Network Working Group   A. Morton
Internet-Draft   AT&T Labs
Intended status: Informational   October 25, 2010
Expires: April 28, 2011

Lab Test Results for Advancing Metrics on the Standards Track
draft-morton-ippm-advance-metrics-02

Abstract

This memo supports the process of progressing performance metric RFCs along the standards track. Observing that the metric definitions themselves should be the primary focus rather than the implementations of metrics, this memo describes results of example lab test procedures to evaluate specific metric RFC requirement clauses to determine if the requirement has been implemented as intended. A single implementation has been tested against the key specifications of RFC 2679 on One-way Delay.

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

Copyright Notice

Table of Contents

1. Introduction

The IETF (IP Performance Metrics working group) has been considering 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 path equal for both, and because considerable variation was allowed in the parameters of each metric.

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. An earlier version of this draft, titled "Problems and Possible Solutions for Advancing Metrics on the Standards Track", brought many issues to light and offered some solutions. Sections from the earlier draft has now been combined with [draft-geib-ippm-metrictest] resulted in an IPPM working group draft, [draft-ippm-metrictest-00.txt]. The plan now emphasizes evaluating the metric specifications themselves, as a result of this interaction.

There is now consensus that the metric definitions should be the primary focus 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 add more support for the current approach as the author perceives it to be. 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-79 in Beijing.

Another aspect of the metric RFC advancement process which has received limited attention is the requirement to document the work and results. The procedures of [RFC2026] are expanded in[RFC5657], including sample implementation and interoperability reports. Section 3 of this memo can serve as a template 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.

We have also 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 investigates that topic by outlining a procedure that includes same-implementation comparisons to help set the equivalence threshold.

This memo also discusses an issue with some network emulators, namely correlated loss or burst loss generation.

Finally, this memo is also an open invitation to developers or testers who would be willing to use their equipment to help advance the IPPM metrics through lab tests, like the tests described below.

2. A Definition-centric metric advancement process

The process described in Section 3.5 of [draft-ippm-metrictest-00.txt] 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:

```
      ,---.
     /     \
    ( Start )
     \     /    Implementations
      ‘-+-’        +-------+
        |         /|   1   ‘.
    +---+----+   / +-------+ ‘.-----------+      ,-------.
    |  RFC   |  /             |Check for  |    ,’ was RFC ‘.  YES
    |        | /              |Equivalence.....  clause x   -------+
    |        |/    +-------+  |under      |    ‘. clear?  ,’       |
    | Metric \.....|   2   ....relevant   |      ‘---+---’    +----+---+
    | Metric |\    +-------+  |identical  |       No |        |Report  |
    | Metric | \              |network    |      +---+---.    |results+|
    |  ...   |  \             |conditions |      |Modify |    |Advance |
    |        |   \ +-------+  |           |      |Spec   +----+  RFC   |
    +--------+    \|   n   |.’+-----------+      +-------+    |request?|
                   +-------+                                  +--------+
```

3. Lab test results to check metric definitions

This section describes some results from lab tests with test devices and a network emulator to create relevant conditions and 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 [draft-ippm-metrictest-00.txt]. The principle modification the use of the mean statistic for comparisons.

The metric implementation used was NetProbe version 5.8.5, (an earlier version is used in the WIPM system and deployed world-wide). Accuracy of NetProbe measurements is usually limited by NTP synchronization performance (~1ms error or greater), although this lab environment often exhibits errors much less than typical for NTP.

The network emulator is a host running Fedora Core Linux [http://fedoraproject.org/] with IP forwarding enabled and the NIST Net emulator 2.0.12b [http://snad.ncsl.nist.gov/nistnet/] loaded and operating.

The links between NetProbe hosts and the NIST Net emulator host were 100baseTx-FD (100Mbps full duplex) as reported by "mii-tool", except as noted below.

For these tests, a stream of at least 30 packets were sent from Source to Destination in each implementation. Periodic streams (as per [RFC3432]) with 1 second spacing were used, except as noted.

These examples do not entirely avoid the problem of declaring equivalence with a statistical test, but the lab conditions should simplify the problem by removing as much variability as possible.

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

3.1. 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 a path with 1 sec one-way constant delay

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

3. configure the path with 3 sec one-way delay (or change the path delay while test is in progress, when there are sufficient packets at the first delay setting)

4. repeat/continue measurements

5. observe that the increase measured in step 4 caused all packets with 3 sec delay to be declared lost, and that all packets that arrive successfully in step 2 are assigned a valid one-way delay.

3.1.1. NetProbe Lab 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 2 seconds, all packets with one-way delay >2 seconds are marked "Lost" and included in the Lost Packet list with their transmission time (as required in Section 3.3 of [RFC2680]). 22 of 38 packets were declared lost.

3.1.2. XXX Lab Results for Loss Threshold

>>> Comment: this section is a placeholder

3.1.3. Conclusions on Lab Results for Loss Threshold

>>> Comment: this section is a placeholder

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

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.

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

1. configure a path with X ms one-way constant delay, and ideally including a low-speed link

2. measure (average) one-way delay with 2 or more implementations, using identical options and equal size small packets (e.g., 100 octet IP payload)

3. maintain the same path with X ms one-way delay

4. measure (average) one-way delay with 2 or more implementations, using identical options and equal size large packets (e.g., 1500 octet IP payload)

5. observe that the increase measured in 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.

3.2.1. NetProbe Lab results for Serialization

For this test only, the link between the NetProbe Source host and the NIST Net emulator host was changed to 10baseT-FD (10Mbps full duplex) as configured by "mii-tool".

The value of X = 1000 ms was used in the NIST Net emulator.

When the UDP payload size was increased from 32 octets to 1400 octets, the NIST Net emulator exhibited a bi-modal delay distribution. Investigation confirmed that the NetProbe implementations tested did not exhibit bi-modal delay on an alternate (network management) path.

1400 byte payload Delay for each mode microseconds	32 byte payload (one mode) microseconds	Delay Diff microseconds	Expected Diff microseconds
1001621	1000356	1265	1094.4
1002735	1000356	2379	1094.4

Average Delay over 60 packets for different payload sizes with Delay computations and comparison with expected delay difference for serialization.

For the lower-delay mode, the Delay Difference between payload sizes is about 170 microseconds higher than expected. However, it is clear that delay increased with a larger payload as expected when the measurement is conducted First-bit to Last-bit and includes serialization time.

The higher mode appears on almost every other packet in the stream, and comments are sought on possible configuration changes that would remove this bi-modal behavior without significant sacrifices in other dimensions of performance.

UPDATE: Additional investigation appears to conclude that the modal behavior is related to interrupt-to-frame arrival settings of the specific interface board. Various options appear to be configurable, but only when the interface driver is compiled as a module. Also, the board/driver does not support the "coalesce" options of ethtool. Until we can rebuild the Linux machine with this and other planned modifications, confirmation will have to wait.

3.3. 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 a path with X ms one-way constant delay

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.

3.3.1. NetProbe Lab results for Differential Delay

In this test, X=1000ms and Y=2000ms.

Average pre-increase delay, microseconds	1000276.6
Average post 2s additional, microseconds	3000282.6
Difference (should be ~= Y = 2s)	2000006

Average delays before/after 2 second increase

The NetProbe implementation exhibited a 2 second increase with a 6 microsecond error (assuming that the NIST Net emulated delay difference is exact).

3.4. One-way Delay, ADK Sample Metric (Lab)

This test determines if implementations produce results that appear to come from the same delay distribution. In addition, same-implementation 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].

1. Configure a path with X ms one-way constant delay.

2. Measure a sample of one-way delay singletons with 2 or more implementations, using identical options.

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

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

5. Take the largest resolution and confidence factor for distribution equivalence from the same-implementation pairs as the equivalence threshold for these experimental conditions. >>> Question: do we need to account for additional cross-implementation error? How much?

6. Compare the cross-implementation ADK performance with the equivalence threshold determined in step 4 to determine if equivalence can be declared.

3.4.1. NetProbe Lab results for ADK

To be provided, the same-implementation lab tests have been completed, but the analysis was not ready in time for publication.

ADK Results for same-implementation

3.5. Error Calibration, RFC 2679

This is a simple check to determine if an implementation reports the error calibration as required in Section 4.8 of [RFC2679]. Note that the context (Type-P) must also be reported.

3.5.1. Net Probe Error and Type-P

NetProbe error is dependent on the specific version and installation details, and was discussed briefly above.

Type-P for this test was IP-UDP with Best Effort DCSP.

4. Notes on Network Emulator Loss Generation

While network emulators can be expect to generate independent random loss, it is well-understood that real loss tends to be correlated to some extent.

NistNet and many earlier and current network emulators use the same effective function to generate correlated values for delay and correlated values for comparison with a loss threshold. The correlation relationship in many emulator descriptions takes the following form:

Corr_value = Last_value * corr_coeff + New_value * (1-corr_coeff)

where:

- o New_value is the random value from some distribution
- o Last_value is the result of this equation for the previous packet
- o corr_coeff is the correlation coefficient, [+1, -1]

o Corr_value is the revised random value with correlation

This seems to work adequately for delay, as seen in [NistNet]. However, it does not appear to be possible to produce long loss bursts with low probability using this equation. We note that a somewhat more complicated relationship is implemented in the NistNet code, and avoids range violations that may be possible with correlations at the end of range.

Investigation of similar, but alternative relationship to generate loss bursts has begun as part of this effort, and a candidate equation has been developed. Integration with an existing emulator is in-progress.

It bears note that some network emulators can produce deterministic loss durations in time and/or in lost packets, but the frequent appearance of the relationship above is disturbing, given its poor ability to produce burst loss, as far as existing tests show.

5. Security Considerations

There are no security issues raised by discussing the topic of metric RFC advancement along the standards track.

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

6. IANA Considerations

This memo makes no requests of IANA, and hopes that IANA will leave it alone, as well.

7. Acknowledgements

The author would like to thank Len Ciavattone for continued consultations on the laboratory aspects of this work, and Yaakov Stein for a useful discussion on the bi-modal delay behavior observed in the Linux-based router and network emulator used here.

8. 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.

[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.

Author's Address

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/