

Internet Engineering Task Force (IETF)
Request for Comments: 7290
Category: Informational
ISSN: 2070-1721

L. Ciavattone
AT&T Labs
R. Geib
Deutsche Telekom
A. Morton
AT&T Labs
M. Wieser
Technical University Darmstadt
July 2014

Test Plan and Results for Advancing RFC 2680 on the Standards Track

Abstract

This memo provides the supporting test plan and results to advance RFC 2680, a performance metric RFC defining one-way packet loss metrics, along the Standards Track. 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 2680.

Status of This Memo

This document is not an Internet Standards Track specification; it is published for informational purposes.

This document is a product of the Internet Engineering Task Force (IETF). It represents the consensus of the IETF community. It has received public review and has been approved for publication by the Internet Engineering Steering Group (IESG). Not all documents approved by the IESG are a candidate for any level of Internet Standard; see Section 2 of RFC 5741.

Information about the current status of this document, any errata, and how to provide feedback on it may be obtained at <http://www.rfc-editor.org/info/rfc7290>.

Copyright Notice

Copyright (c) 2014 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	3
1.1. Requirements Language	4
1.2. RFC 2680 Coverage	5
2. A Definition-Centric Metric Advancement Process	5
3. Test Configuration	5
4. Error Calibration and RFC 2680	9
4.1. Clock Synchronization Calibration	9
4.2. Packet Loss Determination Error	10
5. Predetermined Limits on Equivalence	10
6. Tests to Evaluate RFC 2680 Specifications	11
6.1. One-Way Loss: ADK Sample Comparison	11
6.1.1. 340B/Periodic Cross-Implementation Results	12
6.1.2. 64B/Periodic Cross-Implementation Results	14
6.1.3. 64B/Poisson Cross-Implementation Results	15
6.1.4. Conclusions on the ADK Results for One-Way Packet Loss	16
6.2. One-Way Loss: Delay Threshold	16
6.2.1. NetProbe Results for Loss Threshold	17
6.2.2. Perfas+ Results for Loss Threshold	17
6.2.3. Conclusions for Loss Threshold	17
6.3. One-Way Loss with Out-of-Order Arrival	17
6.4. Poisson Sending Process Evaluation	19
6.4.1. NetProbe Results	19
6.4.2. Perfas+ Results	20
6.4.3. Conclusions for Goodness-of-Fit	22
6.5. Implementation of Statistics for One-Way Loss	23
7. Conclusions for a Revision of RFC 2680	23
8. Security Considerations	24
9. Acknowledgements	24
10. Appendix - Network Configuration and Sample Commands	25
11. References	28
11.1. Normative References	28
11.2. Informative References	29

1. Introduction

The IETF IP Performance Metrics (IPPM) working group has considered how to advance their metrics along the Standards Track since 2001.

The renewed work effort sought to investigate ways in which the measurement variability could be reduced in order to thereby simplify the problem of comparison for equivalence. As a result, there is consensus (captured in [RFC6576]) that equivalent results from independent implementations of metric specifications are sufficient evidence that the specifications themselves are clear and unambiguous; it is the parallel concept of protocol interoperability

for metric specifications. The advancement process either (1) produces confidence that the metric definitions and supporting material are clearly worded and unambiguous or (2) identifies ways in which the metric definitions should be revised to achieve clarity. It is a non-goal to compare the specific implementations themselves.

The process also permits identification of options described in the metric RFC 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 [RFC2680] and document the results.

In particular, this memo documents consensus on the extent of tolerable errors when assessing equivalence in the results. In discussions, the IPPM working group agreed that the 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 [RFC6808] for the report that accompanies the protocol action request submitted to the Area Director, including a description of the test setup, procedures, results for each implementation, and conclusions.

The conclusion reached is that [RFC2680], with modifications, should be advanced on the Standards Track. The revised text of RFC 2680 [LOSS-METRIC] is ready for review but awaits work in progress to update the IPPM Framework [RFC2330]. Therefore, this memo documents the information to support the advancement of [RFC2680], and the approval of a revision of RFC 2680 is left for future action.

1.1. 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]. Some of these key words were used in [RFC2680], but there are no requirements specified in this memo.

1.2. RFC 2680 Coverage

This plan is intended to cover all critical requirements and sections of [RFC2680].

Note that there are only five relevant instances of the requirement term "MUST" in [RFC2680], outside of the boilerplate and [RFC2119] reference; the instance of "MUST" in the Security Considerations section of [RFC2680] is not a basis for implementation equivalence comparisons.

Statements in RFC 2680 that have the character of requirements may be included if the community reaches consensus that the wording implies a requirement. At least one instance of an implied requirement has been found in Section 3.6 of [RFC2680].

2. A Definition-Centric Metric Advancement Process

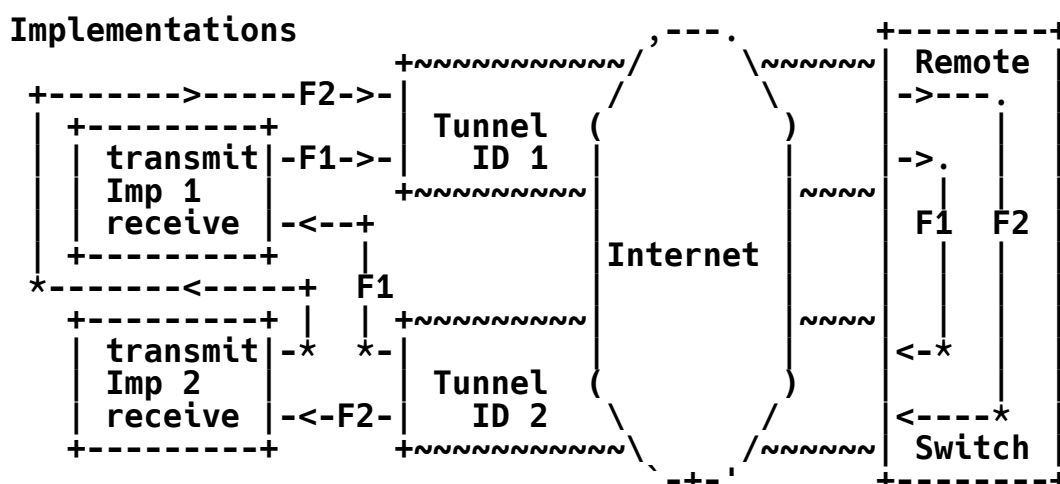
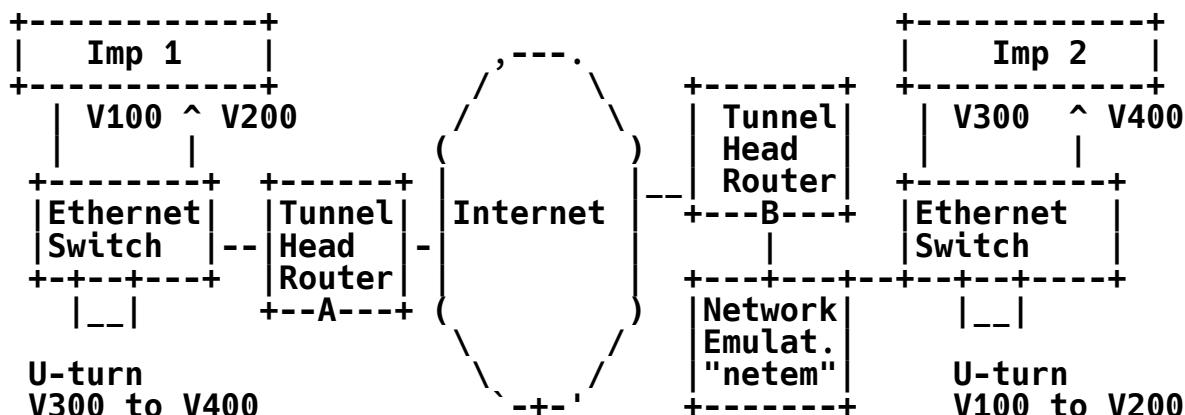
The process described in Section 3.5 of [RFC6576] takes as a first principle that the metric definitions, embodied in the text of the RFCs, are the objects that require evaluation and possible revision in order to advance to the next step on the Standards Track. This memo follows that process.

3. Test Configuration

One metric implementation used was NetProbe version 5.8.5 (an earlier version is used in the WIPM system and deployed worldwide [WIPM]). 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]. Perfas+ uses UDP unicast packets of variable size (but also supports TCP and multicast). Test streams with Periodic, Poisson, or uniform sample distributions may be used.

Figure 1 shows a view of the test path as each implementation's test flows pass through the Internet and the Layer 2 Tunneling Protocol version 3 (L2TPv3) [RFC3931] tunnel IDs (1 and 2), based on Figure 1 of [RFC6576].



Illustrations of a test setup with a bidirectional tunnel. The upper diagram emphasizes the VLAN connectivity and geographical location (where "Imp #" is the sender and receiver of implementation 1 or 2 -- either Perfast+ or NetProbe in this test). The lower diagram shows example flows traveling between two measurement implementations. For simplicity, only two flows are shown, and the netem emulator is omitted (it would appear before or after the Internet, depending on the flow).

Figure 1

The testing employs the 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 point of view of the test equipment.

The network emulator is a host running Fedora 14 Linux [FEDORA], with IP forwarding enabled and the "netem" Network emulator as part of the Fedora Kernel 2.6.35.11 [NETEM] loaded and operating. The standard kernel is "tickless", replacing the previous periodic timer (250 Hz, with 4 ms uncertainty) interrupts with on-demand interrupts. 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 operated on test streams traveling in one direction. In some tests, independent netem instances operated separately on each VLAN. See the Appendix for more details.

The links between the netem emulator host, the router, and the switch were found to be 100BaseTX-HD (100 Mbps half duplex), as reported by "mii-tool" [MII-TOOL] when testing was complete. The 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, 10 pps) Poisson/Periodic distributions, and IP packet sizes of 64, 340, and 500 bytes.

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

As required in Section 2.8.1 of [RFC2680], packet Type-P must be reported. The packet Type-P for this test was IP-UDP with Best Effort Differentiated Services Code Point (DSCP). These headers were encapsulated according to the L2TPv3 specification [RFC3931] and were unlikely to influence the treatment received as the packets traversed the Internet.

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-Packet-Loss-<StreamType>-Stream

With (Section 3.2 of [RFC2680]) 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 2.8.2 of [RFC2680])

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

- + T, a time, and
- + L, either a zero or a one

The values of T in the sequence are monotonically increasing. Note that T would be a valid parameter of *singleton* Type-P-One-way-Packet-Loss and that L would be a valid value of Type-P-One-way-Packet-Loss (see Section 3.3 of [RFC2680]).

Also, Section 2.8.4 of [RFC2680] recommends that the path SHOULD be reported. In this test setup, most of the path details will be concealed from the implementations by the L2TPv3 tunnels; thus, a more informative path traceroute can be conducted by the routers at each location.

When NetProbe is used in production, a traceroute is conducted in parallel 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 * * *
```

NetProbe Traceroute

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 and RFC 2680

An implementation is required to report calibration results on clock synchronization per Section 2.8.3 of [RFC2680] (also required in Section 3.7 of [RFC2680] for sample metrics).

Also, it is recommended to report the probability that a packet successfully arriving at the destination network interface is incorrectly designated as lost due to resource exhaustion in Section 2.8.3 of [RFC2680].

4.1. Clock Synchronization Calibration

For NetProbe and Perfas+ clock synchronization test results, refer to Section 4 of [RFC6808].

4.2. Packet Loss Determination Error

Since both measurement implementations have resource limitations, it is theoretically possible that these limits could be exceeded and a packet that arrived at the destination successfully might be discarded in error.

In previous test efforts [ADV-METRICS], NetProbe produced six multicast streams with an aggregate bit rate over 53 Mbit/s, in order to characterize the one-way capacity of an emulator based on NIST Net. Neither the emulator nor the pair of NetProbe implementations used in this testing dropped any packets in these streams.

The maximum load used here between any two NetProbe implementations was 11.5 Mbit/s divided equally among three unicast test streams. We concluded that steady resource usage does not contribute error (additional loss) to the measurements.

5. Predetermined 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 that the tested specification is therefore clear.

A key point is that the allowable errors, corrections, and confidence levels only need to be sufficient to detect any misinterpretation of the tested specification that would indicate 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) [ADK] comparisons, the required confidence factor for the cross-implementation comparisons SHALL be the smallest of:

- o 0.95 confidence factor at 1-packet resolution, or
- o the smallest confidence factor (in combination with resolution) of the two same-implementation comparisons for the same test conditions (if the number of streams is sufficient to allow such comparisons).

For Anderson-Darling Goodness-of-Fit (ADGoF) [RADGOF] comparisons, the required level of significance for the same-implementation Goodness-of-Fit (GoF) SHALL be 0.05 or 5%, as specified in Section 11.4 of [RFC2330]. This is equivalent to a 95% confidence factor.

6. Tests to Evaluate RFC 2680 Specifications

This section describes some results from production network (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 similar to those contained in Appendix A.1 of [RFC6576] for one-way delay.

6.1. One-Way Loss: ADK Sample Comparison

This test determines if implementations produce results that appear to come from a common packet loss distribution, as an overall evaluation of Section 3 of [RFC2680] ("A Definition for Samples of One-way Packet Loss"). 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 2, 3, and 4 of [RFC2680].

By testing the extent to which the counts of one-way packet loss on different test streams of two [RFC2680] implementations appear to be from the same loss process, we reduce comparison steps because comparing the resulting summary statistics (as defined in Section 4 of [RFC2680]) would require a redundant set of equivalence evaluations. We can easily check whether the single statistic in Section 4 of [RFC2680] was implemented and report on that fact.

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 packet loss singletons with two or more implementations, using identical options and network emulator settings (if used).

3. Measure a sample of one-way packet loss singletons with *four or more* instances of the *same* implementations, using identical options, noting that connectivity differences SHOULD be the same as for cross-implementation testing.
4. If less than ten test streams are available, skip to step 7.
5. Apply the ADK comparison procedures (see Appendix B of [RFC6576]), and determine the resolution and confidence factor for distribution equivalence of each same-implementation comparison and each cross-implementation comparison.
6. 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.
7. Compare the cross-implementation ADK performance with the equivalence threshold determined in step 5 to determine if equivalence can be declared.

The metric parameters varied for each loss test, and they are listed first in each sub-section below.

The cross-implementation comparison uses a simple ADK analysis [RT00L] [RADK], where all NetProbe loss counts are compared with all Perfast+ loss results.

In the results analysis of this section:

- o All comparisons used 1-packet resolution.
- o No correction factors were applied.
- o The 0.95 confidence factor (and ADK criterion for $t_{obs} < 1.960$ for cross-implementation comparison) was used.

6.1.1. 340B/Periodic Cross-Implementation Results

Tests described in this section used:

- o IP header + payload = 340 octets
- o Periodic sampling at 1 packet per second
- o Test duration = 1200 seconds (during April 7, 2011, EDT)

The netem emulator was set for 100 ms constant delay, with a 10% loss ratio. In this experiment, the netem emulator was configured to operate independently on each VLAN; thus, the emulator itself is a potential source of error when comparing streams that traverse the test path in different directions.

=====

```
A07bps_loss <- c(114, 175, 138, 142, 181, 105) (NetProbe)
A07per_loss <- c(115, 128, 136, 127, 139, 138) (Perfas+)
```

```
> A07bps_loss <- c(114, 175, 138, 142, 181, 105)
> A07per_loss <- c(115, 128, 136, 127, 139, 138)
>
> A07cross_loss_ADK <- adk.test(A07bps_loss, A07per_loss)
> A07cross_loss_ADK
Anderson-Darling k-sample test.
```

```
Number of samples: 2
Sample sizes: 6 6
Total number of values: 12
Number of unique values: 11
```

```
Mean of Anderson Darling Criterion: 1
Standard deviation of Anderson Darling Criterion: 0.6569
```

```
T = (Anderson Darling Criterion - mean)/sigma
```

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

	t.obs	P-value	extrapolation
not adj. for ties	0.52043	0.20604	0
adj. for ties	0.62679	0.18607	0

```
>
```

=====

The cross-implementation comparisons pass the ADK criterion (t.obs < 1.960).

6.1.2. 64B/Periodic Cross-Implementation Results

Tests described in this section used:

- o IP header + payload = 64 octets
- o Periodic sampling at 1 packet per second
- o Test duration = 300 seconds (during March 24, 2011, EDT)

The netem emulator was set for 0 ms constant delay, with a 10% loss ratio.

=====

```
> M24per_loss <- c(42,34,35,35)      (Perfas+)
> M24apd_23BC_loss <- c(27,39,29,24) (NetProbe)
> M24apd_loss23BC_ADK <- adk.test(M24apd_23BC_loss,M24per_loss)
> M24apd_loss23BC_ADK
Anderson-Darling k-sample test.
```

```
Number of samples: 2
Sample sizes: 4 4
Total number of values: 8
Number of unique values: 7
```

```
Mean of Anderson Darling Criterion: 1
Standard deviation of Anderson Darling Criterion: 0.60978
```

```
T = (Anderson Darling Criterion - mean)/sigma
```

Null Hypothesis: All samples come from a common population.

	t.obs	P-value	extrapolation
not adj. for ties	0.76921	0.16200	0
adj. for ties	0.90935	0.14113	0

```
Warning: At least one sample size is less than 5.
p-values may not be very accurate.
```

```
>
```

=====

The cross-implementation comparisons pass the ADK criterion.

6.1.3. 64B/Poisson Cross-Implementation Results

Tests described in this section used:

- o IP header + payload = 64 octets
- o Poisson sampling at $\lambda = 1$ packet per second
- o Test duration = 1200 seconds (during April 27, 2011, EDT)

The netem configuration was 0 ms delay and 10% loss, but there were two passes through an emulator for each stream, and loss emulation was present for 18 minutes of the 20-minute (1200-second) test.

=====

```
A27aps_loss <- c(91,110,113,102,111,109,112,113) (NetProbe)
A27per_loss <- c(95,123,126,114) (Perfas+)
```

```
A27cross_loss_ADK <- adk.test(A27aps_loss, A27per_loss)
```

```
> A27cross_loss_ADK
Anderson-Darling k-sample test.
```

```
Number of samples: 2
Sample sizes: 8 4
Total number of values: 12
Number of unique values: 11
```

```
Mean of Anderson Darling Criterion: 1
Standard deviation of Anderson Darling Criterion: 0.65642
```

```
T = (Anderson Darling Criterion - mean)/sigma
```

Null Hypothesis: All samples come from a common population.

	t.obs	P-value	extrapolation
not adj. for ties	2.15099	0.04145	0
adj. for ties	1.93129	0.05125	0

```
Warning: At least one sample size is less than 5.
p-values may not be very accurate.
```

```
>
```

=====

The cross-implementation comparisons barely pass the ADK criterion at 95% = 1.960 when adjusting for ties.

6.1.4. Conclusions on the ADK Results for One-Way Packet Loss

We conclude that the two implementations are capable of producing equivalent one-way packet loss measurements based on their interpretation of [RFC2680].

6.2. One-Way Loss: Delay Threshold

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 2.8.2 of [RFC2680].

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 second of one-way constant delay in one direction of transmission.
3. Measure (average) one-way delay with two or more implementations, using identical waiting time thresholds (Thresh) for loss set at 3 seconds.
4. Configure the network emulator to add 3 seconds of one-way constant delay in one direction of transmission equivalent to 2 seconds of additional one-way delay (or change the path delay while the 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 seconds of 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:

- o IP header + payload = 64 octets
- o Poisson sampling at $\lambda = 1$ packet per second
- o Test duration = 900 seconds total (March 21, 2011 EDT)

The netem emulator settings added constant delays as specified in the procedure above.

6.2.1. NetProbe Results for Loss Threshold

In NetProbe, the loss threshold was 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 were 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 [RFC2680]. This is a simple way to enforce the constant loss threshold envisioned in [RFC2680] (see Section 2.8.2 of [RFC2680]). We take the position that the assumption of post-processing is compliant and that the text of the revision of RFC 2680 should be revised slightly to include this point.

6.3. One-Way Loss with Out-of-Order Arrival

Section 3.6 of [RFC2680] indicates, with a lowercase "must" in the text, that implementations need to ensure that reordered packets are handled correctly. In essence, this is an implied requirement because the correct packet must be identified as lost if it fails to arrive before its delay threshold under all circumstances, and reordering is always a possibility on IP network paths. See [RFC4737] for the definition of reordering used in IETF standard-compliant measurements.

The netem emulator can produce packet reordering because each packet's delay is drawn from an independent distribution. Here, significant delay (2000 ms) and delay variation (1000 ms) were

sufficient to produce packet reordering. Using the procedure described in Section 6.1, the netem emulator was set to introduce 10% loss while reordering was present.

The tests described in this section used:

- o IP header + payload = 64 octets
- o Periodic sampling = 1 packet per second
- o Test duration = 600 seconds (during May 2, 2011, EDT)

=====

```
> Y02aps_loss <- c(53,45,67,55)      (NetProbe)
> Y02per_loss <- c(59,62,67,69)      (Perfas+)
> Y02cross_loss_ADK <- adk.test(Y02aps_loss, Y02per_loss)
> Y02cross_loss_ADK
Anderson-Darling k-sample test.
```

```
Number of samples: 2
Sample sizes: 4 4
Total number of values: 8
Number of unique values: 7
```

```
Mean of Anderson Darling Criterion: 1
Standard deviation of Anderson Darling Criterion: 0.60978
```

```
T = (Anderson Darling Criterion - mean)/sigma
```

Null Hypothesis: All samples come from a common population.

	t.obs	P-value	extrapolation
not adj. for ties	1.11282	0.11531	0
adj. for ties	1.19571	0.10616	0

```
Warning: At least one sample size is less than 5.
p-values may not be very accurate.
```

```
>
```

=====

The test results indicate that extensive reordering was present. Both implementations capture the extensive delay variation between adjacent packets. In NetProbe, packet arrival order is preserved in the raw measurement files, so an examination of arrival packet sequence numbers also reveals reordering.

Despite extensive continuous packet reordering present in the transmission path, the distributions of loss counts from the two implementations pass the ADK criterion at 95% = 1.960.

6.4. Poisson Sending Process Evaluation

Section 3.7 of [RFC2680] indicates that implementations need to ensure that their sending process is reasonably close to a classic Poisson distribution when used. Much more detail on sample distribution generation and Goodness-of-Fit testing is specified in Section 11.4 of [RFC2330] and the Appendix of [RFC2330].

In this section, each implementation's Poisson distribution is compared with an idealistic version of the distribution available in the base functionality of the R-tool for Statistical Analysis [RT00L] and performed using the Anderson-Darling Goodness-of-Fit test package (ADGofTest) [RADGOF]. The Goodness-of-Fit criterion derived from [RFC2330] requires a test statistic value $AD \leq 2.492$ for 5% significance. The Appendix of [RFC2330] also notes that there may be difficulty satisfying the ADGofTest when the sample includes many packets (when 8192 were used, the test always failed, but smaller sets of the stream passed).

Both implementations were configured to produce Poisson distributions with $\lambda = 1$ packet per second and to assign received packet timestamps in the measurement application (above the UDP layer; see the calibration results in Section 4 of [RFC6808] for error assessment).

6.4.1. NetProbe Results

Section 11.4 of [RFC2330] suggests three possible measurement points to evaluate the Poisson distribution. The NetProbe analysis uses "user-level timestamps made just before or after the system call for transmitting the packet".

The statistical summary for two NetProbe streams is below:

```
=====
> summary(a27ms$s1[2:1152])
  Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
0.0100  0.2900  0.6600  0.9846  1.3800  8.6390
> summary(a27ms$s2[2:1152])
  Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
0.010  0.280   0.670   0.979   1.365   8.829
=====
```

We see that both of the means are near the specified $\lambda = 1$.

The results of ADGoF tests for these two streams are shown below:

=====

```
> ad.test( a27ms$s1[2:101], pexp, 1)
```

Anderson-Darling GoF Test

data: a27ms\$s1[2:101] and pexp

AD = 0.8908, p-value = 0.4197

alternative hypothesis: NA

```
> ad.test( a27ms$s1[2:1001], pexp, 1)
```

Anderson-Darling GoF Test

data: a27ms\$s1[2:1001] and pexp

AD = 0.9284, p-value = 0.3971

alternative hypothesis: NA

```
> ad.test( a27ms$s2[2:101], pexp, 1)
```

Anderson-Darling GoF Test

data: a27ms\$s2[2:101] and pexp

AD = 0.3597, p-value = 0.8873

alternative hypothesis: NA

```
> ad.test( a27ms$s2[2:1001], pexp, 1)
```

Anderson-Darling GoF Test

data: a27ms\$s2[2:1001] and pexp

AD = 0.6913, p-value = 0.5661

alternative hypothesis: NA

=====

We see that both sets of 100 packets and 1000 packets from two different streams (s1 and s2) all passed the $AD \leq 2.492$ criterion.

6.4.2. Perfas+ Results

Section 11.4 of [RFC2330] suggests three possible measurement points to evaluate the Poisson distribution. The Perfas+ analysis uses "wire times for the packets as recorded using a packet filter".

However, due to limited access at the Perfas+ side of the test setup, the captures were made after the Perfas+ streams traversed the production network, adding a small amount of unwanted delay variation to the wire times (and possibly error due to packet loss).

The statistical summary for two Perfas+ streams is below:

```
=====
> summary(a27pe$p1)
  Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
 0.004  0.347   0.788   1.054   1.548   4.231
> summary(a27pe$p2)
  Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
0.0010 0.2710  0.7080  0.9696  1.3740  7.1160
=====
```

We see that both of the means are near the specified $\lambda = 1$.

The results of ADGoF tests for these two streams are shown below:

```
=====
> ad.test(a27pe$p1, pexp, 1 )

      Anderson-Darling GoF Test

data:  a27pe$p1 and pexp
AD = 1.1364, p-value = 0.2930
alternative hypothesis: NA

> ad.test(a27pe$p2, pexp, 1 )

      Anderson-Darling GoF Test

data:  a27pe$p2 and pexp
AD = 0.5041, p-value = 0.7424
alternative hypothesis: NA

> ad.test(a27pe$p1[1:100], pexp, 1 )

      Anderson-Darling GoF Test

data:  a27pe$p1[1:100] and pexp
AD = 0.7202, p-value = 0.5419
alternative hypothesis: NA
```

```
> ad.test(a27pe$p1[101:193], pexp, 1 )
```

Anderson-Darling GoF Test

```
data: a27pe$p1[101:193] and pexp
AD = 1.4046, p-value = 0.201
alternative hypothesis: NA
```

```
> ad.test(a27pe$p2[1:100], pexp, 1 )
```

Anderson-Darling GoF Test

```
data: a27pe$p2[1:100] and pexp
AD = 0.4758, p-value = 0.7712
alternative hypothesis: NA
```

```
> ad.test(a27pe$p2[101:193], pexp, 1 )
```

Anderson-Darling GoF Test

```
data: a27pe$p2[101:193] and pexp
AD = 0.3381, p-value = 0.9068
alternative hypothesis: NA
```

```
>
```

=====

We see that sets of 193, 100, and 93 packets from two different streams (p1 and p2) all passed the $AD \leq 2.492$ criterion.

6.4.3. Conclusions for Goodness-of-Fit

Both NetProbe and Perfas+ implementations produce adequate Poisson distributions according to the Anderson-Darling Goodness-of-Fit at the 5% significance ($1-\alpha = 0.05$, or 95% confidence level).

6.5. Implementation of Statistics for One-Way Loss

We check to see which statistics were implemented and report on those facts, noting that Section 4 of [RFC2680] does not specify the calculations exactly and only gives some illustrative examples.

	NetProbe	Perfas+
Type-P-One-way-Packet-Loss-Average (this is more commonly referred to as "loss ratio")	yes	yes

Implementation of RFC 2680 Section 4 Statistics

We note that implementations refer to this metric as a loss ratio, and this is an area for likely revision of the text to make it more consistent with widespread usage.

7. Conclusions for a Revision of RFC 2680

This memo concludes that [RFC2680] should be advanced on the Standards Track and recommends the following edits to improve the text (which are not deemed significant enough to affect maturity).

- o Revise Type-P-One-way-Packet-Loss-Ave to Type-P-One-way-Delay-Packet-Loss-Ratio.
- o Regarding implementation of the loss delay threshold (Section 6.2), the assumption of post-processing is compliant, and the text of the revision of RFC 2680 should be revised slightly to include this point.
- o The IETF has reached consensus on guidance for reporting metrics [RFC6703], and this memo should be referenced in a revision of RFC 2680 to incorporate recent experience where appropriate.

We note that there are at least two errata for [RFC2680], and it appears that these minor revisions should be incorporated in a revision of RFC 2680.

The authors that revise [RFC2680] should review all errata filed at the time the document is being written. They should not rely upon this document to indicate all relevant errata updates.

We recognize the existence of BCP 170 [RFC6390], which provides guidelines for development of documents describing new performance metrics. However, the advancement of [RFC2680] represents fine-tuning of long-standing specifications based on experience that

helped to formulate BCP 170, and material that satisfies some of the requirements of [RFC6390] can be found in other RFCs, such as the IPPM Framework [RFC2330]. Thus, no specific changes to address BCP 170 guidelines are recommended for a revision of RFC 2680.

8. Security Considerations

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

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 Customer Premises Equipment (CPE) router for the NetProbe side of the test setup 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 on statistical interpretation with Ganga Maguluri.

Constructive comments and helpful reviews were also provided by Bill Cervený, Joachim Fabini, and Ann Cervený.

10. Appendix - Network Configuration and Sample Commands

This Appendix provides some background information on the host configuration and sample tc commands for the "netem" network emulator, as described in Section 3 and Figure 1 of this memo. These details are also applicable to the test plan in [RFC6808].

The host interface and configuration are shown below. Due to the limit of 72 characters per line, line breaks were added to the "tc" commands in the output below.

```
[system@dell4-4 ~]$ su
Password:
[root@dell4-4 system]# service iptables save
iptables: Saving firewall rules to /etc/sysconfig/iptables:[ OK ]
[root@dell4-4 system]# service iptables stop
iptables: Flushing firewall rules: [ OK ]
iptables: Setting chains to policy ACCEPT: nat filter [ OK ]
iptables: Unloading modules: [ OK ]
[root@dell4-4 system]# brctl show
bridge name      bridge id                STP enabled    interfaces
virbr0           8000.00000000000000      yes
[root@dell4-4 system]# ifconfig eth1.300 0.0.0.0 promisc up
[root@dell4-4 system]# ifconfig eth1.400 0.0.0.0 promisc up
[root@dell4-4 system]# ifconfig eth2.400 0.0.0.0 promisc up
[root@dell4-4 system]# ifconfig eth2.300 0.0.0.0 promisc up
[root@dell4-4 system]# brctl addbr br300
[root@dell4-4 system]# brctl addif br300 eth1.300
[root@dell4-4 system]# brctl addif br300 eth2.300
[root@dell4-4 system]# ifconfig br300 up
[root@dell4-4 system]# brctl addbr br400
[root@dell4-4 system]# brctl addif br400 eth1.400
[root@dell4-4 system]# brctl addif br400 eth2.400
[root@dell4-4 system]# ifconfig br400 up
[root@dell4-4 system]# brctl show
bridge name      bridge id                STP enabled    interfaces
br300            8000.0002b3109b8a       no              eth1.300
                  eth2.300
br400            8000.0002b3109b8a       no              eth1.400
                  eth2.400
virbr0           8000.00000000000000      yes
```

```
[root@dell4-4 system]# brctl showmacs br300
port no mac addr          is local?      ageing timer
  2    00:02:b3:10:9b:8a    yes            0.00
  1    00:02:b3:10:9b:99    yes            0.00
  1    00:02:b3:c4:c9:7a    no             0.52
  2    00:02:b3:cf:02:c6    no             0.52
  2    00:0b:5f:54:de:81    no             0.01
[root@dell4-4 system]# brctl showmacs br400
port no mac addr          is local?      ageing timer
  2    00:02:b3:10:9b:8a    yes            0.00
  1    00:02:b3:10:9b:99    yes            0.00
  2    00:02:b3:c4:c9:7a    no             0.60
  1    00:02:b3:cf:02:c6    no             0.42
  2    00:0b:5f:54:de:81    no             0.33
[root@dell4-4 system]# tc qdisc add dev eth1.300 root netem
delay 100ms

[root@dell4-4 system]# ifconfig eth1.200 0.0.0.0 promisc up
[root@dell4-4 system]# vconfig add eth1 100
Added VLAN with VID == 100 to IF -:eth1:-

[root@dell4-4 system]# ifconfig eth1.100 0.0.0.0 promisc up

[root@dell4-4 system]# vconfig add eth2 100
Added VLAN with VID == 100 to IF -:eth2:-

[root@dell4-4 system]# ifconfig eth2.100 0.0.0.0 promisc up
[root@dell4-4 system]# ifconfig eth2.200 0.0.0.0 promisc up
[root@dell4-4 system]# brctl addbr br100
[root@dell4-4 system]# brctl addif br100 eth1.100
[root@dell4-4 system]# brctl addif br100 eth2.100
[root@dell4-4 system]# ifconfig br100 up
[root@dell4-4 system]# brctl addbr br200
[root@dell4-4 system]# brctl addif br200 eth1.200
[root@dell4-4 system]# brctl addif br200 eth2.200
[root@dell4-4 system]# ifconfig br200 up
[root@dell4-4 system]# brctl show
bridge name      bridge id                STP enabled    interfaces
br100            8000.0002b3109b8a        no             eth1.100
                                                         eth2.100
br200            8000.0002b3109b8a        no             eth1.200
                                                         eth2.200
br300            8000.0002b3109b8a        no             eth1.300
                                                         eth2.300
br400            8000.0002b3109b8a        no             eth1.400
                                                         eth2.400
virbr0           8000.000000000000        yes
```

```

[root@dell4-4 system]# brctl showmacs br100
port no mac addr          is local?      ageing timer
  2    00:02:b3:10:9b:8a    yes            0.00
  1    00:02:b3:10:9b:99    yes            0.00
  1    00:0a:e4:83:89:07    no             0.19
  2    00:0b:5f:54:de:81    no             0.91
  2    00:e0:ed:0f:72:86    no             1.28
[root@dell4-4 system]# brctl showmacs br200
port no mac addr          is local?      ageing timer
  2    00:02:b3:10:9b:8a    yes            0.00
  1    00:02:b3:10:9b:99    yes            0.00
  2    00:0a:e4:83:89:07    no             1.14
  2    00:0b:5f:54:de:81    no             1.87
  1    00:e0:ed:0f:72:86    no             0.24
[root@dell4-4 system]# tc qdisc add dev eth1.100 root netem
                        delay 100ms
[root@dell4-4 system]#

```

=====

Some sample tc command lines controlling netem and its impairments are given below.

```

tc qdisc add dev eth1.100 root netem loss 0%
tc qdisc add dev eth1.200 root netem loss 0%
tc qdisc add dev eth1.300 root netem loss 0%
tc qdisc add dev eth1.400 root netem loss 0%

```

Add delay and delay variation:

```

tc qdisc change dev eth1.100 root netem delay 100ms 50ms
tc qdisc change dev eth1.200 root netem delay 100ms 50ms
tc qdisc change dev eth1.300 root netem delay 100ms 50ms
tc qdisc change dev eth1.400 root netem delay 100ms 50ms

```

Add delay, delay variation, and loss:

```

tc qdisc change dev eth1 root netem delay 2000ms 1000ms loss 10%

```

=====

11. References

11.1. Normative References

- [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.
- [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.
- [RFC4737] Morton, A., Ciavattone, L., Ramachandran, G., Shalunov, S., and J. Perser, "Packet Reordering Metrics", RFC 4737, November 2006.
- [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.
- [RFC6390] Clark, A. and B. Claise, "Guidelines for Considering New Performance Metric Development", BCP 170, RFC 6390, October 2011.
- [RFC6576] Geib, R., Morton, A., Fardid, R., and A. Steinmitz, "IP Performance Metrics (IPPM) Standard Advancement Testing", BCP 176, RFC 6576, March 2012.

- [RFC6703] Morton, A., Ramachandran, G., and G. Maguluri, "Reporting IP Network Performance Metrics: Different Points of View", RFC 6703, August 2012.
- [RFC6808] Ciavattone, L., Geib, R., Morton, A., and M. Wieser, "Test Plan and Results Supporting Advancement of RFC 2679 on the Standards Track", RFC 6808, December 2012.

11.2. Informative References

- [ADK] Scholz, F. and M. Stephens, "K-Sample Anderson-Darling Tests of Fit, for Continuous and Discrete Cases", University of Washington, Technical Report No. 81, May 1986.
- [ADV-METRICS] Morton, A., "Lab Test Results for Advancing Metrics on the Standards Track", Work in Progress, October 2010.
- [FEDORA] "Fedora", <<http://fedoraproject.org/>>.
- [LOSS-METRIC] Almes, G., Kalidindi, S., Zekauskas, M., and A. Morton, Ed., "A One-Way Loss Metric for IPPM", Work in Progress, July 2014.
- [MII-TOOL] Hinds, D., Becker, D., and B. Eckenfels, "Linux System Administrator's Manual", February 2013, <<http://man7.org/linux/man-pages/man8/mii-tool.8.html>>.
- [NETEM] Linux Foundation, "netem", <<http://www.linuxfoundation.org/collaborate/workgroups/networking/netem>>.
- [Perfas] Heidemann, C., "Qualitaet in IP-Netzen Messverfahren", published by ITG Fachgruppe, 2nd meeting 5.2.3, November 2001, <www.itg523.de/oeffentlich/01nov/Heidemann_QOS_Messverfahren.pdf>.
- [RADGOF] Bellosta, C., "ADGofTest: Anderson-Darling Goodness-of-Fit Test. R package version 0.3.", R-Package Version 0.3, December 2011, <<http://cran.r-project.org/web/packages/ADGofTest/index.html>>.
- [RADK] Scholz, F., "ADK: Anderson-Darling K-Sample Test and Combinations of Such Tests. R package version 1.0.", 2008.

- [RFC3931] Lau, J., Townsley, M., and I. Goyret, "Layer Two Tunneling Protocol - Version 3 (L2TPv3)", RFC 3931, March 2005.
- [RT00L] R Development Core Team, "R: A Language and Environment for Statistical Computing", ISBN 3-900051-07-0, 2014, <<http://www.R-project.org/>>.
- [WIPM] AT&T, "AT&T Global IP Network", 2014, <<http://ipnetwork.bgtmo.ip.att.net/pws/index.html>>.

Authors' Addresses

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

Phone: +1 732 420 1239
EMail: lencia@att.com

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
Technical University Darmstadt
Darmstadt
Germany

EMail: matthias_michael.wieser@stud.tu-darmstadt.de