SPRING
Internet-Draft
Intended status: Standards Track
Expires: 4 January 2026

B. Varga, Ed.
F. Fejes
Ericsson

3 July 2025

Deterministic Networking specific SID
draft-varga-spring-preof-sid-02

Abstract

Replication, Elimination and Ordering functions of the DetNet Architecture require packet sequence information (i.e., sequence number) to provide service protection by the DetNet service sub-layer. This document extends SRv6 Network Programming [RFC8986] with new SR endpoint and transit behaviors to be performed on packets of DetNet flows to support the specific service protection treatment.

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 4 January 2026.

Table of Contents

1. Introduction

The DetNet Working Group has defined Packet Replication (PRF), Packet Elimination (PEF), and Packet Ordering (POF) Functions (represented as PREOF) to provide service protection by the DetNet service sub-layer [RFC8655]. This service protection method relies on copies of the same packet sent over multiple maximally disjoint paths and uses sequencing information to eliminate duplicates before delivered to its destination.

DetNet over an SRv6 data plane can provide a solution to transport sequencing information within a SID. This document describes a DetNet-specific SID (SID = Segment Identifier, [RFC8402]) and a set of related packet processing rules inside an SRv6 domain. The DetNet-specific SID provides Flow-ID and Sequence-Number information for the DetNet service sub-layer functions (i.e., PREOF).

The usage of DetNet-specific SID provides a native IPv6 data plane for DetNet networks and supports the implementation of PREOF functionalities on IPv6-only DetNet nodes. It is using native SRv6 technology and does not require additional tunneling or implementation of other protocol stack(s) (e.g., MPLS).

2. Terminology

2.1. Terms Used in This Document

This document uses the terminology established in the DetNet architecture [RFC8655] and in the SRv6 Network Programming [RFC8986]. The reader is assumed to be familiar with that document and its terminology.

2.2. Abbreviations

The following abbreviations are used in this document:

ARG	Arguments.
DetNet	Deterministic Networking.
FUNCT	Function.
LOC	Locator.
PEF	Packet Elimination Function.
POF	Packet Ordering Function.
PREOF	Packet Replication, Elimination and Ordering Functions.
PRF	Packet Replication Function.
SeqNum	Sequence Number.

2.3. 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. DetNet-specific SID

In SRv6, a SID represents a 128-bit value containing the following three parts [RFC8986]:

* Locator (LOC): first part of the SID with most significant bits and represents a specific SRv6 node.

* Function (FUNCT): the portion of the SID that is local to the owner node and designates a specific SRv6 function (network instruction) that is executed locally on a particular node (specified by Locator).

* Arguments (ARG): optional field and represents optional argument(s) to the function.

For PREOF processing, two arguments are needed:

1. Flow-ID: defines which DetNet flow the packet belongs to (what is used to determine which PREOF instance has to be used on a node). Its size is 20 bits for the DetNet MPLS data plane [RFC8986] and same size is appropriate for DetNet IP data plane as well.

2. Sequence Number: defines the sequencing information, it is created at the DetNet edge node (or by the first PRF node) and used by PEF/POF functionalities. For DetNet MPLS data plane the following sizes are defined: 0/16/28 bits [RFC8964].

The required size for these two arguments are maximum 48 bits.

The explicit format of DetNet-specific SID is network addressing design specific. PREOF specific parameters are encoded as follows:

* LOC: specifies the DetNet Relay node (same allocation rule applies as for any SRv6-enabled node).

* FUNCT: a single value represents all PREOF instances of a DetNet Relay node.

* ARG: Contains the Flow-ID and the Sequence Number parameters.

Note: if Function=PREOF, Arg=0 is also a meaningful value and does not refer to the lack of arguments.

The DetNet-specific SID must be the last segment in an SR Policy and it is associated with the PREOF functionality!

The following packet processing rules are defined as a new set of SRv6 SID behaviors regarding the DetNet-specific SID: (i) End.DPREOF, (ii) H.Encaps.PREOF, (iii) H.Encaps.PREOF.Red, (iv) H.Encaps.PREOF.L2, and (v) H.Encaps.PREOF.L2.Red.

4. SRv6 endpoint behaviors

This section describes the PREOF specific behaviors that can be associated with a SID.

End.DPREOF	Endpoint with decapsulation and PREOF Processing

Figure 1: PREOF Endpoint Behavior

4.1. End.DPREOF: Endpoint with PREOF

When a node "N" receives a packet whose IPv6 DA is "S" and "S" is a local End.DPREOF SID, "N" does the following:

```
S01. When an SRH is processed {
S02.   If (Segments Left != 0) {
S03.    Send an ICMP Parameter Problem message to the Source Address
             with Code 0 (Erroneous header field encountered),
             and Pointer set to the Segments Left field,
             interrupt packet processing and discard the packet
S04.   }
S05.   Extract the ARG part of the SID
S06.   Remove the outer IPv6 header with all its extension headers
S07.   Forward the exposed payload and the ARG part to the PREOF
       functionality
S08. }
```

5. SR Policy Headend Behaviors

This section describes a set of SRv6 Policy Headend [RFC8402] behaviors.

H.Encaps.PREOF	SR Headend with PREOF Encapsulation
H.Encaps.PREOF.Red	H.Encaps with Reduced PREOF Encapsulation
H.Encaps.PREOF.L2	H.Encaps.PREOF Applied to Received L2 Frames
H.Encaps.PREOF.L2.Red	H.Encaps.PREOF.Red Applied to Received L2 Frames

Figure 2: PREOF specific SR Policy Headend Behaviors

5.1. H.Encaps.PREOF: SR Headend with PREOF

When a node "N" receives a packet P=(A, B) identified as a DetNet Flow. B is neither a local address nor SID of "N". It executes the DetNet Flow related PREOF function(s), resulting in one or more member flow (P1=(A, B), P2=(A, B), ...) with related parameters ([Flow-ID1, SeqNum], [Flow-ID2, SeqNum], ...).

Node "N" is configured with an IPv6 address "T" (e.g., assigned to its loopback). "N" steers the egress packet P1 into an SRv6 Policy with a Source Address T and a segment list SP1=<S11, S12, S13>, where S13 is a DetNet-specific SID (LOC+FUNCT) with 0 as ARG.

The H.Encaps.PREOF encapsulation behavior is defined as follows (SA: source address, DA: destination address):

S01. Push an IPv6 header with its own SRH
     Set the ARG part of the LAST SID in the segment list
S02. Set outer IPv6 SA = T and outer IPv6 DA to the first SID
     in the segment list
S03. Set outer Payload Length, Traffic Class, Hop Limit, and
     Flow Label fields
S04. Set the outer Next Header value
S05. Decrement inner IPv6 Hop Limit or IPv4 TTL
S06. Submit the packet to the IPv6 module for transmission to S11

After the H.Encaps.PREOF behavior, P1, and P2 respectively look like:

* (T, S11) (S13, S12, S11; SL=2) (A, B), note: S13.ARG=Flow-ID1, SeqNum

* (T, S21) (S23, S22, S21; SL=2) (A, B), note: S23.ARG=Flow-ID2, SeqNum

The member flow packet is encapsulated unmodified (with the exception of the IPv4 TTL or IPv6 Hop Limit that is decremented).

The push of the SRH MAY be omitted when the SRv6 Policy only contains one segment and there is no need to use any flag, tag, or TLV. In such cases the outer destination address is the DetNet-specific SID.

5.2. H.Encaps.PREOF.Red: H.Encaps.PREOF with Reduced Encapsulation

The H.Encaps.PREOF.Red behavior is an optimization of the H.Encaps.PREOF behavior.

H.Encaps.PREOF.Red reduces the length of the SRH by excluding the first SID in the SRH of the pushed IPv6 header. The first SID is only placed in the Destination Address field of the pushed IPv6 header.

After the H.Encaps.PREOF.Red behavior, P1, and P2 respectively look like:

* (T, S11) (S13, S12; SL=2) (A, B), note: S13.ARG=Flow-ID1, SeqNum

* (T, S21) (S23, S22; SL=2) (A, B), note: S23.ARG=Flow-ID2, SeqNum

5.3. H.Encaps.PREOF.L2: H.Encaps.PREOF Applied to Received L2 Frames

The H.Encaps.PREOF.L2 behavior encapsulates a received Ethernet frame and its attached VLAN header, if present, in an IPv6 packet with an SRH. The Ethernet frame becomes the payload of the new IPv6 packet.

The H.Encaps.PREOF.L2 encapsulation behavior is similar to H.Encaps.PREOF but sets an Ethernet specific outer Next Header and lacks the TTL/Hop Limit related action. H.Encaps.PREOF.L2 is defined as follows:

S01. Push an IPv6 header with its own SRH
Set the ARG part of the LAST SID in the segment list
S02. Set outer IPv6 SA = T and outer IPv6 DA to the first SID in the segment list
S03. Set outer Payload Length, Traffic Class, Hop Limit, and Flow Label fields
S04. Set the outer Next Header value
S05. <N/A>
S06. Submit the packet to the IPv6 module for transmission to S11

The Next Header field of the SRH MUST be set to 143.

The push of the SRH MAY be omitted when the SRv6 Policy only contains one segment and there is no need to use any flag, tag, or TLV.

The encapsulating node MUST remove the preamble (if any) and frame check sequence (FCS) from the Ethernet frame upon encapsulation, and the decapsulating node MUST regenerate, as required, the preamble and FCS before forwarding the Ethernet frame.

5.4. H.Encaps.PREOF.L2.Red: H.Encaps.PREOF.L2 with Reduced Encapsulation

The H.Encaps.PREOF.L2.Red behavior is an optimization of the H.Encaps.PREOF.L2 behavior.

H.Encaps.PREOF.L2.Red reduces the length of the SRH by excluding the first SID in the SRH of the pushed IPv6 header. The first SID is only placed in the Destination Address field of the pushed IPv6 header.

The push of the SRH MAY be omitted when the SRv6 Policy only contains one segment and there is no need to use any flag, tag, or TLV.

6. DetNet-specific SID related counters

PREOF implementation may provide counters per DetNet flow. However, in order to be inline with the intention of RFC8986 (section 6. Counters), its recommendation may apply on the DetNet-specific SID and the above described set of SR Behaviors. It means, a node supporting DetNet-specific SID should implement a pair of traffic counters (one for packets and one for bytes) per local SID entry, for traffic that matched that SID and was processed successfully (i.e., packets that generate ICMP Error Messages or are dropped are not counted). The retrieval of these counters from MIB, NETCONF/YANG, or any other data structure is outside the scope of this document.

7. Security Considerations

DetNet PREOF related security considerations are described in section 3.3 of [RFC9055]. There are no additional related security considerations originating from this document.

SRv6 Network Programming related security considerations are described in section 9 of [RFC8986]. There are no additional related security considerations originating from this document.

8. IANA Considerations

This document requires registration of End.DPREOF behavior in "SRv6 Endpoint Behaviors" sub-registry of "Segment Routing Parameters" registry. IANA is requested to make one new assignments from the First Come First Served portion of the registry as follows:

Value	Hex	Endpoint Behavior	Reference	Change Controller
TBD1	xTBD1	End.DPREOF	[This.I-D]	IETF

9. Acknowledgements

Authors extend their appreciation to Janos Farkas, Istvan Moldovan and Miklos Mate for their insightful comments and productive discussion that helped to improve the document.

10. Appendix A. Illustrations

This appendix shows how the described End.DPREOF mechanisms can be used in an SRv6 network.

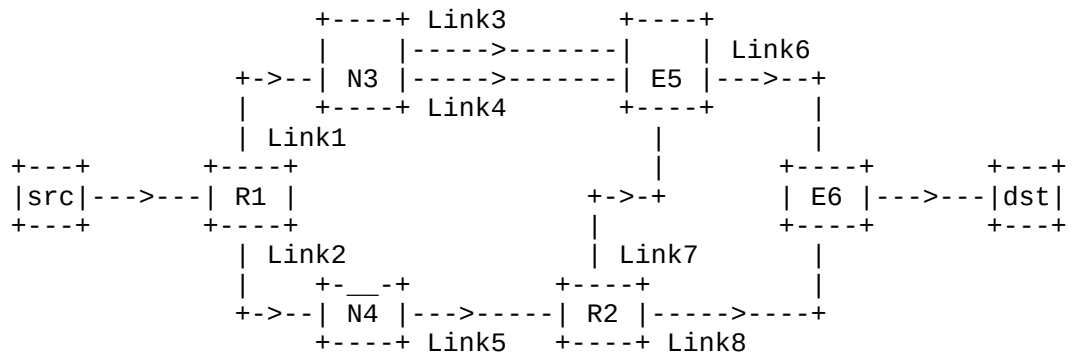

Figure 3: Example Topology

In the reference topology:

* Nodes N3, R1, R2, E5 and E6 are SRv6-capable nodes.

* Nodes R1, R2, E5 and E6 are PREOF nodes.

* Nodes N4 is an IPv6 node that is not SRv6-capable.

* Node j has an IPv6 loopback address 2001:db8:L:j::/128.

* A SID at node j with locator block 2001:db8:K::/48 and function U is represented by 2001:db8:K:j:U::.

* 2001:db8:K:j:P:: is explicitly allocated as the End.DPREOF SID at node j. For example, 2001:db8:K:2:P:: represents End.DPREOF at node R2.

* 2001:db8:K:j:Xin:: is explicitly allocated as the End.X SID at node j towards neighbor node i via the nth link between nodes i and j. For example, 2001:db8:K:3:X51:: represents End.X at node N3 towards node E5 via link3 (the first link between nodes N3 and E5). Similarly, 2001:db8:K:3:X52:: represents the End.X at node N3 towards node E5 via link4 (the second link between nodes N3 and E5).

If the src node sends a packet to the dst node for which per packet redundancy is configured, then the nodes with PREOF functions provide the required replication or elimination functions. For instance, in the example in Figure 3:

* Node src sends a UDP packet as follows: (2001:db8:src::1, 2001:db8:dst::1, NH = UDP)(UDP payload).

* Node R1, which is an SRv6-capable PREOF node, identifies the flow the packet belongs to. As replication is configured for the given flow, R1 performs the replication action and intends to send the packet to the next PREOF nodes (E5 and R2). These nodes are reachable via SRv6, so R1 performs H.Encaps.PREOF(.Red) on the replicas with a path specific SRH. The argument part of the End.DPREOF SID involves the Flow-ID and the SeqNum. Specifically, one replica is sent on link-1 towards E5 (2001:db8:L:1::, 2001:db8:K:3:X51::) (2001:db8:K:5:P:arg::, 2001:db8:K:3:X51::, SL=1, NH = IPv6) (2001:db8:src::1, 2001:db8:dst::1, NH = UDP)(UDP payload) and the other replica is sent on link-2 towards R2 (2001:db8:L:1::, 2001:db8:K:2:P:arg::, NH = IPv6) (2001:db8:src::1, 2001:db8:dst::1, NH = UDP)(UDP payload).

* Node N3, which is an SRv6-capable node, performs the standard SRH processing. Specifically, it executes the End.X behavior indicated by the 2001:db8:K:3:X51:: SID and forwards the packet on link3 to node E5.

* Node N4, which is a non-SRv6-capable node, performs the standard IPv6 processing. Specifically, it forwards the UDP packet based on DA 2001:db8:K:2:P:arg:: in the IPv6 header towards node R2.

* Node R2, which is an SRv6-capable PREOF node, identifies the packet as targeted to the local PREOF function. R2 performs the decapsulation and forwards the exposed payload and the ARG part to the PREOF functionality. The PREOF function identifies the flow the packet belongs to. As replication is configured for the given flow, R2 performs the replication action and intends to send the packet to the next PREOF nodes (E5 and E6). These nodes are reachable via SRv6, so R2 performs H.Encaps.PREOF(.Red) on the replicas with a path specific SRH. The argument part of the End.DPREOF SID involves the Flow-ID and the SeqNum. Specifically, one replica is sent on link-7 towards E5 (2001:db8:L:2::, 2001:db8:K:5:P:arg::, NH = IPv6) (2001:db8:src::1, 2001:db8:dst::1, NH = UDP)(UDP payload) and the other replica is sent on link-8 towards E6 (2001:db8:L:2::, 2001:db8:K:6:P:arg::, NH = IPv6) (2001:db8:src::1, 2001:db8:dst::1, NH = UDP)(UDP payload).

* Node E5, which is an SRv6-capable PREOF node, identifies the packets as targeted to the local PREOF function. E5 performs the decapsulation and forwards the payload and the ARG part to the PREOF functionality. The PREOF function identifies the flow the packet belongs to. As elimination is configured for the given flow, the elimination action is performed on the packets received over Link3 and Link7. E5 intends to send the packet to the next PREOF node (E6), which is reachable via SRv6, so E6 performs H.Encaps.PREOF(.Red) with a path specific SRH. The argument part of the End.DPREOF SID involves the Flow-ID and the SeqNum. Specifically, the replica received first is sent on link-6 towards E6 (2001:db8:L:5::, 2001:db8:K:6:P:arg::, NH = IPv6) (2001:db8:src::1, 2001:db8:dst::1, NH = UDP)(UDP payload).

* Node E6, which is an SRv6-capable PREOF node, identifies the packets as targeted to the local PREOF function. It performs the decapsulation and forwards the payload and the ARG part to the PREOF functionality. The PREOF function identifies the flow the packet belongs to. As elimination is configured for the given flow, the elimination action is performed on the packets received over Link6 and Link8. E6 is the last PREOF node, so after the PREOF function it send the UDP packet towrds the destination. Specifically, the replica received first is sent towards the destination (2001:db8:src::1, 2001:db8:dst::1, NH = UDP)(UDP payload).

The example topology shown in Figure 3 is constructed to show the usage of PREOF-SID. Note that any of the links can be replaced with an SRv6 network segment. The above described principles are applicable to such more complex network topologies as well.

11. 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>.

[RFC8402] Filsfils, C., Ed., Previdi, S., Ed., Ginsberg, L., Decraene, B., Litkowski, S., and R. Shakir, "Segment Routing Architecture", RFC 8402, DOI 10.17487/RFC8402, July 2018, <https://www.rfc-editor.org/info/rfc8402>.

[RFC8655] Finn, N., Thubert, P., Varga, B., and J. Farkas, "Deterministic Networking Architecture", RFC 8655, DOI 10.17487/RFC8655, October 2019, <https://www.rfc-editor.org/info/rfc8655>.

[RFC8964] Varga, B., Ed., Farkas, J., Berger, L., Malis, A., Bryant, S., and J. Korhonen, "Deterministic Networking (DetNet) Data Plane: MPLS", RFC 8964, DOI 10.17487/RFC8964, January 2021, <https://www.rfc-editor.org/info/rfc8964>.

[RFC8986] Filsfils, C., Ed., Camarillo, P., Ed., Leddy, J., Voyer, D., Matsushima, S., and Z. Li, "Segment Routing over IPv6 (SRv6) Network Programming", RFC 8986, DOI 10.17487/RFC8986, February 2021, <https://www.rfc-editor.org/info/rfc8986>.

[RFC9055] Grossman, E., Ed., Mizrahi, T., and A. Hacker, "Deterministic Networking (DetNet) Security Considerations", RFC 9055, DOI 10.17487/RFC9055, June 2021, <https://www.rfc-editor.org/info/rfc9055>.

Authors' Addresses

Balazs Varga (editor)
Ericsson
Budapest
Magyar Tudosok krt. 11.
1117
Hungary
Email: balazs.a.varga@ericsson.com

Ferenc Fejes
Ericsson
Budapest
Magyar Tudosok krt. 11.
1117
Hungary
Email: ferenc.fejes@ericsson.com