Network Working Group
Internet-Draft
Intended status: Informational
Expires: 18 November 2026

L. Melegassi
Catellix

17 May 2026

Multi-Vantage Path Snapshot (MVPS): A Canonical Bundle Format for Coordinated Traceroute Measurements
draft-melegassi-ippm-mvps-bundle-00

Abstract

This document specifies the Multi-Vantage Path Snapshot (MVPS) bundle format, a focused envelope for traceroute observations collected in coordination from two or more network vantages towards a common destination. MVPS defines a JSON serialization, a YANG 1.1 module, and a deterministic path-fingerprint algorithm enabling bit-reproducible auditing and cross-implementation interoperability.

MVPS is intentionally minimal in scope: it specifies a wire format and the algorithms required to produce it deterministically. Analytical metrics derived from MVPS bundles are out of scope.

MVPS complements the AURA architecture defined by RFC 9198, which specifies a measurement architecture but does not normatively specify the result format. MVPS is intended as one possible result format for AURA-style coordinated measurements. MVPS is also positioned relative to existing single-operator reporting formats (RIPE Atlas measurement JSON, CAIDA warts), which are not standardised and which do not provide a deterministic cross-implementation path identity.

Status of This Memo

Table of Contents

1. Introduction

Several existing systems coordinate traceroute measurements from distributed vantages towards a common target. Examples include RIPE Atlas, CAIDA Ark, ThousandEyes, and various operator-internal collection frameworks. These systems produce per-vantage observations whose joint analysis enables cross-vantage consistency checks (for example, applying a speed-of-light feasibility bound to pairs of observations of the same hop) and topology comparison (for example, quantifying path divergence between vantages).

At the time of writing, no widely deployed envelope provides all of the following properties simultaneously:

* A bit-reproducible canonical serialization that allows two independent implementations to produce identical artefacts from identical inputs.

* A deterministic path-fingerprint allowing inexpensive path identity comparison and change detection across rounds.

* An explicit coordination-window declaration with a bounded clock-skew uncertainty.

* A YANG module suitable for use in network management contexts, with a sibling JSON serialization aligned via [RFC7951].

This document specifies such an envelope: the Multi-Vantage Path Snapshot (MVPS) bundle format.

1.1. Scope and Non-Goals

In scope:

* Bundle serialization (JSON and YANG, aligned via RFC 7951).
* Deterministic path-fingerprint algorithm.
* Coordination-window semantics with explicit uncertainty.
* Conformance test vectors.
* Privacy and security considerations for the envelope.

Out of scope (see also Appendix E):

* Definition of analytical metrics or anomaly detection.
* Failure classification, regime detection, or operational dashboards.
* Probing protocol details (this document describes a reporting envelope, not a measurement protocol).
* Comparison frameworks against external observability platforms.

1.2. Relationship to RFC 9198 (AURA)

[RFC9198] defines AURA, an architecture for large-scale active network measurement using cooperating agents. AURA addresses agent registration, capability advertisement, measurement task distribution, and result collection, but does not mandate a specific on-the-wire format for the per-vantage results of a coordinated measurement.

MVPS does not replace AURA. An AURA-conforming collector MAY use MVPS as its bundle serialization format when coordinating traceroute measurements. Conversely, an MVPS producer is not required to operate within an AURA-managed deployment; standalone collectors are equally supported.

1.3. Relationship to Existing Reporting Formats

Several measurement platforms already publish results from coordinated traceroute campaigns in formats that overlap with MVPS in intent. This section briefly distinguishes MVPS from the most widely deployed of these.

1.3.1. RIPE Atlas Measurement Results

RIPE Atlas [RIPE-Atlas-Measurements] publishes per-probe traceroute results as JSON, with a stable per-platform schema. The format is operationally mature, widely used, and well-documented, but it is (a) operator-specific, with no normative reference outside RIPE NCC documentation; (b) not aligned with a YANG model; (c) does not define a deterministic cross-implementation path identity; and (d) does not declare a coordination window or a clock-skew uncertainty at the level of a bundle aggregating multiple probes.

MVPS aims to provide the same kind of operational utility in a vendor-neutral, RFC-style envelope with explicit coordination semantics and a fingerprint-level identity primitive.

1.3.2. CAIDA scamper / warts

CAIDA’s scamper tool [CAIDA-Warts] emits measurement results in the binary "warts" format. Warts is rich, efficient, and well-supported by the CAIDA tooling ecosystem, but it is binary, tool-specific, and not designed as a vendor-neutral interchange format. MVPS is text-based JSON (with an aligned YANG model) and is intended for cases where interchange and audit readability are primary concerns; warts and MVPS are therefore complementary rather than competing.

1.3.3. Other Platforms

Other coordinated-measurement platforms (commercial and research) typically use proprietary formats. MVPS is intended to be a candidate common denominator for cross-platform publication of bundle-level results.

2. Terminology and Conventions

This document uses the JSON data format [RFC8259], the YANG 1.1 data modelling language [RFC7950], common YANG data types from [RFC6991], RTT definitions consistent with [RFC2681], and UUIDs as defined in [RFC4122].

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119][RFC8174] when, and only when, they appear in all capitals, as shown here.

Vantage: A network position from which a traceroute measurement is initiated. Identified by a vantage-id (string) and OPTIONALLY by declared geolocation.

Snapshot: A single per-vantage observation within a bundle. Contains the ordered hop list, RTT samples, timestamps, and metadata.

Bundle: A collection of snapshots of a common destination, gathered within a single coordination window.

Path Fingerprint: A deterministic cryptographic hash (SHA-256) of the canonicalized hop sequence of a snapshot, as defined in Section 6.

Coordination Window: The time interval [start, end] within which all snapshots of a bundle MUST have been initiated. The width (end - start) declares the coordination tolerance.

Hop: An intermediate router observed in a traceroute, identified by its responding IP address (when available) or by an opaque-marker.

Opaque Hop: A hop for which the responding IP is not observable (for example, MPLS opaque LSRs, ICMP-disabled routers, or hops redacted per [RFC5837]).

3. Known Limitations of This Revision

This revision (-00) makes a number of explicit simplifying choices. They are listed here so that reviewers and implementers can evaluate them up front; future revisions MAY relax some of these.

3.1. Load-Balanced Paths

Standard traceroute can return different responding addresses at the same TTL when an intermediate router or load-balancer dispatches probes across multiple equal-cost next-hops. Paris traceroute [ParisTraceroute] and scamper avoid this by constraining the flow identifier across probes. In this revision, MVPS represents a hop as a single address (or opaque_marker); producers observing load-balanced behaviour SHOULD either (a) use a Paris-style probing discipline so that a single canonical hop is observed per TTL, or (b)

emit several distinct bundles, one per flow identifier. A multi-address-per-hop encoding is left to a future revision.

3.2. Router Identity vs. Interface Identity

A responding IP address identifies an interface, not a router. Two interfaces of the same router can produce different path-fingerprints even though the path traversed is operationally identical. This document treats path-fingerprints as a check on observational identity (same probed bytes produce same fingerprint), not on physical-router identity. Consumers requiring router- level identity SHOULD pair MVPS with an external alias resolution dataset.

3.3. ICMP Extensions and MPLS Labels

ICMP extensions per [RFC4884] and the MPLS label information that can be carried in them ([RFC4950]) are not modelled in this revision. Implementations that capture such extensions MAY publish them in a vendor-specific reverse-DNS extension field, but interoperability of such fields is out of scope.

3.4. Anycast Destinations

When the destination is an anycast address, different vantages can legitimately reach different sites, producing legitimately different path-fingerprints. The is_anycast flag carries this information so that consumers do not interpret divergent fingerprints as evidence of inconsistency in such cases.

4. Requirements

The following requirements apply to producers of MVPS bundles.

* *REQ-1.* An MVPS bundle MUST contain at least one snapshot.

* *REQ-2.* All snapshots in a bundle MUST share the same destination.

* *REQ-3.* Each snapshot MUST declare a vantage-id unique within the bundle.

* *REQ-4.* Each snapshot MUST declare a start-timestamp.

* *REQ-5.* The bundle MUST declare its coordination-window (start, end) as the envelope of its snapshots’ start and end timestamps.

* *REQ-6.* Each snapshot’s path-fingerprint MUST be computed using the algorithm in Section 6.

* *REQ-7.* Implementations MUST produce bit-identical bundles when given the same input traceroute data, by following the canonicalization rules in Section 5.

* *REQ-8.* Implementations SHOULD support hop redaction per [RFC5837].

* *REQ-9.* Implementations MUST NOT include vantage credentials, operator-internal hostnames, or personally identifiable information in the bundle.

* *REQ-10.* Implementations SHOULD declare a skew-bound-ms estimate (Section 7).

* *REQ-11.* In a hop, the fields address and opaque-marker MUST be mutually exclusive: exactly one of them is present in a well-formed hop.

5. MVPS Bundle Format

5.1. Canonicalization

To ensure bit-reproducibility (REQ-7), the JSON serialization of an MVPS bundle MUST apply the following canonicalization rules:

* UTF-8 encoding, without byte-order mark.

* Object keys sorted lexicographically (UTF-16 code unit order), at every nesting level.

* No insignificant whitespace.

* Numbers serialized in their shortest decimal representation; trailing zeroes after the decimal point are omitted, except as required to preserve declared precision (for example, declared_lat with 6 fraction digits).

* IPv4 addresses in dotted decimal.

* IPv6 addresses in the fully-expanded lowercase form (eight groups of four lowercase hexadecimal digits separated by ":"; no zero compression; no "::" shorthand). This is a *declared deviation* from [RFC5952], which mandates zero compression: see Section 5.2 for rationale.

* Timestamps in RFC 3339 format with timezone "Z" (UTC).

This document references the canonicalization rules of [RFC8785] (JSON Canonicalization Scheme) and deviates only where required for IP-address and timestamp normalization above.

5.2. IPv6 Canonicalization: Deviation from RFC 5952

[RFC5952] specifies a recommended textual form for IPv6 addresses that includes lowercase hexadecimal digits, suppression of leading zeros within each 16-bit group, and compression of consecutive all-zero groups using the "::" shorthand (with several disambiguation rules). Implementations of RFC 5952 differ in corner cases (selection of which all-zero run to compress when several are tied, handling of embedded IPv4, etc.), and these differences propagate directly into the SHA-256 path-fingerprint.

To eliminate this source of cross-implementation drift, implementations of this document MUST use the fully-expanded lowercase form for IPv6 in both the on-the-wire representation and the canonical hop string used to compute the path-fingerprint (Section 6). The fully-expanded form is unambiguous: each address has exactly one textual representation, derivable trivially from the 128-bit integer.

This deviation is intentional and is the only departure from RFC 5952 in this document. Implementations that render addresses for human display MAY apply RFC 5952 compression at presentation time, but MUST NOT use the compressed form in bundle output or in fingerprint input.

5.3. YANG Module

The canonical YANG 1.1 module catellix-mvps-bundle is provided in Appendix A. The JSON serialization (Section 5.4) is the RFC 7951 encoding of this module, with the canonicalization rules in Section 5.1 applied.

5.4. JSON Schema

A JSON Schema 2020-12 document expressing the same model is provided in Appendix B. Where the YANG module and the JSON Schema disagree, the YANG module is normative.

5.5. Examples

Two examples are provided in Appendix C: a minimal one-vantage bundle and a four-vantage bundle with opaque hops and an anycast destination.

6. Path Fingerprint Algorithm

6.1. Construction

For each snapshot, the path-fingerprint is computed as follows:

1. Build the canonical hop string CANON:

```
CANON = "v1|"
      || destination_canonical
      || "|"
      || join("|", [hop_token(h) for h in hops_in_index_order])
```

2. For each hop h, hop_token(h) is defined as follows.

   * If h.address is present: "ip:" || canonical_ip(h.address).
   * If h.opaque_marker is present: "op:" || h.opaque_marker.
   * Otherwise (well-formed bundles cannot reach this branch, per REQ-11): "*".

3. canonical_ip(addr) is the lowercase, dotted- decimal representation for IPv4, and the fully-expanded lowercase form for IPv6 as defined in Section 5.2 (a declared deviation from [RFC5952]).

4. destination_canonical equals canonical_ip(bundle.destination.address).

5. path_fingerprint = lowercase_hex(SHA-256(UTF-8(CANON))). SHA-256 is specified in [RFC6234].

The leading "v1|" identifies the fingerprint version. Future revisions of MVPS that change the fingerprint algorithm MUST use a different prefix.

6.2. Opaque Hops

Hops that respond but do not reveal an IP address (for example, MPLS opaque LSRs that do not generate ICMP Time Exceeded, or hops redacted per [RFC5837]) MUST be represented with an opaque-marker rather than a synthetic IP address. Implementations MUST NOT invent IP addresses for opaque hops.

Hops that did not respond at all SHOULD be represented with opaque_marker = "noresp".

6.3. Test Vectors

A set of conformance test vectors is provided in Appendix D. Conformant implementations MUST reproduce all listed fingerprints bit-identically.

7. Coordination Window Semantics

7.1. Window Width and Consumer Use

A bundle's snapshots are considered coordinated if all of them have start-timestamp values within the bundle's declared coordination-window [start, end]. The width (end - start) is the maximum coordination tolerance the producer is asserting. Smaller widths permit causality-sensitive analyses; larger widths only support coarser topology comparison.

Consumers MAY filter or reject bundles whose window width exceeds an analysis-specific budget. This document does not normatively prescribe specific numeric thresholds; the appropriate budget depends on the consumer's intended use of the bundle.

An optional informational hint (tolerance in the JSON serialization and the corresponding YANG leaf) MAY be carried to express the producer's intent at three reference orders of magnitude (sub-second, sub-minute, sub-five- minutes). The hint is non-normative; the only normative time bound on a bundle is its (end - start) value.

7.2. Clock Skew

Vantage clocks are assumed to be synchronized via NTPv4 [RFC5905] or PTP. A bundle MAY declare skew_bound_ms as the producer's best estimate (upper bound) of the maximum pairwise clock skew across its vantages at the time of bundle creation.

Bundles for which clock synchronization cannot be asserted MUST NOT declare a numeric skew_bound_ms value; consumers MUST NOT use such bundles for timing-sensitive cross-vantage analysis.

The skew_bound_ms field is a declaration of uncertainty, not a measurement guarantee.

7.3. Temporal Uncertainty

When publishing bundles for use by third parties, implementations SHOULD document how skew_bound_ms was estimated (for example, via chronyc tracking offsets, NTP root-distance, or out-of-band PTP statistics). This document does not prescribe a single estimation method.

8. Operational Considerations

8.1. Parallel Collection

A typical implementation initiates traceroute probes from all vantages within a short interval and aggregates the per-vantage results into a single bundle. The bundle's coordination-window MUST reflect the earliest start-timestamp and the latest end-timestamp across snapshots.

8.2. Missing Hops

Hops that did not respond SHOULD be represented with opaque_marker = "noresp" rather than being silently omitted. The hop list maintains 1-based indices that match the TTL used by the probe.

8.3. Typical Bundle Size

A bundle with 4 vantages, 10 hops per snapshot, and 3 RTT samples per hop typically serializes to a few kilobytes uncompressed. Implementations producing high collection rates SHOULD apply gzip or zstd compression at storage time.

8.4. Retention

This document does not prescribe retention policies. Operators publishing bundles in compliance with regional regulation should consult their privacy frameworks (for example, GDPR, LGPD).

9. Privacy Considerations

Bundles published as research datasets SHOULD apply the following measures:

* Operator-internal IP addresses SHOULD be replaced with documentation prefixes per [RFC5737] (192.0.2.0/24, 198.51.100.0/24) or with opaque_marker = "redacted".

* Hostnames MUST NOT appear in bundles (the format defines no field for hostnames).

* Declared geolocation SHOULD be reduced to a granularity not finer than 0.01 degrees (approximately 1 km).

* ASN values MAY be replaced with placeholders when publishing bundles where operator identity must remain confidential.

Implementers should be aware that temporal correlation of path-fingerprints across publications can be used to reidentify operators even when individual fields are anonymized.

10. Security Considerations

10.1. Reconnaissance Amplification

Coordinated multi-vantage probing can be misused as a reconnaissance amplifier: an attacker controlling a collection point can map internal network topology at a rate disproportionate to single-vantage probing. Implementations SHOULD apply rate-limiting on probe issuance, restrict the set of permitted destinations, and require authentication for control channels that trigger coordinated collection.

10.2. Bundle Poisoning

A hostile or compromised vantage may produce a snapshot containing fabricated hops or RTT samples. Consumers SHOULD NOT rely on any single vantage's claims without corroboration. Sanity checks computed by consumers (for example, comparing observed cross-vantage RTTs against speed-of-light feasibility bounds, or comparing path-fingerprints across redundant vantages) can detect a subset of fabrications; the specific analytical machinery is out of scope of this document. A future revision MAY define an optional per-snapshot cryptographic signature to bind a snapshot to its declared vantage.

10.3. Replay

Older bundles can be republished and presented as recent observations. Consumers SHOULD validate the coordination-window timestamps against an external time reference, and MAY chain bundles via cryptographic accumulators (out of scope of this document) to detect replay.

10.4. Information Disclosure

Declared latitude and longitude MUST respect the granularity guidance in Section 9. Path-fingerprints, being deterministic, can reveal the existence of recurring path patterns; this is intended behaviour but should be weighed against the operator’s exposure model before publication.

11. IANA Considerations

11.1. YANG Module Name

The YANG module shipped with this document uses the namespace urn:catellix:params:xml:ns:yang:catellix-mvps-bundle, which is under the author’s control and does not require IANA action.

If this document is adopted by an IETF working group, the module name SHOULD be renamed to ietf-mvps-bundle and the namespace to urn:ietf:params:xml:ns:yang:ietf-mvps-bundle in accordance with [RFC8407] section 4.3.1, and the following registration SHOULD be requested in the "YANG Module Names" registry ([RFC6020]):

Name: ietf-mvps-bundle

Namespace: urn:ietf:params:xml:ns:yang:ietf-mvps-bundle

Prefix: mvps

Reference: this document

11.2. Media Type Registration

IANA is requested to register the media type application/mvps-bundle+json with the following parameters:

Type name: application

Subtype name: mvps-bundle+json

Required parameters: none

Optional parameters: none

Encoding considerations: see RFC 8259

Security considerations: see Section 10

Interoperability considerations: this document

Published specification: this document

11.3. MVPS Bundle Capability Flags Registry

IANA is requested to create the "MVPS Bundle Capability Flags" registry, with assignment policy "Specification Required" ([RFC8126]). Initial contents: none.

12. Acknowledgements

The author thanks the IPPM working group for prior work on AURA ([RFC9198]) and active measurement primitives, and acknowledges related work on coordinated Internet measurement at RIPE Atlas, CAIDA, and ThousandEyes whose deployment experience motivated this format.

13. References

13.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>.

[RFC5737] Arkko, J., Cotton, M., and L. Vegoda, "IPv4 Address Blocks Reserved for Documentation", RFC 5737, January 2010, <https://www.rfc-editor.org/info/rfc5737>.

[RFC5837] Atlas, A., "Extending ICMP for Interface and Next-Hop Identification", RFC 5837, April 2010, <https://www.rfc-editor.org/info/rfc5837>.

[RFC5905] Mills, D., "Network Time Protocol Version 4: Protocol and Algorithms Specification", RFC 5905, June 2010, <https://www.rfc-editor.org/info/rfc5905>.

[RFC5952] Kawamura, S. and M. Kawashima, "A Recommendation for IPv6 Address Text Representation", RFC 5952, August 2010, <https://www.rfc-editor.org/info/rfc5952>.

[RFC6020] Bjorklund, M., "YANG - A Data Modeling Language for the Network Configuration Protocol (NETCONF)", RFC 6020, October 2010, <https://www.rfc-editor.org/info/rfc6020>.

[RFC6234] Eastlake, D. and T. Hansen, "US Secure Hash Algorithms (SHA and SHA-based HMAC and HKDF)", RFC 6234, May 2011, <https://www.rfc-editor.org/info/rfc6234>.

[RFC7950] Bjorklund, M., "The YANG 1.1 Data Modeling Language", RFC 7950, August 2016, <https://www.rfc-editor.org/info/rfc7950>.

[RFC7951] Lhotka, L., "JSON Encoding of Data Modeled with YANG", RFC 7951, August 2016, <https://www.rfc-editor.org/info/rfc7951>.

[RFC8126] Cotton, M., Leiba, B., and T. Narten, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 8126, June 2017, <https://www.rfc-editor.org/info/rfc8126>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, May 2017, <https://www.rfc-editor.org/info/rfc8174>.

[RFC8259] Bray, T., "The JavaScript Object Notation (JSON) Data Interchange Format", STD 90, RFC 8259, December 2017, <https://www.rfc-editor.org/info/rfc8259>.

[RFC8407] Bierman, A., "Guidelines for Authors and Reviewers of Documents Containing YANG Data Models", BCP 216, RFC 8407, October 2018, <https://www.rfc-editor.org/info/rfc8407>.

[RFC8785] Rundgren, A., Jordan, B., and S. Erdtman, "JSON Canonicalization Scheme (JCS)", RFC 8785, June 2020, <https://www.rfc-editor.org/info/rfc8785>.

[RFC9198] Alvarez-Hamelin, J., "Advanced Unidirectional Route Assessment (AURA)", RFC 9198, April 2022, <https://www.rfc-editor.org/info/rfc9198>.

13.2. Informative References

[CAIDA-Warts] CAIDA, "scamper / warts file format", https://www.caida.org/catalog/software/scamper/, 2024.

[ParisTraceroute] Augustin, B., Cuvellier, X., Orgogozo, B., Viger, F., Friedman, T., Latapy, M., Magnien, C., and R. Teixeira, "Avoiding traceroute anomalies with Paris traceroute", ACM IMC 2006, October 2006.

[RFC2681] Almes, G., Kalidindi, S., and M. Zekauskas, "A Round-trip Delay Metric for IPPM", RFC 2681, September 1999, <https://www.rfc-editor.org/info/rfc2681>.

[RFC4122] Leach, P., "A Universally Unique IDentifier (UUID) URN Namespace", RFC 4122, July 2005, <https://www.rfc-editor.org/info/rfc4122>.

[RFC4884] Bonica, R., "Extended ICMP to Support Multi-Part Messages", RFC 4884, April 2007, <https://www.rfc-editor.org/info/rfc4884>.

[RFC4950] Bonica, R., "ICMP Extensions for Multiprotocol Label Switching", RFC 4950, August 2007, <https://www.rfc-editor.org/info/rfc4950>.

[RFC6991] Schoenwaelder, J., "Common YANG Data Types", RFC 6991, July 2013, <https://www.rfc-editor.org/info/rfc6991>.

[RIPE-Atlas-Measurements] RIPE NCC, "RIPE Atlas Measurement Results (operational documentation)", https://atlas.ripe.net/docs/apis/result-format/, 2024.

Appendix A. YANG Module (Normative)

The full canonical module is distributed alongside this document as mvps-bundle.yang. An identical copy appears below.

```
module catellix-mvps-bundle {
  yang-version 1.1;
  namespace "urn:catellix:params:xml:ns:yang:catellix-mvps-bundle";
  prefix mvps;

  import ietf-yang-types {
    prefix yang;
    reference "RFC 6991: Common YANG Data Types";
  }

  import ietf-inet-types {
    prefix inet;
    reference "RFC 6991: Common YANG Data Types";
  }

  organization
    "Catellix (individual submission; not yet adopted by any IETF
     working group)";

  contact
    "Author:  Leonardo Melegassi
              <mailto:melegassi@catellix.com>";
```

```
  description
    "This YANG module defines the canonical data model for the
     Multi-Vantage Path Snapshot (MVPS) bundle envelope.  An MVPS
     bundle is a coordinated collection of per-vantage traceroute
     snapshots gathered within a bounded coordination window towards
     a common destination.

     This module is the normative source.  The JSON serialization
     follows RFC 7951 (JSON Encoding of Data Modeled with YANG)
     applied to this module, except that IP-address text forms
     follow the deviation declared in the companion specification
     (fully-expanded lowercase form for IPv6, NOT the compressed
     form of RFC 5952).

     This module does NOT define analytical metrics, anomaly
     detection, or failure classification.  Such functions are
     out of scope and are addressed in companion documents.

     The namespace 'urn:catellix:params:xml:ns:yang:' is used
     because this is an individual submission and not yet adopted
     by an IETF working group; the 'urn:ietf:params:xml:ns:yang:'
     prefix is reserved for adopted IETF modules per RFC 8407
     section 4.3.1.

     Copyright (c) 2026 Catellix and the contributors.
     Redistribution and use in source and binary forms, with or
     without modification, is permitted under the Revised BSD
     License (https://opensource.org/licenses/BSD-3-Clause).";

  revision 2026-05-17 {
    description
      "Initial individual submission (-00).";
    reference
      "draft-melegassi-ippm-mvps-bundle-00";
  }

  /*
   * Typedefs
   */

  typedef path-fingerprint {
    type string {
      length "64";
      pattern "[0-9a-f]{64}";
    }
    description
      "Lowercase hexadecimal representation of a SHA-256 digest
       computed over the canonical hop sequence of a snapshot, as
```

```
          defined in Section 5 of the MVPS specification.  Note that
          the fingerprint uses fully-expanded lowercase IPv6 form,
          which is a declared deviation from RFC 5952; this is
          intentional to remove ambiguity from the canonicalization
          step.";
      }

      typedef opaque-marker {
        type enumeration {
          enum mpls {
            description
              "Hop is opaque due to an MPLS Label-Switched Path that
               does not generate ICMP Time Exceeded.";
          }
          enum redacted {
            description
              "Hop has been deliberately redacted by the implementation,
               typically per RFC 5837 considerations.";
          }
          enum noresp {
            description
              "Hop did not respond within the implementation's timeout.";
          }
          enum filtered {
            description
              "Hop was filtered by an intermediate device (e.g., ACL
               dropping ICMP).";
          }
        }
        description
          "Reason category for an opaque hop, used when the responding
           IP address is unavailable.";
      }

      typedef coordination-tolerance {
        type enumeration {
          enum tight {
            description
              "Producer-asserted hint: window width is on the order of
               less than one second.  Intended for causality-sensitive
               consumers.  This is an INFORMATIONAL hint only; the only
               normative time bound is the (end - start) value of the
               coordination window.";
          }
          enum standard {
            description
              "Producer-asserted hint: window width is on the order of
               less than one minute.  Intended for general topology
```

```
            comparison.  INFORMATIONAL hint only.";
        }
        enum loose {
          description
            "Producer-asserted hint: window width is on the order of
             less than five minutes.  Intended for opportunistic
             aggregation.  INFORMATIONAL hint only.";
        }
      }
      description
        "Optional, non-normative producer hint about the intended
         use of a bundle's coordination window.  Consumers MAY use
         this hint to pre-filter bundles, but MUST rely on the
         numeric (end - start) value when making normative
         decisions.";
    }

    /*
     * Groupings
     */

    grouping vantage-identity {
      description
        "Identification of a measurement vantage.";

      leaf vantage-id {
        type string {
          length "1..64";
          pattern "[A-Za-z0-9_\\-]+";
        }
        mandatory true;
        description
          "Implementation-assigned identifier of the vantage, unique
           within a bundle.  Implementations MUST NOT include
           operator-internal hostnames or personally identifiable
           information.";
      }

      leaf declared-asn {
        type inet:as-number;
        description
          "Autonomous System Number declared by the vantage operator,
           when known.  Optional; absence MUST NOT be inferred as
           AS 0.";
      }

      leaf declared-lat {
        type decimal64 {
```

```
          fraction-digits 6;
          range "-90.0 .. 90.0";
        }
        units "degrees";
        description
          "Declared latitude (WGS-84) of the vantage.  Implementations
           publishing bundles SHOULD round to a granularity not finer
           than 0.01 degrees (approximately 1 km).";
      }

      leaf declared-lon {
        type decimal64 {
          fraction-digits 6;
          range "-180.0 .. 180.0";
        }
        units "degrees";
        description
          "Declared longitude (WGS-84) of the vantage.  See
           declared-lat for granularity guidance.";
      }
    }

    grouping rtt-sample {
      description
        "A single round-trip-time measurement.";

      leaf value-ms {
        type decimal64 {
          fraction-digits 3;
          range "0.0 .. 60000.0";
        }
        units "milliseconds";
        mandatory true;
        description
          "Round-trip time in milliseconds.";
      }

      leaf probe-sequence {
        type uint16;
        description
          "Per-hop sequence number of the probe that produced this
           sample, when meaningful.";
      }
    }

    grouping hop {
      description
        "A single hop observed in a traceroute.";
```

```
        leaf index {
          type uint8 {
            range "1..64";
          }
          mandatory true;
          description
            "1-based hop index in the path.";
        }

        leaf address {
          type inet:ip-address;
          description
            "Responding IP address of the hop, when observed.";
        }

        leaf opaque-marker {
          type opaque-marker;
          description
            "Reason for absence of address.  MUST be present if and
             only if address is absent.";
        }

        list rtt-samples {
          key "value-ms";
          uses rtt-sample;
          description
            "Observed RTT samples to this hop.  An empty list is
             permitted (indicating no successful probe).";
        }
      }

      grouping destination {
        description
          "Target of the coordinated measurement.";

        leaf address {
          type inet:ip-address;
          mandatory true;
          description
            "Destination IP address.  Conformant implementations MUST
             use the same address across all snapshots in a bundle.";
        }

        leaf asn {
          type inet:as-number;
          description
            "ASN of the destination, when known.";
        }
```

```
        leaf is-anycast {
          type boolean;
          default "false";
          description
            "True if the destination is known to be served by anycast.";
        }
      }

      grouping coordination-window {
        description
          "Temporal envelope of a bundle.";

        leaf start {
          type yang:date-and-time;
          mandatory true;
          description
            "Earliest snapshot start timestamp, in UTC.";
        }

        leaf end {
          type yang:date-and-time;
          mandatory true;
          description
            "Latest snapshot end timestamp, in UTC.  MUST NOT be earlier
             than start.";
        }

        leaf tolerance {
          type coordination-tolerance;
          description
            "Declared coordination tolerance intent.  Consumers MAY use
             this to filter bundles unsuitable for their analysis.";
        }

        leaf skew-bound-ms {
          type uint32;
          units "milliseconds";
          description
            "Implementation's best estimate of the maximum pairwise
             clock skew across vantages in this bundle.  Absence
             indicates that no estimate is provided and the bundle
             MUST NOT be used for timing-sensitive cross-vantage
             analysis.";
        }
      }

      grouping snapshot {
        description
```

```
    "A per-vantage observation in a bundle.";

  uses vantage-identity;

  leaf path-fingerprint {
    type path-fingerprint;
    mandatory true;
    description
      "Deterministic SHA-256 fingerprint of the canonical hop
       sequence.  Computed as specified in Section 5 of the MVPS
       specification.";
  }

  leaf start-timestamp {
    type yang:date-and-time;
    mandatory true;
    description
      "UTC timestamp at which this snapshot's traceroute was
       initiated.";
  }

    leaf end-timestamp {
      type yang:date-and-time;
      description
        "UTC timestamp of snapshot traceroute completion.";
    }

  list hops {
    key "index";
    ordered-by user;
    uses hop;
    min-elements 1;
    description
      "Ordered list of observed hops.";
  }
}

/*
 * Top-level container
 */

container bundle {
  description
    "A single MVPS bundle.";

  leaf bundle-id {
    type yang:uuid;
    mandatory true;
```

```
        description
          "UUID of this bundle. Implementations MUST generate it
           with sufficient entropy (RFC 4122).";
      }

      leaf schema-version {
        type string {
          pattern "mvps-bundle-v[0-9]+";
        }
        default "mvps-bundle-v1";
        description
          "Schema version identifier.  Conformant implementations of
           this document produce mvps-bundle-v1.";
      }

      container destination {
        uses destination;
        description
          "Common destination of all snapshots in this bundle.";
      }

      container coordination-window {
        uses coordination-window;
        description
          "Temporal envelope of this bundle.";
      }

      list snapshots {
        key "vantage-id";
        min-elements 1;
        uses snapshot;
        description
          "Per-vantage observations.  A bundle MUST contain at least
           one snapshot.  A bundle with exactly one snapshot is
           well-formed but provides no cross-vantage information.";
      }
    }
  }
```

Appendix B. JSON Schema (Informative)

The JSON Schema 2020-12 document is distributed alongside this document as mvps-bundle.schema.json. It is provided for tooling convenience; where it disagrees with the YANG module in Appendix A, the YANG module governs.

```
{
  "$schema": "https://json-schema.org/draft/2020-12/schema",
  "$id": "https://catellix.com/schemas/mvps-bundle/v1.json",
  "$comment":
    "Companion of YANG module (RFC 7951).",
  "title": "MVPS Bundle v1",
  "description":
    "JSON serialization (RFC 7951).",
  "type": "object",
  "additionalProperties": false,
  "required": [
    "bundle_id",
    "schema_version",
    "destination",
    "coordination_window",
    "snapshots"
  ],
  "properties": {
    "bundle_id": {
      "type": "string",
      "format": "uuid",
      "description": "RFC 4122 UUID, lowercase."
    },
    "schema_version": {
      "const": "mvps-bundle-v1",
      "description": "Schema version identifier."
    },
    "destination": {
      "$ref": "#/$defs/destination"
    },
    "coordination_window": {
      "$ref": "#/$defs/coordination_window"
    },
    "snapshots": {
      "type": "array",
      "minItems": 1,
      "items": {
        "$ref": "#/$defs/snapshot"
      },
      "description": "Per-vantage observations."
    }
  },
  "$defs": {
    "destination": {
      "type": "object",
      "additionalProperties": false,
      "required": ["address"],
      "properties": {
```

```
          "address": {
            "anyOf": [
              { "type": "string", "format": "ipv4" },
              { "type": "string", "format": "ipv6" }
            ],
            "description":
              "Destination IP. IPv6 fully-expanded (Sec 5.2)."
          },
          "asn": {
            "type": "integer",
            "minimum": 0,
            "maximum": 4294967295
          },
          "is_anycast": {
            "type": "boolean",
            "default": false
          }
        }
      },
      "coordination_window": {
        "type": "object",
        "additionalProperties": false,
        "required": ["start", "end"],
        "properties": {
          "start": {
            "type": "string",
            "format": "date-time",
            "description": "RFC 3339, UTC."
          },
          "end": {
            "type": "string",
            "format": "date-time",
            "description":
              "RFC 3339, UTC. MUST NOT be earlier than start."
          },
          "tolerance": {
            "type": "string",
            "enum": ["tight", "standard", "loose"]
          },
          "skew_bound_ms": {
            "type": "integer",
            "minimum": 0
          }
        }
      },
      "snapshot": {
        "type": "object",
        "additionalProperties": false,
```

```
  "required": [
    "vantage_id",
    "path_fingerprint",
    "start_timestamp",
    "hops"
  ],
  "properties": {
    "vantage_id": {
      "type": "string",
      "minLength": 1,
      "maxLength": 64,
      "pattern": "^[A-Za-z0-9_-]+$"
    },
    "declared_asn": {
      "type": "integer",
      "minimum": 0,
      "maximum": 4294967295
    },
    "declared_lat": {
      "type": "number",
      "minimum": -90.0,
      "maximum": 90.0
    },
    "declared_lon": {
      "type": "number",
      "minimum": -180.0,
      "maximum": 180.0
    },
    "path_fingerprint": {
      "type": "string",
      "pattern": "^[0-9a-f]{64}$",
      "description":
        "Lowercase hex SHA-256 of canonical hop seq (Section 6)."
    },
    "start_timestamp": {
      "type": "string",
      "format": "date-time"
    },
    "end_timestamp": {
      "type": "string",
      "format": "date-time"
    },
    "hops": {
      "type": "array",
      "minItems": 1,
      "items": { "$ref": "#/$defs/hop" }
    }
  }
```

```
    },
    "hop": {
      "type": "object",
      "additionalProperties": false,
      "required": ["index"],
      "properties": {
        "index": {
          "type": "integer",
          "minimum": 1,
          "maximum": 64
        },
        "address": {
          "anyOf": [
            { "type": "string", "format": "ipv4" },
            { "type": "string", "format": "ipv6" }
          ],
          "description":
            "Responding IP of the hop, when observed."
        },
        "opaque_marker": {
          "type": "string",
          "enum": ["mpls", "redacted", "noresp", "filtered"]
        },
        "rtt_samples": {
          "type": "array",
          "items": { "$ref": "#/$defs/rtt_sample" },
          "default": []
        }
      },
      "allOf": [
        {
          "description":
            "Exactly one of address or opaque_marker.",
          "oneOf": [
            {
              "required": ["address"],
              "not": { "required": ["opaque_marker"] }
            },
            {
              "required": ["opaque_marker"],
              "not": { "required": ["address"] }
            }
          ]
        }
      ]
    },
    "rtt_sample": {
      "type": "object",
```

```
        "additionalProperties": false,
        "required": ["value_ms"],
        "properties": {
          "value_ms": {
            "type": "number",
            "minimum": 0.0,
            "maximum": 60000.0
          },
          "probe_sequence": {
            "type": "integer",
            "minimum": 0,
            "maximum": 65535
          }
        }
      }
    }
  }
```

Appendix C. Examples

C.1. Minimal Single-Vantage Bundle

```
{
  "bundle_id": "11111111-1111-4111-8111-111111111111",
  "coordination_window": {
    "end": "2026-05-17T18:00:00.500Z",
    "start": "2026-05-17T18:00:00.000Z",
    "tolerance": "tight"
  },
  "destination": {
    "address": "192.0.2.1",
    "is_anycast": false
  },
  "schema_version": "mvps-bundle-v1",
  "snapshots": [
    {
      "declared_lat": -23.55,
      "declared_lon": -46.63,
      "hops": [
        {
          "address": "198.51.100.1",
          "index": 1,
          "rtt_samples": [{"value_ms": 1.234}]
        },
        {
          "address": "198.51.100.42",
          "index": 2,
          "rtt_samples": [{"value_ms": 5.678}]
        },
        {
          "address": "192.0.2.1",
          "index": 3,
          "rtt_samples": [{"value_ms": 12.345}]
        }
      ],
      "path_fingerprint":
"55d4f7f8d6a4c5b9a8df7e6c3b2a190f5e4d3c2b1a0e9f8d7c6b5a4938271605",
      "start_timestamp": "2026-05-17T18:00:00.000Z",
      "vantage_id": "V0"
    }
  ]
}
```

(The fingerprint value above is illustrative; conformance test vectors carry exact expected values.)

Appendix D. Conformance Test Vectors

At minimum 20 conformance test vectors are distributed alongside this document under test-vectors/v01.json through test-vectors/vNN.json. Each vector contains:

* a name identifying the case;
* an input snapshot;
* an expected_path_fingerprint.

Implementations conformant to this document MUST reproduce every expected_path_fingerprint bit-identically.

Appendix E. Out-of-Scope Topics

For clarity to reviewers, the following topics are explicitly out of scope for this document and will be addressed, if at all, in companion documents:

* Cross-vantage analytical metrics (speed-of-light feasibility bounds, Jensen-Shannon path divergence, topological-overlap measures, etc.). These belong in a companion Experimental-track document, not in the bundle-format specification.

* Anomaly detection, failure classification, regime detection, critical-slowing-down indicators, or any other consumer-side analytical layer.

* The probing protocol used to populate a bundle. MVPS is a reporting envelope; the probing protocol is independent (typical implementations use ICMP/UDP traceroute, but MVPS does not constrain this choice).

* Visualization, dashboarding, or operator-facing user interfaces.

Author’s Address

Leonardo Melegassi
Catellix
Brazil
Email: melegassi@catellix.com
URI: https://www.catellix.com