Workgroup: 6man
Internet-Draft:
draft-ietf-6man-comp-rtg-hdr-05
Published: 10 April 2024
Intended Status: Experimental
Expires: 12 October 2024
Authors: R. Bonica Y. Kamite
Juniper Networks NTT Communications Corporation
A. Alston D. Henriques L. Jalil
Liquid Telecom Liquid Telecom Verizon

The IPv6 Compact Routing Header (CRH)

Abstract

This document describes an experiment in which two new IPv6 Routing headers are implemented and deployed. Collectively, they are called the Compact Routing Headers (CRH). Individually, they are called CRH-16 and CRH-32.

One purpose of this experiment is to demonstrate that the CRH can be implemented and deployed in a production network. Another purpose is to demonstrate that the security considerations, described in this document, can be addressed with access control lists. Finally, this document encourages replication of the experiment.

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 12 October 2024.

Table of Contents

1. Introduction

IPv6 [RFC8200] source nodes use Routing headers to specify the path that a packet takes to its destination. The IETF has defined several Routing header types [IANA-RH]. This document defines two new Routing header types. Collectively, they are called the Compact Routing Headers (CRH). Individually, they are called CRH-16 and CRH-32.

The CRH allows IPv6 source nodes to specify the path that a packet takes to its destination. The CRH can be encoded in relatively few

bytes. The following are reasons for encoding the CRH in as few bytes as possible:

* Many ASIC-based forwarders copy headers from buffer memory to on-chip memory. As header sizes increase, so does the cost of this copy.

* Because Path MTU Discovery (PMTUD) [RFC8201] is not entirely reliable, many IPv6 hosts refrain from sending packets larger than the IPv6 minimum link MTU (i.e., 1280 bytes). When packets are small, the overhead imposed by large Routing Headers is excessive.

This document describes an experiment whose purposes are:

* To demonstrate that the CRH can be implemented and deployed.

* To demonstrate that the security considerations, described in this document, can be addressed with access control lists.

* To encourage replication of the experiment.

2. 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.

3. The Compressed Routing Headers (CRH)

Both CRH versions (i.e., CRH-16 and CRH-32) contain the following fields:

* Next Header - Defined in [RFC8200].

* Hdr Ext Len - Defined in [RFC8200].

* Routing Type - Defined in [RFC8200]. (CRH-16 value is 5. CRH-32 value is 6).

* Segments Left - Defined in [RFC8200].

* Type-specific Data - Described in [RFC8200].

In the CRH, the Type-specific data field contains a list of Segment Identifiers (SIDs). Each SID identifies an entry in the CRH Forwarding Information Base (CRH-FIB) (Section 4). Each CRH-FIB

entry identifies an interface on the path that the packet takes to its destination.

SIDs are listed in reverse order. So, the first SID in the list represents the final interface in the path. Because segments are listed in reverse order, the Segments Left field can be used as an index into the SID list. In this document, the "current SID" is the SID list entry referenced by the Segments Left field.

The first segment in the path can be omitted from the list. See Appendix A for an example.

In the CRH-16 (Figure 1), each SID is encoded in 16-bits. In the CRH-32 (Figure 2), each SID is encoded in 32-bits.

In all cases, the CRH MUST end on a 64-bit boundary. So, the Type-specific data field MUST be padded with zeros if the CRH would otherwise not end on a 64-bit boundary.

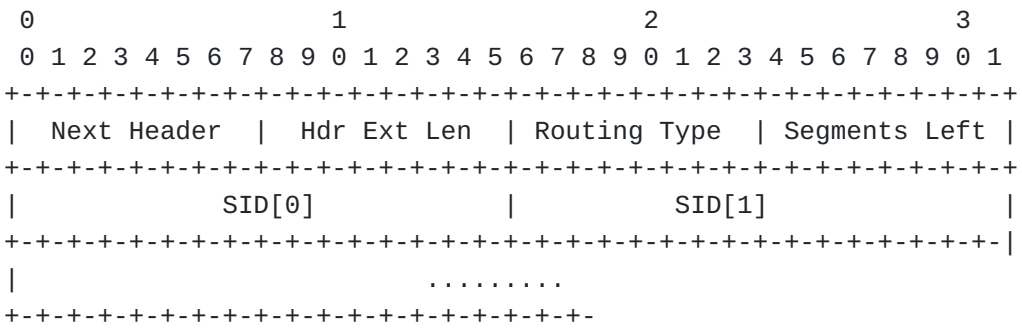

Figure 1: CRH-16

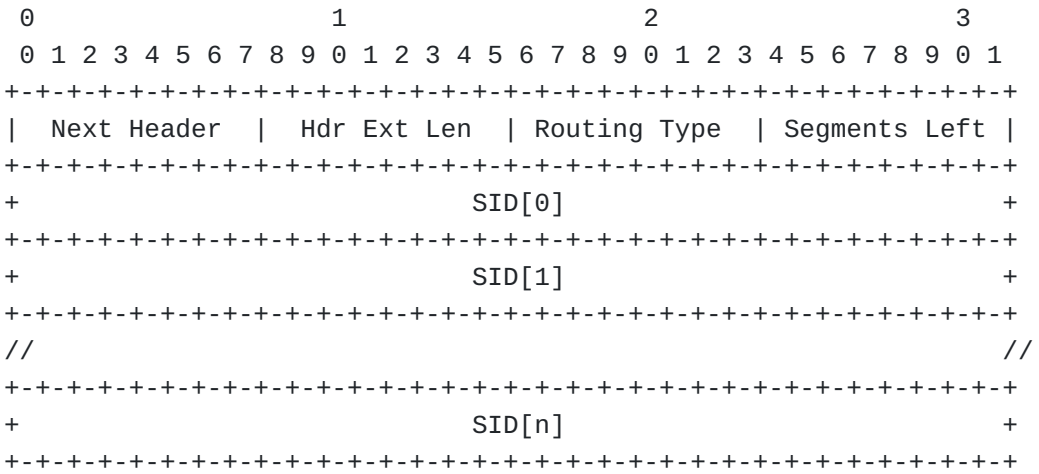

Figure 2: CRH-32

4. The CRH Forwarding Information Base (CRH-FIB)

Each SID identifies a CRH-FIB entry.

Each CRH-FIB entry contains:

- *An IPv6 address.
- *A topological function.
- *Arguments for the topological function. (Optional).

The topological function specifies how the processing node forwards the packet to the interface identified by the IPv6 address. The following are examples:

- *Forward the packet through the least-cost path to the interface identified by the IPv6 address (i.e., loose source routing).
- *Forward the packet through a specified interface to the interface identified by the IPv6 address (i.e.,strict source routing)

Some topological functions require parameters. For example, a topological function might require a parameter that identifies the interface through which the packet is forwarded.

The CRH-FIB can be populated:

- *By an operator, using a Command Line Interface (CLI).
- *By a controller, using the Path Computation Element (PCE) Communication Protocol (PCEP) [RFC5440] or the Network Configuration Protocol (NETCONF) [RFC6241].
- *By a distributed routing protocol [ISO10589-Second-Edition], [RFC5340], [RFC4271].

The above-mentoned mechanisms are not defined here and are beyond the scope of this document

5. Processing Rules

The following rules describe CRH processing:

- *If Segments Left equals 0, skip over the CRH and process the next header in the packet. The IPv6 address in the IPv6 Header's Destination Address field is that of the ultimate recipient.
- *If Hdr Ext Len indicates that the CRH is larger than the implementation can process, discard the packet and send an ICMPv6

[RFC4443] Parameter Problem, Code 0, message to the Source Address, pointing to the Hdr Ext Len field.

*Compute L, the minimum CRH length (Section 5.1).

*If L is greater than Hdr Ext Len, discard the packet and send an ICMPv6 Parameter Problem, Code 0, message to the Source Address, pointing to the Segments Left field.

*Decrement Segments Left.

*Search for the current SID in the CRH-FIB. In this document, the "current SID" is the SID list entry referenced by the Segments Left field.

*If the search does not return a CRH-FIB entry, discard the packet and send an ICMPv6 Parameter Problem, Code 0, message to the Source Address, pointing to the current SID.

*If Segments Left is greater than 0 and the CRH-FIB entry contains a multicast address, discard the packet and send an ICMPv6 Parameter Problem, Code 0, message to the Source Address, pointing to the current SID.

*Copy the IPv6 address from the CRH-FIB entry to the Destination Address field in the IPv6 header.

*Decrement the IPv6 Hop Limit.

*Submit the packet, its topological function and its parameters to the IPv6 module. See NOTE.

NOTE: By default, the IPv6 module determines the next-hop and forwards the packet. However, the topological function may elicit another behavior. For example, the IPv6 module may forward the packet through a specified interface.

NOTE: In the description above, ICMPv6 messages are subject to rate limits.

5.1. Computing Minimum CRH Length

The algorithm described in this section accepts the following CRH fields as its input parameters:

*Routing Type (i.e., CRH-16 or CRH-32).

*Segments Left.

It yields L, the minimum CRH length. The minimum CRH length is measured in 8-octet units, not including the first 8 octets.

<CODE BEGINS>

```
switch(Routing Type) {
    case CRH-16:
        if (Segments Left <= 2)
            return(0)
        sidsBeyondFirstWord = Segments Left - 2;
        sidPerWord = 4;
    case CRH-32:
        if (Segments Left <= 1)
            return(0)
        sidsBeyondFirstWord = Segments Left - 1;
        sidsPerWord = 2;
    case default:
        return(0xFF);
    }

words = sidsBeyondFirstWord div sidsPerWord;
if (sidsBeyondFirstWord mod sidsPerWord)
    words++;

return(words)
```

<CODE ENDS>

6. Mutability

In the CRH, the Segments Left field is mutable. All remaining fields are immutable.

7. Destination Address Transparency

When a packet containing the CRH header leaves its source, it does not include its final destination address. The final destination address is not added to the packet until the final SID is resolved.

While destination address transparency enhances privacy, it prevents intermediate nodes from verifying transport layer checksums.

8. Applications And SIDs

A CRH contains one or more SIDs. Each SID is processed by exactly one node.

Therefore, a SID is not required to have domain-wide significance. Applications can:

*Allocate SIDs so that they have domain-wide significance.

*Allocate SIDs so that they have node-local significance.

9. Management Considerations

PING and TRACEROUTE [RFC2151] both operate correctly in the presence of the CRH. TCPDUMP and Wireshark have been extended to support the CRH.

10. Textual Representation

A 16-bit SID can be represented by four hexadecimal digits. Leading zeros SHOULD be omitted. However, the all-zeros SID MUST be represented by a single 0. The following are examples:

*beef

*eef

*0

A 16-bit SID also can be represented in dotted-decimal notation. The following are examples:

*192.0

*192.51

A 32-bit SID can be represented by four hexadecimal digits, a colon (:), and another four hexadecimal digits. Leading zeros MUST be omitted. The following are examples:

*dead:beef

*ead:eef

*:beef

*beef:

*:

A 32-bit SID can also be represent in dotted-decimal notation. The following are examples:

*192.0.2.1

*192.0.2.2

*192.0.2.3

11. Security Considerations

In this document, one node trusts another only if both nodes are operated by the same party.

A node can encounter security vulnerabilities by indiscriminately processing packets that contain Routing Headers [RFC5095]. Therefore, nodes MUST discard packets containing the CRH when both of the following conditions are true:

*The Source Address does not identify an interface on a trusted node.

*The Destination Address identifies an interface on the local node.

The above-state rule does not protect the node from attack packets that contain a forged (i.e., spoofed) Source Address. In order to mitigate this risk, nodes MAY also discard packets containing the CRH when all of the following conditions are true:

*The Source Address identifies an interface on a trusted node.

*The Destination Address identifies an interface on the local node.

*The packet does not pass an Enhanced Feasible-Path Unicast Reverse Path Forwarding (RPF) [RFC8704],

The RPF check eliminates some, but not all packets with forged source addresses. Therefore, a network operator that deploys CRH MUST implement Access Control Lists (ACL) on each of its edge nodes. The ACL discards packets whose source address identifies an interface on a trusted node.

12. Implementation and Deployment Status

Juniper Networks has produced experimental implementations of the CRH on the MX-series (ASIC-based) router

Liquid Telecom has produced experimental implementations of the CRH on software based routers.

The CRH has carried non-production traffic in CERNET and Liquid Telecom.

Interoperability among these implementations has not yet been demonstrated.

13. Experimental Results

Parties participating in this experiment should publish experimental results within one year of the publication of this document. Experimental results should address the following:

- *Effort required to deploy
  - -Was deployment incremental or network-wide?
  - -Was there a need to synchronize configurations at each node or could nodes be configured independently
  - -Did the deployment require hardware upgrade?
  - -Did SIDs have domain-wide or node-local significance?
- *Effort required to secure
- *Performance impact
- *Effectiveness of risk mitigation with ACLs
- *Cost of risk mitigation with ACLs
- *Mechanism used to populate the FIB
- *Scale of deployment
- *Interoperability
  - -Did you deploy two inter-operable implementations?
  - -Did you experience interoperability problems?
  - -Did implementations generally implement the same topological functions with identical arguments?
  - -Were topological function semantics identical on each implementation?
- *Effectiveness and sufficiency of OAM mechanism
  - -Did PING work?
  - -Did TRACEROUTE work?
  - -Did Wireshark work?

-Did TCPDUMP work?

14. IANA Considerations

This document makes the following registrations in the "Internet Protocol Version 6 (IPv6) Parameters" "Routing Types" subregistry maintained by IANA:

Value	Description	Reference
5	CRH-16	This document
6	CRH-32	This document

15. Acknowledgements

Thanks to Dr. Vanessa Ameen, Dale Carder, Brian Carpenter, Adrian Farrel, Fernando Gont, Naveen Kottapalli, Joel Halpern, Mark Smith, Reji Thomas, Tony Li, Xing Li, Gerald Schmidt, Nancy Shaw, Ketan Talaulikar, and Chandra Venkatraman for their contributions to this document.

16. Contributors

Gang Chen

Baidu

No.10 Xibeiwang East Road Haidian District

Beijing 100193 P.R. China

Email: phdgang@gmail.com

Yifeng Zhou

ByteDance

Building 1, AVIC Plaza, 43 N 3rd Ring W Rd Haidian District

Beijing 100000 P.R. China

Email: yifeng.zhou@bytedance.com

Gyan Mishra

Verizon

Silver Spring, Maryland, USA

Email: hayabusagsm@gmail.com

17. References

17.1. 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>.

[RFC4443] Conta, A., Deering, S., and M. Gupta, Ed., "Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification", STD 89, RFC 4443, DOI 10.17487/RFC4443, March 2006, <https://www.rfc-editor.org/info/rfc4443>.

[RFC5095] Abley, J., Savola, P., and G. Neville-Neil, "Deprecation of Type 0 Routing Headers in IPv6", RFC 5095, DOI 10.17487/RFC5095, December 2007, <https://www.rfc-editor.org/info/rfc5095>.

[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>.

[RFC8200] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", STD 86, RFC 8200, DOI 10.17487/RFC8200, July 2017, <https://www.rfc-editor.org/info/rfc8200>.

[RFC8201] McCann, J., Deering, S., Mogul, J., and R. Hinden, Ed., "Path MTU Discovery for IP version 6", STD 87, RFC 8201, DOI 10.17487/RFC8201, July 2017, <https://www.rfc-editor.org/info/rfc8201>.

[RFC8704] Sriram, K., Montgomery, D., and J. Haas, "Enhanced Feasible-Path Unicast Reverse Path Forwarding", BCP 84, RFC 8704, DOI 10.17487/RFC8704, February 2020, <https://www.rfc-editor.org/info/rfc8704>.

17.2. Informative References

[IANA-RH]
IANA, "Routing Headers", <https://www.iana.org/assignments/ipv6-parameters/ipv6-parameters.xhtml#ipv6-parameters-3>.

[ISO10589-Second-Edition] International Organization for Standardization, ""Intermediate system to Intermediate system intra-domain routeing information exchange protocol for use in conjunction with the protocol for providing the connectionless-mode Network Service (ISO 8473)", ISO/IEC 10589:2002, Second Edition,", November 2001.

[RFC2151] Kessler, G. and S. Shepard, "A Primer On Internet and TCP/IP Tools and Utilities", FYI 30, RFC 2151, DOI 10.17487/RFC2151, June 1997, <https://www.rfc-editor.org/info/rfc2151>.

[RFC4271] Rekhter, Y., Ed., Li, T., Ed., and S. Hares, Ed., "A Border Gateway Protocol 4 (BGP-4)", RFC 4271, DOI 10.17487/RFC4271, January 2006, <https://www.rfc-editor.org/info/rfc4271>.

[RFC5340] Coltun, R., Ferguson, D., Moy, J., and A. Lindem, "OSPF for IPv6", RFC 5340, DOI 10.17487/RFC5340, July 2008, <https://www.rfc-editor.org/info/rfc5340>.

[RFC5440] Vasseur, JP., Ed. and JL. Le Roux, Ed., "Path Computation Element (PCE) Communication Protocol (PCEP)", RFC 5440, DOI 10.17487/RFC5440, March 2009, <https://www.rfc-editor.org/info/rfc5440>.

[RFC6241] Enns, R., Ed., Bjorklund, M., Ed., Schoenwaelder, J., Ed., and A. Bierman, Ed., "Network Configuration Protocol (NETCONF)", RFC 6241, DOI 10.17487/RFC6241, June 2011, <https://www.rfc-editor.org/info/rfc6241>.

Appendix A. CRH Processing Examples

This appendix demonstrates CRH processing in the following scenarios:

* The SID list contains one entry for each segment in the path (Appendix A.1).

* The SID list omits the first entry in the path (Appendix A.2).

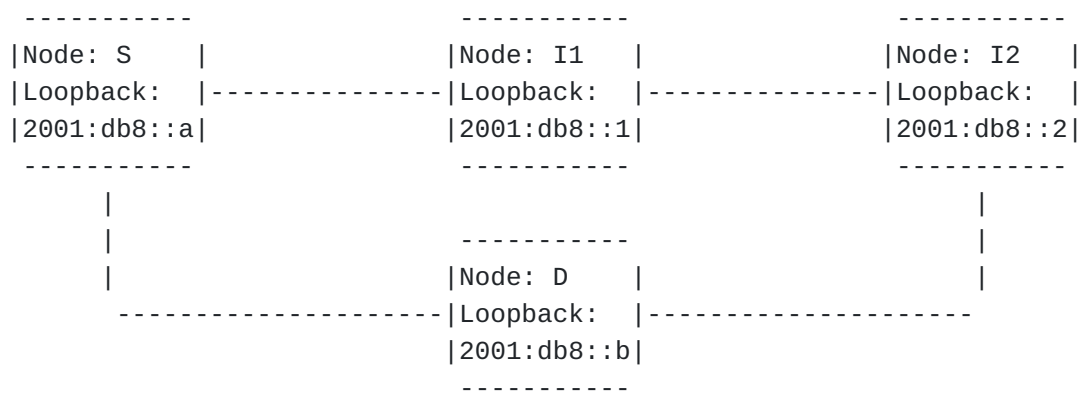

Figure 3: Reference Topology

Figure 3 provides a reference topology that is used in all examples.

SID	IPv6 Address	Forwarding Method
2	2001:db8::2	Least-cost path
11	2001:db8::b	Least-cost path

Table 1: Node SIDs

Table 1 describes two entries that appear in each node's CRH-FIB.

A.1. The SID List Contains One Entry For Each Segment In The Path

In this example, Node S sends a packet to Node D, via I2. In this example, I2 appears in the CRH segment list.

As the packet travels from S to I2:	
Source Address = 2001:db8::a	Segments Left = 1
Destination Address = 2001:db8::2	SID[0] = 11
	SID[1] = 2

Table 2

As the packet travels from I2 to D:	
Source Address = 2001:db8::a	Segments Left = 0
Destination Address = 2001:db8::b	SID[0] = 11
	SID[1] = 2

Table 3

A.2. The SID List Omits The First Entry In The Path

In this example, Node S sends a packet to Node D, via I2. In this example, I2 does not appear in the CRH segment list.

As the packet travels from S to I2:	
Source Address = 2001:db8::a	Segments Left = 1
Destination Address = 2001:db8::2	SID[0] = 11

Table 4

As the packet travels from I2 to D:	
Source Address = 2001:db8::a	Segments Left = 0
Destination Address = 2001:db8::b	SID[0] = 11

Table 5

Authors' Addresses

Ron Bonica
Juniper Networks
2251 Corporate Park Drive
Herndon, Virginia 20171
United States of America

Email: rbonica@juniper.net

Yuji Kamite
NTT Communications Corporation
3-4-1 Shibaura, Minato-ku,
108-8118
Japan

Email: y.kamite@ntt.com

Andrew Alston
Liquid Telecom
Nairobi
Kenya

Email: Andrew.Alston@liquidtelecom.com

Daniam Henriques
Liquid Telecom
Johannesburg
South Africa

Email: daniam.henriques@liquidtelecom.com

Luay Jalil
Verizon
Richardson, Texas
United States of America

Email: luay.jalil@one.verizon.com