Internet Engineering Task Force (IETF)
Request for Comments: 9631
Category: Experimental
ISSN: 2070-1721

R. Bonica
Juniper Networks
Y. Kamite
NTT Communications Corporation
A. Alston
Alston Networks
D. Henriques
Liquid Telecom
L. Jalil
Verizon

August 2024

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 Header (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 (ACLs). Finally, this document encourages replication of the experiment.

Status of This Memo

This document is not an Internet Standards Track specification; it is published for examination, experimental implementation, and evaluation.

This document defines an Experimental Protocol for the Internet community. This document is a product of the Internet Engineering Task Force (IETF). It represents the consensus of the IETF community. It has received public review and has been approved for publication by the Internet Engineering Steering Group (IESG). Not all documents approved by the IESG are candidates for any level of Internet Standard; see Section 2 of RFC 7841.

Information about the current status of this document, any errata, and how to provide feedback on it may be obtained at https://www.rfc-editor.org/info/rfc9631.

Table of Contents

1. Introduction

IPv6 [RFC8200] source nodes use Routing headers to specify the path that a packet takes to its destination(s). The IETF has defined several Routing Types; see [IANA-RT]. This document defines two new Routing Types. Collectively, they are called the Compact Routing Header (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 forwarders based on Application-Specific Integrated Circuits (ASICs) 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 with the following purposes:

* To demonstrate that the CRH can be implemented and deployed

* To demonstrate that the security considerations described in this document can be addressed with ACLs

* 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 Compact Routing Header (CRH)

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

* Next Header, as defined in [RFC8200]

* Hdr Ext Len, as defined in [RFC8200]

* Routing Type, as defined in [RFC8200] (CRH-16 value is 5, and CRH-32 value is 6.)

* Segments Left, as defined in [RFC8200]

* type-specific data, as described in [RFC8200]

In the CRH, the type-specific data field contains a list of CRH Segment Identifiers (CRH SIDs). Each CRH 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.

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

The first CRH SID in the path is omitted from the list unless there is some reason to preserve it. See Appendix A for an example.

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

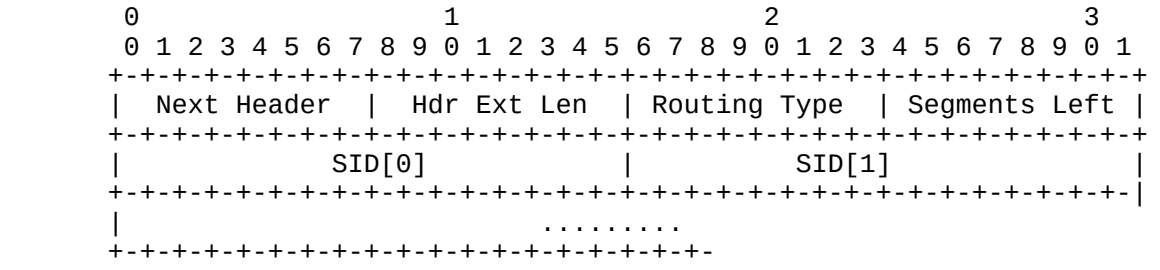

Figure 1: CRH-16

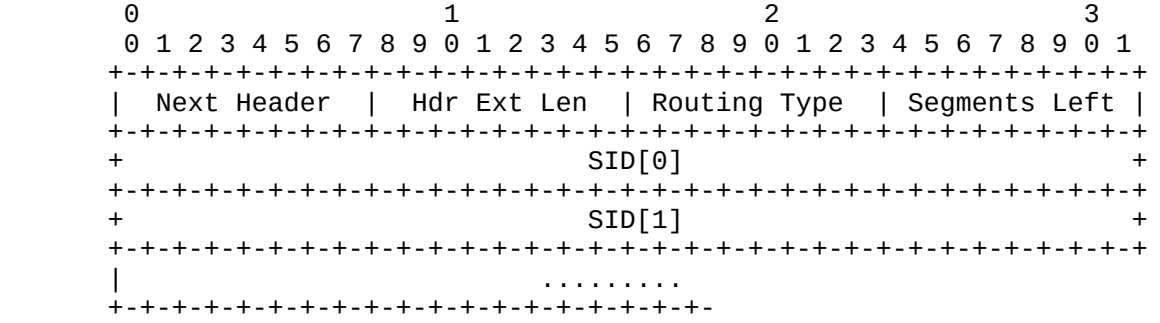

Figure 2: CRH-32

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

Each CRH SID identifies a CRH-FIB entry.

Each CRH-FIB entry contains:

* An IPv6 address

* A topological function

* Arguments for the topological function (optional)

The IPv6 address can be a Global Unicast Address (GUA), a Link-Local Unicast (LLU) address, or a Unique Local Address (ULA). When the IPv6 address is the final address in a path, it can also be a multicast address.

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)

* A controller, using the Path Computation Element Communication Protocol (PCEP) [RFC5440] or the Network Configuration Protocol (NETCONF) [RFC6241]

* A distributed routing protocol, such as those defined in [ISO10589-Second-Edition], [RFC5340], and [RFC4271]

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

5. Processing Rules

The following rules describe CRH processing:

* 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 6, message to the Source Address, pointing to the Segments Left field.

* Decrement Segments Left.

* Search for the current CRH SID in the CRH-FIB. In this document, the "current CRH SID" is the CRH 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. (This prevents packet storms.)

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

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

  | 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.

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. Applications and CRH SIDs

A CRH contains one or more CRH SIDs. Each CRH SID is processed by exactly one CRH-configured router whose one address matches the packet Destination Address.

Therefore, a CRH SID is not required to have domain-wide significance. Applications can allocate CRH SIDs so that they have either domain-wide or node-local significance.

8. Operational Considerations

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

PING and Traceroute report 16-bit CRH SIDs for CRH-16 and 32-bit CRH SIDs for CRH-32. It is recommended that the experimental versions of PING use the textual representations described in Section 9.

9. Textual Representations

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

* beef

* eef

* 0

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

* 192.0

* 192.51

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

* dead:beef

* ead:eef

* :beef

* beef:

* :

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

* 192.0.2.1

* 192.0.2.2

* 192.0.2.3

10. Security Considerations

In this document, one node trusts another only if both nodes are operated by the same party. A node determines whether it trusts another node by examining its IP address. In many networks, operators number their nodes using a small number of prefixes. This facilitates identification of trusted nodes.

A node can encounter security vulnerabilities when it processes a Routing header that originated on an untrusted node [RFC5095]. Therefore, nodes MUST deploy ACLs that 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-mentioned ACLs do 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 (EFP-uRPF) [RFC8704] check.

The EFP-uRPF check eliminates some, but not all, packets with forged Source Addresses. Therefore, a network operator that deploys CRH MUST implement ACLs on each of its edge nodes. The ACL discards packets whose Source Address identifies an interface on a trusted node.

The CRH is compatible with end-to-end IPv6 Authentication Header (AH) [RFC4302] processing. This is because the source node calculates the Integrity Check Value (ICV) over the packet as it arrives at the destination node.

11. 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 a hardware upgrade?
  - Did the CRH 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 CRH-FIB
* Scale of deployment
* Interoperability
  - Did you deploy two interoperable 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 Operations, Administration, and Maintenance (OAM) mechanisms

  - Did PING work?

  - Did Traceroute work?

  - Did Wireshark work?

  - Did TCPDUMP work?

12. IANA Considerations

IANA has registered the following in the "Routing Types" subregistry within the "Internet Protocol Version 6 (IPv6) Parameters" registry:

Value	Description	Reference
5	CRH-16	RFC 9631
6	CRH-32	RFC 9631

Table 1

13. References

13.1. 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>.

[RFC4302] Kent, S., "IP Authentication Header", RFC 4302, DOI 10.17487/RFC4302, December 2005, <https://www.rfc-editor.org/info/rfc4302>.

[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>.

13.2. Informative References

[IANA-RT] IANA, "Routing Types", <https://www.iana.org/assignments/ipv6-parameters>.

[ISO10589-Second-Edition]
ISO/IEC, "Information technology - Telecommunications and information exchange between systems - 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)", Second Edition, ISO/IEC 10589:2002, November 2002, <https://www.iso.org/standard/30932.html>.

[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>.

[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>.

Appendix A. CRH Processing Examples

This appendix demonstrates CRH processing in the following scenarios:

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

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

Figure 3 provides a reference topology that is used in all examples, and Table 2 describes two entries that appear in each node's CRH-FIB.

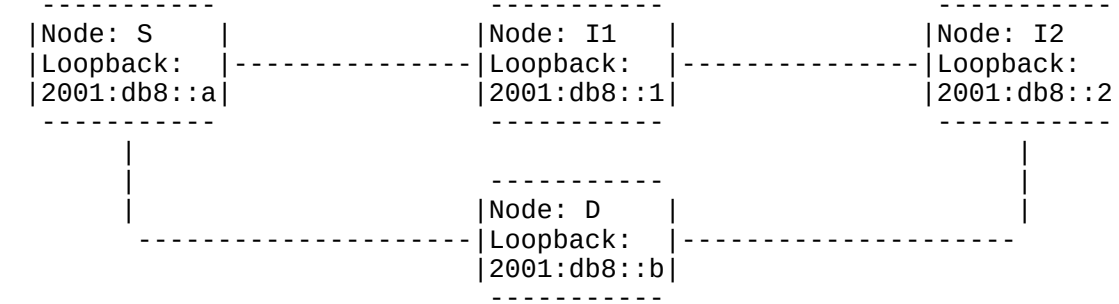

Figure 3: Reference Topology

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

Table 2: Node SIDs

A.1. The CRH SID list contains one entry for each segment in the path.

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

Source Address = 2001:db8::a	Segments Left = 1
Destination Address = 2001:db8::2	SID[0] = 11
	SID[1] = 2

Table 3: Packet Travels from S to I2

Source Address = 2001:db8::a	Segments Left = 0
Destination Address = 2001:db8::b	SID[0] = 11
	SID[1] = 2

Table 4: Packet Travels from I2 to D

A.2. The CRH SID list omits the first entry in the path.

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

Source Address = 2001:db8::a	Segments Left = 1
Destination Address = 2001:db8::2	SID[0] = 11

Table 5: Packet Travels from S to I2

Source Address = 2001:db8::a	Segments Left = 0
Destination Address = 2001:db8::b	SID[0] = 11

Table 6: Packet Travels from I2 to D

Acknowledgements

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

Contributors

Gang Chen
Baidu
No.10 Xibeiwang East Road
Haidian District
Beijing
100193
China
Email: phdgang@gmail.com

Yifeng Zhou
ByteDance
Building 1, AVIC Plaza
43 N 3rd Ring W Rd
Haidian District
Beijing
100000
China
Email: yifeng.zhou@bytedance.com

Gyan Mishra
Verizon
Silver Spring, MD
United States of America
Email: hayabusagsm@gmail.com

Authors' Addresses

Ron Bonica
Juniper Networks
2251 Corporate Park Drive
Herndon, VA 20171
United States of America
Email: rbonica@juniper.net

Yuji Kamite
NTT Communications Corporation
3-4-1 Shibaura, Minato-ku, Tokyo
108-8118
Japan
Email: y.kamite@ntt.com

Andrew Alston
Alston Networks
Nairobi
Kenya
Email: aa@alstonnetworks.net

Daniam Henriques
Liquid Telecom
Johannesburg
South Africa
Email: daniam.henriques@liquidtelecom.com

Luay Jalil
Verizon
Richardson, TX

United States of America
Email: luay.jalil@one.verizon.com