A National Programmable Infrastructure to Experiment with Next-Generation Networks

Paola Grosso
University of Amsterdam

Cristian Hesselman
SIDN Labs and
University of Twente

Luuk Hendriks
University of Twente

Joseph Hill
University of Amsterdam

Stavros Konstantaras
AMS-IX

Ronald van der Pol
SURF

Victor Reijs
SIDN Labs

Joeri de Ruiter
SIDN Labs

Caspar Schutijser
SIDN Labs

2STiC

AMS-IX, NDIX, NLnet Labs, SIDN Labs, SURF, Technical University of Delft, the University of Amsterdam and the University of Twente work together in the joint research consortium called 2STiC (pronounced "to-stick"). 2STiC consortium's goal is to develop and evaluate mechanisms to increase security, stability and transparency of inter-network communications by experimenting with and contributing to emerging internet architectures.

For more information see https://2stic.nl

Testbed

- First nation-wide multi-domain P4-programmable network in The Netherlands
- Goal: develop, deploy, and experiment with new network-level systems and protocols that increase the trust and autonomy (sovereignty) of users
- Experiment with future Internet or networks based on SCION, RINA, or the "responsible Internet" paradigm
- Currently six sites with P4 programmable equipment
- Connected via SURF's network through dedicated 200 Gbit/s optical wavelengths

Use cases

SCION

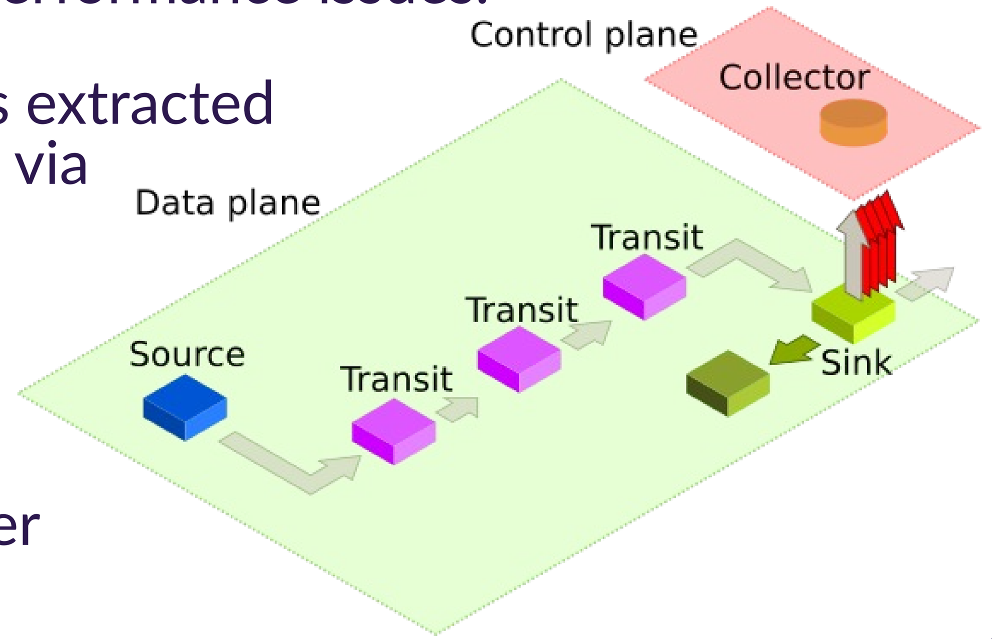
We are implementing the SCION internet architecture in P4 for the Intel Tofino ASIC. The implementation is work in progress, but already led to several improvements of the SCION protocol headers, making them easier and more efficient to implement in hardware.

In-band Monitoring of Flows

We evaluated two methodologies for gathering and storing in-band monitoring data across routers.

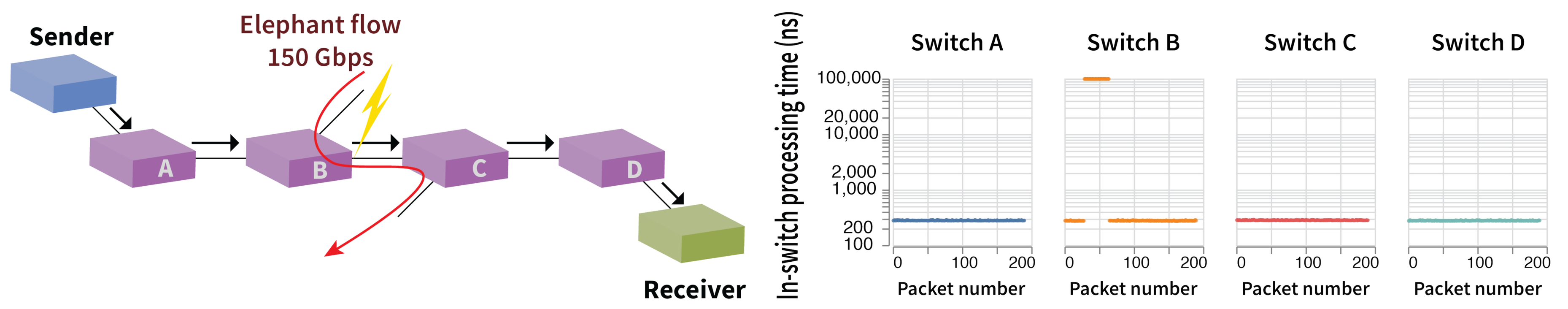
- **Control plane:** the in-band data is directly sent to the control plane, introducing slow-path performance issues.
- **Data plane:** the in-band data is extracted from the traffic and forwarded via the data plane to an external collector.

Using the testbed, we found the data plane-oriented data collection to be faster, and closer to the concept of INT.



In-switch Latency Measurements

We use P4 switches to add latency information to an IPv6 extension header. Our switches record timestamps when a packet enters a switch and when it leaves a switch.



Telemetry Collection with RDMA

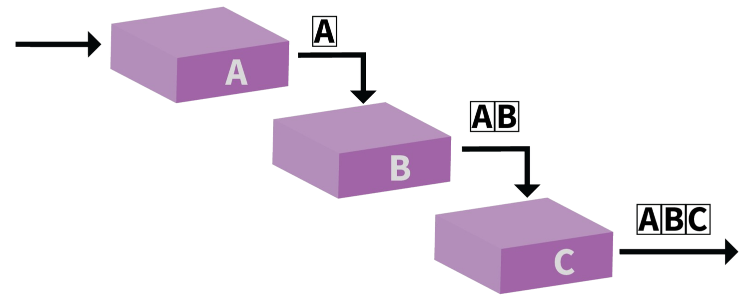
We employed P4 to craft a telemetry message for each packet that passes through the data plane of a device. By creating a telemetry message for each individual packet, the amount of state the switch needs to track is reduced.

We used RDMA (Remote Direct Memory Access) to reduce the workload of the system receiving the telemetry messages. With RDMA, a NIC can write data directly to memory without involving the CPU. The P4 switch extracts telemetry data from the network traffic and sends it encapsulated in an RDMA packet to the collector, which monitors the buffer and writes the data to disk.

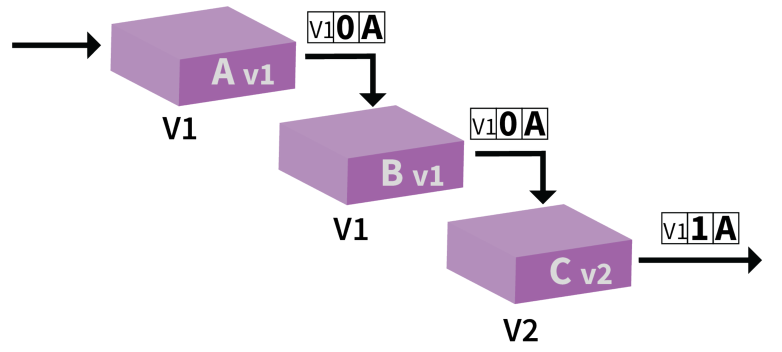
Path Tracking

We developed two P4-based methods to track an IPv6 packet's path:

- **Hop recording:** at each node we append the ID of this node to the IPv6 extension header. The complete path a packet took can be extracted from the packet in the last node of the path.



- **Forward state logging:** we assume that the routing information is stored and available. At each node we include in the IPv6 extension header the information about the routing table version that is used to make the forwarding decision. The complete path can be re-constructed based on the node where the packet entered a network.



2STiC