Data plane monitoring and measurements with P4

Paola Grosso
University of Amsterdam
P.Grosso@uva.nl

Luuk Hendriks
University of Twente
Luuk.Hendriks@utwente.nl

Joseph Hill
University of Amsterdam
J.D.Hill@uva.nl

Marijke Kaat
SURFnet
Marijke.Kaat@surfnet.nl

Silke Knossen
University of Amsterdam
S.Knossen@uva.nl

Ronald van der Pol
SURFnet
Ronald.vanderPol@surfnet.nl

Victor Reijs
SIDN labs
Victor.Reijs@sidn.nl

2STiC - Security, Stability and Transparency in inter-network Communication

AMS-IX, NLnet Labs, SIDN Labs, SURFnet, 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 internet communications by experimenting with and contributing to emerging internet architectures such as SCION, RINA, and NDN.

The 2STiC partners envisage that such new types of internet will complement and co-exist with the current Internet, in particular for critical applications such as intelligent transport systems, smart homes and smart grids.Our long-term objectives are to establish a centre of expertise in the field of trusted and resilient internets, and to help put the Dutch (and European) networking communities in a leading position in the field.
For more information see http://2stic.nl.

P4 as enabling technology

P4 is a domain-specific programming language that allows developers to program the data plane of a packet forwarding device *[1]*. There is an increasing interest in evaluating P4 devices by more and more research networks, driven by the plethora of applications and use cases they could support.

The 2STiC researchers are tackling the question: "Can P4 devices offer novel methods to measure and monitor traffic flowing through them? How?"

We implemented P4 programs to perform in-band monitoring of the active flows and to track the full path across the network in a transparent way.

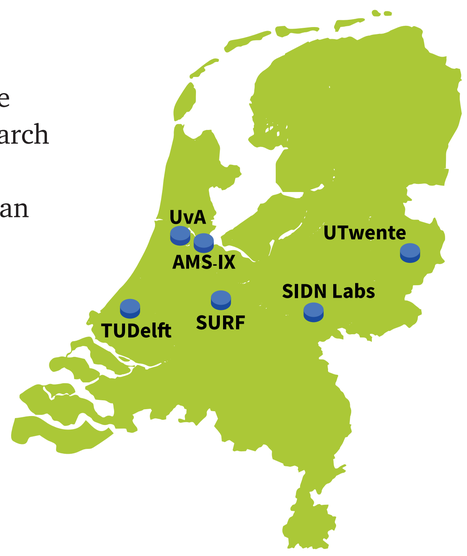
[1] The P4 Language Consortium. P4-16 language specification. 2018. doi: 10.1093/mnras/118.4.379. https://p4.org/p4-spec/docs/P4-16- v1.1.0-spec.pdf

2STiC Testbed

The 2STiC testbed currently consists of six nodes across the Netherlands. Each node contains a **P4 programmable switch** and a server with a programmable smart NIC. The switches are interconnected with multiple direct links on which Ethernet framing is used.

With P4 we can implement any protocol over these Ethernet links. This makes it very suitable for research on new protocols, such as SCION and RINA.
By extending existing protocols with sensors, we can also do all kinds of measurements on them.

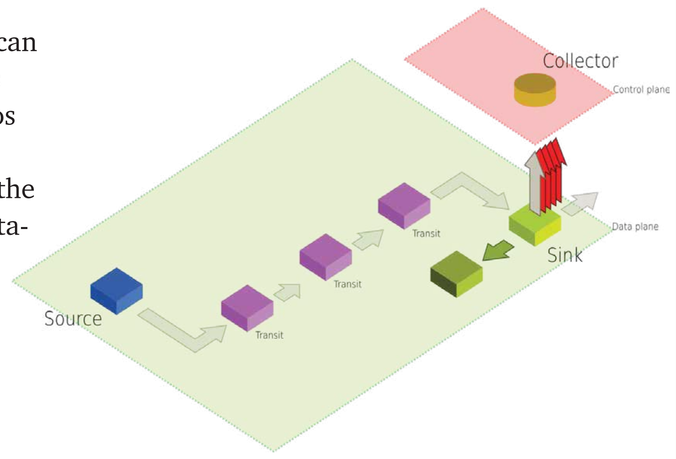
The usage of P4 hardware switches and smart NICs is an important addition to research done with network simulators and software switches.



Collecting Monitoring Data

Extracting information added by one or more hops in-band can be done in two ways: via the control plane, which does not require additional hardware but does introduce typical slow-path performance issues.
Or, the INT *[2]* information can be extracted from the traffic and forwarded via the data plane to an external machine running dedicated software to process the incoming INT payloads.

With the testbed set up, we can evaluate how these methods perform in different scenarios by varying the nature and volume of traffic, as well as the number and types of INT datapoints inserted in-band.

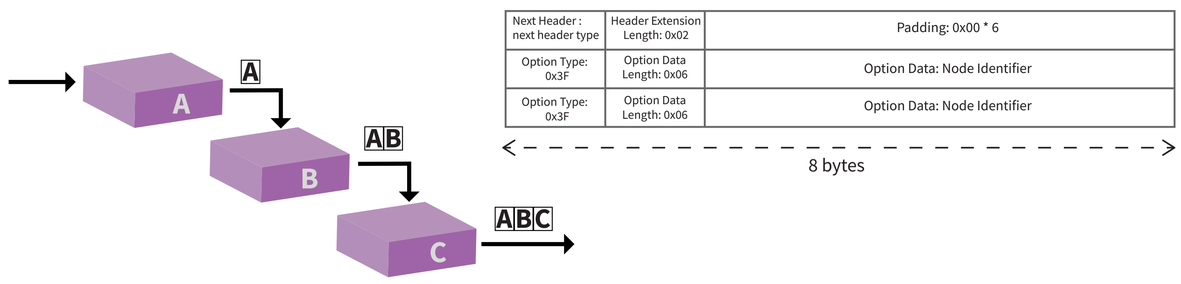


[2] Changhoon Kim et al. In-band Network Telemetry (INT). https://p4.org/assets/INT-current-spec.pdf

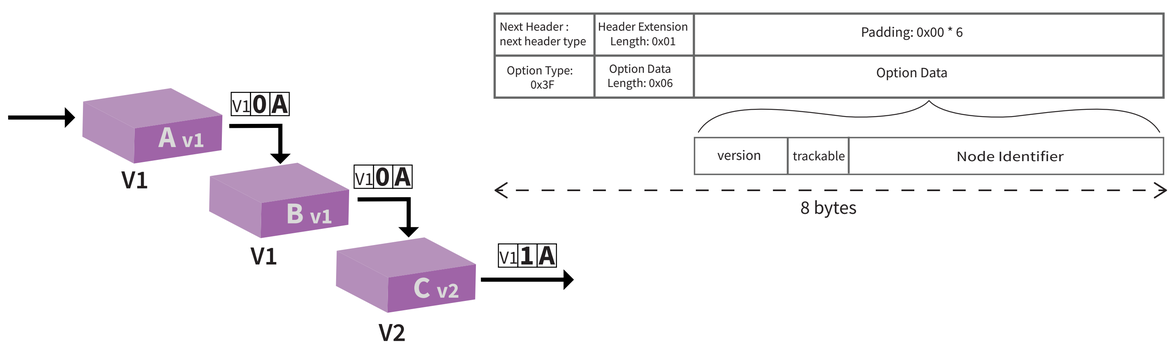
Path Tracking

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 reconstructed based on the node where the packet entered a network.
In this figure the version field identifies the routing table version at the entry point in the network. The **trackable** field is used to flag version changes along the path.



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.