A registry for IPPM metrics

draft-bagnulo-ippm-new-registry-00
draft-bagnulo-ippm-new-registry-independent-00
M. Bagnulo, T. Burbridge S. Crawford, P. Eardley, A. Morton
Motivation

• Allow a **Controller** to request a **Measurement Agent** to perform a specific test
• Allow a **Measurement Agent** to report the results of a specific test to a **Collector**.
• Protocol Independent registry: Allows multiple protocols to be defined between Controller, Measurement Agents and Collector
  – e.g. IPFIX between the MA and the Collector and YANG/NETCONF between the Controller and MA
  – This may requires additional protocol specific fields
Well-defined and operational metrics

- RFC4148 defined an IPPM metric registry
- RFC6248 obsoleted RFC4148
  - it was "found to be insufficiently detailed to uniquely identify IPPM metrics... [there was too much] variability possible when characterizing a metric exactly"
- New registry: tightly defined metric with few open parameters (don’t affect the nature of the test)
  - e.g. source and dest address and the like
- Less is more: reduced number of metrics proven useful
  - We require both specification AND expert review for new assignments
  - Specification covers the well defined and expert review the operational
- Side benefits:
  - Inventory of useful and used metrics
  - Comparable test results even if performed by different implementations and in different networks
Information to be conveyed

• "Dear measurement agent: Please start test RTT(server.com,150) every day at 2000 GMT. Run RTT test every 1 second for 50 times. Do that when the network is idle. Generate both raw results and 99th percentile mean. Send measurement results to collector.com in IPFIX format"."
Registries

• The commonly used metric registry
  – Scheduling registry
  – Environment registry
  – Output-type registry
  – Metric registry
Scheduling Registry

• Defines the scheduling strategy for the metric
  – Initial allocations
    • Singleton (additional input time of execution)
    • Periodic (additional parameters, initial time, end time and rate)
    • Poisson (additional parameters, initial time, end time and rate)
Environment Registry

• Defines the conditions where the metric is to be executed.
  – Initial allocations
    • Undefined: no additional environment settings are provided
    • No-cross traffic: The "No cross traffic" condition is satisfied when, during the 5 seconds preceding measurement of the metric:
      – the level of traffic flowing through the interface that will be used to send measurement packets in either direction is less than a threshold value of 1% of the line rate of the aforementioned interface.
Output-type Registry

• Defines the type of output the metric produces, either raw or some statistics.
  – Initial allocations
    • Raw: the raw output of the test
    • Xth percentile interval
    • Xth percentile mean
Metric Registry

• Two approaches:
  – Independent registries, the metric being one more
    • The metric registry entries are simply metric IDs
    • A test is defined with as the tuple (metric, scheduling, Output-type, Environment)
    • Possible explosion (implementation side) due to combinations
  – Sub-registries of the metric registry.
    • Each entry of the metric registry fully defines a test
    • The metric registry has the following entries:
      – Metric ID
      – Scheduling (value out of the Scheduling sub-registry)
      – Output-Type (value out of the Output-Type sub registry)
      – Environment (value out of the Environment sub registry)
    • Combinations are defined explicitly, possible explosion of entries
Metric registry

• Initial Allocations (inspired in MBA report):
  – UDP latency
  – UDP packet loss
  – ICMP packet loss
  – DNS latency
  – VOIP latency
  – VOIP loss
  – VOIP Jitter