

Workgroup: Network Working Group  
Internet-Draft:  
draft-lhan-satellite-instructive-routing-03  
Published: 1 September 2023  
Intended Status: Informational  
Expires: 4 March 2024  
Authors: L. Han, Ed.

Futurewei Technologies, Inc.  
A. Retana  
Futurewei Technologies, Inc.  
R. Li  
Futurewei Technologies, Inc.

## **Semantic Address Based Instructive Routing for Satellite Network**

### **Abstract**

This document presents a method to do IP routing over satellite network that consists of LEO (Low Earth Orbit) satellites and ground-stations. The method uses the source routing mechanism. The whole routing info is obtained by path calculation. The routing path information is converted to be a list of instructions and embedded into user packet's IPv6 extension header. At each hop or each satellite, the routing process engine will forward the packet based on the specified instruction for the satellite. Until the packet reaches the edge of satellite network, or the last satellite, the packet will be sent to a ground station.

### **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 March 2024.

### **Copyright Notice**

Copyright (c) 2023 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. Terminology](#)
- [3. Review of LEO satellite constellation for future Internet](#)
- [4. Basics of Instructive Routing](#)
  - [4.1. Forwarding Directions](#)
  - [4.2. Forwarding Segments](#)
  - [4.3. Forwarding Instructions](#)
  - [4.4. Example](#)
- [5. IPv6 Routing Header for Instructive Routing](#)
- [6. Instruction List for Instructive Routing](#)
- [7. Instructive Routing Behaviors](#)
  - [7.1. Fwd.Inc.Sat\\_ID](#)
  - [7.2. Fwd.Dec.Sat\\_ID](#)
  - [7.3. Fwd.Inc.Opb\\_ID](#)
  - [7.4. Fwd.Dec.Opb\\_ID](#)
  - [7.5. Fwd.Inc.Shl\\_ID](#)
  - [7.6. Fwd.Dec.Shl\\_ID](#)
  - [7.7. End.Intf\\_ID](#)
  - [7.8. End.Punt](#)
  - [7.9. End.Lookup](#)
  - [7.10. End.Lookup.IPv4](#)
  - [7.11. End.Lookup.IPv6](#)
  - [7.12. Fwd.Sat\\_Addr](#)
  - [7.13. Fwd.Sat\\_MacAddr](#)
  - [7.14. Forwarding\\_API\(Packet,Input\\_Satellite,Input\\_Direction\)](#)
  - [7.15. Forwarding\\_API\\_SAT\(Packet,Input\\_Satellite,Sat\\_Addr\)](#)
  - [7.16. Forwarding\\_API\\_MAC\(Packet,Input\\_Satellite,Sat\\_MacAddr\)](#)
  - [7.17. Forwarding\\_GS\\_API\(Packet,Input\\_Interface\)](#)
- [8. Other notes](#)
- [9. IANA Considerations](#)
- [10. Security Considerations](#)
- [11. Contributors](#)
- [12. Acknowledgements](#)
- [13. References](#)
  - [13.1. Normative References](#)
  - [13.2. Informative References](#)
- [Appendix A. Change Log](#)

## 1. Introduction

Massive LEO constellation is expected to be used for future Internet. It has raised challenges to the current IP networking technologies to support such super-fast-moving network. [\[I-D.lhan-problems-requirements-satellite-net\]](#) has analyzed the problems when using the regular routing protocols in such network.

Since all satellites in a LEO constellation are well organized and form a kind of multi-layered grid network, each satellite's relative position in the satellite network will be steady during its life time. [\[I-D.lhan-satellite-semantic-addressing\]](#) has proposed to use couple of indexes to identify each satellite in the network. The combination of the indexes is called the satellite semantic address. The semantic address can be embedded into the field of the interface identifier (i.e., the rightmost 64 bits) of the IPv6 address, if IPv6 is used in the satellite network.

This memo proposes a method for routing for LEO satellite network, it is based on the satellite semantic address. It is a source routing mechanism and conceptually similar to SRv6 (IPv6 Segment Routing) [\[RFC8754\]](#) with loose-hop, but with many differences in the architecture and details. The routing information is embedded into the IPv6 packet as routing extension header defined in [\[RFC8200\]](#). Unlike the SRv6 [\[RFC8754\]](#) and programming [\[RFC8986\]](#), The new method will not use IPv6 SID (Segment Identifier) to represent the segments on the routing path. Instead, it will convert the segments on the path to be a list of instructions since each satellite could be represented by the semantic address. Each instruction can tell each satellite how to forward the packet to an adjacent satellite and when to stop, either on the same orbit, or on the adjacent orbit.

Compared with the traditional IP forwarding, the new method will not use TCAM (Ternary Content-addressable Memory) lookup for IP prefix. Each satellite only needs to store a simple adjacency table. Therefore, the new method can save significant TCAM and the processing time for routing/forwarding tables.

It must be noted this memo just describes one aspect of the whole solution for satellite constellation used for Internet access and NTN (Non-Terrestrial Network) integration with 5G, following areas are not covered in this memo and will be addressed in other documents separately:

1. IP forwarding path determination for a LEO constellation. There are different strategies and algorithms to determine the IP path. One example using modified OSPF and Dijkstra algorithm

[[I-D.retana-lsr-ospf-monitor-node](#)] to get the shortest geographic path can be found in [[Large-Scale-LEO-Network-Routing](#)].

2. Data planes for different scenarios, such as Internet access and NTN integration.
3. Other protocols for control plane.

## 2. Terminology

**LEO** Low Earth Orbit with the altitude from 180 km to 2000 km.

**LEO constellation** LEO constellation consists of certain number of LEOs. Each LEO has pre-assigned orbit element.

**ISL** Inter Satellite Link

**GS** Ground Station, a device on ground connecting satellite. In the document, GS will hypothetically provide L2 and/or L3 functionality in addition to process/transmit/receive radio wave. It might be different as the reality that the device to process/transmit/receive radio wave and the device to provide L2 and/or L3 functionality could be separated.

**L2** Layer 2, or Data Link Layer in OSI model [[OSI-Model](#)]

**L3** Layer 3, or Network Layer in OSI model [[OSI-Model](#)], it is also called IP layer in TCP/IP model

**OS** Operating System

**NTN** Non-Terrestrial Network

**SID** Segment Identifier

**Sat-GS Links** Wireless links between satellites and ground-stations, it consists of uplink (from ground to satellite) and downlink (from satellite to ground).

**Link Metrics** The cost of the outgoing interface for routing, typically, it may indicate the bandwidth, delay or other costs for the interface.

**Sat\_ID** Satellite Index, the Index for the satellite in a orbit plane, see [[I-D.lhan-satellite-semantic-addressing](#)]

**Obp\_ID** Orbit Plane Index, the Index for the orbit plane in a shell group of satellite, see [[I-D.lhan-satellite-semantic-addressing](#)]

**Shl\_ID**

Shell Index, the Index for the shell group of satellite in a satellite constellation, see [\[I-D.lhan-satellite-semantic-addressing\]](#)

**Intf\_ID** Interface Index

**Sat\_Addr** Satellite Semantic Address, it consists of indexes Shl\_ID, Obp\_ID and Sat\_ID. It is 32-bit long and is defined in Section 5.4 in [\[I-D.lhan-satellite-semantic-addressing\]](#)

**Sat\_MacAddr** The MAC (Media Access Control) Address for a satellite

### 3. Review of LEO satellite constellation for future Internet

LEO satellite constellation is expected to be integrated with terrestrial network in future Internet. StarLink project [\[StarLink\]](#) has launched its satellites and provided the beta service in some areas. 3GPP [\[ThreeGPP\]](#) has studied the issues when NTN is integrated with Internet and 5G. 3GPP [\[TR38-821\]](#) has also proposed the Satellite-based NG-RAN architectures for NTN integration. In the 3GPP new Release 18 (in-progress), there is a working item "Study on 5G System with Satellite Backhaul" [\[TR23-700\]](#). In which, LEO satellite network will provide the transport functionality for 5G RAN access network. As a summary, the targets of LEO constellation for future Internet and NTN integration are as follows:

1. Global coverage: The Satellite network should cover all places on earth and any flying objects as long as the place or objects are below LEO attitude and within the coverage footprint of satellite constellation, the satellite network should be the complementary to terrestrial network.
2. Internet access: The Satellite network can provide the Internet access service for covered areas.
3. NTN integration: The Satellite network is fully integrated with Internet including Wireless such as 4G or 5G.
4. Competitive service: The Satellite network can provide the services that are competitive to terrestrial network in terms of service stability, Quality of Service, especially the latency for Satellite network is shorter.

As a new form of network, LEO constellation has lots of difference with the steady terrestrial network especially in the mobility. [\[I-D.lhan-problems-requirements-satellite-net\]](#) has analyzed the movement and coverage of satellite. For a massive LEO constellation, all satellites are moving on the allocated orbits, and form one or

multiple layers of network. Finally, the massive LEO constellation will have the following unprecedented mobility:

1. Each LEO moves at the speed of 7.x km/s.
2. Ground Stations move at the speed of 463 m/s due to earth rotation.
3. Half of LEOs move on the direction that is different with another half of LEOs.
4. Huge number of links between satellites and ground-stations, and all of them are constantly flipping within short period of time. All Link Metrics of Sat-GS Links are also constantly changing.
5. All Link Metrics of ISL on the Longitude direction are constantly changing.
6. All Links of ISL on the Longitude direction may be interrupted at two polar areas.
7. All Link Metrics of ISL on the radius direction (for satellites with different altitude) are constantly changing.
8. All Links of ISL on the radius direction can only last for a limited time.

#### **4. Basics of Instructive Routing**

In IP routing or forwarding, the IP path consists of a list of IP nodes (hops). In LEO satellite network, the IP forwarding path is a list of satellites. Instructive routing essentially is a mechanism that converts satellites on the path to a list of segment and then to a list of instructions. It will utilize the special characters of LEO satellite network to achieve the minimized packet overhead while the forwarding functions can be executed quickly.

A typical LEO satellite network is an interleaved and meshed network moving constantly. Each satellite only has limited adjacent satellites, thus the limited packet forwarding directions (see [Section 4.1](#)).

The satellites on a forwarding path can be converted to a list of segments. The number of segments is normally much smaller than the number of satellites on the path.

The number of segment type will determine the number of instruction type. Since the segment type is also limited (see [Section 4.2](#)), so the instruction type is limited.

Finally, combining the above characters and with the use of semantic address, the Instructive Routing will only introduce limited overhead that is much smaller than SRv6 and SRv6 with compressed SID.

#### 4.1. Forwarding Directions

When using ISL for satellites in a LEO constellation, each layer of network will have satellite nodes connected by limited ISLs. A typical satellite will have about six ISL to connected to its adjacent satellites in 3D space. Additionally, there might have very few numbers of ISL working as un-steady link to connect to other satellites. Un-steady links are those between satellites moving to different directions, see [\[I-D.lhan-problems-requirements-satellite-net\]](#) for the detailed explanation. After using the semantic address for each satellite, the satellite relationship will be static. [Figure 1](#) illustrates one satellite and its six direct connected adjacent satellites, it is easy to determine some indexes of its adjacent satellites:

1. S0, S1 and S2 have the same Shl\_ID, the difference of Obp\_ID between S0 and S1, S0 and S2 are both equal to one.
2. S0, S3 and S4 have the same Shl\_ID and Obp\_ID, the difference of Sat\_ID between S0 and S3, S0 and S4 are both equal to one.
3. S0, S5 and S6 have different Shl\_ID, and the difference of Shl\_ID between S0 and S5, S0 and S6 are both equal to one.

Another benefit to use the semantic address is that the packet forwarding for routing and switching will be simplified significantly. There will be only six major forwarding directions to the directly connected adjacent satellites described above, plus one or few specified directions probably. The specified direction is to forward packet to a specified adjacent satellite through an un-steady link. The un-steady link can connect to any satellite but only last for a short time. The usages of un-steady links are expected to be limited and are not major scenarios in a LEO constellation. Following are all directions for forwarding:

1. Forward to the Sat\_ID Incremental or Decremental directions.
2. Forward to the Obp\_ID Incremental or Decremental directions.
3. Forward to the Shl\_ID Incremental or Decremental directions.
4. Forward to a specified satellite through an un-steady link.

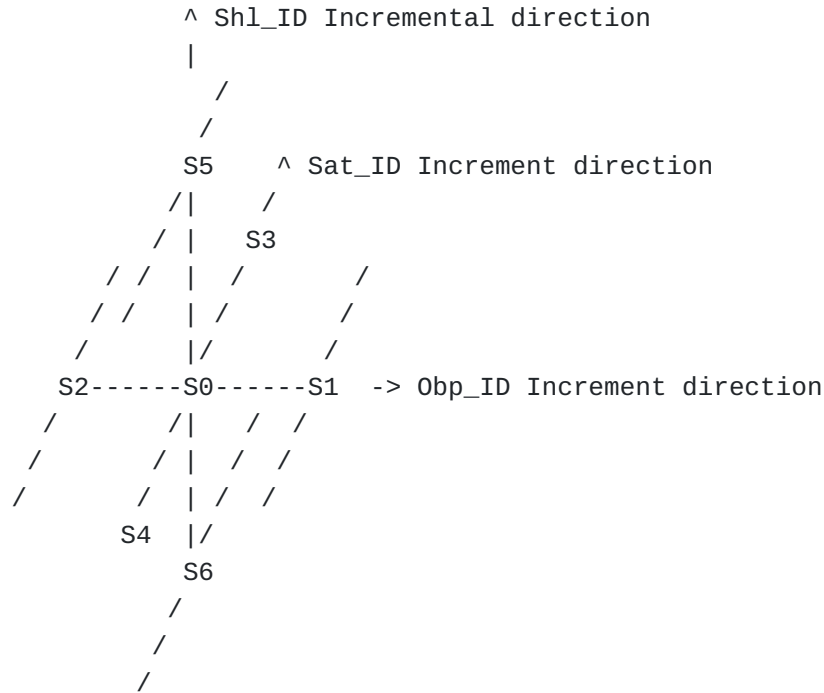


Figure 1: The LEO Satellite Relationship in 3D Space

#### 4.2. Forwarding Segments

A forwarding segment is defined as a list of satellites, and four type segments are defined for LEO satellite network where semantic address is used:

1. Segment with adjacent Shl\_ID: For any direct adjacent satellites on the segment, their Shl\_ID are also adjacent (differ by one).
2. Segment with adjacent Obp\_ID: For any direct adjacent satellites on the segment, their Obp\_ID are also adjacent (differ by one), the Shl\_ID are the same.
3. Segment with adjacent Sat\_ID: For any direct adjacent satellites on the segment, their Sat\_ID are also adjacent (differ by one), the Obp\_ID and Shl\_ID are identical.
4. Segment with non-adjacent index: this segment only has two satellites and two satellites do not belong to the above three categories.

#### 4.3. Forwarding Instructions

Each forwarding instruction consists of Functional Code and Argument (see [Section 6](#)). For the most often used instructions, the Argument



represents one specified index (Sat\_ID or Obp\_ID or Shl\_ID) of a satellite semantic address and only has the size of one octet.

Each segment maps to a forwarding instruction that can guide the packet forwarded at each satellite from the start to the end of the segment. For the segment types (1) to (3) described in [Section 4.2](#), there are two directions to forward packet, each direction can be defined as either an increment or a decrement of a specified index. For type (4), there is one direction to forward packet. In total we have seven directions to forward packets among all satellites: to the satellite ahead or behind; to either sides; above or below; or to another non-adjacent satellite.

When an IP packet is forwarded on a segment by an instruction, at each satellite, the forwarding logic needs to check if the packet reaches the end of the segment. In the regular segment routing, the long size of SID is used to do such indication. But for satellite network, since 32-bit satellite's semantic address is embedded into the IPv6 address, it is not needed to include the long SID into the packet header. Instead, we only need to compare one octet index of the current satellite's semantic address, instead of whole IPv6 address, with the Argument in the instruction.

#### 4.4. Example

[Figure 2](#) illustrates a 2D example. It shows how a packet is forwarded in a grid satellite network. Intuitively, we can obtain the list of instructions to guide the packet and get the forwarding behaviors at different satellites. Following is an example:

1. At S1 to S2, forward packet to the Sat\_ID Incremental direction, until the packet reaches S2
2. At S2 to S3, forward packet to the Obp\_ID Incremental direction, until the packet reaches the orbit plane of S3
3. At S3 to S4, forward packet to the Sat\_ID Incremental direction, until the packet reaches S4
4. At S4 to S5, forward packet to the Obp\_ID Decremental direction, until the packet reaches the orbit plane of S5
5. At S5 to S6, forward packet to the Sat\_ID Decremental direction, until the packet reaches S6

By using a specified index of semantic address as the argument as described in [Section 4.3](#), we can further simplify the above instructions as:

1. At S1 to S2, forward packet to the Sat\_ID Incremental direction, until the packet reaches a satellite and the satellite's Sad\_ID is equal to the given instruction argument (S2's Satellite Index)
2. At S2 to S3, forward packet to the Obp\_ID Incremental direction, until the packet reaches a satellite and the satellite's Obp\_ID is equal to the given instruction argument (S3's Orbit Plane Index)
3. At S3 to S4, forward packet to the Sat\_ID Incremental direction, until the packet reaches a satellite and the satellite's Sat\_ID is equal to the given instruction argument (S4's Satellite Index)
4. At S4 to S5, forward packet to the Obp\_ID Decremental direction, until the packet reaches a satellite and the satellite's Obp\_ID is equal to the given instruction argument (S5's Orbit Plane Index)
5. At S5 to S6, forward packet to the Sat\_ID Decremental direction, until the packet reaches a satellite and the satellite's Sat\_ID is equal to the given instruction argument (S6's Satellite Index)

```

^ Sat_ID Incremental Direction
|
|
+-----> Obp_ID Incremental Direction

```

x: The ISL is down

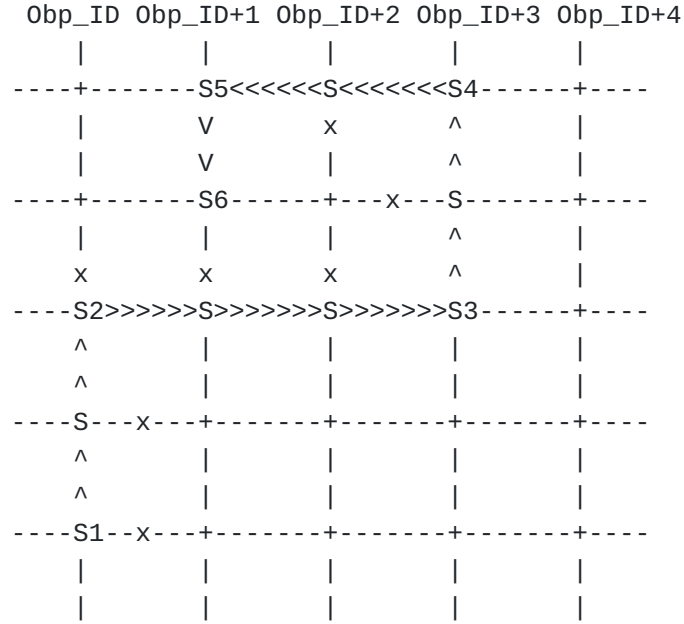


Figure 2: Packet Forwarding in 2D LEO satellite constellation network

## 5. IPv6 Routing Header for Instructive Routing

For instructive routing, IPv6 routing header is used with a new routing type "Instructive Routing Type". The format of the new routing header is illustrated in [Figure 3](#).

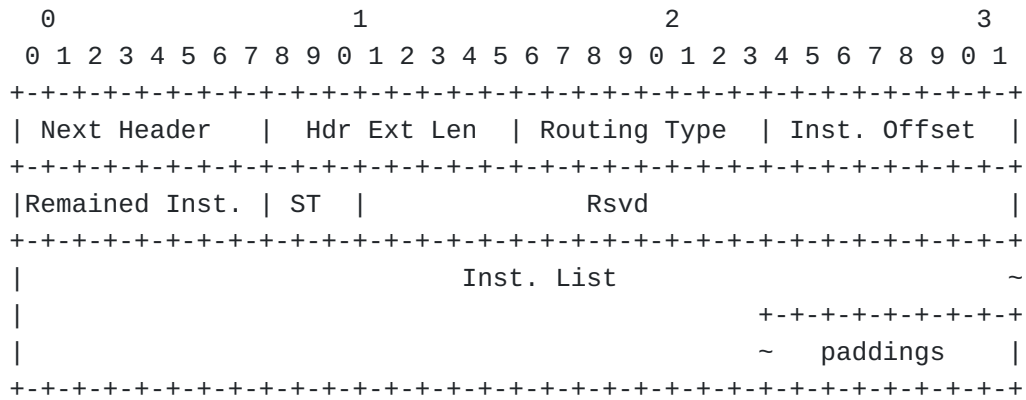


Figure 3: The IPv6 Routing Hdr for Instructive Routing

Routing Type	Instructive	Routing Type
--------------	-------------	--------------

**Inst. Offset** The offset in the number of octets from the start of Instruction List. The initial value is set to 0 and it points to the 1st instruction to be executed. The value is incremented by the number of octets of the total size of an instruction after the instruction is executed.

**Remained Inst.** Remained Number of Instructions. The initial value is set to the total number of instructions. The value will be decremented by one after one instruction is executed. The minimum number is one, and it indicates that the end of instruction stack is reached.

**ST** The satellite address type, default is 0.

**Inst. List** A list of instructions, the size is variable.

**Paddings** Pad1 or PadN options to make the packet extension header alignment, see [\[RFC8200\]](#)

## 6. Instruction List for Instructive Routing

For instructive routing, the instruction list is used to instruct each satellite how to do routing job. The format of the instruction list is illustrated in [Figure 4](#). Each instruction consists of Function Code and Arguments.

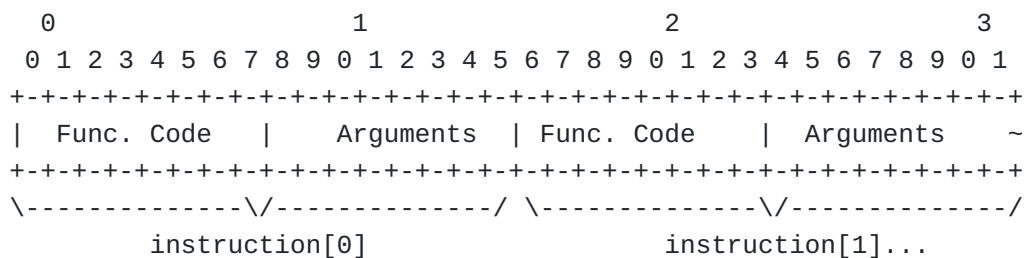


Figure 4: The Instruction List for Instructive Routing

<b>Func. Code</b>	Function Code, size is 1 octet
-------------------	--------------------------------

**Arguments** Arguments for the function, Variable length

## 7. Instructive Routing Behaviors

The behavior for each satellite for instructive routing is described here. Table 1 is the summary of the name, Hex values of all

functions, arguments and size. New functions can be defined if needed.

The subsections below are the detailed explanation for each function.

Func Name/Hex Value	Arguments/Size(Octet)	Reference
Fwd.Inc.Sat_ID/0X01	Sat_ID/1	<a href="#">Section 7.1</a>
Fwd.Dec.Sat_ID/0X02	Sat_ID/1	<a href="#">Section 7.2</a>
Fwd.Inc.Obp_ID/0X03	Obp_ID/1	<a href="#">Section 7.3</a>
Fwd.Dec.Obp_ID/0X04	Obp_ID/1	<a href="#">Section 7.4</a>
Fwd.Inc.Shl_ID/0X05	Shl_ID/1	<a href="#">Section 7.5</a>
Fwd.Dec.Shl_ID/0X06	Shl_ID/1	<a href="#">Section 7.6</a>
End.Intf_ID/0X07	Intf_ID/1	<a href="#">Section 7.7</a>
End.Punt/0X08	0X0/1	<a href="#">Section 7.8</a>
End.Lookup/0X09	0X0/1	<a href="#">Section 7.9</a>
End.Lookup.IPv4/0X0A	IPv4_Addr/4	<a href="#">Section 7.10</a>
End.Lookup.IPv6/0X0B	IPv6_Addr/16	<a href="#">Section 7.11</a>
Fwd.Sat_Addr/0X0C	Sat_Addr/4	<a href="#">Section 7.12</a>
Fwd.Sat_MacAddr/0X0D	Sat_MacAddr/6	<a href="#">Section 7.13</a>

Table 1: Functions, Arguments and Reference

The functions in [Section 7.1](#) to [Section 7.6](#) are used for the instructions to forward packet to one of the six major directions discussed in [Section 4](#). They will call API in [Section 7.14](#) to forward the packet to the specified direction.

The functions in [Section 7.12](#) and [Section 7.13](#) are used for the instructions to forward packet to a specified adjacent satellite discussed in [Section 4](#). They will call APIs in [Section 7.15](#) and [Section 7.16](#) respectively to forward the packet to the specified adjacent satellite.

In order to forward packet, each satellite should have an adjacency table stored locally; the table should contain the information about all adjacent satellites, it should at least store:

1. Each adjacent satellite's semantic address.
2. The ID of local interface connecting to each adjacent satellite.
3. The MAC address for the remote interface of each adjacent satellite.

### 7.1. Fwd.Inc.Sat\_ID

The definition of this function is "Forward the packet on the Satellite Index Incremental Direction until the packet reaches a Satellite whose Satellite Index is equal to the value specified in the argument"

This function is used for the instruction to forward packet to one of the six major directions discussed in [Section 4](#).

When a satellite receives a packet with new routing header, assume the satellite indexes in the address are Shl\_index, Obp\_index, Sat\_index respectively, the satellite does the following. During the forwarding, the Forwarding\_API in [Section 7.14](#) is called to forward the packet to the specified direction.

```
S01. When an IRH is processed {
S02.   If ((RI > 1) and (Argument != Sat_index)) {
S03.     Input_Satellite = Current Satellite;
S04.     Input_Direction = Satellite Index Incremental direction;
S05.     Forwarding_API(Packet,Input_Satellite,Input_Direction);
S06.   } else {
S07.     IOF += 2;
S08.     RI --;
S09.     if (RI <= 0)
        Send an ICMP Parameter Problem to the Source Address
        with Code 0 (Erroneous header field encountered)
        and Pointer set to the RI field,
        interrupt packet processing, and discard the packet;
S10.   Proceed to execute the next Instruction;
S11. }
S12.}
```

### 7.2. Fwd.Dec.Sat\_ID

The definition of this function is "Forward the packet on the Satellite Index Decremental Direction until the packet reaches a Satellite whose Satellite Index is equal to the value specified in the argument"

This function is used for the instruction to forward packet to one of the six major directions discussed in [Section 4](#).

When a satellite receives a packet with new routing header, assume the satellite indexes in the address are Shl\_index, Obp\_index, Sat\_index respectively, the satellite does the following. During the forwarding, the Forwarding\_API in [Section 7.14](#) is called to forward the packet to the specified direction.

```

S01. When an IRH is processed {
S02.   If ((RI > 1) and (Argument != Sat_index)) {
S03.     Input_Satellite = Current Satellite;
S04.     Input_Direction = Satellite Index Decremental direction;
S05.     Forwarding_API(Packet,Input_Satellite,Input_Direction);
S06.   } else {
S07.     IOF += 2;
S08.     RI --;
S09.     if (RI <= 0)
        Send an ICMP Parameter Problem to the Source Address
        with Code 0 (Erroneous header field encountered)
        and Pointer set to the RI field,
        interrupt packet processing, and discard the packet;
S10.   Proceed to execute the next Instruction;
S11.   }
S12.}

```

### 7.3. Fwd.Inc.Opb\_ID

The definition of this function is "Forward the packet on the Orbit Plane Index Incremental Direction until the packet reaches a Satellite whose Orbit Plane Index is equal to the value specified in the argument"

This function is used for the instruction to forward packet to one of the six major directions discussed in [Section 4](#).

When a satellite receives a packet with new routing header, assume the satellite indexes in the address are Shl\_index, Obp\_index, Sat\_index respectively, the satellite does the following. During the forwarding, the Forwarding\_API in [Section 7.14](#) is called to forward the packet to the specified direction.

```

S01. When an IRH is processed {
S02.   If ((RI > 1) and (Argument != Obp_index)) {
S03.     Input_Satellite = Current Satellite;
S04.     Input_Direction = Orbit Plane Index Incremental direction;
S05.     Forwarding_API(Packet,Input_Satellite,Input_Direction);
S06.   } else {
S07.     IOF += 2;
S08.     RI --;
S09.     if (RI <= 0)
        Send an ICMP Parameter Problem to the Source Address
        with Code 0 (Erroneous header field encountered)
        and Pointer set to the RI field,
        interrupt packet processing, and discard the packet;
S10.   Proceed to execute the next Instruction;
S11.   }
S12.}

```

#### 7.4. Fwd.Dec.Opb\_ID

The definition of this function is "Forward the packet on the Orbit Plane Index Decremental Direction until the packet reaches a Satellite whose Orbit Plane Index is equal to the value specified in the argument"

This function is used for the instruction to forward packet to one of the six major directions discussed in [Section 4](#).

When a satellite receives a packet with new routing header, assume the satellite indexes in the address are Shl\_index, Obp\_index, Sat\_index respectively, the satellite does the following. During the forwarding, the Forwarding\_API in [Section 7.14](#) is called to forward the packet to the specified direction.

```
S01. When an IRH is processed {
S02.   If ((RI > 1) and (Argument != Obp_index)) {
S03.     Input_Satellite = Current Satellite;
S04.     Input_Direction = Orbit Plane Index Decremental direction;
S05.     Forwarding_API(Packet,Input_Satellite,Input_Direction);
S06.   } else {
S07.     IOF += 2;
S08.     RI --;
S09.     if (RI <= 0)
        Send an ICMP Parameter Problem to the Source Address
        with Code 0 (Erroneous header field encountered)
        and Pointer set to the RI field,
        interrupt packet processing, and discard the packet;
S10.   Proceed to execute the next Instruction;
S11. }
S12.}
```

#### 7.5. Fwd.Inc.Shl\_ID

The definition of this function is "Forward the packet on the Orbit Shell Index Incremental Direction until the packet reaches a Satellite whose Orbit Shell Index is equal to the value specified in the argument"

This function is used for the instruction to forward packet to one of the six major directions discussed in [Section 4](#).

When a satellite receives a packet with new routing header, assume the satellite indexes in the address are Shl\_index, Obp\_index, Sat\_index respectively, the satellite does the following. During the forwarding, the Forwarding\_API in [Section 7.14](#) is called to forward the packet to the specified direction.



```

S01. When an IRH is processed {
S02.   If ((RI > 1) and (Argument != Shl_index)) {
S03.     Input_Satellite = Current Satellite;
S04.     Input_Direction = Orbit Shell Index Incremental direction;
S05.     Forwarding_API(Packet,Input_Satellite,Input_Direction);
S06.   } else {
S07.     IOF += 2;
S08.     RI --;
S09.     if (RI <= 0)
        Send an ICMP Parameter Problem to the Source Address
        with Code 0 (Erroneous header field encountered)
        and Pointer set to the RI field,
        interrupt packet processing, and discard the packet;
S10.   Proceed to execute the next Instruction;
S11.   }
S12.}

```

#### 7.6. Fwd.Dec.Shl\_ID

The definition of this function is "Forward the packet on the Orbit Shell Index Decremental Direction until the packet reaches a Satellite whose Orbit Shell Index is equal to the value specified in the argument"

This function is used for the instruction to forward packet to one of the six major directions discussed in [Section 4](#).

When a satellite receives a packet with new routing header, assume the satellite indexes in the address are Shl\_index, Obp\_index, Sat\_index respectively, the satellite does the following. During the forwarding, the Forwarding\_API in [Section 7.14](#) is called to forward the packet to the specified direction.

```

S01. When an IRH is processed {
S02.   If ((RI > 1) and (Argument != Shl_index)) {
S03.     Input_Satellite = Current Satellite;
S04.     Input_Direction = Orbit Shell Index Decremental direction;
S05.     Forwarding_API(Packet,Input_Satellite,Input_Direction);
S06.   } else {
S07.     IOF += 2;
S08.     RI --;
S09.     if (RI <= 0)
        Send an ICMP Parameter Problem to the Source Address
        with Code 0 (Erroneous header field encountered)
        and Pointer set to the RI field,
        interrupt packet processing, and discard the packet;
S10.   Proceed to execute the next Instruction;
S11.   }
S12.}

```

### 7.7. End.Intf\_ID

The definition of this function is "End of processing for the Instructive routing, remove the Instructive Routing Header, Forward the packet to the interface specified in the argument"

This function is normally used on the Dst\_Sat to forward packet to Dst\_GS.

When a satelllite receives a packet with new routing header, the satellite does the following, Forwarding\_GS\_API in [Section 7.17](#) is called to forward the packet to the specified interface.

```
S01. When an IRH is processed {  
S02.   Change the Next header in the packet header to be  
       the Next Header field in the Instructive Routing header;  
S03.   Remove the Instructive Routing Header;  
S04.   Forwarding_GS_API(Packet, Argument);  
S05.}
```

### 7.8. End.Punt

The definition of this function is "End of processing for the Instructive routing, remove the Instructive Routing Header, Punt the packet to the OS for process"

This function is normally used send packet to a satelllite. At the destination satelllite, the packet is punted to the OS to be processed further.

When a satelllite receives a packet with new routing header, the satellite does the following:

```
S01. When an IRH is processed {  
S02.   Change the Next header in the packet header to be  
       the Next Header field in the Instructive Routing header;  
S03.   Remove the Instructive Routing Header;  
S04.   Punt packet to the local CPU for process;  
S05.}
```

### 7.9. End.Lookup

The definition of this function is "End of processing for the Instructive routing, remove the Instructive Routing Header, Lookup the destination address in packet header and forward the packet accordingly"

This function is normally used to send packet to Dst\_GS. After the packet reaches the Dst\_Sat, the packet is forwarded to Dst\_GS by looking up the destination address in the IPv6 packet header.

When a satellite receives a packet with new routing header, the satellite does the following:

```
S01. When an IRH is processed {  
S02.   Change the Next header in the packet header to be  
       the Next Header field in the Instructive Routing header;  
S03.   Remove the Instructive Routing Header;  
S04.   Lookup the destination address in packet hdr and forward  
       the packet;  
S05.}
```

#### **7.10. End.Lookup.IPv4**

The definition of this function is "End of processing for the Instructive routing, remove the Instructive Routing Header, Lookup the IPv4 address specified in the argument and forward the packet accordingly"

This function is normally used to send packet to Dst\_GS. After the packet reaches the Dst\_Sat, the packet is forwarded to Dst\_GS by looking up the IPv4 destination address specified in the Function Argument.

When a satellite receives a packet with new routing header, the satellite does the following:

```
S01. When an IRH is processed {  
S02.   Fetch the IPv4 addr in the argument;  
S03.   Change the Next header in the packet header to be  
       the Next Header field in the Instructive Routing header;  
S04.   Remove the Instructive Routing Header;  
S05.   Lookup the fetched IPv4 address and forward the packet;  
S06.}
```

#### **7.11. End.Lookup.IPv6**

The definition of this function is "End of processing for the Instructive routing, remove the Instructive Routing Header, Lookup the IPv6 address specified in the argument and forward the packet accordingly"

This function is normally used to send packet to Dst\_GS. After the packet reaches the Dst\_Sat, the packet is forwarded to Dst\_GS by looking up the IPv6 destination address specified in the Function Argument.

When a satellite receives a packet with new routing header, the satellite does the following:

```

S01. When an IRH is processed {
S02.   Fetch the IPv6 addr in the argument;
S03.   Change the Next header in the packet header to be
        the Next Header field in the Instructive Routing header;
S04.   Remove the Instructive Routing Header;
S05.   Lookup the fetched IPv6 address and forward the packet;
S06.}

```

#### 7.12. Fwd.Sat\_Addr

The definition of this function is "Forward the packet to the adjacent satellite with the address specified in the argument"

This function is normally used for the instruction to forward packet to an adjacent satellite specified by its Satellite Semantic Address. The Satellite Semantic Address is 32-bit long and is defined in Section 5.4 in [[I-D.lhan-satellite-semantic-addressing](#)]

When a satellite receives a packet with new routing header, assume the satellite semantic address is Sat\_Addr, the satellite does the following:

```

S01. When an IRH is processed {
S02.   If ((RI > 1) and (Argument != Sat_Addr)) {
S03.     Input_Satellite = Current Satellite;
S04.     SatAddr = Argument;
S05.     Forwarding_API_SAT(Packet, Input_Satellite, SatAddr);
S06.   } else {
S07.     IOF += 4;
S08.     RI --;
S09.     if (RI <= 0)
        Send an ICMP Parameter Problem to the Source Address
        with Code 0 (Erroneous header field encountered)
        and Pointer set to the RI field,
        interrupt packet processing, and discard the packet.
S10.     Proceed to execute the next Instruction;
S11.   }
S12.}

```

#### 7.13. Fwd.Sat\_MacAddr

The definition of this function is "Forward the packet to the adjacent satellite with the MAC address specified as the argument"

This function is normally used for the instruction to forward packet to an adjacent satellite specified by its MAC address.

When a satellite receives a packet with new routing header, assume the satellite Mac address is Sat\_MacAddr, the satellite does the following:

```

S01. When an IRH is processed {
S02.   If ((RI > 1) and (Argument != Sat_MacAddr)) {
S03.     Input_Satellite = Current Satellite;
S04.     SatMacAddr = Argument;
S05.     Forwarding_API_Mac(Packet, Input_Satellite, SatMacAddr);
S06.   } else {
S07.     IOF += 6;
S08.     RI --;
S09.     if (RI <= 0)
        Send an ICMP Parameter Problem to the Source Address
        with Code 0 (Erroneous header field encountered)
        and Pointer set to the RI field,
        interrupt packet processing, and discard the packet.
S10.   Proceed to execute the next Instruction;
S11.   }
S12.}

```

#### **7.14. Forwarding\_API(Packet, Input\_Satellite, Input\_Direction)**

This API will forward a packet to the specified direction. When a satellite executes the API, it will do following:

```

S01. Forwarding_API(Packet, Input_Satellite, Input_Direction) {
S02.   Lookup the local adjacency table to find out
      1) The adjacent satellite of "Input_Satellite" on the
         direction equal to "Input_Direction" (The adjacent
         satellite's semantic address can be inferred by
         the "Input_Satellite" and "Input_Direction").
      2) The L2 address for the adjacent satellite;
      3) The local interface connecting to the adjacent
         satellite;
S03.   Rewrite the L2 header of the Packet by the L2 address;
S04.   Send the Packet to the local interface;
S05.}

```

#### **7.15. Forwarding\_API\_SAT(Packet, Input\_Satellite, Sat\_Addr)**

This API will forward a packet to the specified adjacent satellite with the semantic address as the argument. When a satellite executes the API, it will do following:

```

S01. Forwarding_API_SAT(Packet,Input_Satellite,SatAddr) {
S02.     Lookup the local adjacency table to find out
        1) The adjacent satellite of "Input_Satellite"
           (The adjacent satellite address is SatAddr);
        2) The L2 address for the adjacent satellite;
        3) The local interface connecting to the adjacent
           satellite;
S03.     Rewrite the L2 header of the Packet by the L2 address;
S04.     Send the Packet to the local interface;
S05.}

```

#### **7.16. Forwarding\_API\_MAC(Packet,Input\_Satellite,Sat\_MacAddr)**

This API will forward a packet to the specified adjacent satellite with the MAC address as the argument. When a satellite executes the API, it will do following:

```

S01. Forwarding_API_MAC(Packet,Input_Satellite,SatMacAddr) {
S02.     Lookup the local adjacency table to find out
        1) The adjacent satellite of "Input_Satellite"
           (The adjacent satellite MAC address is SatMacAddr);
        2) The L2 address for the adjacent satellite;
        3) The local interface connecting to the adjacent
           satellite;
S03.     Rewrite the L2 header of the Packet by the L2 address;
S04.     Send the Packet to the local interface;
S05.}

```

#### **7.17. Forwarding\_GS\_API(Packet,Input\_Interface)**

This API will forward a packet to ground station the connected to the specified interface. When a satellite executes the API, it will do following:

```

S01. Forwarding_API(Packet,Input_Interface) {
S02.     Lookup the local adjacency table to find out
        1) The connected GS to the interface
           equal to "Input_Interface";
        2) The L2 address for the GS;
S03.     Rewrite the L2 header of the Packet by the L2 address;
S04.     Send the Packet to the "Input_Interface";
S05.}

```

### **8. Other notes**

Due to the limit of the picture drawing for IETF draft, the pictures in the memo may not be easy to understand. For easier understanding of the method, please refer to the [\[Large-Scale-LEO-Network-Routing\]](#), it provided more vivid pictures

obtained by simulation software Savi [[Savi](#)], and also provided the simulation results.

## 9. IANA Considerations

This document defines a new IPv6 Routing Type: the "Instructive Routing Header". It needs to be assigned a number by IANA.

This document also defines an 8-bit Function Name, for which IANA will create and will maintain a new sub-registry entitled "Instructive Routing Function Name" under the "Internet Protocol Version 6 (IPv6) Parameters" [[IPv6 Parameters](#)] registry. Initial values for the subtype registries are given in [Table 1](#).

## 10. Security Considerations

The instructive routing is only applicable to a satellite network that is using the satellite semantic address. It will add instructive routing header at a GS and the header will be removed before reaching another GS. Normally, a satellite network including all GS is trusted domain. Traffic will be filtered at the domain boundaries. Non-authorized users cannot access the satellite network.

## 11. Contributors

## 12. Acknowledgements

## 13. References

### 13.1. Normative References

- [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>>.
- [RFC8754] Filsfils, C., Ed., Dukes, D., Ed., Previdi, S., Leddy, J., Matsushima, S., and D. Voyer, "IPv6 Segment Routing Header (SRH)", RFC 8754, DOI 10.17487/RFC8754, March 2020, <<https://www.rfc-editor.org/info/rfc8754>>.
- [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>>.

### 13.2. Informative References

**[I-D.lhan-problems-requirements-satellite-net]**

Han, L., Li, R., Retana, A., Chen, M., Su, L., Jiang, T., and N. Wang, "Problems and Requirements of Satellite Constellation for Internet", Work in Progress, Internet-Draft, draft-lhan-problems-requirements-satellite-net-05, 5 July 2023, <<https://datatracker.ietf.org/doc/html/draft-lhan-problems-requirements-satellite-net-05>>.

**[I-D.lhan-satellite-semantic-addressing]** Han, L., Li, R., Retana, A., Chen, M., and N. Wang, "Satellite Semantic Addressing for Satellite Constellation", Work in Progress, Internet-Draft, draft-lhan-satellite-semantic-addressing-04, 1 September 2023, <<https://datatracker.ietf.org/doc/html/draft-lhan-satellite-semantic-addressing-04>>.

**[I-D.retana-lsr-ospf-monitor-node]** Retana, A. and L. Han, "OSPF Monitor Node", Work in Progress, Internet-Draft, draft-retana-lsr-ospf-monitor-node-00, 7 March 2022, <<https://datatracker.ietf.org/doc/html/draft-retana-lsr-ospf-monitor-node-00>>.

**[StarLink]** "Star Link", <<https://en.wikipedia.org/wiki/Starlink>>.

**[ThreeGPP]** "3GPP", <<https://www.3gpp.org/>>.

**[OSI-Model]** "OSI Model", <[https://en.wikipedia.org/wiki/OSI\\_model](https://en.wikipedia.org/wiki/OSI_model)>.

**[TR38-821]** "Solutions for NR to support Non-Terrestrial Networks (NTN)", <<https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3525>>.

**[TR23-700]** "Study on support of satellite backhauling in 5GS", <<https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=4000>>.

**[IPv6\_Parameters]** IANA, "Internet Protocol Version 6 (IPv6) Parameters", <<https://www.iana.org/assignments/ipv6-parameters/ipv6-parameters.xhtml#ipv6-parameters-3>>.

**[Large-Scale-LEO-Network-Routing]** Han, L., Retana, A., Westphal, C. and R. Li, "Large Scale LEO Satellite Networks for the Future Internet: Challenges and Solutions to Addressing and Routing," Computer Networks and Communications, 1(1), 31-58", 2023, <<https://ojs.wiserpub.com/index.php/CNC/article/view/2105>>.

**[Savi]** "Satellite constellation visualization", <<https://savi.sourceforge.io/>>.



## **Appendix A. Change Log**

\*Initial version, 02/28/2022

\*Revision 1, 09/02/2022

\*Revision 2, 03/03/2023

\*Revision 3, 09/01/2023

## **Authors' Addresses**

Lin Han (editor)  
Futurewei Technologies, Inc.  
2330 Central Express Way  
Santa Clara, CA 95050,  
United States of America

Email: [lh@futurewei.com](mailto:lh@futurewei.com)

Alvaro Retana  
Futurewei Technologies, Inc.  
2330 Central Express Way  
Santa Clara, CA 95050,  
United States of America

Email: [alvaro.retana@futurewei.com](mailto:alvaro.retana@futurewei.com)

Richard Li  
Futurewei Technologies, Inc.  
2330 Central Express Way  
Santa Clara, CA 95050,  
United States of America

Email: [rli@futurewei.com](mailto:rli@futurewei.com)