Squeezing Network Adapters

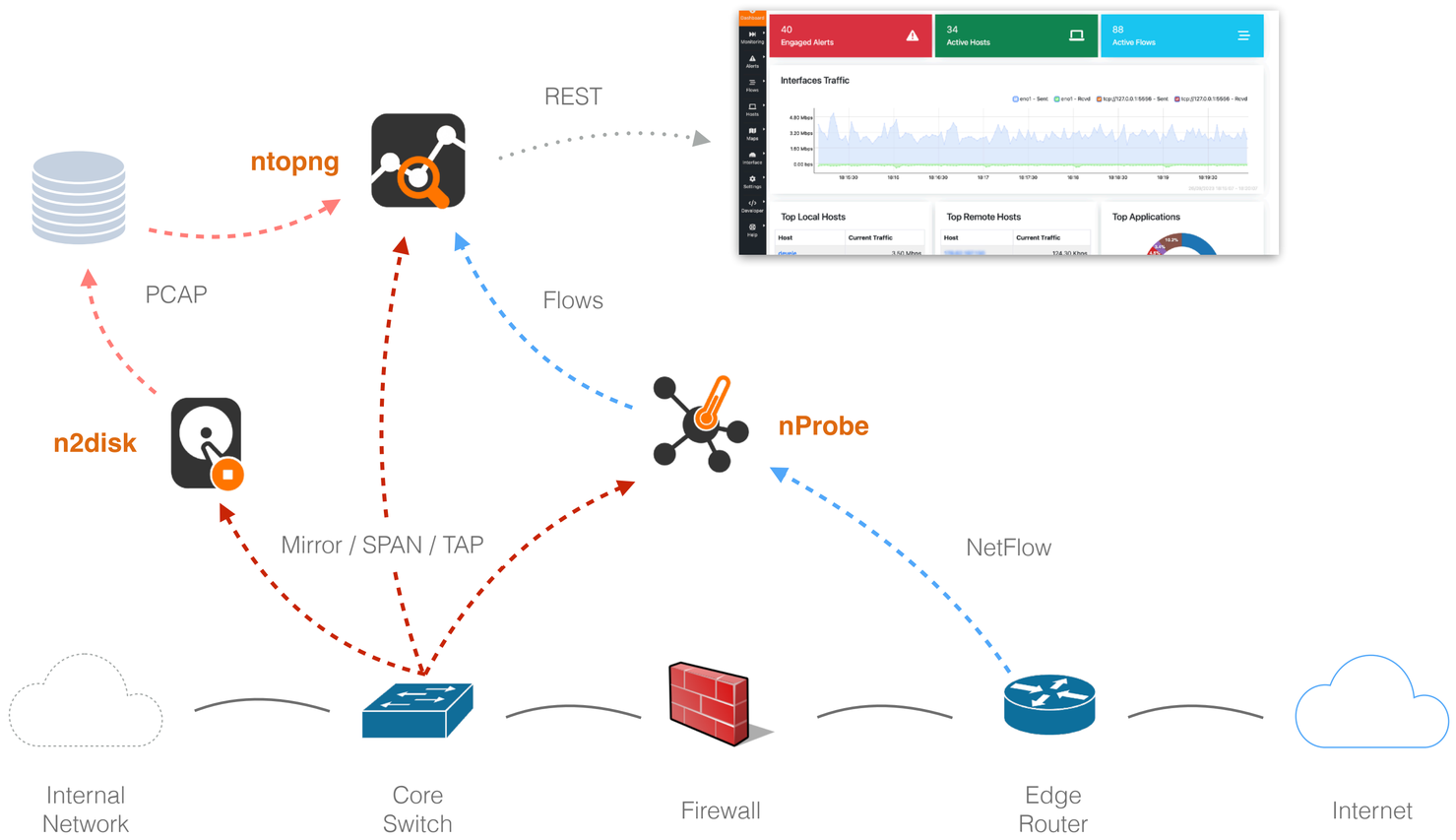
Tips and tricks to offload and scale up.

Alfredo Cardigliano
cardigliano@ntop.org

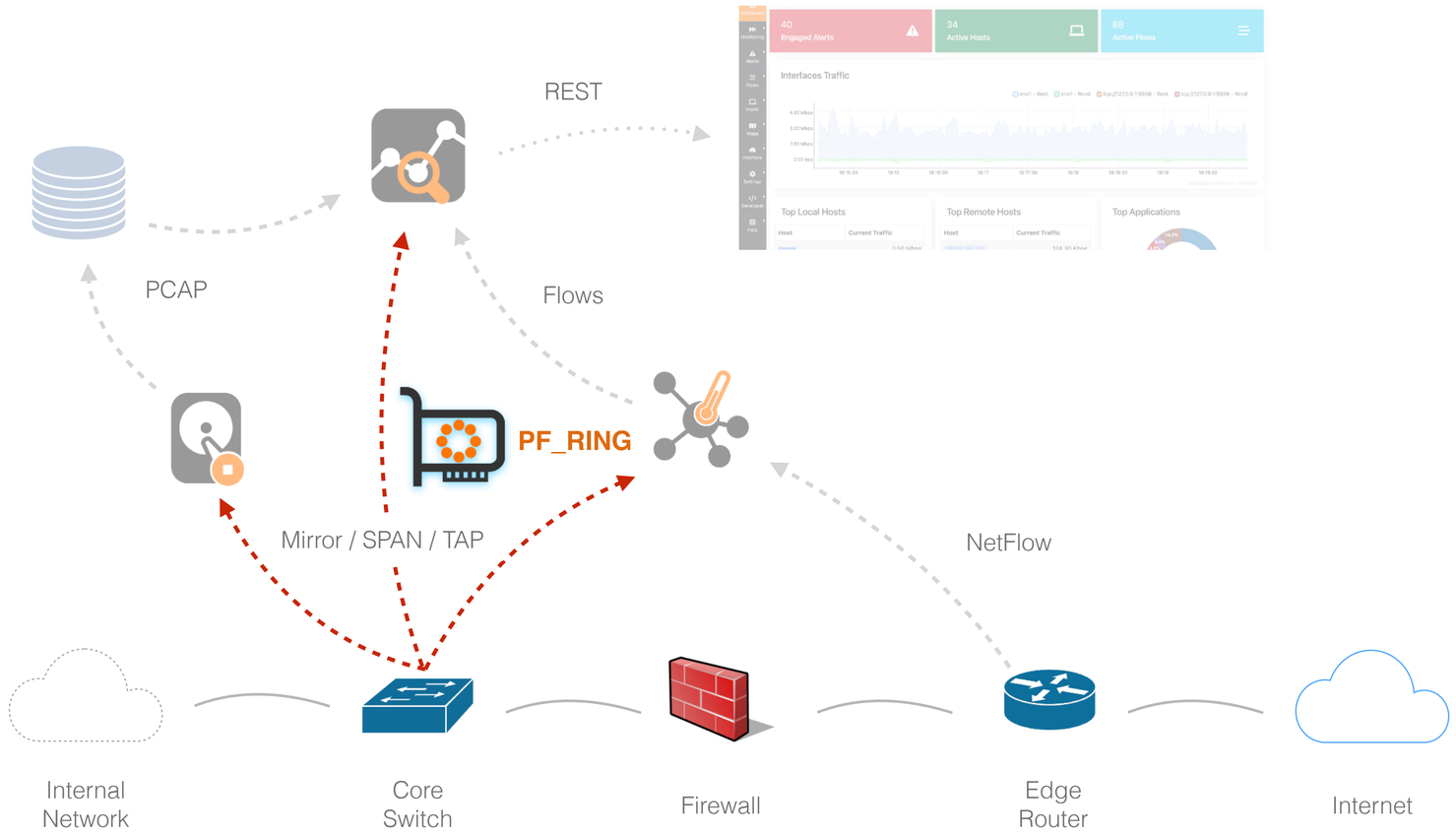
ntop

PacketFest'25

ntop Ecosystem



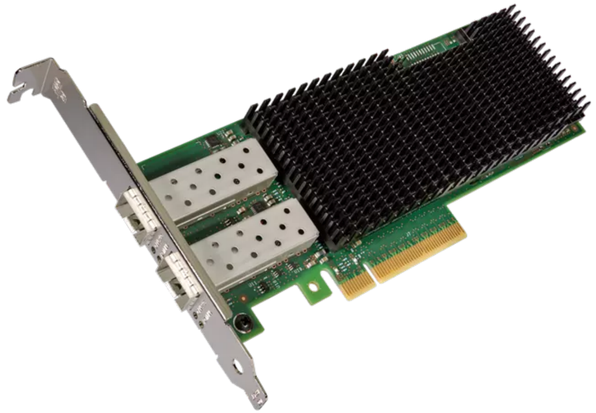
ntop Ecosystem



NIC vs SmartNIC vs SuperNIC

Standard NIC

- Standard network adapter (Intel, Broadcom, ...) with data transmission and reception.
- Features:
  - RSS
  - Limited packet filtering
  - Cheap, but still wire-rate capture with **PF_RING ZC** on Intel, etc.



SmartNIC

- Advanced, typically FPGA-based, adapter (Napatech, Silicom Fiberblaze, ...) able to offload and accelerate specific workloads from the CPU.
- Features:
  - Enhanced packet parsing
  - Load balancing
  - Packet filtering
  - Some programmability (e.g. NTPL)
  - Optimized data transfer (segment mode)

SuperNIC

- Programmable adapter (NVIDIA BlueField, Pensando, ...) with enhanced accelerators and processing capabilities.
- Features:
  - Onboard CPU
  - Hardware accelerators
    - Networking, Encryption, Compression, Storage, etc.
  - AI acceleration (direct GPU connectivity)

(Zero-Copy) Hardware Fanout

on Standard NICs

Packet Sniffing

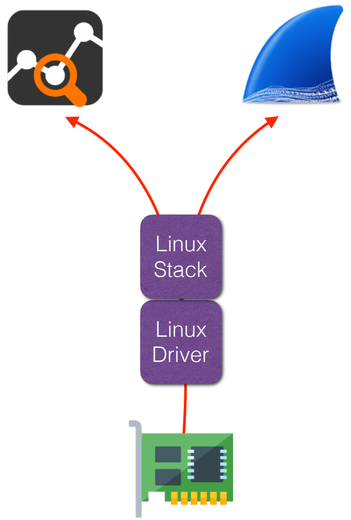
with Standard Drivers

- Running multiple monitoring applications on the same interface (e.g. ntopng, tcpdump, Wireshark, Suricata) is a common practice

```
ntopng -i eth1
```

```
tcpdump -i eth1
```

- The stack takes care of copying packets to all consumers



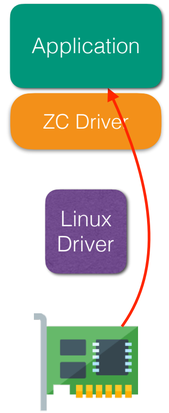
Kernel-Bypass (Zero-Copy)

- Kernel-bypass drivers (including PF_RING ZC) take full control on the network adapter (there is no kernel in the middle, thus no packet copy)
- Pros: optimal performance as packets are delivered to the application directly
- Drawback: on many adapters, including Intel, same traffic cannot be processed by multiple applications simultaneously (**exclusive access**)

```
ntopng -i zc:eth1
```

```
~~tcpdump -i zc:eth1~~
```

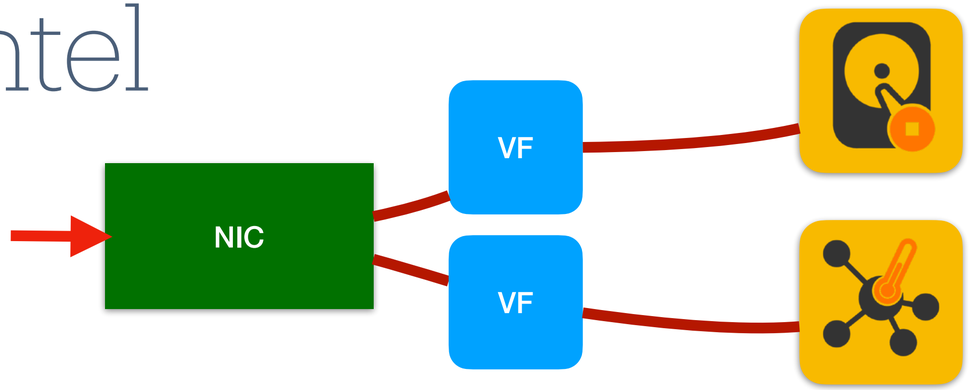
  - Note: this is not the case of NVIDIA ConnectX and some FPGAs which support packet duplication in hardware



VF to the rescue

- Any chance we can enable hardware fan-out on Intel (in zero-copy mode)?
- When enabling SR-IOV (commonly used by Virtual Machines) modern Intel adapters support a working mode called ***trust mode*** which has been introduced to enable promiscuous capture on Virtual Functions (VF)
- This in practice duplicates all traffic hitting the physical interface to all VF
- Intel i40e (X710/XL710) and ice (E810) SR-IOV Virtual Functions are supported (iavf ZC driver enables capture acceleration on them)

Enabling Trust Mode on Intel



- Configure a couple of Virtual Functions

```
echo '2' > /sys/bus/pci/devices/$(ethtool -i eth1 | grep bus-info | cut -d' ' -f2)/sriov_numvfs
```

- Set trust mode on all VF

```
ip link set eth1 vf 0
ip link set eth1 vf 1
ip link set dev eth1 vf 0 trust on
ip link set dev eth1 vf 1 trust on
```

- Run the applications on VF interfaces

```
ntopng -i zc:eth1v0
```

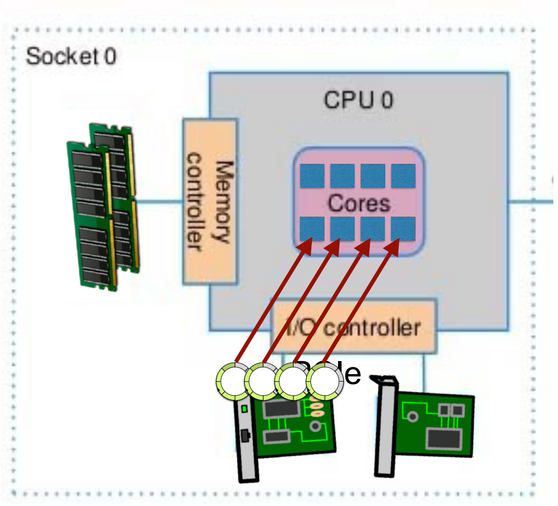
```
tcpdump -i zc:eth1v1
```

Custom RSS

on Standard NICs

RSS

Receive Side Scaling



- Available on almost all adapters (with different names)
- Hardware technology that distributes the load across multiple queues
- Packet distribution is usually based on a **hash on the 5-tuple** (flow key)
  - Support for (bidirectional) flow coherency is a must

Hash Fields (Built-in)

- Many modern adapters (including Intel E810) allow (limited) hash customization
- Changing fields with ethtool:

```
ethtool -N eth1 rx-flow-hash tcp4 sd
```

- Where:
  - s: Source IP address
  - d: Destination IP address
  - f: Source port
  - n: Destination port
- Other fields available with PF_RING ZC drivers (v: VLAN, m: MAC address)

Hash Fields (Custom)

- Further hash customizations are possible on Intel E800 series adapters
- Properties of protocol header fields (e.g. field offset) used by the (hardware) packet parser are configured by the driver
- Example:
  - Use case: Load-balance PPPoE traffic on Session ID
  - Trick: Change the VLAN offset to match the PPPoE Session ID field

```
-       ICE_FLOW_FLD_INFO(ICE_FLOW_SEG_HDR_VLAN, 14, ICE_FLOW_FLD_SZ_VLAN),
+       ICE_FLOW_FLD_INFO(ICE_FLOW_SEG_HDR_VLAN, 16, ICE_FLOW_FLD_SZ_VLAN),
```

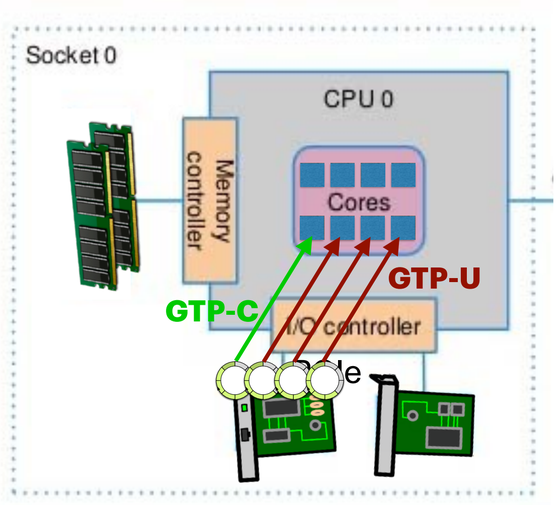
```
ethtool -N eth1 rx-flow-hash tcp4 v
```

RSS and GTP

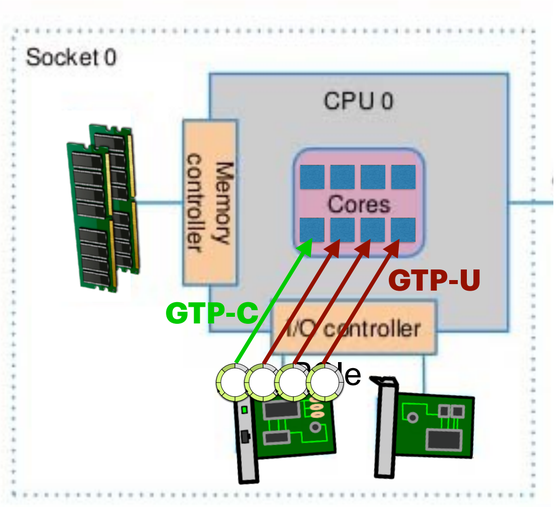
on Standard NICs

RSS on GTP Traffic

- When processing GTP traffic, we usually want to:
  - Process GTP-C traffic on a dedicated process
  - Load balance GTP-U traffic to multiple cores



Combining RSS with Flow Steering



- Configure RSS queues:

```
ethtool --set-channels eth1 combined 8
```

- Configure GTP-C steering to queue 0:

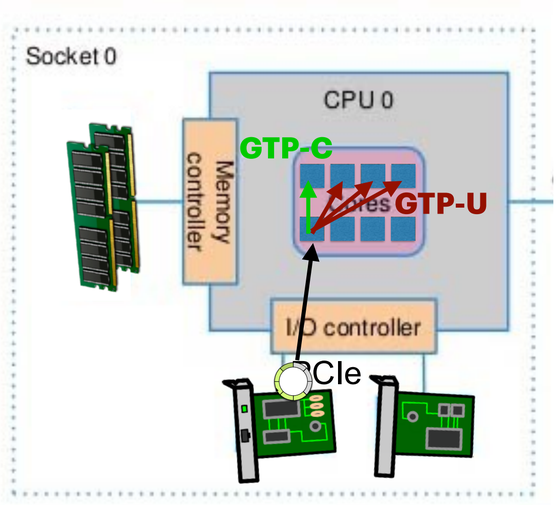
```
ethtool -U eth1 flow-type udp4 src-port 2123 action 0
```

- Distribute traffic with RSS to all other queues:

```
ethtool -X eth1 weight 0 1 1 1 1 1 1 1
```

Software Load-Balancing

with PF_RING ZC



- zbalance_ipc: sample application using the PF_RING ZC framework
- Provides zero-copy flexible Load-Balancing and Fan-Out in software
- Egress to queues or physical interfaces
- Configuration example to load-balance GTP traffic:

```
-i=zc:eth1 # Capture interface
-n=8       # Load-balance to 8 consumers
-m=4       # Hash on GTP content
-G=0:2     # Deliver GTP-C v2 to consumer 0
```

Flow Tracking (Offload)

on SmartNICs

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 (N-tuple)
  - 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 (e.g. nDPI)

Flow Table Offload

on Napatech / Flow Manager

1. Application captures a packet

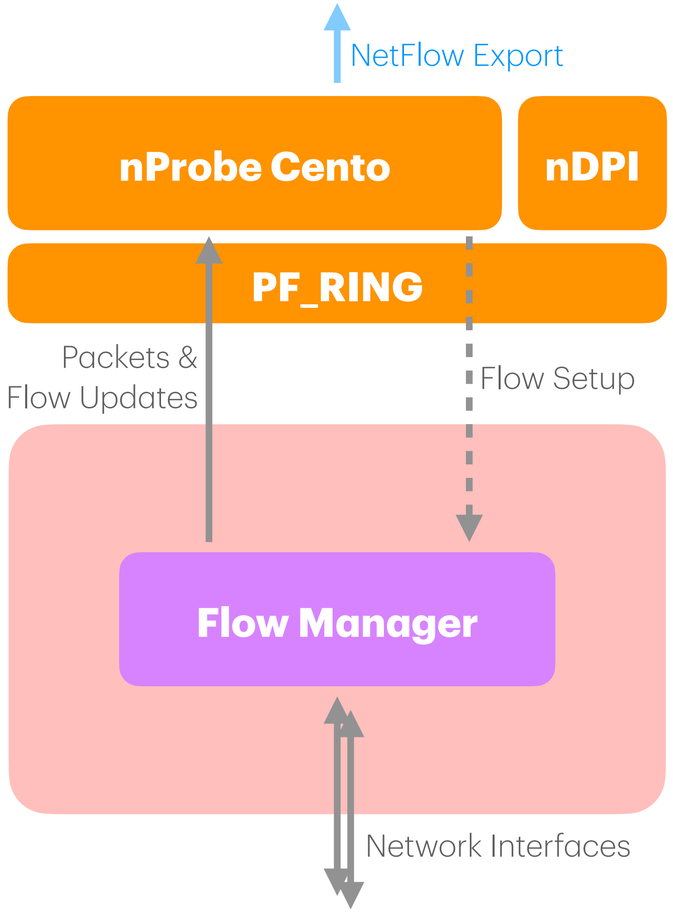
2. Extract the 5-tuple

3. (Optional) Run DPI on the payload in software

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

(Optional) Inline forwarding with L7 filtering



Performance

on Napatech NT200 Flow Manager

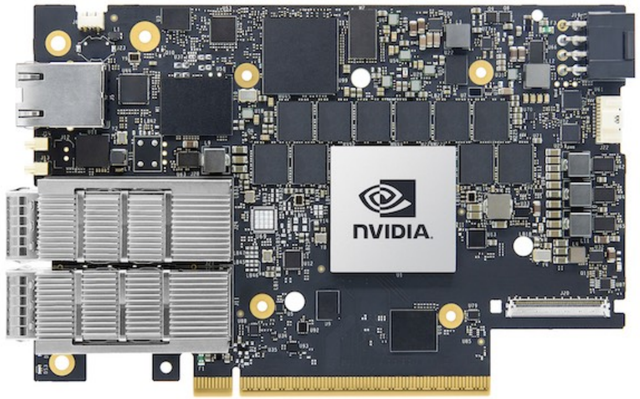
- 100 Gbps full rate packet processing
- Flow creation rate:
  - 1.5 Million flows/sec with one stream
  - 3+ Million flows/sec with multiple streams
- Cache capacity: 140 Million flows

Flow Tracking (Offload)

on SuperNICs

BlueField-3

- Embedded ARM CPU
- Hardware accelerators
  - Networking, Storage, Security, …
- Programmable with DOCA
- ConnectX-7 interfaces

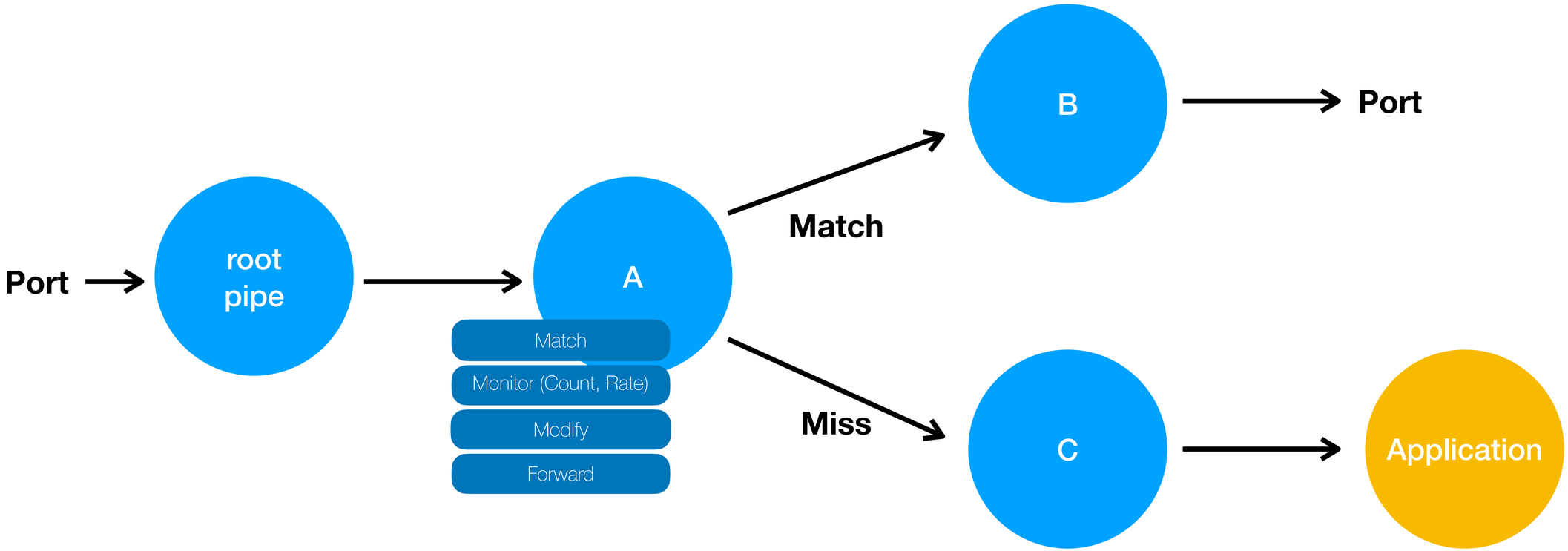


Flow Table Offload

on BlueField / DOCA

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

Pipeline


Port
root
pipe
A
Match
Monitor (Count, Rate)
Modify
Forward
Match
B
Port
Miss
C
Application

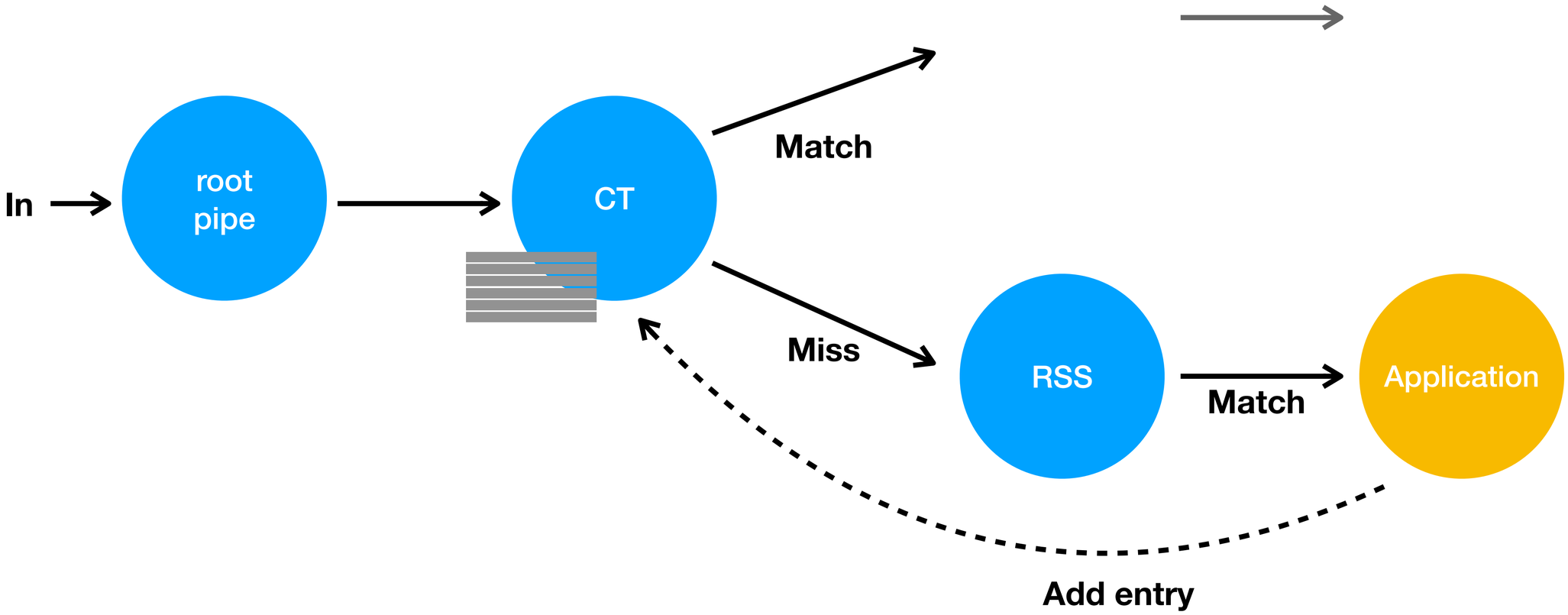
DOCA Flow CT

- Special DOCA Flow pipe providing Connection Tracking:
  - 5-tuple table to store entries (flows)
  - API to add, remove, update entries
  - Per-entry statistics
  - Flow aging support (expiration)
- Available both on the BlueField DPU and plain ConnectX 6/7 adapters

"Kryptonite"

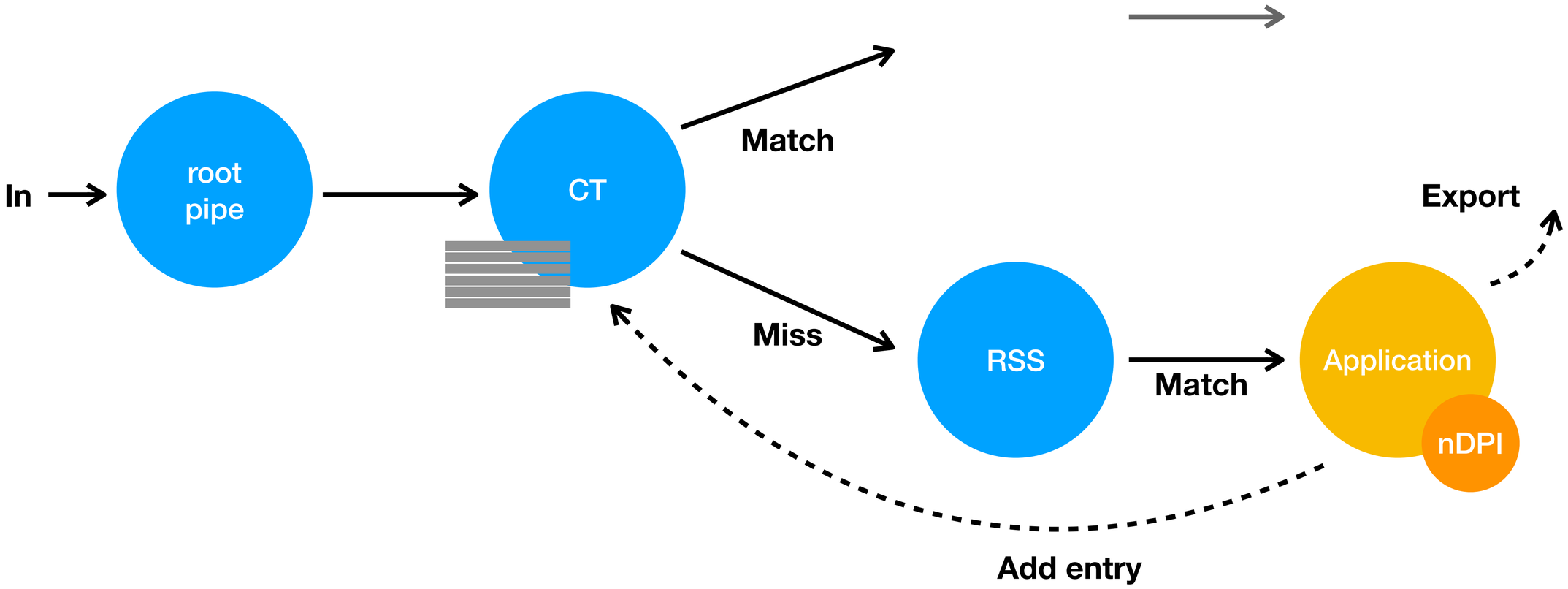
- Implement Flow Offload using DOCA Flow CT
- Include a shadow (Software) Flow Table
  - Keep track of all offloaded flows and metadata
- Dump flows as output (text)
- (Optional) Inline traffic forwarding with flow verdicts

Kryptonite CT-based Pipeline



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

Kryptonite-dpi (Flow Export with DPI)



Performance

on BlueField-3 with DOCA Flow CT

- Tested up to 100 Gbps with 200-byte packets (55.8 Mpps) and 40 Gbps with 60-byte packets (60 Mpps) with no loss *
- Flow creation rate:
  - 1.3-1.5 M flows/s on DPU
  - 1-2.7 Million flows/sec on x86 (NIC mode / ConnectX)
- * Packet loss occurs on the Slow Path when flow rate exceeds the max creation rate

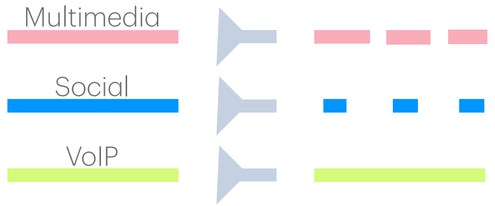
- Cache capacity: 2 Million concurrent flows (max configurable on BlueField-3)

Traffic Shaping

on SuperNICs

Traffic Rating

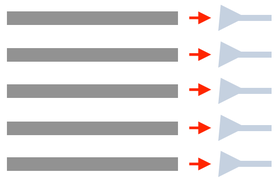
Use Cases



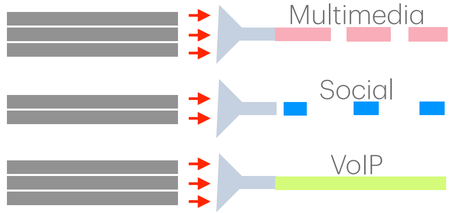
- **QoS and Prioritization** for Business-Critical Applications
  - Prioritize VoIP and video conferencing over recreational traffic
  - Ensure essential operations function smoothly even during peak usage times.
- **Internet Service Plans** Based on Application-Specific Performance
  - Allow monetization of QoS by differentiating traffic classes
- **Application-Specific Boosting** or Throttling
  - Enhances user experience for latency-sensitive services (e.g. gaming) without upgrading full bandwidth

Traffic Meters (Offload)

on BlueField / DOCA



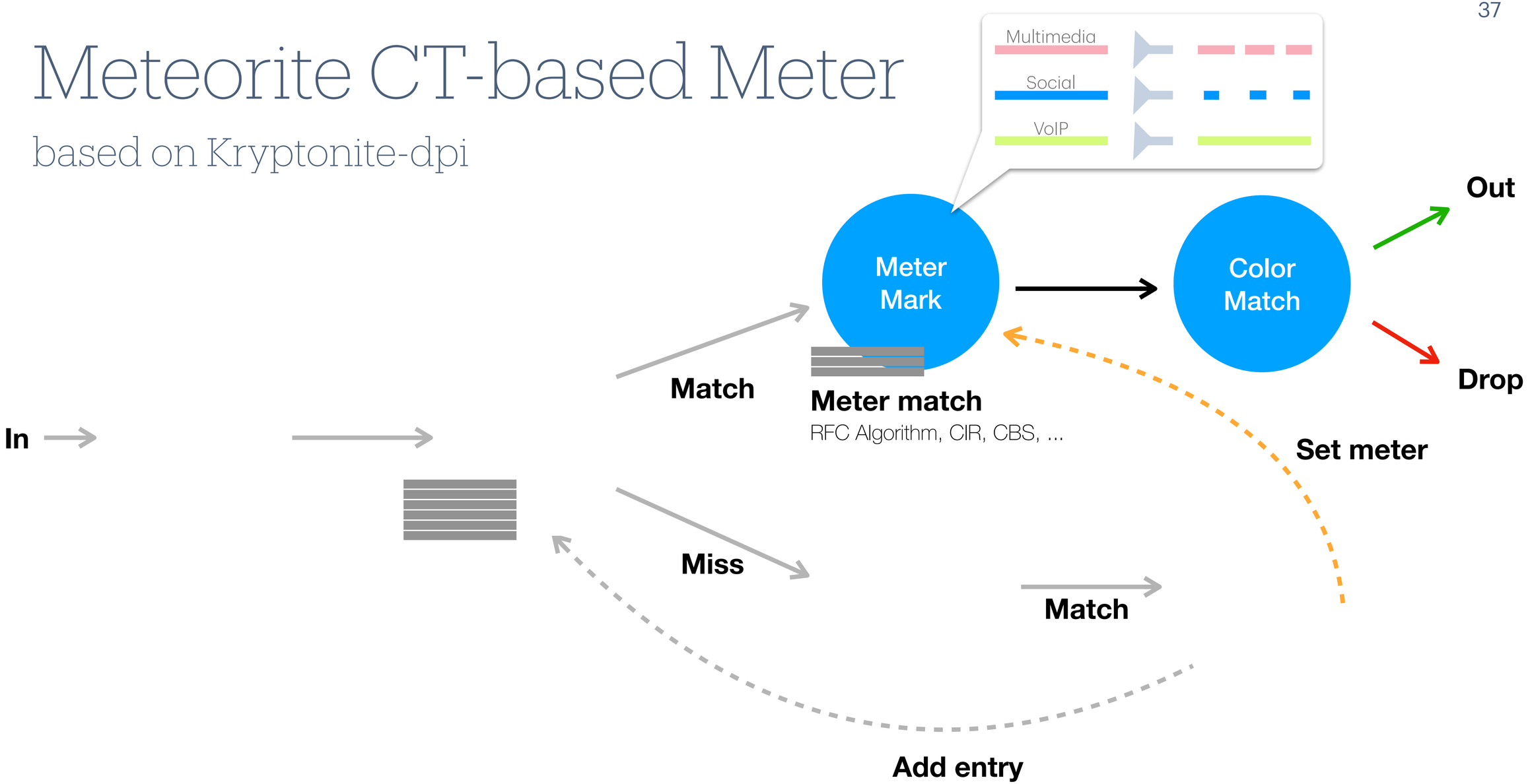
- Simple (Non-shared) meter
  - A dedicated meter per flow
  - Each flow has its own independent rate limit
  - Traffic is evaluated individually per flow



- Shared meter
  - A single meter instance that can be attached to multiple flows
  - All flows that use this meter share the same rate limit (queue/class)
  - Traffic from all these flows is aggregated together when evaluating the rate

Meteorite CT-based Meter

based on Kryptonite-dpi



Thank you!