100G In-band Network Telemetry With Netcope P4

This document explains how the power of the P4 language and Field Programmable Gate Arrays (FPGAs) can be combined to achieve unprecedented time to market in the high bandwidth networking domain. Netcope P4 to FPGA Compiler (Netcope P4) is used to generate a 100 Gbps network probe for datacenter traffic carrying In-band Network Telemetry (INT) headers. While the original traffic (with INT headers stripped away) is forwarded to its destination at wire speed, INT headers are processed, stored and visualized by the Flowmon Collector, a leading commercial platform for network traffic analysis and visibility. The final solution is a clever mix of hardware-accelerated probe and mature network monitoring software, all put together with no need for HDL coding.

INT: Real-time reporting of network status

The best description of INT can be unsurprisingly found in the INT specification[1]:

> Inband Network Telemetry 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. In the INT architectural model, packets contain header fields that are interpreted as telemetry instructions by network devices. These instructions tell an INT-capable device what state to collect and write into the packet as it transits the network. INT traffic sources (applications, endhost networking stacks, hypervisors, NICs, sendside ToRs, etc.) can embed the instructions either in normal data packets or in special probe packets. Similarly, INT traffic sinks retrieve (and optionally report) the collected results of these instructions, allowing the traffic sinks to monitor the exact data plane state that the packets observed while being forwarded.
>
> This powerful concept proves useful in a range of applications:
>
> - Network troubleshooting and performance monitoring
> - Advanced congestion control
> - Advanced routing
> - Network data plane verification

P4: Enabling Truly Programmable Network Dataplane

The P4 language overcomes the limitation of traditional fixed networking devices by providing a way to describe a custom packet processing chain that involves parsing, matching and assembling modified packets. This abstraction allows for a high degree of decoupling the data plane and control plane, enabling new applications and more options in virtualization of networking resources. The language is target independent and can be mapped to CPUs, FPGAs, NPUs, and ASICs. FPGAs are especially well aligned with the ideas of P4 because of their structural programmability, deterministic performance, and massively parallel nature.

The purpose of this whitepaper is to demonstrate how P4 can be used to quickly obtain running FPGA firmware for a particular application. We use INT traffic analysis as the example application. INT has few real implementations today, so it is a perfect example of how **agile** can **FPGA design** with P4 actually be. By

[1] http://p4.org/wp-content/uploads/fixed/INT/INT-current-spec.pdf

integration with Flowmon, we further show that it is possible to build a novel, production-quality, hardware-accelerated and vertically integrated application very efficiently.

Building a 100G INT sink with Netcope P4 and FPGA

We use Netcope FPGA board NFB-100G2 as a hardware platform, together with Netcope Development Kit (NDK), which takes care of 100GE network interfaces, fast DMA transfers over PCI Express, and other hardware infrastructure tasks. To address the INT-specific functionality, we use the Netcope P4 to FPGA Compiler to generate the processing engine. There are two main functions of the resulting firmware: Header stripping and Header export.

Header stripping

The FPGA firmware strips the INT headers away from the packets and forwards the packets back to the network with very low latency. This means that the devices that follow in the network suffer no additional processing load associated with telemetry. By building a dedicated INT probe, we assemble a network monitoring service with minimal impact on the network that is being monitored.

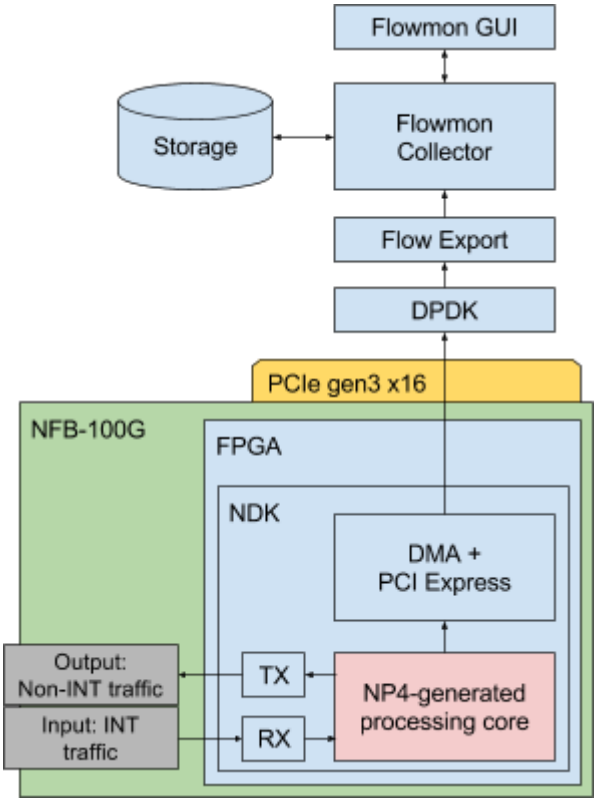


Fig. 1: Architecture of 100GE INT probe.

INT headers are declared in the following (simplified) P4 code snippet:

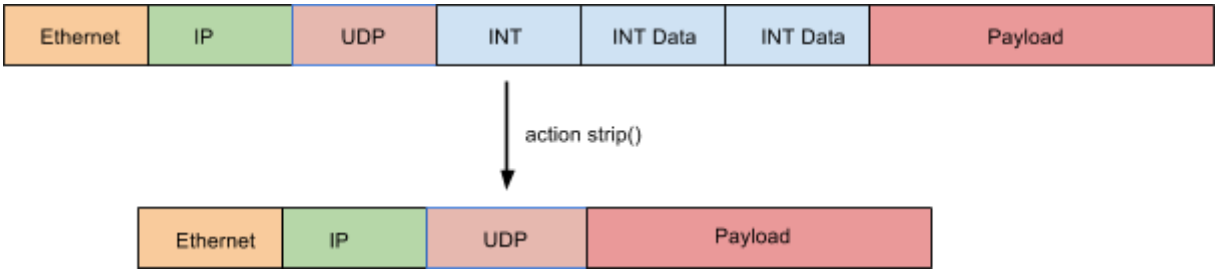
```
header_type int_hdr_t {
    fields {
        ver          : 2;
        rep          : 2;
        c            : 1;
        e            : 1;
        rsvd1        : 5;
        ins_cnt      : 5;
        max_hops     : 8;
        total_hops   : 8;
        inst_mask    : 16;
        rsvd2        : 16;
}   }
```

```
header_type int_payload_t {
    fields {
        id       : 32;
        time     : 32;
}   }
```

The code for stripping of INT headers (max 2 INT payloads allowed) from packets is as simple as:

```
action strip() {
    remove_header(intpayload1);
    remove_header(intpayload0);
    remove_header(inthdr);
}
table strip_tab {
    actions {strip;}
}
```

This results in the following packet modification scheme:



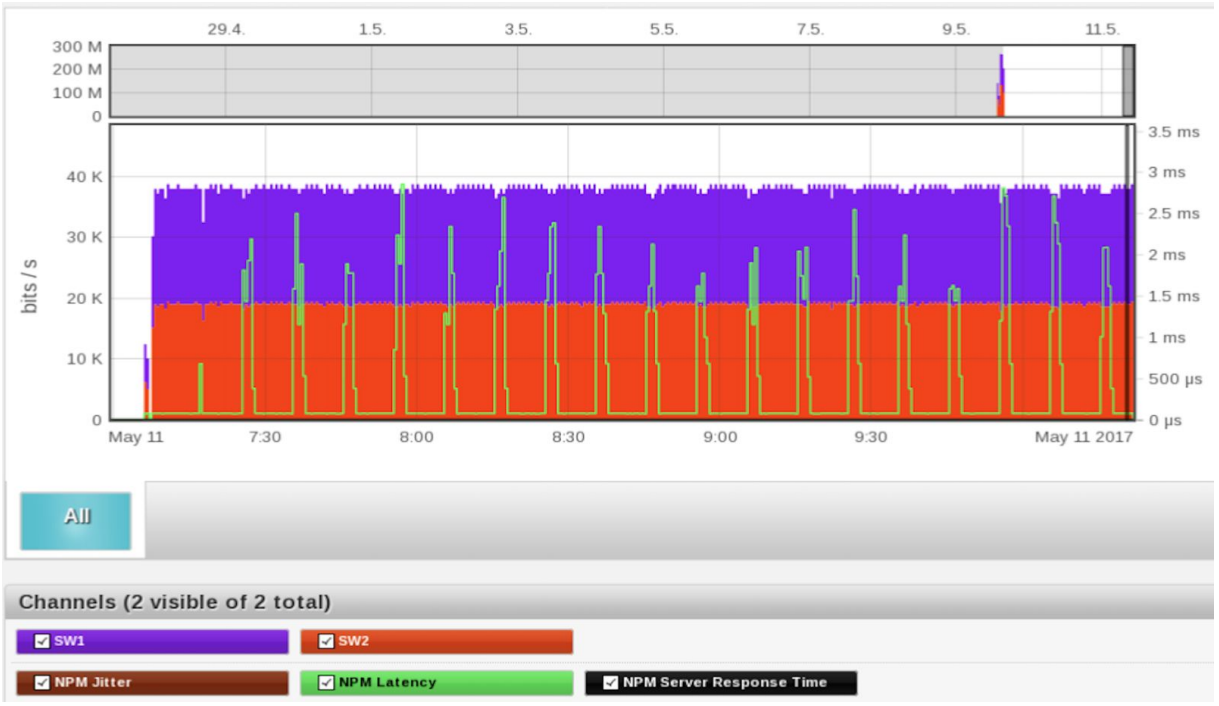
Header export

Any information that is relevant for network traffic monitoring is exported to the host CPU for further processing. This includes INT headers as well as some basic fields from other protocols, such as IP addresses. P4's `generate_digest` action is used to generate this additional output from the P4 pipeline. Optionally, sampling can be enabled to reduce the load in the subsequent steps. The generated headers are sent over PCI Express bus, utilizing the industry standard DPDK API on the software side. The Flow Export tool is where most of the

actual programming was done. It is a rather straightforward C tool that accepts headers from the DPDK API, parses them, and generates standard NetFlow protocol records carrying information about network flows and the telemetry they have gathered on their way. As a proof of concept, we have hacked the static format of NetFlow v5 records to carry telemetry information about packet delay within switches.

The extra mile: Integration with Flowmon

Flowmon Collector is a powerful appliance providing detailed network traffic visibility. The solution is dedicated for collection, visualization, analysis and long-term storage of network statistics (NetFlow v5/v9, IPFIX, and other flow data) generated from routers, switches, firewalls or specialized probes. We have extended the Collector to gather, visualize and allow analysis of packet delay within switches. This means that the already excellent support for graphs and query-based analysis of network problems at L3 and L4 is now enriched by the information from the underlying L2 infrastructure.



Example of Flowmon Collector Web GUI.

Conclusion

Our work demonstrates that it is perfectly possible to innovate by building novel, production-quality, hardware-accelerated and vertically integrated applications very efficiently. Inband Network Telemetry is a nice example of a new and very promising application with very few actual implementations available, not to mention the high-speed ones. By introducing an entirely new protocol and targeting the bandwidth-demanding domain of datacenter networking, INT offers several interesting engineering challenges, but also opportunities.

The Netcope P4 Compiler provides a straightforward way from the high-level problem description to the FPGA firmware running at 100 Gbps. By reducing the need for manual HDL coding, the development time and chance of coding error are minimized. Altogether, Netcope P4 reduces time to market and development cost of a wide range of applications.