Network Working Group
Internet-Draft
Intended status: Informational
Expires: April 23, 2012

L. Ciavattone
AT&T Labs
R. Geib
Deutsche Telekom
A. Morton
AT&T Labs
M. Wieser
University of Applied Sciences
Darmstadt

October 21, 2011

Test Plan and Results for Advancing RFC 2679 on the Standards Track
draft-ietf-ippm-testplan-rfc2679-00

Abstract

This memo proposes to advance a performance metric RFC along the standards track, specifically RFC 2679 on One-way Delay Metrics. Observing that the metric definitions themselves should be the primary focus rather than the implementations of metrics, this memo describes the test procedures to evaluate specific metric requirement clauses to determine if the requirement has been interpreted and implemented as intended. Two completely independent implementations have been tested against the key specifications of RFC 2679.

Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

Status of this Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at http://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on April 23, 2012.

Copyright Notice

Copyright (c) 2011 IETF Trust and the persons identified as the document authors. All rights reserved.

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

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

Table of Contents

1. Introduction 4
  1.1. RFC 2679 Coverage 5
2. A Definition-centric metric advancement process 5
3. Test configuration 6
4. Error Calibration, RFC 2679 10
  4.1. NetProbe Error and Type-P 11
  4.2. Perfas Error and Type-P 13
5. Pre-determined Limits on Equivalence 14
6. Tests to evaluate RFC 2679 Specifications 14
  6.1. One-way Delay, ADK Sample Comparison - Same & Cross Implementation 15
    6.1.1. NetProbe Same-implementation results 16
    6.1.2. Perfas Same-implementation results 17
    6.1.3. One-way Delay, Cross-Implementation ADK Comparison . . 18
    6.1.4. Conclusions on the ADK Results for One-way Delay . . . 18
  6.2. One-way Delay, Loss threshold, RFC 2679 19
    6.2.1. NetProbe results for Loss Threshold 20
    6.2.2. Perfas Results for Loss Threshold 20
    6.2.3. Conclusions for Loss Threshold 20
  6.3. One-way Delay, First-bit to Last bit, RFC 2679 20
    6.3.1. NetProbe and Perfas Results for Serialization 21
    6.3.2. Conclusions for Serialization 22
  6.4. One-way Delay, Difference Sample Metric (Lab) 22
    6.4.1. NetProbe results for Differential Delay 23
    6.4.2. Perfas results for Differential Delay 24
    6.4.3. Conclusions for Differential Delay 24
  6.5. Implementation of Statistics for One-way Delay 24
7. Security Considerations 25
8. IANA Considerations 25
9. Acknowledgements 25
10. References 25
  10.1. Normative References 25
  10.2. Informative References 26
Authors' Addresses 26

1. Introduction

The IETF (IP Performance Metrics working group, IPPM) has considered how to advance their metrics along the standards track since 2001, with the initial publication of Bradner/Paxson/Mankin's memo [ref to work in progress, draft-bradner-metricstest-]. The original proposal was to compare the results of implementations of the metrics, because the usual procedures for advancing protocols did not appear to apply. It was found to be difficult to achieve consensus on exactly how to compare implementations, since there were many legitimate sources of variation that would emerge in the results despite the best attempts to keep the network paths equal, and because considerable variation was allowed in the parameters (and therefore implementation) of each metric. Flexibility in metric definitions, essential for customization and broad appeal, made the comparison task quite difficult.

A renewed work effort sought to investigate ways in which the measurement variability could be reduced and thereby simplify the problem of comparison for equivalence.

There is *preliminary* consensus [I-D.ietf-ippm-metrictest] that the metric definitions should be the primary focus of evaluation rather than the implementations of metrics, and equivalent results are deemed to be evidence that the metric specifications are clear and unambiguous. This is the metric specification equivalent of protocol interoperability. The advancement process either produces confidence that the metric definitions and supporting material are clearly worded and unambiguous, OR, identifies ways in which the metric

definitions should be revised to achieve clarity.

The process should also permit identification of options that were not implemented, so that they can be removed from the advancing specification (this is an aspect more typical of protocol advancement along the standards track).

This memo's purpose is to implement the current approach for [RFC2679]. It was prepared to help progress discussions on the topic of metric advancement, both through e-mail and at the upcoming IPPM meeting at IETF.

In particular, consensus is sought on the extent of tolerable errors when assessing equivalence in the results. In discussions, the IPPM working group agreed that test plan and procedures should include the threshold for determining equivalence, and this information should be available in advance of cross-implementation comparisons. This memo includes procedures for same-implementation comparisons to help set the equivalence threshold.

Another aspect of the metric RFC advancement process is the requirement to document the work and results. The procedures of [RFC2026] are expanded in[RFC5657], including sample implementation and interoperability reports. This memo follows the template in [I-D.morton-ippm-advance-metrics] for the report that accompanies the protocol action request submitted to the Area Director, including description of the test set-up, procedures, results for each implementation and conclusions.

1.1. RFC 2679 Coverage

This plan, in it's first draft version, does not cover all critical requirements and sections of [RFC2679]. Material will be added as it is "discovered" (not all requirements use requirements language).

2. A Definition-centric metric advancement process

The process described in Section 3.5 of [I-D.ietf-ippm-metrictest] takes as a first principle that the metric definitions, embodied in the text of the RFCs, are the objects that require evaluation and possible revision in order to advance to the next step on the

standards track.

IF two implementations do not measure an equivalent singleton or sample, or produce the an equivalent statistic,

AND sources of measurement error do not adequately explain the lack of agreement,

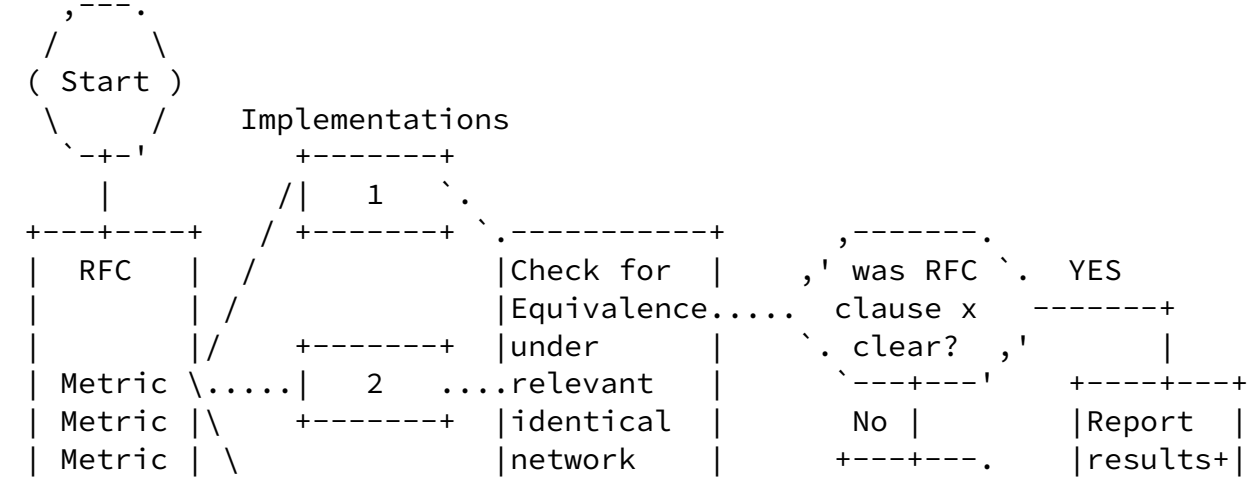
THEN the details of each implementation should be audited along with the exact definition text, to determine if there is a lack of clarity that has caused the implementations to vary in a way that affects the correspondence of the results.

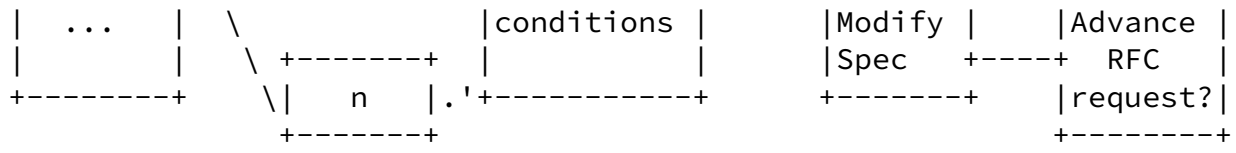
IF there was a lack of clarity or multiple legitimate interpretations of the definition text,

THEN the text should be modified and the resulting memo proposed for consensus and advancement along the standards track.

Finally, all the findings MUST be documented in a report that can support advancement on the standards track, similar to those described in [RFC5657]. The list of measurement devices used in testing satisfies the implementation requirement, while the test results provide information on the quality of each specification in the metric RFC (the surrogate for feature interoperability).

The figure below illustrates this process:



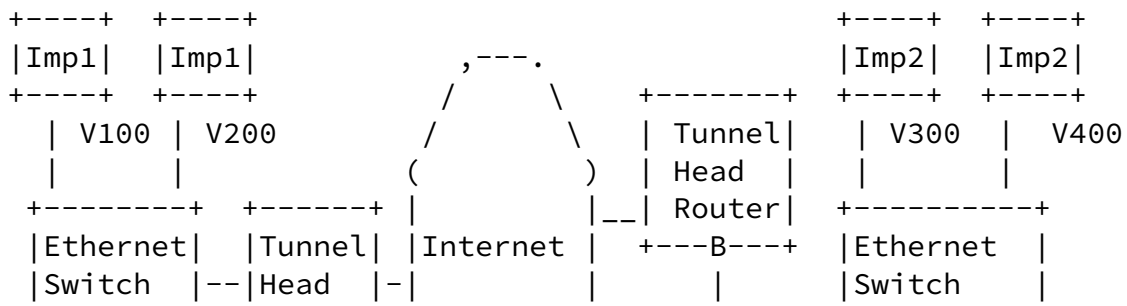


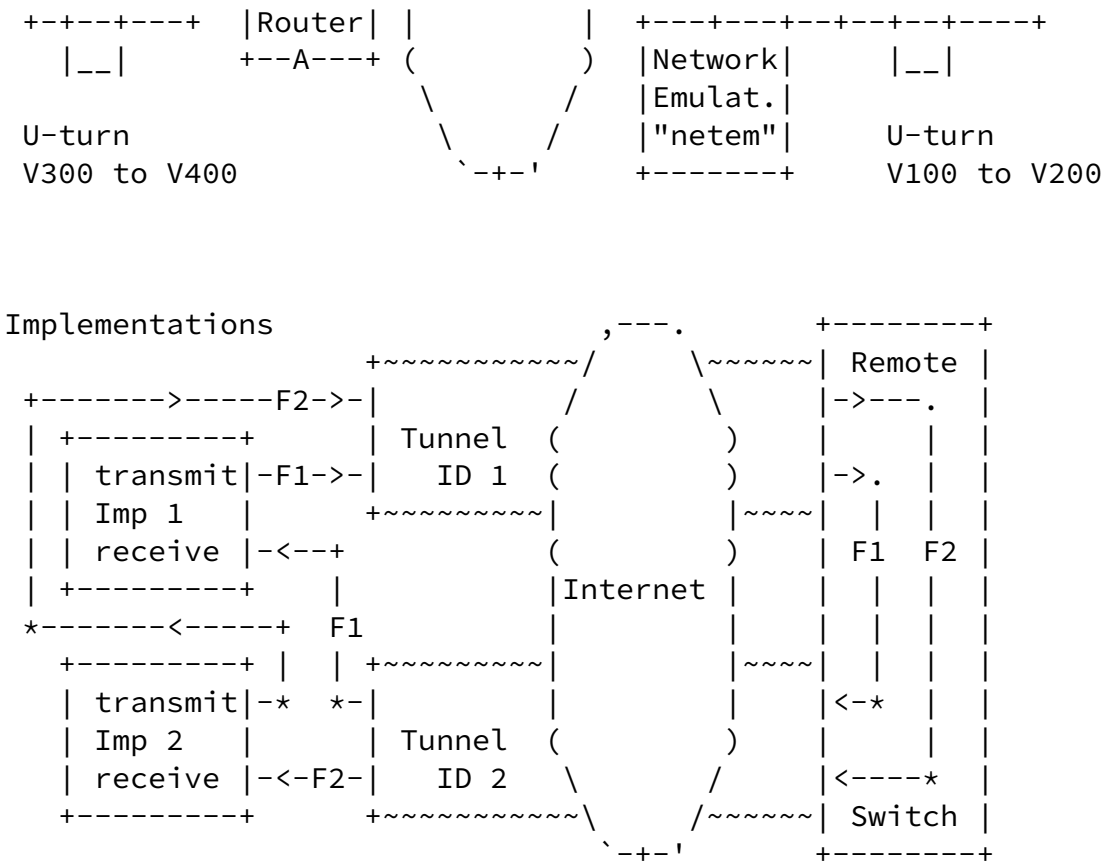
3. Test configuration

One metric implementation used was NetProbe version 5.8.5, (an earlier version is used in the WIPM system and deployed world-wide). NetProbe uses UDP packets of variable size, and can produce test streams with Periodic [RFC3432] or Poisson [RFC2330] sample distributions.

The other metric implementation used was Perfas+ version 3.1, developed by Deutsche Telekom. Perfas+ uses UDP unicast packets of variable size (but supports also TCP and multicast). Test streams with periodic, Poisson or uniform sample distributions may be used.

Figure 2 shows a view of the test path as each Implementation's test flows pass through the Internet and the L2TPv3 tunnel IDs (1 and 2), based on Figure 1 of [I-D.ietf-ippm-metrictest].





Illustrations of a test setup with a bi-directional tunnel. The upper diagram emphasizes the VLAN connectivity and geographical location. The lower diagram shows example flows traveling between two measurement implementations (for simplicity, only two flows are shown).

Figure 1

The testing employs the Layer 2 Tunnel Protocol, version 3 (L2TPv3) [RFC3931] tunnel between test sites on the Internet. The tunnel IP and L2TPv3 headers are intended to conceal the test equipment addresses and ports from hash functions that would tend to spread different test streams across parallel network resources, with likely variation in performance as a result.

At each end of the tunnel, one pair of VLANs encapsulated in the

tunnel are looped-back so that test traffic is returned to each test

site. Thus, test streams traverse the L2TP tunnel twice, but appear to be one-way tests from the test equipment point of view.

The network emulator is a host running Fedora 14 Linux [http://fedoraproject.org/] with IP forwarding enabled and the "netem" Network emulator as part of the Fedora Kernel 2.6.35.11 [http://www.linuxfoundation.org/collaborate/workgroups/networking/netem] loaded and operating. Connectivity across the netem/Fedora host was accomplished by bridging Ethernet VLAN interfaces together with "brctl" commands (e.g., eth1.100 <-> eth2.100). The netem emulator was activated on one interface (eth1) and only operates on test streams traveling in one direction. In some tests, independent netem instances operated separately on each VLAN.

The links between the netem emulator host and router and switch were found to be 100baseTx-HD (100Mbps half duplex) as reported by "mii-tool"when the testing was complete. Use of Half Duplex was not intended, but probably added a small amount of delay variation that could have been avoided in full duplex mode.

Each individual test was run with common packet rates (1 pps, 10pps) Poisson/Periodic distributions, and IP packet sizes of 64, 340, and 500 Bytes.

For these tests, a stream of at least 300 packets were sent from Source to Destination in each implementation. Periodic streams (as per [RFC3432]) with 1 second spacing were used, except as noted.

With the L2TPv3 tunnel in use, the metric name for the testing configured here (with respect to the IP header exposed to Internet processing) is:

Type-IP-protocol-115-One-way-Delay-<StreamType>-Stream

With (Section 4.2. [RFC2679]) Metric Parameters:

+ Src, the IP address of a host (12.3.167.16 or 193.159.144.8)

+ Dst, the IP address of a host (193.159.144.8 or 12.3.167.16)

+ T0, a time

+ Tf, a time

+ lambda, a rate in reciprocal seconds

+ Thresh, a maximum waiting time in seconds (see Section 3.82 of

[RFC2679]) And (Section 4.3. [RFC2679])

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

+ T, a time, and

+ dT, either a 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.

Also, Section 3.8.4 of [RFC2679] recommends that the path SHOULD be reported. In this test set-up, most of the path details will be concealed from the implementations by the L2TPv3 tunnels, thus a more informative path trace route can be conducted by the routers at each location.

When NetProbe is used in production, a traceroute is conducted in parallel with, and at the outset of measurements.

Perfas+ does not support traceroute.

```
 IPLGW#traceroute 193.159.144.8

 Type escape sequence to abort.
 Tracing the route to 193.159.144.8

   1 12.126.218.245 [AS 7018] 0 msec 0 msec 4 msec
   2 cr84.n54ny.ip.att.net (12.123.2.158) [AS 7018] 4 msec 4 msec
     cr83.n54ny.ip.att.net (12.123.2.26) [AS 7018] 4 msec
   3 cr1.n54ny.ip.att.net (12.122.105.49) [AS 7018] 4 msec
     cr2.n54ny.ip.att.net (12.122.115.93) [AS 7018] 0 msec
     cr1.n54ny.ip.att.net (12.122.105.49) [AS 7018] 0 msec
   4 n54ny02jt.ip.att.net (12.122.80.225) [AS 7018] 4 msec 0 msec
     n54ny02jt.ip.att.net (12.122.80.237) [AS 7018] 4 msec
   5 192.205.34.182 [AS 7018] 0 msec
     192.205.34.150 [AS 7018] 0 msec
     192.205.34.182 [AS 7018] 4 msec
   6 da-rg12-i.DA.DE.NET.DTAG.DE (62.154.1.30) [AS 3320] 88 msec 88 msec
 88 msec
   7 217.89.29.62 [AS 3320] 88 msec 88 msec 88 msec
   8 217.89.29.55 [AS 3320] 88 msec 88 msec 88 msec
   9  *  *  *
```

It was only possible to conduct the traceroute for the measured path on one of the tunnel-head routers (the normal trace facilities of the measurement systems are confounded by the L2TPv3 tunnel encapsulation).

4. Error Calibration, RFC 2679

An implementation is required to report on its error calibration in Section 3.8 of [RFC2679] (also required in Section 4.8 for sample metrics). Sections 3.6, 3.7, and 3.8 of [RFC2679] give the detailed formulation of the errors and uncertainties for calibration. In summary, Section 3.7.1 of [RFC2679] describes the total time-varying uncertainty as:

Esynch(t)+ Rsource + Rdest

where:

Esynch(t) denotes an upper bound on the magnitude of clock synchronization uncertainty.

Rsource and Rdest denote the resolution of the source clock and the destination clock, respectively.

Further, Section 3.7.2 of [RFC2679] describes the total wire-time

uncertainty as

Hsource + Hdest

referring to the upper bounds on host-time to wire-time for source and destination, respectively.

Section 3.7.3 of [RFC2679] describes a test with small packets over an isolated minimal network where the results can be used to estimate systematic and random components of the sum of the above errors or uncertainties. In a test with hundreds of singletons, the median is the systematic error and when the median is subtracted from all singletons, the remaining variability is the random error.

The test context, or Type-P of the test packets, must also be reported, as required in Section 3.8 of [RFC2679] and all metrics defined there. Type-P is defined in Section 13 of [RFC2330] (as are many terms used below).

4.1. NetProbe Error and Type-P

Type-P for this test was IP-UDP with Best Effort DCSP. These headers were encapsulated according to the L2TPv3 specifications [RFC3931], and thus may not influence the treatment received as the packets traversed the Internet.

In general, NetProbe error is dependent on the specific version and installation details.

NetProbe operates using host time above the UDP layer, which is different from the wire-time preferred in [RFC2330], but can be identified as a source of error according to Section 3.7.2 of

[RFC2679].

Accuracy of NetProbe measurements is usually limited by NTP synchronization performance (which is typically taken as ~+/-1ms error or greater), although the installation used in this testing often exhibits errors much less than typical for NTP. The primary stratum 1 NTP server is closely located on a sparsely utilized network management LAN, thus it avoids many concerns raised in Section 10 of[RFC2330] (in fact, smooth adjustment, long-term drift analysis and compensation, and infrequent adjustment all lead to stability during measurement intervals, the main concern).

The resolution of the reported results is 1us (us = microsecond) in the version of NetProbe tested here, which contributes to at least +/-1us error.

NetProbe implements a time-keeping sanity check on sending and receiving time-stamping processes. When the significant process interruption takes place, individual test packets are flagged as possibly containing unusual time errors, and are excluded from the sample used for all "time" metrics.

We performed a NetProbe calibration of the type described in Section 3.7.3 of [RFC2679], using 64 Byte packets over a cross-connect cable. The results estimate systematic and random components of the sum of the Hsource + Hdest errors or uncertainties. In a test with 300 singletons conducted over 30 seconds (periodic sample with 100ms spacing), the median is the systematic error and the remaining variability is the random error. One set of results is tabulated below:

(Results from the "R" software environment for statistical computing and graphics - http://www.r-project.org/)

```
> summary(XD4CAL)
      CAL1            CAL2            CAL3
 Min.   : 89.0   Min.   : 68.00   Min.   : 54.00
 1st Qu.: 99.0   1st Qu.: 77.00   1st Qu.: 63.00
 Median :110.0   Median : 79.00   Median : 65.00
 Mean   :116.8   Mean   : 83.74   Mean   : 69.65
 3rd Qu.:127.0   3rd Qu.: 88.00   3rd Qu.: 74.00
 Max.   :205.0   Max.   :177.00   Max.   :163.00
```

>
NetProbe Calibration with Cross-Connect Cable, one-way delay values in microseconds (us)

The median or systematic error can be as high as 110 us, and the range of the random error is also on the order of 116 us for all streams.

Also, anticipating the Anderson-Darling K-sample (ADK) comparisons to follow, we corrected the CAL2 values for the difference between means between CAL2 and CAL3 (as specified in [I-D.ietf-ippm-metrictest]), and found strong support for the (Null Hypothesis that) the samples are from the same distribution (resolution of 1 us and alpha equal 0.05 and 0.01)


```
> XD4CVCAL2 <- XD4CAL$CAL2 - (mean(XD4CAL$CAL2)-mean(XD4CAL$CAL3))
> boxplot(XD4CVCAL2,XD4CAL$CAL3)
> XD4CV2_ADK <- adk.test(XD4CVCAL2, XD4CAL$CAL3)
> XD4CV2_ADK
Anderson-Darling k-sample test.

Number of samples:  2
Sample sizes: 300 300
Total number of values: 600
Number of unique values: 97

Mean of Anderson Darling Criterion: 1
Standard deviation of Anderson Darling Criterion: 0.75896

T = (Anderson Darling Criterion - mean)/sigma

Null Hypothesis: All samples come from a common population.
```

```
                        t.obs P-value extrapolation
not adj. for ties  0.71734 0.17042             0
adj. for ties     -0.39553 0.44589             1
>
```

4.2. Perfas Error and Type-P

Perfas+ is configured to use GPS synchronisation and uses NTP synchronization as a fall-back or default. GPS synchronisation worked throughout this test with the exception of the calibration stated here (one implementation was NTP synchronised only). The time stamp accuracy typically is 0.1 ms.

The resolution of the results reported by Perfas+ is 1us (us = microsecond) in the version tested here, which contributes to at least +/-1us error.

Port	5001	5002	5003
Min.	-227	-226	294
Median	-169	-167	323
Mean	-159	-157	335
Max.	6	-52	376
s	102	102	93

Perfas Calibration with Cross-Connect Cable, one-way delay values in microseconds (us)

The median or systematic error can be as high as 323 us, and the range of the random error is also less than 232 us for all streams.

5. Pre-determined Limits on Equivalence

In this section, we provide the numerical limits on comparisons between implementations, in order to declare that the results are equivalent and therefore, the tested specification is clear.

A key point is that the allowable errors, corrections, and confidence levels only need to be sufficient to detect mis-interpretation of the tested specification resulting in diverging implementations.

Also, the allowable error must be sufficient to compensate for

measured path differences. It was simply not possible to measure fully identical paths in the VLAN-loopback test configuration used, and this practical compromise must be taken into account.

For Anderson-Darling K-sample (ADK) comparisons, the required confidence factor for the cross-implementation comparisons SHALL be the smallest of:

- 0.95 confidence factor at 1ms resolution, or
- the smallest confidence factor (in combination with resolution) of the two same-implementation comparisons for the same test conditions.

A constant time accuracy error of as much as +/-0.5ms MAY be removed from one implementation's distributions (all singletons) before the ADK comparison is conducted.

A constant propagation delay error (due to use of different sub-nets between the switch and measurement devices at each location) of as much as +2ms MAY be removed from one implementation's distributions (all singletons) before the ADK comparison is conducted.

For comparisons involving the mean of a sample or other central statistics, the limits on both the time accuracy error and the propagation delay error constants given above also apply.

6. Tests to evaluate RFC 2679 Specifications

This section describes some results from real-world (cross-Internet) tests with measurement devices implementing IPPM metrics and a network emulator to create relevant conditions, to determine whether the metric definitions were interpreted consistently by implementors.

The procedures are slightly modified from the original procedures contained in Appendix A.1 of [I-D.ietf-ippm-metrictest]. The

modifications include the use of the mean statistic for comparisons.

Note that there are only five instances of the requirement term "MUST" in [RFC2679] outside of the boilerplate and [RFC2119]

reference.

6.1. One-way Delay, ADK Sample Comparison - Same & Cross Implementation

This test determines if implementations produce results that appear to come from a common delay distribution, as an overall evaluation of Section 4 of [RFC2679], "A Definition for Samples of One-way Delay". Same-implementation comparison results help to set the threshold of equivalence that will be applied to cross-implementation comparisons.

This test is intended to evaluate measurements in sections 3 and 4 of [RFC2679].

By testing the extent to which the distributions of one-way delay singletons from two implementations of [RFC2679] appear to be from the same distribution, we economize on comparisons, because comparing a set of individual summary statistics (as defined in Section 5 of [RFC2679]) would require another set of individual evaluations of equivalence. Instead, we can simply check which statistics were implemented, and report on those facts.

1. Configure an L2TPv3 path between test sites, and each pair of measurement devices to operate tests in their designated pair of VLANs.

2. Measure a sample of one-way delay singletons with 2 or more implementations, using identical options and network emulator settings (if used).

3. Measure a sample of one-way delay singletons with *four* instances of the *same* implementations, using identical options, noting that connectivity differences SHOULD be the same as for the cross implementation testing.

4. Apply the ADK comparison procedures (see Appendix C of [I-D.ietf-ippm-metrictest]) and determine the resolution and confidence factor for distribution equivalence of each same-implementation comparison and each cross-implementation comparison.

5. Take the coarsest resolution and confidence factor for distribution equivalence from the same-implementation pairs, or the limit defined in Section 5 above, as a limit on the equivalence threshold for these experimental conditions.

6. Apply constant correction factors to all singletons of the sample distributions, as described and limited in Section 5 above.

7. Compare the cross-implementation ADK performance with the equivalence threshold determined in step 5 to determine if equivalence can be declared.

The common parameters used for tests in this section are:

- IP header + payload = 64 octets
- Periodic sampling at 1 packet per second
- Test duration = 300 seconds (March 29)

The netem emulator was set for 100ms average delay, with uniform delay variation of +/-50ms. In this experiment, the netem emulator was configured to operate independently on each VLAN and thus the emulator itself is a potential source of error when comparing streams that traverse the test path in different directions.

In the result analysis of this section:

- All comparisons used 1 microsecond resolution.
- No Correction Factors were applied.
- The 0.95 confidence factor (1.960 for paired stream comparison) was used.

6.1.1. NetProbe Same-implementation results

A single same-implementation comparison fails the ADK criterion (s1 <-> sB). We note that these streams traversed the test path in opposite directions, making the live network factors a possibility to explain the difference.

All other pair comparisons pass the ADK criterion.

ti.obs (P)	s1	s2	sA
s2	0.25 (0.28)		
sA	0.60 (0.19)	-0.80 (0.57)	
sB	2.64 (0.03)	0.07 (0.31)	-0.52 (0.48)

NetProbe ADK Results for same-implementation

6.1.2. Perfas Same-implementation results

All pair comparisons pass the ADK criterion.

ti.obs (P)	p1	p2	p3
p2	0.06 (0.32)		
p3	1.09 (0.12)	0.37 (0.24)	
p4	-0.81 (0.57)	-0.13 (0.37)	1.36 (0.09)

6.1.3. One-way Delay, Cross-Implementation ADK Comparison

The cross-implementation results are compared using a combined ADK analysis [ref], where all NetProbe results are compared with all Perfas results after testing that the combined same-implementation results pass the ADK criterion.

When 4 (same) samples are compared, the ADK criterion for 0.95 confidence is 1.915, and when all 8 (cross) samples are compared it is 1.85.

```
Combination of Anderson-Darling K-Sample Tests.

Sample sizes within each data set:
Data set 1 :  299 297 298 300 (NetProbe)
Data set 2 :  300 300 298 300 (Perfas)
Total sample size per data set: 1194 1198
Number of unique values per data set: 1188 1192
...
Null Hypothesis:
All samples within a data set come from a common distribution.
The common distribution may change between data sets.

NetProbe           ti.obs P-value extrapolation
not adj. for ties 0.64999 0.21355             0
adj. for ties     0.64833 0.21392             0
Perfas
not adj. for ties 0.55968 0.23442             0
adj. for ties     0.55840 0.23473             0

Combined Anderson-Darling Criterion:
                   tc.obs P-value extrapolation
not adj. for ties 0.85537 0.17967             0
```

adj. for ties 0.85329 0.18010 0

The combined same-implementation samples and the combined cross-implementation comparison all pass the ADK criteria at P>=0.18 and support the Null Hypothesis (both data sets come from a common distribution).

We also see that the paired ADK comparisons are rather critical. Although the NetProbe s1-sB comparison failed, the combined data set from 4 streams passed the ADK criterion easily.

6.1.4. Conclusions on the ADK Results for One-way Delay

Similar testing was repeated many times in the months of March and April 2011. There were many experiments where a single test stream

from NetProbe or Perfas proved to be different from the others in paired comparisons (even same comparisons). When the out lier stream was removed from the comparison, the remaining streams passed combined ADK criterion. Also, the application of correction factors resulted in higher comparison success.

We conclude that the two implementations are capable of producing equivalent one-way delay distributions based on their interpretation of [RFC2679] .

6.2. One-way Delay, Loss threshold, RFC 2679

This test determines if implementations use the same configured maximum waiting time delay from one measurement to another under different delay conditions, and correctly declare packets arriving in excess of the waiting time threshold as lost.

See Section 3.5 of [RFC2679], 3rd bullet point and also Section 3.8.2 of [RFC2679].

1. configure an L2TPv3 path between test sites, and each pair of measurement devices to operate tests in their designated pair of VLANs.

2. configure the network emulator to add 1.0 sec one-way constant delay in one direction of transmission.

3. measure (average) one-way delay with 2 or more implementations, using identical waiting time thresholds (Thresh) for loss set at 3 seconds.

4. configure the network emulator to add 3 sec one-way constant delay in one direction of transmission equivalent to 2 seconds of additional one-way delay (or change the path delay while test is in progress, when there are sufficient packets at the first delay setting)

5. repeat/continue measurements

6. observe that the increase measured in step 5 caused all packets with 2 sec additional delay to be declared lost, and that all packets that arrive successfully in step 3 are assigned a valid one-way delay.

The common parameters used for tests in this section are:

- IP header + payload = 64 octets
- Poisson sampling at lambda = 1 packet per second
- Test duration = 900 seconds total (March 21)

The netem emulator was set to add constant delays as specified in the procedure above.

6.2.1. NetProbe results for Loss Threshold

In NetProbe, the Loss Threshold is implemented uniformly over all packets as a post-processing routine. With the Loss Threshold set at 3 seconds, all packets with one-way delay >3 seconds are marked "Lost" and included in the Lost Packet list with their transmission time (as required in Section 3.3 of [RFC2680]). This resulted in 342 packets designated as lost in one of the test streams (with average delay = 3.091 sec).

6.2.2. Perfas Results for Loss Threshold

Perfas uses a fixed Loss Threshold which was not adjustable during this study. The Loss Threshold is approximately one minute, and emulation of a delay of this size was not attempted. However, it is possible to implement any delay threshold desired with a post-processing routine and subsequent analysis. Using this method, 195 packets would be declared lost (with average delay = 3.091 sec).

6.2.3. Conclusions for Loss Threshold

Both implementations assume that any constant delay value desired can be used as the Loss Threshold, since all delays are stored as a pair <Time, Delay> as required in [RFC2679] . This is a simple way to enforce the constant loss threshold envisioned in [RFC2679] (see specific section references above). We take the position that the assumption of post-processing is compliant, and that the text of the RFC should be revised slightly to include this point.

6.3. One-way Delay, First-bit to Last bit, RFC 2679

This test determines if implementations register the same relative change in delay from one packet size to another, indicating that the first-to-last time-stamping convention has been followed. This test tends to cancel the sources of error which may be present in an implementation.

See Section 3.7.2 of [RFC2679], and Section 10.2 of [RFC2330].

1. configure an L2TPv3 path between test sites, and each pair of measurement devices to operate tests in their designated pair of VLANs, and ideally including a low-speed link (it was not possible to change the link configuration during testing, so the lowest speed link present was the basis for serialization time comparisons).

2. measure (average) one-way delay with 2 or more implementations, using identical options and equal size small packets (64 octet IP header and payload)

3. maintain the same path with additional emulated 100 ms one-way

delay

4. measure (average) one-way delay with 2 or more implementations, using identical options and equal size large packets (500 octet IP header and payload)

5. observe that the increase measured between steps 2 and 4 is equivalent to the increase in ms expected due to the larger serialization time for each implementation. Most of the measurement errors in each system should cancel, if they are stationary.

The common parameters used for tests in this section are:

- IP header + payload = 64 octets
- Periodic sampling at l packet per second
- Test duration = 300 seconds total (April 12)

The netem emulator was set to add constant 100ms delay.

6.3.1. NetProbe and Perfas Results for Serialization

When the IP header + payload size was increased from 64 octets to 500 octets, there was a delay increase observed.

Mean Delays in us
NetProbe

Payload	s1	s2	sA	sB
500	190893	191179	190892	190971
64	189642	189785	189747	189467

```
Diff     1251    1394    1145    1505

Perfas
Payload    p1      p2      p3      p4
500    190908  190911  191126  190709
 64    189706  189752  189763  190220
Diff     1202    1159    1363     489
```

Serialization tests, all values in microseconds

The typical delay increase when the larger packets were used was 1.1 to 1.5 ms (with one outlier). The typical measurements indicate that a link with approximately 3 Mbit/s capacity is present on the path.

Through investigation of the facilities involved, it was determined that the lowest speed link was approximately 45 Mbit/s, and therefore the estimated difference should be about 0.077 ms. The observed differences are much higher.

The unexpected large delay difference was also the outcome when testing serialization times in a lab environment, using the NIST Net Emulator and NetProbe [ref to earlier lab tests].

6.3.2. Conclusions for Serialization

Since it was not possible to confirm the estimated serialization time increases in field tests, we resort to examination of the implementations to determine compliance.

NetProbe performs all time stamping above the IP-layer, accepting that some compromises must be made to achieve extreme portability and measurement scale. Therefore, the first-to-last bit convention is supported because the serialization time is included in the one-way delay measurement, enabling comparison with other implementations.

Perfas >>>>>>>>>>>>>>> TBD

6.4. One-way Delay, Difference Sample Metric (Lab)

This test determines if implementations register the same relative increase in delay from one measurement to another under different delay conditions. This test tends to cancel the sources of error which may be present in an implementation.

This test is intended to evaluate measurements in sections 3 and 4 of [RFC2679].

1. configure an L2TPv3 path between test sites, and each pair of measurement devices to operate tests in their designated pair of VLANs.

2. measure (average) one-way delay with 2 or more implementations, using identical options

3. configure the path with X+Y ms one-way delay

4. repeat measurements

5. observe that the (average) increase measured in steps 2 and 4 is ~Y ms for each implementation. Most of the measurement errors in each system should cancel, if they are stationary.

In this test, X=1000ms and Y=1000ms.

The common parameters used for tests in this section are:

o IP header + payload = 64 octets

o Poisson sampling at lambda = 1 packet per second

o Test duration = 900 seconds total (March 21)

The netem emulator was set to add constant delays as specified in the procedure above.

6.4.1. NetProbe results for Differential Delay

Average pre-increase delay, microseconds	1089868.0
Average post 1s additional, microseconds	2089686.0
Difference (should be ~= Y = 1s)	999818.0

Average delays before/after 1 second increase

The NetProbe implementation observed a 1 second increase with a 182 microsecond error (assuming that the netem emulated delay difference is exact).

We note that this differential delay test has been run under lab conditions and published in prior work [ref to "advance metrics" draft]. The error was 6 microseconds.

6.4.2. Perfas results for Differential Delay

Average pre-increase delay, microseconds	1089794.0
Average post 1s additional, microseconds	2089801.0
Difference (should be ~= Y = 1s)	1000007.0

Average delays before/after 1 second increase

The Perfas implementation observed a 1 second increase with a 7 microsecond error.

6.4.3. Conclusions for Differential Delay

Again, the live network conditions appear to have influenced the results, but both implementations measured the same delay increase within their calibration accuracy.

6.5. Implementation of Statistics for One-way Delay

The ADK tests the extent to which the sample distributions of one-way delay singletons from two implementations of [RFC2679] appear to be from the same overall distribution. By testing this way, we economize on the number of comparisons, because comparing a set of individual summary statistics (as defined in Section 5 of [RFC2679]) would require another set of individual evaluations of equivalence. Instead, we can simply check which statistics were implemented, and report on those facts, noting that Section 5 of [RFC2679] does not specify the calculations exactly, and gives only some illustrative examples.

	NetProbe	Perfas
5.1. Type-P-One-way-Delay-Percentile	yes	no
5.2. Type-P-One-way-Delay-Median	yes	no
5.3. Type-P-One-way-Delay-Minimum	yes	yes
5.4. Type-P-One-way-Delay-Inverse-Percentile	no	no

Implementation of Section 5 Statistics

5.1. Type-P-One-way-Delay-Percentile 5.2. Type-P-One-way-Delay-Median 5.3. Type-P-One-way-Delay-Minimum 5.4. Type-P-One-way-Delay-Inverse-Percentile

7. Security Considerations

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

8. IANA Considerations

This memo makes no requests of IANA, and hopes that IANA will be as accepting of our new computer overlords as the authors intend to be.

9. Acknowledgements

The authors thank Lars Eggert for his continued encouragement to advance the IPPM metrics during his tenure as AD Advisor.

Nicole Kowalski supplied the needed CPE router for the NetProbe side of the test set-up, and graciously managed her testing in spite of issues caused by dual-use of the router. Thanks Nicole!

The "NetProbe Team" also acknowledges many useful discussions with Ganga Maguluri.

10. References

10.1. Normative References

[I-D.ietf-ippm-metrictest]
Geib, R., Morton, A., Fardid, R., and A. Steinmitz, "IPPM standard advancement testing", draft-ietf-ippm-metrictest-03 (work in progress), June 2011.

[RFC2026] Bradner, S., "The Internet Standards Process -- Revision 3", BCP 9, RFC 2026, October 1996.

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

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

[RFC2679] Almes, G., Kalidindi, S., and M. Zekauskas, "A One-way Delay Metric for IPPM", RFC 2679, September 1999.

[RFC2680] Almes, G., Kalidindi, S., and M. Zekauskas, "A One-way Packet Loss Metric for IPPM", RFC 2680, September 1999.

[RFC3432] Raisanen, V., Grotefeld, G., and A. Morton, "Network performance measurement with periodic streams", RFC 3432, November 2002.

[RFC4656] Shalunov, S., Teitelbaum, B., Karp, A., Boote, J., and M. Zekauskas, "A One-way Active Measurement Protocol (OWAMP)", RFC 4656, September 2006.

[RFC4814] Newman, D. and T. Player, "Hash and Stuffing: Overlooked Factors in Network Device Benchmarking", RFC 4814, March 2007.

[RFC5226] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 5226, May 2008.

[RFC5357] Hedayat, K., Krzanowski, R., Morton, A., Yum, K., and J. Babiarz, "A Two-Way Active Measurement Protocol (TWAMP)", RFC 5357, October 2008.

[RFC5657] Dusseault, L. and R. Sparks, "Guidance on Interoperation and Implementation Reports for Advancement to Draft Standard", BCP 9, RFC 5657, September 2009.

10.2. Informative References

[I-D.morton-ippm-advance-metrics]
Morton, A., "Lab Test Results for Advancing Metrics on the Standards Track", draft-morton-ippm-advance-metrics-02 (work in progress), October 2010.

[RFC3931] Lau, J., Townsley, M., and I. Goyret, "Layer Two Tunneling Protocol - Version 3 (L2TPv3)", RFC 3931, March 2005.

Authors' Addresses

Len Ciavattone
AT&T Labs
200 Laurel Avenue South
Middletown, NJ 07748
USA

Phone: +1 732 420 1239
Fax:
Email: lencia@att.com
URI:

Ruediger Geib
Deutsche Telekom
Heinrich Hertz Str. 3-7
Darmstadt, 64295
Germany

Phone: +49 6151 58 12747

Email: Ruediger.Geib@telekom.de

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

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

Matthias Wieser
University of Applied Sciences Darmstadt
Birkenweg 8 Department EIT
Darmstadt,   64295
Germany

Phone:
Email: matthias.wieser@stud.h-da.de