

INTERNET-DRAFT
Intended Status: Experimental
Expires: August 2018

T. Herbert
Quantonium

February 3, 2018

Identifier groups
draft-herbert-idgroups-00

Abstract

This draft describes a means to create logical identifier groups to manage identifiers in a mapping system for identifier-locator protocols. An identifier group consists of identifiers that have similar properties in the context of the mapping system. Identifier groups facilitate bulk operations on the mapping system that would affect multiple identifiers. A primary use case for this is to facilitate mobility of devices that are associated with possibly thousands or even millions of identifiers.

Status of this Memo

This Internet-Draft is submitted to IETF in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

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."

The list of current Internet-Drafts can be accessed at
<http://www.ietf.org/lid-abstracts.html>

The list of Internet-Draft Shadow Directories can be accessed at
<http://www.ietf.org/shadow.html>

Copyright and License Notice

Copyright (c) 2018 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 (<http://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 Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1	Introduction	3
2	Characteristics of identifiers	3
2.1	Identifier addresses	3
2.2	Desired properties	4
2.2	Policy mechanisms for identifiers	4
3	Structure of identifier groups	5
4	Interfaces	7
4.1	Management interface	7
4.2	Query interface	7
5	Security Considerations	8
6	IANA Considerations	10
7	References	10
7.1	Normative References	10
7.2	Informative References	10
	Author's Address	10

1 Introduction

This document describes identifier groups for identifier-locator mapping systems.

Identifier-locator protocols include the concept of identifiers as a type of node addressing. Identifiers are logical endpoints of communications and only differ from canonical addresses in that they are not topological. A node may be assigned multiple ephemeral identifiers so that they be can used to create different source addresses for different communications to benefit privacy and anonymity. It is expected that individual end devices may have thousands of active ephemeral identifiers; a device that connects backend subnets could have millions of associated identifiers.

An identifier-group is an group of identifiers within a mapping system that share some common properties. A grouping is arbitrary, the given application or mapping system may create identifier groups as needed. An identifier may belong to multiple groups, however when an operation is performed it must be clear as to which group applicable properties are be derived from. Groups may also be hierarchical such that groups may be members of other groups and thus inherit properties from their parent groups.

A primary application of identifier groups is mobility where a device has a number of identifiers associated with it. When such a device moves in the network and is assigned a new locator, all of the identifiers associated with the device assume the new locator also. Identifier groups provide a level of indirection so that the locator can be set for all of the associated identifiers for the device in a single operation on the mapping system.

2 Characteristics of identifiers

This section list some salient properties of identifiers that are relevant to a mapping system and privacy.

2.1 Identifier addresses

Identifier addresses are full IP addresses that are either an identifier or contain an identifier as part of the address. Identifier addresses are used by endpoints to achieve communications. In order to reach the end host where the node indicated by an identifier resides, somewhere in the path an identifier-locator operation is performed and the packet is typically modified (either by encapsulation or address translation) to reach the correct node. At the destination node, a reverse operation is done to restore the originally sent packet before presenting the packet to the end node

or application.

Identifier addresses should have the following properties:

2.2 Desired properties

- o They are composed of a global routing prefix and a suffix that is internal to an organization. This is the same property for IP addresses [RFC3513].
- o The registry and organization of an address can be determined by the network prefix. This is true for any global address.
- o The organizational bits in the address should have minimal hierarchy to prevent inferences. It might be reasonable to have an internal prefix that divides identifiers based on broad geographic regions, but detailed information such as location, department in an enterprise, or device type should not be encoded in a globally visible address.
- o Given two identifier addresses and no other information, the desired properties of correlating them are:
 - o It can be inferred if they belong the same organization and registry. This is true for any two global IP addresses.
 - o It may be inferred that they belong to the same broad grouping, such as a geographic region, if the information is encoded in the organizational bits of the address.
 - o No other correlation can be established. For example, it cannot be inferred that the IP addresses address the same device, the IP addresses reside in the same subnet or department, or that the nodes for the two addresses have any geographic proximity to one another.

2.2 Policy mechanisms for identifiers

Other than a globally routable network prefix, identifier addresses require no hierarchy since they are not topological. Therefore all or most of the organizational bits in a publicly visible address form a flat, non-hierarchical space. To create identifier addresses with the properties listed above, the bits in this space are pseudo-randomly assigned to form addresses.

While the routing requirements are satisfied by the identifier-locator protocols and mapping system, the lack of internal hierarchy in addresses is a potential disruption for network deployments that

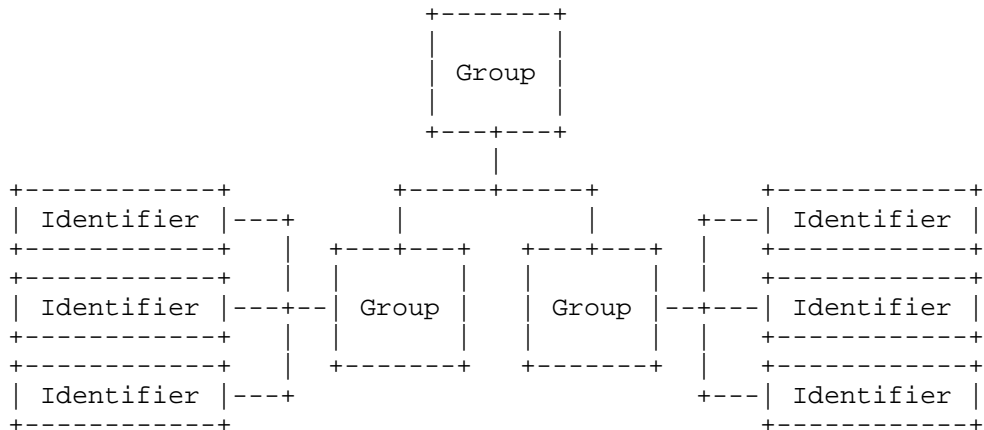
rely on address hierarchy to implement policy. For instance, an enterprise might implement a firewall rule base on destination network prefix that prevents the engineering department from talking to human resources.

In order to apply such policies and still maintain the properties to prevent inference, a firewall could create rules based on identifier groups. So when a packet arrives at the firewall, the mapping system may be consulted and information for a group is returned. A policy decision, i.e. forward or drop, may be made per this information.

In the example above, identifier groups might be created for engineering and human resources. The policy is expressed that members of the engineering group are not allowed to send to members human resources group. Since the groups are not encoded in the addresses there is no means for an external party to infer which packets belong to engineering and which belong to human resources. This is a privacy benefit compared to common method of encoding the department in the address hierarchy. An additional benefit is that such groupings are arbitrarily flexible and are not constrained by the need to format information into addresses (address prefixes for instance). Since the addresses don't contain group information, group membership can be changed for an address without requiring the node to change its address.

3 Structure of identifier groups

Identifier groups can form a hierarchical structure within a mapping system domain. The diagram below illustrates a hierarchy containing two levels of groups and six identifier mapping entries at the leaves.



The diagram below provides an explicit example of using an identifier

to create a locator hierarchy. That is, multiple identifier locator operations are performed to get packets to destination. This is expected to be common in identifier-locator deployments. It is analogous to a packet going through a routing hierarchy where at each level the information applied became progressively more specific to the final destination (i.e. at each layer the prefix match is longer).

4 Interfaces

The mapping system interface is logically divided into the management interface and the query interface.

4.1 Management interface

The management interface is used to create and manipulate mapping entries and identifier groups.

The allowed operations on the management interface are:

- o Create groups
- o Set properties of a group, such as a locator or membership in another group in a group hierarchy
- o Change properties of a group
- o Create identifier mapping entries
- o Set identifier mapping properties such as locator or group membership
- o Change identifier mapping properties
- o Delete an identifier mapping entry
- o Remove all members from a group
- o Delete all identifier mappings in a group
- o Delete a group (that has no members)

Note that there is no public interface defined that will return all the members of a group. This is intended to limit visibility to this sensitive information.

4.2 Query interface

The query interface is used by devices that require identifier to locator mappings. This interface is read-only.

The basic operations in the query interface are:

- o Lookup locator for an identifier. In the case that a group hierarchy is present, the lookup request includes an indication as to which level in the hierarchy is applicable.
- o Lookup group information by group identifier. This is needed if the entry returned in a mapping entry indicates a group in a level of indirection. The internal structure for mapping entries which are members of the same group may reference a single group structure.
- o Request notifications of mapping entry changes if the mapping system supports pub/sub model. This includes notifications that a group membership has changed.
- o Request notifications of group changes. For example, if the locator for an identifier group changes.

5 Security Considerations

Access to mappings of group identifier to member identifiers MUST be strictly controlled. If this information is compromised, then privacy and anonymity of users could be undermined. In the case that the group identifiers refer to a single device, such as a UE in a mobile network, breach of the mapping from group identifier to identifiers may be sufficient to compromise individual user identities. Note that these concerns are not specific to identifier-locator mapping systems, but in any scenario where address assignment is done for devices.

The management interface should provide very strong authorization and employ encryption when communicating with the mapping system. The mapping system should enable security mechanisms associated with databases that contains sensitive information.

The query interface is always read-only, however this should also have strong access authorization methods for security and privacy.

A distributed identifier-locator mapping system should be deployed within a single administratively controlled domain. Low level information that potentially contains PII (Personally Identifiable Information) or specific location information should never be shared between administrative domains. It is conceivable that two networks could share a high level identifier-locator mapping system distinct

from their internal systems to support cross domain identifier-locator mappings. In this case, a locator hierarchy would be employed so as not to reveal any detailed information or PII. Specifically, identifier group information that refers specific devices and end locators for specific devices should not be visible.

6 IANA Considerations

7 References

7.1 Normative References

7.2 Informative References

Author's Address

Tom Herbert
Quantonium
Santa Clara, CA
USA

Email: tom@quantonium.net

INTERNET-DRAFT
Intended Status: Standard
Expires: June 2018

T. Herbert
Quantonium

December 21, 2017

Identifier Locator Addressing Mapping Protocol
draft-herbert-ila-ilamp-00

Abstract

Identifier-locator protocols rely on a mapping system that is able to map identifiers to locators. ILA nodes that perform ILA translations need to access the mapping system via a protocol. This document specifies the ILA Mapping Protocol that is used by ILA forwarding nodes and hosts to populate and maintain a cache of ILA mappings.

Status of this Memo

This Internet-Draft is submitted to IETF in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

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."

The list of current Internet-Drafts can be accessed at
<http://www.ietf.org/lid-abstracts.html>

The list of Internet-Draft Shadow Directories can be accessed at
<http://www.ietf.org/shadow.html>

Copyright and License Notice

Copyright (c) 2017 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

(<http://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 Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1	Introduction	4
2	Reference topology	4
2.1	Functional components	5
2.2	ILA forwarding nodes and hosts	5
2.2.1	ILA forwarding nodes	5
2.2.2	ILA hosts	5
2.2.3	ILA address to SIR address translation	5
2.2.4	ILA Forwarding	5
2.3	ILA routers	6
2.3.1	Forwarding routers	6
2.3.2	Mapping routers	6
2.3.3	ILA router synchronization	7
3	ILA Mapping Protocol	7
3.1	Common header format	8
3.2	Hello messages	8
4	ILAMP Version 0	9
4.1	Map request	9
4.2	Map information	10
4.3	Extended map information	11
4.4	Locator unreachable	12
4.5	Identifier and locator types	13
4.5.1	Identifier types	13
4.5.2	Locator types	13
5	Operation	14
5.1	Version negotiation	14
5.2	Populating an ILA cache	14
5.2.1	ILA Redirects	15
5.2.1.1	Proactive push with redirect	15
5.2.1.2	Redirect rate limiting	15
5.2.2	Map request/reply	15
5.2.3	Push mappings	16
5.3	Cache maintenance	16
5.3.1	Timeouts	16
5.3.2	Cache refresh	16
5.3.3	Cache timeout values	17
5.4	ILA forwarding node and host receive processing	17
5.5	Locator unreachable handling	17

5.6 Control Connections 18
5.7 Protocol errors 18
6 Security Considerations 19
7 IANA Considerations 20
8 References 20
8.1 Normative References 20
8.2 Informative References 20
Author's Address 20

1 Introduction

The ILA Mapping Protocol (ILAMP) is a control plane protocol that provides ILA nodes mapping information. ILA [ILA] nodes that perform ILA translation rely on a mapping system to provide identifier to locator mappings. There are two levels of mapping protocols to be defined: one used by ILA routers that require the full set of ILA mappings for a domain, and one used by ILA nodes that maintain a caches of mappings.

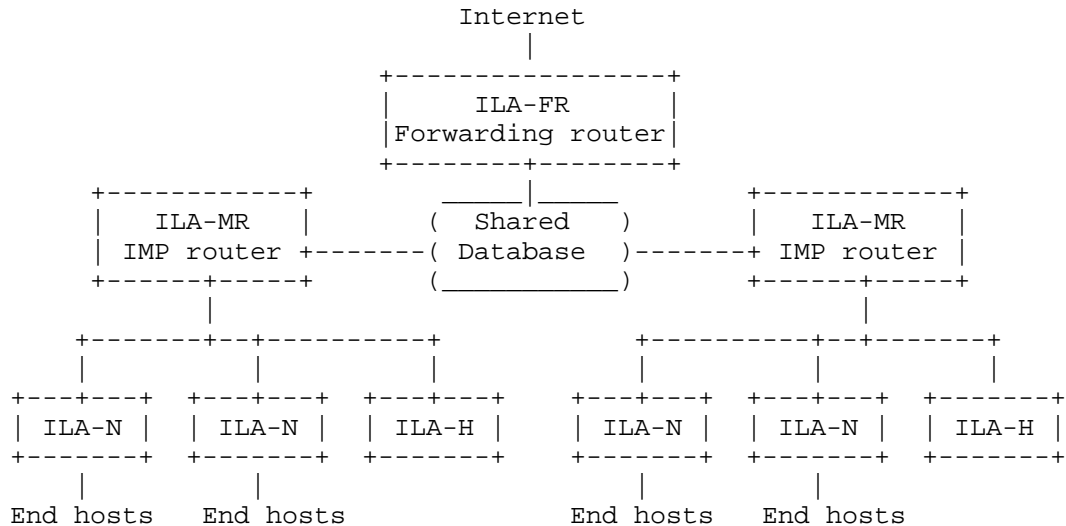
The ILA mapping system is effectively a key/value database that maps identifiers to locators. The protocol for sharing mapping information amongst ILA routers, nodes that maintain a full table of mappings, can thus be a database. A database schema for the ILA mapping system will be described in a separate document.

ILA separates the control plane from the data plane, so alternative control plane protocols may be used with a common data plane. For instance, BGP could be used as a mapping system protocol [ILABGP].

2 Reference topology

This section provides a reference topology for ILA.

The topology is general however can be adapted to specific ILA use cases such as data center virtualization and mobility networks.



2.1 Functional components

There are four types of functional nodes in the ILA architecture:

ILA-N: ILA forwarding nodes

ILA-H: ILA hosts

ILA-FR: ILA forwarding routers

ILA-MR: ILA mapping routers

2.2 ILA forwarding nodes and hosts

2.2.1 ILA forwarding nodes

ILA forwarding nodes (ILA-N) are deployed in the network infrastructure towards the edges to provide caches for ILA forwarding. ILA forwarding nodes have two functions: ILA address to SIR address translation and ILA forwarding. As indicated in the reference topology, forwarding nodes may be deployed near the point of device attachment (e.g. base station, eNodeB) of user devices (e.g. UEs).

2.2.2 ILA hosts

ILA hosts (ILA-H) are end hosts that participate in ILA. These may be servers that provide ILA to work with virtualization techniques such as VMs or containers. ILA hosts perform the same functions as ILA forwarding nodes, however the source of packets is local to the same host so there are some optimizations that may be applied.

2.2.3 ILA address to SIR address translation

ILA forwarding nodes and hosts perform ILA address to SIR address translation. This is a reverse ILA translation in order to restore the original addresses in a packet for delivery. Forwarding the packet on to the destination is done based on the SIR address. For instance, a forwarding node may map a SIR address to a layer 2 address of a directly attached device that has the SIR address. Note that this functionality is required somewhere in the path between the node that writes a locator into an address and the ultimate destination device.

2.2.4 ILA Forwarding

An ILA forwarding node or host may perform ILA translation and forward packets directly to peer ILA nodes in the same domain. The

mappings for this are maintained in a working set cache. As a cache there must be methods to populate, evict, and timeout entries. A cache is considered an optimization so the system should be functional without its use (e.g. if the cache has no entries).

2.3 ILA routers

ILA routers (denoted by ILA-R*) are deployed within the network infrastructure and collectively contain a database of all identifier to locator mappings in the domain, as well as all identifier group information for the domain. The database may be sharded across some number of ILA routers for scalability. ILA routers that maintain the database or a shard may be replicated for scalability and availability.

ILA routers provide two main functions: ILA forwarding and mapping resolution. An ILA router may perform one or the other of these functions or may provide both at the same time.

2.3.1 Forwarding routers

Forwarding routers (ILA-FR) perform ILA translation when packets enter a domain. A destination address of a packet that is a SIR address is translated to an ILA address. The process is that the router performs a lookup on the destination address in the mapping table and a locator is returned. The locator is written into the destination address of the packet (that is the high order sixty-four bits are overwritten with a locator).

In the case of a sharded database being used, the high order bits of the identifier indicate the shard number. This is included in a routing prefix so that the packet is routed to an ILA router that contains the database for the relevant shard.

2.3.2 Mapping routers

A mapping router (ILA-MR) provides ILA forwarding nodes and hosts mapping information. A mapping router may also perform ILA translations and forwarding. A mapping router implements ILAMP to communicate with ILA forwarding nodes and hosts. Mapping information is provided by a request/reply protocol, ILA redirects, or by a push mechanism.

An ILA router that is performing mapping resolution will respond to mapping requests from ILA forwarding nodes or ILA hosts. The mapping request protocol allows a node to request the locator information for an identifier address.

A mapping router can send ILA redirects to ILA forwarding nodes and ILA hosts in order to inform them of a direct ILA path. A redirect is sent to the upstream ILA forwarding node or host of the source which is determined by an ILA lookup the source address.

2.3.3 ILA router synchronization

ILA routers, both those that are forwarding and those that provide mapping resolution, must synchronize the contents of the database. This synchronization is done for each shard. When a change occurs to an identifier-locator mapping, for instance the locator for an identifier changes, the shard that contains the identifier must be synchronized in as little time to converge as possible.

There are a number of options to use for implementing the ILA mapping system and router protocol. One option is to use a key/value database (such as a NoSQL database like Redis).

The idea of the database is that each shard is a distributed database instance with some number of replicas. When a write is done in the database, the change is propagated throughout all of the replicas for the shard using the standard database replication mechanisms. Mapping information is written to the database using common database API and requires authenticated write permissions. Each ILA router can read the database for the associated shard to perform its function.

The specifics of applying a database and a database schema for ILA will be provided in other documents.

3 ILA Mapping Protocol

The ILA Mapping Protocol (ILAMP) is used between ILA forwarding nodes and ILA mapping routers. The purpose of the protocol is to populate and maintain the ILA mapping cache in forwarding nodes.

ILAMP defines redirects, a request/response protocol, and a push mechanism to populate the mapping table. Unlike traditional routing protocols that run over UDP, this protocol is intended to be run over TCP. TCP provides reliability, statefulness implied by established connections, ordering, and security in the form of TLS. Secure redirects are facilitated by the use of TCP.

ILAMP is used to send message between ILA routers and ILA forwarding nodes or hosts. The messages are sent over the TCP stream and must be delineated by a receiver. Different versions of ILAMP are allowed and the version used for communication is negotiated by Hello messages.

3.1 Common header format

All ILAMP messages begin with a two octet common header:

```

+-----+
| Type   | Length           |
+-----+

```

The contents of the common header are:

- o Type: Indicates the type of message. A type 0 message is a hello message. Types greater than zero are interpreted per the negotiated version.
- o Length: Length of the message in octets. This includes the common header. The minimal length of a message is 2 octets and the maximum length is 1,048,575 octets.

Following the two octet common header is variable length data that is specific to the version and type the message.

3.2 Hello messages

Hello messages indicate the versions of ILAMP that a node supports. Hello message MUST be sent by each side as the first message in the connection.

The format of an ILAMP Hello message is:

```

+-----+
| 0     | 4           |R| Rsvd      | MinV | MaxV |
+-----+

```

The contents of the Hello message are:

- o Type = 0. This is indicates the type is a Hello message.
- o Router bit: Indicates the sender is an ILA router. If the sender is an ILA forwarding node or host this bit is cleared.
- o Rsvd: Reserved bits. Must be set to zero on transmit.
- o MinV: Minimum version number supported by the sending node.
- o MaxV: Maximum version number supported by the sending node.

Version numbers are from 0 to 15. This document describes version 0 of ILAMP.

4 ILAMP Version 0

The message types in version 0 of IMP are:

- o Map request (Type = 1)
- o Map information (Type = 2)
- o Extended map information (Type = 3)
- o Locator unreachable (Type = 4)

4.1 Map request

A map request is sent by an ILA forwarding node or host to an ILA mapping router to request mapping information for a list of identifiers. The format of a map request is:

```

+++++
| 0 | Length | Rsvd | IDType |
+++++
| ~ Identifier ~ | ent
| | |
+++++

```

The contents of the map request message are:

- o Type = 1. This indicates the type is map request.
- o Length: Set to 4 plus the size of the identifier times the number of identifiers in the list.
- o Rsvd: Reserved bits. Must be set to zero when sending.
- o IDType: Identifier type. Specifies the identifier type. This also implies the length of each identifier in the request list. Identifier types are defined below.
- o Identifier: An identifier of type indicated by IDType. The size of an identifier is specified by the type.

The Identifier field is repeated for each identifier in the list. The number of identifiers being requested is $(\text{message length} - 4) / (\text{identifier size})$.

4.2 Map information

Map information messages are sent by an ILA router to an ILA forwarding node or host. This message provides a list of identifier to locator mappings. The format of a map information message is:

```

+-----+-----+-----+-----+-----+-----+-----+-----+-----+
|  2  |          Length          | Rsvd  | SubType|LocType| IDType|
+-----+-----+-----+-----+-----+-----+-----+-----+
|                                     |                                     | \
|                                     | Identifier                         |  | e
|                                     |                                     |  | n
+-----+-----+-----+-----+-----+-----+-----+-----+ t
|                                     |                                     | ~
|                                     | Locator                           |  |
|                                     |                                     | /
+-----+-----+-----+-----+-----+-----+-----+

```

The contents of the map information message are:

- o Type = 2. This indicates the type is map information.
- o Length: Set to 4 plus the size of an identifier and locator times the number of identifier/locator pairs in the list.
- o Rsvd: Reserved bits. Must be set to zero when sending.
- o SubType: Specifies the reason that ILA-R sent this message. Sub types are
 - o 0: Redirect
 - o 1: Map reply to a map request
 - o 2: Push map information
- o LocType: Locator type. Specifies the locator type. This also implies the length of each locator in the list. Locator types are defined below.
- o IDType: Identifier type. Specifies the identifier type. This also implies the length of each identifier in the list. Identifier types are defined below.
- o Identifier: An identifier of type indicated by IDType. The size of an identifier is specified by the type.
- o Locator: A locator of type indicated by LocType. The size of a

locator is specified by the type.

The Identifier/Locator pair is repeated for each mapping being reported in the list. The number of identifiers being requested is (message length - 4) / (identifier size + locator size).

4.3 Extended map information

An extended locator map information message is sent by an ILA router to associate more than one locator with an identifier as well as providing an expiration time for an identifier locator mapping and additional locator specific attributes. The format of an extended map information is:

```

+-----+-----+-----+-----+-----+-----+-----+-----+-----+
|   3   |           Length           | Rsvd  | SubType|LocType|IDType |
+-----+-----+-----+-----+-----+-----+-----+-----+ <-+
|                                           | ~ |
|                               Identifier | ~ |
|                                           |  r
+-----+-----+-----+-----+-----+-----+-----+-----+ e
| Num locator |           Record timeout           | c
+-----+-----+-----+-----+-----+-----+-----+ \ o
| Prio  | Rsvd  | Weight          |           Rsvd          | e r
+-----+-----+-----+-----+-----+-----+-----+ t d
|                                           | t |
|                               Locator   | ~ |
|                                           | / |
+-----+-----+-----+-----+-----+-----+-----+ <-+

```

The contents of the map reply message are:

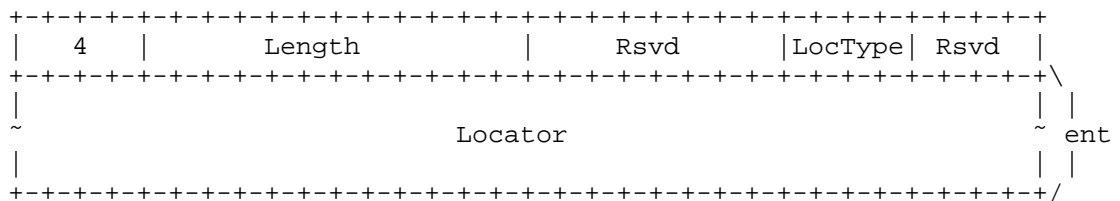
- o Type = 3. This indicates an extended map information message
- o Length: Set to 4 plus the sum of sizes for each identifier record in the list.
- o Rsvd: Reserved bits. Must be set to zero when sending.
- o SubType: Specifies the reason that an ILA router sent this messages. Sub types are:
 - o 0: Map reply to a map request
 - o 1: Redirect
 - o 2: Push map information

- o LocType: Locator type. Specifies the locator type. This also implies the length of each locator in the list. Locator types are defined below.
- o IDType: Identifier type. Specifies the identifier type. This also implies the length of each identifier in the list. Identifier types are defined below.
- o Num locator: Number of locators being reported for an identifier.
- o Record timeout: The time to live for the identifier information in seconds. A value of zero indicates the default is used.
- o Priority: Relative priority of a locator. Locators with higher priority values have preference to be used. Locators that have the same priority may be used for load balancing.
- o Weight: Relative weights assigned to each locator. In the case that locators have the same priority the weights are used to control how traffic is distributed. A weight of zero indicates no weight and the mapping is not used unless all locators for the same priority have a weight of zero.
- o Locator: A locator of type specified in LocType.

The identifier record is repeated for each mapping being reported and the locator entry is repeated for each locator being reported for an identifier. The total number of identifiers being reported is determined by parsing the message.

4.4 Locator unreachable

A locator unreachable message is sent by an ILA router to ILA forwarding node or host in the event that a locator or locators are known to no longer be reachable. The format of a locator unreachable message is:



The contents of the locator unreachable message are:

- o Type = 4. This is indicates the type is a locator unreachable

message.

- o Length: Set to 4 plus the size of the locator times the number of locators in the list.
- o Rsvd: Reserved bits. Must be set to zero when sending.
- o LocType: Locator type. Specifies the locator type. This also implies the length of each locator in the list. Locator types are defined below.
- o Locator: A locator of type indicated by LocType. The size of a locator is specified by the type.

The Locator field is repeated for each locator in the list. The number of locators being reported is $(\text{message length} - 4) / (\text{locator size})$.

4.5 Identifier and locator types

4.5.1 Identifier types

Identifier types used in IDType fields of ILAMP messages are:

- o IPv6 address (IDType = 1): 128 bit IPv6 address
- o ILA Identifier (IDType = 2): 64 bit ILA identifier
- o 32 bit index (IDType = 3): 32 bit index into an identifier table
- o 64 bit index (IDType = 4): 64 bit index into an identifier table

For the table index types it is assumed that a table mapping index to and identifier is shared with ILA forwarding nodes and hosts.

4.5.2 Locator types

Locator types used in LocType fields of ILAMP messages are:

- o IPv6 address (LocType = 1): 128 bit IPv6 address
- o ILA Identifier (LocType = 2): 64 bit ILA locator
- o 32 bit index (LocType = 3): 32 bit index into an locator table
- o 64 bit index (LocType = 4): 64 bit index into an locator table

For the table index types it is assumed that a table mapping index to

and locator is shared with ILA forwarding nodes and hosts.

5 Operation

5.1 Version negotiation

The first message sent by each side of an ILAMP connection is a Hello message. Hello messages contain the minimum and maximum versions of ILAMP supported. The minimum and maximum values form an inclusive range.

When a host receives an ILAMP Hello it determines which version is negotiated. The negotiated version is the maximum version number supported by both sides. For instance, if a node advertises a minimum version of 0 and maximum of 1 and receives a peer Hello message with a minimum version of 0 and maximum of 2; then the negotiated version is 1 since that is the greatest version supported by both sides. The peer host will also determine that 1 is the negotiated version.

If there is no common version supported between the peers, that is their supported version ranges are disjoint, then version negotiation fails. The connection **MUST** be terminated and error message **SHOULD** be logged.

If both sides set the router bit or both clear the router bit in a Hello message, then this is an error and the connection **MUST** be terminated and error message **SHOULD** be logged. Both sides cannot have the same role in an ILAMP session.

5.2 Populating an ILA cache

ILA forwarding nodes and ILA hosts maintain a cache of identifier to locator mappings. There are three means that this cache can be populated by ILAMP:

- o ILA redirect
- o Mapping request/reply
- o Push mappings

ILA redirects are **RECOMMENDED** to be the primary means of obtaining mapping information. Request/reply and push mappings may be used in limited circumstances, however generally these techniques don't scale.

Note that forwarding nodes and hosts do not hold packets that are pending mapping resolution. If a node does not have a mapping for a

destination in its cache then packet is forwarded in the network. The packet should be translated by an ILA router and sent to the proper destination node.

5.2.1 ILA Redirects

A mapping router can send ILA redirects in conjunction with forwarding packets. Redirects are sent to ILA forwarding nodes and ILA hosts in order to inform them of a direct ILA path. A redirect is sent to the upstream ILA forwarding node or host of the source which is determined by an ILA lookup on the source address of the packet being forwarded. The found locator is used to infer an address of the ILA forwarding node or host. For instance, the address of the forwarding node might be <locator>::1 where <locator> is a sixty-four bit prefix and 0:0:0:1 is reserved as a special identifier. Note that this technique assumes a symmetric path towards the source. If a redirect is sent then the received packet that motivated the redirect MUST be ILA translated and forwarded by the router.

5.2.1.1 Proactive push with redirect

In addition to sending an ILA redirect to the ILA forwarding node or host, a mapping router MAY send an ILA push to the ILA forwarding node or host of the destination to inform it of the identifier to locator mapping for the source address in a packet. This is an optimization to push the ILA translation that will be used in the reverse direction of the communications. In order to do this, the mapping router performs an ILA lookup on the source address (which should already be done to perform the redirect). An ILA push message is then sent to the forwarding node or host based on its locator.

5.2.1.2 Redirect rate limiting

A mapping router SHOULD rate limit the number of redirects it sends to a forwarding node or host for each redirected address. The rate limit SHOULD be configurable. The default SHOULD be that no more than one redirect is sent every one half of the minimum identifier timeout being used. The minimum rate limit SHOULD be to send no more than one redirect per second per redirected identifier. If a mapping change is detected the rate limiting SHOULD be reset so that redirects for a new mapping can be sent immediately.

5.2.2 Map request/reply

A forwarding node or host may send a map request message to obtain mapping information for a locator. If the receiving mapping router has the mapping information it responds with a map information message. If the mapping router does not have a mapping entry for the

requested identifier it MAY reply with in all zeros locator (LocType = 2 and 64 bit locator is all zeroes).

Map requests are NOT RECOMMENDED to be used to populate entries in the cache table that are not present. The problem with this technique is that an ILA forwarding node or host may generate a map request for each new destination that it gets from a downstream end host. A downstream end host could launch a Denial of Service (DOS) attack whereby it sends packets with random destination addresses that requires a mapping looking. In the worse case scenario the mapping router would send a map request for every packet received. Rate limiting the sending of map requests does not mitigate the problem since that would prevent the cache from getting mappings for legitimate destinations.

5.2.3 Push mappings

A mapping router may push mappings to an ILA forwarding node or host without being requested to do so. This mechanism could be used to pre-populate an ILA cache. Pre-populating the cache might be done if the network has a very small number of identifiers or there are a set of identifiers that are likely to be used for forwarding in most ILA forwarding nodes and hosts (identifiers for common services in the network for instance). When a mapping router detects a changed mapping, the locator changes for instance, the new mapping can be pushed to the ILA nodes and hosts.

The push model is NOT RECOMMENDED as a primary means to populate an ILA cache since this does not scale. Conceivably, one could keep track of all ILA mappings and to which nodes the mapping information was provided. When a mapping changes, mapping information could be sent to those nodes that expressed interest. Such a scheme will not scale in deployments that have many mappings.

5.3 Cache maintenance

5.3.1 Timeouts

A node SHOULD apply a timeout for the mapping entry using either the default timeout or record timeout if one was received in an extended map information message. If the timeout fires then the mapping entry is removed. Subsequent packets may cause a mapping router to send a redirect so that the mapping entry gets repopulated in the cache.

5.3.2 Cache refresh

In order to avoid cycling a mapping entry with a redirect for a mapping that times out, a node MAY try to refresh the mapping before

timeout. This should only be done if the cache entry has been used to forward a packet during the timeout interval.

A cache refresh is performed by sending a map request for an identifier before its cache entry expires. If a map information messages is received for the identifier then the timeout can be reset and there are no other side effects.

5.3.3 Cache timeout values

The RECOMMENDED default timeout for identifiers is one minute. If a node sends a map request to refresh a mapping, the RECOMMENDED default is to send the request ten seconds before the the mapping expires.

5.4 ILA forwarding node and host receive processing

If an ILA forwarding node or host receives an ILA addressed packet with its locator it will check its local mapping database to determine if the identifier is local. If the identifier is local, a forwarding node will forward the packet to its destination after ILA to SIR address translation has been done on the packet's destination address. Similarly, an ILA host will receive the packet into it's local stack after ILA to SIR address translation.

If the identifier is not local then the ILA forwarding node or host will perform ILA to SIR address translation on the destination address and forward the packet into the network. This may happen if an end node has moved to be attached to a different ILA forwarding node in host and the new locator has not yet been propagated to all ILA nodes. The packet should traverse a mapping router which can send an ILA redirect back the source's ILA forwarding node or host as described above.

When a node migrates its point of attachment from one forwarding node or host to another, the local mapping on the old node is removed so that any packets that are received and destined to the migrated identifier are re-injected with a SIR address as described above. A "negative" mapping with timeout may also be set ensure that the node is able to infer the SIR address from a destination address (e.g. would be needed with foreign identifiers).

5.5 Locator unreachable handling

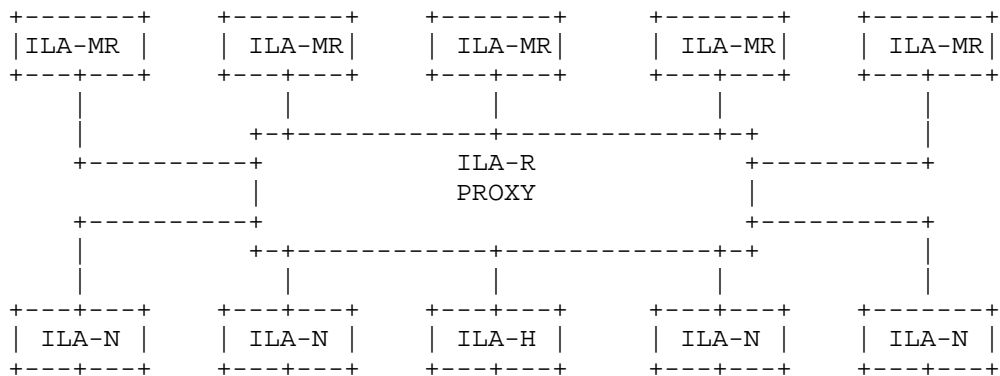
When connectivity to a locator is loss the mapping system should detect this. A locator unreachable message MAY be sent by an ILA router to ILA forwarding nodes or hosts informing them that a locator is no longer reachable. Each forwarding node or host SHOULD remove

any cache entries using that locator and MAY send a map request for the affected identifiers.

5.6 Control Connections

ILA nodes and routers must create ILAMP connections to all the mapping routers that might provide routing information. In a simple network there may be just one mapping router to connect to. In a more complex network with ILA routers for sharded and replicated mapping system database there may be many. A list of ILA routers to connect to is provided to each ILA forwarding node and host. This list could be provided by configuration, a shared database, or an external protocol to ILAMP.

Conceivably, the number of mapping routers in a network that might report mapping information to a host could be quite large (into the thousands). If managing a large number of connections at the ILA forwarding nodes or hosts is problematic, ILA mapping router proxies could be used that consolidate connections as illustrated below:



In the above diagram a single ILA mapping router proxy serves five ILA routers and five ILA forwarding nodes and nodes. The proxy creates one connection to each router and each ILA forwarding node and host creates one connection to the proxy.

5.7 Protocol errors

If a protocol error is encountered in processing ILAMP messages a peer MUST terminate the connection. It SHOULD log the error and MAY attempt to restart the connection. There are no error messages defined in ILAMP.

Protocol errors include mismatch of length for given data, reserved bit not set to zero, unknown identifier type or locator types,

unknown type, unknown sub-type, and loss of message synchronization in a TCP stream. Note that if the end of a message does not end on field or record boundary this also considered a protocol error.

6 Security Considerations

ILAMP must have protection against message forgery. In particular secure redirects and mapping information message are required to prevent and attacked from spoofing messages and illegitimately redirecting packets. This security is provided by using TCP connections so that origin of the messages is never ambiguous.

Transport Layer Security (TLS) [RFC5246] MAY be used to provide secrecy, authentication, and integrity check for ILAMP messages.

The TCP Authentication Option [RFC5925] MAY be used to provide authentication for ILAMP messages.

7 IANA Considerations

8 References

8.1 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>>.
- [RFC4291] Hinden, R. and S. Deering, "IP Version 6 Addressing Architecture", RFC 4291, February 2006.
- [ILA] Herbert, T., and Lapukhov, P., "Identifier Locator Addressing for IPv6" draft-herbert-intarea-ila-00

8.2 Informative References

- [RFC5246] Dierks, T. and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.2", RFC 5246, DOI 10.17487/RFC5246, August 2008, <<https://www.rfc-editor.org/info/rfc5246>>.
- [RFC5925] Touch, J., Mankin, A., and R. Bonica, "The TCP Authentication Option", RFC 5925, DOI 10.17487/RFC5925, June 2010, <<https://www.rfc-editor.org/info/rfc5925>>.
- [BGPILA] Lapukhov, P., "Use of BGP for dissemination of ILA mapping information" draft-lapukhov-bgp-ila-afi-02

Author's Address

Tom Herbert
Quantonium
Santa Clara, CA
USA

Email: tom@quantonium.net

INTERNET-DRAFT
Intended Status: Informational
Expires: September 2018

T. Herbert
Quantonium
K. Bogineni
Verizon

March 6, 2018

Identifier Locator Addressing for Mobile User-Plane
draft-herbert-ila-mobile-01

Abstract

This document discusses the applicability of Identifier Locator Addressing (ILA) to the user-plane of mobile networks. ILA allows a means to implement network overlays without the overhead, complexities, or anchor points associated with encapsulation. This solution facilitates highly efficient packet forwarding and provides low latency and scalability in mobile networks. ILA can be used in conjunction with techniques such as network slices and Network Function Virtualization to achieve optimal service based forwarding.

Status of this Memo

This Internet-Draft is submitted to IETF in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

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."

The list of current Internet-Drafts can be accessed at
<http://www.ietf.org/lid-abstracts.html>

The list of Internet-Draft Shadow Directories can be accessed at
<http://www.ietf.org/shadow.html>

Copyright and License Notice

Copyright (c) 2018 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 (<http://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 Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1	Introduction	4
2	Conventions and Terminology	5
3	Motivation	5
4	Reference topology	6
4.1	ILA routers (ILA-R)	6
4.1.1	Forwarding routers	7
4.1.2	Mapping resolution	7
4.2	ILA forwarding nodes (ILA-N)	7
4.2.1	ILA to SIR address transformation	7
4.2.2	ILA forwarding	8
4.3	ILA hosts (ILA-H)	8
4.4	ILA management (ILA-M)	9
5	Data plane operation	9
5.1	SIR to ILA transformation	10
5.2	ILA to SIR transformation	11
5.3	Data path efficiency	11
5.4	Alternative data path use cases	12
5.5	Locator changing	12
5.6	ICMP handling	12
6	Control plane	12
6.1	ILA router mapping database	13
6.1.1	ILA with BGP	13
6.1.2	Key/value store	13
6.2	ILA Mapping Protocol	13
6.3	Address assignment	14
6.3.1	Singleton address assignment	14
6.3.2	Network prefix assignment	14
7	ILA in 5G networks	15
7.1	Architecture	15
7.2	Protocol layering	16
7.3	Control plane between ILA and network	16
7.4	ILA and network slices	17

8 Security considerations 18

8.1 Data plane security 18

8.2 Control plane security 19

8.3 Privacy in address assignment 20

9 References 21

9.1 Normative References 21

9.2 Informative References 22

Authors' Addresses 22

1 Introduction

In mobile networks, mobility management systems provide connectivity while mobile nodes move around. A control-plane system signals movements of a mobile node, and a user-plane establishes tunnels between mobile nodes and anchor nodes over IP based backhaul and core networks.

This document discusses the applicability of Identifier Locator Addressing (ILA) to those mobile networks. ILA is a form of identifier/locator split where identity and location of a node are disassociated in IP addresses. ILA nodes transform destination addresses of packets by overwriting part of the address with a locator. The locator provides the topological address for forwarding a packet towards its destination. Before a packet is delivered to the end destination, the destination address is reverted to its original value.

An ILA mobile user-plane implementation needs both data plane and control plane components.

The data plane includes the ILA transformation processing as well as handling to maintain conformance with IP protocols. The ILA data plane is described in [ILA].

The control plane's primary function is to maintain a mapping database that is shared amongst ILA nodes. The mapping database contains entries for the mobile nodes in the network, and the number of mapping entries is expected scale into the billions. In order to scale, a two level hierarchy of ILA nodes is defined by ILA routers and ILA forwarding nodes.

ILA routers maintain a full set of ILA mappings. Routers may be replicated for redundancy and load balancing. The mapping system may also be sharded, so that each router is responsible for a shard. Routers use a protocol to synchronize the mappings for each shard.

ILA forwarding nodes perform a reverse ILA transformations to restore the destination address in packets before delivery. A forwarding node can also maintain a cache of ILA mappings to perform transformations on intra domain traffic as an optimization to avoid having to forward packets through ILA routers. Forwarding nodes are typically located close to the mobile nodes. The ILA Mapping Protocol [ILAMP] is used between forwarding nodes and ILA routers to manage the cache.

2 Conventions and Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

ILA related terms are defined in [ILA].

3 Motivation

Emerging applications such as VR, AR, and autonomous vehicle communication require very low latency, high bandwidth, and high reliability. For mobile devices, these requirements must not only be met when the device is stationary, but also across handover during mobile events. Mobility needs to be a seamless operation where IP addresses and connections are maintained. In a second dimension, the number of connected mobile devices, including a large contingent of IoT devices, is expected to grow by several orders of magnitude within a few years as enabling technologies such as 5G are deployed.

The convergence of mobile networks and datacenter networks is also pertinent. Simple physics (i.e. speed of light) dictates that very low latency for applications (order of less than five milliseconds) can only be achieved by placing application servers in close proximity to clients and minimizing the number of network hops. An emerging trend is for providers to house datacenters within the network to run applications. For similar reasons, many providers are integrating multi-tenant cloud services directly in their mobile networks. The upshot is that mobile networks need to support the convergence of mobile devices, datacenter virtualization, and cloud. A single solution framework for all of this is desirable.

Current mobile architecture is hitting the limits of scalability and performance. In particular, anchor points used in 3GPP have become single points of failure, bottlenecks, and lead to sub-optimal triangular routing. The anchor model is also inflexible in attempts to leverage services of the transport network such as network slices and Network Function Virtualization. The control plane to manage millions of GTP tunnels is complex and difficult to scale. GTP-U is narrowly defined for a particular use case, which makes it difficult to leverage for other use cases. The use of any in-network tunneling, including GTP, raises issues of overhead, MTU and fragmentation, security, and other complexities.

ILA is a proposed alternative to GTP-U and encapsulation. It does not require anchors and simplifies both the data plane and control plane. ILA has zero wire overhead so there are no issues around MTU and fragmentation. Its use is transparent to the network, and it is

compatible with existing hardware and commonly deployed protocol optimizations. ILA is a general network overlay protocol to meet the requirements of use cases in a converged network. User Plane Functions (UPF) with ILA are lightweight and stateless such that they can be brought up quickly as needed.

4 Reference topology

Figure 1 shows an example topology of ILA in a mobile network.

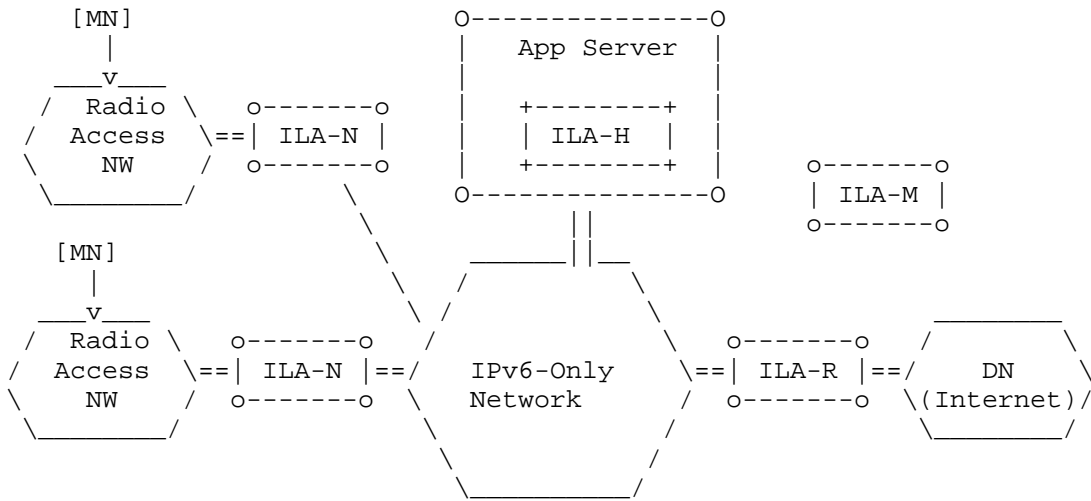


Figure 1: Mobile User-plane with ILA

There are four types of functional nodes in the ILA architecture:

- o ILA routers (ILA-R)
- o ILA forwarding nodes (ILA-N)
- o ILA hosts (ILA-H)
- o ILA managment (ILA-M)

4.1 ILA routers (ILA-R)

ILA routers are deployed within the network infrastructure and collectively contain a mapping database of all identifier to locator mappings in an ILA domain. The database may be sharded across the identifier space by some number of ILA routers for scalability. ILA

routers may also be replicated for scalability and availability.

ILA routers provide two main functions: ILA forwarding and mapping resolution. An ILA router may perform one both of these functions at the same time. If a router performs both functions it may send ILA redirects.

4.1.1 Forwarding routers

Forwarding routers perform ILA transformations when packets enter an ILA domain. A destination address of a packet that is a SIR address is transformed to an ILA address. The process is that the router performs a lookup on the destination address in a mapping table and a locator is returned. The locator is written into the destination address of the packet (typically the high order sixty-four bits are overwritten with a locator).

In the case of a sharded database, the high order bits of the identifier indicate the shard number. This is included in a routing prefix so that the packet is routed to an ILA router that contains the database for the indicated shard.

4.1.2 Mapping resolution

An ILA router that is performing mapping resolution will respond to mapping requests from ILA forwarding nodes or ILA hosts (these are described below). The mapping request protocol allows the caller to request the locator for an identifier address.

4.2 ILA forwarding nodes (ILA-N)

ILA forwarding nodes are deployed in the network infrastructure towards the edges to provide ILA transformations for end devices. ILA forwarding nodes have two functions: ILA to SIR address transformation and ILA forwarding. As indicated in the reference topology, forwarding nodes may be deployed near the point of device attachment (e.g. base station, eNodeB) of mobile nodes.

4.2.1 ILA to SIR address transformation

In the path towards the end devices, forwarding nodes perform ILA to SIR address transformation. That is, they perform a reverse ILA transformation in order to restore the original addresses in packet. Forwarding the packet on to the destination is done based on the SIR address. For instance, an eNodeB may map a SIR address to a layer 2 address of the attached device that has the SIR address. Note that this functionality is required somewhere in the path between the ILA node that writes a locator into an address and the ultimate

destination device (e.g. a UE). It is not recommended that this functionality is implemented on end user devices.

When a node migrates its point of attachment from one ILA-N to another, the local mapping on the old ILA-N is removed. If an ILA addressed packet is received by an ILA-N for which there is no local mapping, then the packet is forwarded back into the network with a destination SIR address. The packet should be forwarded through an ILA router that can perform the transformation for the new ILA-N. A "negative" mapping with timeout may also be set in an ILA-N to ensure that ILA-N is able to infer the SIR address (e.g. would be needed with non-local identifiers).

4.2.2 ILA forwarding

A forwarding node may perform ILA transformation and forward packets directly to peer ILA nodes in the same domain. The mappings for this are maintained in a working set cache in each ILA-N. As a cache there must be methods to populate, evict, and timeout entries. A cache is considered an optimization, so the system should be functional without its use (e.g. if the cache has no entries). The possibility of Denial of Service attack (DOS) on a cache being populated by unmanaged outside events, in this case mobile devices sending packets to arbitrary destinations, must be considered in the cache design.

If a packet is received by an ILA forwarding node from a downstream node that is destined to another node in the same ILA domain for which there is no existing cache entry, then:

- o The packet is forwarded by address. The SIR address plus shard identifier prefix will route the packet to a forwarding ILA router which will perform ILA transformation of the packet to reach its destination.
- o An ILA router may return an ILA redirect to inform the forwarding node of a direct ILA mapping.
- o If the forwarding node gets a mapping from an ILA router, then subsequent packets for the destination can be directly sent using the mapping. Note that a forwarding node does not hold packets that are pending mapping resolution.

4.3 ILA hosts (ILA-H)

ILA host are forwarding nodes that are embedded in end servers to provide ILA transformation. Since an ILA host is integrated with the host stack sourcing packets, there are opportunities for optimizing processing.

ILA is not recommended to run on end user devices, however there may be servers or other end devices that are in the provider network that might benefit from participating in ILA (this is illustrated in the reference topology above). A server that implements ILA forwarding can directly send to ILA peers in the same domain to avoid triangular routing.

4.4 ILA management (ILA-M)

The ILA management node provides the interface between the ILA infrastructure and mobile management of a network. Similar to ILA-Rs there may be multiple ILA-Ms in the network and they can be replicated for redundancy and load distribution. Data managed by ILA-Ms needs to be synchronized across ILA-Ms. It is conceivable that the set of ILA-Ms could be split into shards serving different geographic area in order to localize data. ILA-Ms may be co-located with ILA-Rs so that there is a fast path between them.

The management nodes are responsible for:

- o Receiving notifications from the session management in the mobile network. Notifications of interest include: when mobile nodes attach to the network, are removed from the network, or change their point of attachment in the network (i.e. they move).
- o Managing identifier groups. Identifier groups are sets of identifiers (nodes) that share common properties [ILAGRPS]. In a mobile network, identifier groups are used to represent all the identifiers assigned to a mobile node. Each mobile node will have its own identifier group.
- o Writing identifier locator mappings into the ILA mapping database. The written content is based on the information provide by session management.
- o Changing the mapping table when a locator for an identifier, group, or mobile node changes. A locator for a device changes when its point of attachment changes.
- o Creating identifiers for attached devices. Identifiers may be persistent so that each time a device attaches it gets the same identifier.
- o Registering ILA-Rs, ILA-Ns and their locators. ILA-Ms coordinate the operation of ILA nodes in the network.

5 Data plane operation

ILA performs transformations on IPv6 addresses of packets in flight. A SIR to ILA address transformation overwrites the destination address with a locator address for forwarding over a network. An ILA to SIR address transformation restores an IP address to its original contents. The transformations are always paired so that a SIR to ILA address transformation is always undone before delivery. End hosts and applications only see SIR addresses. Effectively, ILA is a mechanism to implement transparent network overlays. Note the process is specifically called a "transformation" as opposed to "translation" which distinguishes ILA from NAT. NAT translations are not undone before reception and NAT is not transparent to the end points.

5.1 SIR to ILA transformation

SIR to ILA address transformations may be performed by ILA routers, ILA forwarding nodes, and ILA hosts.

SIR to ILA transformation is done by a lookup on the destination address in a mapping table. On an ILA router this table contains all the entries for the shard the router serves. On a forwarding node or host, the table is a cache of entries. If a corresponding entry is found, then a locator is returned. The locator is written into the destination address.

If checksum neutral mapping is being used to preserve transport layer checksums, then that is indicated in the mapping entry. Checksum neutral mangles the low order sixteen bits of the identifier portion of the address. The checksum difference between the SIR prefix and the locator is added into to the low order sixteen bits of the identifier.

If an ILA router does not find a match on the destination address in its table then the packet is dropped as having no route to host.

If an ILA forwarding node or host does not find a match on the destination address, then it forwards the packet unchanged. The packet may encounter an ILA router that performs the transformation.

In response to forwarding a packet, a router might send an ILA redirect to an ILA forwarding node. A redirect informs a node of an ILA mapping that may be cached to avoid triangular routing when forwarding subsequent packets. The destination of a redirect is the upstream forwarding node of the source of packet. An ILA router can determine this by performing an ILA lookup on the source address of the packet being forwarded. This assumes that the source is a SIR address for the ILA domain and that the use of ILA is symmetric so that the lookup reveals the correct forwarding node; this needs to be accounted for in network design.

5.2 ILA to SIR transformation

Transformed packets are forwarded to an ILA-N or ILA-H based on normal routing of the packet with a locator in the its destination address (upper sixty-four bits). When a node receives the packet it first performs an ILA to SIR address transformation by mapping the received locator (one local to the node) to a SIR address. If checksum neutral mapping has been done, the lower sixteen bits in the identifier must be fixed up. This is done by subtracting the checksum difference of the SIR address and locator from the low order bits of the identifier (the opposite operation of setting the checksum neutral bits).

After transforming a destination back to SIR address, a lookup is performed on the identifier to determine if it is local (that is it refers to a node that is downstream of the ILA node). If the node is local, it is forwarded downstream using normal mechanisms of the network. If the node is not local, the SIR addressed packet is forwarded back into the network. The packet should traverse an ILA router that can transform its destination to the correct locator and possibly send a redirect towards the source.

5.3 Data path efficiency

There basic operations of ILA address transformation, either SIR to ILA or ILA to SIR, are:

- 1) Read destination address from a packet.
- 2) Lookup all or are part of the destination address in table. This is a fixed length lookup.
- 3) Overwrite all or part of the destination address with a locator value returned from the lookup.
- 4) Fix the checksum neutral mapping bits in the identifier.
- 5) Forward the resultant packet.

The computationally intensive operations in this path are the lookup and checksum neutral processing.

The lookup operation is on a fixed length key so a simple hash table can be used. It is also amenable for use with a hardware TCAM. On an ILA host, an ILA mapping may be cached with a connection context so that a lookup does not to be performed for every packet sent on the connection.

Checksum neutral processing entails 1's complement arithmetic over sixty-four or 128 bit values. In the case that the full 128 bit identifier address is a one-to-one mapping with a locator address, then the checksum computation is constant for a mapping and can be precomputed and saved with the mapping.

5.4 Alternative data path use cases

ILA supports multicast encoding, virtual networking modes, and IPv4/IPv6 translation. These require different processing, and in the case of IPv4/IPv6 translation the size of the packet increases. However, these alternative cases should not fundamentally increase the cost of the lookups since instructions for alternative processing can be returned by a lookup.

5.5 Locator changing

ILA allows multiple locator transformations to effectively implement hop-by-hop source routing. This can be used to deliberately have a packet visit some set of nodes. This might also be used in the case where two domains exchange ILA mappings, but only share locators that are ingress points in their network and not final locators of a node. This would be done to protect user location from being exposed.

5.6 ICMP handling

A packet whose destination address is an ILA address may generate an ICMP error. In this case the ICMP data will contain an IPv6 header whose destination is an ILA address. If a sender receives an ICMP error with an ILA address as the destination of the original packet, it won't recognize the destination address as one that it sent to and this may leak information about internal nodes of the network. To prevent this from happening, upstream ILA-Ns or ILA-Hs of an end node can filter ICMP packets. When an ICMP packet is received by these nodes, an ILA destination address can be transformed back to a SIR address by performing a reverse lookup.

6 Control plane

This section describes the ILA control plane for the mobile user-plane.

The ILA control plane is separate from the control plane of the mobile network. An interface between the session management of the network and the control plane is needed to get device information and point of attachment. The intent is that the interface is well compartmentalized to minimize the amount of specialization needed to adapt ILA for use in different access technologies.

6.1 ILA router mapping database

There are a number of options to use for implementing the ILA mapping system and router protocol amongst ILA-Rs. The mapping database must be able to scale and provide fast converge when mobile nodes move within the network.

6.1.1 ILA with BGP

A traditional routing protocol could be used for route dissemination. [BGPILA] defines multiprotocol extensions to BGP for distributing ILA mappings.

6.1.2 Key/value store

A mapping database is logically a simple key/value store where the lookup key is fixed length (sixty-four or 128 bytes). This characteristic affords the possibility of using a key/value database in lieu of traditional routing protocols.

The idea of the key/value database is that each shard is a distributed database instance with some number of replicas. When a write is done in the database, the change is propagated throughout all of the replicas for the shard using the standard database replication mechanisms. Mapping information is written to the database using common database API and requires authenticated write permissions. Each ILA router can read the database for the associated shard to perform its function.

The database is assumed to be (mostly) persistent and recoverable if database nodes are lost. The selection of an ILA router shard and shard instance is idempotent and stateless per packet, so that shards and shard replicas can be dynamically added or removed.

6.2 ILA Mapping Protocol

The ILA Mapping Protocol [ILAMP] is used between ILA forwarding nodes and ILA mapping resolution routers. The purpose of the protocol is to populate and maintain the ILA mapping cache in forwarding nodes.

ILA forwarding nodes can use a pull model (request/response), push model (pub/sub), or redirects to populate the mapping table. ILAMP runs over TCP which provides reliability, statefulness implied by established connections, allows use of HTTP and RESTful APIs, and standard security in the form of TLS.

The protocol is composed of message primitives:

- o Map request: Sent by an ILA-N or ILA-H to an ILA-R to request mapping information for an IPv6 address.
- o Map information: Sent by an ILA-R to an ILA-N or ILA-H and provides mappings. A map information message can be sent in response to a map request, when mappings are pushed in pub/sub, or a mapping is being advertised by ILA redirect. The reason the mapping information was sent in included in a message.
- o Subscribe/unsubscribe: Sent by an ILA-N or ILA-H to an ILA-R. "Subscribe" requests mapping notifications for the listed identifiers. Notifications are sent when a mapping entry for an identifier changes. "Unsubscribe" requests that notifications for the listed identifiers stop.
- o Locator unreachable: sent by an ILA-R to an ILA-N or ILA-H to indicate that another ILA-N is no longer reachable so all cache entries using that ILA-N or ILA-H should be evicted.

6.3 Address assignment

Mobile nodes are assigned addresses that serve as identifiers. A node may be assigned singleton addresses or a network prefix. Privacy is an important consideration in address assignment.

6.3.1 Singleton address assignment

DHCPv6 or static address configuration can be used to assign singleton addresses to a node. These addresses have no topological component and are not meaningfully aggregable for routing, so an entry in the ILA mapping table would be created for each address. Nodes may be assigned thousand of addresses or even millions of IPv6 addresses. Given the large IPv6 address space there are few concerns about address depletion, however to the mapping system each address is represented in a identifier to locator mapping. Scaling this needs to be carefully considered. Sharding, replication, and caching on forwarding nodes are meant to provide scalability.

6.3.2 Network prefix assignment

A node may be assigned a /64 address via SLAAC as is common in many provider networks. In this scenario, the low order sixty-four bits contains IIDs arbitrarily assigned by a device for its own purposes; these bits cannot be used as an identifier in identifier/locator split.

To support /64 prefix assignment with ILA, the ILA identifier can be encoded in the the upper sixty-four bits of an address and the lower

sixty-four bits are ignored by ILA. Since only a subset of bits are available, a level of indirection can be used so that ILA transforms the upper sixty four bits to contain both a locator and an index into a locator (ILA-N) specific table. The entry in the table provides the original sixty-four bit prefix so that ILA to SIR address transformation can be done.

7 ILA in 5G networks

The section describes applying ILA for use in a 5G network. ILA is instantiated as a function in the 5G services architecture described in [3GPPTS].

7.1 Architecture

Figures 2 and 3 depict two architectural options for the use of ILA in a 5G architecture. ILA is logically a network function and ILA interfaces to the 5G control plane via service based interfaces. In this architecture, ILA replaces GTP use over the N9 interface. Identifier address to locator address transformations in the downlink from the data network are done by an ILA-R. Transformations for intra domain traffic can be done by an ILA-N close to the gNB or by an ILA-R in the case of a cache miss. Locator address to identifier address transformation happen at ILA-Ns. ILA could be supported on a gNB. In this case, an ILA-N would be co- resident at a gNB and ILA is used over N3 interface in lieu GTP-U.

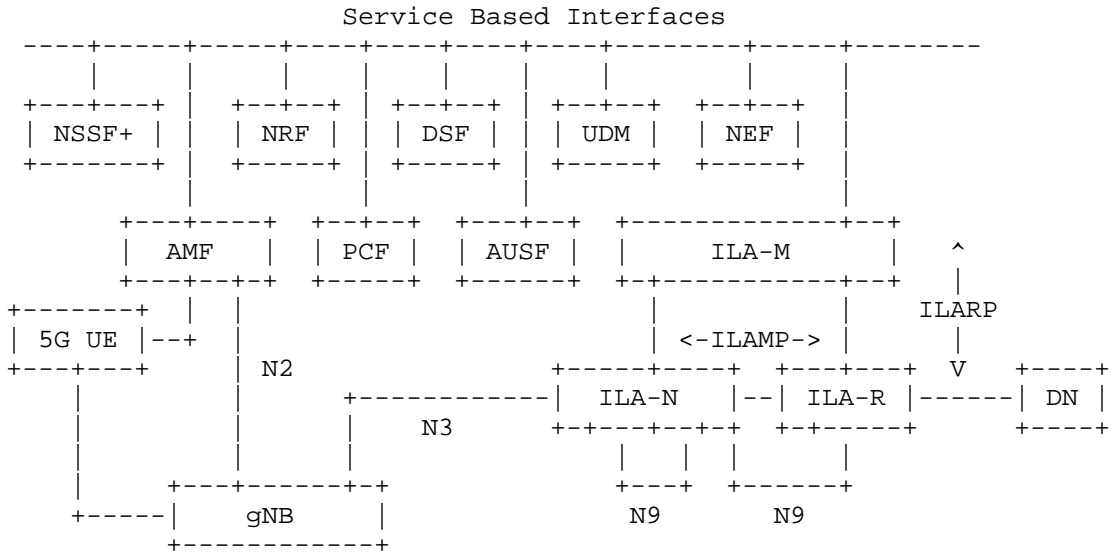


Figure 2: ILA in 5G architecture - Option 1

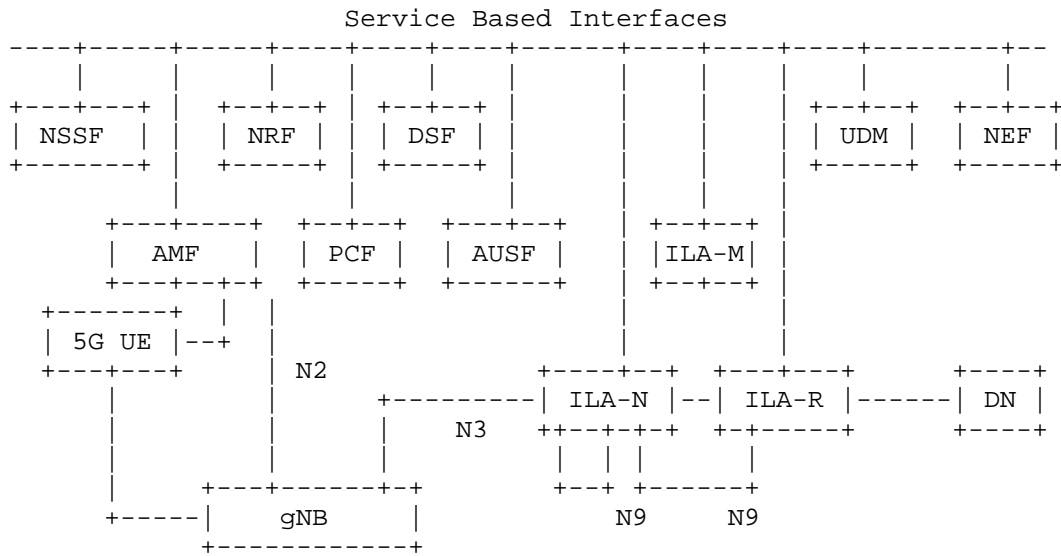


Figure 3: ILA in 5G architecture - Option 2

7.2 Protocol layering

Figure 3 illustrates the protocol layers of packets sent over various data plane interfaces in the downlink direction of data network to a mobile node. Note that this assumes the topology shown in Figure 2 where GTP-U is used over N3 and ILA is used on N9.

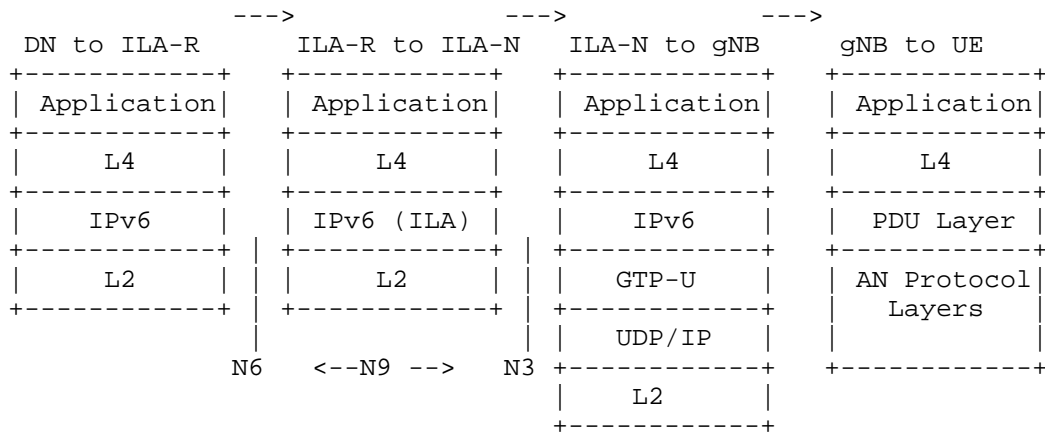


Figure 3: ILA and protocol layer in 5G

7.3 Control plane between ILA and network

in a slice. ILA can be deployed incrementally or in parts of the network. As demonstrated, the use of network slices can provide domain isolation for this.

Slice #2 supports ILA. Some number of ILA-Ns and ILA-Rs are deployed. ILA transformations are performed over the N9 interface. ILA-Rs would be deployed at the N6 interface to perform transformations on packets received from a data network. ILA-Ns will be deployed deeper in the network at one side of the N3 interface. ILA-Ns may be supplemented by ILA-Rs that are deployed in the network. ILA-M manages the ILA nodes and mapping database within the slice.

Slice #3 shows another slice that supports ILA. In this scenario, the slice is for Mobile Edge Computing. The slice contains ILA-Rs and ILA-Ns, and as illustrated, it may also contain ILA_Hs that run directly on edge computing servers. Note in this example, one ILA-M, and hence one ILA domain, is shared between slice #2 and slice #3. Alternatively, the two slices could each have their own ILA-M and define separate ILA domains.

8 Security considerations

A mobile public infrastructure has many considerations in security as well as privacy. Fundamentally, a system must protect against misdirection for the purposes of hijacking traffic, spoofing, revealing user identities, exposing accurate geo-location, and Denial of Service attacks on the infrastructure. Security must be considered for both the data and control planes.

8.1 Data plane security

The ILA data plane must protect against spoofing, inadvertent leakage of sensitive information, and Denial of Service attack.

Locator addresses must be contained within an ILA domain. ILA to SIR transformations MUST be performed before allowing a packet to egress an ILA domain.

Nodes outside of an ILA domain MUST NOT be permitted to send packets into the domain that have an ILA address in either the source or destination. A stateless firewall at the domain boundary can be used to drop such packets. Note that in the ILA protocol, ILA addresses are not used in source addresses.

Section 5.6 describes the handling of ICMP with ILA to avoid leaking locators outside the ILA domain.

When a cache is employed that is populated by events from an outside

party there is the possibility of Denial of Service attack. A conceptual attack on ILA-N would be for an attacker will flood its link with packets destined to random SIR addresses. The intent is to exhaust the cache memory so that legitimate traffic is blocked from using the cache and hence needs to take sub-optimal routing. The attack can also generate vast numbers of control messages to DOS the infrastructure.

It is recommended that ILA redirects, as opposed to query model or pub/sub, is used to mitigate attacks. The reasoning is:

- o On a cache miss the packet is forwarded and might encounter a router that sends a redirect. The packet itself implies a request for a mapping so no additional control message are needed.
- o An ILA router will send a redirect only if there is a mapping to the destination. It doesn't sent negative information. In particular, if the identifier space is reasonably sparse a random address attack will not be very effective.
- o A cache entry is created only when a valid redirect is received. This can be contrasted with a query mechanism that might create state for pending resolutions.
- o An inactivity timeout can used to evict cache entries. Given the incoming packet rate and a preferred inactivity timeout, a cache can be sized to absorb an attack.
- o An ILA router may apply its knowledge to rate limit, prioritize, and shape the use of redirects to manage caches. For instance, an ILA router might identify "hot nodes" in the network that receive a lot of traffic and provide the most benefit when cached in forwarding nodes.

8.2 Control plane security

A mapping system contains sensitive privacy information that could be used to make inferences about user's identity or their geo-location. This information needs to be protected.

Mapping protocols must be secured to prevent an attacker from injecting mapping entries to redirect traffic to their own devices. To this end, mapping protocols for ILA are intended to use TCP. The statefulness of TCP deters spoofing of messages and allows for privacy and identity verification in the form of X.509 certificates. The control protocol includes "secure" redirects that must be authenticated to originate from a legitimate ILA router.

Mapping protocols must also be resilient to DOS attack, especially in a scenario where a cache of mappings is being employed. Such a cache might be populated in response to the activities of a third party (for instance an application sending packets to different destinations). An attack on the cache whereby an attacker attempts to fill the cache with entries to random destinations must be mitigated. The recommendation of ILA is to use "secure redirects" as a scalable and secure means to populate a forwarding cache.

Write access to the ILA mapping database must be strictly controlled. In the ILA architecture only ILA-Ms write to the mapping database. Write access to the database should require strong credentials, validation of each operation, and encryption and authentication of operations being sent over the network.

Read access the ILA routing database should also be controlled. Devices should only access data on a "need to know" basis. For instance, ILA routers might need identifier to group mappings to perform forwarding, but they should not need to retrieve all the identifiers for a group. The latter information can be contained in the ILA-Ms.

8.3 Privacy in address assignment

A node may use multiple addresses to prevent inferences by third parties that break privacy. Properties of addresses to maintain strong privacy are:

- o They are composed of a global routing prefix and a suffix that is internal to an organization or provider. This is the same property for IP addresses [RFC3513].
- o The registry and organization of an address can be determined by the network prefix. This is true for any global address.
- o The organizational bits in the address should have minimal hierarchy to prevent inference. It might be reasonable to have an internal prefix that divides identifiers based on broad geographic regions, but detailed information such as accurate location, department in an enterprise, or device type should not be encoded in a globally visible address.
- o Given two addresses and no other information, the desired properties of correlating them are:
 - o It can be inferred if they belong the same organization and registry. This is true for any two global IP addresses.

- o It may be inferred that they belong to the same broad grouping, such as a geographic region, if the information is encoded in the organizational bits of the address (e.g. are in the same shard).
- o No other correlation can be established. For example, it cannot be inferred that the IP addresses address the same node, the addressed nodes reside in the same subnet, rack, or department, or that the nodes for the two addresses have any geographic proximity to one another.

Ostensibly, assigning a /64 prefix to a node is good for security. The end device can create its own random addresses in the low order sixty-four bits which mitigates address scanning attacks. However, the upper sixty four bits of the address become a static identifier for the node that potentially allows DOS on the device, as well as third party correlations on addresses that deduce that different flows are sourced from the same user.

[RFC4941] recommends rotating addresses to protect privacy. In the case of sixty-four bit address assignments this would entail that a new prefix for the device is periodically requested. There is no recommendation for the frequency of address change and there is no quantitative description of the effects of periodic address change.

For maximum privacy, a different address could be used for each connection. If this were done for every connection in the network, it would create network state for each connection (note that is sort of thing already exists with stateful NAT). Scaling the mapping system to accommodate this is challenging. One alternative to be investigated is use a reversible cryptographic hash to aggregate identifiers and reduce the number of mappings needed.

9 References

9.1 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>>.
- [ILA] Herbert, T., and Lapukhov, P., "Identifier Locator Addressing for IPv6" draft-herbert-intarea-ila-00
- [ILAMP] Herbert, T., "Identifier Locator Addressing Mapping Protocol" draft-herbert-ila-ilamp-00

9.2 Informative References

- [RFC3513] Hinden, R. and S. Deering, "Internet Protocol Version 6 (IPv6) Addressing Architecture", RFC 3513, DOI 10.17487/RFC3513, April 2003, <<https://www.rfc-editor.org/info/rfc3513>>.
- [RFC4941] Kolkman, O. and R. Gieben, "DNSSEC Operational Practices", RFC 4641, DOI 10.17487/RFC4641, September 2006, <<https://www.rfc-editor.org/info/rfc4641>>.
- [ILAGRPS] Herbert, T., "Identifier Groups in ILA", To be published
- [BGPILA] Lapukhov, P., "Use of BGP for dissemination of ILA mapping information" draft-lapukhov-bgp-ila-afi-02
- [3GPPTS] 3rd Generation Partnership Project (3GPP), "3GPP TS 23.502", <http://www.3gpp.org/DynaReport/23-series.htm>

Authors' Addresses

Tom Herbert
Quantonium
Santa Clara, CA
USA

Email: tom@quantonium.net

Kalyani Bogineni
Verizon
One Verizon Way, Basking Ridge, NJ 07920
USA

Email: kalyani.bogineni@verizon.com

INTERNET-DRAFT
Intended Status: Informational
Expires: July 2018

T. Herbert
Quantonium

January 22, 2018

Identifier Locator Addressing: Problem areas, Motivation, and Use Cases
draft-herbert-ila-motivation-00

Abstract

This document describes the problems, motivation, and use cases for Identifier-Locator Addressing.

Status of this Memo

This Internet-Draft is submitted to IETF in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

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."

The list of current Internet-Drafts can be accessed at
<http://www.ietf.org/lid-abstracts.html>

The list of Internet-Draft Shadow Directories can be accessed at
<http://www.ietf.org/shadow.html>

Copyright and License Notice

Copyright (c) 2018 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 (<http://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 Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1	Introduction	3
2	Problem areas	3
2.1	Identifier/locator split	3
2.2	Efficiency of network overlay techniques	3
2.3	Ramifications of network tunneling	4
2.4	Networking hardware compatibility	4
2.5	Mobility	5
2.6	Mapping systems	5
2.6.1	Nodes with complete set of mappings	5
2.6.2	Mapping caches	5
2.7	Privacy	6
2.7.1	Geo-location	6
2.7.2	Privacy in addresses	7
2.8	Address assignment	8
2.9	Scaling	9
2.10	Security	10
2.10.1	Mapping information	10
2.10.2	Mapping protocols	10
2.12	ICMP	10
2.13	Multicast	11
3	Motivation for ILA	11
3.1	Alternative network overlay technologies	11
3.1.1	ILNP	11
3.1.2	Encapsulation	12
3.1.3	Segment routing	13
3.1.4	Network Address Translation	13
3.1.5	Transport layer mechanisms	14
3.2	Benefits of ILA	14
3.3	Limitations and caveats of ILA	16
4	Use cases	16
4.1	Mobility networks	16
4.2	Datacenter virtualization	17
4.3	Network virtualization	17
5	References	17
5.1	Normative References	17
5.2	Informative References	17
	Author's Address	17

1 Introduction

Identifier-locator addressing (ILA) is a protocol based on identifier/locator split that provides network overlays without the use of encapsulation or extension headers. ILA operates at the network layer and is intended to be transparent to both applications and transport layer protocols.

This document highlights problem areas, motivation, and use cases of ILA. Problem areas include protocol efficiency, scalability, mobility, privacy, and security for network overlays and identifier/locator split. The motivation for ILA is provided in terms of how ILA addresses the problems and its advantages over alternative approaches. The use cases of ILA include mobility, datacenter virtualization, and network virtualization; some details and considerations for these use cases are provided.

2 Problem areas

This section highlights some of the problems faced for use of identifier/locator split and network overlays.

2.1 Identifier/locator split

Identifier/locator split is a technique that has long been discussed within IETF. This is the answer to the problem that IP addresses have traditionally been overloaded with two characteristics: they indicate both the identity of a node and its location. Identifier/locator split endeavors to separate these two notions. A node has identity and location, but they are separate elements in addressing. This disassociation of identity and location allows nodes to become "virtual" and mobile. This distinction has been made in network virtualization and mobility networks for some time, however demand for mobility and virtualization is continuously increasing so that in the future a majority of nodes might be virtualized and subject to identifier/locator split.

Deployment of identifier/locator split in existing networks can raise a number of issues. Since addresses no longer contain location, routing needs to change and routing tables need to scale to include many more destinations. Logging and tools also need to change to take into account that location and identity may be separate notions in an address.

2.2 Efficiency of network overlay techniques

With the emergence of applications such as AR, VR, machine learning, and road safety information, the demands for low latency, high

throughput, and highly reliable networking are only increasing. Low latency networking is no longer confined to the purview of specialized application centric networks, requirements for very low latency are being introduced into public access networking technologies such as 5G. As such, the efficiency and performance of network overlays becomes important.

Network overlays are a benefit since they facilitates node mobility and malleability in use of network resources. These benefits tend to be architectural in the network and not necessarily obvious to the users or applications in the network. For instance, network overlays facilitate mobility, however the cost of that capability to applications must be taken into a account. Mobility is likely a rare event, and there are many nodes that will never move during their lifetime. When a node is not moving, the cost incurred for having the capability to move should be near zero.

2.3 Ramifications of network tunneling

The ramifications and issues with tunneling in the network have been well documented and discussed in IETF.

If encapsulation is being used in the network to implement a tunnel then the packet size increases so exceeding the MTU is a concern; [RFC4459] discusses MTU and fragmentation considerations for network tunneling. [RFC2983] discusses the interactions of differentiated services with tunnels and procedures to translate diffserv for an inner packet to the outer one. [RFC6040] specifies handling of Explicit Congestion Notification in a tunnel. [RFC6935] and [RFC6936] discuss at length the requirements that must be meant for allowing a zero UDPv6 checksum to be used for tunnels.

The upshot is that defining and correctly implementing tunnels in the network is a nontrivial exercise.

2.4 Networking hardware compatibility

The effects of deploying a protocol with existing network hardware implementation must be considered. Generally, hardware implementations (switches, router, NICs, etc.) are optimized for certain protocols and protocol features. Most devices implement optimizations for the two most common transport protocols, namely TCP and UDP. These optimizations include features like ECMP, checksum offload, and segmentation offload. As one moves away from use of commonly supported protocols, the benefits of optimizations and even feasibility of protocols or protocol features dwindles. For example, IPv6 extension headers have had a checkered history of being properly supported by intermediate nodes, and hence are considered precarious

for use on the Internet [RFC7872].

Note that many new encapsulation protocols (GUE, GRE/UDP, LISP, Geneve, etc.) employ encapsulation in UDP. The use of UDP makes packets more palatable to network devices, albeit at the cost of UDP header overhead and additional processing overhead.

2.5 Mobility

For seamless mobility, a node retains its IP address and connections remain established across mobile events. If network mobility is handled in the network layer then moving should be transparent to the application with only the possibly that latency is increased for a few packets.

Mobility is present in different use cases whenever a node changes its point of attachment in the network. When this happens the location of the node and hence its locator changes. A key attribute of an identifier/locator solution is how the network converges when a change occurs. During the convergence period, latency is expected to be bounded and packet loss is expected to be minimized or avoided entirely.

2.6 Mapping systems

Identifier/locator split solutions employ a mapping system that provides identifier to locator mappings. Similar to IP routing, there may be both nodes that maintain a full list of mappings (analogous to core routers) and nodes that maintain a cache of mappings (analogous to nodes with a neighbor discovery cache).

2.6.1 Nodes with complete set of mappings

The complete set of mappings in a network might be sharded across some number of nodes. Each node maintains a complete list of mappings for their respective shard. Furthermore, each shard may be replicated on several nodes for redundancy and load balancing. All the nodes for a shard must synchronize information upon changes using a protocol amongst themselves. The time it takes for all nodes to converge when a change happens can correlate to perceived application latency.

2.6.2 Mapping caches

Anchorless mobility is a goal of identifier/locator split. For achieving low latency, direct routing is preferred. Forwarding nodes that contain a cache of mapping entries may be deployed at or near source hosts to optimize forwarding. These nodes maintain a cache of mapping entries used to directly forward packet to peers in the same

identifier/locator domain. The use of a cache avoids triangular routing through an intermediate device that has a complete list of mappings.

Mapping caches may be populated either by "push" or "pull" model. In the push model, nodes are informed of changes to the mappings for nodes they are interested in. A node may register to get notifications of changes from authoritative nodes in the mapping system. This is the pub/sub model. In the pull model, a node queries an authoritative node to resolve a mapping for an identifier it is interested in; typically this is done on demand when a packet is being forwarded to a destination address whose mapping is yet not in the cache.

Both the pull model and push model have their advantages and disadvantages. The push model (aka pub/sub) should result in faster and more accurate convergence, however may require more communications and be harder to scale than the pull model. The pull model may be more susceptible to Denial of Service attack.

Secure redirects are hybrid of the push and pull model. A cache entry is populated by an authoritative node that is forwarding a packet. This "redirect" eliminates triangular routing from the source to the destination. Redirects must be secure since to prevent destination hijacking.

2.7 Privacy

Identifier/locator split can benefit user privacy, particularly in what is exposed in IP addresses. The benefit is only viable if locators that imply geo-location and identities are not revealed to untrusted parties.

2.7.1 Geo-location

An effect of identifier/locator split is that location is no longer an inherent component of IP addresses. This is a benefit to user privacy as it can reduce the inference of geo-location of a user based on IP addresses. However, strong privacy implies that locators, which could very well reveal geo-location, are only visible to trusted entities.

Conceivably, an identifier-locator protocol could be run at the end user devices in a public network (e.g. UEs in a mobile network). This section provides a privacy argument against that.

The major problem with running an identifier/locator protocol on end user devices is that the devices are not controlled by the network

infrastructure. User devices on a public network, such as Android devices, can easily be hacked to allow root access to the device. Once a user has root access they can install any program they wish on the device including those that could disable or circumvent security or accounting related to identifier/locator split protocol.

If root access can be gained on an end user device, this leads to the stalker problem which would be a very easy means to track individuals. This exploit is described below:

- * Suppose that a user device participates in an identifier/locator split protocol such that they cache locators and use locators to directly send to peer devices.
- * A hacker might tap all packets sent or received on the network interfaces which makes locators visible to them.
- * In order to be able to tap packets, a user needs root access to the device. There are instructions on the web to root an Android device. Similarly, jailbreaking can be done to circumvent restrictions on an iPhone to gain the equivalent of root access.
- * Once root access has been obtained, packets can be tapped using tcpdump or similar packet sniffer applications.
- * With the tap running and packet addresses being captured, the hacker just needs to drive around sending traffic between two devices in their car. They can observe the locators that are assigned to the their device, and from this can create a geo map of locators.
- * Given that one hacker can do this, then thousands will do it and web sites will spring up that provide locator geo maps. Efforts to obfuscate or rotate identifiers does not help much here. Obfuscation complicates routing in the network such that more transformations need to happen there. Locator rotation is defeated if there are enough devices to keep the maps up to date in a mashup.
- * The net effect is that this enables a stalker attack. An individual simply initiates a communication with their target. For instance, this could be a chat or phone call. If in doing this the locators for the device belonging to the target are made visible to the hacker, then the physical location of the target can be deduced using the locator geo maps described above.

2.7.2 Privacy in addresses

A node may use multiple addresses to prevent inferences by third parties that break privacy. Properties of addresses to maintain strong privacy are:

- * They are composed of a global routing prefix and a suffix that is internal to an organization or provider. This is the same property for IP addresses [RFC3513].
- * The registry and organization of an address can be determined by the network prefix. This is true for any global address.
- * The organizational bits in the address should have minimal hierarchy to prevent inference. It might be reasonable to have an internal prefix that divides identifiers based on broad geographic regions, but detailed information such as accurate location, department in an enterprise, or device type should not be encoded in a globally visible address.
- * Given two addresses and no other information, the desired properties of correlating them are:
 - * It can be inferred if they belong the same organization and registry. This is true for any two global IP addresses.
 - * It may be inferred that they belong to the same broad grouping, such as a geographic region, if the information is encoded in the organizational bits of the address (e.g. are in the same shard).
 - * No other correlation can be established. For example, it cannot be inferred that the IP addresses address the same node, the addressed nodes reside in the same subnet, rack, or department, or that the nodes for the two addresses have any geographic proximity to one another.

2.8 Address assignment

In an identifier/locator split protocol, end hosts are assigned addresses that serve as identifiers. A device may be assigned a network prefix or singleton addresses.

A end host may be assigned a /64 address via SLAAC as is common in many provider networks. In this scenario, the low order sixty-four bits contains IIDs arbitrarily assigned by devices for its purposes; so these bits cannot be used as an identifier in identifier/locator split. Effectively, the upper sixty-four bits is the identifier of the node.

Ostensibly, assigning a /64 prefix to a node is good for security. The end device can create its own random addresses in the low order sixty-four bits which mitigates address scanning attacks. However, the upper sixty four bits of the address become a static identifier for the device which potentially allows DOS on the device as well as correlating different addresses in the Internet as being sourced from the same device.

[RFC4941] recommends rotating addresses to protect privacy. In the case of sixty-four bit address assignments this would entail that a new prefix for the device is periodically requested. There is no recommendation for the frequency of address change and there is no quantitative description of the effects of periodic address change.

The following exploit is proposed to defeat the privacy goal of periodic address rotation:

- * An attacker creates an "always connected" app that provides some seemingly benign service so that users download the app.
- * The app includes some sort of persistent identity. For instance, this could be an account login.
- * The backend server for the app logs the user identity and IP address each time a user connects.
- * When an address change happens, existing connections on the user device are disconnected. The app will receive a notification and immediately attempt to reconnect using the new source address.
- * The backend server will see the new connection and log the new IP address as being used by the user. Thus, the server has a real-time record of users and the IP addresses they are using.
- * The attacker gains access to packet traces taken at some point in the Internet. The addresses in the captured packets can be time correlated with the server database to deduce the identity of the source of packets for communications completely unrelated to the app.

This exploit would defeat address rotation with any frequency except the for case that that a different source address is used for each flow.

2.9 Scaling

Since identity is no longer associated with location, each node becomes separately routable in the network. In identifier/locator

split, a table that maps identifiers to locators is maintained. Each destination effectively becomes a host route, and hierarchical routing is generally not usable. For instance, a VM may have a virtual address and might be located anywhere in a network. The mapping table would contain a mapping of the virtual address of the VM to the physical address of the server where the VM is running.

The number of virtualized or mobile nodes in a network is expected to grow into the billions. This need for scaling is similar in both mobile networks as well as datacenter and multi-tenant virtual networks. In mobile networks, the explosion of IoT devices drives scaling growth. In the datacenter it is the desire to use fine grained addresses for tasks or more generally addressable virtual objects.

2.10 Security

2.10.1 Mapping information

An identifier/locator solution will contain sensitive information that includes identity and location of nodes. In the case that there is a one-to-one correspondence between a network node and user, for instance the node is a smart phone owned by an individual, this information is Personally Identifiable Information (PII). A mapping system needs to ensure security of this information.

2.10.2 Mapping protocols

Mapping protocols have a couple of security ramifications.

A mapping protocol must be authenticated in order to prevent spoofing of messages. In particular, an attacker cannot be able to hijack a mapping entry to redirect packets to their own node.

A mapping protocol must also be resilient to DOS attack, especially in a scenario where a cache of mappings is being employed. Such a cache might be populated in response to the activities of a third party (for instance an application sending packets to different destinations). An attack on the cache whereby an attacker attempts to fill the cache with entries to random destinations must be mitigated.

2.12 ICMP

ICMP presents problems for network overlays as well as identifier/locator split. Specifically, the problem is how to return ICMP errors to the sender that were caused as a result of using a network overlay. ICMP errors that are returned to the source may

require translation of address in the ICMP data or other modifications. There may also be security ramifications with ICMP, for instance filtering ICMP may be necessary to prevent locator information from leaking out of a network.

2.13 Multicast

Multicast is problematic in identifier/locator split since the routing depends on the source address of a packet. If using the network layer multicast, the source address must be a locator not an identifier.

3 Motivation for ILA

3.1 Alternative network overlay technologies

A number of solutions for network overlays have been defined or proposed in IETF. This section considers layer 3 network overlay solutions, and a few related layer 4 solutions for comparison. An overview of each is provided along with a description of how they deal with the problems enumerated in section 2 and where they are deficient.

3.1.1 ILNP

Identifier-Locator Network Protocol (ILNP) is similar to ILA and in fact some of the concepts of ILA were adapted from ILNP.

ILNP explicitly replaces the use of IP Addresses with two distinct name spaces, each having distinct and different semantics:

- a) Identifier: a non-topological name for uniquely identifying a node.
- b) Locator: a topologically bound name for an IP subnetwork.

Characteristics of ILNP are:

- * ILNP changes the meaning of IP addresses.
- * ILNP requires changes to transport layer protocols. Transport layer endpoints are no longer IP addresses they are identifier values.
- * The pseudo header for TCP and UDP checksum changes. This might break intermediate nodes that perform checksum calculation such as NICs that provides checksum offload.

- * ILNP is an end to end overlay mechanism. There is no prescribed method to use ILNP in intermediate nodes.
- * ILNP defines a Nonce in Destination Options extension header.
- * ILNP requires applications to use fully qualified domain names. Applications that use IP addresses presumably need to change.

3.1.2 Encapsulation

Various encapsulation techniques are used to achieve layer 3 network overlays. These includes IPIP, LISP, GRE, VXLAN, GUE, GTP-U, Geneve, etc. These encapsulation protocols provide the means to create overlays over IP networks via IP over IP encapsulation. They differ in format and extensibility. For instance, IPIP is the simplest method that just encapsulates one IP packet in another. GUE is a UDP based encapsulation that is both generic and extensible.

Characteristics of encapsulation are:

- * While encapsulation has proven functional and useful, it incurs significant on-the-wire overhead, require substantial processing, and may be incompatible with transport layer specific network optimizations for TCP and UDP
- * Outer IP header overhead. Adoption of IPv6 exacerbates the overhead of encapsulation. Where simple IPv4 over IPv4 encapsulation has an overhead of twenty bytes, IPv6 or IPv4 over IPv6 incurs overhead of forty bytes.
- * Possible additional header overhead if UDP is used or there is an encapsulation header
- * Can be used at intermediate nodes for tunneling, so all the issues involving tunneling must be addressed.
- * Compatibility with hardware is an issue. UDP based encapsulation overcomes some of the issues, but in itself creates new ones.
- * Checksum handling must be considered in various contexts. Encapsulation may break checksum offload feature commonly implemented in NICs. Some network devices are incapable of computing checksums, so if UDPv6 is used the checksum is often set to zero. Some protocols allow a non-zero UDP checksum to be ignored during reception in violation of [RFC1122].
- * Issues around tunneling within the network have to be addressed (described in section 2.3). These include dealing with MTU, IPv6

checksum, traceroute, ECN, and how to translate diffserv from an inner header to an outer header.

- * Encapsulation can be used in the network or at end hosts and doesn't require any changes to transport layer implementation.

3.1.3 Segment routing

Segment routing (SR) has been proposed as a method to provide identifier/locator split for mobile networks. SR leverages the source routing paradigm. A node steers a packet through an ordered list of instructions, called segments. A segment can represent any instruction, topological or service-based. A segment can have a semantic local to an SR node or global within an SR domain

Characteristics of segment routing are:

- * Requires use of extension headers, specifically a routing header.
- * [RFC820] prohibits extension header insertion at intermediate nodes. Encapsulation is required at ingress intermediate node to use segment routing.
- * The segment header itself be significant overhead. A segment routing header with just a single address would be twenty four bytes of overhead.
- * Transport layer checksum is not kept correct when destination address is changed. This could break checksum offload.
- * Transport layer checksum does not protect the segment routing header, so additional overhead is needed to detect corruption of the SR header.
- * Extension headers are not transparent to intermediate nodes and this may cause incompatibility with network hardware implementation resulting in loss of optimizations or relegation to slow path processing.

3.1.4 Network Address Translation

ILA is similar to NAT (address translation not port translation) in that it operates by rewriting the destination address of packet en route. However, the transformation by ILA is always undone before the packet is delivered to its ultimate destination.

Characteristics of NAT:

- * No additional header overhead. Checksum neutral mapping might be used to maintain correct transport layer checksum.
- * Not useful as an overlay mechanism since NAT translation is not undone before reception at a receiver

3.1.5 Transport layer mechanisms

There are a number of techniques used in the transport layer or applications to handle mobility. Strictly speaking, these are not network overlay techniques, however they can be used to provide similar effects in mobility.

The simplest way to deal with an address change is just to require an application to reconnect when a connection is disconnected. This is not transparent to an application, it must have a method to checkpoint progress on the connection and implement the reconnect logic (this could be handled in a library). The latency to detect that a connection is dead, reconnect, and then recover to a checkpoint is likely much greater than that of a transparent network layer solution.

Alternatively, a transport protocol may employ subflows to construct a logical flow. This is the technique used by MPTCP and QUIC. These techniques are transport layer specific, tend to be driven by one sided, and require network layer information.

Proxies can also provide network overlay semantics. However, they require statefulness in the network that creates single point of failure and a potential bottleneck.

3.2 Benefits of ILA

This section enumerates the benefits of ILA and highlights how the problems described in section 3 are addressed.

- * ILA has zero on-the-wire overhead.
- * Processing for ILA is efficient. A basic ILA transformation is done by reading the destination address in a packet, performing a fixed length lookup, and writing the destination address with found locator.
- * ILA does not employ tunneling so considerations for network tunneling are not a concern.

- * The ILA domain is effectively a virtual link layer or an underlay network for the traffic being carried between hosts outside of the ILA domain. As long as the ILA domain is perfectly transparent to the overlay network and its hosts, then what ever happens within the ILA domain doesn't matter, similar to how link layer compression, as long as fully and perfectly reversed, also doesn't matter.
- * ILA maintains a correct transport layer checksum via checksum neutral mapping.
- * ILA can be deployed either in the network or on end hosts. When deployed at end hosts, certain optimizations are available since ILA is integrated into the host stack
- * ILA is implemented at network layer. It requires no changes to either applications or transport layer implementations.
- * ILA is transparent to intermediate nodes so that it is compatible with existing networking hardware and protocol optimizations. A TCP/IP packet is still a TCP/IP packet after being transformed by ILA.
- * ILA enables singleton address assignment for privacy. It also supports /64 address assignment.
- * It is recommended that ILA be contained within an ILA domain that is one network under administrative control. Locators are not shared with parties outside of the domain.
- * ILA espouses the use of secure redirects as the primary means to populate a mapping cache. Push and pull models can be used, however secure redirects should be effective in mitigating DOS attacks and scalable.
- * ILA allows alternative address representations for identifier/locator split other than the canonical 64/64 split. Non-local identifiers are defined as a method to use identifiers to map to 128 bit IP addresses that might not be local to a network.
- * ILA defines optional addressing schemas for IPv4 to IPv6 translation, network virtualization with an embedded virtual networking identifier, and encoding of IP multicast addresses.
- * ILA defines identifier groups as a convenient way to group identifiers together that have common characteristics. Identifier groups should reduce the number of operations needed

on the mapping system.

- * A reference datapath implementation is supported in stock Linux since version 4.15. A userspace control path implementation will be open sourced.

3.3 Limitations and caveats of ILA

This section describes limitations and caveats of ILA.

- * While ILA has much less overhead than encapsulation or extension headers, this does limit the amount of information that can be expressed. ILA is not extensible like some encapsulations so there is no means to associate ancillary information with ILA.
- * /64 address assignment is feasible in ILA, however requires a level of indirection in addressing.
- * ILA operates by transforming destination IP addresses in packets. Source addresses are not transformed. This works very well for unicast traffic, but creates some complexity for multicast in using the network layer multicast with ILA.
- * If the network generates an ICMP error for a packet whose destination contains a transformed address with a locator, the embedded packet in ICMP data contains a destination address with a locator. Before delivery to the original source host this address should be converted back to the original destination address.
- * Firewalls should filter addresses in packets before ILA translation. The typical scenario is that when a packet is forwarded to a network ingress point, the firewall inspects the packet before ILA is applied. An firewall internal to the network may see ILA addresses as destinations; this should be taken into account.
- * Logging and tools need to be adapted since they may be operating on ILA addresses. Logged addresses can be mapped to standard identifier representation either by a fixed mapping or by reverse mapping the address by a lookup in the mapping table. The latter would be needed in the case of non-local identifier addresses.

4 Use cases

4.1 Mobility networks

4.2 Datacenter virtualization

4.3 Network virtualization

5 References

5.1 Normative References

5.2 Informative References

Author's Address

Tom Herbert
Quantonium
Santa Clara, CA
USA

Email: tom@quantonium.net

INTERNET-DRAFT
Intended Status: Standard
Expires: September 5, 2018

Tom Herbert
Quantonium
Petr Lapukhov
Facebook

March 5, 2018

Identifier-locator addressing for IPv6
draft-herbert-intarea-ila-01

Abstract

This specification describes identifier-locator addressing (ILA) for IPv6. Identifier-locator addressing differentiates between location and identity of a network node. Part of an address expresses the immutable identity of the node, and another part indicates the location of the node which can be dynamic. Identifier-locator addressing can be used to efficiently implement overlay networks for network virtualization as well as solutions for use cases in mobility.

Status of this Memo

This Internet-Draft is submitted to IETF in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

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."

The list of current Internet-Drafts can be accessed at
<http://www.ietf.org/lid-abstracts.html>

The list of Internet-Draft Shadow Directories can be accessed at
<http://www.ietf.org/shadow.html>

Copyright and License Notice

Copyright (c) 2018 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 (<http://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 Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1	Introduction	4
1.1	Terminology	4
1.2	Use cases	6
1.3	Scope	6
2	Architecture overview	7
2.1	Addressing	7
2.2	Network topology	8
2.3	Transformations and mappings	8
2.4	ILA routing	9
2.5	ILA domains	10
2.6	ILA control plane	10
3	Address formats	10
3.1	ILA address format	10
3.2	Locators	11
3.3	Identifiers	11
3.4	Standard identifier representation addresses	12
4	Optional identifier formats	13
4.1	Checksum neutral mapping	13
4.2	Identifier types	13
4.2.1	Interface identifiers	15
4.2.2	Locally unique identifiers	15
4.2.3	Virtual networking identifiers for IPv4	15
4.2.4	Virtual networking identifiers for IPv6 unicast	16
4.2.5	Virtual networking identifiers for IPv6 multicast	17
4.2.6	Non-local address identifiers	18
4.3	SIR addresses with formatted identifiers	19
4.3.1	SIR for locally unique identifiers	20
4.3.2	SIR for virtual addresses	20
4.3.2	SIR for non-local address identifiers	20
5	Operation	20
5.1	Identifier to locator mapping	20
5.2	Address transformations	21

5.2.1	SIR to ILA address transformation	21
5.2.2	ILA to SIR address transformation	21
5.3	Virtual networking operation	22
5.3.1	Crossing virtual networks	22
5.3.2	IPv4/IPv6 protocol translation	22
5.4	Transport layer checksums	23
5.4.1	Checksum-neutral mapping	23
5.4.2	Sending an unmodified checksum	25
5.5	Non-local address mapping	25
5.6	Address assignment	26
5.6.1	Singleton address assignment	26
5.6.2	Network prefix assignment	26
5.6.3	Strong privacy addresses	27
5.7	Address selection	27
5.8	Duplicate identifier detection	27
5.9	ICMP error handling	28
5.9.1	Handling ICMP errors by ILA capable hosts	28
5.9.2	Handling ICMP errors by non-ILA capable hosts	28
5.10	Multicast	29
6	Motivation for ILA	29
6.1	Use cases	29
6.1.1	Multi-tenant virtualization	29
6.1.2	Datacenter virtualization	30
6.1.3	Mobile networks	30
6.2	Alternative methods	31
6.2.1	ILNP	31
6.2.2	Flow label as virtual network identifier	31
6.2.3	Extension headers	32
6.2.4	Encapsulation techniques	32
7	Security Considerations	32
8	IANA Considerations	33
9	References	34
9.1	Normative References	34
9.2	Informative References	34
10	Acknowledgments	35
	Appendix A: Communication scenarios	36
A.1	Terminology for scenario descriptions	36
A.2	Identifier objects	37
A.3	Reference network for scenarios	37
A.4	Scenario 1: Object to task	38
A.5	Scenario 2: Object to Internet	38
A.6	Scenario 3: Internet to object	38
A.7	Scenario 4: Tenant system to service	39
A.8	Scenario 5: Object to tenant system	39
A.9	Scenario 6: Tenant system to Internet	40
A.10	Scenario 7: Internet to tenant system	40
A.11	Scenario 8: IPv4 tenant system to object	40
A.12	Tenant to tenant system in the same virtual network	41

A.12.1 Scenario 9: TS to TS in the same VN using IPV6	41
A.12.2 Scenario 10: TS to TS in same VN using IPv4	41
A.13 Tenant system to tenant system in different virtual networks	41
A.13.1 Scenario 11: TS to TS in different VNs using IPV6	41
A.13.2 Scenario 12: TS to TS in different VNs using IPv4	42
A.13.3 Scenario 13: IPv4 TS to IPv6 TS in different VNs	42
A.14 Scenario 14: Non-local address to tenant system	42
Appendix B: unique identifier generation	43
B.1 Globally unique identifiers method	43
B.2 Universally Unique Identifiers method	44
Appendix C: Datacenter task virtualization	44
C.1 Address per task	44
C.2 Job scheduling	45
C.3 Task migration	45
C.3.1 Address migration	46
C.3.2 Connection migration	46
Appendix D: Mobility in wireless networks	47

1 Introduction

This specification describes the address formats, protocol operation, and communication scenarios of identifier-locator addressing (ILA). In identifier-locator addressing, an IPv6 address is split into a locator and an identifier component. The locator indicates the topological location in the network for a node, and the identifier indicates the node's identity which refers to the logical or virtual node in communications. Locators are routable within a network, but identifiers typically are not. An application addresses a peer destination by identifier. Identifiers are mapped to locators for transit in the network. The on-the-wire address is composed of a locator and an identifier: the locator is sufficient to route the packet to a physical host, and the identifier allows the receiving host to translate and forward the packet to the application.

Some of the concepts for ILA are adapted from Identifier-Locator Network Protocol (ILNP) ([RFC6740], [RFC6741]) which defines a protocol and operations model for identifier-locator addressing in IPv6.

Section 6 provides a motivation for ILA and comparison of ILA with alternative methods that achieve similar functionality.

1.1 Terminology

ILA	Identifier-locator addressing.
ILA host	An end host that is capable of performing ILA

translations on transmit or receive.

- ILA router A network node that performs ILA translation and forwarding of translated packets.
- ILA node A network node capable of performing ILA translations. This can be an ILA router or ILA host.
- Locator A network prefix that routes to a physical host. Locators provide the topological location of an addressed node. ILA locators are typically sixty-four bit prefixes, however other prefix sizes can be used.
- Locator address
An IPv6 address than contains a locator.
- Identifier A number that identifies an addressable node in the network independent of its location. ILA identifiers are typically sixty-four bit values, however other sized values may be used.
- Identifier address
An IPv6 address that contains an identifier but not a locator. Identifier addresses are visible to applications and provide a means to address nodes independent of their location.
- ILA address
An IPv6 address composed of a locator and an identifier. In the canonical format the locator occupies the upper sixty-four bits of an address and the identifier is in the lower sixty-four bits.
- ILA domain A unique identifier namespace. This may be indicated by a SIR prefix where each SIR prefix maps to an ILA domain.
- ILA transformation
The process of transforming an identifier address to a locator address or vice versa.
- SIR Standard identifier representation.
- SIR prefix A network prefix used to identify a SIR address. In the canonical format SIR prefixes are sixty-four bits.
- SIR address
An identifier address composed of a SIR prefix

(typically upper sixty-four bits) and an identifier (typically lower sixty-four bits).

Virtual address

An IPv6 or IPv4 address that resides in the address space of a virtual network. Such addresses may be translated to identifier addresses as an external representation of the address outside of the virtual network, or they may be translated to locator addresses for transit over an underlay network.

Topological address

An address that refers to a non-virtual node in a network topology. These address physical hosts in a network.

Checksum-neutral mapping

A method to preserve a correct transport layer checksum when performing ILA transformation. When the upper bits of an address are overwritten in an ILA transformation, a modification can be made to the low order bits of the identifier to offset the checksum difference.

1.2 Use cases

ILA use cases include datacenter virtualization, network virtualization, and mobility in cellular and other mobile networks. Section 6 provides details on these use cases. ILA operates at the network layer so it works with any transport layer protocol and can be used at intermediate devices or end nodes. An ILA implementation may include optimizations depending on where in the network it runs.

1.3 Scope

Architecturally, ILA is a protocol to implement transparent network overlays without encapsulation. It is also an identifier/locator split protocol where location of a node is decoupled from its identity. ILA works by transforming addresses between identifier and locator addresses. ILA does address "transformation" as opposed to "translation" since address modifications are always undone before delivery to a destination node.

With identifier-locator addressing, network virtualization and addressing for mobility can be implemented in an IPv6 network without any additional encapsulation headers. Packets sent with identifier-locator addresses look like plain unencapsulated packets (e.g. TCP/IP packets). This method is transparent to the network, so protocol specific mechanisms in network hardware work seamlessly. These

mechanisms include hash calculation for ECMP, NIC large segment offload, checksum offload, etc.

ILA includes both a data plane and control plane. The data plane defines the address structure and mechanisms for transforming application visible identifier addresses to locator addresses. The control plane's primary focus is a mapping system that includes a database of identifier to locator mappings. This mapping database drives ILA transformations. Control plane protocols disseminate identifier to locator mappings amongst ILA nodes.

This specification is mostly concerned with the data plane for ILA. The control plane is specified elsewhere.

2 Architecture overview

This section describes the architectural aspects of ILA.

2.1 Addressing

ILA performs transformations on IPv6 addresses. There are two types of addresses introduced for ILA: locator addresses and identifier addresses.

Locator addresses are IPv6 addresses that are composed of a locator (typically upper sixty-four bits) and an identifier (typically low order sixty-four bits). The identifier serves as the logical address of a node, and the locator indicates the location of a node on the network.

Identifier addresses are IPv6 addresses that contain an identifier but not a locator. Identifier addresses are visible to applications and provide a means to address nodes independent of their location.

A SIR address (Standard Identifier Representation) is an identifier address that contains an identifier and an application visible SIR prefix. SIR addresses are visible to the application and can be used as connection endpoints. When a packet is sent to a SIR address, an ILA router or host overwrites the SIR prefix with a locator corresponding to the identifier. When a peer receives the packet, the locator is overwritten with the original SIR prefix before delivery to the application. In this manner applications only see SIR addresses, they do not have visibility into ILA addresses.

ILA transformations can transform addresses from one type to another. In network virtualization, virtual addresses can be transformed into locator or identifier addresses, and conversely locator and identifier addresses can be translated to virtual addresses.

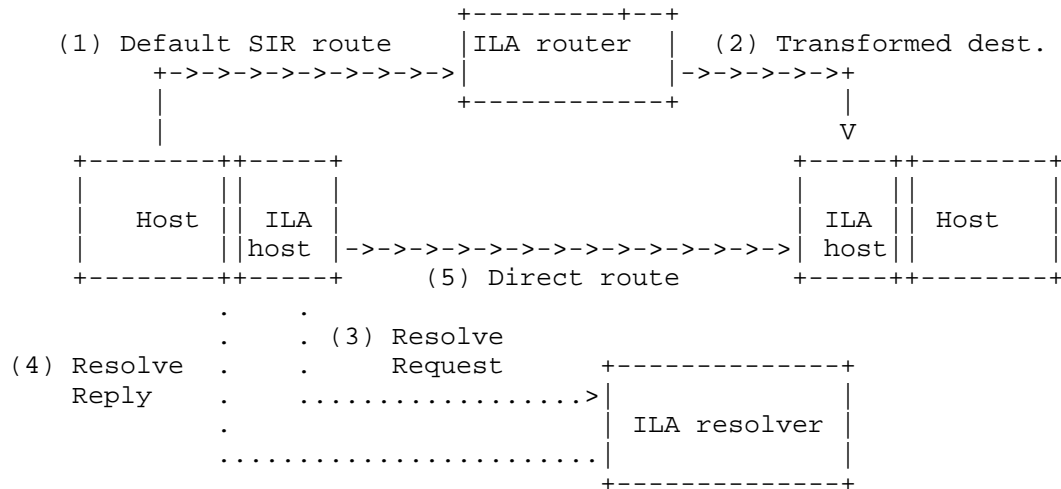
transport to the proper destination. In the canonical ILA addressing format, transformation occurs in the upper sixty-four bits of an address, the low order sixty-four bits contains an identifier that is immutable and is not used to route a packet. The identifier/locator split in addresses may have alternate arrangements for different use cases. For instance, transformations on non-local identifier address (Section 4.2.6) are performed across the full 128 bit address.

Each ILA node maintains a mapping table. This table maps identifiers to locators. The mappings are dynamic as nodes with identifiers can be created, destroyed, or move in the network. Mappings are propagated amongst ILA routers or hosts in a network using mapping propagation protocols (mapping propagation protocols will be described in other specifications).

Identifiers are not statically bound to a host on the network, and in fact their binding (or location) may change. This is the basis for network virtualization and device mobility. An identifier is mapped to a locator at any given time, and a set of identifier to locator mappings is propagated throughout a network to allow communications. The mappings are kept synchronized so that if an identifier migrates to a new location, its identifier to locator mapping is updated.

2.4 ILA routing

ILA is intended to be sufficiently lightweight so that all the hosts in a network could potentially send and receive ILA addressed packets. In order to scale this model and allow for hosts that do not participate in ILA, a routing topology may be applied. A simple routing topology is illustrated below.



An ILA router can be addressed by an "anycast" SIR prefix so that it receives packets sent on the network with SIR addresses. When an ILA router receives a SIR addressed packet (step (1) in the diagram) it will perform the ILA transformation and send the ILA addressed packet to the destination ILA node (step (2)).

If a sending host is ILA capable the triangular routing can be eliminated by performing an ILA resolution protocol. This entails a host sending an ILA resolve request that specifies the SIR address to resolve (step (3) in the figure). An ILA resolver can respond to a resolve request with the identifier to locator mapping (step (4)). Subsequently, the ILA host can perform ILA transformation and send directly to the destination specified in the locator (step (5) in the figure). The ILA resolution protocol will be specified in a companion document.

In this model an ILA host maintains a cache of identifier mappings for identifiers that it is currently communicating with. ILA routers are expected to maintain a complete list of identifier to locator mappings within the ILA domains that they service.

2.5 ILA domains

An ILA domain defines a namespace for identifiers. Identifiers must be unique within an ILA domain. Each SIR prefix maps to one ILA domain so that the combination of a SIR prefix and an identifier (a SIR address) uniquely identifies a node. More than one SIR prefix may be associated a domain where each SIR prefix combined with the same identifier refers to the same node.

Locators **MUST** map to only one ILA domain in order to ensure that transformation from a locator to SIR prefix is unambiguous.

2.6 ILA control plane

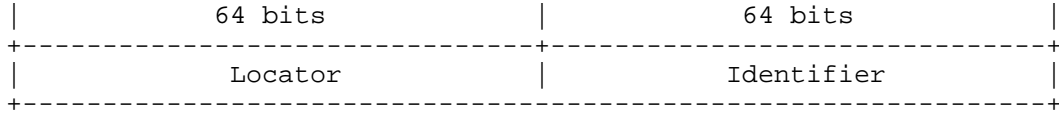
ILA routers and ILA hosts require a control plane that propagates the tables that map identifier addresses to locator address (or just identifier to locator mappings). There are several possible methods for control planes that have been proposed including synchronized configuration, BGP, DNS, and NoSQL databases. Defining a specific control plane for ILA is out of scope of this document.

3 Address formats

3.1 ILA address format

In the canonical format, an ILA address is composed of a locator and an identifier where each occupies sixty-four bits (similar to the

encoding in ILNP [RFC6741]).

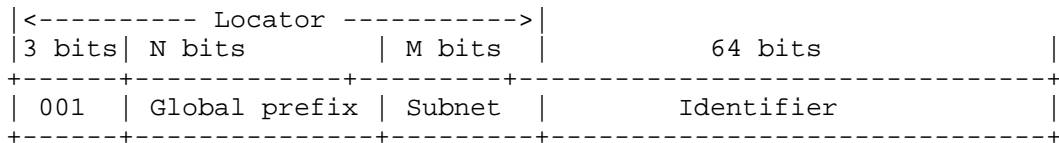


Note that there is no technical reason why identifiers and locators must be sixty-four bits. Different sizes could be used. The split is somewhat arbitrary, however it does simplify the description and implementation. For instance, sixty-four bits is the size of a "long long" native data type in several computer architectures. It is conceivable that a different arrangement could be used for some ILA domain. However, for the purposes of this document we assume that the 64/64 split is the canonical format.

3.2 Locators

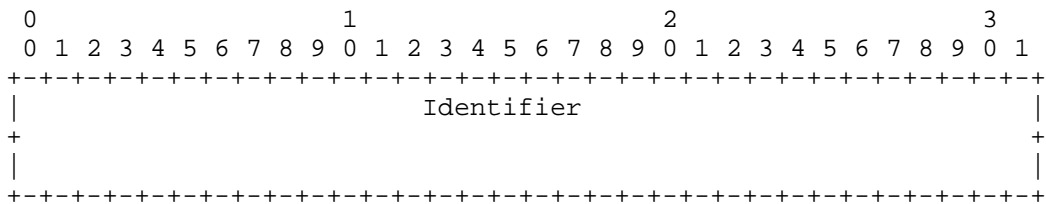
Locators are routable network address prefixes that create topological addresses for physical hosts within the network. They SHOULD be assigned from a global address block [RFC3587].

The format of an ILA address with a global unicast locator is:



3.3 Identifiers

Identifiers uniquely identify logical nodes in an ILA domain. The format of an ILA identifier is:



Identifiers are specified to be sixty-four bit values that are unstructured. A structure and format for identifiers MAY be defined for a domain; for instance the operator of an ILA domain may define the use of prefixes for its identifiers in order to facilitate hierarchies of its identifiers. Section 4 defines optional ILA

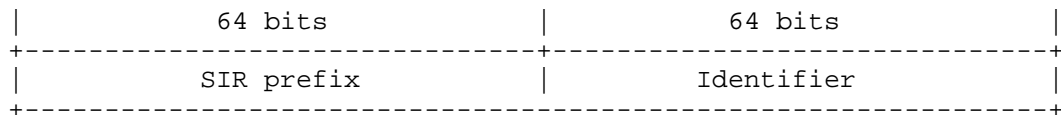
formats that an ILA domain might impose locally that allow different types of identifiers as well as an indication of checksum neutral mapping.

3.4 Standard identifier representation addresses

An identifier identifies objects or nodes in a network. For instance, an identifier may refer to a specific host, virtual machine, or tenant system. When a host initiates a connection or sends a packet, it uses an identifier address to indicate the peer endpoint of the communication. The endpoints of an established connection context are also referenced by identifiers (encoded in identifier addresses). It is only when the packet is actually being sent over a network that the locator for the identifier needs to be resolved.

In order to maintain compatibility with existing networking stacks and applications, identifiers are encoded in IPv6 addresses using a standard identifier representation (SIR) address. A SIR address is a combination of a prefix which occupies what would be the locator portion of an ILA address, and the identifier in its usual location.

The format of a SIR address is:



A SIR prefix SHOULD be a globally routable prefix per [RFC3587]. A globally routable SIR prefix facilitates connectivity between hosts on the Internet and ILA nodes. An ILA router between a site's network and the Internet can translate between SIR prefix and locator for an identifier. A network may have multiple SIR prefixes where each prefix defines a unique identifier space.

Locators MUST only be associated with one SIR prefix. This ensures that if a transformation from a SIR address to an ILA address is performed when sending a packet, the reverse transformation at the receiver yields the same SIR address that was seen at the transmitter. This also ensures that a reverse checksum-neutral mapping can be performed at a receiver to restore the addresses that were included in a pseudo header for setting a transport checksum.

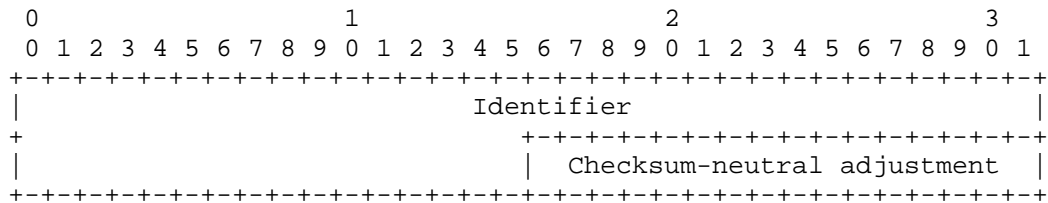
An identifier address can be used as the externally visible address for a node. This can be used throughout the network, returned in DNS AAAA records [RFC3363], used in logging, etc. An application can use an identifier address without knowledge that it encodes an identifier.

4 Optional identifier formats

This section describes optional identifier formats that allow for different types of identifiers, groups of identifiers, and checksum neutral mapping being applied. Note that identifiers are defined as unstructured fields, there is no required structure imposed on them. An administrator MAY impose an identifier format within an ILA domain. Any imposed structure is local only to the domain and all ILA nodes within the domain must agree on the format. A format might include optional elements as described below, or may include other elements customized for a domain.

4.1 Checksum neutral mapping

Checksum neutral mapping is an optional mechanism that may be applied to an ILA address (see section 5.4.1 for description of checksum-neutral mapping). When employed the checksum neutral mapping occupies the low order sixteen bits of the identifier in a locator address.



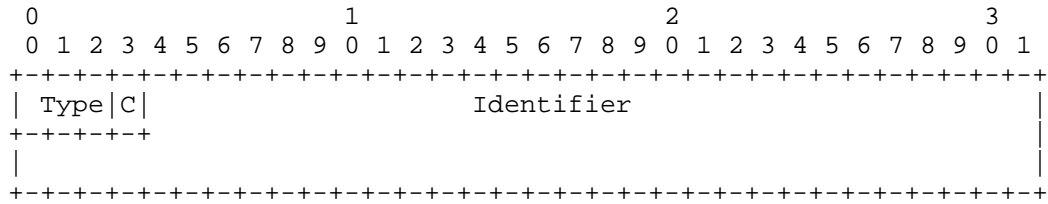
The presence of the checksum-neutral adjustment field must be unambiguous. An optional C-bit flag could be used in the identifier to indicate the checksum-neutral field is valid. The use of the C-bit is demonstrated below. Alternatively, within an ILA domain an operator could require it to be assumed that all ILA addresses have the checksum-neutral field set so that an explicit flag is not needed. Note that checksum-neutral adjustment is not used with identifier addresses.

4.2 Identifier types

This section describes an optional identifier format that allows for different types of identifiers and an indication of checksum neutral mapping being applied.

Note that the identifier type format is optional. If this is not used within an ILA domain then all ILA nodes assume that all identifiers are of the same type (locally unique identifier for instance).

The optional type format of an ILA identifier with the checksum adjust flag is:



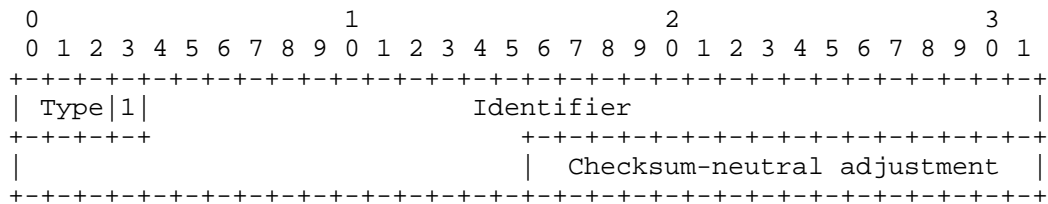
Fields are:

- o Type: Type of the identifier (see below).
- o C: The C-bit. This indicates that checksum-neutral mapping applied (see below). Presence of this field is optional.
- o Identifier: Identifier value.

Identifier types allow standard encodings for common uses of identifiers. Defined identifier types are:

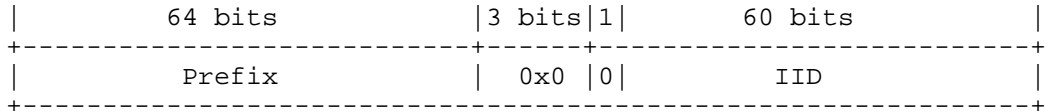
- 0: interface identifier
- 1: locally unique identifier
- 2: virtual networking identifier for IPv4 address
- 3: virtual networking identifier for IPv6 unicast address
- 4: virtual networking identifier for IPv6 multicast address
- 5: non-local address identifier
- 6-7: Reserved

If the C-bit is set then the low order sixteen bits of an identifier contain the adjustment for checksum-neutral mapping (see section 4.4.1 for description of checksum-neutral mapping). The format of an identifier with checksum neutral mapping is:



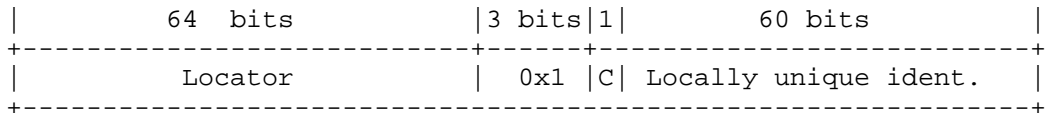
4.2.1 Interface identifiers

The interface identifier type indicates a plain local scope interface identifier. When this type is used the address is a normal IPv6 address without identifier-locator semantics. The purpose of this type is to allow normal IPv6 addresses to be defined within the same networking prefix as ILA addresses. Type bits and C-bit MUST be zero. The format of an ILA interface identifier address is:

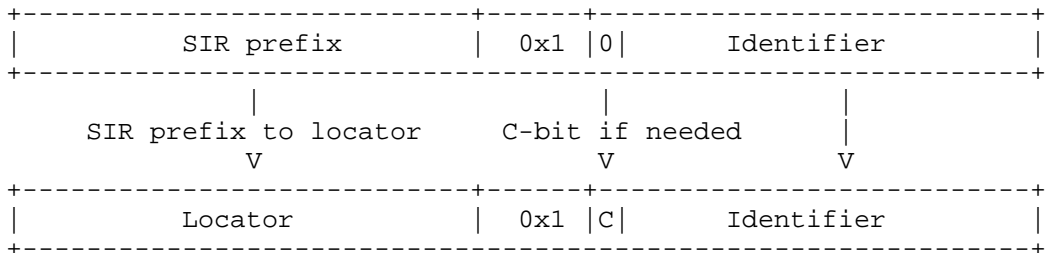


4.2.2 Locally unique identifiers

Locally unique identifiers (LUI) can be created for various addressable objects within a network. These identifiers are in a flat space and must be unique within a SIR domain (unique within a site for instance). To simplify administration, hierarchical allocation of locally unique identifiers may be performed. The format of an ILA address with locally unique identifiers is:



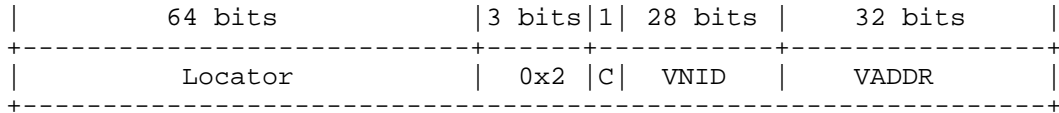
The figure below illustrates the transformation from SIR address to an ILA address as would be performed when a node sends to a SIR address. Note the low order 16 bits of the identifier may be modified as the checksum-neutral adjustment. The reverse transformation of ILA address to SIR address is symmetric.



4.2.3 Virtual networking identifiers for IPv4

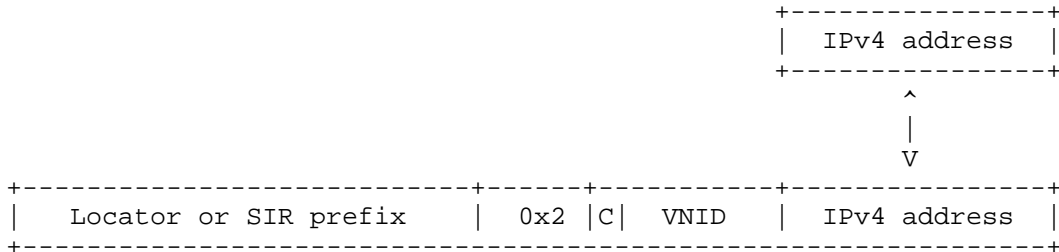
This type defines a format for encoding an IPv4 virtual address and virtual network identifier within an identifier. The format of an ILA

address for IPv4 virtual networking is:



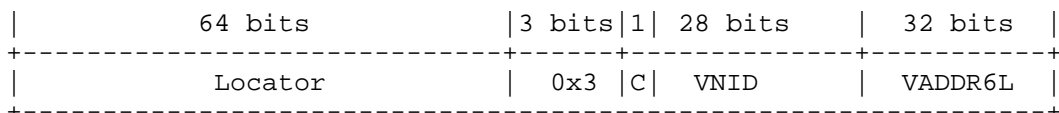
VNID is a virtual network identifier and VADDR is a virtual address within the virtual network indicated by the VNID. The VADDR can be an IPv4 unicast or multicast address, and may often be in a private address space (i.e. [RFC1918]) used in the virtual network.

Translating a virtual IPv4 address into an ILA or SIR address and the reverse transformation are straight forward. Note that the low order 16 bits of the IPv6 address may be modified as the checksum-neutral adjustment and that this transformation implies protocol translation between IPv4 and IPv6.



4.2.4 Virtual networking identifiers for IPv6 unicast

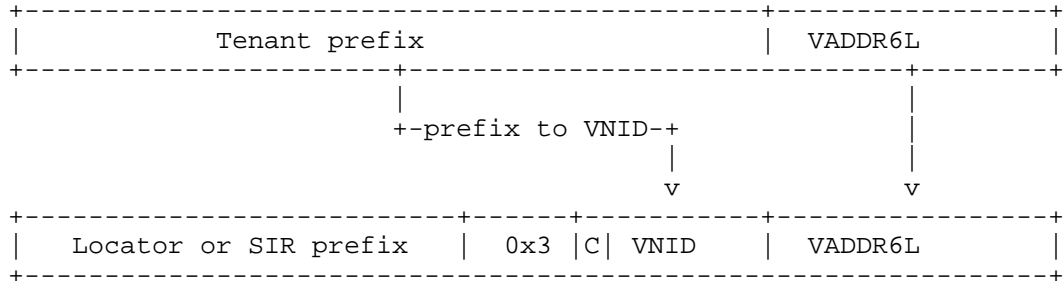
In this format, a virtual network identifier and virtual IPv6 unicast address are encoded within an identifier. To facilitate encoding of virtual addresses, there is a unique mapping between a VNID and a ninety-six bit prefix of the virtual address. The format an IPv6 unicast encoding with VNID in an ILA address is:



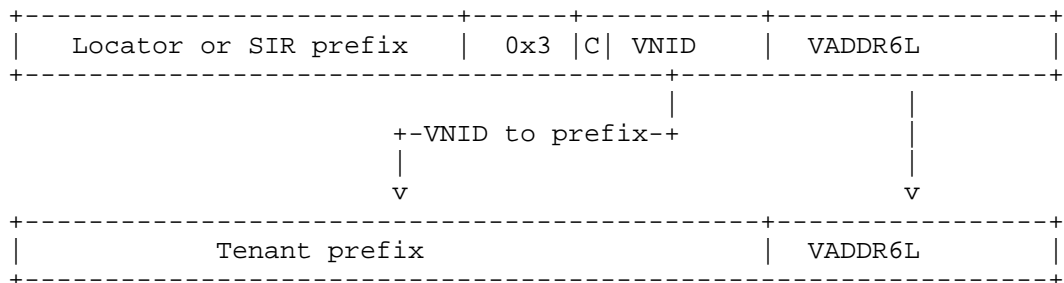
VADDR6L contains the low order 32 bits of the IPv6 virtual address. The upper 96 bits of the virtual address inferred from the VNID to prefix mapping. Note that for ILA transformations the low order sixteen of the VADDR6L may be modified for checksum-neutral adjustment.

The figure below illustrates encoding a tenant IPv6 virtual unicast

address into a ILA or SIR address.

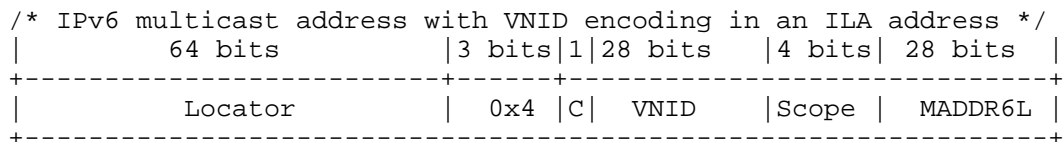


This encoding is reversible, given an ILA address, the virtual address visible to the tenant can be deduced:



4.2.5 Virtual networking identifiers for IPv6 multicast

In this format, a virtual network identifier and virtual IPv6 multicast address are encoded within an identifier.



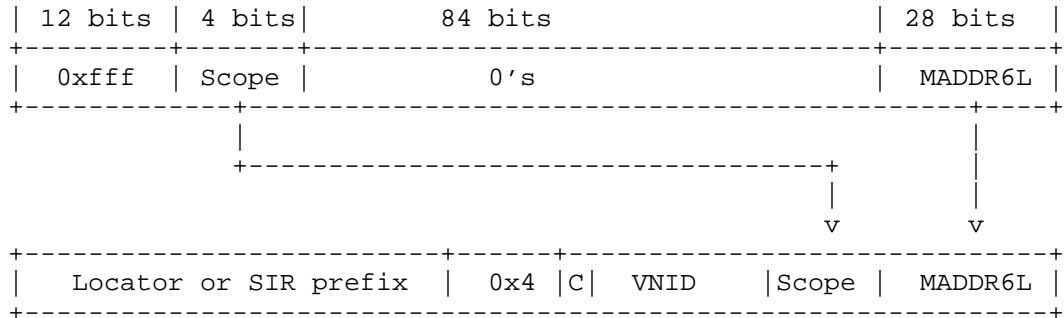
This format encodes an IPv6 multicast address in an identifier. The scope indicates multicast address scope as defined in [RFC7346]. MADDR6L is the low order 28 bits of the multicast address. The full multicast address is thus:

```
ff0<Scope>::<MADDR6L high 12 bits>:<MADDR6L low 16 bits>
```

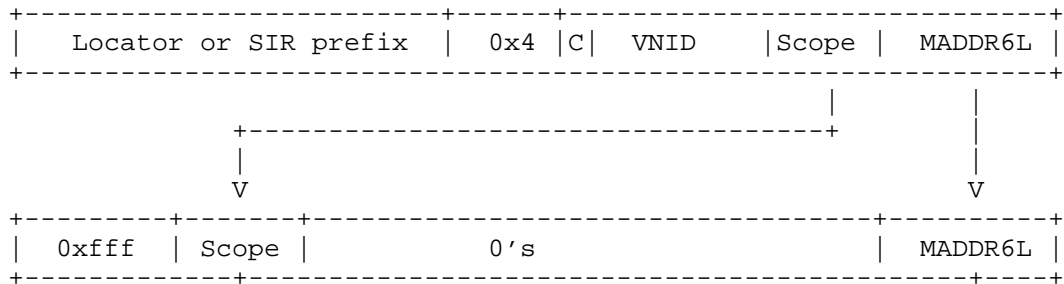
And so can encode multicast addresses of the form:

```
ff0X::0 to ff0X::0fff:ffff
```

The figure below illustrates encoding a tenant IPv6 virtual multicast address in an ILA or SIR address. Note that low order sixteen bits of MADDR6L may be modified to be the checksum-neutral adjustment.



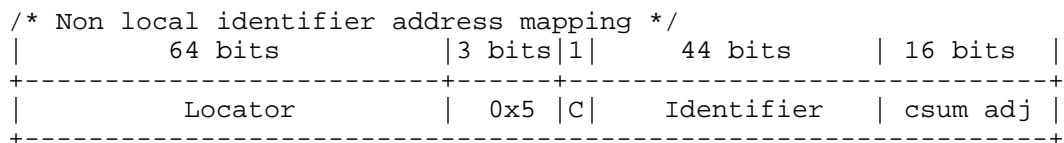
This transformation is reversible:



4.2.6 Non-local address identifiers

Non-local address identifiers allow mapping an arbitrary address to an ILA address. The mapping system contains an entry that associates an IPv6 address with an identifier. The associated IP address does not need to be a SIR address or even in the same routing domain.

The format of a locator address for a non-local address identifier is:

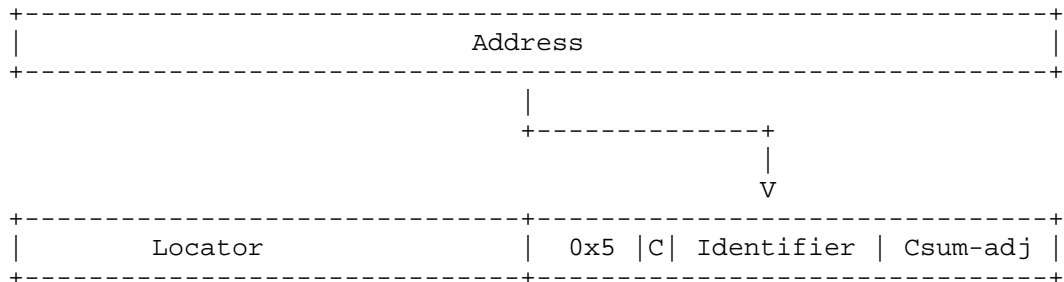


If the checksum adjust field is present it is not part of the identifier that is used in the mapping lookup. The high order bits of the address were originally not a SIR prefix, so it cannot be assumed

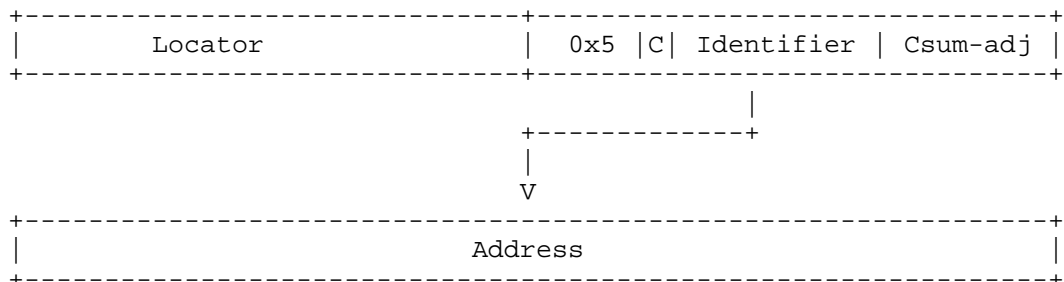
the checksum adjustment is based on a SIR prefix. The identifier is taken to be the forty-four bits that precede the checksum adjustment field. When creating the ILA address, the checksum adjustment field is initialized to zero and then set based on checksum difference between the original non-local address and the ILA address.

The figure below illustrates encoding an address into a locator address.

```
/* Non local address identifier */
```

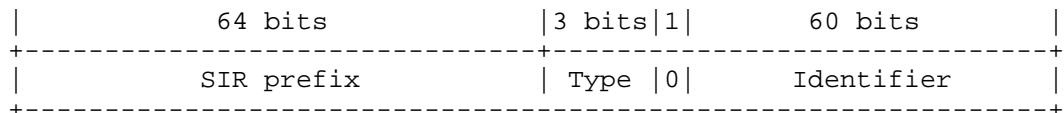


A reverse transformation is performed based on a lookup in the mapping table on the identifier (44 bits as shown above). The result of the lookup provides the original address.



4.3 SIR addresses with formatted identifiers

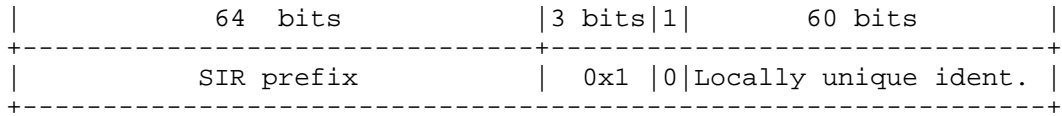
The format of a SIR address with a formatted identifier is:



The C-bit (checksum-neutral mapping) MUST be zero for a SIR address. Type may be any identifier type except zero (interface identifiers)

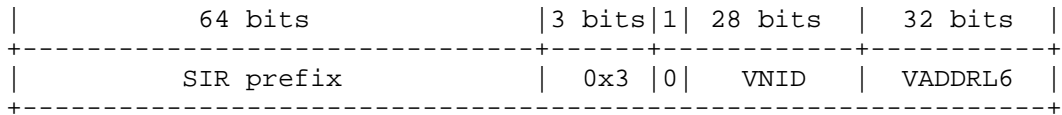
4.3.1 SIR for locally unique identifiers

The SIR address for a locally unique identifier has format:



4.3.2 SIR for virtual addresses

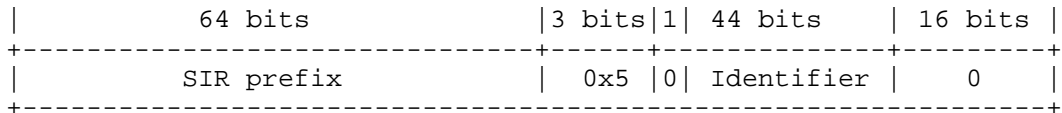
A virtual address can be encoded using the standard identifier representation. For example, the SIR address for an IPv6 virtual address may be:



Note that this allows three representations of the same address in the network: as a virtual address, a SIR address, and an ILA address.

4.3.2 SIR for non-local address identifiers

A non-local address identifier can be encoded using the standard identifier representation. For example, an encoding may be:



Note that lower order sixteen bits are set to zero since that would be the checksum adjustment value bits if transformed to an ILA address.

5 Operation

This section describes operation methods for using identifier-locator addressing.

5.1 Identifier to locator mapping

An application initiates a communication or flow using an identifier address or virtual address for a destination. In order to send a packet on the network, the destination address is transformed by an ILA node in the path. An ILA node maintains a list of mappings from

identifier to locator to perform this transformation.

The mechanisms of propagating and maintaining identifier to locator mappings are outside the scope of this document.

5.2 Address transformations

With ILA, address transformation is performed to convert identifier addresses to locator addresses, and locator addresses to identifier addresses. Transformation is usually done on a destination address as a form of source routing, however transformation on source virtual addresses to identifier addresses can also be done to support some network virtualization scenarios (see section Appendix A for examples).

5.2.1 SIR to ILA address transformation

When translating a SIR address to an ILA address, the SIR prefix in the address is overridden with a locator, and checksum neutral mapping may be performed. Since this operation is potentially done for every packet the process should be very efficient (particularly the lookup and checksum processing operations).

The typical steps to transmit a packet using ILA are:

- 1) Host stack creates a packet with source address set to a local address (possibly a SIR address) for the local identity, and the destination address is set to the SIR address or virtual address for the peer. The peer address may have been discovered through DNS or other means.
- 2) An ILA node translates the packet to use the locator. If the original destination address is a SIR address then the SIR prefix is overwritten with the locator. If the original packet is a virtually addressed tenant packet then the virtual address is transformed per section 4.2. The locator is discovered by a lookup in the locator to identifier mappings.
- 3) The ILA node performs checksum-neutral mapping if configured for that (section 5.4).
- 4) Packet is forwarded on the wire. The network routes the packet to the node indicated by the locator.

5.2.2 ILA to SIR address transformation

When a destination node (ILA router or end host) receives an ILA addressed packet, the ILA address MUST be transformed back to a SIR

address (or virtual address) before upper layer processing.

The steps of receive processing are:

- 1) Packet is received. The destination locator is verified to match a locator assigned to the node.
- 2) A lookup is performed on the destination identifier to find if it addresses a local identifier. If match is found, either the locator is overwritten with SIR prefix (for locally unique identifier type) or the address is transformed back to a tenant virtual address.
- 3) Perform reverse checksum-neutral mapping if C-bit is set (section 5.4).
- 4) Perform any optional policy checks; for instance that the source may send a packet to the destination address, that packet is not illegitimately crossing virtual networks, etc.
- 5) Forward packet to the application.

5.3 Virtual networking operation

When using ILA with virtual networking identifiers, address transformation is performed to convert tenant virtual network and virtual addresses to ILA addresses, and ILA addresses back to a virtual network and tenant's virtual addresses. Transformation may occur on either source address, destination address, or both (see scenarios for virtual networking in Appendix A). Address transformation is performed similar to the SIR transformation cases described above.

5.3.1 Crossing virtual networks

With explicit configuration, virtual network hosts may communicate directly with virtual hosts in another virtual network by using identifier addresses for virtualization in both the source and destination addresses. This might be done to allow services in one virtual network to be accessed from another (by prior agreement between tenants). See appendix A.13 for example of ILA addressing for such a scenario.

5.3.2 IPv4/IPv6 protocol translation

An IPv4 tenant may send a packet that is converted to an IPv6 packet with ILA addresses. Similarly, an IPv6 packet with ILA addresses may be converted to an IPv4 packet to be received by an IPv4-only tenant.

These are IPv4/IPv6 stateless protocol translations as described in [RFC6144] and [RFC6145]. See appendix A.12 for a description of these scenarios.

5.4 Transport layer checksums

Packets undergoing ILA transformation may encapsulate transport layer checksums (e.g. TCP or UDP) that include a pseudo header that is affected by the transformation.

ILA provides two alternatives do deal with this:

- o Perform a checksum-neutral mapping to ensure that an encapsulated transport layer checksum is kept correct on the wire.
- o Send the checksum as-is, that is send the checksum value based on the pseudo header before transformation.

Some intermediate devices that are not the actual end point of a transport protocol may attempt to validate transport layer checksums. In particular, many Network Interface Cards (NICs) have offload capabilities to validate transport layer checksums (including any pseudo header) and return a result of validation to the host. Typically, these devices will not drop packets with bad checksums, they just pass a result to the host. Checksum offload is a performance benefit, so if packets have incorrect checksums on the wire this benefit is lost. With this incentive, using checksum-neutral mapping is recommended. If it is known that the addresses of a packet are not included in a transport checksum, for instance a GRE packet is being encapsulated, then a source may choose not to perform checksum-neutral mapping.

5.4.1 Checksum-neutral mapping

When a change is made to one of the IP header fields in the IPv6 pseudo-header checksum (such as one of the IP addresses), the checksum field in the transport layer header may become invalid. Fortunately, an incremental change in the area covered by the Internet standard checksum [RFC1071] will result in a well-defined change to the checksum value [RFC1624]. So, a checksum change caused by modifying part of the area covered by the checksum can be corrected by making a complementary change to a different 16-bit field covered by the same checksum.

ILA can perform a checksum-neutral mapping when a SIR prefix or virtual address is transformed to a locator in an IPv6 address, and performs the reverse mapping when translating a locator back to a SIR

prefix or virtual address. The low order sixteen bits of the identifier contain the checksum adjustment value for ILA.

On transmission, the transformation process is:

- 1) Compute the one's complement difference between the SIR prefix and the locator. Fold this value to 16 bits (add-with-carry four 16-bit words of the difference).
- 2) If the C-bit is to be used then add-with-carry the bit-wise not of the 0x1000 (i.e. 0xefff) to the value from #1. This compensates the checksum for setting the C-bit.
- 3) Add-with-carry the value from #2 to the low order sixteen bits of the identifier.
- 4) Set the resultant value from #3 in the low order sixteen bits of the identifier and set the C-bit if it is to be present.

Note that the "adjustment" (the 16-bit value set in the identifier) is fixed for a given SIR to locator mapping, so the adjustment value can be saved in an associated data structure for a mapping to avoid computing it for each transformation.

On reception of an ILA addressed packet, if checksum-neutral mapping is applied to the packet (either the C-bit is set or its used is assumed for the ILA domain):

- 1) Compute the one's complement difference between the locator in the address and the SIR prefix that the locator is being transformed to. Fold this value to 16 bits (add-with-carry four 16-bit words of the difference).
- 2) If the C-bit is used then add-with-carry 0x1000 to the value from #1. This compensates the checksum for clearing the C-bit.
- 3) Add-with-carry the value from #2 to the low order sixteen bits of the identifier.
- 4) Set the resultant value from #3 in the low order sixteen bits of the identifier and clear the C-bit if its present. This restores the original identifier sent in the packet.

Note that receive checksum-neutral mapping process requires that the original upper sixty four bits in the address can be deduced. The method for this is different based on identifier type:

- o interface identifier: checksum-neutral mapping is not used.

- o locally unique identifier: the SIR prefix is inferred from the one to one mapping with a locator.
- o virtual network identifier for IPv4: the original upper sixty-four bits are assumed to be zero.
- o virtual network identifier for IPv6 unicast: the VNID in the identifier is mapped to a tenant prefix that includes the original upper sixty-four bits.
- o virtual network identifier for IPv6 multicast: the original upper sixty-four bits can be deduced by from the scope field in the identifier and fixed field of the multicast address.
- o non-local address identifier: the identifier, not including the low order sixteen bits of the address, is used to lookup the original address. Since the full address is provided by the lookup, the process to undo a checksum-neutral mapping can be obviated in this case

5.4.2 Sending an unmodified checksum

When sending an unmodified checksum, the checksum is incorrect as viewed in the packet on the wire. At the receiver, ILA transformation of the destination ILA address back to the SIR address occurs before transport layer processing. This ensures that the checksum can be verified when processing the transport layer header containing the checksum. Intermediate devices are not expected to drop packets due to a bad transport layer checksum.

5.5 Non-local address mapping

Non-local addresses may be mapped into ILA addresses using non-local address identifiers. This allows transit of such addresses across the underlay of an ILA domain. This would be useful for handling addresses in a network that originate from an external source. An example of this would be roaming in cellular network so that a device can continue using addresses that are part of its home network.

A packet may be forwarded to an ILA router that has a non-local destination address which is not a identifier address for the domain. An ILA router can perform a lookup on the full address in an alternate mapping table. If there is a match, an identifier is returned that reverses maps to the address. This identifier is in the ILA domain space and identifies the node with the non-local address. A normal mapping table lookup can then be done to get the locator for the node in the ILA domain.

At a peer ILA router, a lookup is performed on the destination identifier in a table that maps the non-local address identifier to the original non-local address. If an entry is found, the address is set in the destination address and the packet is forward to the destination.

Note that the non-local address to identifier mapping and its reverse mapping must be set in the table before hand.

5.6 Address assignment

ILA supports single address assignments as well as prefix assignments. ILA will also support strong privacy in addressing [ADDRPRIV].

5.6.1 Singleton address assignment

Singleton addresses can use a canonical 64/64 locator/identifier split. Singleton addresses can be assigned by DHCPv6.

5.6.2 Network prefix assignment

Prefix assignment can be done via SLAAC or DHCPv6-PD.

To support /64 prefix assignment with ILA, the ILA identifier can be encoded in the upper sixty-four bits of an address. A level of indirection is used so that ILA transforms the upper sixty four bits to contain both a locator and an index into a locator (ILA node) specific table. The entry in the table provides the original sixty-four bit prefix so that locator to identifier address transformation can be done. As an example of this scheme, suppose network has a /24 prefix. The identifier address format for /64 assignment might be:

24 bits	40 bits	64 bits	
Network	Identifier	IID	

The IID part is arbitrarily assigned by the device, so that is ignored by ILA. All routing, lookups, and transformations (excepting checksum neutral mapping) are based on the upper sixty-four bits. For identifier to locator address transformation, a lookup is done on the upper sixty-four bits. That returns a value that contains a locator and a locator table index. The resulting packet format may be something like:

24 bits	20 bits	20 bits	64 bits
Network	Locator	Loc index	IID

The packet is forwarded and routed as addressed by locator (/44 route in this case). At the ILA forwarding node, the locator index is used as a key to an ILA node specific table that returns a 40 bit Identifier. This value is then written in the packet do ILA to identifier address transformation thereby restoring the original destination address. The locator index is not globally unique, it is specific to each ILA node. When an end node attaches to an ILA node, an index is chosen so that the table is populated at the ILA node and the ILA mapping includes the locator and index. When a node detaches from on ILA, it's entry in the table is removed and the index can be reused after a hold-down period to allow stale mappings to be purged.

5.6.3 Strong privacy addresses

Note that when a /64 is assigned to end hosts (such as UEs in a mobile network), the assigned prefix may become a persistent identifier for a device. This is a potential privacy issue. [ADDPRIV] describes this problem and suggests some solutions that may be used with ILA.

5.7 Address selection

There may be multiple possibilities for creating either a source or destination address. A node may be associated with more than one identifier, and there may be multiple locators for a particular identifier. The choice of locator or identifier is implementation or configuration specific. The selection of an identifier occurs at flow creation and must be invariant for the duration of the flow. Locator selection must be done at least once per flow, and the locator associated with the destination of a flow may change during the lifetime of the flow (for instance in the case of a migrating connection it will change). ILA address selection should follow specifications in Default Address Selection for Internet Protocol Version 6 (IPv6) [RFC6724].

5.8 Duplicate identifier detection

As part of implementing the locator to identifier mapping, duplicate identifier detection should be implemented in a centralized control plane. A registry of identifiers could be maintained (possibly in association with the identifier to locator mapping database). When a node creates an identifier it registers the identifier, and when the identifier is no longer in use the identifier is unregistered. The

control plane should be able to detect a registration attempt for an existing identifier and deny the request.

5.9 ICMP error handling

A packet that contains an ILA address may cause ICMP errors within the network. In this case the ICMP data contains an IP header with an ILA address. ICMP messages are sent back to the source address in the packet. Upon receiving an ICMP error the host will process it differently depending on whether it is ILA capable.

5.9.1 Handling ICMP errors by ILA capable hosts

If a host is ILA capable it can attempt to reverse translate the ILA address in the destination of a header in the ICMP data back to a SIR address that was originally used to transmit the packet. The steps are:

- 1) Assume that the upper sixty-four bits of the destination address in the ICMP data is a locator. Match these bits to a SIR address. If the host is only in one SIR domain, then the mapping to SIR address is implicit. If the host is in multiple domains then a locator to SIR addresses table can be maintained for this lookup.
- 2) If the identifier includes checksum-neutral mapping, undo the checksum-neutral mapping using the SIR address found in #1 and the process in section 5.4.1. The resulting identifier address is potentially the original address used to send the packet.
- 3) Lookup the identifier in the identifier to locator mapping table. If an entry is found compare the locator in the entry to the locator (upper sixty-four bits) of the destination address in the IP header of the ICMP data. If these match then proceed to next step.
- 4) Overwrite the upper sixty-four bits of the destination address in the ICMP data with the found SIR prefix and overwrite the low order sixty-four bits with the found identifier (the result of undoing checksum-neutral mapping). The resulting address should be the original SIR address used in sending. The ICMP error packet can then be received by the stack for further processing.

5.9.2 Handling ICMP errors by non-ILA capable hosts

A non-ILA capable host may receive an ICMP error generated by the network that contains an ILA address in IP header contained in the

ICMP data. This would happen in the case that an ILA router performed transformation on a packet the host sent and that packet subsequently generated an ICMP error. In this case the host receiving the error message will attempt to find the connection state corresponding to the packet header in the ICMP data. Since the host is unaware of ILA the lookup for connection state should fail. Because the host cannot recover the original addresses it used to send the packet, it won't be able any to derive any useful information about the original destination of the packet that it sent.

If packets for a flow are always routed through an ILA router in both directions, for example ILA routers are coincident with edge routers in a network, then ICMP errors could be intercepted by an intermediate node which could translate the destination addresses in ICMP data back to the original SIR addresses. A receiving host would then see the destination address in the packet of the ICMP data to be that it used to transmit the original packet.

5.10 Multicast

ILA is generally not intended for use with multicast. In the case of multicast, routing of packets is based on the source address. Neither the SIR address nor an ILA address is suitable for use as a source address in a multicast packet. A SIR address is unroutable and hence would make a multicast packet unroutable if used as a source address. Using an ILA address as the source address makes the multicast packet routable, but this exposes ILA address to applications which is especially problematic on a multicast receiver that doesn't support ILA.

If all multicast receivers are known to support ILA, a local locator address may be used in the source address of the multicast packet. In this case, each receiver will translate the source address from an ILA address to a SIR address before delivering packets to an application.

6 Motivation for ILA

6.1 Use cases

6.1.1 Multi-tenant virtualization

In multi-tenant virtualization overlay networks are established for tenants to provide virtual networks. Each tenant may have one or more virtual networks and a tenant's nodes are assigned virtual addresses within virtual networks. Identifier-locator addressing may be used as an alternative to traditional network virtualization encapsulation protocols used to create overlay networks (e.g. VXLAN [RFC7348]).

Tenant systems (e.g. VMs) run on physical hosts and may migrate to different hosts. A tenant system is identified by a virtual address and virtual networking identifier of a corresponding virtual network. ILA can encode the virtual address and a virtual networking identifier in an ILA identifier. Each identifier is mapped to a locator that indicates the current host where the tenant system resides. Nodes that send to the tenant system set the locator per the mapping. When a tenant system migrates, its identifier to locator mapping is updated and communicating nodes will use the new mapping.

6.1.2 Datacenter virtualization

Datacenter virtualization virtualizes networking resources. Various objects within a datacenter can be assigned addresses and serve as logical endpoints of communication. A large address space, for example that of IPv6, allows addressing to be used beyond the traditional concepts of host based addressing. Addressed objects can include tasks, virtual IP addresses (VIPs), pieces of content, disk blocks, etc. Each object has a location which is given by the host on which an object resides. Some objects may be migratable between hosts such that their location changes over time.

Objects are identified by a unique identifier within a namespace for the datacenter (appendix B discusses methods to create unique identifiers for ILA). Each identifier is mapped to a locator that indicates the current host where the object resides. Nodes that send to an object set the locator per the mapping. When an object migrates its identifier to locator mapping is updated and communicating nodes will use the new mapping.

A datacenter object of particular interest is tasks, units of execution for applications. The goal of virtualizing tasks is to maximize resource efficiency and job scheduling. Tasks share many properties of tenant systems, however they are finer grained objects, may have a shorter lifetimes, and are likely created in greater numbers. Appendix C provides more detail and motivation for virtualizing tasks using ILA.

6.1.3 Mobile networks

ILA may be applied as a solution for mobility in mobile networks (such as cellular networks). In mobile networks, devices such as smart phones move physically within the network. When a device moves it changes its point of attachment in the network. The goal of mobility is to provide a seamless transition when a device moves from one attachment point to another. Appendix D provides more detail and motivation for ILA in wireless networks.

Each mobile device in a network may be assigned one or more identifiers to use in communications. The ILA mapping table has an entry for each identifier that maps to a locator indicating the current network point of attachment for the device. Nodes that send to the device set the locator per the mapping. When a mobile device moves to a new attachment point, then mapping table entries all of its associated identifiers are updated with a new locator.

6.2 Alternative methods

This section discusses the merits of alternative solution that have been proposed to provide network virtualization or mobility in IPv6.

6.2.1 ILNP

ILNP splits an address into a locator and identifier in the same manner as ILA. ILNP has characteristics, not present in ILA, that prevent it from being a practical solution:

- o ILNP requires that transport layer protocol implementations must be modified to work over ILNP.
- o ILNP can only be implemented in end hosts, not within the network. This essentially requires that all end hosts need to be modified to participate in mobility.

6.2.2 Flow label as virtual network identifier

The IPv6 flow label could conceptually be used as a 20-bit virtual network identifier in order to indicate a packet is sent on an overlay network. In this model the addresses may be virtual addresses within the specified virtual network. Presumably, the tuple of flow-label and addresses could be used by switches to forward virtually addressed packets.

This approach has some issues:

- o Forwarding virtual packets to their physical location would require specialized switch support.
- o The flow label is only twenty bits, this is too small to be a discriminator in forwarding a virtual packet to a specific destination. Conceptually, the flow label might be used in a type of label switching to solve that.
- o The flow label is not considered immutable in transit, intermediate devices may change it.

- o The flow label is not part of the pseudo header for transport checksum calculation, so it is not covered by any transport (or other) checksums.

6.2.3 Extension headers

To accomplish network virtualization an extension header, such as a destination or routing option, could be used that contains the virtual destination address of a packet. The destination address in the IPv6 header would be the topological address for the location of the virtual node. Conceivably, segment routing could be used to implement network virtualization in this manner.

This technique has some issues:

- o Intermediate devices must not insert extension headers [RFC8200].
- o Extension headers introduce additional packet overhead which may impact performance.
- o Extension headers are not covered by transport checksums (as the addresses would be) nor any other checksum.
- o Extension headers are not widely supported in network hardware or devices. For instance, several NIC offloads don't work in the presence of extension headers.

6.2.4 Encapsulation techniques

Various encapsulation techniques have been proposed for implementing network virtualization and mobility. LISP is an example of an encapsulation that is based on locator identifier separation similar to ILA. The primary drawback of encapsulation is complexity and per packet overhead. For instance, when LISP is used with IPv6 the encapsulation overhead is fifty-six bytes and two IP headers are present in every packet. This adds considerable processing costs, requires considerations to handle path MTU correctly, and certain network accelerations may be lost.

7 Security Considerations

Security must be considered when using identifier-locator addressing. In particular, the risk of address spoofing or address corruption must be addressed. To classify this risk the set possible destinations for a packet are classified as trusted or untrusted. The set of possible destinations includes those that a packet may inadvertently be sent due to address or header corruption.

If the set of possible destinations are trusted then packet misdelivery is considered relatively innocuous. This might be the case in a data center if all nodes were tightly controlled under single management. Identifier-locator addressing can be used in this case without further additional security.

If the set of possible destinations contains untrusted hosts, then packet misdelivery could be a risk. This may be the case that virtual machines with untrusted third party applications or OSes are running in the network. A malicious user may be snooping for misdelivered packets, or may attempt to spoof addresses. Identifier-locator addressing should be used with stronger security and isolation mechanisms such as IPsec or GUESEC.

8 IANA Considerations

There are no IANA considerations in this specification.

9 References

9.1 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>>.
- [RFC4291] Hinden, R. and S. Deering, "IP Version 6 Addressing Architecture", RFC 4291, February 2006.
- [RFC6296] Wasserman, M. and F. Baker, "IPv6-to-IPv6 Network Prefix Translation", RFC 6296, June 2011.
- [RFC1071] Braden, R., Borman, D., Partridge, C., and W. Plummer, "Computing the Internet checksum", RFC 1071, September 1988.
- [RFC1624] Rijssinghani, A., "Computation of the Internet Checksum via Incremental Update", RFC 1624, May 1994.
- [RFC6724] Thaler, D., Ed., Draves, R., Matsumoto, A., and T. Chown, "Default Address Selection for Internet Protocol Version 6 (IPv6)", RFC 6724, September 2012.

9.2 Informative References

- [RFC6740] RJ Atkinson and SN Bhatti, "Identifier-Locator Network Protocol (ILNP) Architectural Description", RFC 6740, November 2012.
- [RFC6741] RJ Atkinson and SN Bhatti, "Identifier-Locator Network Protocol (ILNP) Engineering Considerations", RFC 6741, November 2012.
- [RFC1918] Rekhter, Y., Moskowitz, B., Karrenberg, D., de Groot, G., and E. Lear, "Address Allocation for Private Internets", BCP 5, RFC 1918, February 1996.
- [RFC3363] Bush, R., Durand, A., Fink, B., Gudmundsson, O., and T. Hain, "Representing Internet Protocol version 6 (IPv6) Addresses in the Domain Name System (DNS)", RFC 3363, August 2002.
- [RFC3587] Hinden, R., Deering, S., and E. Nordmark, "IPv6 Global Unicast Address Format", RFC 3587, August 2003.

- [RFC6144] Baker, F., Li, X., Bao, C., and K. Yin, "Framework for IPv4/IPv6 Translation", RFC 6144, April 2011.
- [RFC8014] Black, D., Hudson, J., Kreeger, L., Lasserre, M., and T. Narten, "An Architecture for Data-Center Network Virtualization over Layer 3 (NVO3)", RFC 8014, DOI 10.17487/RFC8014, December 2016, <<https://www.rfc-editor.org/info/rfc8014>>.
- [GUE] Herbert, T., and Yong, L., "Generic UDP Encapsulation", draft-ietf-intarea-gue-04, work in progress.
- [GUESEC] Yong, L., and Herbert, T. "Generic UDP Encapsulation (GUE) for Secure Transport", draft-hy-gue-4-secure-transport-03, work in progress
- [ADDRPRIV] Herbert, T., "Privacy in IPv6 Network Prefix Assignment", draft-herbert-ipv6-prefix-address-privacy-00

10 Acknowledgments

The authors would like to thank Mark Smith, Lucy Yong, Erik Kline, Saleem Bhatti, Blake Matheny, Doug Porter, Pierre Pfister, Fred Baker, and Fred Baker for their insightful comments for this draft; Roy Bryant, Lorenzo Colitti, Mahesh Bandewar, and Erik Kline for their work on defining and applying ILA; Kalyani Bogineni, Niranjan Avula, Behcet Sarikaya, Dirk von-Hugo, and Ratul Guha for insights regarding the mobility use case.

Appendix A: Communication scenarios

This section describes the use of identifier-locator addressing in several scenarios.

A.1 Terminology for scenario descriptions

A formal notation for identifier-locator addressing with ILNP is described in [RFC6740]. We extend this to include for network virtualization cases.

Basic terms are:

- A = IP Address
- I = Identifier
- L = Locator
- LUI = Locally unique identifier
- VNI = Virtual network identifier
- VA = An IPv4 or IPv6 virtual address
- VAX = An IPv6 networking identifier (IPv6 VA mapped to VAX)
- SIR = Prefix for standard identifier representation
- VNET = IPv6 prefix for a tenant (assumed to be globally routable)
- Iaddr = IPv6 address of an Internet host

An ILA IPv6 address is denoted by

L:I

A SIR address with a locally unique identifier and SIR prefix is denoted by

SIR:LUI

A virtual identifier with a virtual network identifier and a virtual IPv4 address is denoted by

VNI:VA

An ILA IPv6 address with a virtual networking identifier for IPv4 would then be denoted

L:(VNI:VA)

The local and remote address pair in a packet or endpoint is denoted

A,A

An address translation sequence from SIR addresses to ILA addresses

for transmission on the network and back to SIR addresses at a receiver has notation:

A,A -> L:I,A -> A,A

A.2 Identifier objects

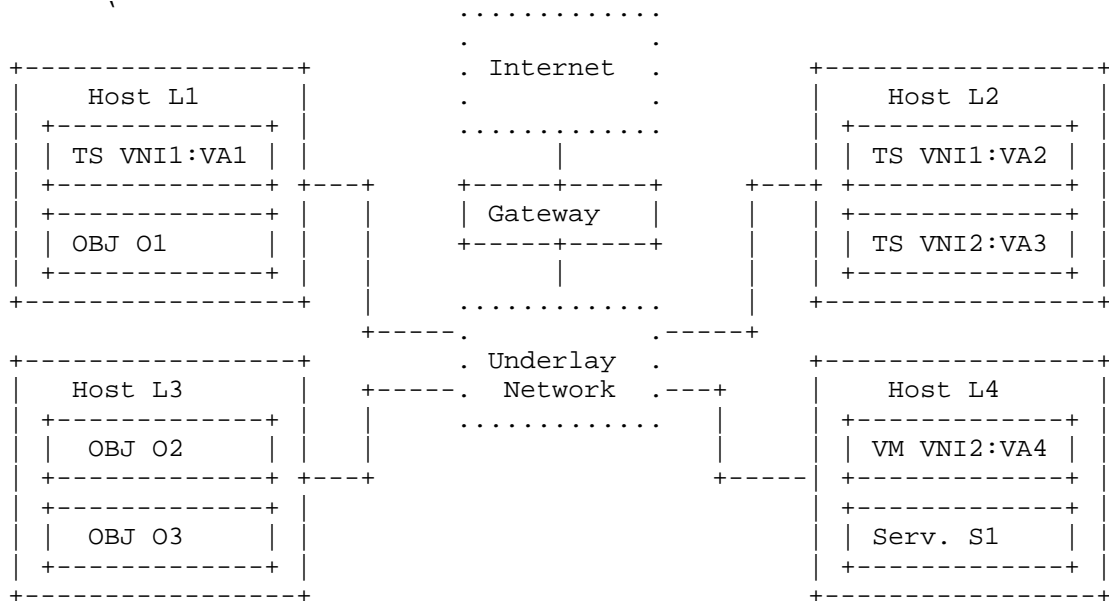
Identifier-locator addressing is broad enough in scope to address many different types of networking entities. For the purposes of this section we classify these as "objects" and "tenant systems".

Objects encompass uses where nodes are address by local unique identifiers (LUI). In the scenarios below objects are denoted by OBJ.

Tenant systems are those associated with network virtualization that have virtual addresses (that is they are addressed by VNI:VA). In the scenarios below tenant systems are denoted by TS.

A.3 Reference network for scenarios

The figure below provides an example network topology with ILA addressing in use. In this example, there are four hosts in the network with locators L1, L2, L3, and L4. There three objects with identifiers O1, O2, and O3, as well as a common networking service with identifier S1. There are two virtual networks VNI1 and VNI2, and four tenant systems addressed as: VA1 and VA2 in VNI1, VA3 and VA4 in VNI2. The network is connected to the Internet via a gateway.



Several communication scenarios can be considered:

- 1) Object to object
- 2) Object to Internet
- 3) Internet to object
- 4) Tenant system to local service
- 5) Object to tenant system
- 6) Tenant system to Internet
- 7) Internet to tenant system
- 8) IPv4 tenant system to service
- 9) Tenant system to tenant system same virtual network using IPv6
- 10) Tenant system to tenant system in same virtual network using IPv4
- 11) Tenant system to tenant system in different virtual network using IPv6
- 12) Tenant system to tenant system in different virtual network using IPv4
- 13) IPv4 tenant system to IPv6 tenant system in different virtual networks
- 14) Non-local address to tenant system

A.4 Scenario 1: Object to task

The transport endpoints for object to object communication are the SIR addresses for the objects. When a packet is sent on the wire, the locator is set in the destination address of the packet. On reception the destination addresses is converted back to SIR representation for processing at the transport layer.

If task T1 is communicating with task T2, the ILA translation sequence would be:

```
SIR:O1,SIR:O2 ->           // Transport endpoints on O1
SIR:O1,L3:O2 ->           // ILA used on the wire
SIR:O1,SIR:O2             // Received at O2
```

A.5 Scenario 2: Object to Internet

Communication from an object to the Internet is accomplished through use of a SIR address (globally routable) in the source address of packets. No ILA translation is needed in this path.

If object O1 is sending to an address Iaddr on the Internet, the packet addresses would be:

```
SIR:O1,Iaddr
```

A.6 Scenario 3: Internet to object

An Internet host transmits a packet to a task using an externally routable SIR address. The SIR prefix routes the packet to a gateway for the datacenter. The gateway translates the destination to an ILA address.

If a host on the Internet with address Iaddr sends a packet to object O3, the ILA translation sequence would be:

```
Iaddr,SIR:O3 -> // Transport endpoint at Iaddr
Iaddr,L1:O3 -> // On the wire in datacenter
Iaddr,SIR:O3 // Received at O3
```

A.7 Scenario 4: Tenant system to service

A tenant can communicate with a datacenter service using the SIR address of the service.

If TS VA1 is communicating with service S1, the ILA translation sequence would be:

```
VNET:VA1,Saddr-> // Transport endpoints in TS
SIR:(VNET:VA1):Saddr-> // On the wire
SIR:(VNET:VA1):Saddr // Received at S1
```

Where VNET is the address prefix for the tenant and Saddr is the IPv6 address of the service.

The ILA translation sequence in the reverse path, service to tenant system, would be:

```
Saddr,SIR:(VNET:VA1) // Transport endpoints in S1
Saddr,L1:(VNET:VA1) // On the wire
Saddr,VNET:VA1 // Received at the TS
```

Note that from the point of view of the service task there is no material difference between a peer that is a tenant system versus one which is another task.

A.8 Scenario 5: Object to tenant system

An object can communicate with a tenant system through it's externally visible address.

If object O2 is communicating with TS VA4, the ILA translation sequence would be:

```
SIR:O2,VNET:VA4 -> // Transport endpoints at T2
SIR:O2,L4:(VNI2:VAX4) -> // On the wire
```

```
SIR:O2,VNET:VA4 // Received at TS
```

A.9 Scenario 6: Tenant system to Internet

Communication from a TS to the Internet assumes that the VNET for the TS is globally routable, hence no ILA translation would be needed.

If TS VA4 sends a packet to the Internet, the addresses would be:

```
VNET:VA4,Iaddr
```

A.10 Scenario 7: Internet to tenant system

An Internet host transmits a packet to a tenant system using an externally routable tenant prefix and address. The prefix routes the packet to a gateway for the datacenter. The gateway translates the destination to an ILA address.

If a host on the Internet with address Iaddr is sending to TS VA4, the ILA translation sequence would be:

```
Iaddr,VNET:VA4 -> // Endpoint at Iaddr
Iaddr,L4:(VNI2:VAX4) -> // On the wire in datacenter
Iaddr,VNET:VA4 // Received at TS
```

A.11 Scenario 8: IPv4 tenant system to object

A TS that is IPv4-only may communicate with an object using protocol translation. The object would be represented as an IPv4 address in the tenant's address space, and stateless NAT64 should be usable as described in [RFC6145].

If TS VA2 communicates with object O3, the ILA translation sequence would be:

```
VA2,ADDR3 -> // IPv4 endpoints at TS
SIR:(VNI1:VA2),L3:O3 -> // On the wire in datacenter
SIR:(VNI1:VA2),SIR:O3 // Received at task
```

VA2 is the IPv4 address in the tenant's virtual network, ADDR4 is an address in the tenant's address space that maps to the network service.

The reverse path, task sending to a TS with an IPv4 address, requires a similar protocol translation.

For object O3 communicate with TS VA2, the ILA translation sequence would be:

```

SIR:O3,SIR:(VNI1:VA2) ->           // Endpoints at T4
SIR:O3,L2:(VNI1:VA2) ->           // On the wire in datacenter
ADDR4,VA2                           // IPv4 endpoint at TS

```

A.12 Tenant to tenant system in the same virtual network

ILA may be used to allow tenants within a virtual network to communicate without the need for explicit encapsulation headers.

A.12.1 Scenario 9: TS to TS in the same VN using IPV6

If TS VA1 sends a packet to TS VA2, the ILA translation sequence would be:

```

VNET:VA1,VNET:VA2 ->               // Endpoints at VA1
VNET:VA1,L2:(VNI1,VAX2) ->        // On the wire
VNET:VA1,VNET:VA2 ->               // Received at VA2

```

A.12.2 Scenario 10: TS to TS in same VN using IPv4

For two tenant systems to communicate using IPv4 and ILA, IPv4/IPv6 protocol translation is done both on the transmit and receive.

If TS VA1 sends an IPv4 packet to TS VA2, the ILA translation sequence would be:

```

VA1,VA2 ->                         // Endpoints at VA1
SIR:(VNI1:VA1),L2:(VNI1,VA2) ->    // On the wire
VA1,VA2                             // Received at VA2

```

Note that the SIR is chosen by an ILA node as an appropriate SIR prefix in the underlay network. Tenant systems do not use SIR address for this communication, they only use virtual addresses.

A.13 Tenant system to tenant system in different virtual networks

A tenant system may be allowed to communicate with another tenant system in a different virtual network. This should only be allowed with explicit policy configuration.

A.13.1 Scenario 11: TS to TS in different VNs using IPV6

For TS VA4 to communicate with TS VA1 using IPv6 the translation sequence would be:

```

VNET2:VA4,VNET1:VA1->              // Endpoint at VA4
VNET2:VA4,L1:(VNI1,VAX1)->         // On the wire
VNET2:VA4,VNET1:VA1                 // Received at VA1

```

Note that this assumes that VNET1 and VNET2 are globally routable between the two virtual networks.

A.13.2 Scenario 12: TS to TS in different VNs using IPv4

To allow IPv4 tenant systems in different virtual networks to communicate with each other, an address representing the peer would be mapped into each tenant's address space. IPv4/IPv6 protocol translation is done on transmit and receive.

For TS VA4 to communicate with TS VA1 using IPv4 the translation sequence may be:

```
VA4,SADDR1 ->                // IPv4 endpoint at VA4
SIR:(VNI2:VA4),L1:(VNI1,VA1)-> // On the wire
SADDR4,VA1                    // Received at VA1
```

SADDR1 is the mapped address for VA1 in VA4's address space, and SADDR4 is the mapped address for VA4 in VA1's address space.

A.13.3 Scenario 13: IPv4 TS to IPv6 TS in different VNs

Communication may also be mixed so that an IPv4 tenant system can communicate with an IPv6 tenant system in another virtual network. IPv4/IPv6 protocol translation is done on transmit.

For TS VA4 using IPv4 to communicate with TS VA1 using IPv6 the translation sequence may be:

```
VA4,SADDR1 ->                // IPv4 endpoint at VA4
SIR:(VNI2:VA4),L1:(VNI1,VAX1)-> // On the wire
SIR:(VNI2:VA4),VNET1:VA1      // Received at VA1
```

SADDR1 is the mapped IPv4 address for VA1 in VA4's address space.

In the reverse direction, TS VA1 using IPv6 would communicate with TS VA4 with the translation sequence:

```
VNET1:VA1,SIR:(VNI2:VA4)      // Endpoint at VA1
VNET1:VA1,L4:(VNI2:VA4)      // On the wire
SADDR1,VA4                    // Received at VA4
```

A.14 Scenario 14: Non-local address to tenant system

A tenant system may have a global address that is non-local to the network. A host on the Internet or a tenant system may send packet to this address. The packet is forwarded by some means to a gateway or other ILA node (tunneling could be used to accomplish this). An ILA

node can create an ILA address for this using a non-local address identifier.

For a node sending to a non-local address that is an address of task T2, the ILA translation sequence would be:

```
SADDR,A           // Endpoint at a host
SADDR,L3:X        // On the wire
SADDR,A           // Received by TS 2
```

Note that A is the non-local address, and X is an identifier that maps to the non-local address.

Appendix B: unique identifier generation

The unique identifier type of ILA identifiers can address 2^{60} objects (assuming the typed identifier format is used as described in section 4). This appendix describes some method to perform allocation of identifiers for objects to avoid duplicated identifiers being allocated.

B.1 Globally unique identifiers method

For small to moderate sized deployments the technique for creating locally assigned global identifiers described in [RFC4193] could be used. In this technique a SHA-1 digest of the time of day in NTP format and an EUI-64 identifier of the local host is performed. N bits of the result are used as the globally unique identifier.

The probability that two or more of these IDs will collide can be approximated using the formula:

$$P = 1 - \exp(-N^2 / 2^{L+1})$$

where P is the probability of collision, N is the number of identifiers, and L is the length of an identifier.

The following table shows the probability of a collision for a range of identifiers using a 60-bit length.

Identifiers	Probability of Collision
1000	$4.3368 \cdot 10^{-13}$
10000	$4.3368 \cdot 10^{-11}$
100000	$4.3368 \cdot 10^{-09}$
1000000	$4.3368 \cdot 10^{-07}$

Note that locally unique identifiers may be ephemeral, for instance a task may only exist for a few seconds. This should be considered when

determining the probability of identifier collision.

B.2 Universally Unique Identifiers method

For larger deployments, hierarchical allocation may be desired. The techniques in Universally Unique Identifier (UUID) URN ([RFC4122]) can be adapted for allocating unique object identifiers in sixty bits. An identifier is split into two components: a registrar prefix and sub-identifier. The registrar prefix defines an identifier block which is managed by an agent, the sub-identifier is a unique value within the registrar block.

For instance, each host in a network could be an agent so that unique identifiers for objects could be created autonomously by the host. The identifier might be composed of a twenty-four bit host identifier followed by a thirty-six bit timestamp. Assuming that a host can allocate up to 100 identifiers per second, this allows about 21.8 years before wrap around.

```

/* LUI identifier with host registrar and timestamp */
|3 bits|1|      24 bits      |              36 bits              |
+-----+-----+-----+-----+-----+-----+-----+-----+
| 0x1  |C| Host identifier |              Timestamp Identifier   |
+-----+-----+-----+-----+-----+-----+-----+-----+

```

Appendix C: Datacenter task virtualization

This section describes some details to apply ILA to virtualizing tasks in a datacenter.

C.1 Address per task

Managing the port number space for services within a datacenter is a nontrivial problem. When a service task is created, it may run on arbitrary hosts. The typical scenario is that the task will be started on some machine and will be assigned a port number for its service. The port number must be chosen dynamically to not conflict with any other port numbers already assigned to tasks on the same machine (possibly even other instances of the same service). A canonical name for the service is entered into a database with the host address and assigned port. When a client wishes to connect to the service, it queries the database with the service name to get both the address of an instance as well as its port number. Note that DNS is not adequate for the service lookup since it does not provide port numbers.

With ILA, each service task can be assigned its own IPv6 address and therefore will logically be assigned the full port space for that

address. This a dramatic simplification since each service can now use a publicly known port number that does not need to be unique between services or instances. A client can perform a lookup on the service name to get an IP address of an instance and then connect to that address using a well known port number. In this case, DNS is sufficient for directing clients to instances of a service.

C.2 Job scheduling

In the usual datacenter model, jobs are scheduled to run as tasks on some number of machines. A distributed job scheduler provides the scheduling which may entail considerable complexity since jobs will often have a variety of resource constraints. The scheduler takes these constraints into account while trying to maximize utility of the datacenter in terms of utilization, cost, latency, etc. Datacenter jobs do not typically run in virtual machines (VMs), but may run within containers. Containers are mechanisms that provide resource isolation between tasks running on the same host OS. These resources can include CPU, disk, memory, and networking.

A fundamental problem arises in that once a task for a job is scheduled on a machine, it often needs to run to completion. If the scheduler needs to schedule a higher priority job or change resource allocations, there may be little recourse but to kill tasks and restart them on a different machine. In killing a task, progress is lost which results in increased latency and wasted CPU cycles. Some tasks may checkpoint progress to minimize the amount of progress lost, but this is not a very transparent or general solution.

An alternative approach is to allow transparent job migration. The scheduler may migrate running jobs from one machine to another.

C.3 Task migration

Under the orchestration of the job scheduler, the steps to migrate a job may be:

- 1) Stop running tasks for the job.
- 2) Package the runtime state of the job. The runtime state is derived from the containers for the jobs.
- 3) Send the runtime state of the job to the new machine where the job is to run.
- 4) Instantiate the job's state on the new machine.
- 5) Start the tasks for the job continuing from the point at which it was stopped.

This model is similar to virtual machine (VM) migration except that the runtime state is typically much less data-- just task state as

opposed to a full OS image. Task state may be compressed to reduce latency in migration.

C.3.1 Address migration

ILA facilitates address (specifically identifier address) migration between hosts as part of task migration or for other purposes. The steps in migrating an address might be:

- 1) Configure address on the target host.
- 2) Suspend use of the address on the old host. This includes handling established connections (see next section). A state may be established to drop packets or send ICMP destination unreachable when packets to the migrated address are received.
- 3) Update the identifier to locator mapping database. Depending on the control plane implementation this may include pushing the new mapping to hosts.
- 4) Communicating hosts will learn of the new mapping via a control plane either by participation in a protocol for mapping propagation or by the ILA resolution protocol.

C.3.2 Connection migration

When a task and its addresses are migrated between machines, the disposition of existing TCP connections needs to be considered.

The simplest course of action is to drop TCP connections across a migration. Since migrations should be relatively rare events, it is conceivable that TCP connections could be automatically closed in the network stack during a migration event. If the applications running are known to handle this gracefully (i.e. reopen dropped connections) then this may be viable.

For seamless migration, open connections may be migrated between hosts. Migration of these entails pausing the connection, packaging connection state and sending to target, instantiating connection state in the peer stack, and restarting the connection. From the time the connection is paused to the time it is running again in the new stack, packets received for the connection should be silently dropped. For some period of time, the old stack will need to keep a record of the migrated connection. If it receives a packet, it should either silently drop the packet or forward it to the new location.

Appendix D: Mobility in wireless networks

ILA can be used in public wireless networks to provide a solution for mobility.

Devices in a carrier network are referred to as User Equipment (UE) and can include smart phones, automobiles, and other IoT devices. UEs attach to provider network at base stations (eNodeB in carrier terminology). As the device moves, it may change it's point of attachment to a geographically close base station. A cellular network is composed of cells each of which has an eNodeB.

A node may change cells several times over a time period. In order to provide seamless communications it is desirable that the existing connections of the device are preserved. ILA provides for this by assigning SIR addresses to UEs and deploying ILA routers in the network infrastructure.

In a canonical architecture each base station (eNodeB) would have an ILA router, and there would be a number of ILA routers that serve as gateways between a provider's network and the Internet. When a host on the Internet sends to a UE's SIR address, a gateway ILA router will translate the address. The locator addresses the base station that is the current point of attachment. At the base station ILA router, the destination is transformed back to a SIR address and delivered to a UE. A similar process can happen when two UEs in the network communicate.

The wireless network use case is conceptually similar to network virtualization. In both scenarios, nodes have a point of attachment and can move to other points of attachment. The difference is that in network virtualization it is virtual machines that are mobile, in wireless networks it is real devices.

The wireless use case has some unique properties:

- o These are often public networks so that privacy is a consideration. It is likely that devices may have many addresses assigned to promote privacy. Strong privacy addresses may be needed [ADDRPRIV].
- o A single device might have many identifiers assigned to it. When a device moves, all of the identifiers must change to map to the same locator.
- o Devices move on their own accord so that mobility is unpredictable.

- o There are mostly real humans using devices so that human identity and exposing geo location are concerns.

Author's Address

Tom Herbert
Quantonium
4701 Patrick Henry Dr.
Santa Clara, CA

EMail: tom@herbertland.com

Petr Lapukhov
1 Hacker Way
Menlo Parck, CA

EMail: petr@fb.com

INTERNET-DRAFT
Intended Status: Informational
Expires: August 2018

T. Herbert
Quantonium

February 20, 2018

Privacy in IPv6 Network Prefix Assignment
draft-herbert-ipv6-prefix-address-privacy-00

Abstract

This document discusses privacy concerns around network prefix assignment in IPv6. It evaluates the privacy threat, proposes a set of ideal criteria for strong privacy, and suggests solutions to achieve a high degree of privacy in addressing.

Status of this Memo

This Internet-Draft is submitted to IETF in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

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."

The list of current Internet-Drafts can be accessed at
<http://www.ietf.org/lid-abstracts.html>

The list of Internet-Draft Shadow Directories can be accessed at
<http://www.ietf.org/shadow.html>

Copyright and License Notice

Copyright (c) 2018 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

(<http://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 Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1	Introduction	3
2	The privacy concern	3
3	Prior work	4
3.1	SLAAC and DHCPv6-PD	4
3.2	Privacy addresses	4
3.3	Privacy in IPv6 address generation mechanisms	5
3.4	Host address availability recommendations	6
3.5	IPWAVE	6
4	Practical effects	7
4.1	Mobile networks	7
4.2	Connected cars	7
4.3	Privacy implications of NAT	8
4.4	Exploit to defeat prefix rotation	8
5	Criteria for strong privacy	9
6	Identifier/locator split solution	10
6.1	Overview	10
6.2	Scaling identifier/locator address assignment	11
6.2.1	Scaling the amount of mapping state	11
6.2.1.1	Hybrid address assignment	11
6.2.1.2	Hidden aggregation	11
6.2.2	Scaling bulk address assignment	12
6.2.2.1	Bulk assignment using DHCPv6	12
6.2.2.2	Hidden aggregation assignment	13
6.2.3	Practicality of hidden aggregation methods	13
6.3	Law enforcement considerations	14
7	Security considerations	15
8	References	16
8.1	Normative References	16
8.2	Informative References	16
9	Acknowledgments	17
	Authors' Addresses	17

1 Introduction

This document discusses privacy of network prefix assignment in IPv6.

A common address assignment method is for a network to assign prefixes to devices. SLAAC and DHCP-PD are two mechanisms for doing this. In the common case of a /64 assignment (as in SLAAC) the device generates IIDs (interface identifiers) to create individual addresses within an assigned prefix. While significant effort has gone into IID generation techniques to protect privacy ([RFC4941], [RFC7721]), the privacy aspects of the prefix itself have not been fully examined.

This document is focused on privacy within the network layer and specifically with privacy in addressing. There are many other privacy issues that arise from persistent identifiers used in higher (and lower) protocol layers (MAC address, session IDs, certificates, etc.). Discussion of these are out of scope for this document, however it is clear that to achieve a level of privacy that users deserve all layers will need to be considered.

2 The privacy concern

In the original IPv6 addressing model, subnets (links) were assigned a sixty-four bit prefix [RFC4291]. Hosts in the subnet would then generate IIDs that are combined with the subnet prefix to create IPv6 addresses. This model was subsequently extended to assign network prefixes, such as /64s, to general purpose hosts ([RFC3314], [RFC7934]).

When a prefix is assigned to an end host, the prefix becomes an identifier for the host. So, if two such addresses have the same prefix (i.e. same upper sixty-four bits) then they can be assumed to refer to the same host. The IID portion of the addresses (lower sixty-four bits) are immaterial in this inference, so IID generation techniques don't affect the ability to make correlations.

The fact that two addresses can be correlated to be from the same host implies the privacy concern. If an attacker knows that a network provider assigns /64 prefixes to end hosts, as is common in mobile networks, then it can deduce that two addresses in the provider prefix sharing the same sixty-four bit prefix refer to the same host. This correlation can be made between addresses of different flows independently of IIDs in those addresses. Furthermore, with a little more information (see Section 4.3), an attacker may not only deduce two addresses refer to the same end host, but also may be able to discover the identities of individuals in communications.

3 Prior work

Several RFCs describe prefix assignment mechanisms and the privacy and security considerations for them.

3.1 SLAAC and DHCPv6-PD

SLAAC [RFC4862] and DHCPv6-PD [RFC3633] are mechanisms to assign network prefixes to devices. Their respective specifications do not address privacy issues of prefix assignment. Security considerations are focused on the mechanisms.

3.2 Privacy addresses

[RFC4941] addresses issues with persistent identifiers in IPv6. It describes the risks of extended use of the same identifier, and recommends using random interface identifiers and changing addresses periodically to deter inferences to reveal identify, location, or other privacy sensitive attributes of parties in communication. Addresses created by following RFC4941 recommendations are often called "privacy addresses".

RFC4941 is mostly concerned with privacy and security aspects of IID generation. It mentions the problem of privacy of network prefixes in passing:

Although it might appear that changing an address regularly in such environments would be desirable to lessen privacy concerns, it should be noted that the network prefix portion of an address also serves as a constant identifier. All nodes at, say, a home, would have the same network prefix, which identifies the topological location of those nodes. This has implications for privacy, though not at the same granularity as the concern that this document addresses. Specifically, all nodes within a home could be grouped together for the purposes of collecting information. If the network contains a very small number of nodes, say, just one, changing just the interface identifier will not enhance privacy at all, since the prefix serves as a constant identifier.

Nevertheless, it's reasonable that some of the recommendations could be extrapolated to apply to prefix assignment for providing privacy. For instance, RFC4941 suggests to periodically do address rotation by generating a new IID. Conceivably, a node could periodically request a new network prefix via SLAAC. The new prefix would be randomized so that no correlation can be drawn between it and the old prefix.

As for the frequency of changing addresses, RFC4941 states:

having large numbers of clients change their address on a daily or weekly basis is likely to be sufficient to alleviate most privacy concerns.

The statement is neither normative nor quantified. Intuitively, one might assume that a higher frequency of address rotation reduces the probability of privacy being compromised. However, other than the case where a different address is used for each flow (see below), there is no known way to quantify the relationship between frequency of changing addresses and privacy provided to users.

A second concern with recommendations of RFC4941 is that it is was written eleven years ago. The sophistication and capabilities of attackers have increased substantially, so recommendations, such as changing addresses on a daily or weekly basis, may no longer be sufficient even if they were eleven years ago.

Presumably, one could try to achieve a high degree of privacy by changing addresses at a high frequency (every few seconds for instance). The effect on privacy is still unquantifiable, however there is another problem in the disruption caused by changing addresses. An address change would require termination of existing flows, so a high frequency of address rotation would constantly thrash connections. A potential mitigation would be to allow a host to retain network prefixes for which it's still using for flows; however, managing that would be cumbersome and likely wouldn't scale since hosts could accumulate many prefixes over time.

The postulated exploit described in Section 4.4 would defeat the privacy protection of any frequency of address rotation except for the case where a different address is used per flow.

3.3 Privacy in IPv6 address generation mechanisms

[RFC7721] mainly focuses on security and privacy considerations for IID generation. The concern around privacy in network prefix assignment is raised:

As [RFC4941] notes, if a very small number of nodes (say, only one) use a particular prefix for an extended period of time, the prefix itself can be used to correlate the host's activities regardless of how the IID is generated. For example, [RFC3314] recommends that prefixes be uniquely assigned to mobile handsets where IPv6 is used within General Packet Radio Service (GPRS). In cases where this advice is followed and prefixes persist for extended periods of time (or get reassigned to the same handsets

whenever those handsets reconnect to the same network router), hosts' activities could be correlatable for longer periods than the analysis below would suggest.

RFC7721 does not suggest any requirements or guidelines for privacy in network prefixes. Similar to RFC4941, RFC7721 frames the problem with an unquantified description as using a prefix for "extended periods of time".

Note that RFC7721 points out that mobile handsets are often assigned a single prefix. In this case, there is one to one relationship between a prefix and device. For a personal device, such as a smart phone or tablet, there would then be a one to one relationship between a prefix and an individual user.

3.4 Host address availability recommendations

[RFC7934] recommends that general-purpose hosts are assigned multiple globally IPv6 addresses when they attach. RFC7934 advocates prefix assignment and /64 assignment with SLAAC in particular.

RFC7934 includes a section on host tracking (Section 9.1 of RFC7934), however this section focuses on facilitating tracking of hosts in provider networks to satisfy legal requirements.

From RFC7934:

Using SLAAC with a dedicated /64 prefix for each host simplifies tracking, as it does not require logging every address formed by the host

RFC7934 references RFC4941, but does not otherwise address issues with privacy in prefix assignment.

3.5 IPWAVE

[IPWAVE] provides the problem statement for IPWAVE. The issue of address tracking is raised in the Security Considerations section. From the draft:

To prevent an adversary from tracking a vehicle by with its MAC address or IPv6 address, each vehicle should periodically update its MAC address and the corresponding IPv6 address as suggested in [RFC4086][RFC4941]. Such an update of the MAC and IPv6 addresses should not interrupt the communications between a vehicle and an RSU.

As in the RFCs cited above, the draft suggests that addresses should

be changed periodically, however there is no guidance as to what an acceptable frequency of change is to prevent tracking. It is noteworthy that address change is expected to not interrupt communications.

4 Practical effects

This section discusses the current characteristics and effects on privacy in network prefix assignment to hosts.

4.1 Mobile networks

Privacy in prefix addressing is of particular concern in mobile networks. It is often the case that UEs (devices such as smart phones) are assigned a unique /64 prefix that is not shared with other devices. As pointed out by RFC4941 and RFC7721, these network prefixes allow the device to be tracked through correlations. For personal devices, such as smart phones or tablets, correlations on IP addresses could be used to infer user identities in communication. The correlation to a user may require additional information that might be relatively easy to acquire as demonstrated by the exploit described in section 4.4.

Most mobile providers follow the advice of [RFC3314] and assign single a /64 to each device. They may implement a method to force a device to periodically request a new /64 assignment.

A sample implementation in a mobile network could assign a /64 prefix to each IPv6 PDN, and the same prefix is retained for Idle to Active to Idle transitions for the duration of the PDN session. If the UE is idle without transmitting/receiving any packets, the PDN session is dropped when the Idle Timer expires (e.g. 2 hours) and the prefix allocation is released. So in this case the minimum amount of time between addresses change is 2 hrs., but a device could keep its prefix allocation indefinitely as long as the device remains active.

4.2 Connected cars

Connected cars are projected to become ubiquitous over the next decade. By some estimates there will be 381 million connected cars on the road by 2020, and by 2025 all new cars manufactured will be connected. Today many vehicles are already connected to the Internet via 4G LTE, and in the future they will connect using 5G, WiFi, DSRC or other radio technologies. In-vehicle networks connect sensors, displays, navigation, entertainment, as well as personal devices being used by passengers.

Privacy in such a network is potentially a more difficult problem

since there are two independent parties that are involved in address assignment. The vehicle as a mobile node must be assigned addresses by the mobile network, and in turn the vehicle delegates addresses to devices attached to the vehicle network.

A /64 prefix could be assigned to vehicles which is a common mobile network assignment. Devices attached to the vehicle network are delegated IPv6 address within the prefix assigned to the vehicle. This results in all the attached devices sharing fate with respect to privacy. For instance, if an attacker is able to determine the location of just one device with an assigned prefix, then it can infer the location of all devices that share the same prefix. If identity of a user can be separately surmised, this raises the prospect that location of individuals can be tracked.

Periodically changing prefixes in this environment is problematic. As described in Section 3.2, a prefix change is potentially disruptive to communications as this results in an address change for each attached device. In the case of a vehicle network, the attached devices and applications they are running may be very heterogeneous such that their response and recovery for an address change may vary significantly. For instance, a laptop might attach to a vehicle network. A laptop is not normally considered a "mobile device" like a smart phone and many applications they might run don't assume addresses constantly change. Periodically changing addresses for privacy benefit may wreak havoc on such applications.

4.3 Privacy implications of NAT

Network Address Translation (NAT) is a method of remapping one IP address space into another by modifying addresses in the IP header of packets while they are in transit across a routing node. NAT has been extensively deployed to allow hosts that are assigned IPv4 private addresses [RFC1918] to communicate with hosts in the global Internet. NAT has been used to extend the usefulness of IPv4 in the face of address depletion.

A side effect of NAT (possibly accidental) is that NAT modifies addresses such that it obfuscates the identity of the source host behind a NAT. With a significant population of users sharing a pool of NAT addresses, an external observer can draw little correlation based on addresses between flows that have gone through a NAT device. The result is that NAT provides strong privacy in addressing. NAT use is of particular concern to law enforcement since its privacy characteristics complicate criminal investigation [EUROPOL].

4.4 Exploit to defeat prefix rotation

As mentioned in a Section 3.2, one might try to provide privacy in addressing by changing addresses with a high frequency. The following exploit is postulated as a way to defeat the privacy goals of periodic address rotation at any frequency except when a different address is used for each connection.

The exploit is:

- o An attacker creates an "always connected" app that provides some seemingly benign service and users download the app.
- o The app includes some sort of persistent identity. For instance, this could be an account login.
- o The backend server for the app logs the identity and IP address of a user each time they connect.
- o When an address change happens, existing connections on the user device are disconnected. The app will receive a notification and immediately attempt to reconnect using the new source address.
- o The backend server will see the new connection and log the new IP address as being associated to the user. Thus, the server has a real-time record of users and the IP address they are using.
- o The attacker intercepts packets at some point in the Internet. The addresses in the captured packets can be time correlated with the server database to deduce identities of parties in communications that are unrelated to the app.

5 Criteria for strong privacy

A set of "ideal" criteria for strong privacy in addressing can be established. These criteria are intended to be specific, such that when applied to a solution the amount of information that can be inferred by correlating addresses is quantifiable.

The ideal criteria for IPv6 addresses that provide strong privacy are:

- o Addresses are composed of a global routing prefix and a suffix that is internal to an organization or provider. This is the same property for IP addresses [RFC4291].
- o The registry and organization of an address can be determined by the network prefix. This is true for any global address. The organizational bits in the address should have minimal hierarchy to prevent inference. It might be reasonable to have an internal

prefix that divides identifiers based on broad geographic regions, but detailed information such as location, department in an enterprise, or device type should not be encoded in a globally visible address.

- o Given two addresses and no other information, the desired properties of correlating them are:
 - o It can be inferred if they belong to the same organization and registry. This is true for any two global IP addresses.
 - o It may be inferred that they belong to the same broad grouping, such as a geographic region, if the information is encoded in the organizational bits of the address.
 - o No other correlation can be established. It cannot be inferred that the IP addresses address the same node, the addressed nodes reside in the same subnet, rack, or department, or that the nodes for the two addresses have any geographic proximity to one another.

Note that if NAT is deployed with a sufficiently large population of users sharing a pool of IP addresses then these criteria are met. Thus NAT can be considered a baseline for strong privacy in addressing.

6 Identifier/locator split solution

This section proposes using identifier/locator split to meet the strong privacy criteria for addressing in IPv6.

6.1 Overview

Identifier/locator split separates the notions of location and identity in IP addresses. Identifier addresses are addresses that don't contain topological information for routing within a network. Nodes are assigned identifier addresses that can be used as endpoints in communications. Locator addresses indicate the topological location of a logical node. In order to forward a packet to a destination with an identifier address, an ingress node for a network maps an identifier address to a locator address. A network overlay method is used to forward the packet to the location in the network of the logical or mobile node.

Since identifier addresses are non-topological they don't require any hierarchy in address assignment beyond the global network prefix. Therefore the network can randomly generate identifier addresses within a portion of the address in a space of at least sixty-four

bits.

Strong privacy in addressing can be achieved by using a different randomly generated identifier address for each flow. Conceptually, this would entail that the network creates and assigns a unique and untrackable address to a host for every flow created by a host. Some suggestions for scaling this technique are discussed below.

Note that this technique parallels what NAT does in that NAT effectively creates a different source address per connection. Unlike NAT however, address assignments in identifier/locator split are stateless in the network and transparent to the end points.

6.2 Scaling identifier/locator address assignment

Assigning an address per connection is a potential scaling problem on two accounts:

- o The amount of state needed in the mapping system is significant.
- o Bulk host address assignment is inefficient.

6.2.1 Scaling the amount of mapping state

The amount of state necessary to assign each flow its own unique source IP address is equivalent, or at least proportional, to the amount of state needed for NAT-- basically this is one state element for every connection in the network. So in one sense this solution should scale as well as NAT has.

6.2.1.1 Hybrid address assignment

Not all communications might require strong privacy, so it is conceivable that a hybrid approach to address assignment might be taken. A network might assign prefixes for use with communications that are not privacy sensitive, and may assign singleton addresses that meet strong privacy criteria for privacy sensitive communications. Assuming that most communications don't need strong privacy this could reduce the amount of state needed in the mapping system considerably. The decision as to whether strong privacy is required for a communication would be made by the user or application.

6.2.1.2 Hidden aggregation

A possible solution to reduce state is to make addresses aggregable, but use an aggregation method that is known only by the network provider and hidden to the rest of the world. The network could use a

reversible hash or encryption function to create addresses.

The input to an address generation function includes a device identifier, a secret key, and a generation index.

The function may have the form:

$$\text{Address} = \text{Func}(\text{key}, \text{dev_ident}, \text{gen})$$

Where "key" is secret to network, "dev_ident" is a network internal identifier for a device (roughly equivalent to "identity" in IDEAS), and "gen" is generation number 0,1,2,... N. The generation value is changed for each invocation to create different addresses for assignment to a device.

When a network ingress node is forwarded a packet it performs the inverse function on an address.

The inverse function has the form:

$$(\text{dev_ident}, \text{gen}) = \text{FuncInv}(\text{key}, \text{Address})$$

The returned dev_ident value is used as the identifier in the mapping lookup for a locator address. In this manner, the network can generate many addresses to assign to a device where they all share a single entry in the mapping system.

6.2.2 Scaling bulk address assignment

Assigning multiple addresses without aggregation is difficult to scale. Each address would need to be individually specified in an assignment sent to a host.

6.2.2.1 Bulk assignment using DHCPv6

DHCPv6 might allow bulk singleton address assignment. As stated in [RFC7934]:

Most DHCPv6 clients only ask for one non-temporary address, but the protocol allows requesting multiple temporary and even multiple non-temporary addresses, and the server could choose to provide multiple addresses. It is also technically possible for a client to request additional addresses using a different DHCP Unique Identifier (DUID), though the DHCPv6 specification implies that this is not expected behavior ([RFC3315], Section 9). The DHCPv6 server will decide whether to grant or reject the request based on information about the client, including its DUID, MAC address, and more. The maximum number of IPv6

addresses that can be provided in a single DHCPv6 packet, given a typical MTU of 1500 bytes or smaller, is approximately 30.

6.2.2.2 Hidden aggregation assignment

By extending the concept of hidden aggregation assignment (section 6.2.1.2), it is conceptually possible that a host could work in concert with the network to generate addresses that meet strong privacy criteria. In this method, a host autonomously generates addresses as needed. The network, but no one outside the network, is then able aggregate the addresses as belonging to the device.

End hosts are generally considered untrusted nodes by the network, so they cannot be given access to the network secret key used for the address generation function. Public key encryption might be used.

A host may perform an encryption function to generate addresses:

```
Address = Encrypt(pub_key, dev_inet, gen)
```

Where "pub_key" is a public key for the network, "dev_inet" is a network identifier for the device and is visible to the device (so it may be leaked). "gen" is a generation number 0,1,2,... N. The generation value is changed for each invocation to create different addresses.

When a network ingress node is forwarded a packet it decrypts an address using the network private key.

The decryption function has the form:

```
(dev_inet, gen) = decrypt(priv_key, Address)
```

Where "priv_key" is the secret private key of the network associated with the public key. The returned dev_inet value is used as the identifier in the mapping lookup for a locator address.

Note that this method would require an new address assignment protocol.

6.2.3 Practicality of hidden aggregation methods

The premise of hidden aggregation is that only trusted devices in the network are able decode the aggregation hidden within IPv6 addresses. This implies that the network must keep secrets about the process. In the above examples, the secrets are keys used in the hash or encryption. The security of the key is then paramount, so techniques for key management, rotation, and using different key sets for

obfuscation are pertinent.

To perform a mapping lookup a node must apply the inverse address generation function to map addresses to locators. This lookup would occur in the critical data path so performance is important. Encryption and hashing are notoriously time consuming and computationally complex functions.

Some possible mitigating factors for performance impact are:

- o The input to address generation functions is a small amount of data and has fixed size. The input is a key (presumably 128 or 256 bits), part of all of an IPv6 address (128 bits), and a generation number (sixteen to twenty-four bits should work).
- o Given that the input is fixed size, specialized hardware might be used to optimize performance of the inverse address generation function. For instance, modern CPUs include instructions to perform crypto [AES-NI]. Since the keys used in these functions are secret to the network and there are relatively few of them, they might be preloaded into a crypto engine to reduce setup costs.
- o The output of an inverse address generation function is cacheable. A cache on a device could contain address to locator mappings. When the inverse function and lookup on dev_ident are performed, a mapping of address to the discovered locator could be created in the cache. The device could then map addresses in subsequent packets sent on the same flow to the proper locator by looking up the address in the cache.

6.3 Law enforcement considerations

This section discusses law enforcement considerations for host tracking when using an identifier/locator split solution for strong privacy. NAT is used as a reference point for discussion.

There are two sub-problems expressed by law enforcement about NAT [EUROPOL]:

- 1) It is difficult to map a NAT address and port back to a user.
- 2) Many Internet servers do not log the client source port of connections.

The first problem is one of maintaining a log of NAT mappings. If the log contains the inner address, outer address and port, and timestamp when the NAT mapping was created-- then given the log and a NATed

packet, the original sender can be revealed. Note that NAT logs are kept internal to the provider network, and securing them is the responsibility of the provider. The same model can be applied to identifier/locator split where the infrastructure keeps a log of identifier to locator mappings and a timestamp for when they were created.

In the second problem, the source port is needed to be logged in servers in order correlate a flow to an entry in the NAT logs of a provider. The source port is relevant to a NAT mapping; however, in identifier/locator split it's not since identification of a host node contained with an address. Therefore the client source port is not required for tracking users in an identifier/locator solution.

7 Security considerations

The subject of this draft is privacy assigning network prefixes. Implicit to this is that any address assignment technique requires security on the parties entities involved.

In the identifier/locator split the mapping of identifier to locator is privacy sensitive information. The locator may very well imply the geo location of a device. As such, it is recommended that locators that might contain accurate location information are strictly contained within a trusted infrastructure.

In mobile environments, it is natural to group identifiers (addresses) together that have the same attributes [IDGROUP]. For instance, if as in section 6.1 a different source address is used for each flow, all of the addresses assigned to a device form a group. When the device moves, all of the addresses move with it; this can be efficiently implemented as single operation on the mapping system. The group information is thus privacy sensitive information that must be secured by the infrastructure to prevent use of the information to make inferences of identity similar to /64 assignment.

Hidden aggregation is a means of grouping identifiers together similar to the above description. The secret keys used in these algorithms are thus critical information that must be kept secure. Security by obscurity should be avoided here, divulging the algorithm used to generate addresses should not reduce security or privacy.

End hosts must implement appropriate security to ensure privacy. For instance, if an address is assigned per flow as described in Section 6.2, applications must be isolated from one another so that they cannot infer addresses or privacy properties of other applications running within the same system. Also, if a host is completely compromised then that fact should not impact the privacy and security

of other hosts and applications within a network.

8 References

8.1 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>>.
- [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>>.

8.2 Informative References

- [RFC4941] Narten, T., Draves, R., and S. Krishnan, "Privacy Extensions for Stateless Address Autoconfiguration in IPv6", RFC 4941, DOI 10.17487/RFC4941, September 2007, <<https://www.rfc-editor.org/info/rfc4941>>.
- [RFC7721] Cooper, A., Gont, F., and D. Thaler, "Security and Privacy Considerations for IPv6 Address Generation Mechanisms", RFC 7721, DOI 10.17487/RFC7721, March 2016, <<https://www.rfc-editor.org/info/rfc7721>>.
- [RFC4291] Hinden, R. and S. Deering, "IP Version 6 Addressing Architecture", RFC 4291, DOI 10.17487/RFC4291, February 2006, <<https://www.rfc-editor.org/info/rfc4291>>.
- [RFC3314] Wasserman, M., Ed., "Recommendations for IPv6 in Third Generation Partnership Project (3GPP) Standards", RFC 3314, DOI 10.17487/RFC3314, September 2002, <<https://www.rfc-editor.org/info/rfc3314>>.
- [RFC7934] Colitti, L., Cerf, V., Cheshire, S., and D. Schinazi, "Host Address Availability Recommendations", BCP 204, RFC 7934, DOI 10.17487/RFC7934, July 2016, <<https://www.rfc-editor.org/info/rfc7934>>.
- [RFC4862] Thomson, S., Narten, T., and T. Jinmei, "IPv6 Stateless Address Autoconfiguration", RFC 4862, DOI 10.17487/RFC4862, September 2007, <<https://www.rfc-editor.org/info/rfc4862>>.

- [RFC3633] Troan, O. and R. Droms, "IPv6 Prefix Options for Dynamic Host Configuration Protocol (DHCP) version 6", RFC 3633, DOI 10.17487/RFC3633, December 2003, <<https://www.rfc-editor.org/info/rfc3633>>.
- [RFC3315] Droms, R., Ed., Bound, J., Volz, B., Lemon, T., Perkins, C., and M. Carney, "Dynamic Host Configuration Protocol for IPv6 (DHCPv6)", RFC 3315, DOI 10.17487/RFC3315, July 2003, <<https://www.rfc-editor.org/info/rfc3315>>.
- [RFC1918] Rekhter, Y., Moskowitz, B., Karrenberg, D., de Groot, G., and E. Lear, "Address Allocation for Private Internets", BCP 5, RFC 1918, DOI 10.17487/RFC1918, February 1996, <<https://www.rfc-editor.org/info/rfc1918>>.
- [RFC3315] Droms, R., Ed., Bound, J., Volz, B., Lemon, T., Perkins, C., and M. Carney, "Dynamic Host Configuration Protocol for IPv6 (DHCPv6)", RFC 3315, DOI 10.17487/RFC3315, July 2003, <<https://www.rfc-editor.org/info/rfc3315>>.
- [EUROPOL] EUROPOL/EC3 to delegations of the Council of the European Union, "Carrier-Grade Network Address Translation (CGN) and the Going Dark Problem", January 2017
- [AES-NI] Gueron, S., "Intel Advanced Encryption Standard (AES) New Instructions", <<https://software.intel.com/sites/default/files/article/165683/aes-wp-2012-09-22-v01.pdf>>
- [IDGROUP] Herbert, T., "Identifier groups", draft-herbert-idgroups-00

9 Acknowledgments

The author would like to thank Robert Moskowitz for insightful comments and contributions to this draft.

Authors' Addresses

Tom Herbert
Quantonium
Santa Clara, CA
USA

Email: tom@quantonium.net

Inter-Domain Routing
Internet-Draft
Intended status: Standards Track
Expires: May 4, 2017

P. Lapukhov
Facebook
October 31, 2016

Use of BGP for dissemination of ILA mapping information
draft-lapukhov-bgp-ila-afi-02

Abstract

Identifier-Locator Addressing [I-D.herbert-nvo3-ila] relies on splitting the 128-bit IPv6 address into identifier and locator parts to implement identifier mobility, and network virtualization. This document proposes a method for distributing the identifier to locator mapping information using Multiprotocol Extensions for BGP-4 [RFC4760].

Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

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 <http://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 May 4, 2017.

Copyright Notice

Copyright (c) 2016 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

(<http://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 Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	2
2. BGP ILA AFI	3
3. Capability Advertisement	3
4. Disseminating Identifier-Locator mapping information	3
4.1. Advertising ILA mapping information	3
4.2. Withdrawing ILA mapping information	4
5. Interpreting the mapping information	4
5.1. Unicast SAFI	4
5.2. Multicast SAFI	5
6. Inter-domain mapping exchange	5
7. BGP Next-Hop attribute handling with ILA	6
8. Use of Add-Paths extension with ILA AFI	7
9. IANA Considerations	7
10. Manageability Considerations	7
11. Security Considerations	8
12. Acknowledgements	8
13. References	8
13.1. Normative References	8
13.2. Informative References	8
Author's Address	9

1. Introduction

Under the ILA proposal, IPv6 address is split in 64-bit identifier (lower address bits) and locator (higher address bits) portions. The locator part is determined dynamically from a mapping table that maintains associations between the location-independent identifiers and topologically significant locators. The hosts that collectively implement and maintain such mappings are referred to as "ILA domain" in this document. The ILA domain has a globally unique 64-bit SIR (Standard Identifier Representation) prefix assigned to it (see [I-D.herbert-nvo3-ila] for more details on SIR prefix use).

This document proposes a new address family identifier (AFI) for the purpose of disseminating the locator-identifier mappings among the nodes participating in the ILA domain. For example, this extension could be used to propagate the mappings from ILA hosts to ILA routers and allow the routers to perform their function (see

[I-D.herbert-nvo3-ila] for definition of the "ILA router" functions). Additional information that provides more detailed examples of deployment scenarios using BGP could be found in [I-D.lapukhov-ila-deployment].

2. BGP ILA AFI

This document introduces a new AFI known as a "Identifier-Locator Addressing AFI" (ILA AFI) with the actual value to be assigned by IANA. The purpose of this AFI is disseminating the mapping information in the ILA domain, e.g. between ILA hosts and ILA routers. This document defines the use of SAFI values of "1" (unicast) and "2" (multicast) only.

3. Capability Advertisement

A BGP speaker that wishes to exchange ILA mapping information MUST use the Multiprotocol Extensions Capability Code, as defined in [RFC4760], to advertise the corresponding AFI/SAFI pair.

4. Disseminating Identifier-Locator mapping information

4.1. Advertising ILA mapping information

For the purpose of ILA mapping encoding, the 8-octet locator field SHALL be encoded in the "next-hop address" field. The "length of the next-hop address" MUST be set to "8". The identifiers bound to the locator SHALL be encoded within the NLRI portion of MP_REACH_NLRI attribute. The NLRI portion of MP_REACH_NLRI starts with the two-octet "Length of identifiers" field, with the value being multiple of 8. The rest of the NLRI is a collection of 8-octet identifiers that are bound to the locator specified in the "next-hop address" field.

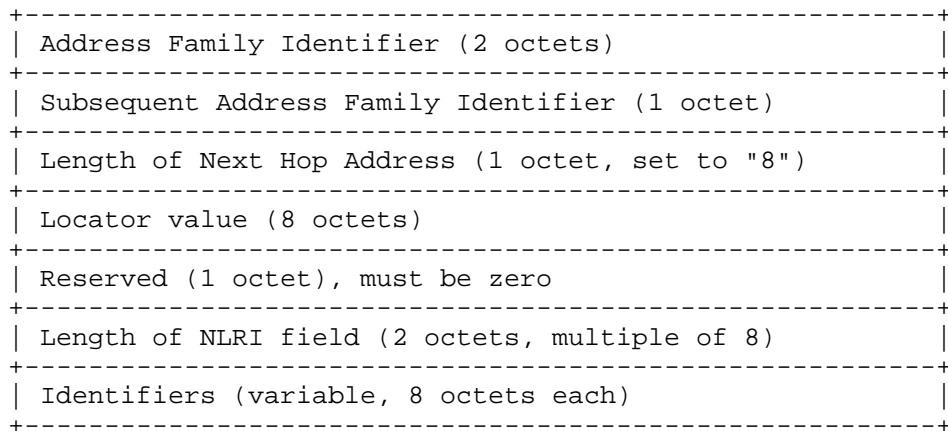


Figure 1: MP_REACH_NLRI Layout

4.2. Withdrawing ILA mapping information

Withdrawal of ILA mapping information is performed via an MP_UNREACH_NLRI attribute advertisement organized as following:

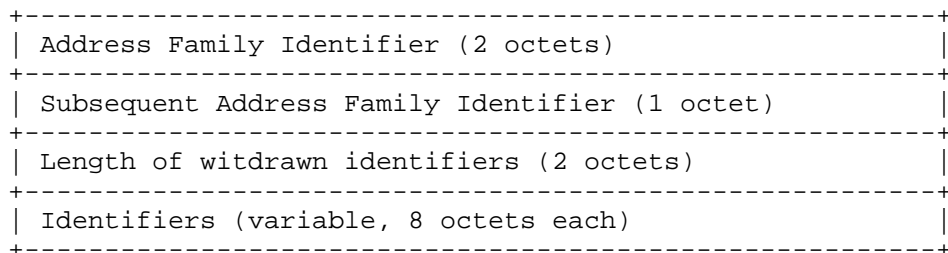


Figure 2: MP_UNREACH_NLRI Layout

5. Interpreting the mapping information

5.1. Unicast SAFI

Only the locator part of ILA address is used for packet routing, and every node that hosts an identifier is expected to have a unique /64 prefix routable within the scope of the ILA domain. The identifiers advertised under the ILA AFI are expected to be used by the data-plane implementation to perform match on a full IPv6 address and decide whether the locator portion of the address needs a re-write. It is up to the implementation to decide which full 128-bit IPv6 addresses need a rewrite, e.g. by matching on a Standard Identifier Representation (SIR) prefix as defined in [I-D.herbert-nvo3-ila].

The locator rewrite information comes from the next-hop "address" associated with the identifier. The next-hop field of MP_REACH_NLRI attribute in general is not expected to be used for any routing resolutions/lookups by the BGP process. It should primarily be used to create a rewrite rule in the data-plane forwarding table. The actual forwarding decision is then based on subsequent lookup in the forwarding table to find the next hop to send the packet to.

In some scenarios, resolving the next-hop attribute via additional lookups tables might be necessary. For example, for environments that deploy MPLS ([RFC3031]) forwarding, the locator may resolve to a label stack that is required to perform further forwarding. In this case, the ILA address rewrite will be accompanied by additional actions, such as label stack imposition. These decisions have to be made in implementation dependant fashion.

5.2. Multicast SAFI

Multicast SAFI retain the same encoding format, with a different SAFI value. For multicast packets, the RPF check process SHALL be modified for use with ILA source addresses. Specifically, source ILA IPv6 addresses with the identifier portion matching the mapping table SHALL be mapped to proper locator, prior to performing the RPF check. The ILA source addresses need to be identified by some means specific to ILA implementation, e.g. by matching on configured SIR prefixes. The ILA addresses that do not match any mapping entry SHALL be considered as failing the RPF check.

6. Inter-domain mapping exchange

The ILA mappings are only unique with an ILA domain - it is possible that different domains may re-use the same identifiers. To make identifiers globally unique, they MUST be concatenated with the SIR prefix assigned to each ILA domain. These globally unique identifiers may then be exchanged between multiple ILA domains. IANA will be requested to allocate new SAFI value called "VPN-ILA" SAFI to facilitate exchange of inter-domain ILA mappings. The MP_REACH_NLRI attributes exchanged over this SAFI will look as following:

```

+-----+
| Address Family Identifier (2 octets) |
+-----+
| Subsequent Address Family Identifier (1 octet) |
+-----+
| SIR prefix (8 octets) |
+-----+
| Length of Next Hop Address (1 octet, set to "8") |
+-----+
| Locator value (8 octets) |
+-----+
| Reserved (1 octet), must be zero |
+-----+
| Length of NLRI field (2 octets, multiple of 8) |
+-----+
| Identifiers (variable, 8 octets each) |
+-----+

```

The main difference from the Unicast/Multicast SAFI's is presence of the SIR prefix field in the announcement. Correspondingly, the MP_UNREACH_NLRI will look as following:

```

+-----+
| Address Family Identifier (2 octets) |
+-----+
| Subsequent Address Family Identifier (1 octet) |
+-----+
| SIR Prefix (8 bytes) |
+-----+
| Length of withdrawn identifiers (2 octets) |
+-----+
| Identifiers (variable, 8 octets each) |
+-----+

```

The use of domain-specific identifiers would require the ILA hosts and routers to make their ILA cache lookups based on the full 128-bit prefix.

7. BGP Next-Hop attribute handling with ILA

This document proposes that the BGP next-hop attribute value encode the locator associated with all identifiers found in MP_REACH_NLRI attribute. It is possible that an intermediate speaker may change the next-hop value. This may be required to ensure all traffic for the associated identifiers is routed through that intermediate speaker. The speaker is expected to maintain the original ILA mappings in its mapping table, and perform additional destination

address translation for the ILA packets. This way, a form of loose hop traffic engineering could be realized within an ILA domain.

It is common that BGP implementations reset the next-hop value for announcements made over eBGP sessions. Such scenario may be common between two different ILA domains.

8. Use of Add-Paths extension with ILA AFI

It could be useful to bind multiple locators to the same identifier, e.g. for the purpose of load-sharing. To make announcing multiple locators possible, the MP_REACH_NLRI and MP_REACH_NLRI attributes are extended to encode the path-identifier per [I-D.ietf-idr-add-paths], correspondingly enabling the capability for the AFI/SAFI. Specifically, the NLRI field will look as following:

```

+-----+
| Path identifier (4 octets) |
+-----+
| Length of NLRI field (2 octets, multiple of 8) |
+-----+
| Identifiers (variable, 8 octets each) |
+-----+

```

Such encoding instructs the receiver to create multiple locator entries for an identifier, differentiated by their path identifiers. Optionally, a weight could be associated with each path using the "link bandwidth" extended community defined in [I-D.ietf-idr-link-bandwidth]

9. IANA Considerations

For the purpose of this work, IANA would be asked to allocate a value for the new AFI, and the VPN-ILA SAFI.

10. Manageability Considerations

The ILA mappings distribution is likely to be done using separate infrastructure, independent from the BGP topology used for regular routing information distribution. Most likely there will be a collection of iBGP route-reflectors deployed within each ILA domain, and peering with a BGP process running on each ILA router and ILA host. More details and deployment examples could be found in [I-D.lapukhov-ila-deployment].

11. Security Considerations

This document does not introduce any changes in terms of BGP security. Defining ILA security model is outside of scope of this document.

12. Acknowledgements

The author would like to thank Doug Porter for the original idea and discussion of this proposal.

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, <<http://www.rfc-editor.org/info/rfc2119>>.
- [RFC3031] Rosen, E., Viswanathan, A., and R. Callon, "Multiprotocol Label Switching Architecture", RFC 3031, DOI 10.17487/RFC3031, January 2001, <<http://www.rfc-editor.org/info/rfc3031>>.
- [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, <<http://www.rfc-editor.org/info/rfc4271>>.
- [I-D.ietf-idr-add-paths] Walton, D., Retana, A., Chen, E., and J. Scudder, "Advertisement of Multiple Paths in BGP", draft-ietf-idr-add-paths-15 (work in progress), May 2016.
- [I-D.ietf-idr-link-bandwidth] Mohapatra, P. and R. Fernando, "BGP Link Bandwidth Extended Community", draft-ietf-idr-link-bandwidth-06 (work in progress), January 2013.

13.2. Informative References

- [RFC4760] Bates, T., Chandra, R., Katz, D., and Y. Rekhter, "Multiprotocol Extensions for BGP-4", RFC 4760, DOI 10.17487/RFC4760, January 2007, <<http://www.rfc-editor.org/info/rfc4760>>.

[I-D.herbert-nvo3-ila]

Herbert, T., "Identifier-locator addressing for IPv6",
draft-herbert-nvo3-ila-03 (work in progress), October
2016.

[I-D.lapukhov-ila-deployment]

Lapukhov, P., "Deploying Identifier-Locator Addressing
(ILA) in datacenter", draft-lapukhov-ila-deployment-00
(work in progress), March 2016.

Author's Address

Petr Lapukhov
Facebook
1 Hacker Way
Menlo Park, CA 94025
US

Email: petr@fb.com

Network Working Group
Internet-Draft
Intended status: Informational
Expires: May 4, 2017

P. Lapukhov
Facebook
October 31, 2016

Deploying Identifier-Locator Addressing (ILA) in datacenter networks
draft-lapukhov-ila-deployment-01

Abstract

Identifier-Locator Addressing architecture defined in [I-D.herbert-nvo3-ila] proposes the use of locator-identifier split in IPv6 address to realize workload mobility and more efficient use of network resources. This document describes how ILA can be implemented in datacenter using BGP as the control-plane protocol. Generally speaking, ILA could be built using different control planes, and BGP is one particular instantiation. The motivation is BGP being a well-known protocol, sufficient for small to medium size deployments, on scale of few millions of identifier to locator mappings. Defining more generic and scalable control plane variants is outside of scope of this document.

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 <http://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 May 4, 2017.

Copyright Notice

Copyright (c) 2016 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 (<http://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 Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	3
2. Terminology	3
3. ILA deployment process	5
4. Preparing the network	6
4.1. Data-center network topology	6
4.2. Configuring locator addressing	7
5. Deploying ILA routers	10
5.1. ILA Redirect Message	10
5.2. Configuration parameters	10
5.3. ILA router operation	11
5.4. Scaling considerations	12
6. Deploying ILA hosts	13
6.1. Configuration parameters	13
6.2. Providing task isolation	13
6.3. ILA host operation	14
7. Using BGP as the ILA control plane	16
7.1. BGP topology	16
7.2. Any-to-any mapping distribution	17
7.3. Hub-and-spoke mapping distribution	17
8. Push vs pull mapping distribution modes	18
9. ILA address management	18
9.1. Decentralized address management	18
9.2. Centralized address management	19
9.3. Role of Task scheduler	19
10. ILA domain federation	20
11. Operational Considerations	20
11.1. Operational procedures for ILA routers	21
11.2. ICMPv6 Message generation by transit devices	21
11.3. Multicast routing	22
11.4. Potential ILA mapping table complications	22
11.5. Potential ILA routers complications	23
12. Deployment Scenario Primer	24
13. IANA Considerations	25
14. Manageability Considerations	25
15. Security Considerations	26
15.1. ILA host security	26
15.2. BGP Security	26
15.3. ILA router security	26
15.4. Tenant security	26

16. Acknowledgements	27
17. Informative References	27
Author's Address	29

1. Introduction

This document provides high-level guidelines for building an ILA-enabled datacenter using BGP [RFC4271] as the protocol for ILA mapping information dissemination. The reader is expected to be familiar with the principles presented in [I-D.herbert-nvo3-ila]. Reading on ILNP architecture defined in [RFC6740] is also recommended, but not needed for understanding of this document. While ILA does not implement the original ILNP proposal, it's based on the same idea of maintaining the Identifier vs Locator split in the IPv6 address.

ILA benefits from routed datacenter networks, i.e. networks that do not rely on spanning Layer-2 domains across multiple network devices. Endpoint mobility made possible by ILA is one of the key benefits ILA brings to the datacenter networks. Combining ILA with fully routed network design allows for achieving the robustness of routed network with the flexibility of endpoint mobility. Some practical recommendations for building a fully-routed datacenter network could be found in [RFC7938] or [ROUTED-DESIGN].

Though workload mobility could also be achieved in L3 switched networks by using "host-route injection" technique, such approach has limited applicability, due to high stress put on the underlying control and data planes. The mobile prefix needs to be removed, re-injected and propagated to all network devices every time an address moves.

ILA is an alternative to "encapsulation" approaches, such as LISP ([RFC6830]), for realizing the endpoint mobility and network virtualization. Using simple address rewrites significantly reduces the processing overhead on the hosts, and makes various hardware and software network acceleration functions easier to implement (e.g. checksum computation offload). Furthermore, ILA keeps the underlying network fully visible to the applications that use ILA addresses, which makes network troubleshooting easier, as compared to the "encapsulation" approaches.

2. Terminology

This section defines ILA-specific terminology that will be used through the document.

ILA domain: a collection of ILA hosts and ILA routers that collectively support ILA identifier mobility and network virtualization model. The ILA domain is assigned a single 64-bit IPv6 prefix known as SIR (Standard Identifier Representation, see [I-D.herbert-nvo3-ila]) prefix, which is made known to all hosts and routers in the domain. This prefix is used to construct the complete 128-bit IPv6 addresses for ILA identifies found in the domain.

SIR Address: IPv6 address constructed from SIR prefix concatenated with the 64-bit identifier. This is the address visible to the applications and transport layer on ILA hosts.

ILA Address: IPv6 address constructed from actual valid 64-bit locator and 64-bit identifier. This address is what being seen by transit network devices - it is expected to be routable in the underlying network.

ILA mapping table: The table for mapping identifiers to locators present in ILA host or ILA router. This table is updated either via BGP, or ILA redirect messages. ILA routers maintain full authoritative copy of the table, while ILA hosts may have their own smaller view of the global mapping state.

ILA host: network endpoint that is capable of accepting and originating packets with ILA addresses, by performing stateless rewrite between SIR addresses and ILA addresses. The host maintains its own local version of the ILA mapping table and has at least one ILA locator (64-bit prefix) assigned.

Non-ILA host: network endpoint that is not aware of ILA addressing structure and does not participate in ILA address translations. To this host, the SIR and ILA addresses look like regular IPv6 addresses.

ILA router: network endpoint that is responsible for two main functions:

- * Storing and disseminating the authoritative ILA mapping information within the ILA domain (NVA role per [I-D.ietf-nvo3-arch]).
- * Serving as the gateway between the ILA-hosts and non-ILA hosts, as well as the gateway for communicating with other ILA domains (NVE role per [I-D.ietf-nvo3-arch]).

Task: the unit of mobility in ILA domain. Each task is assigned an identifier unique within the ILA domain, which follows the task

as it changes the hosts and, consequently, the locators. Implementation wise, the task can run within a container or a virtual machine, for example.

Tenant: owner of the tasks executed in the shared environment.

Common Locator Address (CLA): Special ILA address constructed as <locator>::1 and identifying the physical host itself. This address is used to send and receive of the ILA redirect messages.

3. ILA deployment process

The ILA domain consists of the following conceptual elements:

- o Routed network that provides reachability among physical hosts, i.e. provides routing within the locator address space.
- o ILA hosts, each assigned a unique /64 prefix reachable within the network. ILA hosts maintain their own local version of ILA mapping table.
- o ILA routers, each injecting the domain's SIR prefix into the routed network and maintaining the full mapping table for the ILA domain. The routers could be implemented in software, or using specialized hardware appliances.
- o Centralized BGP router-reflector nodes that peer with all of the ILA hosts and all of the ILA routers within the domain for the purpose of mapping information dissemination. ILA hosts and routers run the BGP processes to communicate with the reflectors.

Deploying ILA in datacenter requires the following logical steps:

- o Preparing the network. Assigning locator addressing to the hosts (servers) in the network and providing routed interconnection among the locator prefixes.
- o Configuring ILA hosts and ILA routers. Each ILA domain requires a set of ILA routers to facilitate mapping function and provide connectivity to other ILA domains and the Internet. Each ILA domain is assigned a /64 SIR prefix, which scopes all identifiers in the domain. All ILA hosts and ILA routers within a domain are aware of the SIR prefix of this domain.
- o Enabling the ILA control plane. Configuring the BGP mesh for mapping information dissemination within the ILA domain and injecting the SIR prefix into routed network from the ILA routers to facilitate communications among the ILA domain and from / to

the Internet. See [I-D.lapukhov-bgp-ila-afi] for definition of the corresponding BGP extension.

- o Deploying an address management solution to coordinate allocation of ILA identifiers. In simplest cases, the addresses could be generated on each host individually, without central coordination.

4. Preparing the network

This section provides overview of the network-related configuration needed for ILA.

4.1. Data-center network topology

For ease of reference, this document adopts the Clos topology described in [RFC7938] along with the terminology developed in that document.

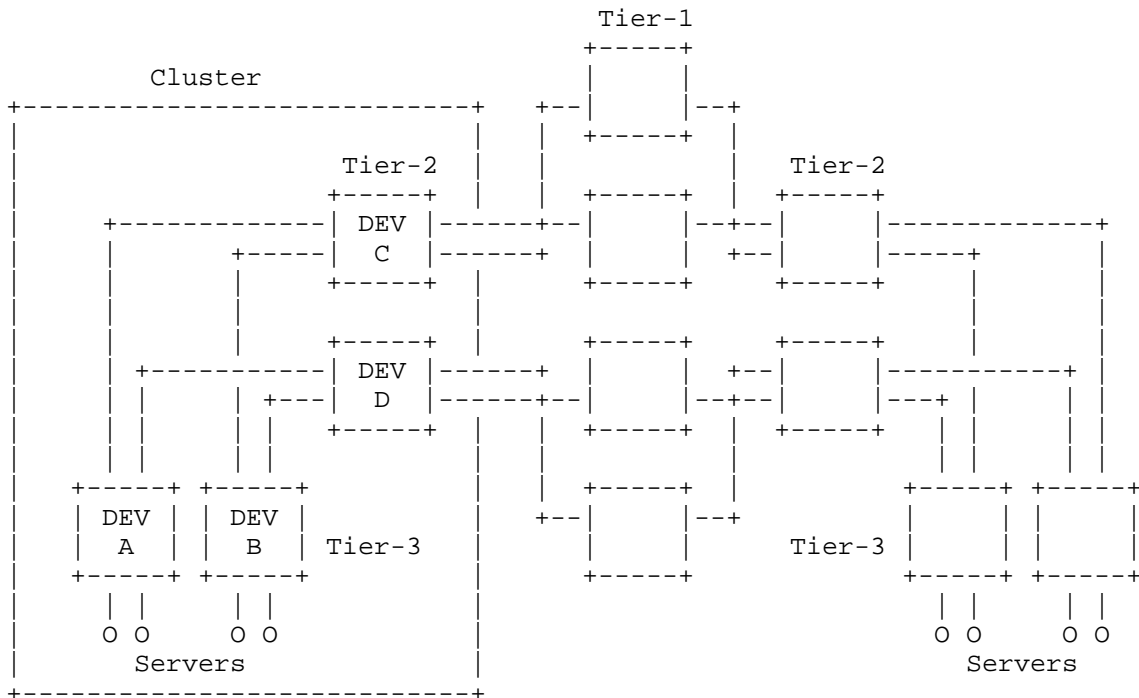


Figure 1: 5-Stage Clos topology

The network is partitioned hierarchically in three tiers, with tier numbering starting at the "middle" stage of the Clos network. The "middle" tier is often called as the "spine" of the network.

A set of directly connected Tier-2 and Tier-3 devices along with their attached servers will be referred to as a "cluster".

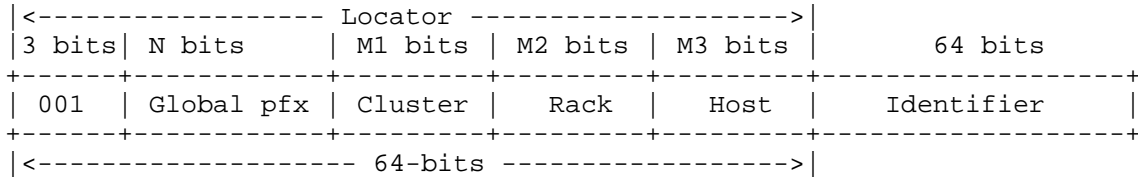
Tier-3 switches that connect the servers, are often referred to as "ToR" (Top of Rack) switches or simply "rack switches".

4.2. Configuring locator addressing

A mandatory prerequisite for ILA deployment is enabling IPv6 routing in the network. This could be done using either dual-stack IPv4/IPv6 deployment or IPv6-only deployments. This document assumes the network has been already configured to forward IPv6 traffic. See [I-D.ietf-v6ops-dc-ipv6] for operational considerations on deploying IPv6 in the datacenter.

ILA requires every ILA host to have at least one 64-bit locator assigned. This means that every host (server) in the datacenter network needs to have at least one /64 IPv6 prefix configured on one of its interfaces. These /64 prefixes could be either globally routable or unique-local.

The use of the globally routable addressing scheme allows for deploying highly scalable hierarchical addressing scheme, and make the locators accessible from the Internet. The figure below illustrates the structure of a globally-routable locator:



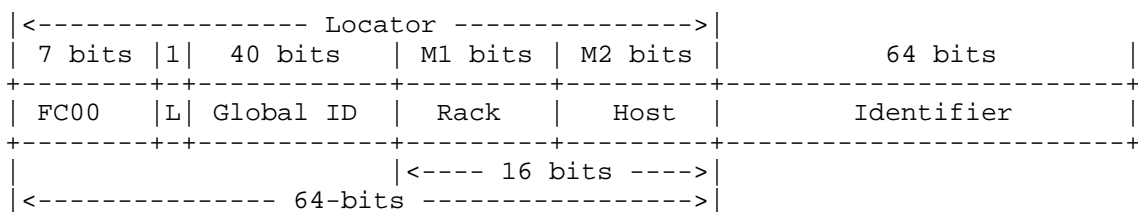
For example, a global /32 prefix (N=29) allows for sub-allocation of 2^32 locators. This sub-allocation could be done hierarchically, mapping to the tiers of network topology. Following the /32 example prefix:

Allocate 256 /64 prefixes per Tier-3 switch (M3 = 8 bits), which allows for up to 256 physical hosts in a rack, with /56 prefix assigned per rack.

Assuming 256 Tier-3 switches per cluster, one would allocate /48 per cluster (M2 = 8 bits).

This leaves room for 16-bits (64K) cluster per datacenter (M1 = 16 bits). This space could be further sub-divided if multiple Clos network fabrics have been deployed.

The use of unique-local addressing for locators is more limiting in terms of available space, as it only offers 16-bits for sub-allocation. It does, however, have the benefit of ad-hoc allocation. This could work better for smaller deployment, e.g. allocating 10-bits to enumerate Tier-3 switches (physical racks of servers) and 6 bits to enumerate hosts within a rack. For instance, the address structure may look as following, here M1 = 10 bits and M2 = 6 bits.



In either case, the addressing scheme is hierarchical, allowing for simple route summarization logic and better routing system scaling (see [RFC2791]). This is especially important in case of IPv6, since contemporary datacenter network switches often have smaller IPv6 lookup tables as compared to IPv4. Route summarization also requires certain network design changes to avoid packet black-holing under link failures. This problem gets more complicated in Clos topologies, and analyzed in more details in [RFC7938].

In greenfield deployments, each ILA host could be assigned a /64 locator prefix during provisioning phase. There are multiple options to accomplish this:

- o Assigning static link-local addresses to servers and statically routing /64 prefixes from Tier-3 switches to the servers over those link-local addresses. In this model, the operator would plan and pre-allocate per ILA-host prefixes beforehand, and configure the Tier-3 switches accordingly. From operational risks perspective, if the server is not present while the static route is configured on Tier-3 switch, packets destined to the corresponding /64 prefix will cause the switch to continuously generate IPv6 NDP packets ("gleaning"), which puts extra stress on the device's CPU.
- o The servers may request the /64 prefix using IPv6 Prefix Delegation mechanism as defined in [RFC3633]. This allocation

could be made "permanent" by proper DHCPv6 server configuration and ensuring the same prefix is always being delegated to the same server. The Tier-3 switch would act as DHCPv6 relay and will install the corresponding /64 IPv6 route dynamically. This approach addresses both the allocation and the routing problem, but makes the setup potentially more fragile operationally (reliance on additional protocol) and harder to debug (additional process involved).

- o The server may run a routing daemon (e.g. BGP process) and inject the pre-allocated /64 prefix into Tier-3 switch. The address allocation in this case needs to happen by some other means. This is more suitable for ad-hoc ILA testing and small, rapid deployments.

The server itself may use one of the IPv6 addresses in /64 prefix for its own addressing, e.g. for remote access or management purposes. Alternatively, the server may obtain another IPv6 address from a different (non-locator) IPv6 address range allocated for the datacenter. This document recommends using <locator>::1 as the special identifier, naming it as "Common Locator Address" (CLA). Such choice of identifier make it easy to differentiate from regular identifiers. This identifier could be used for connectivity testing.

Route summarization for the locator prefixes is highly desirable to reduce the stress on the network switches forwarding tables and improve control-plane stability, and need to be implemented at least on Tier-3 switches. In simplest case, the switches could be statically preconfigured with the summary routes. These routes need to agree with the prefixes that are assigned to the servers, especially in the case when dynamic prefix injection is used. As a possible alternative, simple virtual aggregation could be employed, where hosts inject both the specific and the summary route, and installation of corresponding FIB entries is suppressed as per the rules defined in [RFC6769]. The latter approach does not improve the control plane scalability, but solves the issues with packet black-holing in presence of network summarization. It also requires the network hardware support, which may not be present.

In retrofitting scenarios, the servers are likely to already have 128-bit IPv6 addresses assigned, allocated from the datacenter address space, e.g. by using a single /64 prefix per Tier-3 switch. In this case, the additional locator prefix needs to be assigned in the same way as described above for greenfield deployments. The only difference is that the new prefix and the old server address may be allocated from different IPv6 address ranges.

5. Deploying ILA routers

ILA routers perform multiple functions within the ILA domain:

- o Serve as the centralized store of the identifier-to-mapper information in the domain. The mappings are delivered to the ILA routers as described in Section 7.
- o Act as the gateway between the ILA hosts and non-ILA capable hosts, e.g. the Internet.

The ILA hosts will send the packets destined to identifiers they don't have mappings for to the ILA routers initially to perform the ILA translation, and the hosts outside of the ILA domain will use the ILA routers for all communications with the domain. The ILA routers may also act as ILA hosts and have one or more identifiers assigned.

5.1. ILA Redirect Message

ILA routers may originate and ILA hosts must receive and process ILA redirect messages. The ILA redirect message is carried in UDP packet and destined toward a well-known port. It carries the information binding an identifier to its locator. For security purposes, this message is expected to be authenticated by cryptographic means, such as by using keyed HMAC (message authentication code) procedure. Every host in the domain is then required to be configured with the key information to be able to validate the redirected messages.

The ILA redirect message might be signed with multiple HMAC keys to facilitate key transition in the domain. The redirect message will carry multiple signatures along with corresponding numeric key-identifier, and the ILA hosts are expected to use the signature with the highest locally known identifier. As the old key leaves rotation, eventually every host will get updated and the signature made using the old key could be removed.

5.2. Configuration parameters

The ILA routers need the following configured for their operation:

- o Regular, non-anycast 128-bit IPv6 address to connect the ILA router to the datacenter network.
- o Cryptographic material to authenticate ILA redirect messages, for example key to be used with HMAC scheme.
- o The /64 SIR prefix for the ILA domain, shared by all ILA routers. This prefix is advertised into the network in anycast fashion and

"intercepts" all traffic destined from hosts outside of ILA domains to the SIR addresses in the domain. The prefix could be injected in "always-on" fashion, e.g. by using BGP injectors on ILA routers. This couples the ILA router's life-cycle with the prefix injection cycle.

- o Control-plane configuration, i.e. the IPv6 addresses of BGP route reflectors, and possibly some configuration for the local BGP process. This is discussed in more details in Section 7.
- o Management settings, such as maximum rate of ILA redirect messages, and associated security attributes (e.g. the key pair used for message signing).
- o A configuration flag that instructs the router whether the ILA redirect messages needs to be sent out. The ILA router does not receive ILA redirect messages, since by design it knows of all active mappings in the domain.

5.3. ILA router operation

Upon booting, the ILA router is first required to join the control plane mesh and learn of the mappings that exist in the ILA domain. It is also aware of the SIR prefix that is used within its domain. After the router has learned of the mappings, it may inject the anycast SIR prefix in the datacenter network and join the operational group of ILA routers.

Just like any ILA node, the ILA router is required to have a 64-bit locator configured. Special identifier `::1` is used to build the source and destination addresses of the ILA redirect messages.

When ILA router receives a packet with the upper 64-bits of the destination IPv6 address matching its configured SIR prefix, it performs the following:

- o If the destination address does not match the SIR prefix, the ILA router discards the packet, as it is not supposed to be received by the ILA router.
- o Attempts to resolve the source identifier (bottom 64-bits of the source address), if applicable. If the source address matches SIR prefix, it is coming from an ILA host. The route then needs to translate the identifier found in the source address to its locator. If the translation fails, send back the ILA "Mapping Not Found" message. If the source address does not match the SIR prefix, then no translation is needed, and no redirect messages need to be sent back.

- o Attempts to find the locator matching for the destination identifier (the bottom 64-bits of the destination IPv6 address). If the mapping for destination identifier is not found, the original packet is dropped, and an ICMPv6 "Destination Unreachable" message, type "3" is sent back to the message originator. Otherwise, the router does the following:
 - * Rewrites the SIR prefix in the destination IPv6 address with the new locator and forwards the packet back to the network.
 - * If sending of ILA redirect messages is permitted, the router sends the ILA redirect message back to the originator of the packet, by looking up the source identifier and finding the corresponding locator. The redirect informs the source of the actual destination locator. The redirect messages must be rate-limited to avoid sending ILA redirect for every incoming IPv6 packet.
 - * As mentioned previously, the source and the destination ILA addresses of the redirect message IPv6 header use the identifier value "::1", which designated them to be delivered to the ILA control process.

If the source IPv6 address check reveals that the packet is not coming from the ILA domain the router belongs to (i.e. the SIR prefix does not match), the ILA router does not need to send back the ILA redirect message, but instead simply continue to forward the packet as if the locator for the destination identifier could be found. The ILA router will still send the ICMPv6 "Destination Unreachable" message for unknown mappings.

5.4. Scaling considerations

Due to high load and reliability concerns, the ILA domain needs multiple ILA routers. The simplest way to provide redundancy is by letting the ILA routers inject the /64 SIR IPv6 prefix into the datacenter network in anycast fashion ([RFC4786]). This will allow to naturally use the datacenter network's Equal-Cost Multipath (ECMP) capabilities to distribute traffic among the ILA routers.

For redundancy purposes, the ILA routers would need to be spread across multiple physical racks in the datacenter. More ILA routers could be added incrementally to reduce the load and scale capacity horizontally, and join the operational ILA group in non-disruptive fashion, after they have learned the full mapping table for the ILA domain.

Use of anycast method does have some resulting routing implications. For example, using the network described in Section 4.1 will result in ILA hosts preferring to use the ILA routers in the same cluster, since those are closer based on the routing metric. Thus, the network may not evenly spread their packets across all ILA routers in the datacenter. It is therefore possible that some ILA routers will receive more traffic than the others. This issue is specific to anycast routing in general, and not specifically to ILA.

6. Deploying ILA hosts

This section reviews the deployment considerations for the ILA hosts.

6.1. Configuration parameters

The ILA hosts need to be configured with the following:

- o SIR prefix of the ILA domain.
- o IPv6 addresses of the BGP route reflectors.
- o The routable /64 locator assigned to the host.
- o ILA mapping entries expiration time, to time out unused entries.
- o Cryptographic material to allow validation of redirect messages.
- o Boolean flag, defining whether ILA redirection messages sending / receiving is enabled.

By disabling both the ILA mapping expiration time and the sending of ILA redirect messages the host is effectively configured for the "push" ILA mapping distribution distribution mode (see Section 8). In this mode, the BGP (control plane) is assumed to update/synchronize all of the ILA mapping entries in response to the identifier move events, and redirect messages are not used.

The host is expected to receive ILA redirect messages destined to its locator and identifier value of ":::1". The source of such message must also use the identifier value of ":::1" to be considered a redirect message.

6.2. Providing task isolation

In simplest case, the host only needs to implement the ILA address rewrite function and inform the tasks starting on the host of the ILA addresses they can use. However, it might be desirable to provide the tasks with strong networking isolation guarantees, i.e. making

table empty, depending whether "push" or "pull" distribution model has been selected.

When a task starts it will have an ILA identifier allocated, and the corresponding IPv6 address (built out of SIR prefix + the allocated identifier) bound to an interface within the networking namespace created for the task. The mapping is then propagated over BGP peering sessions to all ILA routers.

For outgoing packets, the ILA host performs the following:

- o Matches the destination IPv6 address against the SIR prefix.
- o If prefix matches, attempts to look-up the identifier portion of the address in the local ILA mapping table.
- o If a match is found in ILA mapping table, rewrite the destination address and replace the SIR prefix with the actual locator.

For packets with destination IPv6 addresses that do not match the SIR prefix, usual forwarding rules apply. If no match is found for the SIR address, the packet is sent as is, and is expected to be delivered to the ILA routers, since those advertise the SIR prefix into the routing domain (without getting the locator portion rewritten - the packet has the SIR prefix in place of the locator).

For incoming packets, the ILA host should perform the following:

- o Match their destination IPv6 addresses against the locator prefix (64 bits) of the host.
- o If the destination address matches, deliver the packet to the corresponding namespace, based on the identifier portion.
- o If the destination identifier in the incoming packet does not match any of the ILA mappings, and sending of ILA redirect message is enabled, the host sends an ILA redirect message back to the originator of the packet. The message will have an empty locator value, and informs the sender that the mapping it has for the identifier is no longer valid, prompting to erase the corresponding entry in the sender's ILA mapping table.
- o If the source address is SIR address, the receiving host may increase time-to-live for the corresponding mapping entry, if it is present in the ILA mapping table. This acts as a signal confirming liveness of the remote corresponding, and validity of the existing mapping. Otherwise, the mapping would be expired

based on the time-to-live provided by the original ILA redirect message, if ILA mapping expiration is enabled.

Sending an ILA redirect message by the ILA host requires the host to translate the source identifier of the original message. Assuming that flow was likely bi-directional, the entry should be readily available in the local ILA mapping table. If not, the ILA redirect message will be routed toward the originator via the ILA routers, i.e. sent back with locator equal to the SIR prefix. It is possible that both source and destination identifiers of the flow have moved, resulting in mutual sending of ILA redirect messages, and temporarily falling back to using the ILA routers.

If the ILA mapping entry expiration time is set to non-zero, the unused ILA mapping entries will eventually be deleted. The entry expiration needs to be disabled if the mappings are learned in event-driven fashion via the BGP mesh ("push" distribution mode).

7. Using BGP as the ILA control plane

This section discusses the use of BGP for ILA mapping information dissemination. The choice of BGP is made to allow for easier integration of hardware appliance, e.g. network switches with extended functionality, where BGP is commonly used as the control plane. Furthermore, BGP itself offers a simple way of disseminating data and converging on a key-value mapping across multiple nodes in eventually consistent fashion, and has proven track record of use in the industry. Furthermore, use of BGP allows for leveraging the monitoring extensions developed for the protocol. For example, [I-D.ietf-grow-bmp] could be used to observe ILA mapping changes in the network using existing tooling.

7.1. BGP topology

Per the common practice, a group of BGP route-reflectors (see [RFC4456]) should be deployed and peered over IBGP with all ILA hosts and ILA routers in the ILA domain. The reflectors themselves would also be peered in full-mesh fashion to provide backup paths for mapping information distribution, e.g. in case if one of reflectors loses a session to a host. Those reflectors do not need to be in the data-path, but merely serve for the purpose of information distribution. The number of route-reflectors should be at least two, to allow for redundancy. See below sections for discussion of route-reflection settings.

It is possible to co-locate the BGP route-reflectors with the ILA routers. This saves on having additional nodes for the purpose of just BGP route-reflection, but puts extra memory and CPU stress on

the ILA routers, and therefore is less desirable. Furthermore, it makes capacity-planning more difficult, and therefore is not recommended.

The route-reflectors are required to peer with potentially a very large number of ILA hosts, which may put scaling limits on the size of the ILA domain due to the overhead of maintaining large amount of BGP peering sessions. To alleviate this problem, the pool of ILA hosts may be split into "shards" and each shard would peer with a different group of route-reflectors. For example, the ILA domain may have four groups of route reflectors, each with four route-reflectors. The sixteen route-reflectors may then peer in a full-mesh fashion, to exchange the mappings they have received from the corresponding "shard" of the ILA domain. This method avoid the issues related to maintaining large amount of TCP sessions, but every BGP route-reflector is still required to maintain the full ILA mapping table.

In addition to ILA AFI/SAFI's, other AFI/SAFIs could be configured on BGP speakers, e.g. using [I-D.lapukhov-bgp-opaque-signaling] for opaque information dissemination in the ILA domain, e.g. to facilitate in distributed address allocation.

7.2. Any-to-any mapping distribution

In this mode, the ILA routers could act as IBGP route-reflectors [RFC4456] for all of the IBGP sessions they have, and relay the mapping information among the ILA hosts. This would allow the hosts to avoid initially sending packets to the ILA routers, at the expense of maintaining the ILA mapping table. Additionally, this allows for completely disabling the ILA redirect messages and using only the mapping information propagated by BGP.

7.3. Hub-and-spoke mapping distribution

Alternatively, BGP could be used to deliver the mappings from ILA hosts to ILA routers only. The hosts and the routers would establish IBGP peering sessions with the route-reflectors in hub-and-spoke fashion, with BGP reflectors being the hubs. The ILA router sessions will be configured as the "route-reflector clients" on the route-reflectors, while the ILA hosts sessions will be left as ordinary IBGP sessions. This will propagate all needed mappings to the ILA routers and allow them to properly redirect the hosts. The ILA hosts are responsible for withdrawing and announcing the mappings as they change.

8. Push vs pull mapping distribution modes

The default mode of operations in ILA is "pull" mode, where mappings are learned by the ILA hosts via ILA redirect messages. Effectively, the process of populating the ILA mapping table is reactive and driven by data-plane events. In some case, e.g. upon identifier move, this may result in short periods of packet loss, while the sender receives the ILA redirect message and falls back to forwarding via the ILA routers. Furthermore, the use of ILA redirect messages requires security configuration to avoid message spoofing and cache poisoning attacks.

An alternative to "pull" mapping distribution on the hosts, is "push" mode, where all ILA hosts receive exactly the same mapping information as the ILA routers. In fact, every ILA host may even operate as an ILA router. In this case, the ILA message sending could be disabled in the ILA domain altogether. The "push" mode allows for proactive creation of the ILA mappings, and avoiding the packet loss, provided that the new mapping reaches the sending host before the destination identifier has moved. The trade-off here is the overhead of maintaining full mapping set on all ILA hosts.

For simplicity, this document recommends that all ILA hosts in the domain operate either in "push" or "pull" modes. In "push" mode the ILA mapping entries expiration needs to be turned off, along with sending of ILA messages. If an ILA host receives a packet for the ILA address it cannot map to locally, it is expected to send an ILA redirect message. If sending the ILA messages is disabled, the host must at least send an ICMPv6 "Destination Unreachable" message with code "3" - "Address Unreachable" to aid in debugging of missing mapping message. Notice that the ILA routers always operate in "push" mode, i.e. they only learn of mappings via the control plane exchange.

9. ILA address management

The ILA control plane and redirect messages perform mapping information dissemination, but the identifier allocation needs to be done separately. The address management process also depends on whether there is some hierarchy desired in the ILA namespace, e.g. if allocating a prefix per-tenant is needed.

9.1. Decentralized address management

In simplest case, each ILA host may independently allocate unique identifier per task when it first starts, and the task will retain it for the duration of its lifetime (see Appendix A of [I-D.herbert-nvo3-ila]). The chances of collision are very low given

the 60-bit value of the identifier. The scheduler is responsible for starting and moving the task in the ILA domain. The tasks belonging to the same tenant may discover each other's addresses by some out-of-band signaling mechanism, e.g. a key-value store such as ([MEMCACHED]) or [ETCD] or use BGP for the same purpose as described in [I-D.lapukhov-bgp-opaque-signaling]. For instance, the task may publish its own identifier, consisting of the tenant name and task name, mapped to the SIR address of the task.

Decentralized allocation is still possible even if the unit of address allocation is prefix, e.g. when multiple tenants are sharing the infrastructure, and unique VNID (see [I-D.herbert-nvo3-ila] for definition) is needed per tenant to build the 96-bit prefixes allocated to tenants from the /64 SIR prefix. Since the size of VNID space is rather small, generating random VNIDs becomes more prone to collision. In this case, decentralized address allocation schemes, such as one described in [RFC7695] could be used. These techniques require the ILA nodes to have some shared communication medium for nodes to "claim" the prefixes and avoid collisions. Once again, various distributed key-value stores could be used to accomplish this.

9.2. Centralized address management

In the case where high level of control is needed to allocate the addresses, e.g. per-tenant prefixes, centralized address management schemes could be used in the ILA domain. This could be either proprietary address allocation system, or system built on top of protocols such as DHCPv6.

9.3. Role of Task scheduler

The ILA domain needs a tasks scheduler responsible for resource allocation and starting of tenant's tasks on the ILA nodes. Defining functions of such scheduler is outside of scope of this document. At the very minimum, the scheduler would need agents running on every ILA host, participating in ILA address allocation, and communicating with the ILA control plane to publish and remove the mappings. Since it's the scheduler that is responsible for task movements, it makes sense for the scheduler to update the mappings in the domain.

The scheduler needs some kind of API to interact with the BGP process on the box. Defining the exact API is outside of scope of this document, but as an option the scheduler may use a BGP session to inject prefixes into the BGP process running on the box.

10. ILA domain federation

In default operation mode, the ILA domains act as if the other domain is unaware of mappings that exist in another. It is possible to let the two domains exchange the mapping information and honor the ILA redirect messages from another domain by "merging" full or partial mapping tables of the two domains. For example, one can envision multiple compute clusters, each being its own ILA domain. In standard ILA model, those clusters would need to communicate via the ILA routers only, increasing stress on the data-plane. To allow traffic flowing directly between the hosts in each cluster and bypassing the ILA routers, the ILA domains may exchange the mapping information, and program the ILA mappings in ILA hosts to facilitate direct paths.

Since each domain may re-use the 64-bit identifier space on its own, the use of SIR prefix is required to make the identifiers globally unique. This requirement is easily fulfilled since the SIR prefix is required to be globally routable in the Internet.

To enable ILA domain federation, the BGP route-reflectors in each domain need need to be fully meshed and configured to use the "VPN-ILA" SAFI with "ILA AFI" (see [I-D.lapukhov-bgp-ila-afi]). This will propagate the mappings known to each route-reflector scoped with the SIR prefix of the local domain. If multiple domains are federated in this way, intermediate route-reflectors could be used, and filtering techniques such as described in [RFC5291] and [RFC4684] could be employed. The filtering may be further used to allow leaking of only select mappings, e.g. for the identifiers or tenants that carry lots of traffic.

If "push" distribution model is chosen with ILA domain federation, the ILA hosts will need to be configured to use "VPN-ILA" SAFI on their peering sessions with the BGP route reflectors. The ILA mapping entries lookup then need to be keyed both on the SIR prefix and the identifier to be resolved. Given the large volume of mappings that may exist in federated model, the "pull" model might become more preferable.

11. Operational Considerations

ILA introduces additional step in packet routing and thus adds more complexity to network troubleshooting process. At the same time, relative to the virtualization techniques that employ encapsulation and tunneling, ILA makes the underlying physical network fully visible to the tasks, and thus make tenant-driven troubleshooting simpler. This section discusses some operational procedures specific

to ILA and the additional fault models that are possible in presence of ILA.

11.1. Operational procedures for ILA routers

ILA routers may be added/removed from the network at any time. Adding a router is commonly needed to scale the capacity of the ILA router group when peak loads increases. Adding an ILA router is non-disruptive procedure. It starts by configuring the ILA router to peer with the BGP mesh to learn of all mappings in the domain. The use of BGP graceful restart (see [RFC4724]) would allow the new router to learn when all mappings have been advertised. At this time, the router may inject the SIR prefix, joining the operational group of ILA routers and start forwarding ILA traffic.

To gracefully take the ILA router out of service, it may be instructed to stop announcing the SIR prefix, or, in case of BGP, announce it with less preferable path attributes. This will allow the router to still accept and forward all in-flight packets, but will redirect the remaining packets toward the remaining ILA routers.

11.2. ICMPv6 Message generation by transit devices

Upon some conditions the transit, ILA-unaware devices, may need to generate ICMPv6 messages, e.g. when IPv6 hop limit exceeds. The source of the packet sent by an ILA application would have SIR as the prefix, and hence the ICMPv6 message will need to transit an ILA router before getting back to the host that sent the original packet. This has some operational downside, as it adds path stretch to the control message flow, and needs to be accounted for operational reasons.

When an ICMPv6 message generated by an intermediate device arrives back to the sender of the original packet, the ILA may need to translate the payload of the ICMPv6 message, as it often contain the IPv6 header of the original packet. This is needed so that the control message could be properly correlated to transport level connection. Thus, it is expected that the ILA host stack will be able to perform this translation, and replace the ILA locator with SIR prefix in the destination address field of the encapsulated IPv6 header.

The last case is generating ICMPv6 message by transint device for packet sourced by non-ILA host (or outside of local ILA domain) and translated by an ILA router. In this case, the response will be directed back to the non-ILA host, bypassing the ILA router, and there will be no easy way to perform the translation of the location

portion in ILA destination address back to the SIR prefix. The non-ILA sender would be able to process the ICMPv6 message.

11.3. Multicast routing

Defining multicast routing and group membership dissemination is outside of scope of this document.

11.4. Potential ILA mapping table complications

Every packet egressing from an ILA host and matching the SIR prefix is subject to lookup and translation in the local ILA mapping table. If entry is not found, the packet is forwarded to the ILA router(s) by the virtue of SIR prefix injected in the datacenter network. If the ILA router does not have the mapping, either the ICMPv6 "Destination Unreachable" or "ILA mapping not found" message will be sent back, depending on whether the original sender is ILA or non-ILA host. There are few observations to make here:

- o Packets egressing the ILA host and not matching the SIR prefix are routed as usual.
- o ILA destinations that are not yet present in the ILA mapping table will be initially routed toward the ILA routers (e.g. the ILA routers will show up in the initial "traceroute" command output).
- o In case of missing identifier mapping, it's the ILA router that informs the sender of this event via either an "ILA Mapping not Found" or ICMPv6 "Destination Unreachable" messages.

Thus, the case of missing mapping is easily debuggable, though the "transition period" when the mapping is not yet in the ILA mapping table might confuse the operator using the "traceroute" command.

The most difficult case of ILA mapping table malfunction would be presence of incorrect mapping, i.e mappings pointing to a non-existent or incorrect locator.

- o Non-existent locator. This will route the packet through the network, and eventually result either in packet getting discarded due to missing route or IPv6 NDP entry, or packet dropped due to routing loop and hop-limit expiration. In either case, the original sender may detect this condition either via reception of ICMPv6 "Destination Unreachable" messages, or by observing the output of the "traceroute" command. The ILA host may also be configured to make sure the identifiers fall within the known prefix range.

- o Incorrect locator. In this case, the packet will be delivered to the wrong ILA host, that does not have the mapping for the identifier. Depending on whether the sending of ILA redirect messages is enabled on the host, two scenarios are possible:
 - * The destination ILA host sends back an ILA redirect message with empty locator, informing the sender that mapping is invalid. The sender will invalidate the ILA mapping entry and switch over to forwarding via the ILA routers. The latter will either inform if of the new mapping, or send an ICMPv6 "Destination Unreachable" message back.
 - * The destination ILA host is not configured to send the ILA redirect messages back. In this case, it simply responds with the ICMPv6 "Destination Unreachable" messages for the duration of time the sender keeps sending the packets using the incorrect mapping. The mapping needs to be flushed our updated by some external mean.

Next possible failure is dropped ILA redirect messages. However, given that the ILA redirect message sending process is memoryless, the recipient will eventually receive one of them, or at least finish the communication via an ILA router.

11.5. Potential ILA routers complications

The ILA routers serve as proxies for traffic entering the ILA domain, as well as temporary transit hops for traffic between the ILA hosts when they don't have matching mappings, in case if "pull" distribution model is utilized. The following operational observations apply:

- o Traffic between the ILA domain and external world will necessarily flow asymmetrically. The packets toward the ILA hosts sent from the outside will always cross the ILA routers (see Section 10 for exceptions from this case) and traffic returning from the ILA hosts to the external world will flow directly, bypassing the ILA routers. This will show up in the outputs of the "traceroute" command running from sender and destination and showing asymmetric paths. This being said, asymmetric traffic flows are very common in modern networks, and thus it should be a problem on its own.
- o A failure of ILA router should be handled by re-balancing the load automatically by means of ECMP re-hashing in the network, and therefore should be mostly transparent to the ILA hosts, unless the load increases significantly after the failure. It is possible to have cascading failure and lose all ILA routers, or have them over-utilized. This event should be detected by

external monitoring system, and be acted upon by adding more ILA routers to the domain - either automatically or manually. From troubleshooting perspective, the event will manifest itself via massive packet loss toward all hosts in the ILA domain.

- o A malfunction of single ILA router (e.g. network interface card issue) would manifest itself in somewhat increased packet drop ratios for flows crossing the ILA routers, mostly traffic from external nodes. The more ILA routers the domain has, the harder to notice this ratio would be, since ECMP mostly spreads traffic evenly over all the ILA routers. This problem is more specific to ECMP behavior, and tooling exists to deal with it in datacenter networks.
- o ILA routers are in path of the ICMPv6 messages generated by non-ILA aware routers in the network. Thus, a loss of such packet in the network could not be differentiated from the loss due to the drop by an ILA router. This may potentially complicate network troubleshooting efforts.

To sum the above up - the health of ILA router is critical to the ILA domain functions, even if "push" model is employed and the ILA routers are used mostly for external communications. The ILA routers should be monitored closely for vital parameters, such as CPU and memory utilization, traffic rates on their network interfaces, and packet loss toward the ILA routers themselves.

12. Deployment Scenario Primer

Building upon the concepts presented above, this section provides a simple ILA deployment scenario.

- o For locator addressing, unique-local addresses is used, with 16-bit available for sub-allocation. This allows for 1024 (2^{10}) Tier-3 switches with 64 (2^4) servers under each Tier-3 switch. Using the Clos topology from section Section 4.1 one can build 32 clusters with 32 Tier-3 switches each.
- o The hosts in the network would use BGP to peer with Tier-3 switches and inject their locator prefixes. It's desirable, but not necessary to configure the route summarization on the network switches, depending on the size of the deployment.
- o Given the small to moderate scale of deployment, four IBGP route-reflectors would be deployed in the ILA domain, without the need for extra level of aggregation hierarchy. Each route-reflector will need to be configured to accept the BGP sessions from all of ILA hosts and be able to maintain thousands of peering sessions.

- o The ILA hosts and routers should be configured with a single SIR prefix, and set up for "push" mapping distribution model, by disabling sending the ILA redirect messages. All ILA mappings will be propagated to all hosts and ILA routers via BGP. Each ILA host and router will need to be running a BGP process and peer with all four route-reflectors.
- o The ILA routers will inject the SIR prefix using BGP into the network.
- o For tasks running on ILA hosts, the globally unique ILA identifiers should be allocated independently in pseudo-random fashion by the host that first starts the task.
- o As task is moved, the task scheduler will update the mapping and publish it via BGP, forcing the ILA routers and ILA hosts to update their ILA mapping tables.
- o ILA domain federation is not used, making every ILA domain communicate to each other via the ILA routers only.

13. IANA Considerations

None

14. Manageability Considerations

ILA requires both one-time deployment efforts, and recurring management work. The initial involvement is reasonably high, as it required extending the existing network and host configuration. It does not require any significant changes to the existing applications, though, aside from making the applications use newly allocated IPv6 addresses. Majority of the required changes could be done without any disruption to the existing infrastructure.

ILA address management schemes could be arbitrarily complex, but in the most basic form do not require any centralized coordination. Thus, in many cases it could be a simple local subroutine that generates a pseudo-random identifier.

Recurring management efforts are mostly concentrated on monitoring the component of ILA deployment, primarily the ILA routers and the BGP route reflectors. Troubleshooting these components follows the standard process and uses regular tooling, with the caveat of having more logical components to deal with, primarily the ILA routers and the ILA mapping tables on the ILA hosts. This increases the complexity of troubleshooting process, as more state needs to be inspected and validated.

15. Security Considerations

ILA introduces new security considerations described below.

15.1. ILA host security

If unsecured ILA redirect messages are used, the ILA hosts could be exposed to cache poisoning attacks. This calls for ILA redirect message authentication, e.g. by use of digital signatures, such as [ED25519]. This will also require to use some mechanism for propagation of public keys associated with the SIR prefix (the ILA routers) and every locator in the domain, since the ILA redirect message could be sent by either.

To prevent tasks from every being able to send packets directly bypassing the mapping layer, the ILA hosts should prohibit the task from sending packets toward the address space associated with the locators. Given that all locators will likely to belong to one large prefix, this could be accomplished by installing a single filtering rule on the ILA host.

15.2. BGP Security

Standard means of improving BGP security as described in [RFC7454] could be applied to harden the mapping dissemination system. Among them, the most important one is likely to be the "TCP Authentication Option" described in the referenced document. Notice that the BGP subsystem used to distribute the ILA mappings is not as vulnerable as the Internet BGP mesh, since it only work within the boundaries of a privately managed data-center.

15.3. ILA router security

ILA routers are primarily susceptible to various form of rate-based DDoS attacks. Primary concern would be overrruning the capabilities of ILA routers with too many packets sent from non-ILA hosts toward the SIR addresses, or "thundering herds" problem when ILA translation tables on the ILA hosts expire synchronously, or due to poisoning attack. Primary ways to address this concern would be closely monitoring server utilization and potentially rate-limiting packet flow to the ILA router on the upstream network device (ToR switch).

15.4. Tenant security

ILA does not natively isolate the tenant traffic from each other, nor from the underlying physical infrastructure. In fact, this is seen as one benefit that makes many troubleshooting processes easier. The access control then become responsibility of the tenant itself, by

employing traffic filtering rules. To this point, implementing filtering rules gets simpler if the tenant is allocated single prefix, as opposed to each task getting an unique identifier.

16. Acknowledgements

TBD

17. Informative References

- [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, <<http://www.rfc-editor.org/info/rfc4271>>.
- [RFC4456] Bates, T., Chen, E., and R. Chandra, "BGP Route Reflection: An Alternative to Full Mesh Internal BGP (IBGP)", RFC 4456, DOI 10.17487/RFC4456, April 2006, <<http://www.rfc-editor.org/info/rfc4456>>.
- [RFC4684] Marques, P., Bonica, R., Fang, L., Martini, L., Raszuk, R., Patel, K., and J. Guichard, "Constrained Route Distribution for Border Gateway Protocol/MultiProtocol Label Switching (BGP/MPLS) Internet Protocol (IP) Virtual Private Networks (VPNs)", RFC 4684, DOI 10.17487/RFC4684, November 2006, <<http://www.rfc-editor.org/info/rfc4684>>.
- [RFC5291] Chen, E. and Y. Rekhter, "Outbound Route Filtering Capability for BGP-4", RFC 5291, DOI 10.17487/RFC5291, August 2008, <<http://www.rfc-editor.org/info/rfc5291>>.
- [RFC6740] Atkinson, RJ. and SN. Bhatti, "Identifier-Locator Network Protocol (ILNP) Architectural Description", RFC 6740, DOI 10.17487/RFC6740, November 2012, <<http://www.rfc-editor.org/info/rfc6740>>.
- [RFC2791] Yu, J., "Scalable Routing Design Principles", RFC 2791, DOI 10.17487/RFC2791, July 2000, <<http://www.rfc-editor.org/info/rfc2791>>.
- [RFC3633] Troan, O. and R. Droms, "IPv6 Prefix Options for Dynamic Host Configuration Protocol (DHCP) version 6", RFC 3633, DOI 10.17487/RFC3633, December 2003, <<http://www.rfc-editor.org/info/rfc3633>>.

- [RFC4724] Sangli, S., Chen, E., Fernando, R., Scudder, J., and Y. Rekhter, "Graceful Restart Mechanism for BGP", RFC 4724, DOI 10.17487/RFC4724, January 2007, <<http://www.rfc-editor.org/info/rfc4724>>.
- [RFC4760] Bates, T., Chandra, R., Katz, D., and Y. Rekhter, "Multiprotocol Extensions for BGP-4", RFC 4760, DOI 10.17487/RFC4760, January 2007, <<http://www.rfc-editor.org/info/rfc4760>>.
- [RFC4786] Abley, J. and K. Lindqvist, "Operation of Anycast Services", BCP 126, RFC 4786, DOI 10.17487/RFC4786, December 2006, <<http://www.rfc-editor.org/info/rfc4786>>.
- [RFC6769] Raszuk, R., Heitz, J., Lo, A., Zhang, L., and X. Xu, "Simple Virtual Aggregation (S-VA)", RFC 6769, DOI 10.17487/RFC6769, October 2012, <<http://www.rfc-editor.org/info/rfc6769>>.
- [RFC6830] Farinacci, D., Fuller, V., Meyer, D., and D. Lewis, "The Locator/ID Separation Protocol (LISP)", RFC 6830, DOI 10.17487/RFC6830, January 2013, <<http://www.rfc-editor.org/info/rfc6830>>.
- [RFC7454] Durand, J., Pepelnjak, I., and G. Doering, "BGP Operations and Security", BCP 194, RFC 7454, DOI 10.17487/RFC7454, February 2015, <<http://www.rfc-editor.org/info/rfc7454>>.
- [RFC7695] Pfister, P., Paterson, B., and J. Arkko, "Distributed Prefix Assignment Algorithm", RFC 7695, DOI 10.17487/RFC7695, November 2015, <<http://www.rfc-editor.org/info/rfc7695>>.
- [RFC7938] Lapukhov, P., Premji, A., and J. Mitchell, Ed., "Use of BGP for Routing in Large-Scale Data Centers", RFC 7938, DOI 10.17487/RFC7938, August 2016, <<http://www.rfc-editor.org/info/rfc7938>>.
- [I-D.herbert-nvo3-ila]
Herbert, T., "Identifier-locator addressing for IPv6", draft-herbert-nvo3-ila-03 (work in progress), October 2016.
- [I-D.lapukhov-bgp-opaque-signaling]
Lapukhov, P., Aries, E., Marques, P., and E. Nkposong, "Use of BGP for Opaque Signaling", draft-lapukhov-bgp-opaque-signaling-02 (work in progress), April 2016.

- [I-D.ietf-v6ops-dc-ipv6]
Lopez, D., Chen, Z., Tsou, T., Zhou, C., and A. Servin,
"IPv6 Operational Guidelines for Datacenters", draft-ietf-
v6ops-dc-ipv6-01 (work in progress), February 2014.
- [I-D.lapukhov-bgp-ila-afi]
Lapukhov, P., "Use of BGP for dissemination of ILA mapping
information", draft-lapukhov-bgp-ila-afi-01 (work in
progress), March 2016.
- [I-D.ietf-grow-bmp]
Scudder, J., Fernando, R., and S. Stuart, "BGP Monitoring
Protocol", draft-ietf-grow-bmp-17 (work in progress),
January 2016.
- [I-D.ietf-nvo3-arch]
Black, D., Hudson, J., Kreeger, L., Lasserre, M., and T.
Narten, "An Architecture for Data Center Network
Virtualization Overlays (NVO3)", draft-ietf-nvo3-arch-08
(work in progress), September 2016.
- [ED25519] "Ed25519: high-speed high-security signatures",
<<https://ed25519.cr.yt.to>>.
- [ETCD] "coreos/etcd", <<https://github.com/coreos/etcd>>.
- [MEMCACHED]
"Memcached", <<https://memcached.org/>>.
- [ROUTED-DESIGN]
"High Availability Campus Network Design", 2008, <<http://www.cisco.com/c/en/us/td/docs/solutions/Enterprise/Campus/routed-ex.html>>.
- [LINUX-NAMESPACES]
"Namespaces in operation, part 1: namespaces overview",
2013, <<https://lwn.net/Articles/531114/>>.
- [IPVLAN] "IPVLAN Driver HOWTO", 2013,
<[https://github.com/torvalds/linux/blob/master/
Documentation/networking/ipvlan.txt](https://github.com/torvalds/linux/blob/master/Documentation/networking/ipvlan.txt)>.

Author's Address

Petr Lapukhov
Facebook
1 Hacker Way
Menlo Park, CA 94025
US

Email: petr@fb.com