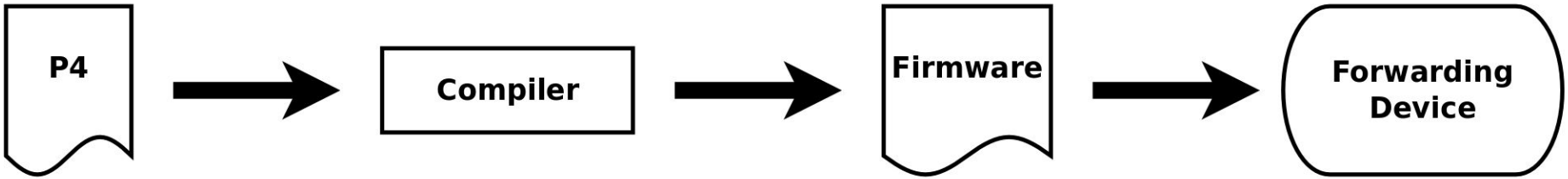
University of Amsterdam
System and Network Engineering
SURF NET

Hardware Implementations of Path Tracking in P4$_{16}$

Silke Knossen
Joseph Hill
Paola Grosso

What is P4?

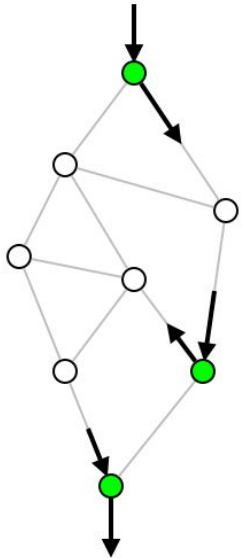
- P4: Programming Protocol-Independent Packet Processors
- Programming language that describes the behavior of the data plane
- Allows the rapid development and deployment of protocols
- An alternative to fixed function devices (e.g. Switches, Routers, Firewalls…)
- Software P4 switch provided by the P4 Language Consortium



Motivation

Can the features provided by P4 be used to enhance the ability to identify and track malicious traffic in networks?

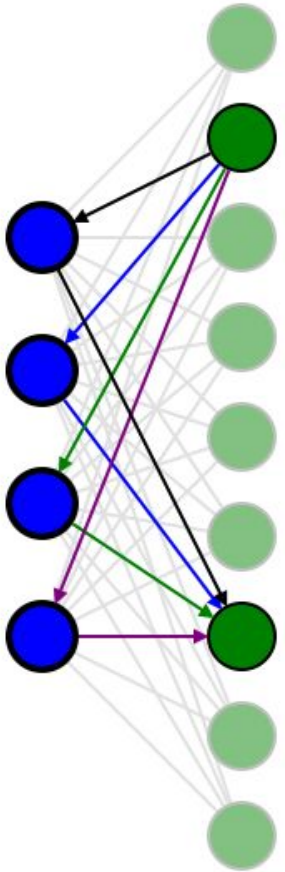
- Goals
  - Identify as much of the path of a flow as possible
  - Minimize impact on system resources
- Assumptions
  - Traffic may be classified as malicious at later time
  - NetFlow is unable to capture all flows at every node



Sampled NetFlow may result in incomplete knowledge of the path

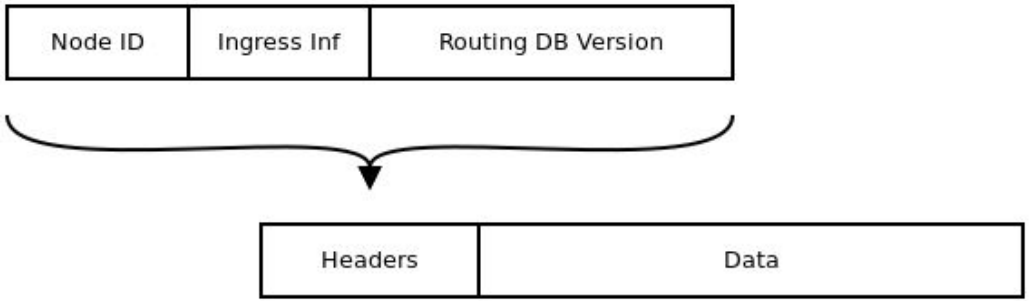
Path Tracking Methods: Hop Recording

- Each node adds its node ID to a header stack
- Flow and Path saved at egress node
- Not a novel approach
  - RFC 791 - Internet Protocol, 1981
  - In-band Network Telemetry (INT), 2015
- Variable amount of data added to packet
- Useful when load balancing over several short paths



Path Tracking Methods: Forwarding State Logging

- Network forwarding state identifier added to packet
- Use a record of forwarding state to recreate path
- Requires global view of how routing will be performed
- Requires archives of each version of routing database



Future work from 2018

- What effect will using $P4_{16}$ have on the implementation?
- How will additional constraints imposed by hardware affect implementation?

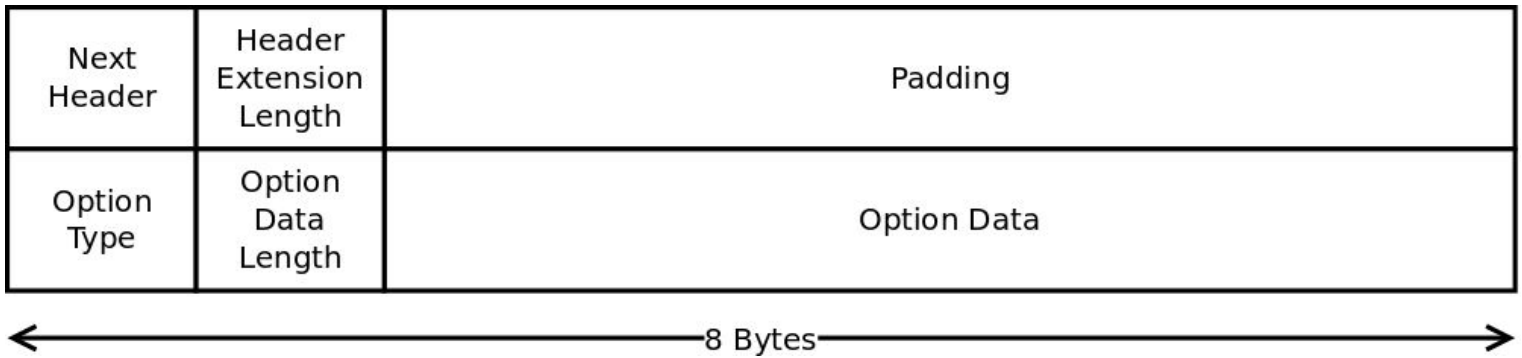
$P4_{16}$ Architectures

- $P4_{16}$ introduces the idea of an architecture model which defines the programmable blocks of a target and their interfaces
- Must be implemented by the target
- The v1model architecture is designed to be identical to the $P4_{14}$ switch architecture

"As an analogy, the PSA is to the $P4_{16}$ language as the C standard library is to the C programming language."

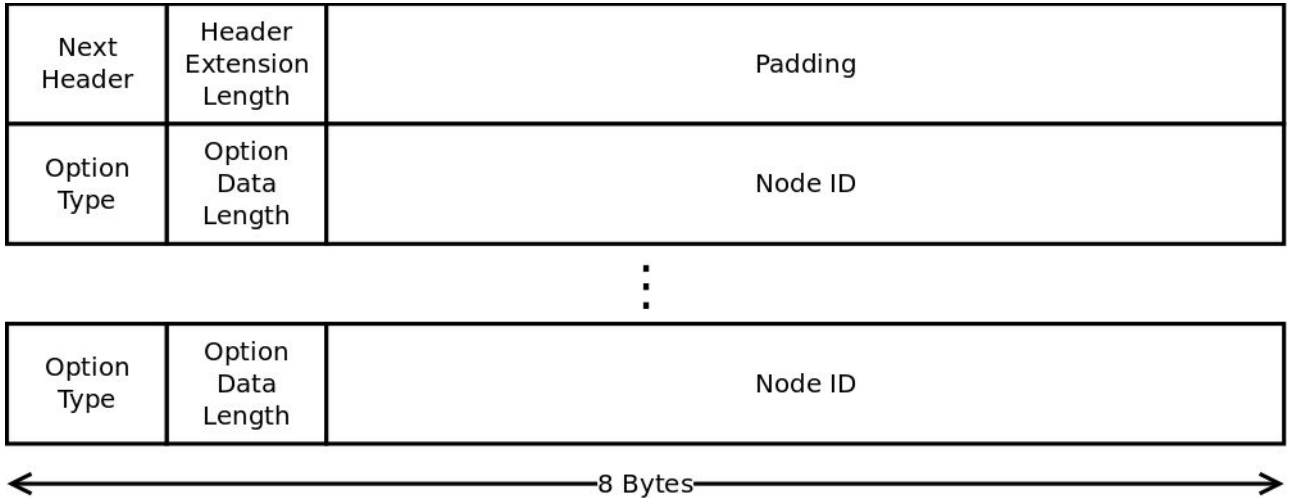
IPv6 Extension Header

- IPv6 Hop-by-Hop option extension header used to encode data
- Option data can be altered in transit
- Header is processed at each node along the path
- Can be ignored by devices that do not support it



Hop Recording Data Encoding

- Node ID encoded in the Option Data
- Each node adds an option
- Arbitrary growth of packets supported up to maximum Hop Limit



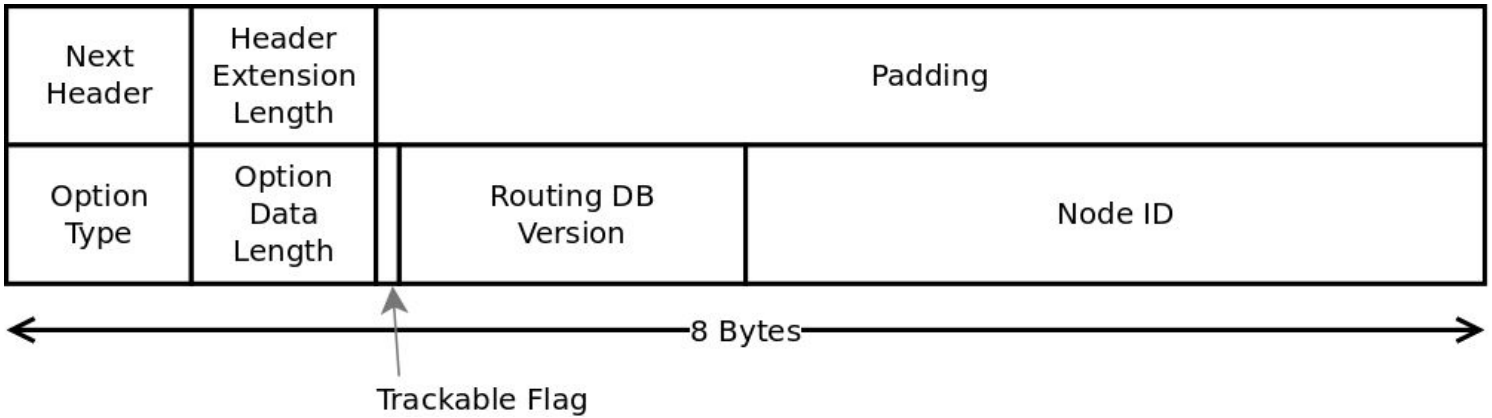
Header Processing Issues

- Initially planned to process each option as a stack of headers
- Alternatively inserting a new option in front of the other options
- Variable length headers not well supported

“P4 is a domain-specific language that is designed to be implementable on a large variety of targets including programmable network interface cards, FPGAs, software switches, and hardware ASICs. As such, the language is restricted to constructs that can be efficiently implemented on all of these platforms.”

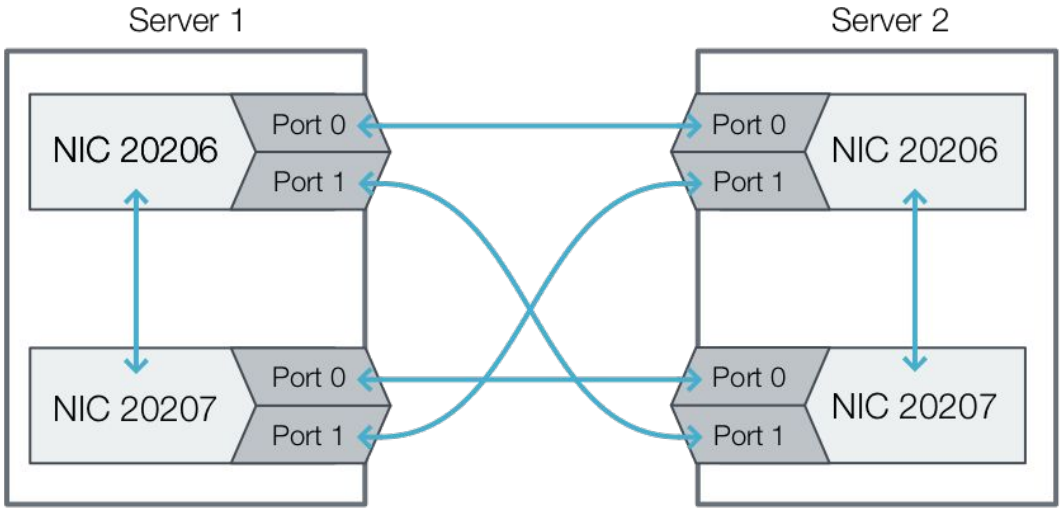
Logging Forwarding State Data Encoding

- Fixed amount of data added to each packet
- Multiple tracking fields encoded into the option data



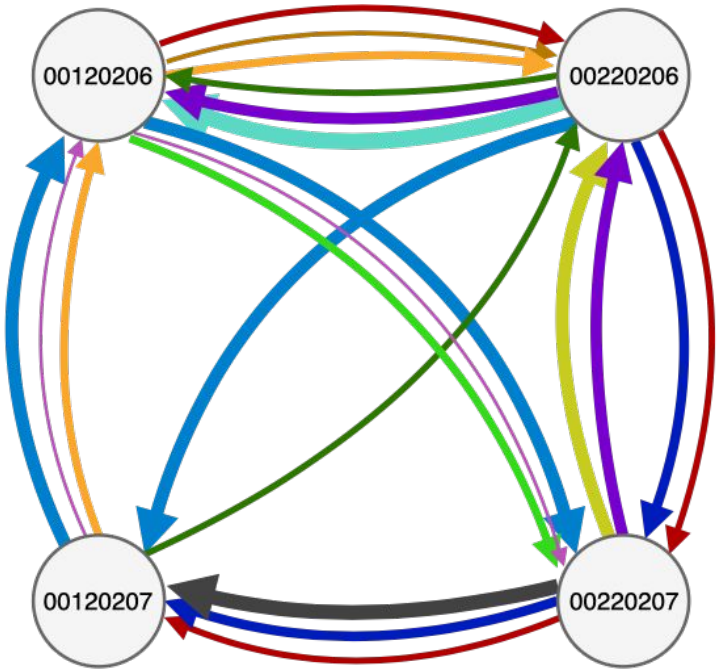
Experimental Setup

- Four Netronome Agilio CX 2x25 GbE SmartNICs
- Each NIC acts as a node in the experimental network



Conclusions and Future Work

- Visualizing Data
- Efficient data extraction
- RFC 8200 compliance
- IPv4 solution
- Layer 2 solution
- Evolution of P4
  - P4 Runtime
  - Portable Switch Architecture
- Data Transfer Nodes (DTNs) and P4

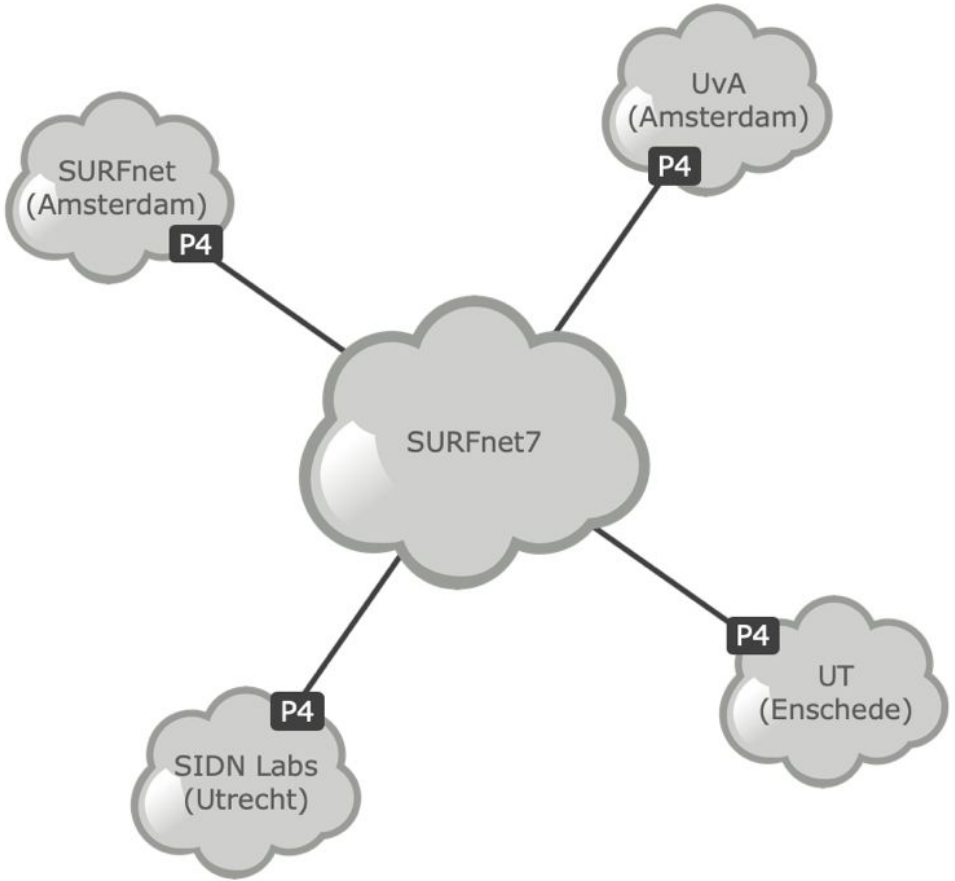


Measuring end-to-end latency with P4 and INT

- Precision timing measurements in the Data Plane (nanoseconds)
- With and without synchronized time between devices
- Implemented on switches with Barefoot Tofino using $P4_{16}$

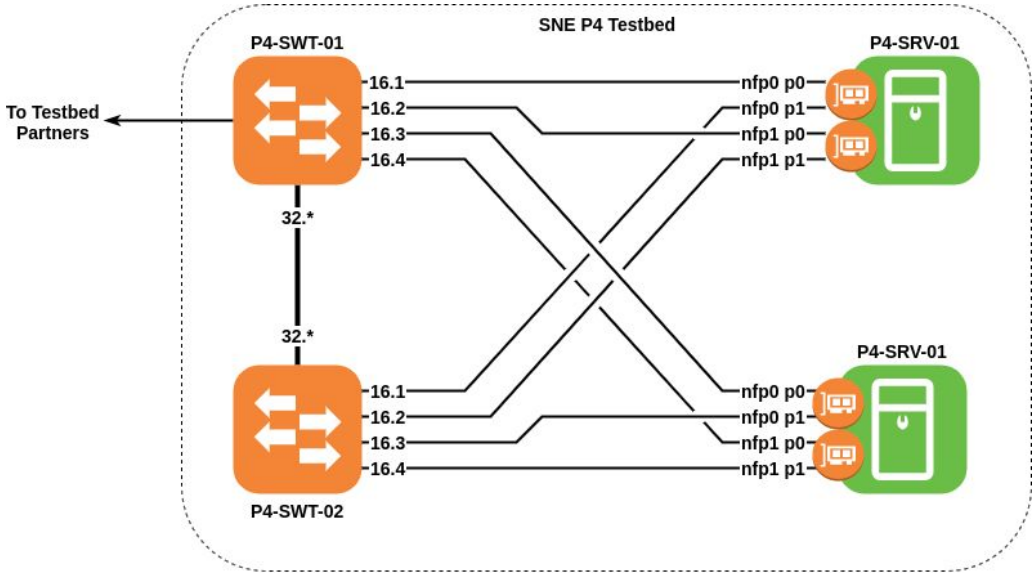
2STiC Program

- AMS-IX
- NLnet Labs
- SIDN Labs
- SURFnet
- TU Delft
- University of Amsterdam
- University of Twente



UvA P4 Testbed

- Edgecore Wedge 100BF-32X
  - 32 x QSFP28 ports
  - $P4_{14}$ and $P4_{16}$ programmable
  - Hardware packet generator
- Netronome SmartNIC
  - 2 x SFP28
  - $P4_{14}$, $P4_{16}$ and C programmable



Questions?