A registry for IPPM metrics

draft-bagnulo-ippm-new-registry-00

draft-bagnulo-ippm-new-registry-independent-00

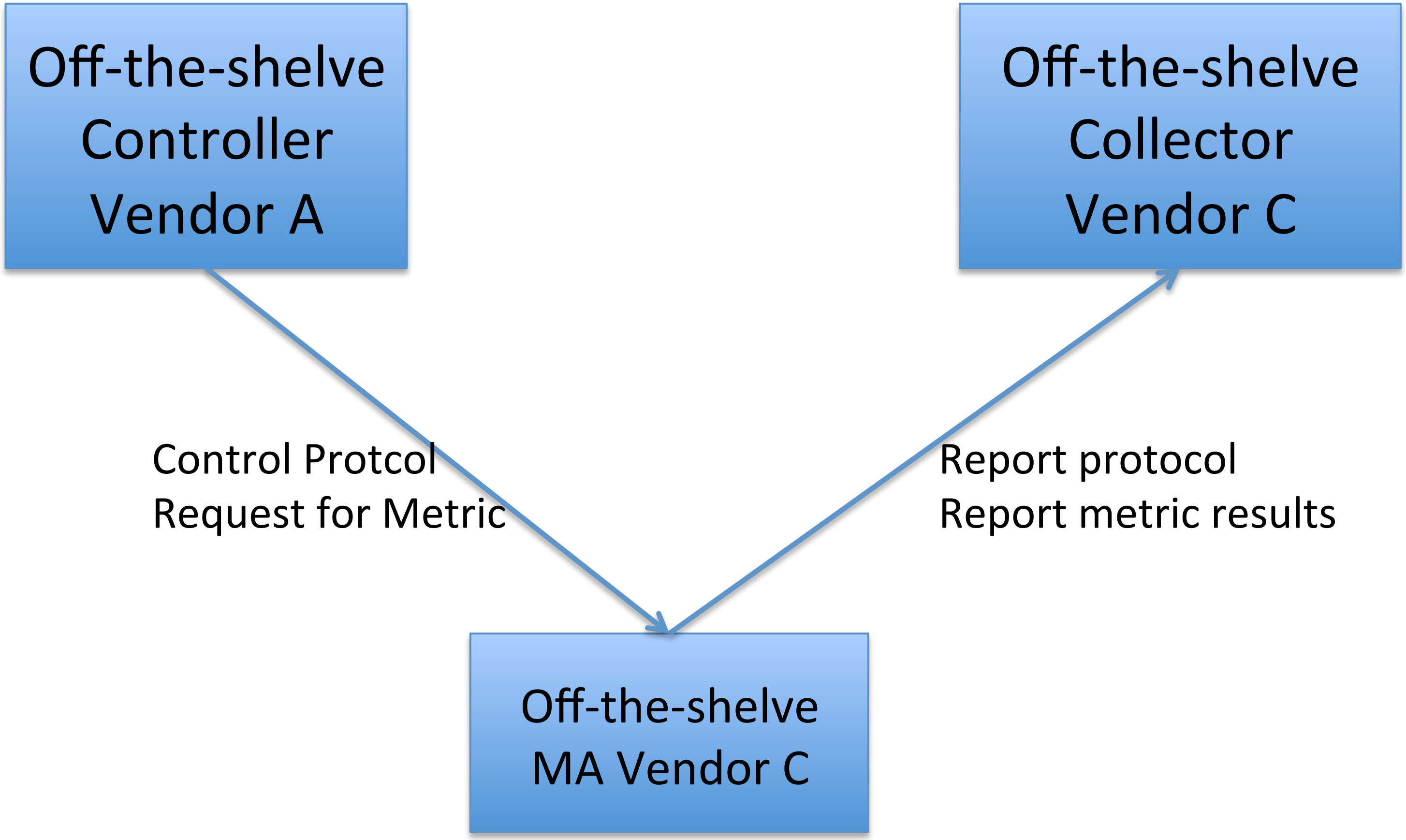
M. Bagnulo, T. Burbridge S. Crawford, P. Eardley, A. Morton

IPPM WG – IETF87

Background

- 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 User: LMAP WG



Off-the-shelve
Controller
Vendor A

Off-the-shelve
Collector
Vendor C

Control Protcol
Request for Metric

Report protocol
Report metric results

Off-the-shelve
MA Vendor C

Well-defined and operational metrics

- 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

The P-Type challenge

- Example: from Type-P-Round-trip-Delay to UDP_Latency
- RFC2681 defines Type-P-Round-trip-Delay
  - Open parameters include P-Type, Timeout, source and destination, time and duration.
- Support for arbitrary P-Type seems costly.
- UDP_Latency: define P-Type (IP fields, plus UDP), leave as open parameters:
  - Source and destination address and UDP ports and time

Schedulling registry

- Currently, each metric defines the sample metrics using some sample distribution
  - e.g. Type-P-Round-trip-Delay-Poisson-Stream
- However, there are a few commonly used sampling strategies with well known input parameters
  - Poisson, Periodic
- Proposal is to define a Schedulling registry that contains the used sampling strategies
- Result: defined number of schedulling strategies to be supported by implementations

Statistics

- Currently, each RFC defines some statistics for each metric
- Again, there are a few statistics that are commonly used.
  - e.g. Xth percentile mean, Xth percentile interval
- Proposal is to define a Output-Type registry with a set of well defined output types that can be specified for each metric

Other enviromental constraints

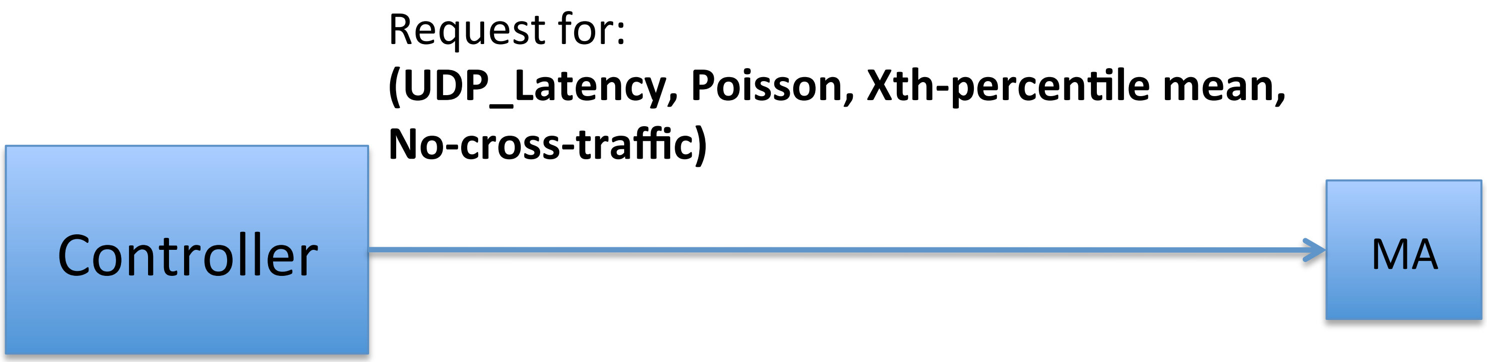
- In some cases, there are additional enviromental constraints that need to be specified as part of the methodology.
- Example: No-cross traffic
- Proposal to define a Environment registry with these constraints.

Registries

- The commnly used metric registry
  - Scheduling registry
  - Environment registry
  - Output-type registry
  - Metric registry

Independent Registries (option 1)

- – Scheduling registry – e.g. Poisson
- – Environment registry – No-cross-traffic
- – Output-type registry – Xth-percentile mean
- – Metric registry – UDP-Latency



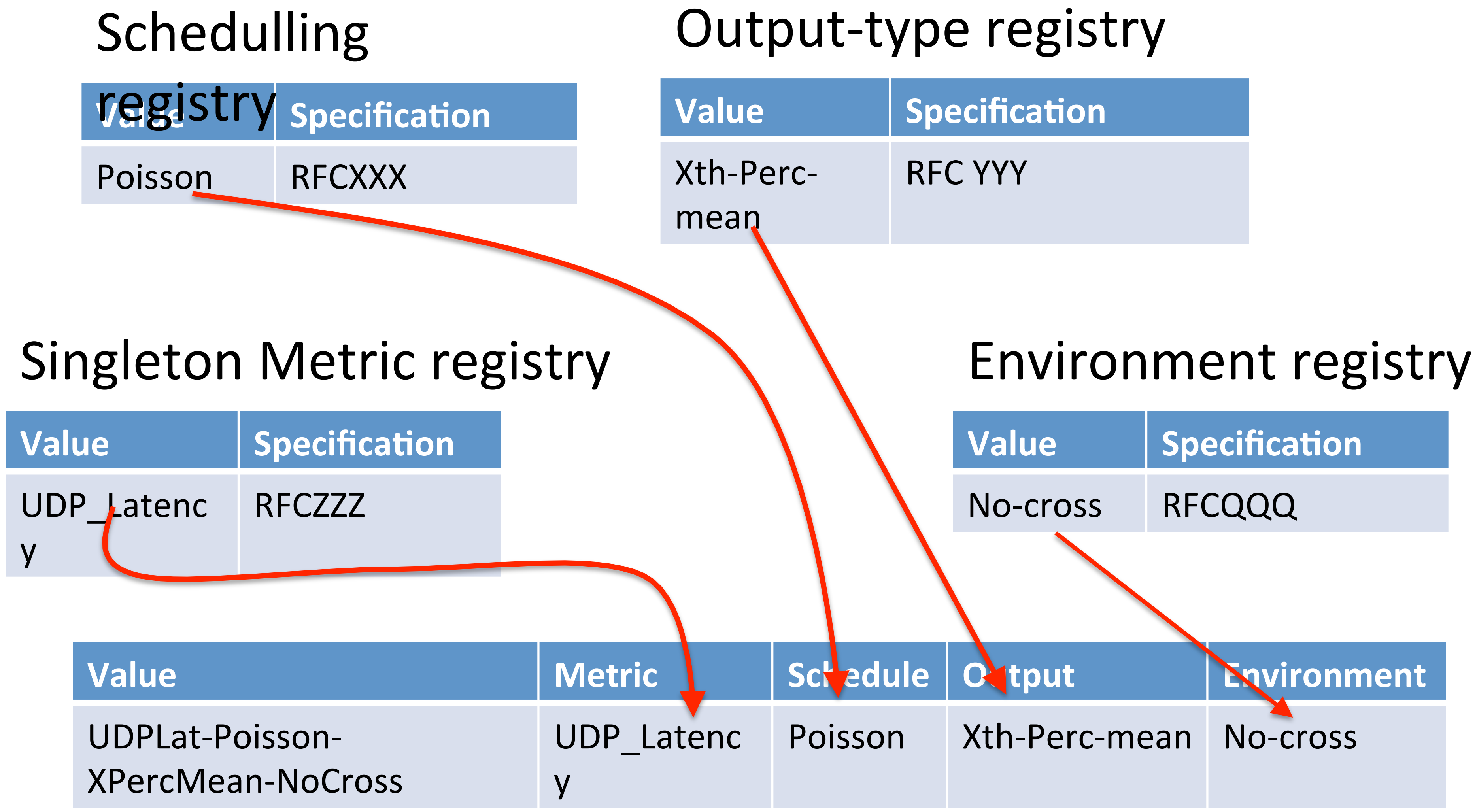
Request for:
(UDP_Latency, Poisson, Xth-percentile mean, No-cross-traffic)

Controller

MA

- Possible drawback: explosion in the implementation side

Sub-registries



Schedulling registry

Value	Specification
Poisson	RFCXXX

Output-type registry

Value	Specification
Xth-Perc-mean	RFC YYY

Singleton Metric registry

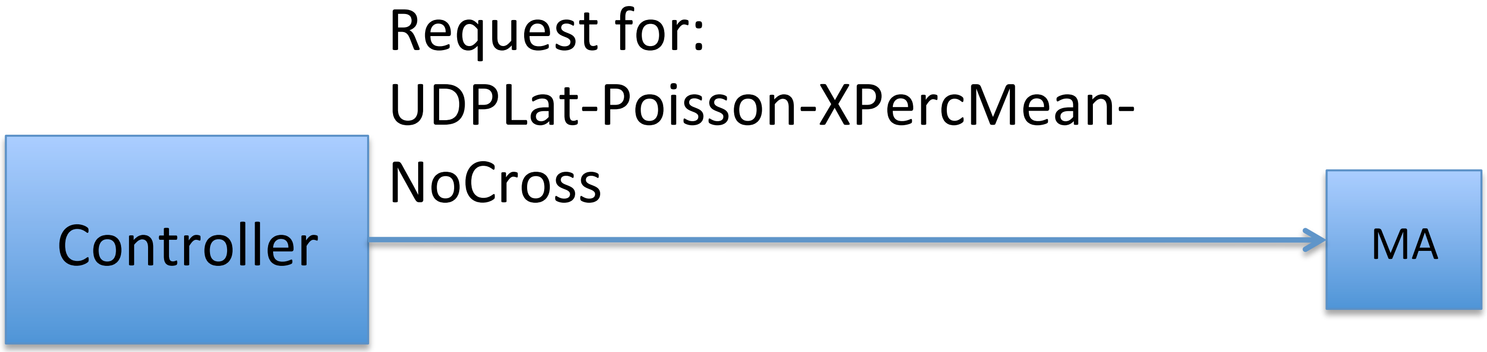
Value	Specification
UDP_Latency	RFCZZZ

Environment registry

Value	Specification
No-cross	RFCQQQ

Value	Metric	Schedule	Output	Environment
UDPLat-Poisson-XPercMean-NoCross	UDP_Latency	Poisson	Xth-Perc-mean	No-cross

Sub-registries (cont)



- Drawback: potential explosion of the regsitry (or lack of maintanance of the registry because the workload)

Changes since v00

- Metric sub-registry defined using RFC6390 format.
- Metric sub-registry now defined as:
  - Metric Name
  - Metric Definition:
  - Method of Measurement or calculation:
  - Units of Measurement.
  - Measurement Accuracy:
- Previously only defined as a pointer to the relevant spec, now include pointers to the relevant sections of the before mentioned spec