

BMWG
Internet-Draft
Intended status: Informational
Expires: 11 October 2026

L. Liu
Zhongguancun Laboratory
N. Geng
Huawei Technologies
9 April 2026

Benchmarking Methodology for Route Origin Validation (ROV)
draft-lg-bmwg-benchmarking-methodology-for-rov-01

Abstract

This document defines a benchmarking methodology for routers that implement ROV. The methodology focuses on device-level behavior, including processing of validated ROA payload (VRP) updates, the interaction between ROV and BGP, control-plane resource utilization, and the scalability of ROV under varying operational conditions. The procedures described here follow the principles and constraints of the Benchmarking Methodology Working Group (BMWG) and are intended to produce repeatable and comparable results across implementations.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <https://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on 11 October 2026.

Copyright Notice

Copyright (c) 2026 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components

extracted from this document must include Revised BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Revised BSD License.

Table of Contents

1. Introduction	2
1.1. Requirements Language	4
2. Scope and Goals	4
3. Terminology	4
4. Test Setup and Laboratory Environment	6
4.1. Test Topology	6
4.2. DUT Configuration Requirements	7
4.3. RTR Data Source Emulation	7
4.4. BGP Traffic Generation Requirements	7
4.5. Traffic Profile Parameters	8
5. Benchmarking Methodology	8
5.1. General Considerations	8
5.2. Test Control and Input Conditions	9
5.3. Metrics and Measurements	9
6. Benchmark Tests	10
6.1. ROV Update Processing Latency	10
6.2. ROV Validation Latency	12
6.3. BGP Convergence with ROV Enabled	13
6.4. VRP Scalability Tests	14
6.5. VRP Churn and Stress Tests	14
6.6. Resource Utilization	15
6.7. RTR Session Behavior Tests	16
7. Reporting Requirements	17
8. Security Considerations	19
9. IANA Considerations	19
10. Normative References	19
Acknowledgements	20
Authors' Addresses	20

1. Introduction

Route Origin Validation (ROV), as specified in [RFC6811], allows routers to use validated Route Origin Authorization (ROA) information, which is distributed via the RPKI-to-Router (RTR) protocol defined in [RFC8210], to classify BGP routes as Valid, Invalid, or NotFound. Deployments of ROV continue to increase across networks, and router vendors have implemented ROV processing as part of their control-plane functions.

While operational experience is growing, there is currently no standardized methodology for measuring the performance impact and behavioral characteristics of ROV on routing devices. As with other

protocol features evaluated by the Benchmarking Methodology Working Group (BMWG), a consistent and repeatable test framework is essential for:

- * Comparing router implementations,
- * Evaluating scalability under controlled conditions,
- * Characterizing the control-plane costs of ROV processing, and
- * Understanding how ROV influences BGP convergence and routing stability.

This document defines a benchmarking methodology for routers that implement ROV, which builds upon the foundational benchmarking principles defined in [RFC1242], [RFC2285], [RFC2544], [RFC2889], and [RFC3918]. The methodology focuses on the Device Under Test (DUT) and uses controlled, reproducible inputs to isolate the effects of ROV from external dependencies. In particular, the benchmarking framework assumes the presence of an RPKI-to-Router (RTR) update source, which may be an RPKI Cache Server or an RTR traffic generator capable of delivering synthetic Validated ROA Payloads (VRPs).

The objective of this document is to define a set of metrics and procedures to quantify:

- * The latency of ROV state updates within the DUT,
- * The impact of ROV on BGP control-plane performance,
- * The scalability of ROV processing under varying VRP and BGP table sizes, and
- * The control-plane resource utilization associated with enabling ROV.

By providing a consistent framework, this document enables vendors, operators, and researchers to evaluate ROV functionality under controlled and repeatable conditions, improving understanding of implementation performance and supporting informed deployment decisions.

1.1. Requirements Language

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.

2. Scope and Goals

This document specifies a laboratory-based benchmarking methodology for evaluating the performance of router implementations of ROV as defined in [RFC6811]. The scope of this benchmarking methodology includes:

- * ***ROV processing performance***: Measurement of the time and resources required for a router to process VRP updates received via the RTR protocol.
- * ***Impact on BGP control-plane performance***: Quantification of how enabling ROV affects BGP convergence times and routing table stability.
- * ***Scalability under controlled conditions***: Evaluation of the router's ability to handle large VRP sets, rapid VRP churn, and BGP updates influenced by ROV.
- * ***Resource utilization***: Measurement of system CPU utilization, system memory consumption, and relevant control-plane process load associated with ROV processing.

The goals of this document are:

- * To define a repeatable, controlled methodology for benchmarking ROV-enabled routers.
- * To provide standardized metrics that allow for comparison across implementations.

3. Terminology

The terminology used in this document follows the conventions of [RFC1242], [RFC2285], and subsequent BMWG publications. The following terms are used with specific meanings in the context of ROV benchmarking.

Route Origin Validation (ROV): A procedure defined in [RFC6811] that compares the origin AS of a BGP announcement with the set of authorized origins derived from validated ROA objects. ROV results in one of three states: Valid, Invalid, or NotFound.

Validated ROA Payload (VRP): The processed output from a relying party containing prefix-origin pairs that routers use for ROV decisions. VRPs are transported via the RPKI-to-Router (RTR) protocol.

RPKI-to-Router (RTR) Session: A protocol session between a router and an RPKI Cache Server. In benchmarking, RTR sessions may be emulated or generated using traffic/test tools to deliver synthetic VRP updates.

ROV Update Processing Latency: The time from when a router receives new VRP data (via RTR) until the updated ROV state is reflected in the router's local Routing Information Base (RIB) or implemented in routing decisions.

VRP-Triggered Revalidation Latency: The time interval between completion of VRP installation and the moment all affected prefixes have updated validation states.

BGP-Triggered ROV Validation Latency: The time interval between receipt of a BGP UPDATE message and completion of the ROV validation procedure for that route.

BGP Convergence Time: The time required for the router's control plane to process BGP updates and reach a stable routing state, while ROV validation is active.

Resource Utilization: System CPU utilization, system memory consumption, and, when observable, per-process utilization of the ROV process and the BGP or routing process while the router performs ROV-related tasks, including processing of VRP updates and applying ROV policy.

ROV Churn: A burst of VRP changes (e.g., many ROA additions or withdrawals) that may trigger significant re-validation and BGP recalculation, which is used in stress tests.

ROV Scalability Limit: The maximum number of VRPs, RTR sessions, or ROV-triggered BGP changes that the router can process while maintaining normal operational performance.

4. Test Setup and Laboratory Environment

This section describes the required test topology, equipment, DUT configuration, RPKI data emulation, and traffic generation conditions. The goal of the test environment is to isolate the DUT and subject it to clearly defined RPKI-RTR and BGP tests, while providing accurate timing and state measurements.

4.1. Test Topology

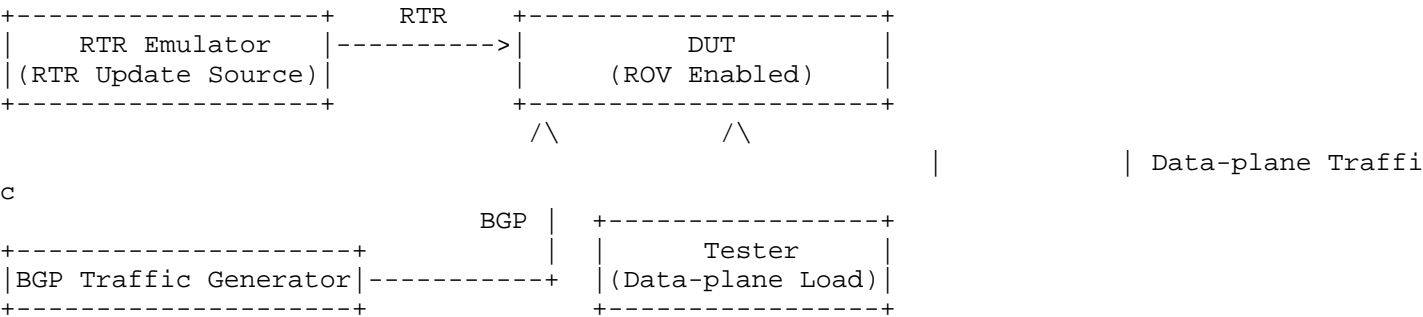


Figure 1: The test topology for ROV benchmarking.

The test topology consists of four primary components: the DUT, an RPKI-RTR update source, a BGP traffic generator, and a tester for generating data-plane traffic load. The DUT is a router equipped with ROV capabilities, supporting the RPKI-RTR protocol and applying ROV policies to received BGP routes. The RPKI-RTR update source may be either a real RPKI cache implementation running in isolated mode or a dedicated emulator capable of producing arbitrary VRP sets and update patterns. This RTR source connects directly to the DUT using the RTR protocol and provides controlled VRP updates, including serial increments, cache resets, and bursty or delayed update sequences.

The BGP traffic generator establishes one or more BGP peering sessions with the DUT and is responsible for delivering a full routing table together with controlled withdrawal or re-announcement events. Because IPv4 and IPv6 tables differ in scale and may exercise different implementation paths, the test setup must state the number of IPv4 routes and the number of IPv6 routes separately. A test MAY use a mixed baseline table (for example, 1,000,000 IPv4 routes and 250,000 IPv6 routes) or MAY benchmark each address family identifier (AFI) separately. In either case, the chosen route counts and AFIs under test must remain fixed across repeated runs for the same condition. The generator should be capable of presenting both stable baseline routing conditions and timed ROV-affected prefixes whose validation status will change in response to VRP updates. A

tester is connected to the DUT to introduce controlled data-plane load during benchmarking. When present, the tester SHOULD generate stable and deterministic traffic loads so that the impact of forwarding load on ROV processing can be evaluated. When data-plane load is applied, its rate, frame size, traffic pattern, and address-selection rules must be documented in the test report.

4.2. DUT Configuration Requirements

The DUT must be configured with ROV enabled on all BGP sessions receiving test routes. The router must establish a stable and fully functional RPKI-RTR session with the RTR emulator. To ensure that performance results are attributable solely to ROV behavior, all non-essential features on the DUT, such as additional routing protocols, unnecessary telemetry mechanisms, and unused services, should be disabled. Logging related to ROV may remain enabled for debugging purposes but must be rate-limited to avoid skewing CPU measurements or affecting test repeatability. All system parameters relevant to routing performance, such as multipath behavior or maximum-prefix limits, must be documented prior to testing.

4.3. RTR Data Source Emulation

The RTR emulator must be capable of generating synthetic VRP data sets with user-defined characteristics. This includes the ability to create arbitrary combinations of prefixes and ASNs, overlapping VRPs, conflicting VRPs, and other edge cases relevant to validation logic. The VRP datasets should mimic realistic global distributions where appropriate, but must also support scaling tests where VRP volumes are substantially higher than today's norm. The data source must further support generating controlled bursts of VRP updates, ranging from 100 to 10,000 VRP changes per second, and must allow for both additive updates and withdrawals.

4.4. BGP Traffic Generation Requirements

The BGP traffic generator must present the DUT with a stable baseline routing table prior to initiating any benchmark. This ensures that the DUT begins each test run in a known, converged state with predictable CPU and memory utilization. The generator must also provide a set of ROV-affected prefixes whose origin AS can be manipulated in concert with VRP updates from the RTR emulator. These prefixes should span a range of prefix lengths and originate from diverse ASes to reflect realistic routing conditions. The traffic generator must support deterministic convergence triggers, such as the precise injection of BGP updates following a VRP change or the simultaneous application of both BGP and VRP events.

4.5. Traffic Profile Parameters

When data-plane traffic is used, the following parameters SHOULD be specified:

- * Fixed frame size used for the measurement. For convergence measurements, a small fixed packet size SHOULD be used to improve time resolution. A 128-byte packet at Layer 3 is one practical choice. Other fixed sizes MAY be used when required by the traffic generator or encapsulation overhead, but the selected size MUST be documented.
- * Traffic rate in packets per second (PPS). For convergence measurements, the traffic SHOULD use a constant rate so that packet arrival times map directly to time resolution. Burst traffic MAY be used in stress scenarios, but such tests MUST be reported separately from constant-rate convergence tests.
- * Traffic pattern. For convergence measurements, constant-rate traffic is RECOMMENDED.
- * Source and destination IP address selection rules. When the setup sends one stream per tested prefix, the destination address for that stream SHOULD be selected from within the tested prefix. The first usable address is one valid example. The selection rule MUST be documented and applied consistently across runs.
- * Whether traffic matches ROV-affected prefixes.

Each frame size and PPS combination SHOULD be reported separately.

5. Benchmarking Methodology

This section describes the general methodology for benchmarking ROV behavior on a DUT. The goal is to ensure that all tests are repeatable, comparable across different environments, and representative of realistic deployment conditions. The methodology defines how to establish a controlled and stable test environment, how to specify and vary input conditions, and how to measure key performance metrics associated with ROV processing.

5.1. General Considerations

Before any measurements are taken, the DUT must reach a well-defined steady state in which the RPKI-RTR session is fully established, the VRP set has been completely synchronized, and the BGP control plane has converged. A warm-up period is recommended to eliminate any cold-start effects that could bias measurement results.

All sources of measurement noise should be avoided. Features such as logging, real-time telemetry export, or periodic background tasks can interfere with timing-sensitive measurements; therefore, such features should be disabled or rate-limited during benchmarking. CPU clock scaling, thermal throttling, or other variable-performance modes should be minimized if the test setup allows it.

5.2. Test Control and Input Conditions

Accurate benchmarking depends on precise control of the input conditions applied to the DUT. All tests should begin from a consistent baseline consisting of:

- * A predefined VRP set size (e.g., tens of thousands to millions of entries).
- * A stable and realistic baseline BGP RIB-in with IPv4 and IPv6 route counts documented separately (for example, 1,000,000 IPv4 routes and 250,000 IPv6 routes in a mixed table, or equivalent AFI-specific baselines).

From this baseline, input variables may be modified to stress different aspects of ROV behavior. These variables include the VRP churn rate, ranging from steady incremental updates to high-intensity bursts, and the type of RPKI-RTR updates provided to the DUT, such as incremental updates versus full-table refreshes. Each of these conditions may trigger different processing strategies within the DUT, and therefore must be explicitly controlled and documented.

5.3. Metrics and Measurements

Benchmarking ROV behavior requires collecting quantitative performance metrics that reflect how the DUT processes validation information and incorporates it into the BGP decision process. Therefore, this document proposes key performance metrics including ROV update processing latency, ROV validation latency, BGP convergence time, VRP storage size, system CPU and memory utilization, per-process utilization where available, and ROV state rebuild time.

ROV update processing latency measures the time from receipt of an RTR update (incremental or full) until the DUT has fully updated its internal validation state. This metric captures the efficiency of ROV data structures and algorithms.

ROV validation latency measures the time interval between a router's receipt of a BGP UPDATE message that contains a new or changed route, and the completion of the ROV procedure for that

route, producing a validation state of Valid, Invalid, or NotFound. This metric isolates the internal validation step, excluding the larger BGP convergence process, and provides insight into the responsiveness of the DUT's validation engine.

BGP convergence time with ROV enabled measures how long the DUT takes to converge on BGP prefixes whose validation states change due to VRP updates. This reflects the real operational behavior of ROV as it interacts with the control plane.

The **VRP storage size** inside the DUT should also be recorded to evaluate the scalability of the implementation when operating with large VRP datasets. Alongside this, **resource utilization** should be monitored to identify performance limits or resource-intensive operations triggered by ROV. At minimum, the benchmark SHOULD collect system-level CPU utilization and system-level memory consumption. When the DUT exposes process-level counters, the benchmark SHOULD also collect utilization for the ROV process and for the BGP process or main routing process.

A recovery-related measurement, **ROV state rebuild time** after RTR session reset, quantifies the time needed for the DUT to re-establish a complete and correct ROV validation state after an RTR session reset or cache outage. This metric reflects robustness and recovery behavior under fault or restart scenarios.

Finally, the DUT should be evaluated under high-pressure scenarios by measuring its behavior when processing VRP bursts, such as surges of 100-10,000 VRPs per second. This measurement reveals whether the implementation can sustain abrupt workload increases without dropping updates, stalling, or entering unstable states.

6. Benchmark Tests

This section defines the individual benchmark tests used to evaluate the performance and behavior of a DUT implementing ROV. Each test focuses on a specific aspect of the ROV processing pipeline, including VRP ingestion, validation, interaction with BGP, scalability limits, and robustness under stress and failure conditions. All tests assume the laboratory setup and input conditions described previously.

6.1. ROV Update Processing Latency

Objective: Measure the latency from the arrival of an RTR PDU until the new VRP information is installed in the DUT's internal ROV tables.

The *test procedures* for ROV update processing latency are listed below:

1. Prepare baseline state
 - * Establish RTR session between DUT and the RTR emulator.
 - * Preload DUT with a selected baseline VRP size (e.g., 100k VRPs).
 - * Ensure BGP is fully converged.
2. Inject controlled RTR update
 - * From the emulator, send a single incremental update modifying a known VRP.
 - * Alternatively, for full-refresh tests, send a full VRP set replacement PDU sequence.
3. Timestamp PDU transmission
 - * Record the exact moment the first update PDU is sent.
4. Monitor DUT internal state
 - * Use device instrumentation (API, CLI, or telemetry) to detect the exact moment the VRP table reflects the update.
 - * Confirm the VRP entry has been added, removed, or modified as expected.
5. Calculate latency
 - * The latency is the time difference between the moment the RTR PDU is sent and the moment the VRP is applied on the DUT.
6. Repeat for multiple VRP table sizes
 - * E.g., 50k, 100k, 500k, and 1M VRPs.
7. Repeat at least 10 times per condition
 - * Compute mean and standard deviation.

6.2. ROV Validation Latency

***Objective*:** Measure how long the DUT takes to apply updated VRPs to the validation states of affected BGP prefixes.

The ***test procedures*** for ROV validation latency are listed below:

1. Establish baseline

- * Load the baseline BGP table with IPv4 and IPv6 route counts stated explicitly (for example, 1,000,000 IPv4 routes and 250,000 IPv6 routes, or an AFI-specific baseline).
- * Ensure all prefixes have a known baseline validation state.

2. Select a controlled prefix set

- * Pick a set of prefixes (e.g., 1,000) whose origin AS is tied to specific VRPs.

3. Trigger validation update

- * Modify VRPs so that these prefixes change validation state (Valid->Invalid or Invalid->Valid).

4. Timestamp VRP installation completion

- * As measured in Section 6.1.

5. Monitor DUT validation table

- * Continuously query validation state for selected prefixes.
- * Note the timestamp when all prefixes reflect the updated state.

6. Compute latency

- * The validation latency is the time difference between the moment the VRP installation is completed and the moment all affected validation states have been updated.

7. Repeat with varying set sizes

- * E.g., 10 prefixes, 100 prefixes, 1,000 prefixes.

6.3. BGP Convergence with ROV Enabled

***Objective*:** Measure BGP convergence time for routes impacted by ROV state changes, and compare to BGP-only convergence without ROV.

The ***test procedures*** for BGP convergence with ROV enabled are listed below:

1. Prepare baseline
 - * Establish full-table BGP adjacency.
 - * Enable ROV on DUT.
 - * Ensure stable initial convergence.
2. Select test prefixes
 - * Choose prefixes that will transition from Valid to Invalid once VRP updates are applied.
3. Trigger VRP state change
 - * Send VRP modifications via RTR.
4. Monitor BGP behavior
 - * Observe best-path selection changes.
 - * Timestamp withdrawal or replacement of Invalid prefixes.
5. Measure convergence
 - * **Convergence Timer Starts:** The convergence timer SHOULD start at the timestamp when the DUT completes installation of the relevant VRP update.
 - * **Convergence Timer Ends:** The convergence timer SHOULD end when both the BGP RIB and FIB reach stable state.
6. Repeat test with ROV disabled
 - * Use identical routing changes for baseline comparison.
7. Record:
 - * Time to withdraw Invalid prefixes.

- * Time until new best paths stabilize.
- * Differences relative to ROV-disabled baseline.

6.4. VRP Scalability Tests

Objective: Evaluate DUT performance with varying VRP table sizes.

The **test procedures** for VRP scalability tests are listed below:

1. Generate VRP datasets at sizes:
 - * E.g., 50k, 100k, 500k, 1M.
2. Load each dataset into the RTR emulator.
3. For each dataset, measure:
 - * Full-table synchronization time.
 - * VRP update processing latency (from Section 6.1).
 - * ROV validation latency (from Section 6.2).
 - * System memory consumption.
 - * System CPU utilization during sync and steady state.
 - * Per-process utilization for the ROV process and the BGP or routing process, when available.
4. Record failures
 - * Session drops
 - * Timeouts
 - * Missing VRPs
 - * ROV process crashes
5. Repeat 10 times per size for statistical stability.

6.5. VRP Churn and Stress Tests

Objective: Stress-test the DUT under rapid VRP changes to measure stability, performance, and correctness.

The **test procedures** for VRP churn and stress tests are listed below:

1. Baseline setup

- * Load a stable VRP table (e.g., 500k).
- * Establish the baseline BGP table with IPv4 and IPv6 route counts stated explicitly.

2. Generate controlled churn patterns

- * Rapid add or remove spikes: 100-10,000 VRPs per second.
- * Sustained churn: continuous modifications for 5-10 minutes.
- * Mixed churn: adds, removes, and changes simultaneously.

3. Measure DUT behavior

- * VRP update backlog or queueing.
- * ROV validation delays.
- * System CPU spikes and, when available, spikes in the ROV process and the BGP or routing process.
- * BGP convergence degradation.
- * Missed or dropped VRP updates.

4. Check correctness

- * Verify that no stale or inconsistent ROV states remain.

5. Record crash, stall, or throttling events.

6.6. Resource Utilization

Objective: Measure resource consumption under various ROV workloads.

The **test procedures** for RTR session behavior tests are listed below:

1. Establish monitoring tools

- * System CPU sampling (100-500 ms interval).

- * System memory usage tracking.
- * Per-process sampling for the ROV process and the BGP or routing process, when available.
- * Hardware counters if available.

2. Measure under conditions

- * Idle ROV.
- * Full VRP sync.
- * VRP churn.
- * BGP convergence triggered by ROV events.

3. Record

- * System CPU load curves.
- * Peak and steady-state system memory consumption.
- * Process-level load curves for the ROV process and the BGP or routing process, when available.
- * Any evidence of saturation (e.g., 100% CPU, memory exhaustion).

4. Identify thresholds

- * Points where performance degrades or ROV processing becomes unstable.

6.7. RTR Session Behavior Tests

***Objective*:** Evaluate robustness and recovery of DUT under RTR failure and failover scenarios.

The ***test procedures*** for resource utilization are listed below:

1. Session reset test

- * Establish normal RTR session.
- * Trigger forced session reset from emulator.

- * Measure time to reestablish RTR session, ROV state rebuild time, time until validation state becomes consistent again.

2. Cache failover test

- * Configure DUT with two RTR servers (primary + secondary).
- * Terminate primary RTR connection.
- * Measure failover time and data consistency after switch.

3. Full resynchronization timing

- * From emulator, force a full Reset Query sequence.
- * Measure full VRP reload time.
- * Compare across different VRP scales.

4. Incremental update performance

- * Send controlled incremental PDUs.
- * Measure processing latency and correctness.
- * Introduce occasional malformed or unexpected PDUs to test robustness.

7. Reporting Requirements

An ROV benchmarking report MUST provide enough detail to allow reproducibility and meaningful comparison across different DUTs. Each report MUST include the following elements:

- * ***Test environment description***: The report MUST specify the DUT hardware and software versions, the testbed topology, and all ROV-related configuration parameters required to replicate the setup.
- * ***Input conditions***: The report MUST document the VRP set size, the RIB-in size with IPv4 and IPv6 counts stated separately, the presence and rate of VRP churn, and whether RTR updates were incremental or full.
- * ***Metrics and results***: Each measured metric MUST include its definition, a brief description of the measurement procedure, and results presented in tabular numerical form (including minimum, average, maximum, and at least P95 values). Graphs MAY be included for clarification.

- * ***Deviations and anomalies***: Any deviation from the expected behavior **MUST** be described, including the conditions under which it occurred and whether the test was repeated.
- * ***Summary of observations***: The report **MUST** include a concise summary of overall DUT performance, scalability limits observed, and any significant effects of enabling ROV on BGP behavior.

In addition, the report **MUST** include, at minimum, the following parameters:

- * DUT hardware model, CPU architecture, memory size, and software version.
- * Complete DUT configuration relevant to ROV and BGP.
- * Testbed topology description.
- * VRP table size.
- * VRP churn rate.
- * RIB-in size, with IPv4 and IPv6 route counts stated separately.
- * AFIs under test (IPv4, IPv6, or mixed).
- * Number of RTR sessions.
- * RTR timer configuration.
- * Presence and parameters of data-plane traffic (if used), including fixed frame size, PPS rate, traffic pattern, and address-selection rule.
- * ROV policy mode (e.g., reject Invalid).
- * System CPU sampling interval and any process-level sampling interval.
- * Measurement repetition count.

For each metric, the report **MUST** provide:

- * Metric definition.
- * Measurement method.
- * Minimum, average, maximum, and at least P95 values.

* Number of samples collected.

8. Security Considerations

This document defines a benchmarking methodology for evaluating ROV on routing devices. As such, it does not introduce new protocols, modify existing security mechanisms, or create new vulnerabilities within the RPKI system or BGP itself. All benchmarking activities are intended to take place in isolated laboratory environments. Nevertheless, a number of security considerations apply to the execution and interpretation of the tests described in this document.

Benchmarking ROV necessarily involves the generation, manipulation, and replay of RPKI objects. These test artifacts **MUST NOT** be injected into production RPKI repositories, production RPKI caches, or live BGP routing systems. Test-generated RPKI data sets **SHOULD** be clearly separated from real-world trust anchors, and laboratory RPKI caches **SHOULD** use isolated test Trust Anchors to prevent accidental propagation.

Similarly, BGP routing information used in the tests including simulated full tables, invalid prefixes, or artificially crafted origin-AS combinations, **MUST NOT** leak into production routing domains. All BGP sessions used for testing **MUST** be confined to a closed environment without external connectivity.

Tests involving stress conditions, such as high churn rates or large-scale VRP updates, may cause elevated CPU or memory consumption on the DUT. Operators performing such tests **SHOULD** ensure that the DUT is not simultaneously connected to any production network to avoid unintended service degradation.

9. IANA Considerations

This document has no actions for IANA.

10. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/rfc/rfc2119>>.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/rfc/rfc8174>>.

- [RFC6811] Mohapatra, P., Scudder, J., Ward, D., Bush, R., and R. Austein, "BGP Prefix Origin Validation", RFC 6811, DOI 10.17487/RFC6811, January 2013, <<https://www.rfc-editor.org/rfc/rfc6811>>.
- [RFC8210] Bush, R. and R. Austein, "The Resource Public Key Infrastructure (RPKI) to Router Protocol, Version 1", RFC 8210, DOI 10.17487/RFC8210, September 2017, <<https://www.rfc-editor.org/rfc/rfc8210>>.
- [RFC1242] Bradner, S., "Benchmarking Terminology for Network Interconnection Devices", RFC 1242, DOI 10.17487/RFC1242, July 1991, <<https://www.rfc-editor.org/rfc/rfc1242>>.
- [RFC2285] Mandeville, R., "Benchmarking Terminology for LAN Switching Devices", RFC 2285, DOI 10.17487/RFC2285, February 1998, <<https://www.rfc-editor.org/rfc/rfc2285>>.
- [RFC2544] Bradner, S. and J. McQuaid, "Benchmarking Methodology for Network Interconnect Devices", RFC 2544, DOI 10.17487/RFC2544, March 1999, <<https://www.rfc-editor.org/rfc/rfc2544>>.
- [RFC2889] Mandeville, R. and J. Perser, "Benchmarking Methodology for LAN Switching Devices", RFC 2889, DOI 10.17487/RFC2889, August 2000, <<https://www.rfc-editor.org/rfc/rfc2889>>.
- [RFC3918] Stopp, D. and B. Hickman, "Methodology for IP Multicast Benchmarking", RFC 3918, DOI 10.17487/RFC3918, October 2004, <<https://www.rfc-editor.org/rfc/rfc3918>>.

Acknowledgements

Many thanks to Giuseppe Fioccola and Christian Giese for their review, comments, and suggestions.

Authors' Addresses

Libin Liu
Zhongguancun Laboratory
Beijing
China
Email: liulb@zgclab.edu.cn

Nan Geng
Huawei Technologies
Beijing
China
Email: gengnan@huawei.com