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 (servers/hypervisors and traditional switches/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 has emerged. Newer specifications like Inband Network Telemetry[3] (INT) and Inband Flow Analyzer[4] (IFA) 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 path 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 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 instructions and places them into packets that are transmitted. One important note is that this node is expected to only add an initial telemetry header to a small, sampled percentage of the traffic transmitted.

1.1.2 Transit Hop or Transit Node

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

1.1.3 Sink or Terminating Node

This is a trusted entity that removes all telemetry headers, instructions, and metadata from in-situ frames or drops out-of-band frames to make the existence of telemetry applications 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. This node will then format and deliver the frame to the Collector node or application.

1.1.4 Collector

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

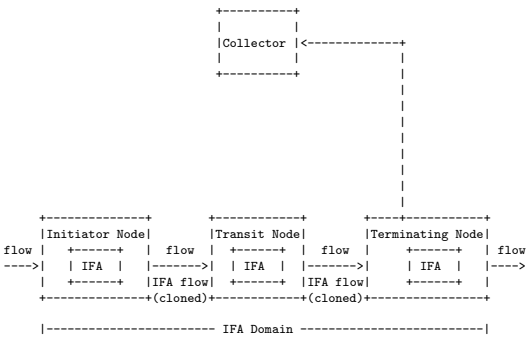
1.1.5 Telemetry Domain

A set of interconnected devices participating in telemetry actions all under the same administrative control. It is critical that devices in the same domain are all configured to behave in a consistent manner.

1.1.6 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 at the time of this writing:



Initiator and Terminating Nodes could be servers with supported hardware and software stacks or switches, routers, or other forwarding elements on a network. The flows may originate or teminate on the IFA/INT node if that node is on the edge of the IFA Domain. If the network extends beyond the IFA Domain, flows originate or terminate outside the IFA Domain.

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. This paper 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:

- A sampled percentage of the frames are updated as they traverse the network
- New *probe packets* are injected and used to collect telemetry information as they traverse 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 for hardware or software that can use a P4 datapath. Support for INT is not limited to devices using a programmable datapath like P4.

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:

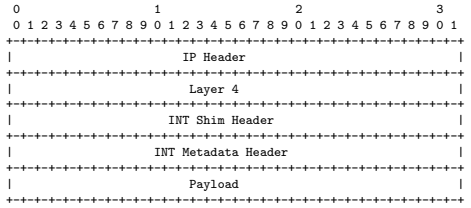
- INT over VXLAN (as VXLAN payload, per GPE extension)
- INT over Geneve (as Geneve option)
- INT over GRE (as a shim between GRE header and encapsulated payload)
- INT over NSH (as NSH payload)

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)

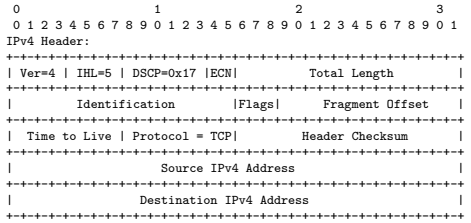
Though many datacenter networks use encapsulated traffic (VXLAN, Geneve, or GRE), the fact that INT does not have native support unencapsulated traffic (standard IPv4/IPv6 and TCP/UDP) could be an issue for some deployments.

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



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:



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. One of the main differences is the ability to send telemetry metadata out-of-band via cloned frames rather than via the original datagram. Like INT however, IFA supports adding metadata to live traffic in-band.

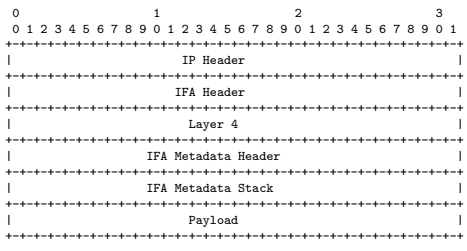
Allowing cloned frames provides benefits over in-band modification of frames. One benefit of cloning is administrators do not need to be concerned about frames growing beyond the MTU. As expected frame payloads will be truncated when needed.

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.

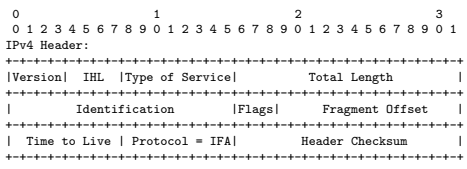
3.0.1 IFA Frame Format

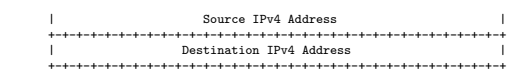
The IFA spec outlines a significantly different scheme for the location to telemetry metadata. From the start IFA aims to interoperate with encapsulated and unencapsulated IPv4 and IPv6 traffic. This is accomplished by using the IPv4 *Protocol* and IPv6 *Next Header* fields to specify that this frame is an IFA frame. There is no current IANA reservation for IFA protocol, so testing should use one of the experimental protocol numbers as described by RFC3692[1].

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

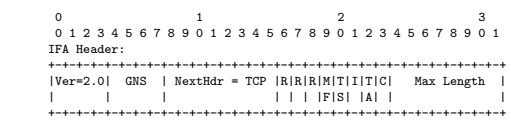


A closer look at the IPv4 header demonstrates that Protocol=IFA would be used to signal that this frame is an IFA frame:





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



4 Linux Kernel 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 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.

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 and TX 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 TX	lwtunnel or XDP/eBPF

4.3 Terminating Node

The ability to match on all of the formats (or the enabled formats used in a deployment) is the critical role of a Terminating 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: Terminating Node

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

4.4 Implementation Summary

The largest volume of work to support INT/IFA in the Linux kernel is included in the TC (specifically related to sampling) and lwtunnel subsystem. While an initial proof-of-concept could be done with eBPF, long-term it seems like use of the lwtunnel infrastructure would be better.

Use of the kernel for performing INT/IFA encapsulation could be seen by some as more work than necessary. Some would argue that this could be solved with a simple userspace daemon. The need to support in-band encapsulation of packets means that the Linux kernel must be aware of this encapsulation.

The desire to add support to configure hardware that is capable of INT/IFA offload also means that this functionality **must** be included in the Linux kernel.

5 Beyond the Linux Kernel Datapath

5.1 Hardware Support for Telemetry

As with any new standards-based technology, manufacturers of networking hardware will make hardware capable of INT and/or IFA when there is significant customer demand. Existing devices with programmable pipelines seem like the primary target, but fixed function devices could also support for IFA/INT if there is enough demand.

5.2 Network-based configuration

There are also use-cases where datacenter administrators will want to enable IFA on a server 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. 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 Source or Initiator Node could make the lives of network adminstrators easier. As NIC hardware that supports INT/IFA emerges configuration in this manner will gain popularity.

A 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. There may ultimately be a common framework that exists to enable INT/IFA on a server or switch with the same API.

5.3 Complete Userspace Implementations

No discussion about new datapath technologies would be complete without addressing whether or not this could be implemented with a poll mode driver like DPDK[2]. It is unlikely that anyone would implement or deploy a DPDK-based application on a server or switch for the sole purpose of performing any of the functions needed for INT/IFA.

It is likely, however, that someone using a DPDK-based application for performing forwarding in the core of a network may want to add INT/IFA functionality to their application. A library that could create, add metadata, or terminate telemetry packets could be written to integrate with any dataplane applcation. At the time of this writing no known library or solution exists.

6 Telemetry Development and Deployment

6.1 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. One way to minimize risk is to be sure that any devices participating are all managed/controlled by the same organization.

6.2 Community Risk

One of the major risks to adding new code to the Linux kernel or any dataplane application is what happens to existing code when standards change. For users of upstream kernels or who build their own applications this is not that difficult – a `git pull` and `make ...` can result in a code updated to the latest standard supported by that project.

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

Unfortunately a standard like INT or IFA will may also require changes to other network infrastructure to handle these changes and not all administrators 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.

7 Acknowledgments

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

References

[1] Assigning Experimental and Testing Numbers Considered Useful. `https://datatracker.ietf.org/doc/rfc3692/`. Accessed: 2019-03-15.

[2] Data Pane Development Kit. `https://www.dpdk.org/`. Accessed: 2019-03-15.

[3] In-band Network Telemetry (INT). `https://p4.org/assets/INT-current-spec.pdf`. Accessed: 2019-02-28.

[4] Inband Flow Analyzer. `https://tools.ietf.org/pdf/draft-kumar-ippm-ifa-01.pdf`. Accessed: 2019-02-28.