Identifier Locator Addressing (ILA)

Introduction

Identifier-locator addressing (ILA) is a technique used with IPv6 that 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.

ILA can be thought of as means to implement an overlay network without encapsulation. This is accomplished by performing network address translation on destination addresses as a packet traverses a network. To the network, an ILA translated packet appears to be no different than any other IPv6 packet. For instance, if the transport protocol is TCP then an ILA translated packet looks like just another TCP/IPv6 packet. The advantage of this is that ILA is transparent to the network so that optimizations in the network, such as ECMP, RSS, GRO, GSO, etc., just work.

The ILA protocol is described in Internet-Draft draft-herbert-intarea-ila.

ILA terminology

- Identifier
  A number that identifies an addressable node in the network independent of its location. ILA identifiers are sixty-four bit values.
- Locator
  A network prefix that routes to a physical host. Locators provide the topological location of an addressed node. ILA locators are sixty-four bit prefixes.
- ILA mapping
  A mapping of an ILA identifier to a locator (or to a locator and meta data). An ILA domain maintains a database that contains mappings for all destinations in the domain.
- SIR address
  An IPv6 address composed of a SIR prefix (upper sixty- four bits) and an identifier (lower sixty-four bits). SIR addresses are visible to applications and provide a means for them to address nodes independent of their location.
- ILA address
  An IPv6 address composed of a locator (upper sixty-four bits) and an identifier (low order sixty-four bits). ILA addresses are never visible to an application.
- 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 forwarding cache
  A type of ILA router that only maintains a working set cache of mappings.
- ILA node
  A network node capable of performing ILA translations. This can be an ILA router, ILA forwarding cache, or ILA host.

Operation

There are two fundamental operations with ILA:

- Translate a SIR address to an ILA address. This is performed on ingress to an ILA overlay.
- Translate an ILA address to a SIR address. This is performed on egress from the ILA overlay.

ILA can be deployed either on end hosts or intermediate devices in the network; these are provided by "ILA hosts" and "ILA routers" respectively. Configuration and datapath for these two points of deployment is somewhat different.

The diagram below illustrates the flow of packets through ILA as well as showing ILA hosts and routers:

```
    +--------+                                            +--------+
    | Host A +-+                                     +--->| Host B |
    |        | |            (2) ILA                 (')   |        |
    +--------+ |          ...addressed....         (   )  +--------+
               V +---+--+ .  packet      . +---+--+ (_)
   (1) SIR     | | ILA  |----->--------->---->| ILA  |  |   (3) SIR
```

```
addressed  +->|router|  .              .  |router|->-+    addressed
packet        +---+--+  .     IPv6     .  +---+--+        packet
                 /      .    Network   .
                /       .              .   +--+-++--------+
+--------+   /          .              .   |ILA ||  Host  |
|  Host  +--+           .              .- -|host||        |
|        |              .              .   +--+-++--------+
+--------+              ................
```

Transport checksum handling

When an address is translated by ILA, an encapsulated transport checksum that includes the translated address in a pseudo header may be rendered incorrect on the wire. This is a problem for intermediate devices, including checksum offload in NICs, that process the checksum. There are three options to deal with this:

- no action Allow the checksum to be incorrect on the wire. Before
  a receiver verifies a checksum the ILA to SIR address translation must be done.
- adjust transport checksum
  When ILA translation is performed the packet is parsed and if a transport layer checksum is found then it is adjusted to reflect the correct checksum per the translated address.
- checksum neutral mapping
  When an address is translated the difference can be offset elsewhere in a part of the packet that is covered by the checksum. The low order sixteen bits of the identifier are used. This method is preferred since it doesn't require parsing a packet beyond the IP header and in most cases the adjustment can be precomputed and saved with the mapping.

Note that the checksum neutral adjustment affects the low order sixteen bits of the identifier. When ILA to SIR address translation is done on egress the low order bits are restored to the original value which restores the identifier as it was originally sent.

Identifier types

ILA defines different types of identifiers for different use cases.

The defined 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

In the current implementation of kernel ILA only locally unique identifiers (LUID) are supported. LUID allows for a generic, unformatted 64 bit identifier.

Identifier formats

Kernel ILA supports two optional fields in an identifier for formatting: "C-bit" and "identifier type". The presence of these fields is determined by configuration as demonstrated below.

If the identifier type is present it occupies the three highest order bits of an identifier. The possible values are given in the above list.

If the C-bit is present, this is used as an indication that checksum neutral mapping has been done. The C-bit can only be set in an ILA address, never a SIR address.

In the simplest format the identifier types, C-bit, and checksum adjustment value are not present so an identifier is considered an unstructured sixty-four bit value:

```
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                            Identifier                         |
+                                                               +
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

The checksum neutral adjustment may be configured to always be present using neutral-map-auto. In this case there is no C-bit, but the checksum adjustment is in the low order 16 bits. The identifier is still sixty-four bits:

```
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                            Identifier                         |
|                               +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

```
|                               |  Checksum-neutral adjustment  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

The C-bit may used to explicitly indicate that checksum neutral mapping has been applied to an ILA address. The format is:

```
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     |C|                    Identifier                         |
|     +-+                       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                               |  Checksum-neutral adjustment  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

The identifier type field may be present to indicate the identifier type. If it is not present then the type is inferred based on mapping configuration. The checksum neutral adjustment may automatically used with the identifier type as illustrated below:

```
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Type|                      Identifier                         |
+-+-+-+                         +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                               |  Checksum-neutral adjustment  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

If the identifier type and the C-bit can be present simultaneously so the identifier format would be:

```
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Type|C|                    Identifier                         |
+-+-+-+-+                       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                               |  Checksum-neutral adjustment  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Configuration

There are two methods to configure ILA mappings. One is by using LWT routes and the other is ila_xlat (called from NFHOOK PREROUTING hook). ila_xlat is intended to be used in the receive path for ILA hosts .

An ILA router has also been implemented in XDP. Description of that is outside the scope of this document.

The usage of for ILA LWT routes is:

ip route add DEST/128 encap ila LOC csum-mode MODE ident-type TYPE via ADDR

Destination (DEST) can either be a SIR address (for an ILA host or ingress ILA router) or an ILA address (egress ILA router). LOC is the sixty-four bit locator (with format W:X:Y:Z) that overwrites the upper sixty-four bits of the destination address. Checksum MODE is one of "no-action", "adj-transport", "neutral-map", and "neutral-map-auto". If neutral-map is set then the C-bit will be present. Identifier TYPE one of "luid" or "use-format." In the case of use-format, the identifier type field is present and the effective type is taken from that.

The usage of ila_xlat is:

ip ila add loc_match MATCH loc LOC csum-mode MODE ident-type TYPE

MATCH indicates the incoming locator that must be matched to apply a the translaiton. LOC is the locator that overwrites the upper sixty-four bits of the destination address. MODE and TYPE have the same meanings as described above.

Some examples

```
# Configure an ILA route that uses checksum neutral mapping as well
# as type field. Note that the type field is set in the SIR address
# (the 2000 implies type is 1 which is LUID).
ip route add 3333:0:0:1:2000:0:1:87/128 encap ila 2001:0:87:0 \
     csum-mode neutral-map ident-type use-format

# Configure an ILA LWT route that uses auto checksum neutral mapping
# (no C-bit) and configure identifier type to be LUID so that the
# identifier type field will not be present.
ip route add 3333:0:0:1:2000:0:2:87/128 encap ila 2001:0:87:1 \
     csum-mode neutral-map-auto ident-type luid

ila_xlat configuration

# Configure an ILA to SIR mapping that matches a locator and overwrites
# it with a SIR address (3333:0:0:1 in this example). The C-bit and
# identifier field are used.
ip ila add loc_match 2001:0:119:0 loc 3333:0:0:1 \
    csum-mode neutral-map-auto ident-type use-format

# Configure an ILA to SIR mapping where checksum neutral is automatically
# set without the C-bit and the identifier type is configured to be LUID
# so that the identifier type field is not present.
ip ila add loc_match 2001:0:119:0 loc 3333:0:0:1 \
    csum-mode neutral-map-auto ident-type use-format
```