Moonshot: Implementing the Next Generation of Network Telemetry Technologies

Andy Gospodarek
Broadcom Corporation
gospo@broadcom.com

Abstract

Current network monitoring and telemetry applications require host-based collectors across all nodes in a network (both on servers/hypervisors and traditional switches and routers). These can be effective solutions, but just as datacenter deployment patterns have evolved new technology to track traffic as it moves through the network had emerged. Newer specifications like Inband Network Telemetry (INT) [2] and Inband Flow Analyzer (IFA) [1] propose standards to add metadata to packets or clone and add metadata as they flow through a network to allow collectors/agents to gather data at the network edges. Hardware that supports INT/IFA can add metadata automatically with application/flowlevel/virtual-port granularity which allows more detailed network monitoring and assurance to customers that service levels for applications are being met.

1 Introduction to Network Telemetry

INT and IFA are all designed create a generic method of reporting and collecting network state information on individual flows as the packets traverse a network. This allows for collection of data from individual hosts or applications as frames that are part of those flows are marked with telemetry headers as they entry a telemetry domain. Network devices can interpret telemetry header fields as *telemetry instructions* and a capable device will update packet headers and header-fields *In-Situ* – as a frame traverses the network. Marking frames as they travel through a network allows detailed reporting of the exact data-plane used by packets on the network as well as enables real-time feedback loops and event detection. This information can also be sent to an external collector for post-processing if desired.

1.1 Network Telemetry Components

Despite using slightly different nomenclature, the fundamental components of the two main telemetry technolgies covered in the paper are similar.

1.1.1 Source or Initiator Node

This is a trusted entity that creates the initial telemetry header and places it into packets that are transmitted.

1.1.2 Transit Hop or Transit Node

Any network element that adds telemetry metadata to a packet that that contains supported telemetry instructions.

1.1.3 Sink or Terminating Node

This is a trusted entity that removes telemetry headers from frames to make the existence of the the headers transparent to applications. This trusted entity will use the headers and other local configuration to determine if information needs to be sent to a collector.

1.1.4 Collector

An application that will receive telemetry data collected by a Sink or Terminating node.

1.1.5 Typical Packet Flow

Below is a diagram of a typical packet path of a network flow and IFA flow through the components described in the previous section. In this case the IFA flow is a sample of the flow, so two frames travel between the Initiator Node and the Terminating Node.

This was adapted from the latest IFA specification and the time fo this writing:

```
+----------+
|          |
|Collector |<-------------+
|          |              |
+----------+              |
                          |
                          |
                          |
                          |
                          |
                          |
                          |
                                                        |
     +--------------+        +------------+        +----+-----------+
     |Initiator Node|        |Transit Node|        |Terminating Node|
flow |   +------+   |  flow  |  +------+  |  flow  |     +------+   | flow
---->|   | IFA  |   |------->|  | IFA  |  |------->|     | IFA  |   |---->
     |   +------+   |IFA flow|  +------+  |IFA flow|     +------+   |
     +--------------+        +------------+        +----------------+
```

In the case where an Initiator and Terminating nodes are switches or other forwarding elements on a network, flows could originate from an external device and exit to another device. Initiator and Terminating Nodes could also be servers with supported hardware and software stacks. In that case flows may originate or teminate on the IFA/INT node rather than originating/terminating from an external device.

This paper does not intend to cover the full scope of each telemetry technology and feature; anyone who would like to learn more should consider reading the latest INT and IFA specifications. It will cover some basic frame formats of each proposal in order to provide context for an implementation discussion.

2 Inband Network Telemetry (INT)

Inband Network Telemetry is a framework suggested by those interested in using P4 to create a programmable pipeline for networking forwarding elements. INT has multiple methods for collecting information about the network: frames are updated as they traverse the network or special *probe packets* are used to collect information about the network. In addition to defining the frame format and fields, the latest INT specification also conveniently provides a P4 program specification for INT Transmit.

2.0.1 INT Frame Format

The current INT specification describes the following formats as being able to support additional encapsulation headers to support INT:

- INT over VXLAN (as VXLAN payload, per GPE extension)
- INT over Geneve (as Geneve option)
- INT over GRE (as a shim between GRE header and encapsulated payload)
- INT over NSH (as NSH payload)

Additionally the INT specification also describes how DSCP bits or *probe markers* can be placed in the payload of packet (after the Layer4 header) to support these packet formats.

- INT over TCP (as payload)
- INT over UDP (as payload)

Though many datacenter networks use encapsulated traffic (VXLAN, Geneve, or GRE), the fact that INT does not have native support unencapsulated traffic (standard IPv4/IPv6 and TCP/UDP) could be an issue for some deployments.

The specified frame format for an INT IPv4/TCP frame would be as follows:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                        IP Header                              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                         Layer 4                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                      INT Shim Header                          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                    INT Metadata Header                        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                         Payload                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Remember that the INT Headers and Payload together are viewed as the full payload to any non-INT-aware device, so anytime INT headers are added to a packet any fields that account for the size of the packet or payload will need to be adjusted.

If the decision to use a reserved DSCP mark (0x17 in this case) to indicate a packet contained INT headers would cause the IPv4 header to look like this:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
IPv4 Header:
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Ver=4 | IHL=5 | DSCP=0x17 |ECN|          Total Length         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|         Identification        |Flags|      Fragment Offset    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Time to Live | Protocol = TCP|         Header Checksum       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                     Source IPv4 Address                       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                  Destination IPv4 Address                     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

The INT specification also outlines suggestions for how to deal with frames as they grow beyond the MTU, how to deal with false detection of *probe markers* contained in payload data of non-INT frames, as well as other deployment scenarios.

3 Inband Flow Analyzer (IFA)

The initial IFA specification was drafted later than other initial telemetry technologies and while similar, it aims to address some of the shortcomings of INT and IOAM. One of the main differences is the ability to send telemetry metadata via a cloned frame rather than via the original datagram.

Allowing cloned frames provides benefits over In-Situ modification of frames. One benefit of cloning is administrators do not need to be concerned about frames growing beyond the MTU since there is also support to allow truncation of frames that are beyond the size of the MTU. The IFA specification also indicates that adding metadata to live traffic is a requirement but this cloning feature is a nice addition to avoid disruption of PMTU discovery.

The proposed frame/header format was modified significantly from the INT specification. The goal was to make the frame format more acceptable to devices that were not IFA-aware.

3.0.1 IFA Frame Format

The IFA spec outlines a significantly different scheme for the location to telemetry metadata. From the start IFA aims to interoperate with encapsulated and unencapsulated IPv4 and IPv6 traffic. This is accomplished by using the IPv4 *Protocol* and IPv6 *Next Header* fields to specify that this frame is an IFA frame. (There is no current reservation for IFA protocol, so testing currently uses one of the experimental protocol numbers.)

The specified frame format for an IFA IPv4/TCP frame would be as follows:

```
  0                   1                   2                   3
  0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |                          IP Header                            |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |                         IFA Header                            |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |                          Layer 4                              |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |                          Payload                              |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |                     IFA Metadata Stack                        |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |                     IFA Metadata Header                       |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

A closer look at the IPv4 header demonstrates that Protocol=IFA would be used to signal that this frame is an IFA frame:

```
  0                   1                   2                   3
  0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
 IPv4 Header:
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |Version|  IHL  |Type of Service|          Total Length         |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |         Identification        |Flags|      Fragment Offset    |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |  Time to Live | Protocol = IFA|         Header Checksum       |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |                       Source IPv4 Address                     |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |                    Destination IPv4 Address                   |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Additionally the IFA Header provides a Next Header field that would indicate that TCP is the next protocol:

```
  0                   1                   2                   3
  0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
 IFA Header:
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |Ver=2.0|  GNS  | NextHdr = TCP |R|R|R|M|T|I|T|C|   Max Length  |
 |       |       |               | | | |F|S| |A| |               |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

4 Software Architecture

In order to add support for a software model for INT or IFA, it is important to understand the discrete componets available within the GNU/Linux ecosystem that could be used with or without modification to add support for INT or IFA. What follows is a list of each of the nodes in a telemetry implementation, the functional blocks needed to implement those features, and what existing kernel components could be the target for performing those functions.

4.1 Initiator Node

The critical datapath operations of sample and mirror (IFA) or sample and redirect(INT/IFA) are the most important part of the Initiator Node implementation. If these are avaiable, the packet encapsulation and transmission could be easily implmented in userspace by an application listening to nflog messages.

Table 1: Initiator Node

Desired Functionality	**Kernel Implementation**
Packet Sampling	TC or netfilter
Packet Mirror(IFA)	TC
Packet Redirect(INT/IFA)	TC or XDP/eBPF
Encapsulate INT/IFA	lwtunnel or XDP/eBPF

4.2 Transit Node

For a device to function as a Transit Node there must be support to recognize the frame in the kernel datapath. Once the frame is recognized one could either pass the traffic to a lwtunnel or eBPF program for modification.

Table 2: Transit Node

Desired Functionality	**Kernel Implementation**
Match on INT/IFA Packet	TC
Update metadata and Transmit	lwtunnel or XDP/eBPF

4.3 Collector Node

The ability to match on all of the formats (or the enabled formats used in a deployment) is the critical role of a Collector Node. Without this functionality in TC or netfilter a system running the Linux kernel would not be able to serve as a collector.

Table 3: Collector Node

Desired Functionality	**Kernel Implementation**
Match on INT/IFA Packet	TC or netfilter
Collect metadata from frame	lwtunnel or XDP/eBPF
Send frame to collector	lwtunnel or XDP/eBPF
Transmit original frame (INT)	lwtunnel or XDP/eBPF

5 Hardware Requirements

[Hardware requirements/implementations for handling IFA/INT today and the challenges facing those wanting to implement them in hardware and software.]

6 Possible Configuration Methods

[Proposals for configuring INT/IFA in both software dataplane and hardware dataplane environments on supported hardware.]

6.1 Host-based configuration

The standard configuration method that most will desire is one that is configured on the host via standard interfaces that are used to communicate between userspace and hardware. Several options exist to accomplish this.

Fundamentally it will go like this:
Create IFA device or tunnel/encap

```
ip link set -> kernel -> nl broadcast -> bnxt\_en -> configure hardware
```

Sample and redirect traffic to tunnel/ifa device

```
tc sample/redirect -> kernel -> bnxt\_en -> configure hardware
```

The kernel interfaces to receive the netlink messages (lwt seems like the best option right now) and the driver work to process the nl messages for IFA and add the parameters to hardware.

6.2 Network-based configuration

There are also use-cases where datacenter administrators will want to enable IFA on a host without requiring communication with the server OS. This could be done via configuration commands on a BMC or management processor or directly with a NIC itself. While this method is traditionally seen as less popular to those within the open-source community who expect to have exclusive control over their hardware, industry deployment trends indicate that this is a popular and prefereed method for many.

The main use-case for network-based configuration is in datacenters where baremetal servers are provided to users. In these deployments network adminstrators still desire the ability to interact more directly with network devices. Servers acting as an INT Source or IFA Initiator Node could make the lives of network adminstrators easier.

A fairly RESTful API could be used to control the INT/IFA configuration of a server much the same way that such APIs are often used to control traditional networking equipment.

7 Telemetry Deployment Risks

7.1 User Risk

Obviously the deployment of any new technology brings along with it some risk. Even looking at the changes between the first two versions of IFA[1] highlights that this is an evolving standard with the addition of new metadata formats and an increased list of packet formats. The risk is minimized when a single organization controls all networking infrastructure.

7.2 Community Risk

One of the major risks to adding new code to the Linux kernel is what happens to existing code when standards change. For users of upstream kernels this is not that difficult as a `git pull` and `make ...` can result in a code updated to the latest standard.

Unfortunately a standard like INT or IFA will may also require changes to other network infrastructure to handle these changes and not everyone may have access to the latest hardware running the latest versions of software. As mentioned in the previous section, unfied management of infrastructure will prevent his from being a major issue.

8 Acknowledgments

Special thanks to all those invoved in IFA and INT specs as well as those at Broadcom who did not particiapte in the writing of the standard, but did provide code, feedback, etc during this process.

References

[1] J. KUMAR, E. A. Inband Flow Analyzer. *https://tools.ietf.org/pdf/draft-kumar-ippm-ifa-01.pdf*.

[2] KIN CHANGHOON, E. A. In-band Network Telemetry (INT). *https://p4.org/assets/INT-current-spec.pdf*.

Notes

[1] https://tools.ietf.org/rfcdiff?url2=draft-kumar-ippm-ifa-01.txt