White Paper

In-band Network Telemetry Detects Network Performance Issues

In-band Network Telemetry (INT) uses data planes or smart devices to insert and collect per packet telemetry data for granular, highly accurate network visibility and improved reliability

Table of Contents

Introduction

Using network telemetry to provide visibility into network performance and health is critical to the management of hyper-scale cloud, high-throughput enterprise, and communications service provider (CoSP) networks that deliver real-time services with measurable reliability. In these environments, visibility into data flow and network conditions is essential to operate and manage the networking elements and meet service-level requirements.

In today's networks, packet mirroring is used to collect data for telemetry analysis. This requires setting up a parallel network packet broker (NPB) network to collect the data and forward it to specialized applications on dedicated servers to analyze the packets. Due to this infrastructure, this solution is expensive to scale and collects too few packets to provide a detailed understanding of the issues facing high-performance production networks (Figure 1). This is especially true at modern link speeds >25Gbps.

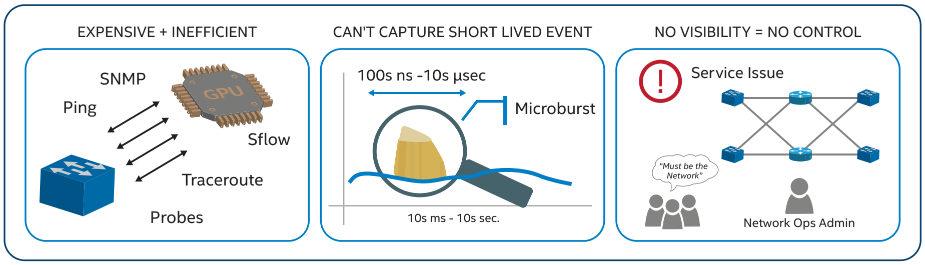

Figure 1. Today's Network Monitoring Pain Points

INT is an open framework managed by the open-source project, P4, which allows the data plane to export telemetry data without the intervention of the control plane. Using INT, network operators can answer four questions about data packets:

- Which path did this packet take?
- Which rules did this packet follow?
- How long did this packet queue at each switch?
- With what other packets did this packet share queues?

With INT, network elements including switch ASICs, virtual switches (vSwitches), or SmartNICs, can track packets through a network by inserting instructions to collect network state metadata into the packet as it traverses the network. Consequently, INT enables unprecedented visibility into network states that each data packet experiences, helping detect throughput issues due to bottlenecks, failures, or configuration errors. The INT mechanism is also being adapted to facilitate an edge-to-edge telemetry mode called, Host-enabled INT, which provides a broader dataset that adds value for both large network operators, as well as application developers.

Fine grain, per-packet measurement of all traffic flows (e.g. compute, storage, RDMA) delivers issue detection at a granular level with INT. Collecting metadata from every packet allows monitoring tools to measure real-time network performance against network performance baselines (path and latency). Further, they can match patterns of behavior and correlate phenomena, empowering network owners to quickly react to network issues and/or anticipate problems before they arise.

This paper details frameworks for both switch-based INT and Host-based INT, starting first with an overview of P4 network programmability, which is the foundation of INT.

Overview of P4

P4 is an open-source programming language managed by the P4 project under the Open Networking Foundation (ONF), that is used to customize programmable data planes in conjunction with SDN control protocols, like OpenFlow. Packet-processing applications written in P4 describe how packets are forwarded and enables these network elements to be programmed using custom or optimized protocols to enable "top-down" control of data flows through the network element. The goals of P4 include:

- Protocol Independence: P4 programs can use any protocol to specify how a switch processes packets
- Target Independence: P4 is suitable for describing everything from high-performance forwarding ASICs to software switches
- Field Reconfigurability: P4 allows network engineers to change the way their switches process packets after they are deployed

P4 Runtime

Deployed P4 elements are optionally configured and changed using P4 Runtime, a control plane specification for controlling the data plane. P4 Runtime is open source and silicon-independent and provides a standard way to control a P4 network element, allowing a network manager to add or delete entries to a networking element's forwarding tables. P4 Runtime is designed for both remote and local control planes and can be operated with or without a remote procedure call (RPC).

INT Framework

INT is a framework designed to allow the collection and reporting of network state by the data plane, without requiring intervention or work by the control plane.

INT utilizes a data plane to match on a particular network flow and executes specific instructions enabling the device to collect the identified data flows. INT endpoints (applications, host networking stack, hypervisor, Network Interface Card (NIC), send-side top of rack switches and others) classify network traffic based on various packet header fields and insert or apply INT instructions. The INT architecture is designed for both normal data packets, cloned packets from a network test access point (TAP), and special probe packets – the last two to accommodate legacy network analysis architectures.

INT uses various header encapsulation mechanisms that enable metadata to be inserted, in-band, into IP data packets by network devices. Figure 2 shows an example of one way a packet might carry the INT metadata. In application, INT metadata is inserted as the packet passes through a forwarding node. The metadata could be, but is not limited to, ingress port ID, ingress timestamp, egress port ID, hop latency, egress port TX Link utilization, or others

Change Detector

For certain INT modes of operation, change detector functionality monitors every data packet against predetermined conditions such as performance thresholds or hysteresis, and reports only on the packets that meet the reporting conditions. For example, queueing latency above a threshold and exhibiting a large enough change against the previously reported value is worthy of a report. Because it minimizes the number of telemetry reports that need to be generated, change detector allows the INT function to scale and support higher-speed networks. Reporting conditions can include flow initiation, flow termination, changes to latency, or any custom programmed value. Change detector's ability to divert packet flows without the latency of a round-trip to an external controller allows INT to analyze every packet at all throughput levels.

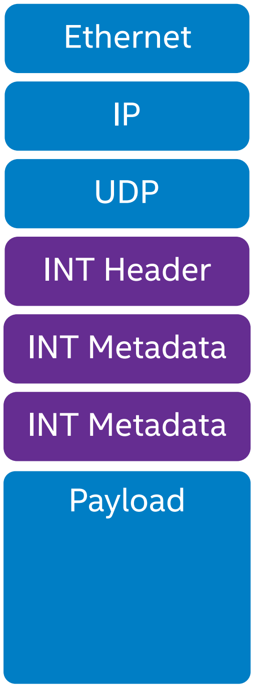

Figure 2. Example INT Packet

How INT Works

There are three key parts to an INT deployment as seen in Figure 3. The INT source is where INT data tracking metadata and instructions are added to the packets in a data flow. Some of the most common data collected include switch ID, arrival time, queue delay, matched rules and more.

As the packets traverse the network, they arrive at forwarding elements that, for INT purposes, are called INT transit nodes. These nodes read the instructions and add the required data to the packet headers.

The INT sink is the node where the metadata that's been collected from the packets is exported and sent for analysis. Some examples of traffic sink behavior include:

- Report to operations, administration, and management (OAM) systems: The INT sink exports its collected state metadata to an external OAM controller
- Real-time control or feedback loops: The collected metadata can be used by a controller to send feedback through upstream network elements to make changes to traffic engineering schemes or forward packets to fix network congestion
- Network event detection: The INT nodes themselves can generate feedback information if INT path data indicates a network fault that requires immediate attention or resolution. This could include severe congestion or violation of certain data plane invariances.

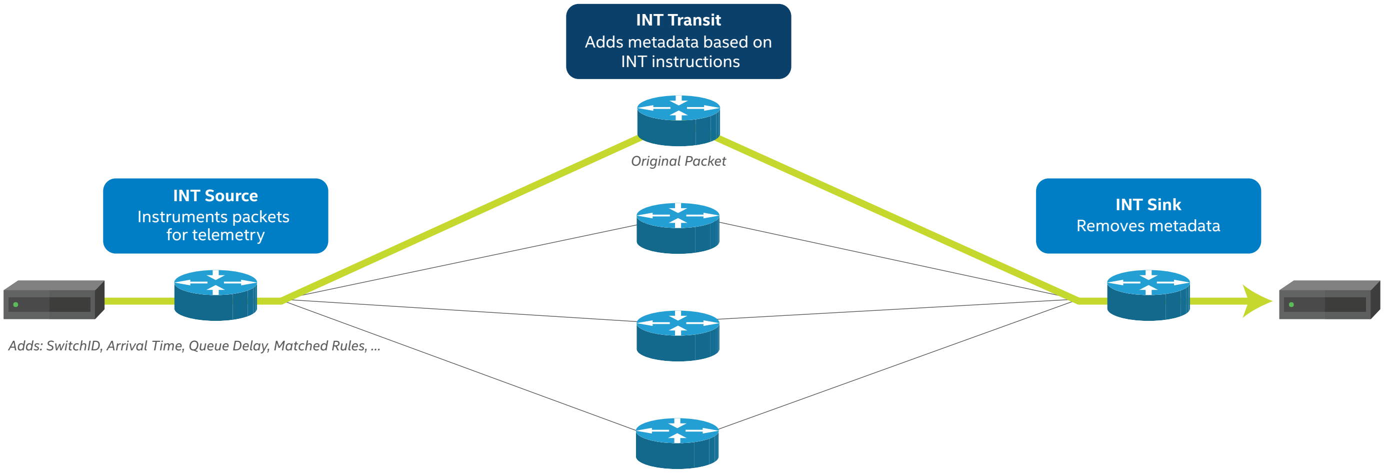
INT Transit
Adds metadata based on INT instructions
Original Packet
INT Source
Instruments packets for telemetry
INT Sink
Removes metadata
Adds: SwitchID, Arrival Time, Queue Delay, Matched Rules, ...

Figure 3. The role that INT Source, INT Transit and INT Sink play in an INT deployment.

INT operates in three modes of operation either from real data packets or packets that come from legacy probes or TAPS (synthetic packets). i.e. (Figure 4)

The modes reflect whether the data is added to real or synthetic packets and also how much information is embedded in the packet. The three operational modes include:

INT-XD (eXport Data): A data plane table, known as a flow watchlist, is established at each INT element. The table supports matching on packet headers and inserting or applying INT instructions on each matched flow. In INT-XD mode, network devices directly export metadata from their data plane to the monitoring system based on the INT instructions configured in their flow watchlists without modifying the packet.

INT-MX (eMbed instruct(X)ions): INT source devices embed INT instructions in the packet header. Every source/transit device then directly sends the metadata to the monitoring system by following the instructions embedded in the packets. INT sink devices strip the instruction header before forwarding the packet to the receiver. The packet modification in this mode is limited to the instruction header and the packet size doesn't increase as the packet interacts with more INT transit devices on its path through the network.

INT-MD (eMbed Data): This mode features the most packet modification and works by having both INT instructions and metadata written into the packets. Packets start at the INT source, which embeds instructions and metadata. Then they travel hop-by-hop through the network with each INT transit node embedding metadata. Finally, the INT sink strips the instructions and aggregated metadata out of the packet and sends selective data to the monitoring system. This mode minimizes the overhead at the monitoring system to collate reports from multiple devices.

INT source devices may generate INT-marked synthetic traffic either by cloning original data packets or by generating special probe packets. INT is applied to this traffic by transit nodes in the same way as other traffic.

When the system is configured for synthetic traffic, the INT sink nodes are programmed to discard the traffic after extracting the collected INT data because it is cloned and not to be sent to any destination. The INT source node can program any of the INT modes to be used on synthetic / probe packets.

INT Monitoring Tools

INT monitoring tools collect data from the INT sink nodes and aggregate that data to provide reports to identify anomalies, bottlenecks, packet drops, latency spikes, and application performance issues. There are also a number of open-source INT monitoring and management tools, as well as commercial software products such as Intel® Deep Insight™ Network Analytics Software.

Intel Deep Insight Network Analytics Software synthesizes and analyzes telemetry data from P4 data planes (such as Intel® Tofino switch ASICs), in addition to edge-to-edge or host-enabled INT implementations. It's reporting capabilities include network topology view, flow table view, event and anomaly dashboards, and top-down troubleshooting workflows to enable rapid diagnosis and resolution of issues. This robust analytics software can also process reports in real time, implementing intelligent analytics, and data

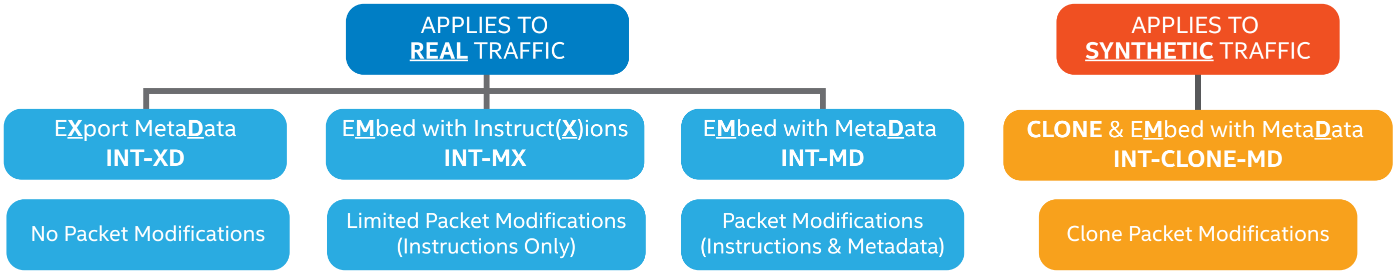
APPLIES TO REAL TRAFFIC
APPLIES TO SYNTHETIC TRAFFIC
EXport MetaData
INT-XD
EMbed with Instruct(X)ions
INT-MX
EMbed with MetaData
INT-MD
CLONE & EMbed with MetaData
INT-CLONE-MD
No Packet Modifications
Limited Packet Modifications (Instructions Only)
Packet Modifications (Instructions & Metadata)
Clone Packet Modifications

Figure 4. INT modes of operation.

correlation techniques to detect network anomalies. These anomalies can be detected and stored for deeper historical analysis. Further, all of this is done at line rate and with multi-terabit/s throughput, without the overhead of additional hardware.

Other features of Deep Insight include:

- Real-time network anomaly detection
- Path and latency tracking
- Topology-aware analytics
- Real-time network event detection
- Congestion analysis
- Packet-drop analysis
- Advanced filtering
- APIs for integration with third-party monitoring solutions
- Easy scale-out and flexible deployment
- System health monitoring

INT Use Cases

Some of the common INT monitoring use cases include network troubleshooting, performance monitoring, congestion control, and routing control. Below are in-depth examples of these applications:

High Precision Congestion Control (HPCC)

Bandwidth intensive networks can create significant congestion as different flows compete for network capacity. This congestion impacts latency. High precision congestion control (HPCC) uses INT to obtain precise link load information and controls traffic precisely. INT provides the granularity and response needed to manage congestion at data center speeds (40Gbps and faster). It also enables HPCC to quickly converge on congestion issues allowing data flows to be rerouted to utilize free bandwidth and reducing in-network queues to near zero.

Micro-burst detection

Micro-bursts are short lived congestion events that impact latency of a queue and can cause problems in low-latency environments. The challenge is having visibility into these bursts – and being able to control them or route packets around them – which takes visibility at timescales on the order of a few round-trip times (RTTs). Additional complications include determining which queues to monitor as the underlying routing could change, and the switch hash functions that affect multipath routing are often proprietary and unknown.

To detect these bursts, INT-enabled packets traverse the network collecting queue-size metadata at each hop. The queue sizes are absolute values at each hop and not averages, which makes them more useful in diagnosing micro-bursts. These packets enable the INT Sink to measure queue occupancy evolution at packet-level granularity allowing them to detect micro-bursts.

End-to-end latency detection

To measure and report latency, an INT source embeds instructions for packets to collect the time stamp of the local egress and local ingress to report the difference as the latency for that network element. The receiving INT transit device can compute the end-to-end latency as a sum of the per-hop latencies. Per-hop latencies in the packet received at the INT transit can also be used to determine which network element(s) contributed most to the end-to-end latency.

The network insight provided by INT is improved as more data is collected. For example, if the combination of switch-ID, input-port, output-port, and egress-link-utilization were all collected, the reports could be used to discover the different paths between a pair of end points and the congestion level along each path.

Host-based INT

INT technology can also be adapted for networks without P4-programmable switches by using a programmable network target at the source and sink. These targets can be small servers, such as universal customer premise equipment (uCPE) devices. Host-based INT doesn't require intermediate network switches to support the INT functionality, which inevitably limits data collection to the network ingress and egress; however, it does provide an outside-in look at network performance, which is valuable for large network owners that have real-time performance requirements and want to track service-level quality. It also provides total network latency measurements (per class of service with change detection) and identification of flow-based packet drops. With this data, app developers and/or service providers can calculate system anomaly detection and alerts, as well as isolate the network's impact on application performance.

In Host-based INT, when a packet comes into an INT source, a header is inserted that includes the node ID and the ingress timestamp. The packet is marked as carrying telemetry data, so at the egress point, an INT sink device can detect the packet and extract the INT header data. The packet is then forwarded, while metadata goes to the control plane for report generation. Subsequently, the INT sink captures the egress timestamp so latency can be calculated. The system also supports change detection.

The implementation of Host-based INT requires a protocol that can insert and extract metadata information from the packet. One approach is to use Express Data Path (XDP) to add metadata at the source and collect it at the sink. The technical advantage of XDP is that it works at the kernel level, and at the sink, it can pass report information to a collection server without leaving the kernel space to go through the network stack. XDP is very high performance and can implement change detector functionality for scaling. An alternative approach now in development is using Linux flowtrace, a user-space active measurement framework built into Linux that enables in-band network measurements using application TCP flows.

Case Study

One organization that has adopted Host-based INT uses it in a nationwide network connecting 3,000 facilities that are providing real-time information for analysis to two data centers. The organization has provisioned Host-based INT on uCPE servers that are at the edge of their network.

This organization uses the Intel Deep Insight Network Analytics Software to compile the data and to analyze and report the performance results monitored by INT. Using this monitoring tool allows the organization to analyze the data so it can measure its service levels to ensure its CoSP is providing promised network quality.

Conclusion

INT is a powerful tool to provide a very high degree of visibility into networks to quickly identify and address problems that previously were hard to detect. In addition, INT enables a closed loop solution that can make real-time changes to improve network throughput and reliability. Using the P4 programming language and programmable network elements, INT extracts telemetry data from the data plane with no impact on packet processing throughput. Through its three modes, INT is able to report its data to OAM systems, take action to alleviate adverse network conditions, as well as detect and report network events.

Learn More

Intel Tofino Switch

INT 2.1 Specification

P4.org

Open Networking Foundation

High Precision Congestion Control

Glossary

- **Monitoring System:** A system that collects telemetry data sent from different network devices. The monitoring system components may be physically distributed but logically centralized.
- **INT Header:** A packet header that carries INT information. There are three types of INT headers – eMbed data (MD-type), eMbed instruction (MX-type) and INT telemetry report header.
- **INT Packet:** A packet containing an INT header.
- **INT Instruction:** Instructions indicating which INT metadata (defined below) to collect at each INT switch. The instructions are either configured at each INT-capable device's Flow Watchlist or written into the INT header.
- **Flow Watchlist:** A data plane table that matches on packet headers and inserts or applies INT instructions on each matched flow. A flow is a set of packets having the same values on the selected header fields.
- **INT Source:** A trusted entity that creates and inserts INT headers into the packets it sends. A Flow Watchlist is configured to select the flows in which INT headers are to be inserted.
- **INT Sink:** A trusted entity that extracts the INT headers and collects the path state contained in the INT headers. The INT Sink is responsible for removing INT headers, so INT is transparent to upper layers. (Note that this does not preclude having nested or hierarchical INT domains.) The INT Sink can decide to send the collected information to the monitoring system.
- **INT Transit Hop:** A networking device that collects metadata from the data plane by following the INT instructions. Based on the instruction, the data may be directly exported to the telemetry monitoring system or embedded into the INT header of the packet.

  Note that one physical device may play multiple roles – INT Source, Transit, Sink – at the same time for the same or different flows. For example, an INT Source device may embed its own metadata into the packet, playing the roles of INT Transit as well.
- **INT Metadata:** Information that an INT Source or an INT Transit Hop device inserts into the INT header.
- **INT Domain:** A set of inter-connected INT devices under the same administration. This specification defines the behavior and packet header formats for interoperability between INT switches from different vendors in an INT domain. The INT devices within the same domain must be configured in a consistent way to ensure interoperability between the devices. Operators of an INT domain should deploy INT Sink capability at domain edges to prevent INT information from leaking out of the domain.

Notices & Disclaimers

Intel technologies may require enabled hardware, software or service activation.

© 2020 Intel Corporation Printed in USA 1220/RE/WPINT/PDF Please Recycle 345568-001 US