

SIDROPS
Internet-Draft
Intended status: Informational
Expires: 26 August 2025

D. Ma
ZDNS
Y. Li
CNIC-CAS
Y. Zhang
Peng Cheng Laboratory
S. Zhang
China Mobile
22 February 2025

RPKI Relying Party with Distributed Systems of Synchronization Nodes
draft-madi-sidrops-rp-dssn-01

Abstract

This document describes a current practice of establishing an RPKI relying party with distributed systems of synchronization nodes.

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 26 August 2025.

Copyright Notice

Copyright (c) 2025 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	3
2. Architecture	3
3. Components of Distributed Synchronization Systems	4
3.1. Console Node (RCN)	5
3.1.1. Data Storage Module	6
3.1.2. Task Scheduling Module	7
3.1.3. Task Dispatch Module	7
3.1.4. Data Aggregation Module	7
3.2. RP Synchronization Nodes (RSN)	7
3.2.1. Task Receiving Module	8
3.2.2. Task Execution Module	8
3.2.3. Data Upload Module	9
4. Scheduling Algorithms of RPKI Data Synchronization	9
4.1. Random Task Assignment	9
4.2. Sequential Task Assignment	9
4.3. Network-Aware Task Assignment	9
4.4. Dynamic Task Assignment	9
4.5. RIR Region-Based Task Assignment	10
4.6. Designated Position Assignment	10
4.7. Comparison of Task Assignment Algorithms	10
5. IANA Considerations	11
6. Security Considerations	11
7. References	11
7.1. Normative References	12
7.2. Informative References	12
Acknowledgements	12
Authors' Addresses	12

1. Introduction

The relying party is a key role in the Resource Public Key Infrastructure (RPKI) ecosystem, whose technical requirements are listed in [RFC8897]. However, how to follow RFC8897 to deploy such a relying party system to make the RPKI data take effect to the routing control system is yet another issue that is worth operational concerns.

As the RPKI-based routing information validation is globally adopted, there is an increasing number of RPKI data published by more and more RPKI repositories, bringing in more complicated validation chains. Granted, a relying party system is always to be expected to timely get all the updates from global RPKI repositories, rapidly validate them and deliver to BGP speakers in time.

To meet this end, this document introduces a current practice of establishing an RPKI relying party (this RP) with distributed systems of synchronization nodes.

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. Architecture

In general, this RP is combined with a set of RP Synchronization Nodes (RSN) and a single RP Console Node (RCN).

The set of RP Synchronization Nodes (RSN) are deployed across the Internet in geographically diverse locations with respect to worldwide RPKI publication points (RPP). The RSN is functioned to take the responsibility of pulling RPKI data from RPPs designated by the RCN.

The single RP Console Node (RCN) is functioned to ascertain the updates from global RPPs, orchestrate RSNs to perform the task of downloading RPKI data respectively, and validate those data.

The overall system architecture is illustrated in Figure 1.

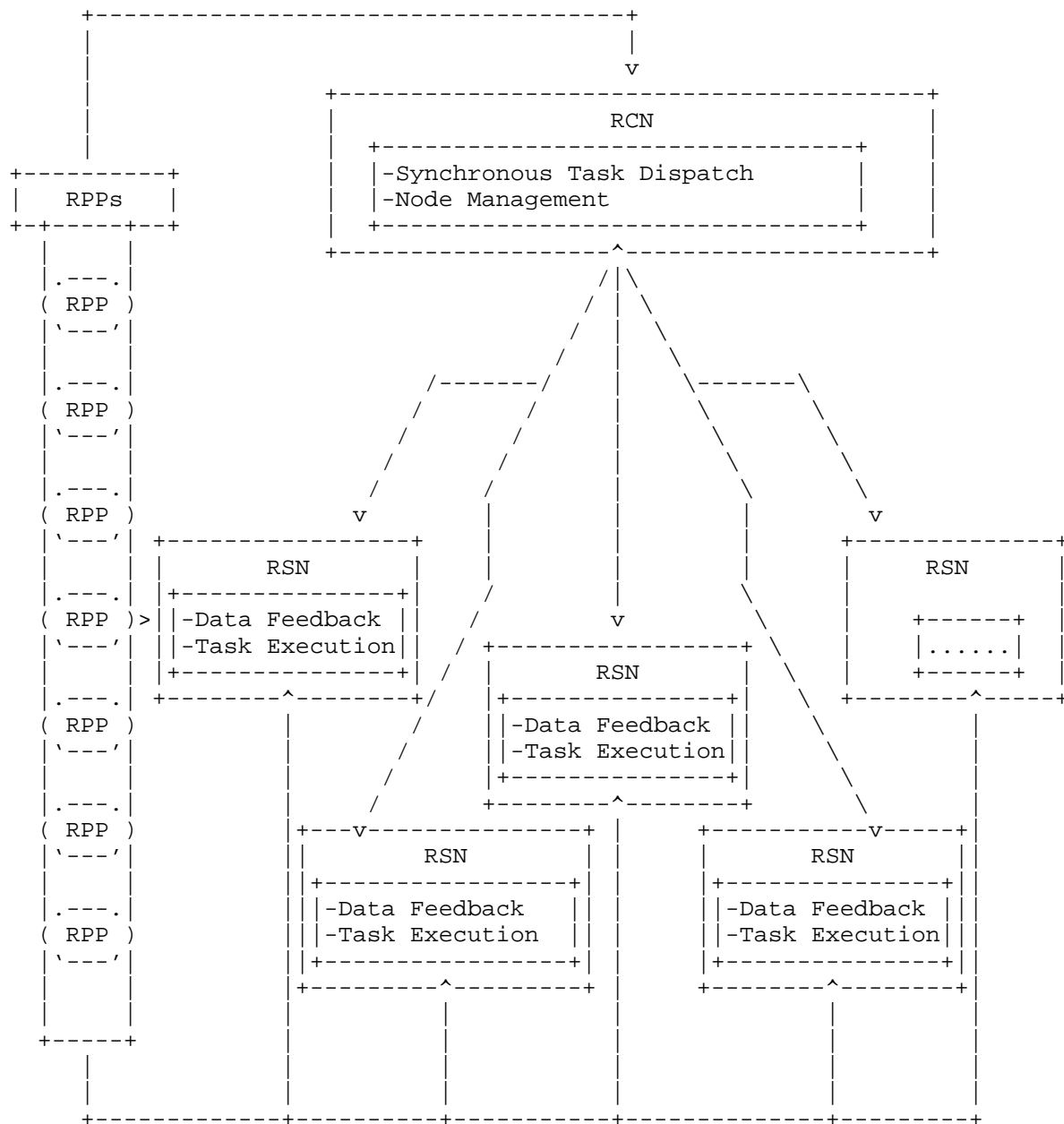


Figure 1: RPKI Relying Party System Architecture

3. Components of Distributed Synchronization Systems

3.1. Console Node (RCN)

The single RCN of this RP is constructed as in Figure 2.

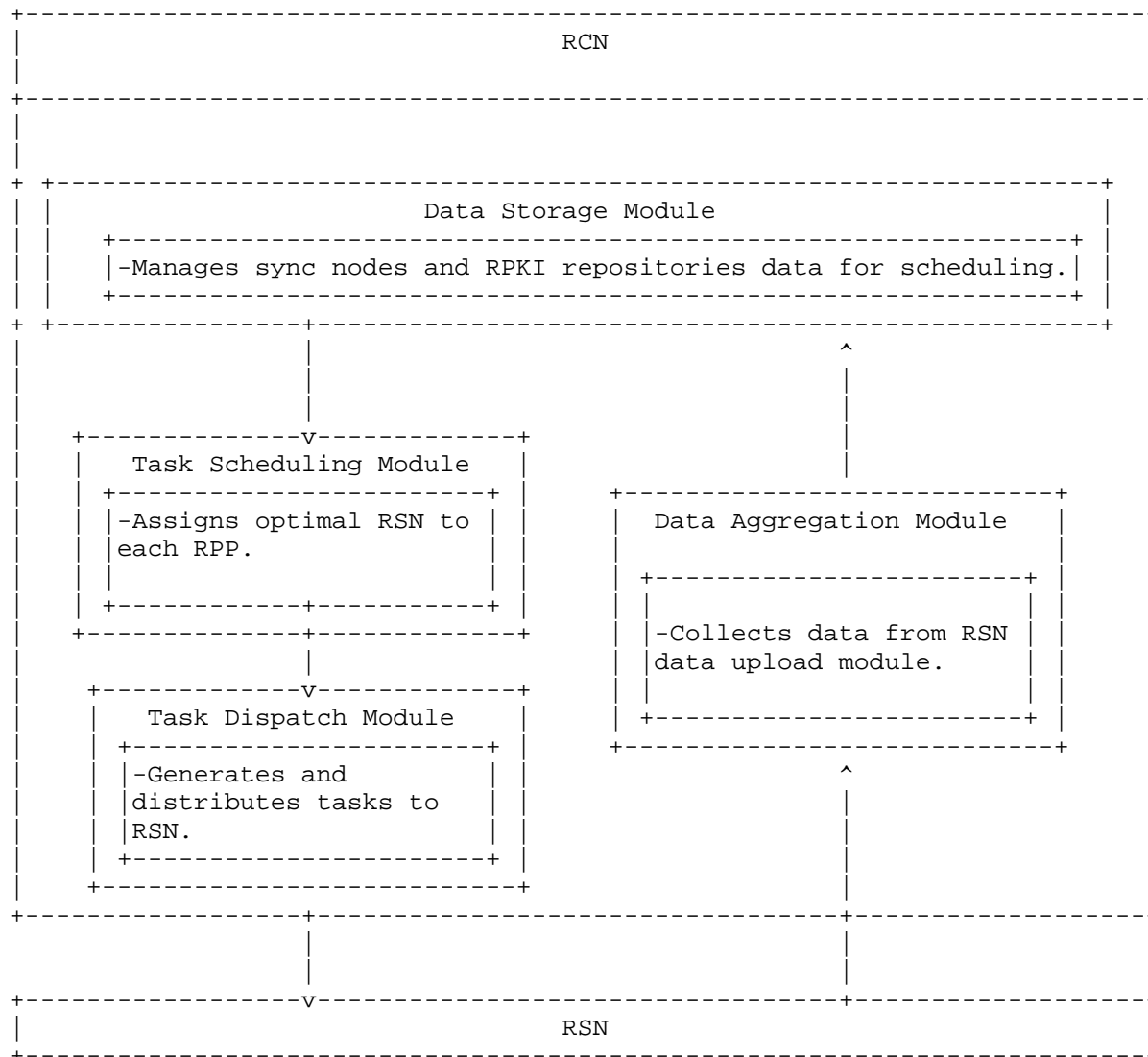


Figure 2: RCN Components

3.1.1. Data Storage Module

The main functions of the module include:

1). Management of RSN Metadata

The module maintains RSN Metadata and Load Information of this RP.

RSN Metadata including:

- URL address of an RSN;
- Geographic information of an RSN;
- RIR service region where an RSN is located.

Load Information including:

- Overall resource usage of an RSN, covering CPU, memory, and network traffic;
- Execution state of an RSN, covering active and queued tasks.

The module dynamically updates the node information to provide real-time data input for the task assignment algorithm to make the selection of an RSN performing the very task.

2). Maintenance of global RPP Information

The module maintains the RPP information pertained to task assignment:

- RPP URL;
- Version serial numbers associated with the RRDP protocol in the local cache.

The module periodically updates this information to ensure that the task assignment algorithm generates synchronized tasks based on the latest RPP status.

Through dynamic management and real-time updating, the data storage module ensures the completeness and timeliness of the information of both RSNs and RPPs, facilitating synchronization task assignment.

3.1.2. Task Scheduling Module

The task scheduling module is responsible for selecting the optimal RSN to synchronize with an given RPP that has RPKI data update. The main functions of this module include:

1). Input

Retrieve the input information from the data storage module to do scheduling:

- RSN Metadata and Load Information;

- RPP Update.

2). Calculation

Use a specific scheduling algorithm to analyze all the RSN Metadata and Load Information of this RP, combined with global RPP Update, to select optimal RSNs to synchronize with given RPPs that have RPKI data update.

3). Output

Push task assignments to Task Dispatch Module, in the form of a set with members, each indicating an specific RSN and its assigned synchronization tasks.

3.1.3. Task Dispatch Module

This module is responsible for communicating with RSNs to accomplish the assignments.

3.1.4. Data Aggregation Module

This module is responsible for collecting all the synchronization data from RSNs and make them aggregated into local cache to facilitate RPKI path validation.

Additionally, it periodically collects RSNs Load Information for scheduling cycle.

3.2. RP Synchronization Nodes (RSN)

An RSN of this RP is constructed as in Figure 3.

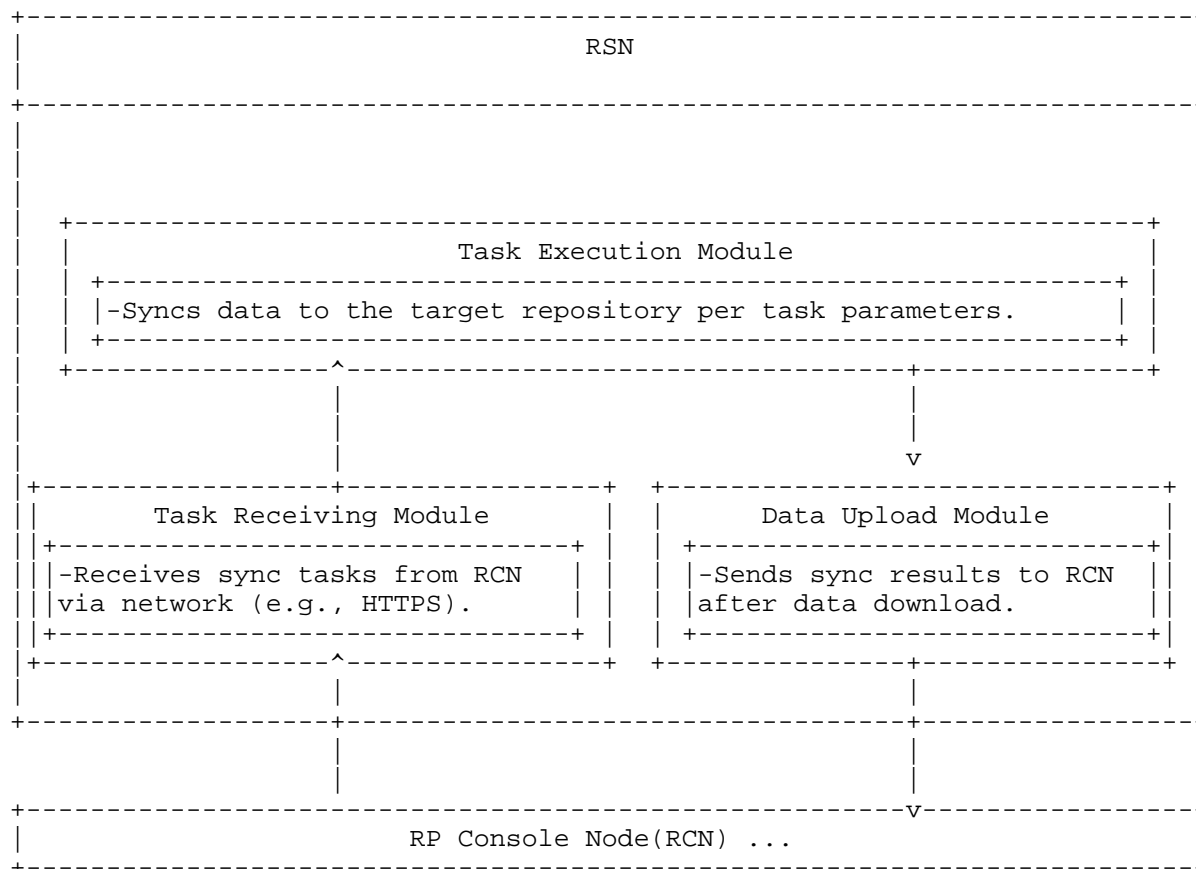


Figure 3: RSN Components

3.2.1. Task Receiving Module

This module is responsible for communicating with RCN to receive the pushed task assignments.

3.2.2. Task Execution Module

The module is responsible for performing synchronization for specific resource URL according to the task parameters.

3.2.3. Data Upload Module

This module is responsible for communicating the encapsulated synchronization data obtained from Task Execution Module to RCN. Additionally, it periodically communicates its load information to RCN.

4. Scheduling Algorithms of RPKI Data Synchronization

In practice, the characteristics of an RSN such as geographic location, network condition, bandwidth, CPU load, vary widely and are subject to change from time to time.

In this RP, scheduling algorithms are optional to the operator in his/her discretion according to local condition. There are six scheduling algorithms employed by this RP in various scenarios, to meet specific operational requirements.

4.1. Random Task Assignment

This algorithm randomly selects an RSN from the list of available RSNs to perform a task. The algorithm is straightforward and does not rely on historical selection records. And is able to achieve roughly even distribution of tasks in homogeneous environments.

4.2. Sequential Task Assignment

This algorithm traverses a sorted RSN list in a cyclical manner, ensuring fair and predictable task assignment. It maintains a selection history to uphold assignment consistency.

4.3. Network-Aware Task Assignment

This algorithm dynamically assigns tasks based on real-time network performance metrics, focusing exclusively on network speed. RSNs continuously measure download speed from designated RPP sources and log performance data for analysis. During task assignment, RCN prioritizes RSN with the highest throughput, ensuring efficient synchronization and minimizing latency.

4.4. Dynamic Task Assignment

This algorithm dynamically adjusts RSN selection weights based on real-time synchronization task load and system performance metrics, prioritizing higher-weighted RSN for pending synchronization tasks. However, its complexity is high due to sensitivity to multiple parameters, making optimization and tuning challenging.

4.5. RIR Region-Based Task Assignment

This algorithm assigns tasks based on the RIR service region where an RSN is located. For task assignment, the RSN with the closest geographic location to the RPP address is prioritized.

4.6. Designated Position Assignment

This algorithm assigns tasks based on pre-specified RSN provided as input, offering high flexibility in task assignment. However, its effectiveness depends on extensive manual tuning and repeated experimentation, which may impact deployment efficiency.

4.7. Comparison of Task Assignment Algorithms

The comparison among those scheduling algorithms are shown in Table 1.

Algorithm	Pros	Cons	Scenarios
Random Task Assignment	Straightforward and fast to deploy.	Ignores node performance and task complexity, causing imbalance.	Best for low-load, balanced-performance scenarios.
Sequential Task Assignment	Fair and efficient assignment.	Lacks dynamic adaptability.	Ideal for stable synchronization tasks with balanced load.
Network-Aware Task Assignment	Network-aware, business-aligned.	Ignores node capacity and load.	Ideal for synchronization tasks with network bottlenecks.
Dynamic Task Assignment	Strong adaptability	High complexity, instability.	Ideal for dynamic synchronization and heterogeneous nodes.
RIR Region-Based Task Assignment	Location-aware efficient routing	Unreliable due to node config or CDN virtualization.	Ideal for geo-optimized network synchronization.
Designated Position Assignment	High control and flexibility.	Manual, inflexible, unscalable.	Suited for manual tuning or strict synchronization environments.

Table 1: Comparison of RPKI Data Synchronization Task Assignment Algorithms

5. IANA Considerations
6. Security Considerations
7. References

7.1. 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/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>>.

7.2. Informative References

- [RFC8897] Ma, D. and S. Kent, "Requirements for Resource Public Key Infrastructure (RPKI) Relying Parties", RFC 8897, DOI 10.17487/RFC8897, September 2020, <<https://www.rfc-editor.org/info/rfc8897>>.

Acknowledgements

The authors would like to extend thanks to NNIX(IX.CN) for its help in designing and deploying this RPKI RP system.

Authors' Addresses

Di Ma
ZDNS
Email: madi@zdns.cn

Yanbiao Li
CNIC-CAS
Email: lybmath@cnic.cn

Yu Zhang
Peng Cheng Laboratory
Email: zhangy08@pcl.ac.cn

Shicong Zhang
China Mobile
Email: zhangshicong@cmhi.chinamobile.com