Initial Performance Metric Registry Entries

~~draft-mornuley-ippm-initial-registry-01,2,3~~

~~draft-morton-ippm-initial-registry-0,1,2,3,4~~

draft-ietf-ippm-initial-registry-07

A. Morton, M. Bagnulo, P. Eardley, K. D’Souza

Registry for Performance Metrics

draft-ietf-ippm-metric-registry-15

M. Bagnulo, B. Claise, P. Eardley,

A. Morton, A. Akhter

Quick Summary

Overall Registry Concept

- Problem: How can we specify with Precision the Metrics and Methods to Implement and Use?
  - Many Standardized Metrics with similar names
  - Registry enables all parties to be sure they're talking about the same Metric: Active, Passive, Hybrid
  - Flexibility and customization of Generic Metrics seen as an advantage in standards development
  - Methods allow variables, system issues out-of-scope
- Provide Unique ID and detailed exposition
  - Raise the bar from Standard to Registered Metrics
  - Fix critical Parameters, but Allow Run-Time flexibility

Quick Summary

all Registry Concept & Format

- draft-ietf-ippm-metric-registry-15 <= No Substantial Revisions!
- Each entry in the registry is a row
  - Series of columns
    - Typically ~1 column may be Not Applicable
  - Clustered in categories
- Each row is indexed by ID
  - 16 bit flat identifier
  - With associated name (i-d defines naming convention)
  - Auto-generate URI (pre-pend urn:ietf:metric: to name)
  - Auto-generate URL (location of text file with registry entry)
- Control & report protocols use URI
- Next slide shows category /column headings
  - Layout is purely presentational (slide not wide enough, neither is anyone’s screen, which is why the text file presentation is available)

Quick Summary

Categories & Columns

Category	Column	....

Summary	ID	Name	URIs	Description	Ref	Change Ctrl	Ver
Metric definition	Reference	Fixed parameters					
Method of measurement	Ref. Meth. (eg Section 3 of RFC XXXX)	Packet stream generation (active tests)	Traffic filter (passive tests)	Sampling distribution (for traffic filter)	**Run-Time** Parameter(s) (eg.MPaddress)	Role(s) (eg sender)	
Output	Type	Reference Method	Units	Calibration			
Admin info	Status	Requestor	Revision #	Date			
Comments	Full history	....					

Maybe a lot of info (~sub-columns)

Don't change nature of Method

Registry/Metric Drafts Updates 07&15

- RFC 2119 and RFC 8174 Req. Terms
- Rev. Passive TCP Metric: Thanks Yakov
  - Heuristics for Methods of Measurement
    - Can improve these if we get some more review…
- DNS (section 6): clarified:
  - “This is a metric for DNS Response performance from a network user's perspective, for a specific named resource. The metric can be measured repeatedly using different resource names.”

Discussion Points

- DNS Response Time **and Loss**, section 6
  - @@@@ <Loss> would require support of ID generation and population in the Message. An alternative would be to use a random Source port on the Query Message, but we would choose ONE method before proceeding.
- UDP Periodic One-way Delay and Loss, sec 8
  - @@@@ should Periodic run-time params be fixed instead? Probably yes if modeling a specific version of tests.

Next Steps

- Discuss and close Open Issues:
  - TCP section review
  - DNS Response Time **and Loss**, section 6
  - Fixed or Run-Time params for Periodic UDP?
- Input from XRBLOCK@IETF-101
  - Agreed to work on new entries (right?)

BACKUP

Registry/Metric Drafts Updates 06&14

- Revised Passive TCP Metric
- add Hand Shake (HS) Metric
  - RTDelay_Passive_IP-TCP-HS_
    RFCXXXXsecY_Seconds_Singleton
  - When the Qualified and Corresponding Packets are a TCP-SYN and a TCP-SYN-ACK,
  - then RTD_fwd == RTD_HS_fwd.
- New elements for Metric Type (Registry Draft)
  - Added HS
  - Added QUIC

Registry/Metric Drafts Updates Future

- Heuristics for Methods of Measurement
  - Can improve these if we get some more review
- Hackathon-101 QUIC Spin bit project table Discussion: How would the TCP Heuristics change?
  - can't use RTT-window analysis with Spin-bit - need a larger window.
  - A connection duration should be expressed in a Number of RTT.
  - Adaptive algs are needed - there are many sources of variation
  - Handshake Can be used as a lower limit filter, but should challenge after N# samples (in case of route-change)
  - Use the detection of Loss to invalidate RTT samples (if available).
  - Need a completely different definition of Corresponding Packets,
  - Need to define Corresponding Sequential Packet PAIR, with a specific edge transition
  - Methods would be QUIC version-specific if Spin-meas not part of the Invariants.
  - For QUIC: RTTM in RFC7323 methods need to be replaced

Registry/Metric Drafts Updates Future (06)

- comment from Brian: there is an interesting method for DNS measurement by encoding information in the query itself. It is a question of what exactly we are trying to measure: a specific RR, or the infrastructure itself. (at this time we measure a specific RR).
- Root Server Service Advisory Committee (RSSAC) (mail from Fred Baker)
  - Need Metrics to support Recommendations
  - “SLA—like”

Initial Performance Metric Registry Entries Part 2: MBM

draft-morton-ippm-mbm-registry-01

A. Morton, M. Mathis

Was the Registry Design efficient?

- short answer: YES
- Many Run-time Parameters:
  - Targets(rate,RTT,MTU), derating, SPRT params
- Additional parameters calculated in the model
- Primary Output: Pass/Fail/Inconclusive, or PFI
- Also Output Loss ratio and Mean RTT results
- Learned more about model specification, especially packet stream generation

A Test of the Registry Design

- Brian/Chair: How can Speed be specified as a Registry Entry, using Model-Based Metrics?
    - Should be part of the initial registry…
    - Name:
- `OWMBM_Active_IP-TCP-SustainedBurst_RFCXXXXsecY_Enumerated_PFI`
- Sustained Burst is described in 8.5.1 of [MBM]
- This test describes stream conditions to evaluate a target_rate, at target RTT and MTU
- Loss and RTT are the primary measurements
    - Needed Loss, have it now.
- Added remaining data formats
- Clarified Packet Stream Generation Parameters:

paced_bursts Send bursts on a timer. Specify any 3 of: average data rate, packet size, burst size (number of packets) and burst headway (burst start to start).

Two-way street

- Started as a “test” of the Registry
- Main impact was on Name Elements
- Authors examined MBM specifications more deeply
  - Feedback is valuable
- Also, some feedback from MBM experimenter

Next Steps

- Further author and WG review
- Combine with Part 1? Or just keep separate?

More Updates

IANA Metric section

- Revised Entries
- Each entry will be mocked-up (partial view)

ID	Name	URIs	Description
1	RTDelay_Active_IP-UDP-Periodic_RFCXXXXsecY_Seconds_95Percentile	urn:ietf:metrics:perf:RTDelay_Active_IP-UDP-Periodic_RFCXXXXsecY_Seconds_95Percentile url: https://tools.ietf.org/html/draft-ietf-ippm-initial-registry-02#section-4	This metric assesses the delay of a stream of packets exchanged between two hosts (which are the two measurement points), and the Output is the Round-trip delay for all successfully exchanged packets expressed as the 95th percentile of their conditional delay distribution.
2	RTLoss_Active_IP-UDP-Periodic_RFCXXXXsecY_Percent_LossRatio	urn:ietf:metrics:perf: RTLoss_Active_IP-UDP-Periodic_RFCXXXXsecY_Percent_LossRatio url: https://tools.ietf.org/html/draft-ietf-ippm-initial-registry-02#section-4	This metric assesses the loss ratio of a stream of packets exchanged between two hosts (which are the two measurement points), and the Output is the Round-trip loss ratio for all successfully exchanged packets expressed as a percentage.
3	OWPDV_Active_IP-UDP-Periodic_RFCXXXXsecY_Seconds_95Percentile	urn:ietf:metrics:perf:OWPDV_Active_IP-UDP-Periodic_RFCXXXXsecY_Seconds_95Percentile url: https://tools.ietf.org/html/draft-ietf-ippm-initial-registry-02#section-5	An assessment of packet delay variation with respect to the minimum delay observed on the stream, and the Output is expressed as the 95th percentile of the packet delay variation distribution.

Registry Updates: New Name Elements

MetricType:

- (ICMP & TCP were already present)

Units:

- Packets
- PPS (Packets Per Second)

Output:

- Count

Name Element Sub-registries (mock-up, partial view)

MetricType: a combination of the directional properties and the metric measured

Element	Name	Description	Reference	Change Controller
MetricType	RTDelay	Round Trip Delay	RFCXXXX Section 7.1.2	IETF
MetricType	RTDNS	Response Time Domain Name Service	RFCXXXX Section 7.1.2	IETF
MetricType	RLDNS	Response Loss Domain Name Service	RFCXXXX Section 7.1.2	IETF
MetricType	OWDelay	One-way Delay	RFCXXXX Section 7.1.2	IETF
MetricType	RTLoss	Round Trip Loss	RFCXXXX Section 7.1.2	IETF

Previous To Do (IETF-99)

- Do the Name Element Sets cover Passive well-enough?
  - Feedback was to prepare a TCP RT Delay Metric
- Brian Trammell provided extensive ref's on ippm-list
- New Passive TCP Metrics in Section 10

TCP Passive Metrics

- Registry accommodates passive (!)
- New name elements (routine now)

New Metrics:

- RTDelay_Passive_IP-TCP_RFCXXXXsecY_Seconds_<statistic>
  - where <statistic> = Mean; Min; Max
- RTLoss_Passive_IP-TCP_RFCXXXXsecY_Packet_Count

Two-part RTDelay Composition

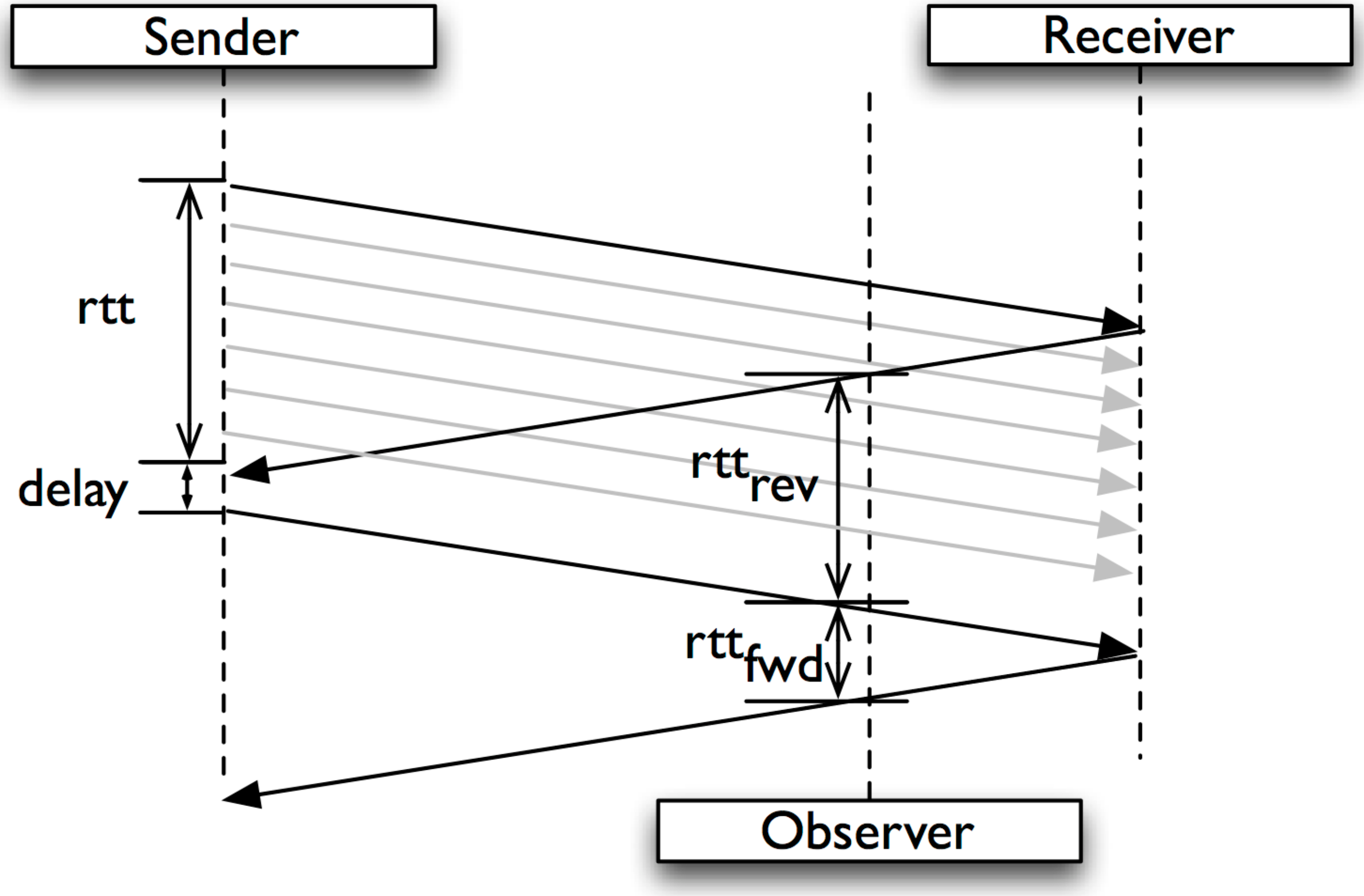

Figure 1: Trammell, et al., “Inline Data Integrity Signals for Passive Measurement”, https://trammell.ch/pdf/qof-tma14.pdf

Passive Metrics for TCP

- No (?) Standard Metric and Method Ref !
  - Assumed Observation Point in the “middle”
  - Major Effort to Define the Metrics
  - Additional work to provide Delay & Loss Methods
  - Included Heuristics from References
  - Delay stats and counts apply to a single TCP connection:
    - (FIN-ACK pairs terminate the measurement interval)

Passive Metrics for TCP: OPENS

- Search for @@@@
- Really, no standardized metrics ?!?!?!?
- Use first-bit -> last-bit in Delay Metric?
- RTDelay-SA: Should we add a separate singleton metric ??
  - (seems reasonable, but no loss metric however)
  - Rachel suggests RTD_fwd and RTD_rev, too
- Realistically, the entire section needs review

Feedback on the Registry Contents

- Seeking feedback on the current contents, and what else the WG (and regu-guests) want
- All Sections (4 thru 8) updated
- **New Metrics**
  - ICMP
  - Passive TCP

Registry/Metric Drafts Updates 05&13

- Replaced several Poisson with Periodic Metrics
  - UDP Round-trip delay & Loss and One-way PDV
- New Metrics for ICMP: RT Delay and RT Loss
  - RT Delay Stats: Mean, Min, and Max
- New packet sending discipline: SendOnRcv
  - Used in some “ping” tools
  - Draws on Periodic Stream and Tmax waiting time

How do I get a registry entry?

- Submit request to IANA, with columns filled in
  - Likely prior review in WG
- Review by performance metric experts
  - If necessary, work on improvements with requester
  - Does the proposed registry entry clearly define the metric & method of measurement?
  - Is it different from existing registry entries?
  - Is it operationally useful (significant industry interest or been deployed)?
- IANA adds to registry
- Similar process for revisions
  - Must be backwards compatible (eg editorial)
  - Otherwise create a new metric (& maybe deprecate old one)

Names, identifiers and URIs

- We keep identifiers, names and we automatically generate URIs
  - Identifiers are flat 16-bit integers
  - Names are unique within the registered metrics
  - URIs are generated by prepending urn:ietf:params:performance:metric to the name
- Also, a URL to a text file containing the Registry Entry

End Review, now some Entries

4. UDP Round-trip Latency Registry Entry
  - 4.1. Summary
    - 4.1.1. ID (Identifier)
    - 4.1.2. Name
    - 4.1.3. URI
    - 4.1.4. Description
  - 4.2. Metric Definition
    - **4.2.1. Reference Definition**
    - **4.2.2. Fixed Parameters** .
  - 4.3. Method of Measurement
    - **4.3.1. Reference Method**
    - 4.3.2. Packet Generation Stream
    - 4.3.3. Traffic Filtering (observation) Details
    - 4.3.4. Sampling Distribution
    - **4.3.5. Run-time Parameters and Data Format**
    - 4.3.6. Roles
  - 4.4. Output
    - 4.4.1. Type
    - 4.4.2. Data Format
    - 4.4.3. Reference
    - 4.4.4. Metric Units
  - 4.5. Administrative items
    - 4.5.1. Status
    - 4.5.2. Requestor (keep?)
    - 4.5.3. Revision
    - 4.5.4. Revision Date
  - 4.6. Comments and Remarks

passive ex: https://tools.ietf.org/html/draft-mornulo-ippm-registry-columns-01#section-6

4.2.1 Reference Definition

<Full bibliographic reference to an immutable doc.>

Almes, G., Kalidindi, S., and M. Zekauskas, "A Round-trip Delay Metric for IPPM", RFC 2681, September 1999.

[RFC2681]

<specific section reference and additional clarifications, if needed>

Section 2.4 of [RFC2681] provides the reference definition of the singleton (single value) Round-trip delay metric. Section 3.4 of [RFC2681] provides the reference definition expanded to cover a multi-value sample. Note that terms such as singleton and sample are defined in Section 11 of [RFC2330].

Note that although the definition of "Round-trip-Delay between Src and Dst at T" is directionally ambiguous in the text, this metric tightens the definition further to recognize that the host in the "Src" role will send the first packet to "Dst", and ultimately receive the corresponding return packet from "Dst" (when neither are lost).

4.2.2 Fixed Parameters

```
Type-P:
   o  IPv4 header values:

      *  DSCP: set to 0

      *  TTL set to 255

      *  Protocol: Set to 17 (UDP)

   o  UDP header values:
      *  Checksum: the checksum must be calculated

   o  Payload
      *  Sequence number: 8-byte integer

      *  Timestamp: 8 byte integer.  Expressed as 64-bit NTP timestamp
         as per section 6 of RFC 5905 [RFC5905]

      *  No padding (total of 9 bytes)

   Timeout, Tmax: 3 seconds
```

4.3.1 Reference Method

<for metric, insert relevant section references and supplemental info>

The methodology for this metric is defined as Type-P-Round-trip-Delay-Poisson-Stream in section 2.6 of RFC 2681 [RFC2681] and section 3.6 of RFC 2681 [RFC2681] using the Type-P and Timeout defined under Fixed Parameters.

The method requires sequence numbers or other send-order information to be retained at the Src or included with each packet to disambiguate packet reordering if it occurs. Sequence number is part of the payload described under Fixed Parameters.

Refer to Section 4.4 of [RFC6673] for expanded discussion of the instruction to "send a Type-P packet back to the Src as quickly as possible" in Section 2.6 of RFC 2681 [RFC2681]. Section 8 of [RFC6673] presents additional requirements which shall be included in the method of measurement for this metric.

4.3.5 Run-time Parameters and Data Format

<list of run-time parameters, and their data formats>

- Src, the IP address of a host (32-bit value for IPv4, 128-bit value for IPv6)

- Dst, the IP address of a host (32-bit value for IPv4, 128-bit value for IPv6)

- T0, a time (start of measurement interval, 128-bit NTP Date Format, see section 6 of [RFC5905]). When T0 is "all-zeros", a start time is unspecified and Tf is to be interpreted as the Duration of the measurement interval.

- Tf, a time (end of measurement interval, 128-bit NTP Date Format, see section 6 of [RFC5905]), interpreted as the Duration of the measurement interval.

- 1/lambda, average packet rate (for Poisson Streams). (1/lambda = 1 packet per second, if fixed)

- Upper limit on Poisson distribution (values above this limit will be clipped and set to the limit value). (if fixed, Upper limit = 30 seconds.)

4.3.5 Run-time Parameters and Data Format

(continued)

The format for 1/lambda and Upper limit of Poisson Dist. are the short format in [RFC5905] (32 bits) and is as follows: the first 16 bits represent the integer number of seconds; the next 16 bits represent the fractional part of a second.

>>> should Poisson run-time params be fixed instead? probably yes if modeling a specific version of MBA tests.

MORE QUESTIONS -=-

>>> Should we require that each Registry entry have a SINGLE output Format and Statistic ?
(now, the answer is yes)

>>> Should we require that each Registry entry specify the Test Protocol used to collect the metric ?
(seems impractical, MUCH duplication)

>>> Current Entries are Detailed. A kind of roadmap to IPPM Literature. Should we retain this practice (at the risk of non-equivalent metrics)? If you were implementing, would you find this detail helpful?

Section

Example Registry Entry Names: