Requirements for Monitoring RPKI-Related Processes on Routers Using BMP

draft-wang-grow-bmp-rpki-mon-reqs-00

Beijing Zhongguancun Lab, Tsinghua University

Shuhe Wang, Mingwei Xu, Yangyang Wang, Jia Zhang

2025.03

Content

What are RPKI-Related Processes on Routers → The Life cycle of RPKI-Related processes on routers

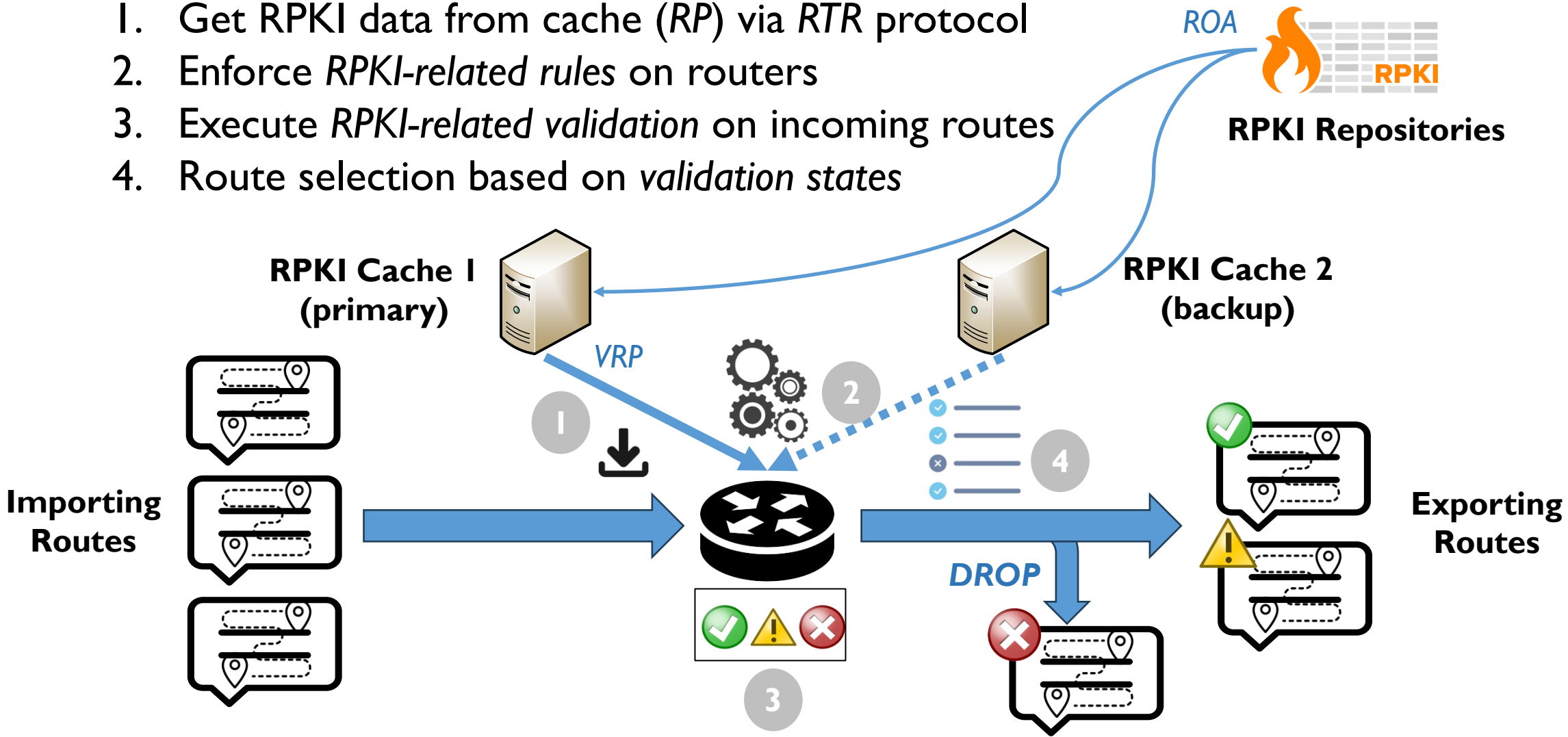
Why should RPKI be monitored → Case studies in which RPKI-related information is needed for monitoring

How should RPKI be monitored → Systematic requirements to extend BMP for RPKI

RPKI-Related Processes on Routers

The whole RPKI lifecycle on the router:

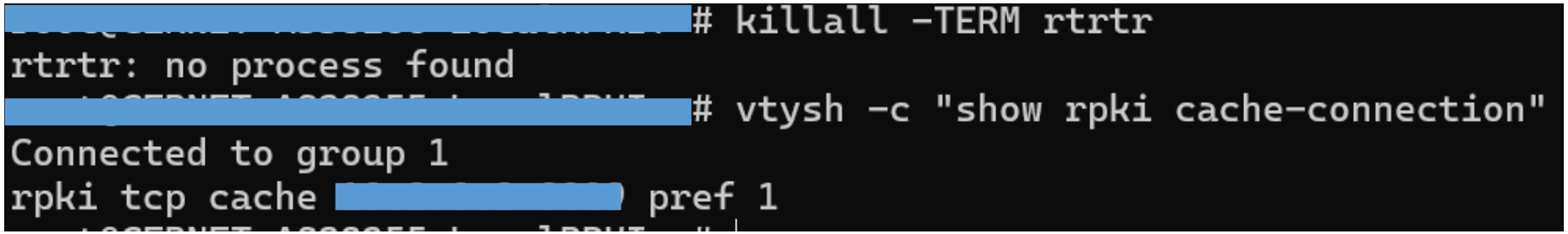
1. Get RPKI data from cache (*RP*) via *RTR* protocol
2. Enforce *RPKI-related rules* on routers
3. Execute *RPKI-related validation* on incoming routes
4. Route selection based on *validation states*



Necessites to monitor RPKI on routers

Case 1: RPKI caches fault (with FRR)

1. The RPKI cache fails, but the VRPs on the router don't get stale and stay the same as before
2. The RPKI cache may fail but soon get back to normal, but the VRPs on the router will still stay unchanged even if the cache get refreshed



The router also needs to know about the RTR connection details in time

Necessites to monitor RPKI on routers

Case 2: No action taken when new VRPs arrive (with FRR)

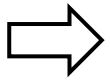
- The route-map is to filter invalid routes, but when cache gets refreshed, the filter clause for prefixes in renewed VRPs is still not invoked

```
route-map rpki-filter permit 10
 match ip address prefix-list acc-all

route-map rpki-filter permit 5
 match rpki invalid
 set community no-export
```

Route-map configuration

Route-map invocation before & after VRPs refresh



It will be normally invoked only after ***manual*** *(soft) reconfiguration of the peer*

```
route-map: rpki-filter Invoked: 410 Optimization: enabled Processed Change: false
 permit, sequence 5 Invoked 0
  Match clauses:
    rpki invalid
  Set clauses:
    community no-export
  Call clause:
  Action:
    Exit routemap
 permit, sequence 10 Invoked 410
  Match clauses:
    ip address prefix-list acc-all
  Set clauses:
  Call clause:
  Action:
    Exit routemap
```

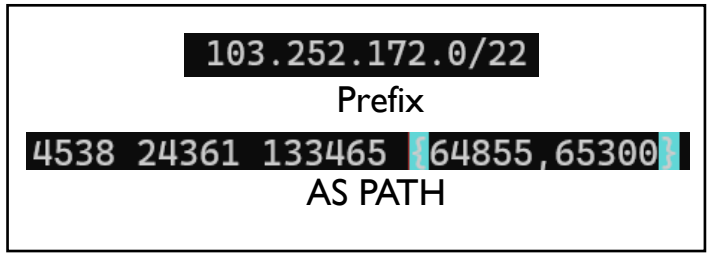
Any renewed RPKI-related routing policies / rules should be reported in time

Necessites to monitor RPKI on routers

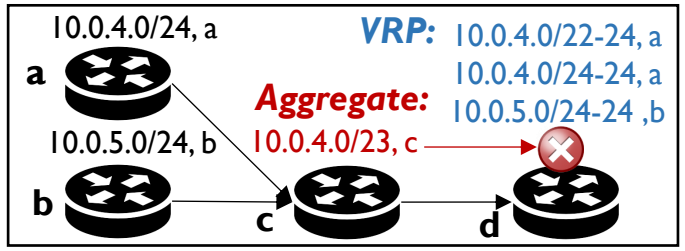
Case 3:Various reasons for getting validated as invalid

1. Invalid origin AS, and the route is truly announced unauthorizedly
2. Invalid origin AS, because an AS-SET is additionally appended to the AS-PATH
3. Invalid origin AS, but the route is altered in the middle (gets aggregated)
4. Invalid length, but the route is altered in the middle (gets de-/aggregated)

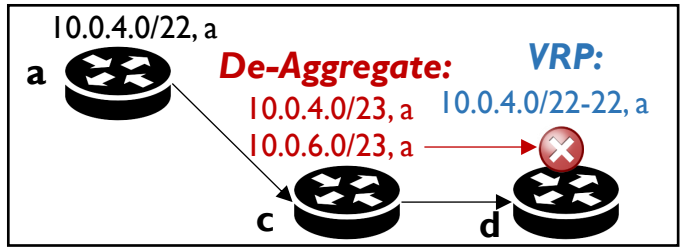
For case 3.3 and case 3.4, welcome to refer to ***draft-wang-sidrops-route-partial-visibility*** for detailed description



Case 3.2



Case 3.3



Case 3.4

Not only the RPKI validation state, but also why such state of the route should be reported

Current Limitation in BMP

Current practice: RPKI-related information not supported

- Standard BMP (RFC 7854): monitoring granularity can be per-router, per-peer and per-route
- Other useful extensions:
  - RFC 8671: extends to *RIB-OUT*
  - RFC 9069: extends to *Local RIB*
  - Draft-ietf-grow-bmp-tlv: self-defined types of data can be sent via *TLV*

Current proposals covering RPKI: not specific and systematic

- Draft-ietf-grow-bmp-path-marking-tlv: extends with *route path state* and RPKI-invalid is one of the states
- Draft-ietf-grow-bmp-rel: extends with *route event* and RPKI-invalid is one of the events
- Draft-ietf-grow-bmp-bgp-rib-stats: extends statistical report message with *counts related to RPKI validation*

Requirements for Supporting RPKI in BMP

Requirement 1: Monitor retrieval of RPKI data from caches

- **Purpose:**
  - Rapid detection and response to faults or outages in cache connectivity
- **Monitor Granularity:**
  - Per-RTR-connection
- **Monitored fields:**
  - RTR connection parameters
    - E.g.: RTR protocol version, TCP connection type, IP address and Port of RPKI cache, etc.
  - Synchronization details
    - E.g.: total number of RPKI records, synchronization state, timestamp of the last successful synchronization, etc.
  - Error metrics
    - E.g.: counts of timeouts, failed synchronization attempts, etc.

Requirements for Supporting RPKI in BMP

Requirement 2: Convey newest configuration of RPKI policies

- **Purpose:**
  - Ensure correct implementation of RPKI-based policies and prompt detection of misconfigurations
- **Monitor Granularity:**
  - Per-peer (or per-router if the configuration is enforced globally to all peers)
- **Monitored fields:**
  - Enable status of RPKI validation mechanisms
    - Whether the peer is enabled with ROV/ASPA
  - Validation rules derived from retrieved RPKI data
    - Total set of VRPs/ASPA entries to use for the peer
  - Configured actions for non-valid (invalid and not-found) routes
    - E.g.: filtering, priority reduction, tagging, or no action.

Requirements for Supporting RPKI in BMP

Requirement 3: Detail validation of routes using RPKI

- **Purpose:**
  - Aid for accurate troubleshooting, verification against unexpected RPKI validation outcomes, and identification of implementation errors of RPKI policies
- **Monitor Granularity:**
  - Per-peer, Per-route
- **Monitored fields:**
  - Statistical summaries of validation outcomes, aggregated per peer
    - Counts of routes in each validation state
    - Other optional statistics, such as the number of routes filtered because of RPKI validation
  - The validation state of each route, along with the specific reason for that state
    - Example reasons: a prefix length exceeding the ROA's maxLength, an origin AS mismatch with the ROA, or an AS path violating ASPA customer-provider relationships.
    - The message should include the relevant VRP/ASPA entry that leads to the invalidation

Requirements for Supporting RPKI in BMP

Requirement 4: Record the impact of RPKI validation on routing decisions

- **Purpose:**
  - Provide visibility of intended outcomes and unintended side effects of RPKI validation
- **Monitor Granularity:**
  - Per-route-event (only for interested routes)
- **Monitored fields:**
  - For routes demoted due to RPKI
    - the new best route selected with RPKI enabled
  - For routes promoted due to RPKI
    - the best route that would have been selected without RPKI
  - More details on the route pairs
    - E.g. : prefix, path, validation state, reason of validation, policy action enforced on such validation state, etc.

Conclusions and Discussions

Requirements summary

- To fully understand RPKI activities on routers, additional types of messages should be added to separately monitor:
  - Retrieval of RPKI data from caches;
  - Configuration of RPKI policies;
  - Validation of routes using RPKI;
  - Impact of RPKI validation on routing decisions.

Future possibilities

- MAYBE a *new unified* BMP message type for RPKI monitoring is better for extension and more systematic, other then 4 separate message types for each specific stage
- Or MAYBE a *new* router monitoring protocol for RPKI alone is better to extend then existing BMP

Thank you!

Welcome to discuss with me at wangsh@mail.zgclab.edu.cn