Scalable and Cost-Effective Generation of Unsampled NetFlow

SWITCH

Alexander Gall
alexander.gall@switch.ch

Telemetry and Big Data Workshop

10.10.2020

NetFlow Export

- One of the oldest sources of network telemetry data
- Originally < source address, destination address, protocol, source port, destination port, interface >
- From Cisco-proprietary to IPFIX IETF standard
- Unsampled: process every packet
- Sampled: process “1 in n” packets only
- Today, most ISPs use sampling due to limitations on the exporting device

Why Unsampled?

Not necessary for volume-based metrics, but e.g.

- Fine-grained analysis of security incidents
- Reliable network problem debugging for low-volume flows, e.g.
  - TCP handshake
  - DNS transactions
- Also: because we can :)

NetFlow @SWITCH

- Used since early implementations on Cisco routers (ca. 1996)
- Unsampled up to Cisco 6500/7600
- Only sampled starting with ASR9k
- 2015: Move to external unsampled NetFlow generation on appliance (Flowmon) using hardware acceleration
- 2020: Replace with pure software implementation on commodity hardware

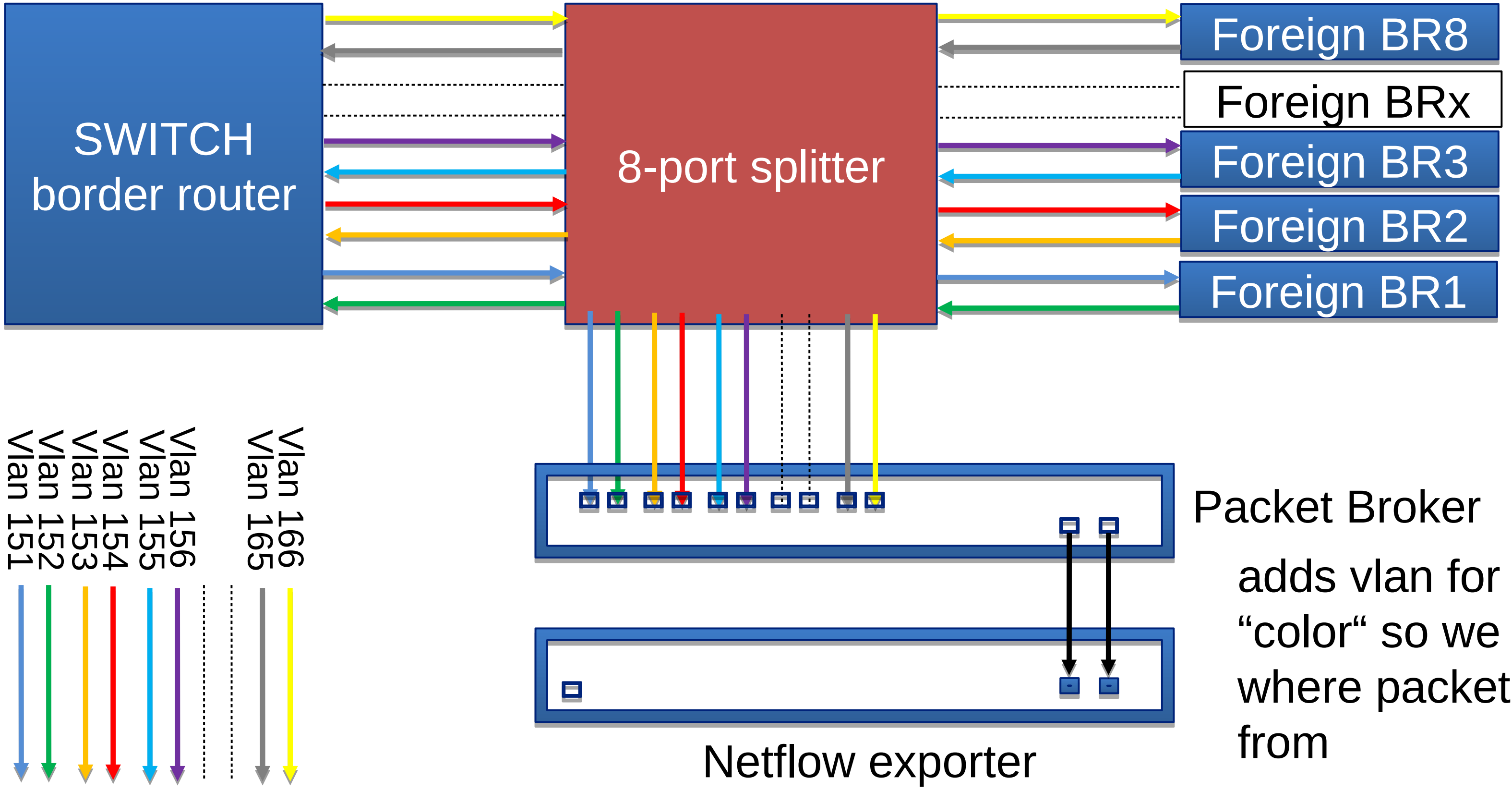
SWITCH External Traffic (Inbound+Outbound)

- Peak values
  - 100Gps
  - 15Mpps
  - 250kfps
- Average flow rate 150kfps, ~1TiB per day
  - Flow analysis is the real Big Data problem here
  - Current method based on `nfdump` is not adequate

Per-PoP Exporter Architecture

- Optical taps on external interfaces to copy packets
- “Packet Broker” to aggregate packets onto 2x100Gbps links to the exporter
  - Use VLAN tags to identify original router ports
- Exporter creates and exports flows

Per-PoP Exporter Architecture



adds vlan for each “color“ so we know where packets came from

Packet Broker

- P4-programmable, based on Tofino NPU from Intel (formerly Barefoot Networks)
- Device from Edgecore, 32xQSFP (WEDGE 100BF-32X), ~6k EUR
- In-house developed P4 program (requires NDA with Intel to obtain SDE) https://github.com/alexandergall/packet-broker
- Easy to add useful features
    - Mirror packets for local analysis

Exporter

- 1RU x86-based server, e.g. AMD Epyc 16-core
- Mellanox ConnectX5 dual-port 100Gbps NIC
- ~4k EUR
- In-house developed IPFIX-compliant exporter based on the Snabb framework (https://github.com/snabbco/snabb)
- Sourcecode at (currently missing documentation) https://github.com/alexandergall/snabbswitch/tree/ipfix

Key Features

- Runs in user-space
- High-Level language (Lua)
  - Includes device-drivers
- Very fast JIT compiler (LuaJIT)
- Uses hardware/software RSS to scale well with the number of cores
- ~1500 cycles per packet (depending on features/templates)
- Easy to include more complex IPFIX templates (currently DNS/HTTP inspection)

Conclusion

- 2 RU, ~10k EUR per PoP
- Should scale up to ~25Mpps on 16 cores @2.6GHz
  - Up to 4x100Gbps between broker and exporter
- Allows us to keep producing unsampled NetFlow for the foreseeable future