UNIVERSITY OF TWENTE.

RIPE71 - Bucharest, Romania

Scalable high-speed packet capture

Using OpenFlow and Intel DPDK

Wouter de Vries

Who am I?

Wouter de Vries
Ph.D. student

Design and Analysis of
Communication Systems
University of Twente

DACS Design and Analysis of Communication Systems

SURF NET

Introduction

We want to capture large-scale DDoS attacks without significant packet loss, why?

- Mitigation is hard
- In-depth analysis could provide valuable insights

Other uses of high-speed packet capture:

- Intrusion detection
- Monitoring (start your own NSA!)

The Problem

The total bandwidth of The Internet^{TM} is ever increasing.

The Problem

The total bandwidth of The Internet^{TM} is ever increasing.

Table: Cisco Visual Networking Index 2015

Year	2014	**2015**	2016	2017	2018	2019
PB per Month	59,8	**72,4**	88,4	109,0	135,5	168,0

The Problem

In order to analyze real-world traffic, the capture methods need to evolve.
At speeds in excess of 10 Gbit/s things start to get difficult:

The Problem

In order to analyze real-world traffic, the capture methods need to evolve.

At speeds in excess of 10 Gbit/s things start to get difficult:

- $\geq$ 14.8 million packets per second

The Problem

In order to analyze real-world traffic, the capture methods need to evolve.
At speeds in excess of 10 Gbit/s things start to get difficult:

- $\geq$ 14.8 million packets per second
- Only a few clockcycles per packet
- Storing $\geq$1.25 Gigabytes per second

The Problem

In order to analyze real-world traffic, the capture methods need to evolve.
At speeds in excess of 10 Gbit/s things start to get difficult:

- $\geq$ 14.8 million packets per second
- Only a few clockcycles per packet
- Storing $\geq$1.25 Gigabytes per second

Goal

A **scalable** system that is able to capture and generate packets at high speed (e.g. $\geq$**40 Gbit/s**)

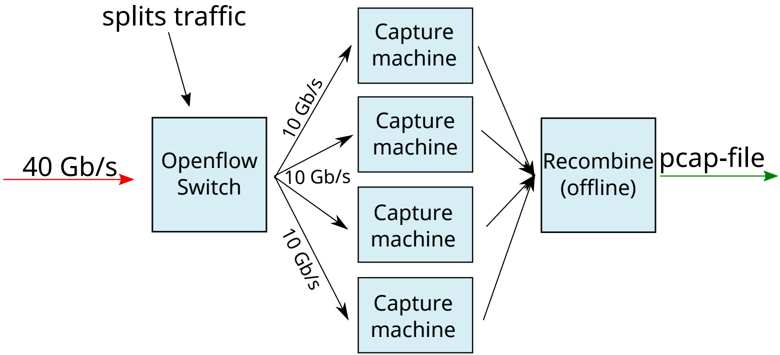
Proposal

- Use DPDK (Data Plane Development Kit) to maximize single machine performance.

Proposal

- Use DPDK (Data Plane Development Kit) to maximize single machine performance.
- Use OpenFlow-switches to distribute traffic over multiple machines

Proposal - Overview



Implementation - What is DPDK?

The **D**ata **P**lane **D**evelopment **K**it is a library for fast packet processing

Main features:

- Zero-Copy
- Fast buffers
- Designed for multicore

Zero-copy allows the network hardware to directly copy data to memory buffers using DMA

Implementation - What is DPDK?

The **D**ata **P**lane **D**evelopment **K**it is a library for fast packet processing

Main features:

- Zero-Copy
- Fast buffers
- Designed for multicore

Fast and thread-safe implementations of (ring) buffers making development of multithreaded applications much easier

DPDK
DATA PLANE DEVELOPMENT KIT

Implementation - What is DPDK?

The **D**ata **P**lane **D**evelopment **K**it is a library for fast packet processing

Main features:

- Zero-Copy
- Fast buffers
- Designed for multicore

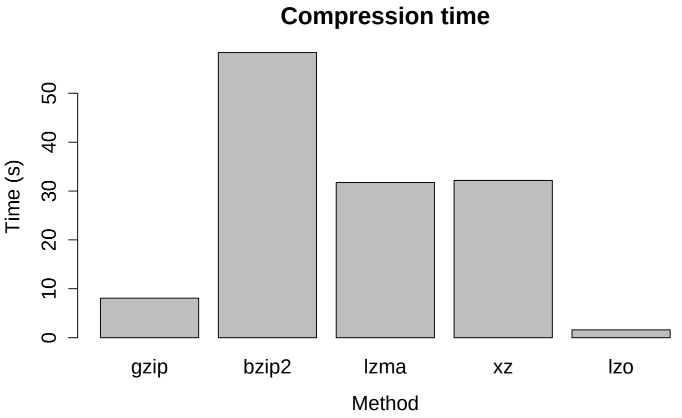
Has been designed from the ground up to support multiple cores, each thread runs on its own core

Implementation - Using DPDK

Capture 64-byte packets at 10 Gbit/s (1.25 GB/s or 1 DVD every 4 seconds) in PCAP-format on commodity hardware.

What to do with all this data?

Implementation - Adding compression



Compressing the linux kernel to a ram disk.

Source: http://catchchallenger.first-world.info

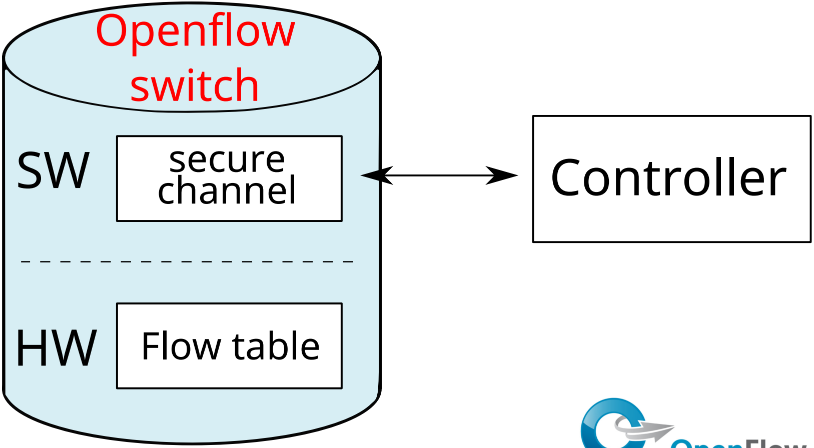
Intermediate results

- Using compression specially crafted 64-byte packets can be captured at line-rate on a single conventional HDD using 3 cores
- Generating packets at line-rate (10 Gbit/s) is possible using a single core

Implementation - What is OpenFlow?

> *”OpenFlow allows direct access to and manipulation of the forwarding plane of network devices such as switches and routers”*
>
> — Open Networking Foundation

Implementation - What is OpenFlow?

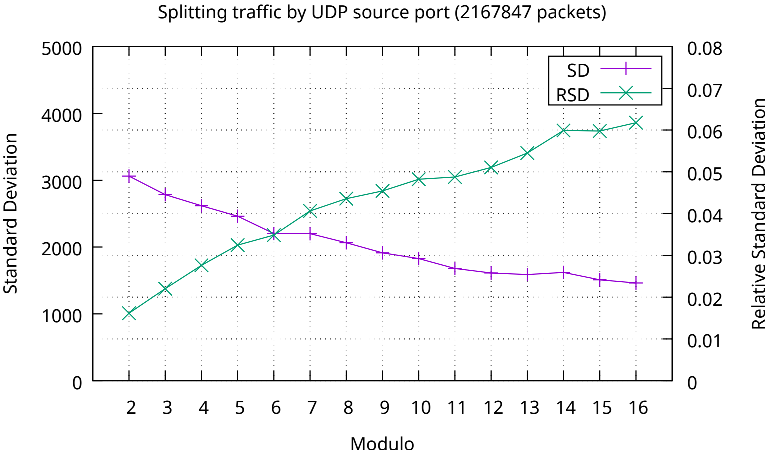


Implementation - OpenFlow

We need to define something that we split the traffic on.
Possible candidates:

- Source port for TCP/UDP (allows mask on Open vSwitch)
- IP-address (allows mask)
- Equal-Cost Multi-Path (ECMP) routing algorithms

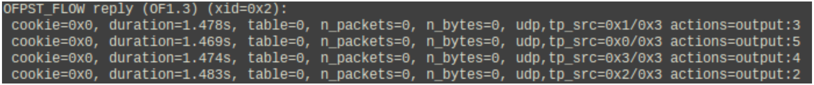
Implementation - UDP



Data: Random DRDoS attack PCAP from simpleweb.org

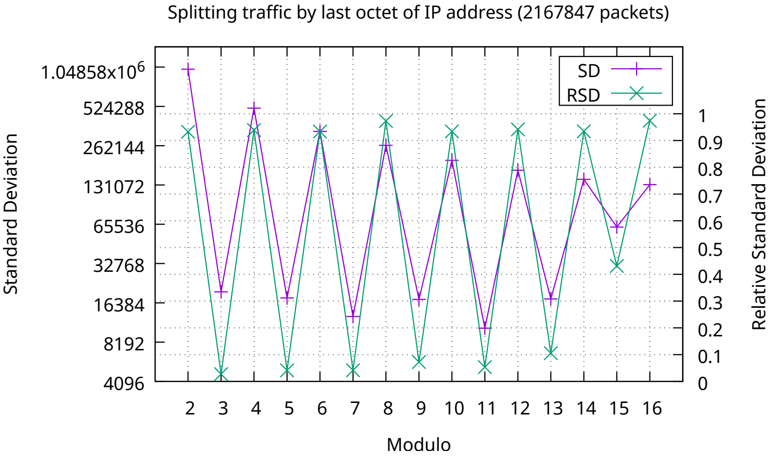
Implementation - Example flow table

Bitmask on last two bits of UDP source port



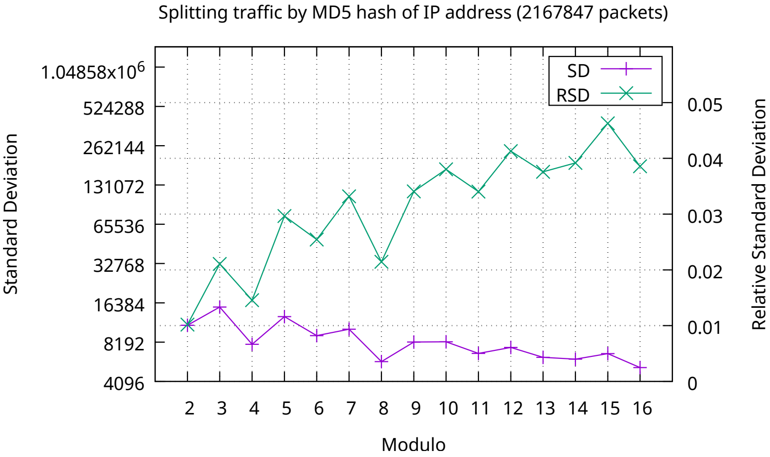
```
OFPST_FLOW reply (OF1.3) (xid=0x2):
 cookie=0x0, duration=1.478s, table=0, n_packets=0, n_bytes=0, udp,tp_src=0x1/0x3 actions=output:3
 cookie=0x0, duration=1.469s, table=0, n_packets=0, n_bytes=0, udp,tp_src=0x0/0x3 actions=output:5
 cookie=0x0, duration=1.474s, table=0, n_packets=0, n_bytes=0, udp,tp_src=0x3/0x3 actions=output:4
 cookie=0x0, duration=1.483s, table=0, n_packets=0, n_bytes=0, udp,tp_src=0x2/0x3 actions=output:2
```

Implementation - IP address



Data: Random DRDoS attack PCAP from simpleweb.org

Implementation - IP address



Data: Random DRDoS attack PCAP from simpleweb.org

Implementation - ECMP

Equal-Cost Multi-Path routing is used to balance traffic over multiple links that have the same cost.

- ECMP Algorithm is not defined by OpenFlow
- Result: ECMP implementation varies by vendor

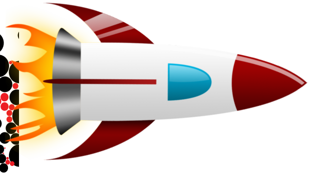
The **definition** of ECMP is a great match to our problem

Current state

- For some types of traffic splitting is easier than others
- On-going work to find a generic way to balance flows
- ECMP is promising, depending on the implementation by the vendor

Conclusion

- Using DPDK allows line-rate packet capture on 10 Gbit/s
- Using OpenFlow-compatible switches has the potential to scale the capture speed horizontally
- Combined, these two technologies allow us to capture $\geq$40 Gbit/s



Open-source

- DPDK-based packet capture tool (DPDKcap): https://github.com/woutifier/dpdkcap

Questions

Thank you for your attention!

Questions and/or comments are welcome!