



SDN-enabled Intrusion Detection System

Adam Krajewski

Liviu Valsan

Stefan Stancu

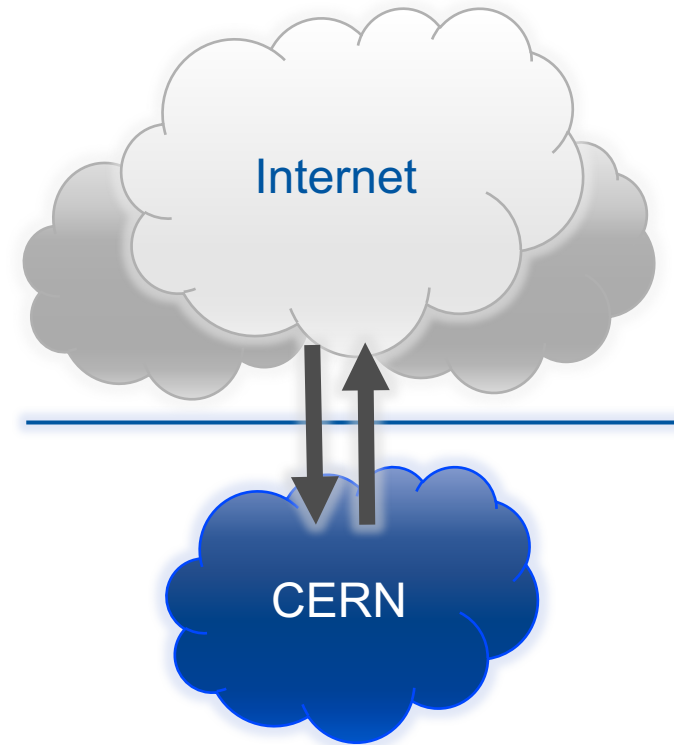
HEPiX 2016 Fall, Berkeley



Introduction

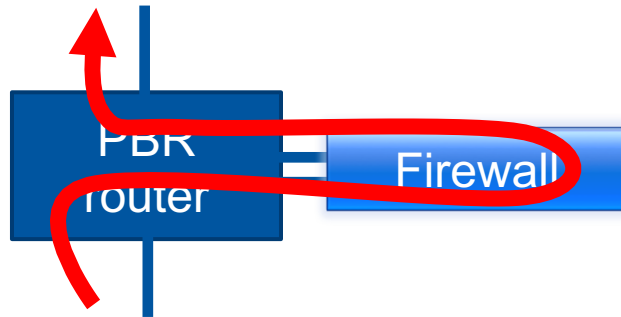
In need of scale-out

- The volume of traffic entering and leaving CERN networks grows continuously
 - LHC data, user activity, network upgrades
- Precise traffic analysis and monitoring is crucial for network security
- Scalable IDS needed

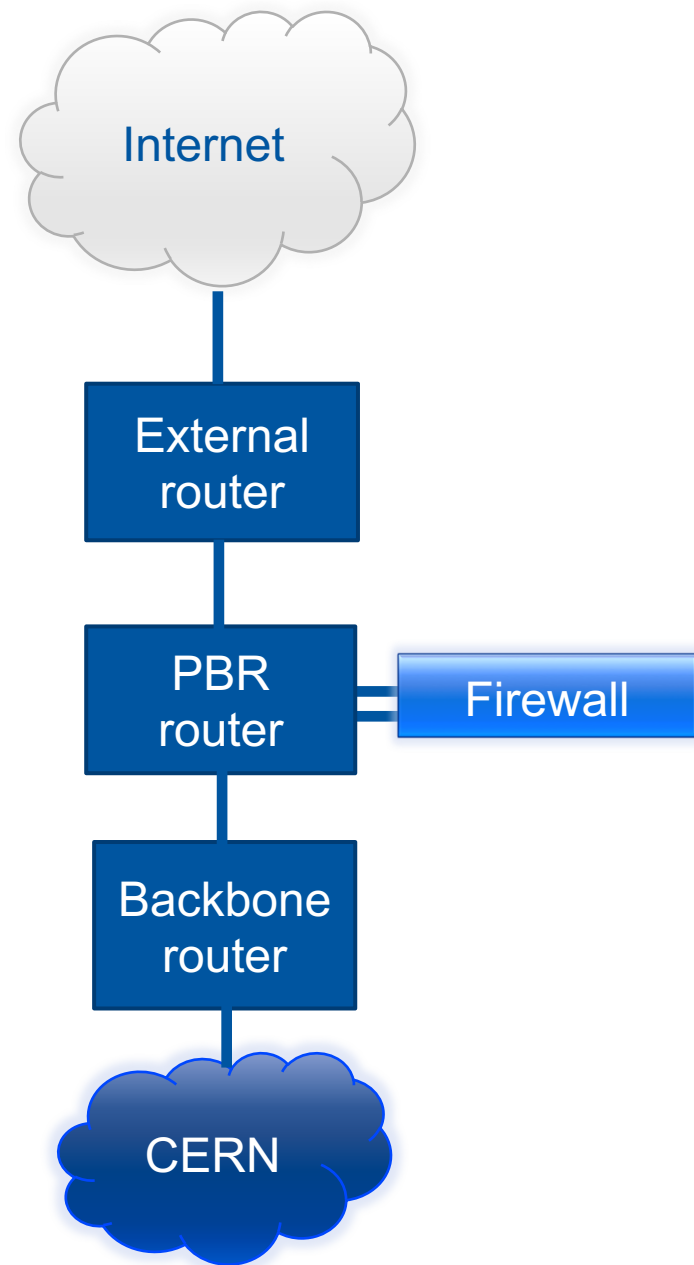
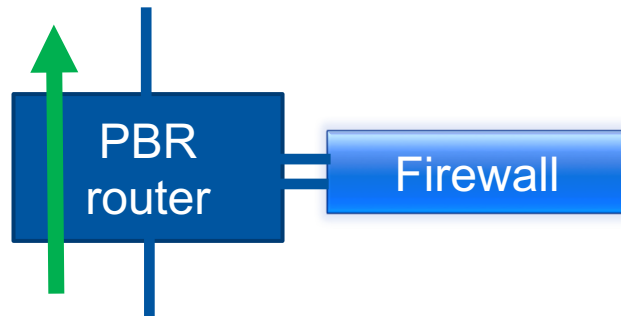


Firewall at CERN

- Regular traffic is firewalled



- Trusted traffic bypasses the firewall



Bro – IDS software

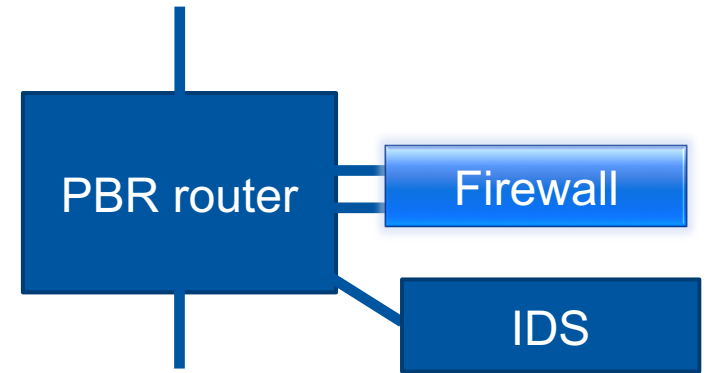
- The Bro Network Security Monitor
 - <https://www.bro.org/>
- Open source
 - Partially developed here in Berkeley at ICSI
- Comprehensive traffic analysis platform
 - Event-based model for Deep Packet Inspection (DPI)
 - Option for calling user-provided code when an event occurs
 - Bro scripts
 - Flexible, **scalable** & extensible
 - Cluster mode
 - Plugins



Current setup

Current setup

- One IDS server for each firewall
 - Receiving traffic mirrored at the firewall
 - 16 physical cores → 16 Bro processes
 - CentOS 7, Bro 2.4
- 20G monitored over a 10G link
 - Traffic losses in case of high load
- Transparent maintenance not possible
- We are looking into delivering a better, scalable setup



Planned setup

Planned setup



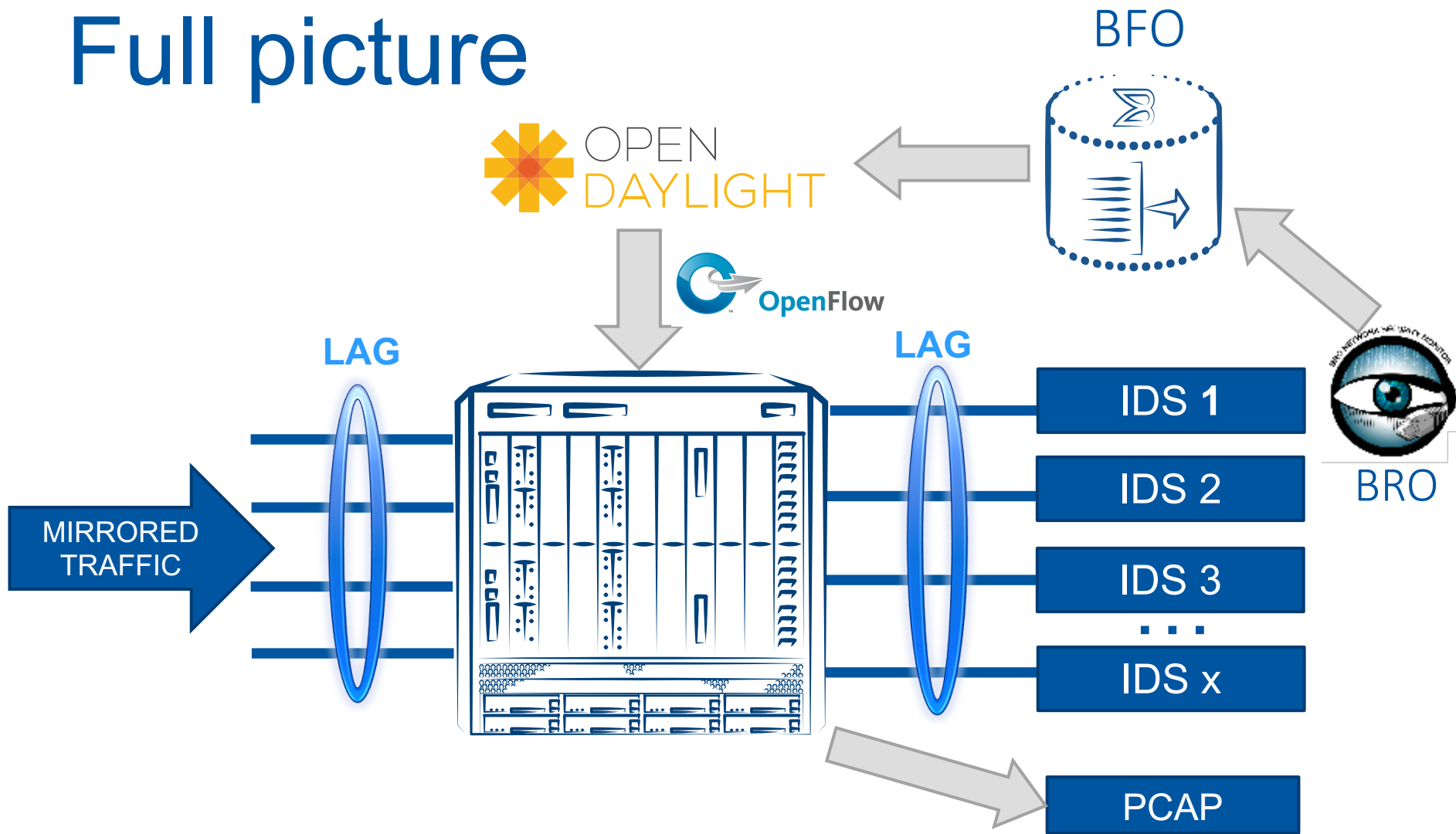
- Inspired by Berkeley Lab [1] **Brocade MLXe-16**
- The traffic mirrored at the CERN firewall is distributed across a pool of 16 servers
 - Both the bypass and the firewalled traffic are monitored
 - Production IDS cluster + test setup (Bro release evaluation) + PCAP servers
- Required features:
 - Symmetrical load-balancing
 - Traffic shunting - filtering out TCP data packets belonging to trusted flows
 - Selective mirroring – mirroring suspicious traffic to a dedicated server for detailed analysis

Software Defined Networking

- Decoupling the network control plane (decision logic) from the forwarding plane
 - Logic centralized in the SDN controller
 - Switching hardware programmed by external software
 - Improved flexibility and programmability
- Our SDN building blocks
 - **OpenFlow**
 - Interface for programming the hardware
 - OpenFlow rule = flow match + action
 - Hybrid OpenFlow simplifies deployment
 - **OpenDaylight**
 - Open-source SDN controller
 - Production-ready and endorsed by the industry
 - **Brocade Flow Optimizer (BFO)**
 - SDN application provided by Brocade
 - Monitoring large traffic flows and organizing them in a controlled manner
 - Bro plugin developed within the openlab collaboration



Full picture



Test design and status

Test case	Name	Status
TC.001	Verify L2 transparent switching on MLXe	Done
TC.002	Verify load balancing fairness	Done
TC.003	Verify load balancing symmetry	In progress
TC.004	Verify traffic shunting	Solution designed, sanity check
TC.005	Verify selective mirroring	Solution designed, sanity check

Test suite findings

Bro issues

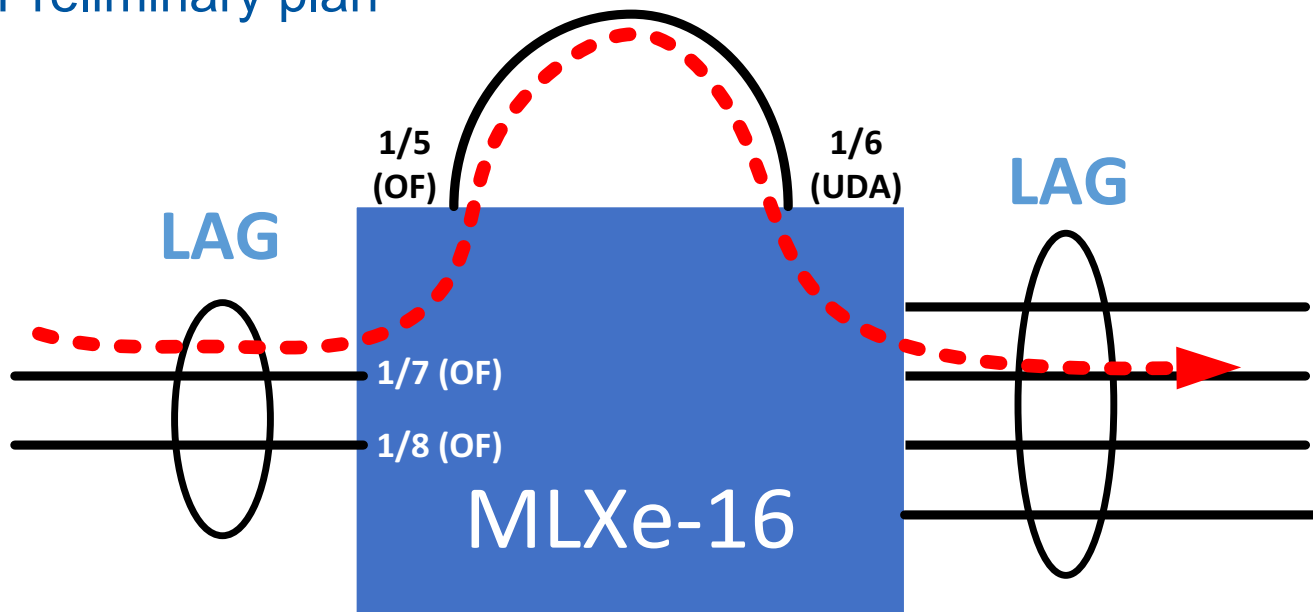
- Software-based traffic splitting to different Bro processes is preferred
 - CERN uses Intel NICs, avoiding specialized hardware and proprietary drivers
- We considered two options:
 - PF_RING
 - Mature
 - Kernel special mode required
 - AF_PACKET
 - Newer
 - Bro plugin developed at CERN
 - Not fully supported in CentOS 7
- No out-of-the-box support for PF_RING in Bro
 - Compiling Bro against upstream packages delivered by ntop provided unreliable results
 - Recompilation of PF_RING and Bro needed
 - Bro 2.5-beta

Traffic shunting – how?

- Selective matching on TCP flags needed
 - src/dst IP, src/dst port, protocol, TCP flag (SYN, FIN)
- Possible solution: User-Defined Access List (UDA) on MLXe
 - Match selected bytes on any offset
 - Problem #1: only 4 offsets allowed – can't match on everything needed
 - Problem #2: cannot enable UDA & OpenFlow at the same time

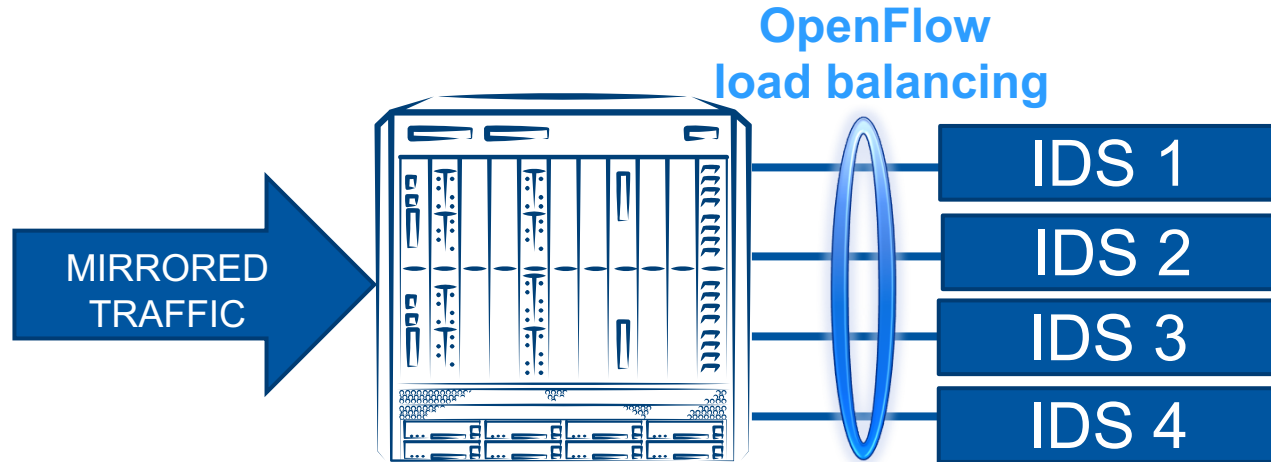
Traffic shunting – OpenFlow solution

- Proposed solution: OpenFlow + UDA
 - Use a loop and apply UDA (TCP flags) on standard port
 - OpenFlow to redirect only selected, trusted flows to the filtering loop
 - Leverage matching capabilities and programmability
- Preliminary plan



Future plans

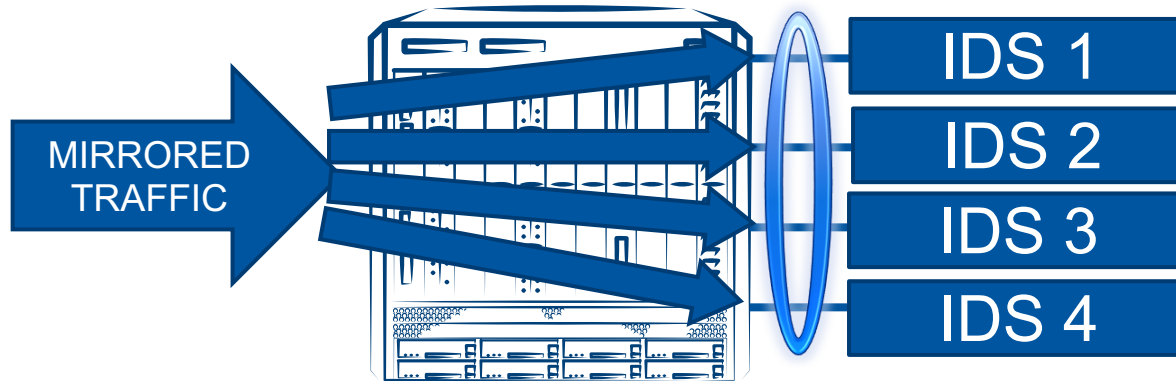
OpenFlow based load balancing



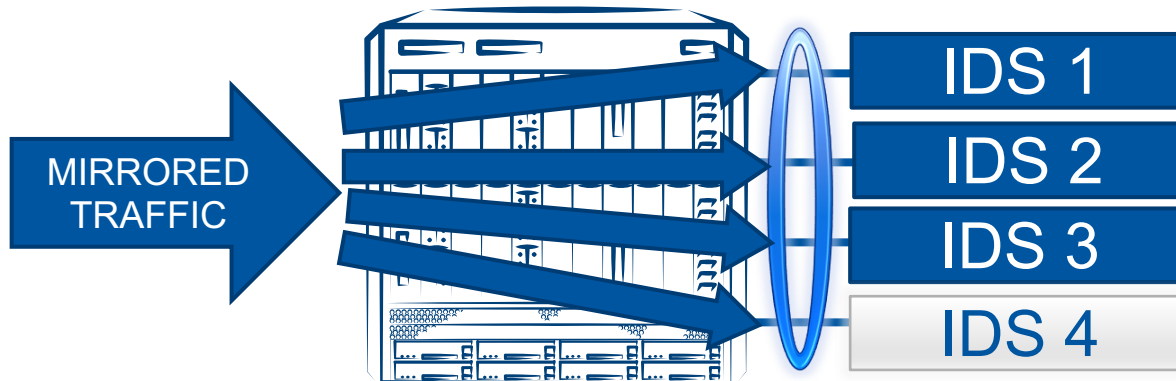
- Goal: replace “traditional” symmetrical load-balancing with software-based solution
- Motivation:
 - Provide better control over traffic redistribution mechanism
 - Integrate with the application layer
 - If the Bro IDS process dies on one of the servers we can redistribute the traffic to the remaining ones
 - Flexible traffic distribution based on server capacity and observed load

Flow draining for distributed IDS (1)

1) The traffic is uniformly distributed

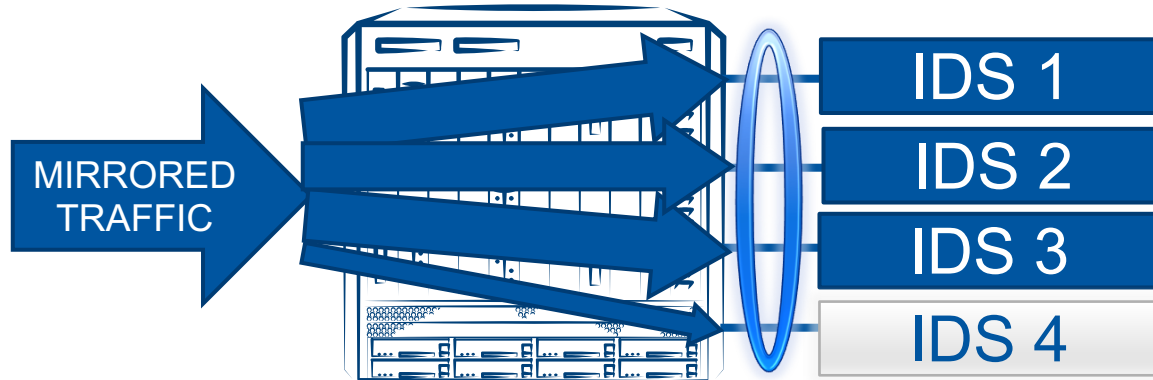


2) We mark IDS 4 as "to be removed"

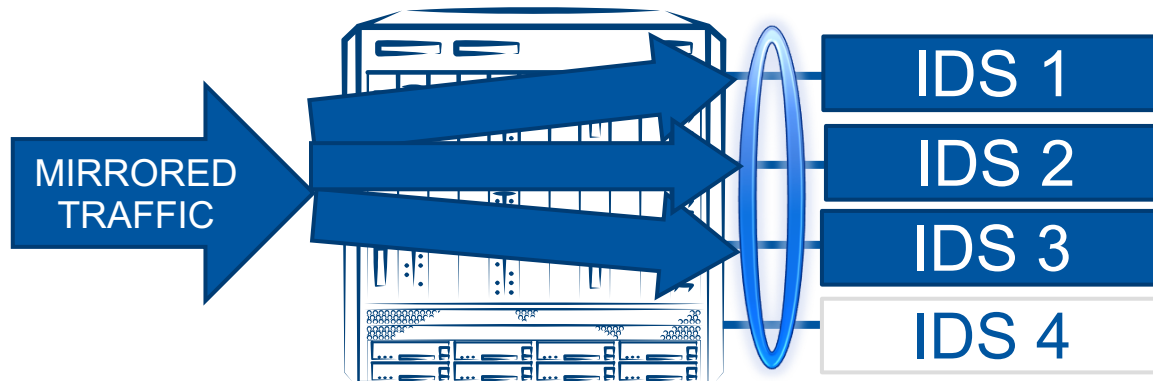


Flow draining for distributed IDS (2)

3) New flows go to IDS1-3, existing flows are still handled by IDS-4 until they end



4) IDS-4 can be safely removed from the pool when all the flows are completed or a predefined timeout is reached



Conclusions

Conclusions

- The solution looks very promising
 - Scalable & programmable
- The very first SDN deployment in the CERN network
 - Gain experience in a production environment
 - Not in the critical data-path
 - Might result in adopting SDN to face other challenges (e.g. firewall load balancing)
- Still work in progress

References

- [1] *100G Intrusion Detection*,
<https://commons.lbl.gov/download/attachments/120063098/100GIntrusionDetection.pdf>

SDN-enabled Intrusion Detection System

Adam Krajewski

Liviu Valsan

Stefan Stancu

HEPiX 2016 Fall, Berkeley

