sidrops
Internet-Draft
Intended status: Standards Track
Expires: 16 October 2025

N. Geng
S. Zhuang
Huawei
Y. Fu
China Unicom
M. Huang
Zhongguancun Laboratory

14 April 2025

Selective Synchronization for RPKI to Router Protocol
draft-geng-sidrops-rtr-selective-sync-05

Abstract

The RPKI-to-Router (RTR) protocol synchronizes all the verified RPKI data to routers. This document proposes to extend the existing RTR protocol to support selective data synchronization. Selective synchronization can avoid some unnecessary synchronizations. The router can obtain only the data that it really needs, and it does not need to save the data that are not needed.

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 16 October 2025.

Copyright Notice

Table of Contents

1. Introduction

The RPKI-to-Router (RTR) protocol is a simple but reliable approach, which help synchronize the validated RPKI data from a trusted cache to routers. There are already several versions of the protocol [RFC6810][RFC8210][I-D.ietf-sidrops-8210bis]. The supported types of data that can be transferred increase, which is shown in Table 1.

Version 0	Version 1	Version 2
IPv4 Prefix	IPv4 Prefix	IPv4 Prefix
IPv6 Prefix	IPv6 Prefix	IPv6 Prefix
	Router Key	Router Key
		ASPA

Table 1: Supported data types in different versions of the RTR protocol

The RTR protocol keeps the synchronization of all types of data, and selective synchronization is not supported. However, routers may be interested in a part of data types, instead of all. In such cases, storing unused data on the router is unreasonable, and synchronizing

all types of data will induce some unnecessary transmission and storage overhead. Since multiple types of data are transmitted together, the router cannot use any type of these data unless it waits for all data to complete transmission. Furthermore, there may be more types of data in the cache, which makes the above issue more significant and worse. The followings are example types, and some of them may be possibly supported in the RTR protocol in the future:

* Secured Routing Policy Specification Language (RPSL) [RFC7909]

* Signed Prefix Lists [I-D.ietf-sidrops-rpki-prefixlist]

* Autonomous Systems Cones [I-D.ietf-grow-rpki-as-cones]

* Mapping Origin Authorizations (MOAs) [I-D.xie-sidrops-moa-profile]

* Signed SAVNET-Peering Information (SiSPI) [I-D.chen-sidrops-sispi]

* Path validation with RPKI [I-D.van-beijnum-sidrops-pathrpki]

* Signed Groupings of Autonomous System Numbers [I-D.spaghetti-sidrops-rpki-asgroup]

* Autonomous System Relationship Authorization (ASRA) [I-D.sriram-sidrops-asra-verification]

This document describes the synchronization problem of the RTR protocol and provides some possible solutions.

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. Problem Statement

The RTR protocol does not distinguish data types in the cache. Different types of data share one serial number and one End of Data PDU. When the Relying Party (RP) synchronizes the cache to the router, various PDUs, such as IPv4 Prefix, IPv6 Prefix, Router Key, and ASPA, are mixed. The router cannot select one or more really required PDUs or deny receiving a certain kind of PDU. For example, if the router supports RTR v2 but does not support or enable ASPA, the ASPA PDU messages will still be transmitted. Another example is the router in an IPv6-only network unreasonably has to receive IPv4

RPKI data. Overall, the transimitted Data PDU type cannot be flexibly selected by the router.

The negative effects of the above problem are as follows:

* Storing unused data on the router, which is unreasonable.

* Unnecessary transmission and storage overhead.

* Inefficient end-of-transmission acknowledgment. Multiple types of data are transmitted together. The router cannot use any type of these data unless it waits for all data to complete transmission.

The above negative effects will become worse when there are more kinds of RPKI data available [I-D.van-beijnum-sidrops-pathrpki][I-D.ietf-grow-rpki-as-cones][I-D.spaghetti-sidrops-rpki-asgroup]. The main problem of the RTR protocol is the lack of selective synchronization capability.

How about using different RTR versions for controlling the synchronized data, e.g., using RTR v0 if ASPA data are unwanted? This is not a good solution. First, the data selection is restricted to RTR versions and thus is not flexible either. Second, upgrading the version of RTR for future new RPKI data is not a proper choice, which is also a problem of existing RTR design. Specifically, existing RTR protocol has low extension capability. When there are new PDUs defined for transmission, a new RTR version needs to be issued. The new version protocol is not well compatible with the older ones, which induces some challenges on version negotiation, protocol implementation, and deployment. This document will primarily focus on the solving the inflexible synchronization problem. How to define an extensible protocol needs to be further discussed.

3. Preliminary Solutions

This section preliminarily proposes some independent solutions for achieving selective synchronization in the RTR protocol, while trying to keep the protocol’s simplicity. A new protocol version may not necessarily be required.

3.1. Subscribing Data PDU

Define a new type of PDU called Subscribing Data PDU. The new PDU will indicate the data types that the router is interested in. An example format of the PDU is shown in Figure 1. The field of PDU type is TBD. The Data Type fields indicate the interested data types (i.e., 4: IPv4 Prefix, 6: IPv6 Prefix, 9: Router Key, 11: ASPA).

The router can send the Subscribing Data PDU to the cache. After finishing the subscribing, the following PDUs, including Serial Notify, Serial Query, Reset Query, Cache Response, and Cache Reset, are only for the subscribed data. If the router wants to modify the subscription, a new Subscribing Data PDU can be sent for overwriting the previous subscription.

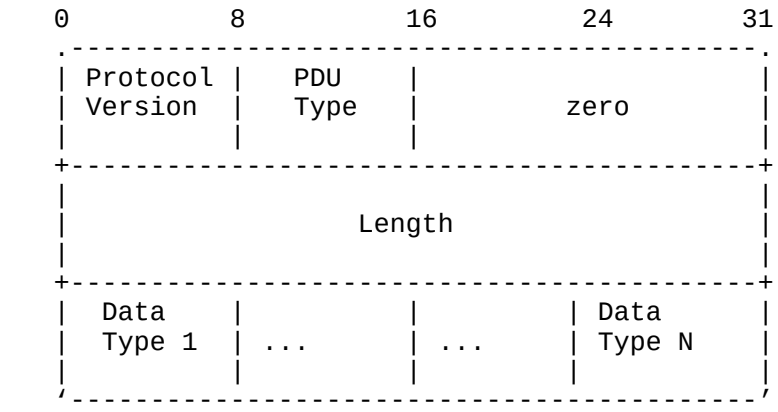

Figure 1: An example format of Subscribing Data PDU

3.2. PDUs with Data Type Field

The existing PDUs, including Serial Notify, Serial Query, Reset Query, Cache Response, and Cache Reset, can be extended to carry the Data Type field. The values of the Data Type field can be 4 for IPv4 Prefix, for IPv6 Prefix, 9 for Router Key, and 11 for ASPA. An example format of the extended Serial Query PDU is shown in Figure 2. A router can send the extended Serial Query PDU for requesting a specific type of data.

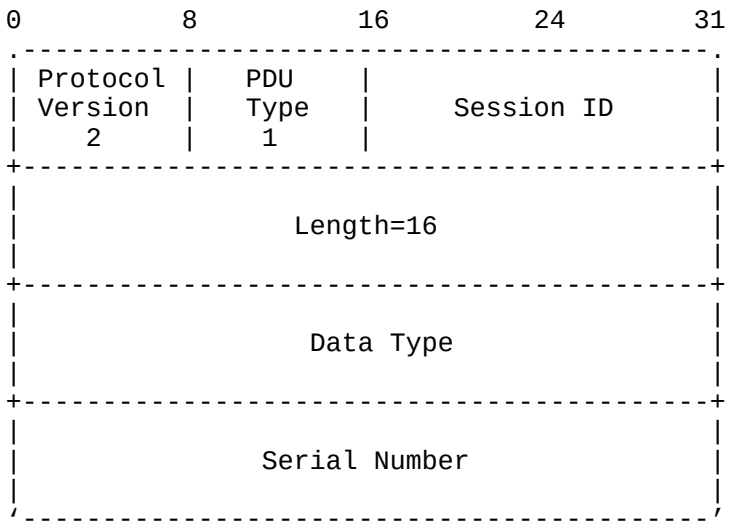

Figure 2: An example format of extended Serial Query PDU

3.3. End of Specific Data PDU

End of Data PDU tells the router that all the requested data are synchronized. The End of Specific Data PDU can be defined for indicating a specific type of data has been synchronized. An example format of End of Specific Data PDU is shown in Figure 3. The field of PDU type is TBD. The Data Type field indicate the interested data types (i.e., 4: IPv4 Prefix, 6: IPv6 Prefix, 9: Router Key, 11: ASPA).

The type of data specified in End of Specific Data PDU will become ready for use. The router does not need to wait for all the data to complete transmission before it can use the specified data.

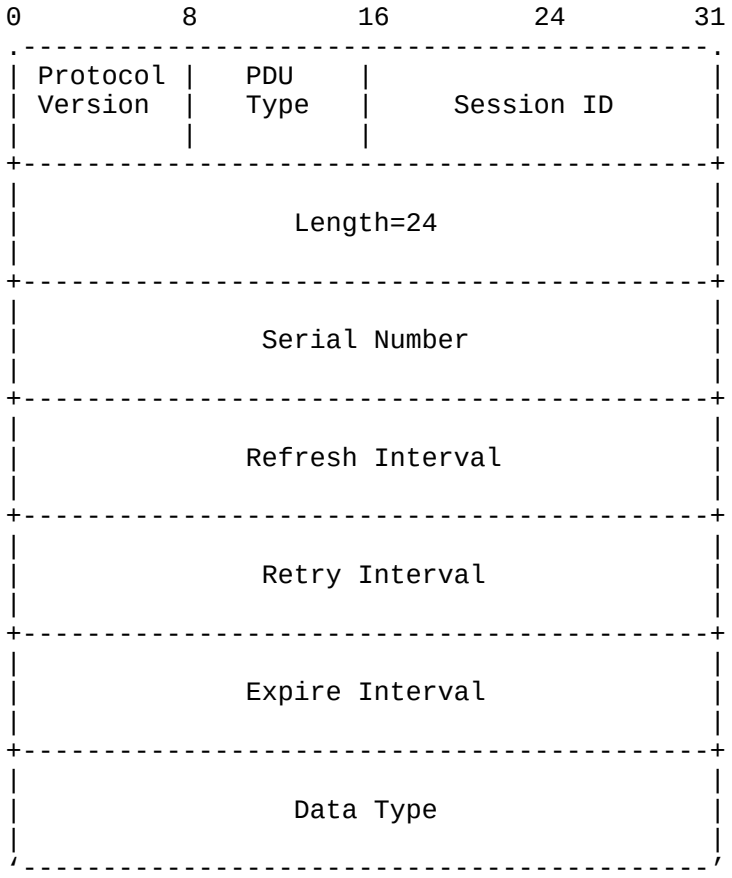

Figure 3: An example format of End of Specific Data PDU

4. Security Considerations

TBD

5. IANA Considerations

TBD

6. References

6.1. Normative References

[RFC6810] Bush, R. and R. Austein, "The Resource Public Key Infrastructure (RPKI) to Router Protocol", RFC 6810, DOI 10.17487/RFC6810, January 2013, <https://www.rfc-editor.org/info/rfc6810>.

[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/info/rfc8210>.

[I-D.ietf-sidrops-8210bis] Bush, R. and R. Austein, "The Resource Public Key Infrastructure (RPKI) to Router Protocol, Version 2", Work in Progress, Internet-Draft, draft-ietf-sidrops-8210bis-19, 10 April 2025, <https://datatracker.ietf.org/doc/html/draft-ietf-sidrops-8210bis-19>.

[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/info/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/info/rfc8174>.

6.2. Informative References

[RFC7909] Kisteleki, R. and B. Haberman, "Securing Routing Policy Specification Language (RPSL) Objects with Resource Public Key Infrastructure (RPKI) Signatures", RFC 7909, DOI 10.17487/RFC7909, June 2016, <https://www.rfc-editor.org/info/rfc7909>.

[I-D.van-beijnum-sidrops-pathrpki] van Beijnum, I., "Path validation with RPKI", Work in Progress, Internet-Draft, draft-van-beijnum-sidrops-pathrpki-00, 20 June 2019, <https://datatracker.ietf.org/doc/html/draft-van-beijnum-sidrops-pathrpki-00>.

[I-D.ietf-grow-rpki-as-cones]
Snijders, J., stucchi-lists@glevia.com, and M. Aelmans, "RPKI Autonomous Systems Cones: A Profile To Define Sets of Autonomous Systems Numbers To Facilitate BGP Filtering", Work in Progress, Internet-Draft, draft-ietf-grow-rpki-as-cones-02, 24 April 2020, <https://datatracker.ietf.org/doc/html/draft-ietf-grow-rpki-as-cones-02>.

[I-D.spaghetti-sidrops-rpki-asgroup]
Snijders, J. and F. Korsb辰ck, "A profile for RPKI Signed Groupings of Autonomous System Numbers (ASGroup)", Work in Progress, Internet-Draft, draft-spaghetti-sidrops-rpki-asgroup-00, 16 November 2022, <https://datatracker.ietf.org/doc/html/draft-spaghetti-sidrops-rpki-asgroup-00>.

[I-D.ietf-sidrops-rpki-prefixlist]
Snijders, J. and G. Huston, "A profile for Signed Prefix Lists for Use in the Resource Public Key Infrastructure (RPKI)", Work in Progress, Internet-Draft, draft-ietf-sidrops-rpki-prefixlist-04, 16 September 2024, <https://datatracker.ietf.org/doc/html/draft-ietf-sidrops-rpki-prefixlist-04>.

[I-D.xie-sidrops-moa-profile]
Xie, C., Dong, G., Li, X., Huston, G., and D. Ma, "A Profile for Mapping Origin Authorizations (MOAs)", Work in Progress, Internet-Draft, draft-xie-sidrops-moa-profile-06, 26 September 2024, <https://datatracker.ietf.org/doc/html/draft-xie-sidrops-moa-profile-06>.

[I-D.chen-sidrops-sispi]
Chen, L., Liu, L., Li, D., and L. Qin, "A Profile of Signed SAVNET-Peering Information (SiSPI) Object for Deploying Inter-domain SAVNET", Work in Progress, Internet-Draft, draft-chen-sidrops-sispi-03, 18 March 2025, <https://datatracker.ietf.org/doc/html/draft-chen-sidrops-sispi-03>.

Authors' Addresses

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

Shunwan Zhuang
Huawei
Beijing
China
Email: zhuangshunwan@huawei.com

Yu Fu
China Unicom
Beijing
China
Email: fuy186@chinaunicom.cn

Mingqing Huang
Zhongguancun Laboratory
Beijing
China
Email: huangmq@mail.zgclab.edu.cn