Unleashing SuperNIC's Superpowers

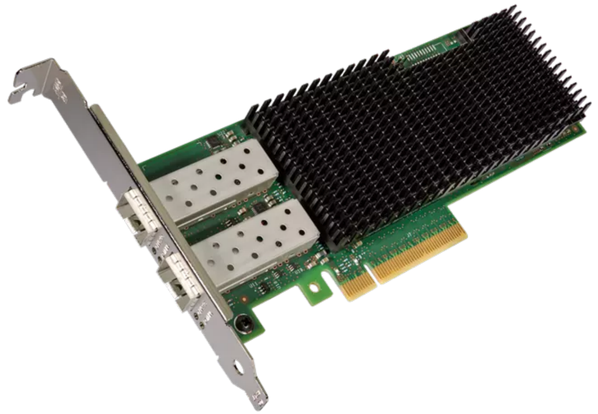
Alfredo Cardigliano <cardigliano@ntop.org>

Are SuperNICs a Game-Changer?

* for traffic analysis

NIC vs SmartNIC vs SuperNIC

- **NIC**
  - Standard network adapter with data transmission and reception.
  - Features:
    - RSS
    - Limited packet filtering

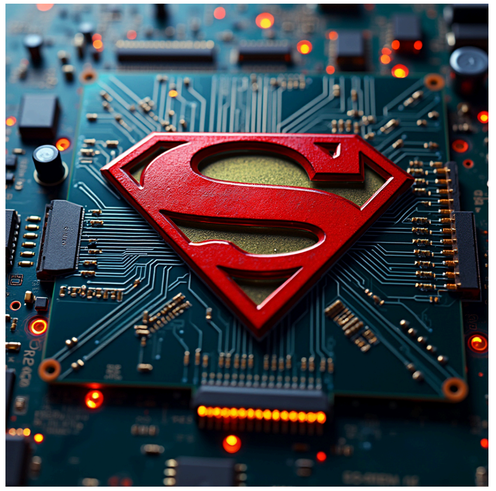


NIC vs SmartNIC vs SuperNIC

- **SmartNIC**
  - Advanced NIC (typically FPGA) able to offload and accelerate specific workloads from the CPU.
  - Features:
    - Enhanced packet parsing
    - Load balancing
    - Packet filtering
    - Limited programmability
    - Optimized data transfer

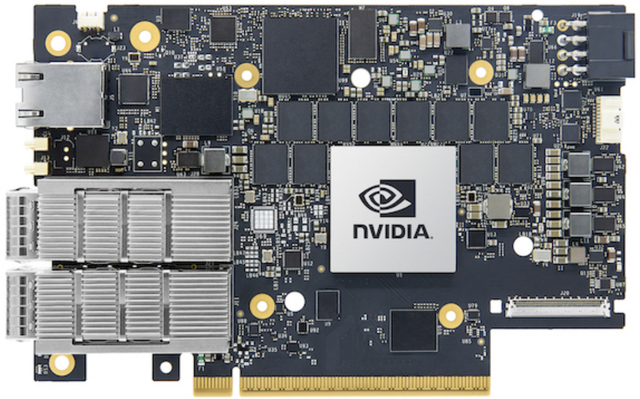
NIC vs SmartNIC vs SuperNIC

- **SuperNIC**
  - More advanced version of a SmartNIC, with next-generation network processing capabilities.
  - Features:
    - Hardware accelerators
      - Network, Encryption, Compression, Storage, etc.
    - Integrated compute (onboard CPU)
    - AI acceleration (direct GPU connectivity)
    - A lot of other marketing stuff depending on vendor



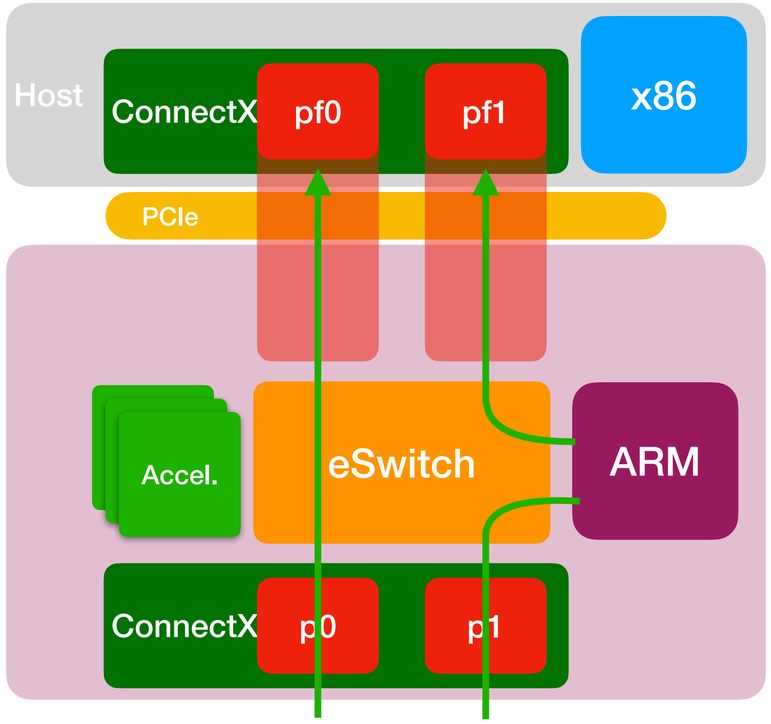
BlueField-3 SuperNIC

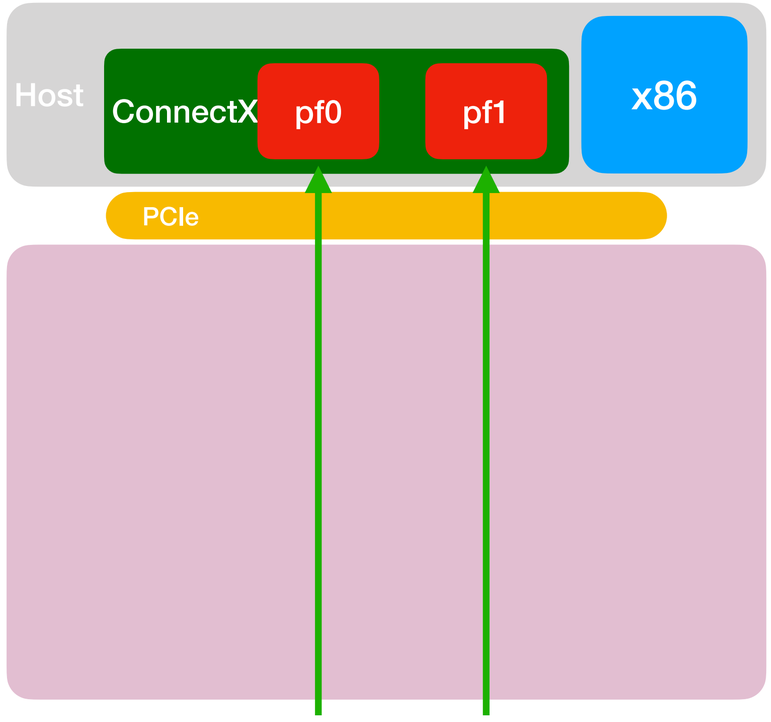
- ARM CPU (16-cores A78)
- 16GB DDR5 on-board
- Up to 400 Gbit connectivity
- PCIe Gen 5.0
- Hardware accelerators (for all tastes)
- Programmable with DOCA
- ConnectX-7 interfaces



Operating Modes

- DPU Mode
  - NIC owned and controlled by the embedded ARM subsystem.
- NIC Mode
  - Behaves exactly like a standard (ConnectX) NIC.

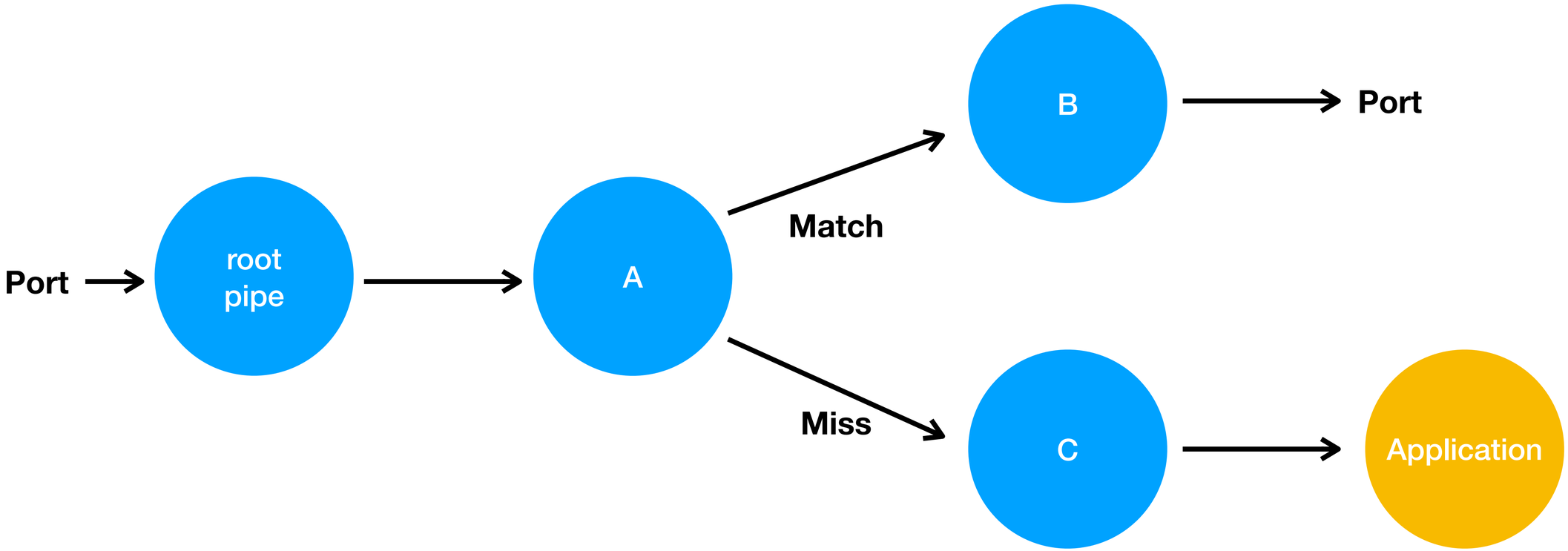




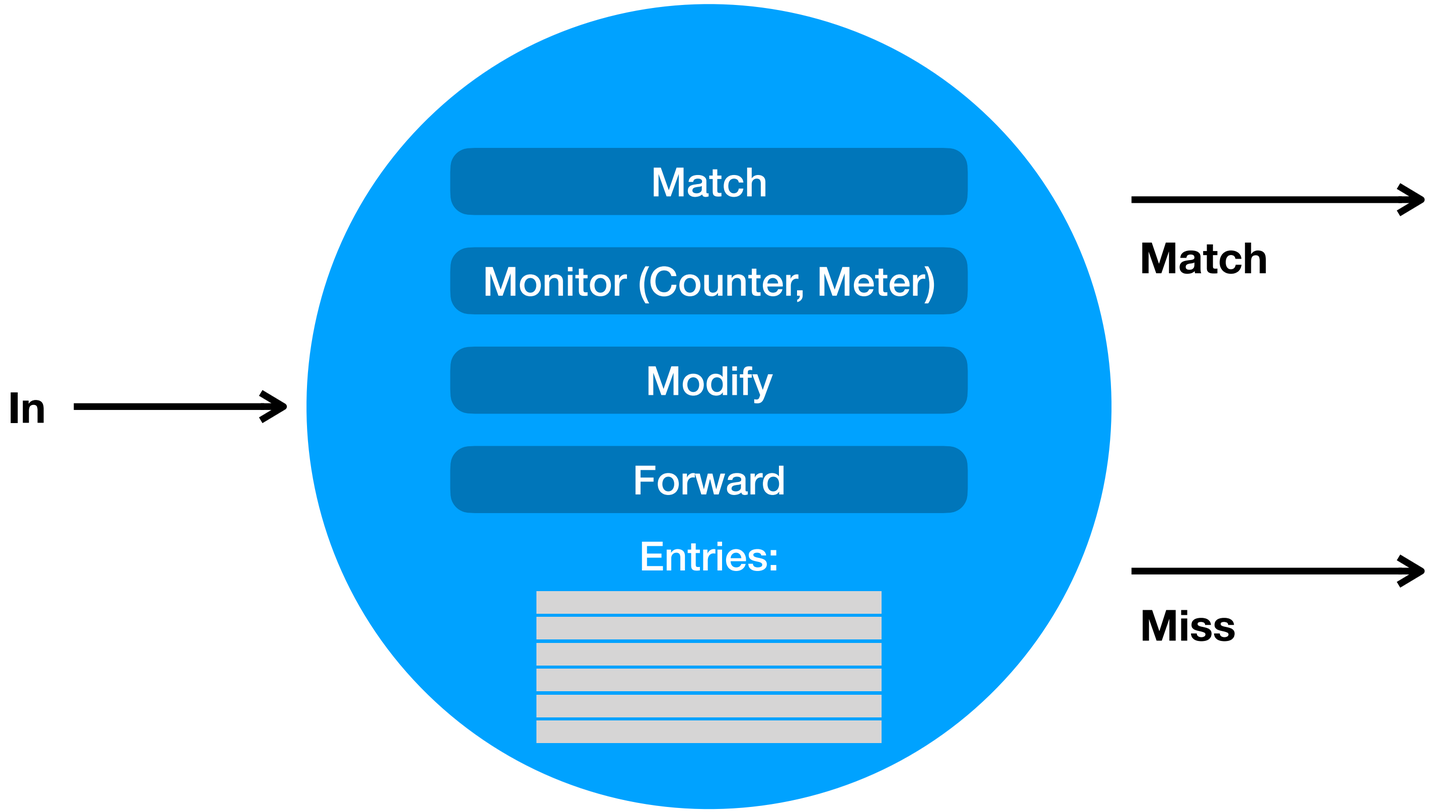
DOCA Flow

- DOCA is an SDK for programming hardware accelerators on NVIDIA adapters
- Divided into components (libraries), e.g. Flow, Compress, etc.
- Supported both on NVIDIA BlueField and ConnectX (depending on the component)
- DOCA Flow is the one we need to accelerate packet processing
- APIs for building processing pipelines by creating pipes and chaining them

Pipeline



Pipe


Match
Monitor (Counter, Meter)
Modify
Forward
Entries:
In
Match
Miss

Are SuperNICs Better Than SmartNICs?

* for traffic analysis

Network Monitoring Use Case

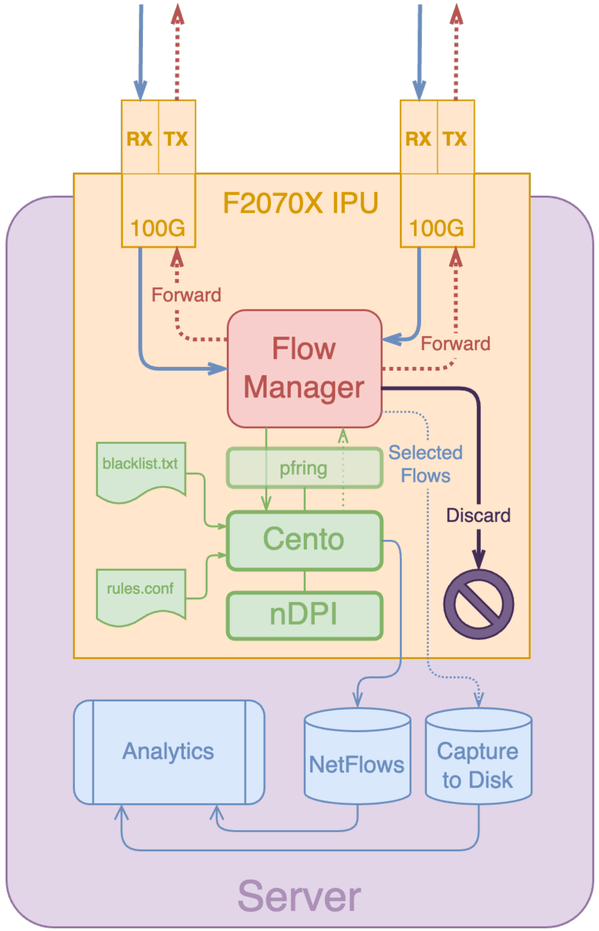
- Feeding a Network Monitoring application on high speed links has been made possible with packet capture acceleration technologies:
  - By enabling kernel-bypass (application-level DMA)
  - Exploiting modern multi-cores CPU by load-balancing traffic in hardware with RSS-like technologies.
  - Other offloads (filtering, etc).

Stateful Traffic Processing

- Monitoring applications, both passive (e.g. NetFlow) or inline (e.g. IPS systems), typically have to analyse and maintain the state of each network communication.
- This requires a flow table, an in-memory data structure where the application keeps the status of network communications (flow key and metadata).
- Metadata may include:
  - Simple statistics about the packet stream (number of packets and bytes)
  - Information from application layer protocols (e.g. the HTTP URL or the VoIP caller) extracted by DPI engines

Flow Table Offload

- Hardware-accelerated flow offload mechanisms can be seen as the next generational step of acceleration technologies.
- Consolidated idea (already working on Napatech SmartNICs) to accelerate stateful traffic processing.
- Software still needs to be involved (e.g. DPI dissectors), at least at the beginning of the communication.



How It Works

1. Capture a packet
2. Extract the 5-tuple
3. (Optional) Run DPI on the payload
4. When it's time to offload (1st packet, or when DPI has done), add a new entry to the hardware flow table
5. Periodically read stats from the hardware entry and handle expiration

Flow Table Offload on BlueField

- Can we use DOCA Flow on BlueField for Network Monitoring acceleration?
- Building a stateful traffic processing implementing Flow Table Offload seems feasible.
  - DOCA Flow seems to be a good fit for this
- Let's build a Proof of Concept to test the DOCA Flow features and performance on BlueField.

DOCA Flow CT Pipe

- DOCA Flow CT (Connection Tracking) seems to provide all the ingredients we need:
  - 5-tuple table to store entries (flows)
  - API to add, remove, update entries
  - Per-entry statistics
  - Flow aging
- Available both on the BlueField DPU and on the host (also on plain ConnectX!)

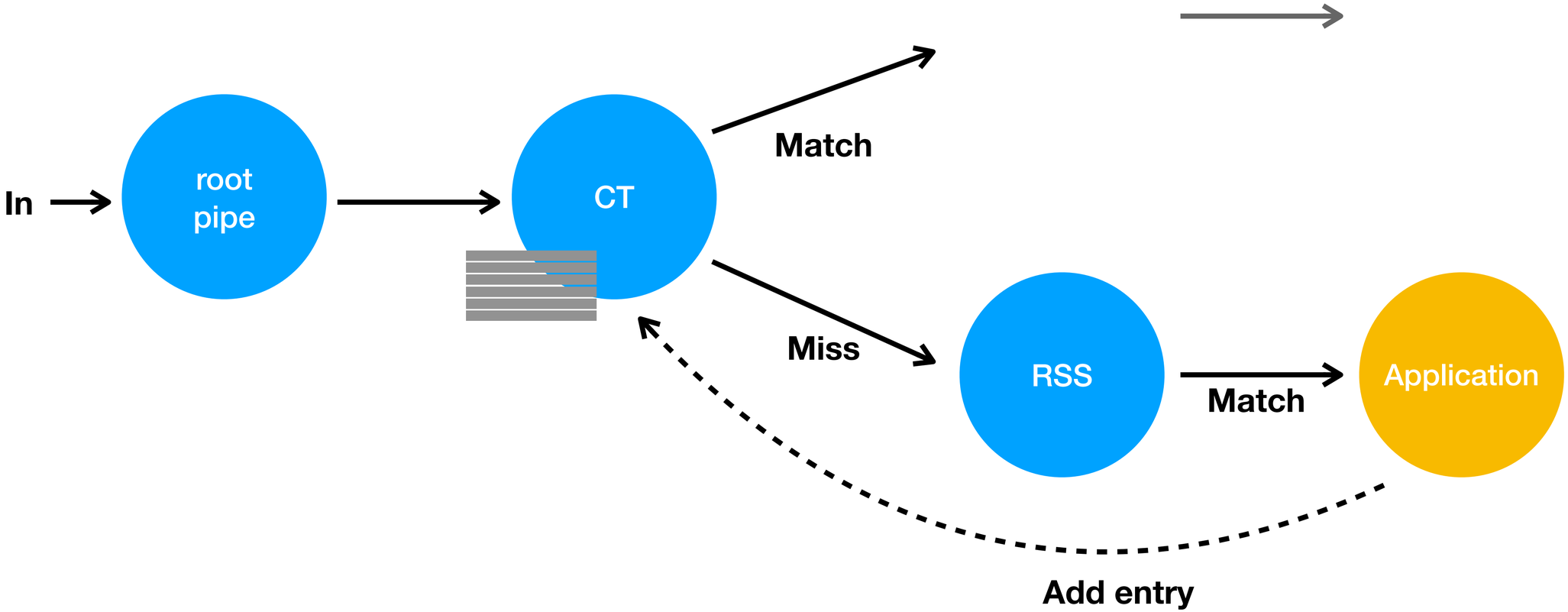
Groping in the Dark

- Documentation (online) sucks
  - You need to guess what most DOCA APIs and data structures do
  - API and docs change quite a bit depending on DOCA version
  - Different features on different adapter model
- Examples don't help much as they are too limited
  - Not designed for live traffic
- Configuring the adapter is a pain in the a**
  - Following the instructions in the guide doesn't always work
- BUT, when everything works, you have a lot of fun

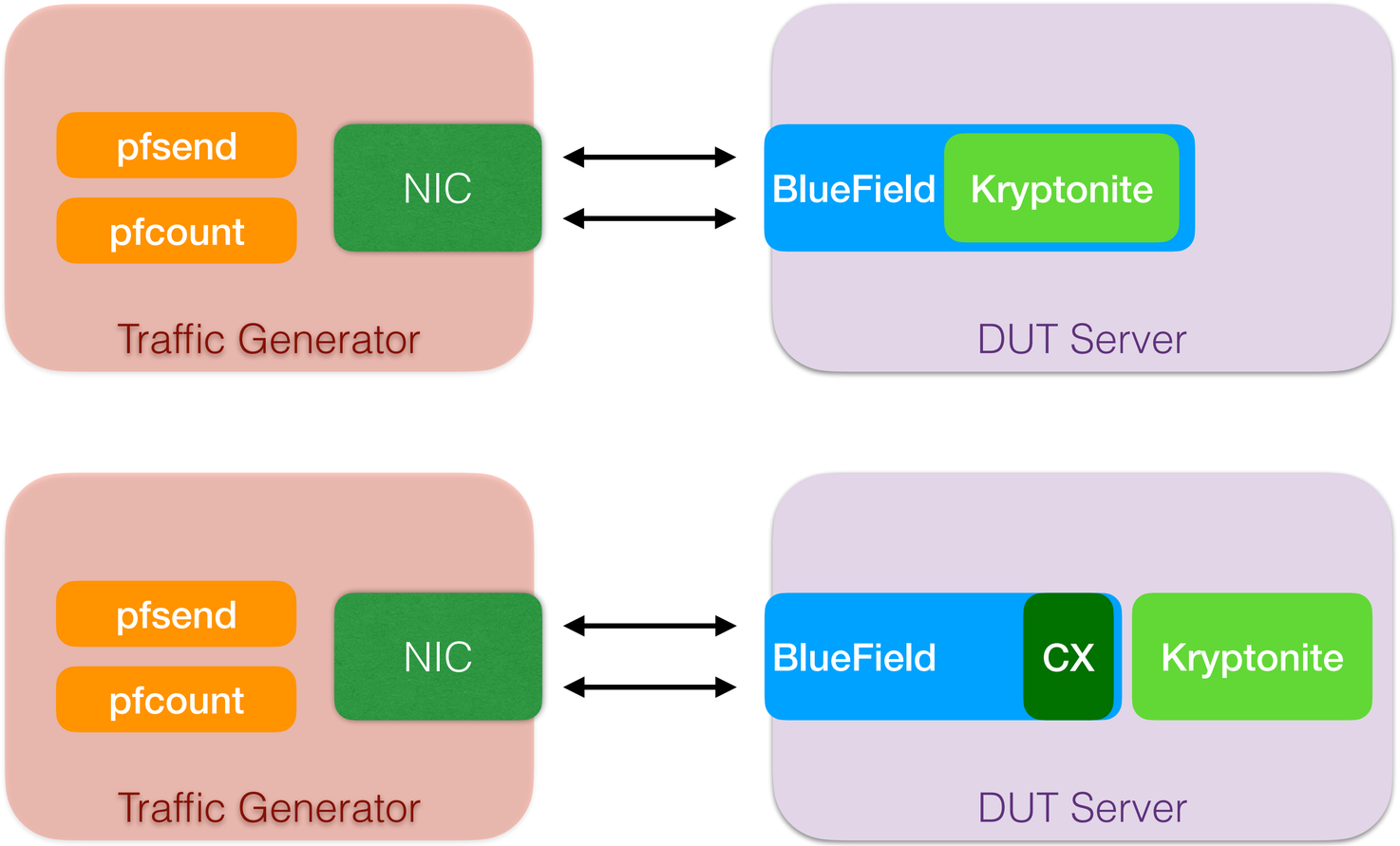
"Kryptonite"

- One-file application using DOCA Flow, aiming to be the ultimate example
- Implement Flow Offload using DOCA Flow CT
- Shadow (Software) Flow Table
  - Keep track of offloaded flows
  - Store additional metadata (e.g. DPI)
- Flow export (dump as text)
- (Optional) Inline traffic forwarding
- Enhanced statistics (performance!)
- Source code: https://github.com/ntop/bluefield-kryptonite

Kryptonite CT-based Pipeline



Test Plant



- On BlueField-3 DPU (ARM)
- On Host (NIC Mode) (Intel Xeon Gold 6526Y)

Test Results

- 2 Million flows generated in all tests (maximum configurable on BlueField-3 according to our tests)
- The traffic generator was able to generate:
  - 100 Gbps with 200-byte packets (55.8 Mpps)
  - 40 Gbps with 60-byte packets (60 Mpps)

➡ Measured CT performance:

  - Max flow creation rate: 2.7 Million flows/sec
  - Full rate (no loss) for traffic handled by the pipeline (Fast Path)
    - Packet loss only occurs (on the Slow Path) when flows rate exceeds the max creation rate

SmartNIC vs SuperNIC

- SmartNIC (Napatech FPGA) with Flow Manager
  - 140 Million flows
  - 1.5 Million flows/sec / 3 Million flows/sec with multiple streams
  - Selected actions
- SuperNIC (BlueField-3) with DOCA Flow CT
  - 2 Million flows (maximum configurable in our tests)
  - 1.3-1.5 M flows/s on DPU / 1-2.7 Million flows/sec on x86 (NIC mode)
  - Programmable actions

Conclusions

- Pros
  - High programmability
  - High (pipeline) performance
  - Offload whole application to the DPU cores
- Cons
  - Lower resources (#sessions) with respect to specialized FPGAs
  - (Development and) configuration is not straightforward
  - Slow path may be a bottleneck, especially on the DPU
  - Tied to DPDK

- Source code available at https://github.com/ntop/bluefield-kryptonite