Network Visibility and Traffic Control with nProbe and nDPI

Luca Deri

deri@ntop.org
@lucaderi

ntop

PacketFest'25

Agenda

- Introduction to flow-based analysis
- Flow collection: NetFlow/IPFIX, sFlow
- "Converting" packets into flows
- nDPI in flow analysis

Part I: Background

Traffic Analysis Simplified [1/2]

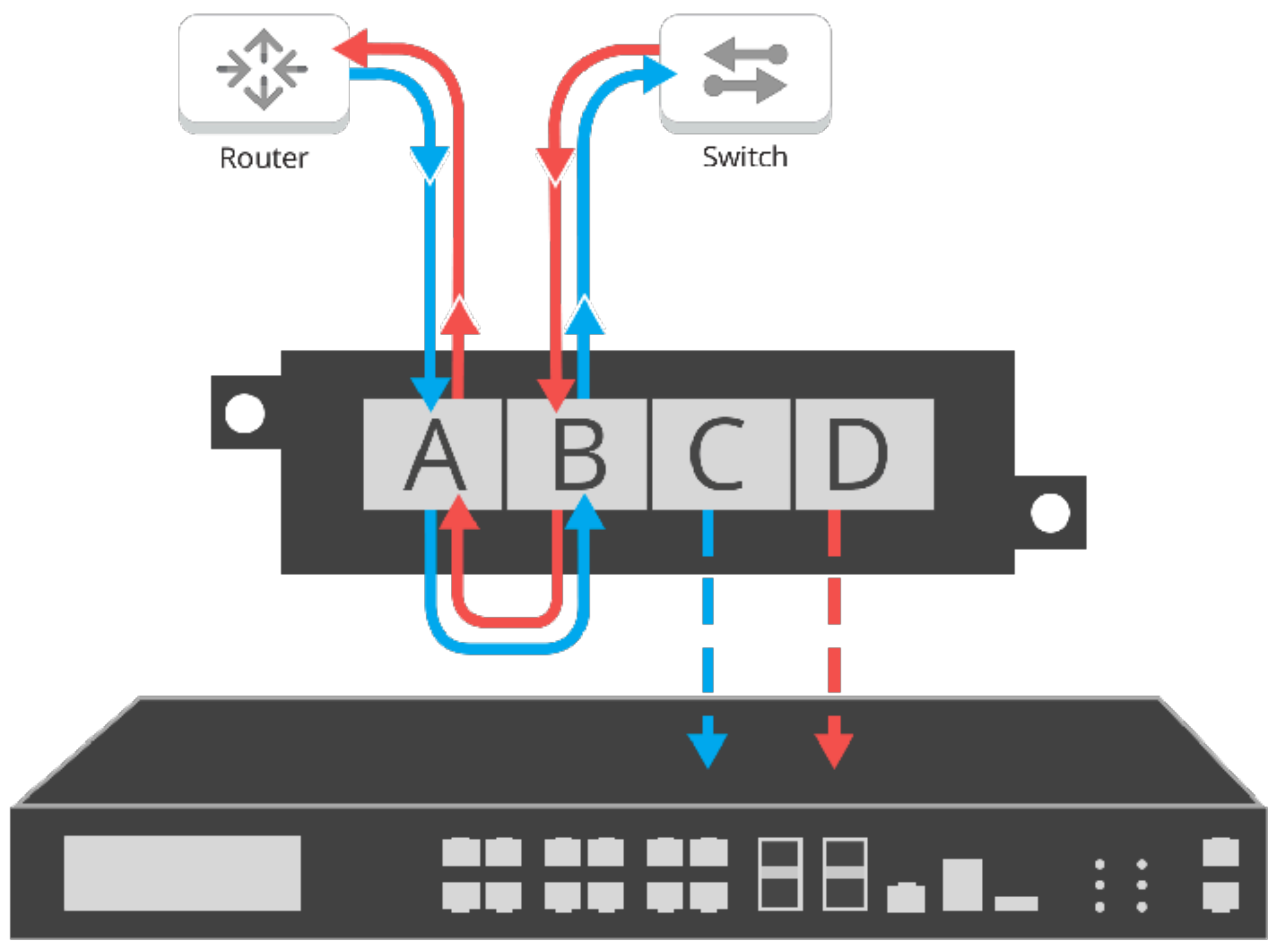
- In traditional packet-based analysis traffic is performed analyzing packets sequentially:
  - Capture traffic
  - Decode packets
  - Classify them into connections
  - Account traffic metrics (packets, bytes, etc), analyze payload …

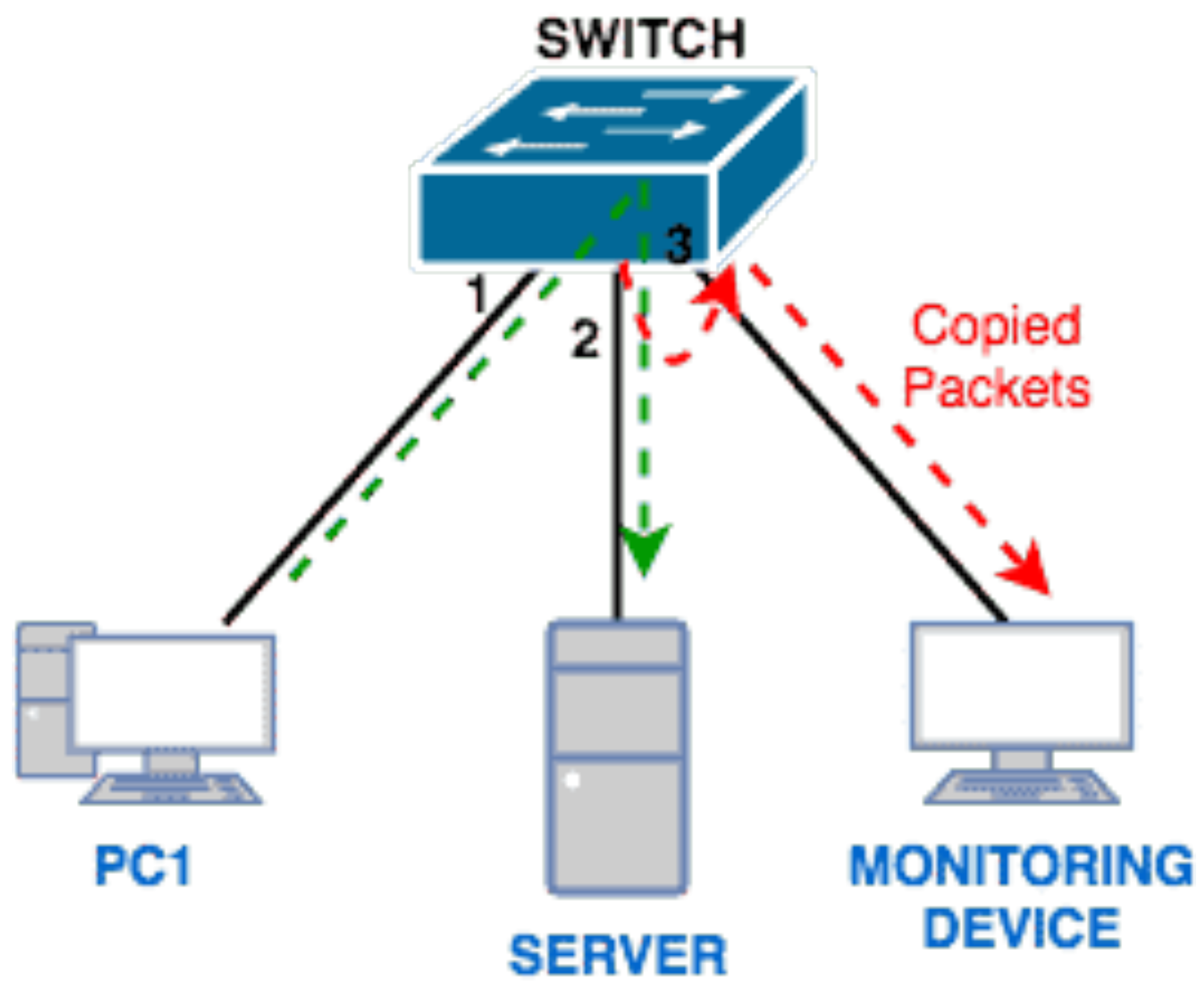
```
deri@iMacM1.local 201> netstat -na
Active Internet connections (including servers)
Proto Recv-Q Send-Q  Local Address          Foreign Address        (state)
tcp4       0      0  192.168.1.29.49685     35.186.224.28.443      ESTABLISHED
tcp4       0      0  192.168.1.29.49684     149.154.167.91.443     ESTABLISHED
tcp4       0      0  192.168.1.29.49683     162.125.69.19.443      ESTABLISHED
tcp4       0      0  192.168.1.29.49679     162.125.21.2.443       ESTABLISHED
tcp4       0      0  192.168.1.29.49675     162.125.21.3.443       ESTABLISHED
tcp4       0      0  192.168.1.29.49669     162.159.130.234.443    ESTABLISHED
tcp4       0      0  192.168.1.29.49666     35.186.224.24.443      ESTABLISHED
tcp4       0      0  192.168.1.29.49638     192.168.1.56.1400      CLOSE_WAIT
tcp4       0      0  192.168.1.29.49637     192.168.1.56.1400      CLOSE_WAIT
tcp4       0      0  192.168.1.29.49635     192.168.1.56.1400      CLOSE_WAIT
tcp4       0      0  192.168.1.29.49634     192.168.1.56.1400      CLOSE_WAIT
tcp4       0      0  192.168.1.29.49636     192.168.1.56.1400      CLOSE_WAIT
tcp4       0      0  192.168.1.29.49631     192.168.1.56.1400      CLOSE_WAIT
tcp4       0      0  192.168.1.29.49633     192.168.1.56.1400      CLOSE_WAIT
tcp4       0      0  192.168.1.29.49632     192.168.1.56.1400      CLOSE_WAIT
tcp4       0      0  192.168.1.29.49533     149.154.167.91.443     ESTABLISHED
tcp4       0      0  192.168.1.29.49491     167.99.215.164.993     ESTABLISHED
tcp4       0      0  *.3400                 *.*                    LISTEN
tcp4       0      0  127.0.0.1.17603        *.*                    LISTEN
tcp4       0      0  127.0.0.1.17600        *.*                    LISTEN
tcp4       0      0  *.17500                *.*                    LISTEN
tcp6       0      0  *.17500                *.*                    LISTEN
tcp6       0      0  ::1.6379               *.*                    LISTEN
tcp4       0      0  127.0.0.1.6379         *.*                    LISTEN
tcp4       0      0  192.168.1.29.49222     35.186.224.44.443      ESTABLISHED
tcp4       0      0  192.168.1.29.49199     34.158.1.133.4070      ESTABLISHED
```

Traffic Analysis Simplified [2/2]

- In order to analyze traffic you need packets. Some options:
  - Your monitoring tools sit on the host where traffic flows (e.g. a gateway).
  - You need to "divert" traffic towards your "observation point"
- Possible alternative:
  - Do not look at packets but ask the network kernel.

Traffic Diversion Technologies



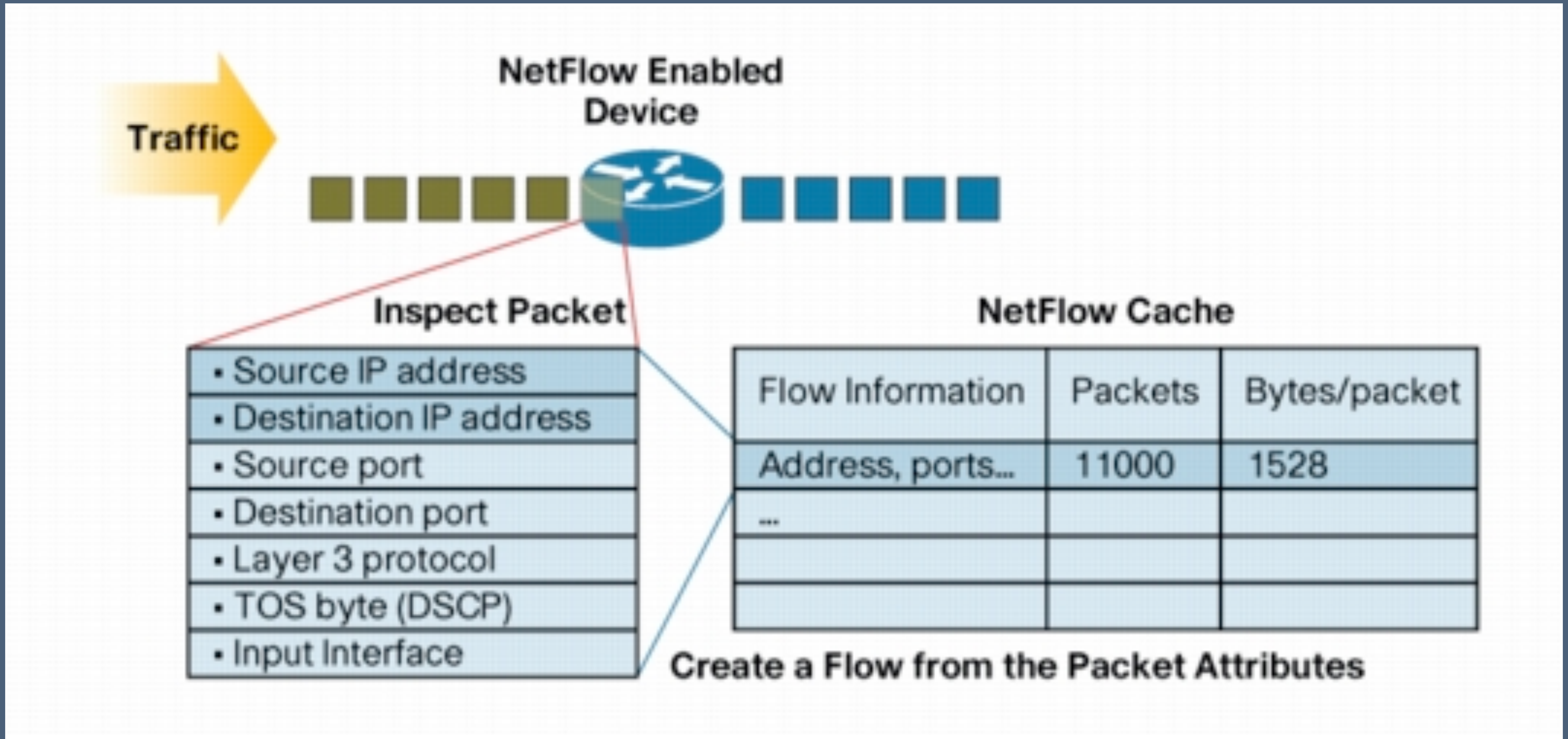


In Kernel Traffic Analysis

```
deri@ubuntu24 224> sudo ./ebpflow
Successfully started! Please run `sudo cat /sys/kernel/debug/tracing/trace_pipe` to see output of the BPF programs.
Ready...
1746262795.493045 (2732979104571889)[RECV][lo][Sent][IPv4/UDP][pid/tid: 767448/767447 [/usr/bin/curl www.ntop.org], uid/gid: 1000/1000][father pid/tid: 767432/0 [/usr/bin/tcsh], uid/gid: 1000/1000][addr: 127.0.0.1:52080 <-> 127.0.0.53:53]
1746262795.493248 (2732979104882657)[RECV][enp1s0][Sent][IPv4/UDP][pid/tid: 619/619 [/usr/lib/systemd/systemd-resolved], uid/gid: 992/992][father pid/tid: 1/0 [/usr/lib/systemd/systemd], uid/gid: 0/0][addr: 192.168.122.62:40403 <-> 192.168.122.1:53]
1746262795.493393 (2732979105047030)[RECV][enp1s0][Sent][IPv4/UDP][pid/tid: 619/619 [/usr/lib/systemd/systemd-resolved], uid/gid: 992/992][father pid/tid: 1/0 [/usr/lib/systemd/systemd], uid/gid: 0/0][addr: 192.168.122.62:44199 <-> 192.168.122.1:53]
1746262795.551084 (2732979162704172)[RECV][lo][Sent][IPv4/UDP][pid/tid: 619/619 [/usr/lib/systemd/systemd-resolved], uid/gid: 992/992][father pid/tid: 1/0 [/usr/lib/systemd/systemd], uid/gid: 0/0][addr: 127.0.0.53:53 <-> 127.0.0.1:52080]
```

```
deri@ubuntu24 240> ss -it
State       Recv-Q       Send-Q                      Local Address:Port                       Peer Address:Port
Process
ESTAB       0            0                     [::ffff:192.168.122.62]:ssh                  [::ffff:192.168.122.1]:60708
         cubic wscale:7,7 rto:201 rtt:0.719/0.776 ato:40 mss:1448 pmtu:1500 rcvmss:1448 advmss:1448 cwnd:10 bytes_sent:5630 bytes_
acked:5630 bytes_received:4254 segs_out:49 segs_in:63 data_segs_out:44 data_segs_in:30 send 161Mbps lastsnd:166682 lastrcv:166971
lastack:166682 pacing_rate 322Mbps delivery_rate 209Mbps delivered:45 app_limited busy:110ms rcv_space:14600 rcv_ssthresh:64076 mi
nrtt:0.142 snd_wnd:75776
ESTAB       0            0                     [::ffff:192.168.122.62]:ssh                  [::ffff:192.168.122.1]:42828
         cubic wscale:7,7 rto:201 rtt:0.265/0.039 ato:40 mss:1448 pmtu:1500 rcvmss:1448 advmss:1448 cwnd:10 bytes_sent:107198 byte
s_acked:107198 bytes_received:28774 segs_out:742 segs_in:1124 data_segs_out:580 data_segs_in:551 send 437Mbps lastsnd:7 lastrcv:8
lastack:6 pacing_rate 871Mbps delivery_rate 145Mbps delivered:581 app_limited busy:330ms rcv_space:14600 rcv_ssthresh:64076 minrtt
:0.101 snd_wnd:81920
```

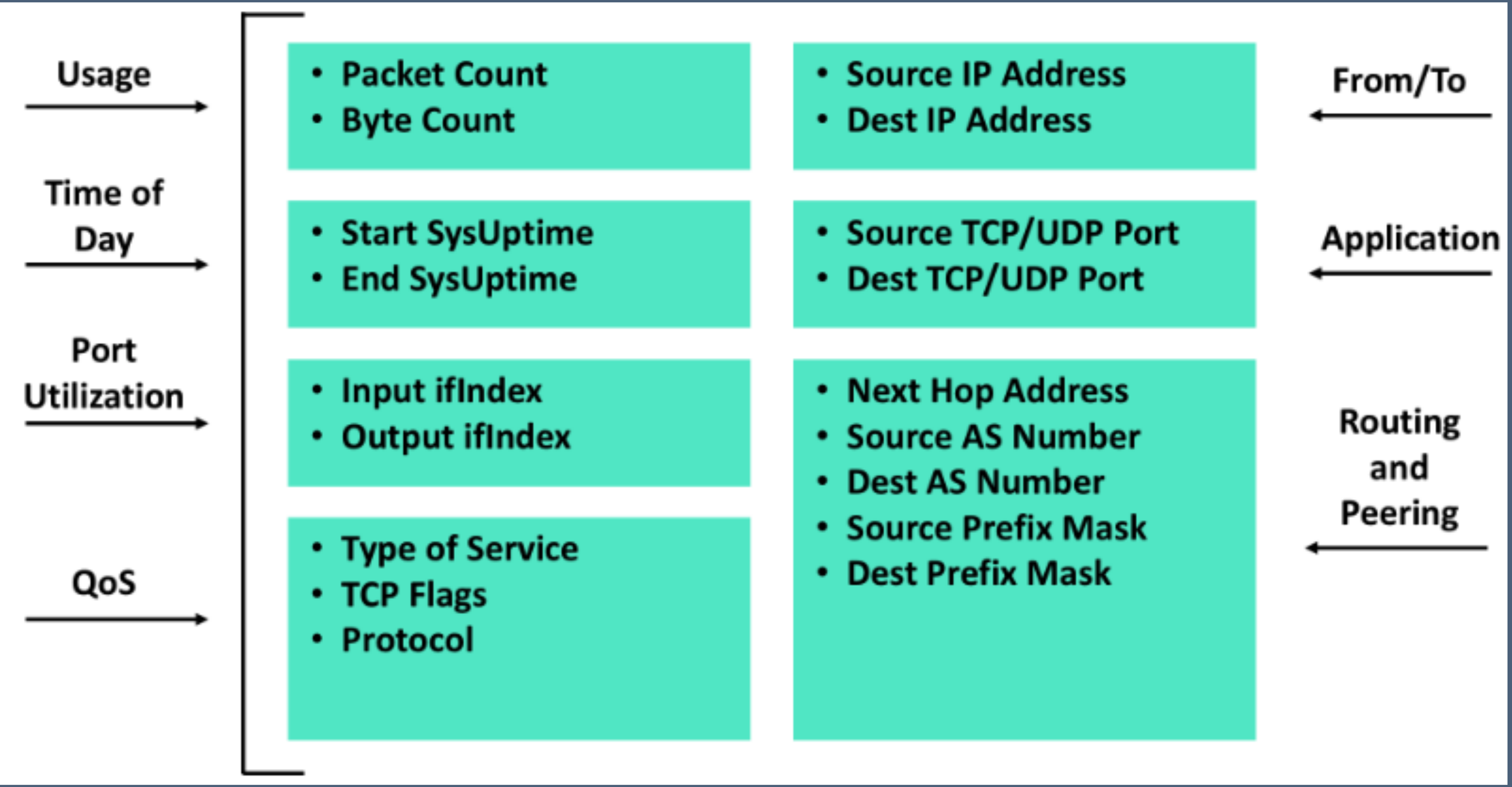
Introduction to Flow-based Analysis: NetFlow/IPFIX [1/3]



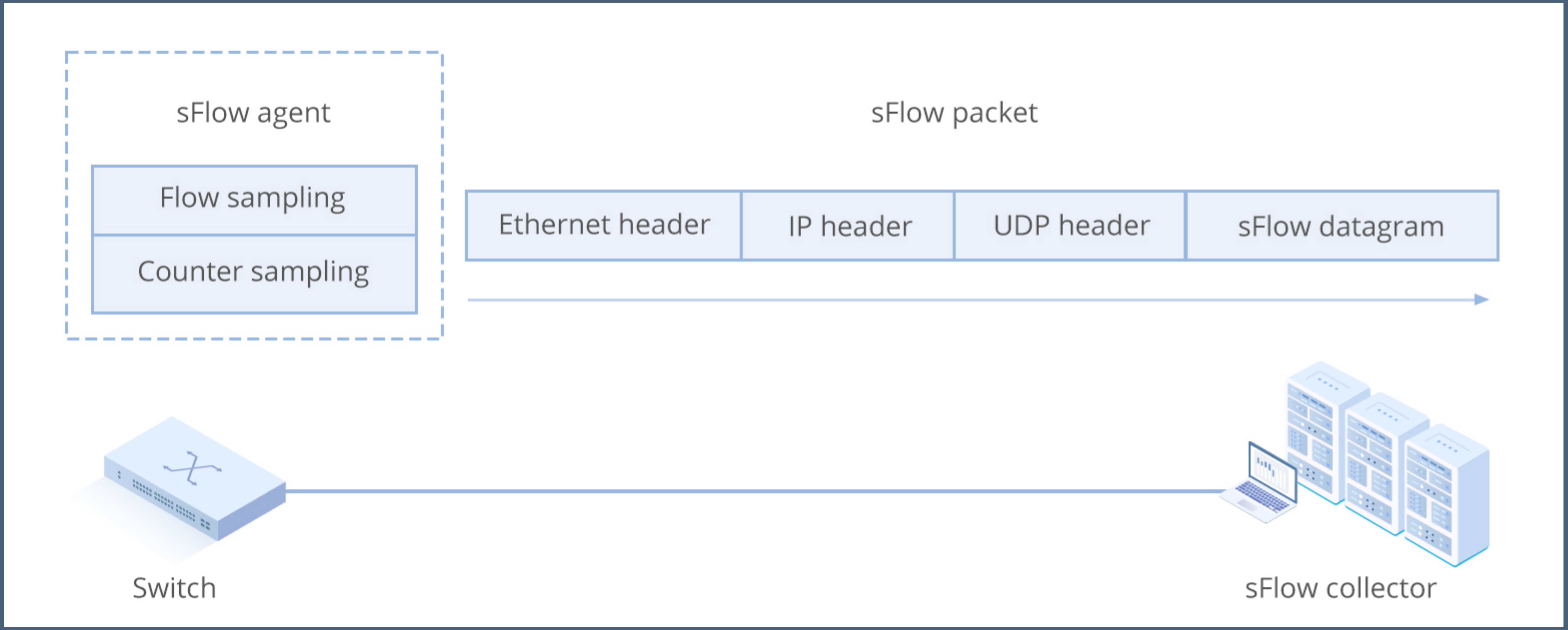
Introduction to Flow-based Analysis: NetFlow/IPFIX [2/3]

- Flows are terminated when one of these conditions are met:
  - The network communication has ended (e.g. a packet contains the TCP FIN flag).
  - The flow lasted too long (default 30 min).
  - The flow has been not active (i.e. no new packets have been received) for too long (default 15 sec).

Introduction to Flow-based Analysis: NetFlow/IPFIX [3/3]



Introduction to Flow-based Analysis: sFlow [1/2]



Introduction to Flow-based Analysis: sFlow [2/2]

- Flow Sample:
  sampled packet capture
  cut to a snaplen

- Counter Sample
  SNMP interface counters
  sent in push-mode

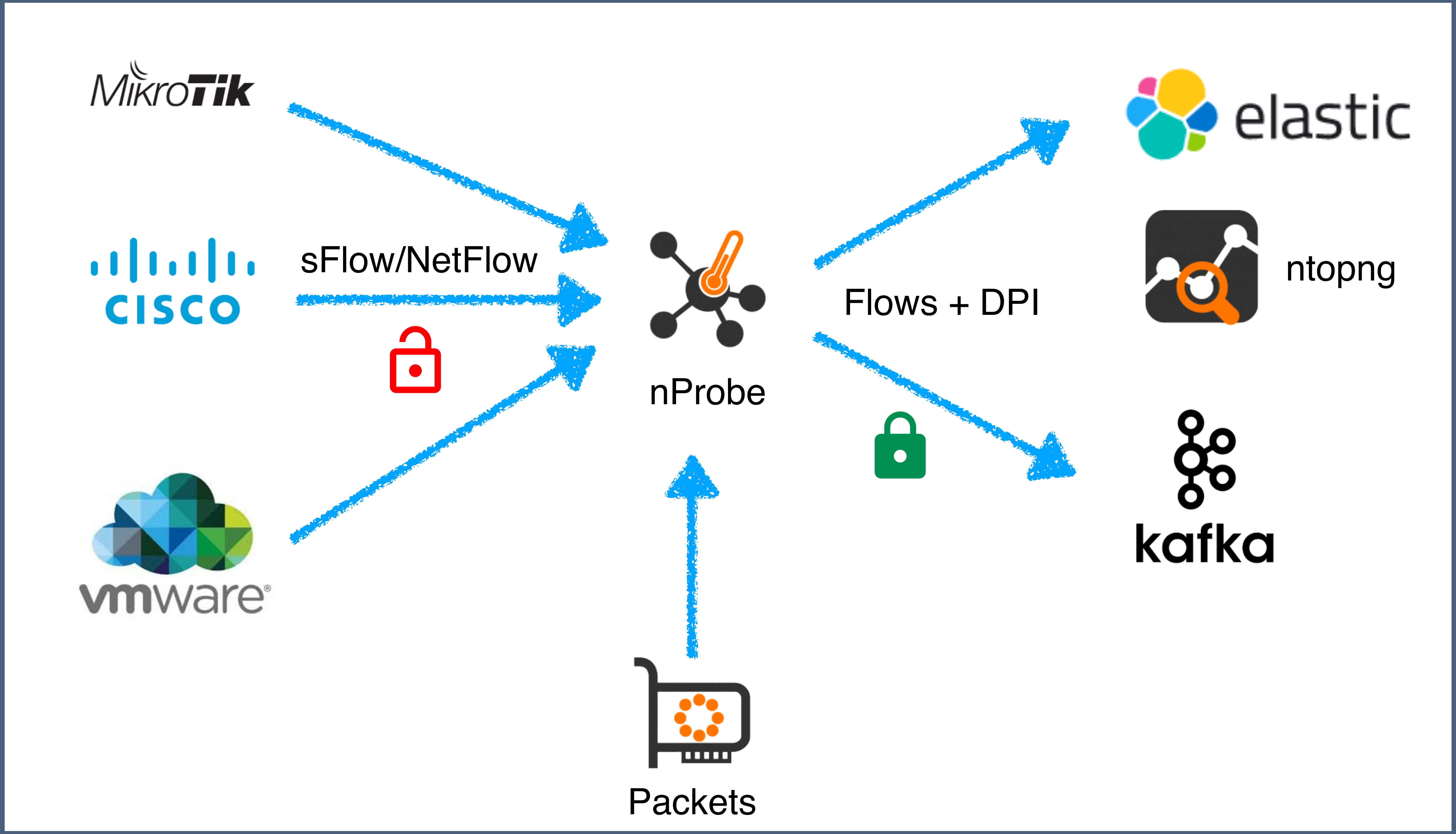
In essence this is a
sampled RSPAN-like
packet capture

```
SysUptime: 2 days, 6 minutes, 37 seconds (173197112ms)
NumSamples: 4
Flow sample, seq 9440
  0000 0000 0000 0000 0000 .... .... .... = Enterprise: standard sFlow (0)
  .... .... .... .... .... 0000 0000 0001 = sFlow sample type: Flow sample (1)
  Sample length (byte): 280
  Sequence number: 9440
  0000 0000 .... .... .... .... .... .... = Source ID class: 0
  .... .... 0000 0000 0000 0000 0000 0010 = Index: 2
  Sampling rate: 1 out of 1024 packets
  Sample pool: 9666560 total packets
  Dropped packets: 0
  Input interface (ifIndex): 2
> Output interface: 0x00000030
  Flow record: 4
> Extended switch data
  Raw packet header
    0000 0000 0000 0000 0000 .... .... .... = Enterprise: standard sFlow (0)
    Format: Raw packet header (1)
    Flow data length (byte): 144
    Header protocol: Ethernet (1)
    Frame Length: 1522
    Payload stripped: 4
    Sampled header length: 128
    Header of sampled packet […]: d4aff786f4a79ce17662a0e08100000b0800450005dc178f400035069108334bca52b9d4e0
    > Ethernet II, Src: Cisco_62:a0:e0 (9c:e1:76:62:a0:e0), Dst: AristaNetwor_86:f4:a7 (d4:af:f7:86:f4:a7)
    > 802.1Q Virtual LAN, PRI: 0, DEI: 0, ID: 11
    > Internet Protocol Version 4, Src: 51.75.202.82, Dst: 185.212.224.18
    > Transmission Control Protocol, Src Port: 58166, Dst Port: 10791, Seq: 2378071091, Ack: 1983703187
    > Data (58 bytes)
> Extended gateway data
> Extended router data
> Flow sample, seq 9441
```

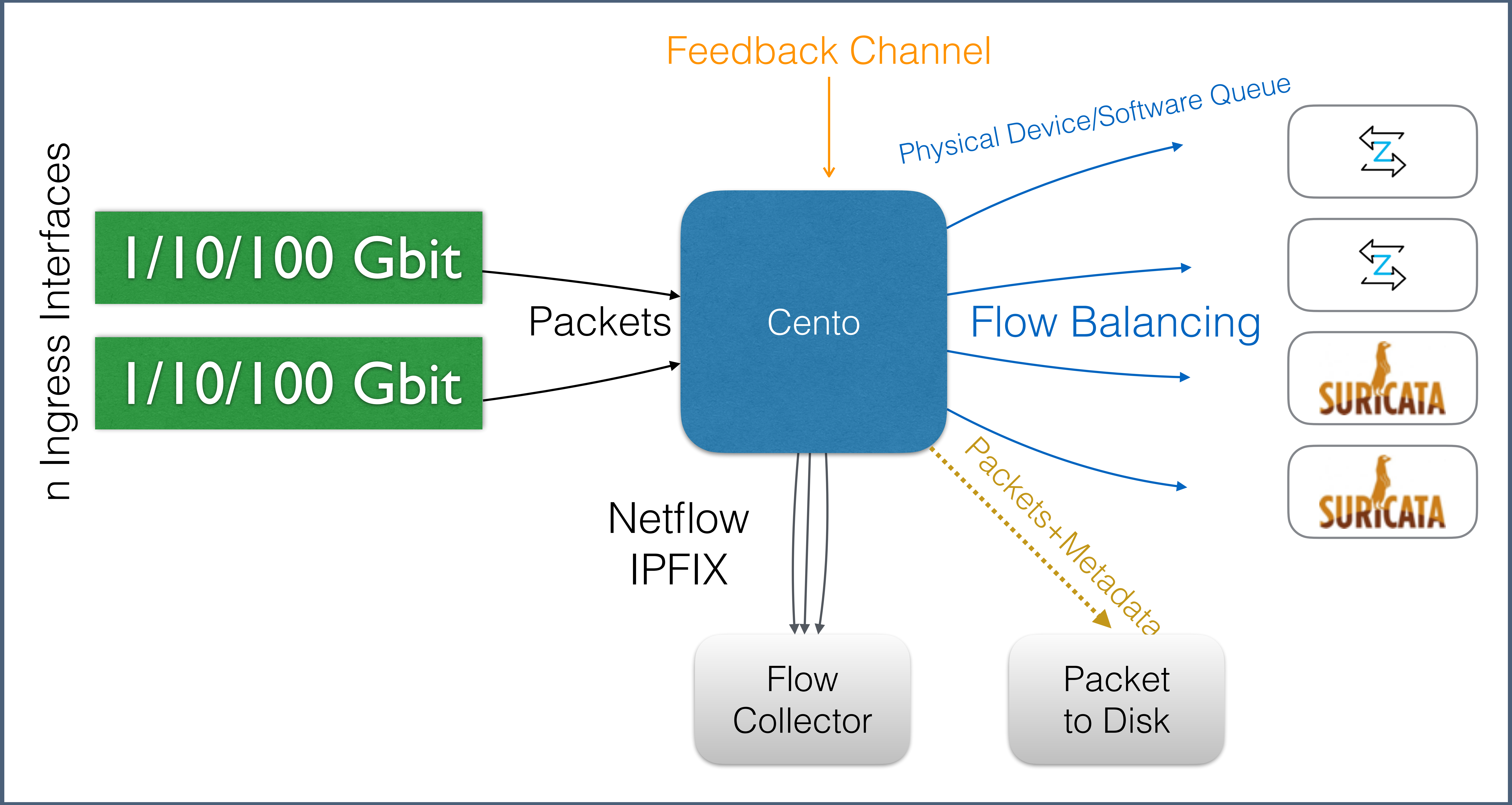
Flow-based Analysis: Evaluation

- Advantages
  - Great idea for "compressing" monitored data by reducing its volume.
  - Contrary to packet-based analysis, most processing happens inside the monitoring device (e.g. a router): less CPU used at the collector side.
- Disadvantages
  - Visibility is limited to what the device is exporting.
  - Monitoring devices often cannot keep-up with network speed (sampling).

nProbe "Classic"



nProbe vs nProbe Cento



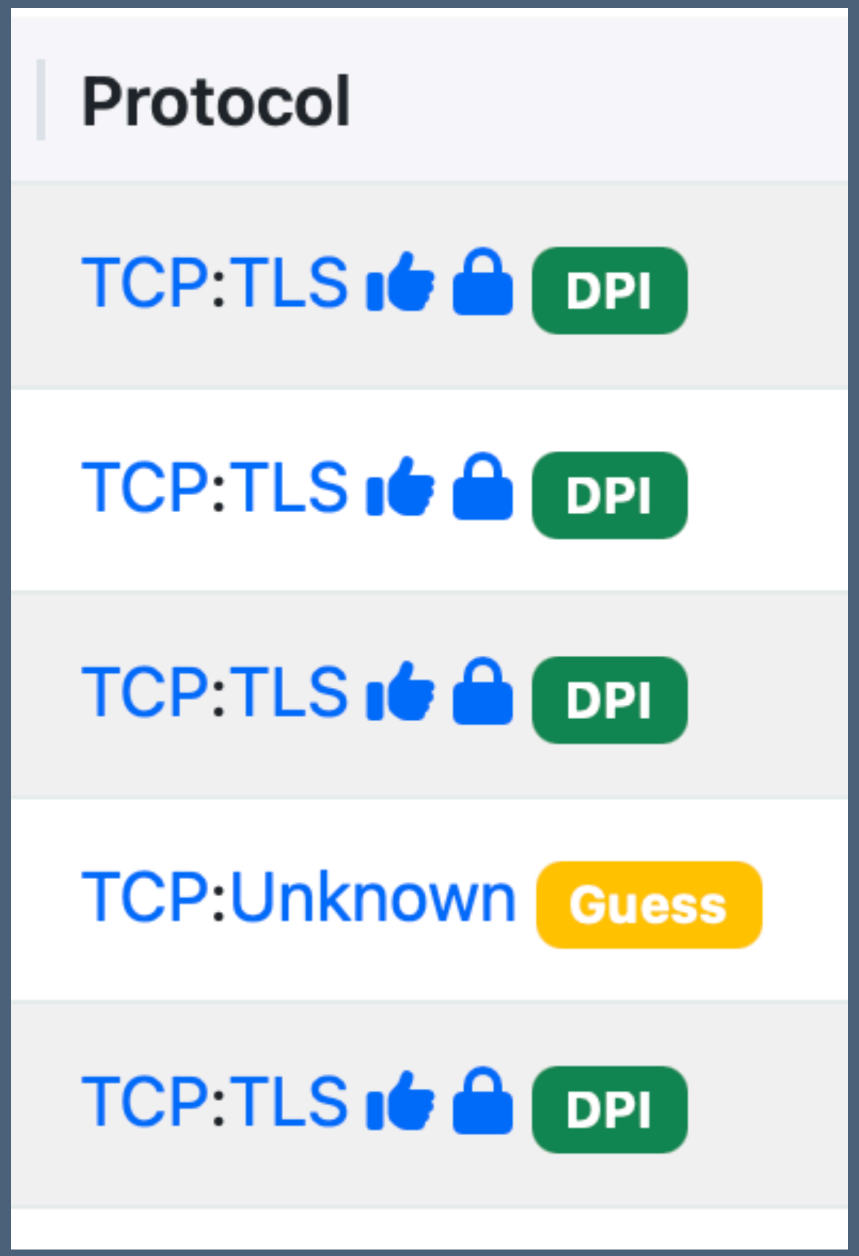
Caveat: no flow collection

nDPI in Flow Analysis

- nDPI, ntop's open source toolkit for DPI, implements Deep Packet Inspection, meaning that the packet payload is inspected. In other words: no packet, no DPI !
  - NetFlow collection: some collection devices export DPI protocol information. However there is not DPI standard (as with flows) hence it's not simple to compare data and protocols.
  - sFlow collection: as packets are sampled and cut to a snaplen, DPI is unlikely to work effectively (beside UDP exceptions such as DNS).

nDPI Without Packets

- Even if the payload cannot be observed, nDPI can partially work using:
  - IP address: nDPI contains the IP addresses of major CDNs (Content Delivery Networks) and services (e.g. WhatsApp, Office365), a well egress nodes for popular VPNs (e.g. Proton) or Tor.
  - Port: old-style protocol/port association.




```
NDPI_CONFIDENCE_UNKNOWN              = 0,    /* Unknown classification */
NDPI_CONFIDENCE_MATCH_BY_PORT,               /* Classification obtained looking only at the L4 ports */
NDPI_CONFIDENCE_NBPF,                        /* PF_RING nBPF (custom protocol) */
NDPI_CONFIDENCE_DPI_PARTIAL,                 /* Classification results based on partial/incomplete DPI information */
NDPI_CONFIDENCE_DPI_PARTIAL_CACHE,           /* Classification results based on some LRU cache with partial/incomplete DPI information */
NDPI_CONFIDENCE_DPI_CACHE,                   /* Classification results based on some LRU cache (i.e. correlation among sessions) */
NDPI_CONFIDENCE_DPI,                         /* Deep packet inspection */
NDPI_CONFIDENCE_MATCH_BY_IP,                 /* Classification obtained looking only at the IP addresses */
NDPI_CONFIDENCE_DPI_AGGRESSIVE,              /* Aggressive DPI: it might be a false positive */
NDPI_CONFIDENCE_CUSTOM_RULE,                 /* Matching a custom rules */
```

nDPI in nProbe [1/2]

- If nProbe is used in flow collection mode, accuracy restrictions just discussed apply.
- Instead if nProbe is used as packet -> flow (packet mode) tool, DPI is performed accurately and reported to the flow collector.
- Caveat:
  - If NetFlow/IPFIX is used, limited DPI information can be reported (e.g. hostname/SNI or protocol Id).
  - If data is delivered via ZMQ/Kafka full DPI information can be reported.

nDPI in nProbe [2/2]

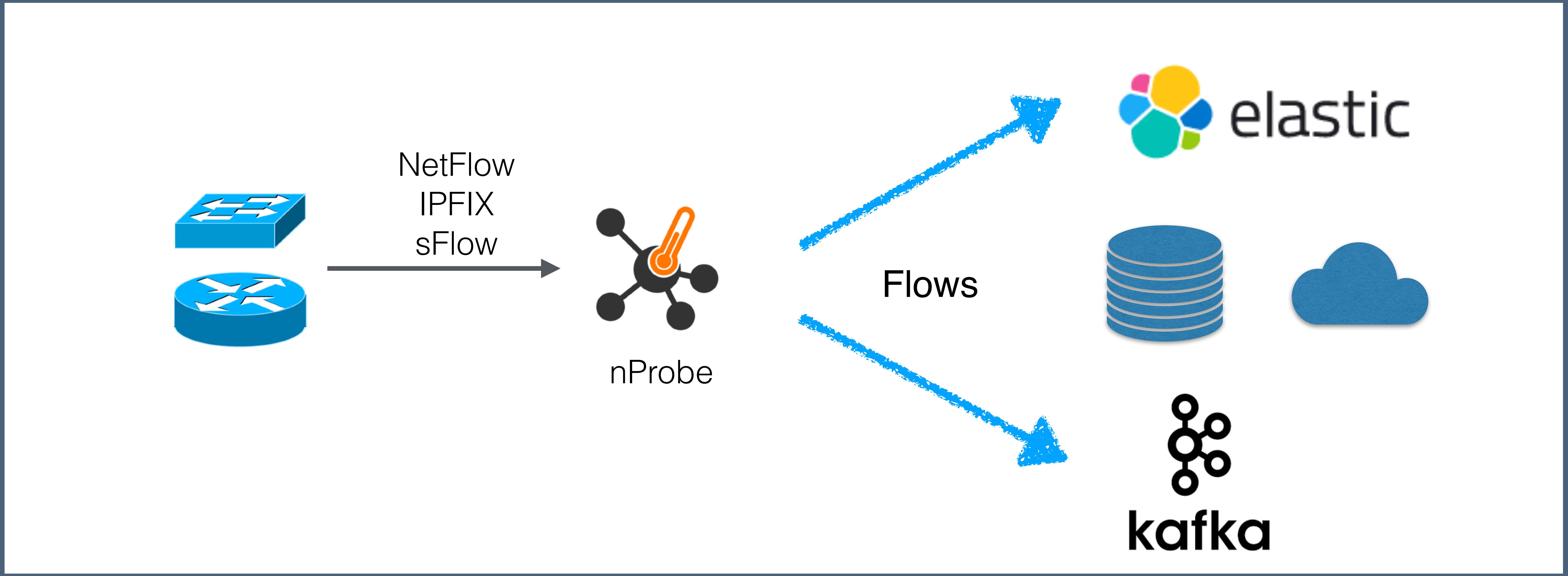
Example of "full" nDPI information:

{"src_ip":"192.168.1.117","dest_ip":"109.94.160.99","src_port":54871,"dst_port":443,"ip":4,"tcp_fing
erprint":"2_64_65535_15db81ff8b0d","proto":"TCP","ndpi": {"flow_risk": {"15": {"risk":"TLS
(probably) Not Carrying HTTPS","severity":"Low","risk_score":
{"total":460,"client":410,"server":50}}},"confidence":
{"6":"DPI"},"proto":"TLS.Zoom","proto_id":"91.189","proto_by_ip":"Unknown","proto_by_ip_id":0,"encry
pted":1,"breed":"Acceptable","category_id":26,"category":"Video","hostname":"zoomfrn99mmr.zoom.us","
domainame":"zoomfrn99mmr.zoom.us","tls":
{"version":"TLSv1.2","server_names":"*.zoom.us,zoom.us","ja3s":"ada793d0f02b028a6c840504edccb652","j
a4":"t12d930700_72a4e8475a2e_4446390ac224","unsafe_cipher":0,"cipher":"TLS_ECDHE_RSA_WITH_AES_128_GC
M_SHA256","issuerDN":"C=US, ST=Arizona, L=Scottsdale, O=GoDaddy.com, Inc., OU=http:\/\/
certs.godaddy.com\/repository\/, CN=Go Daddy Secure Certificate Authority -
G2","subjectDN":"OU=Domain Control Validated,
CN=*.zoom.us","fingerprint":"F7:5A:83:A8:77:24:55:D7:6D:2E:93:F6:6E:9C:C9:7E:AD:9B:3B:E8","blocks":0
}},"detection_completed":1,"check_extra_packets":0,"flow_id":29,"first_seen":1569520471.189,"last_se
en":1569520473.190,"duration":2.001,"vlan_id":0,"bidirectional":1,"xfer":
{"data_ratio":0.511,"data_ratio_str":"Upload","src2dst_packets":127,"src2dst_bytes":54118,"src2dst_g
oodput_bytes":45724,"dst2src_packets":83,"dst2src_bytes":17526,"dst2src_goodput_bytes":12028},"iat":
{"flow_min":1,"flow_avg":27.9,"flow_max":940,"flow_stddev":116.2,"c_to_s_min":0,"c_to_s_avg":16.9,"c
_to_s_max":950,"c_to_s_stddev":93.0,"s_to_c_min":0,"s_to_c_avg":9.2,"s_to_c_max":156,"s_to_c_stddev"
:23.6},"pktlen":
{"c_to_s_min":66,"c_to_s_avg":426.1,"c_to_s_max":1506,"c_to_s_stddev":458.2,"s_to_c_min":66,"s_to_c_
avg":211.2,"s_to_c_max":1506,"s_to_c_stddev":363.6},"tcp_flags":
{"cwr_count":0,"ece_count":0,"urg_count":0,"ack_count":209,"psh_count":108,"rst_count":0,"syn_count"
:2,"fin_count":0,"src2dst_cwr_count":0,"src2dst_ece_count":0,"src2dst_urg_count":0,"src2dst_ack_coun
t":126,"src2dst_psh_count":77,"src2dst_rst_count":0,"src2dst_syn_count":1,"src2dst_fin_count":0,"dst
2src_cwr_count":0,"dst2src_ece_count":0,"dst2src_urg_count":0,"dst2src_ack_count":83,"dst2src_psh_co
unt":31,"dst2src_rst_count":0,"dst2src_syn_count":1,"dst2src_fin_count":0},"c_to_s_init_win":78,"s_t
o_c_init_win":74}

Part II: Use Cases

nProbe Flow Collection [1/3]

- nProbe collects remote NetFlow/sFlow that are delivered to remote ends




```
$ nprobe -i none --collector-port 2055 --kafka <brokers>;<ftopic>;[<otopic>;<ack>;<comp>]
--elastic <format> --mysql=<host[@port]|unix socket>:<dbname>:<prefix>:<user>:<pw>
--clickhouse=<host[@port]>:<dbname>:<prefix>:<user>:<pw>
```

nProbe Flow Collection [2/3]

```
[--collector-port|-3] <port|dir>    | Collect NetFlow/IPFIX/sFlow packets on port <port>
                                    | or directory where AWS VPC flow logs are stored
                                    | You can optionally specify an IPv4 address to bind to.
                                    | Example: -3 6343 (sFlow/NetFlow/IPFIX only) [UDP]
                                    | Example: -3 /data/vpc_flow_logs/ (VPC logs only)
                                    | Example: -3 127.0.0.1:6343       [UDP]
                                    | Example: -3 tcp://127.0.0.1:6343 [TCP]
                                    | Example: -3 tls://127.0.0.1:6343 [TLS]
                                    | Example: -3 tcp://6343           [TCP]
                                    | Example: --collector-port 2055
                                    | NOTE: in collector mode flow cache is disabled. If you
                                    | want to enable it add a trailer 'c'. Example: -3 6343c
                                    | NetFlow/IPFIX/sFlow packets can also be received through
                                    | a ZMQ relay, in which case <port> is used to specify the
                                    | relay endpoint. An implementation of a ZMQ relay
                                    | comes packaged and is available as binary flowRelay.
                                    | Example: -3 zmq://127.0.0.1:5556
```

Note: you can send to the collector port both sFlow/NetFlow/IPFIX and nProbe will handle all of them seamlessly

nProbe Flow Collection [3/3]

- If you are on a cloud environment, often you don't have NetFlow or packets but proprietary log formats.
- nProbe support both Amazon AWS e Google VPC.
- Just use -3 <path> to point to the directory where such logs can be found. nProbe will consume them, convert into flows, and delete after processing.

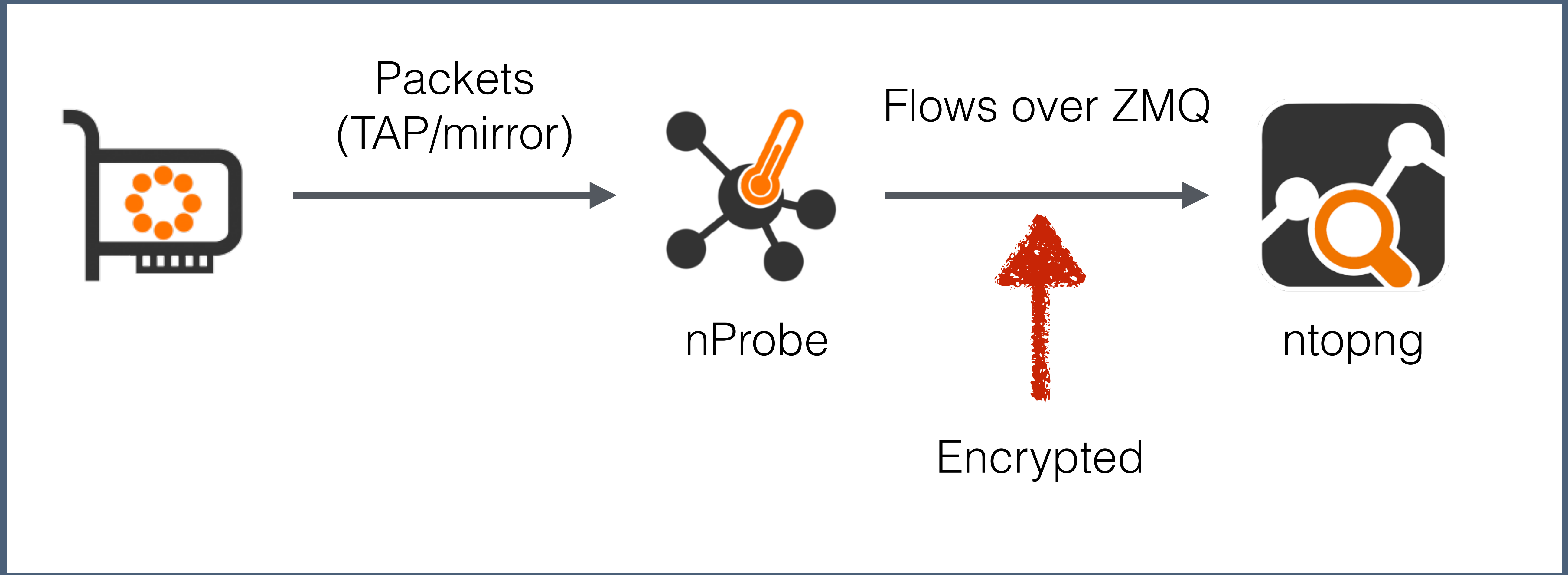

```
{
  "insertId": "sbg3kxf1o13cu",
  "jsonPayload": {
    "bytes_sent": "17792",
    "connection": {
      "dest_ip": "172.116.78.34",
      "dest_port": 58008,
      "protocol": 6,
      "src_ip": "10.128.0.2",
      "src_port": 22
    },
    "end_time": "2024-08-09T16:18:11.199096751Z",
    "packets_sent": "32",
    "reporter": "SRC",
    "rtt_msec": "779",
    "start_time": "2024-08-09T16:18:09.895020934Z"
  },
  "logName": "projects/myproject/logs/compute.googleapis.com%2Fvpc_flows",
  "receiveTimestamp": "2024-08-09T16:18:19.934521405Z",
  "resource": {
    "labels": {
      "location": "us-central1",
      "project_id": "myproject",
      "subnetwork_id": "7139065492556973974",
      "subnetwork_name": "default"
    },
    "type": "gce_subnetwork"
  },
  "timestamp": "2024-08-09T16:18:19.934521405Z"
}
```



```
version account-id interface-id srcaddr dstaddr srcport dstport protocol packets bytes start end action log-status
2 unknown eni-045a9fd7eb76dfbc6 10.253.17.103 10.253.17.27 54264 32261 6 5 407 1661447190 1661447244 ACCEPT OK
2 unknown eni-045a9fd7eb76dfbc6 10.253.17.12 10.253.17.103 32261 20807 6 5 561 1661447190 1661447244 ACCEPT OK
2 unknown eni-045a9fd7eb76dfbc6 10.253.17.103 10.253.17.12 20807 32261 6 5 407 1661447190 1661447244 ACCEPT OK
```

nProbe Probe Mode [1/3]

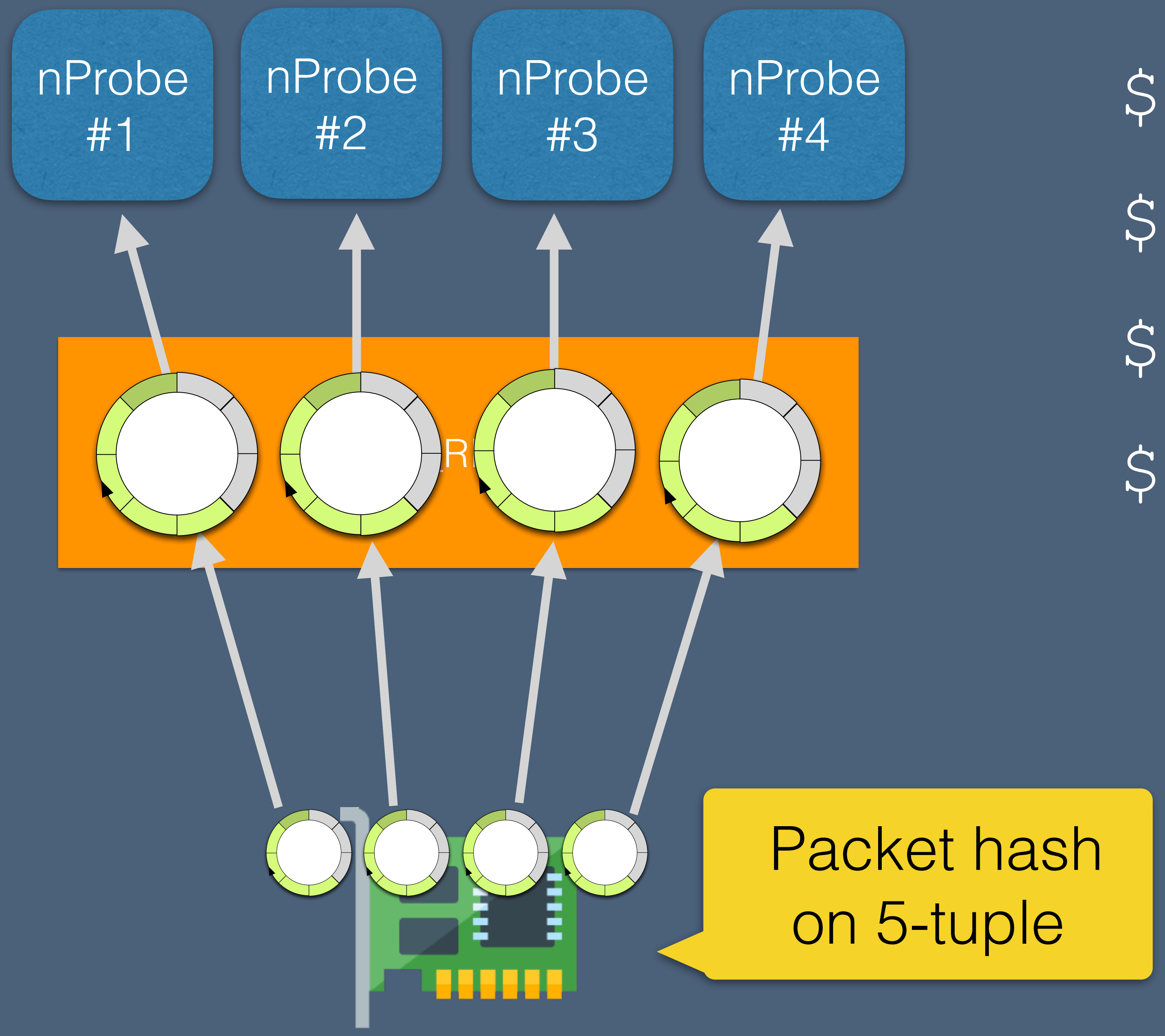
- In probe mode you can specify the packet capture interface with -i.




```
$ nprobe -i eth1 —zmq tcp://*:5556
$ ntopng -i tcp://127.0.0.1:5556
```

nProbe Probe Mode [2/3]

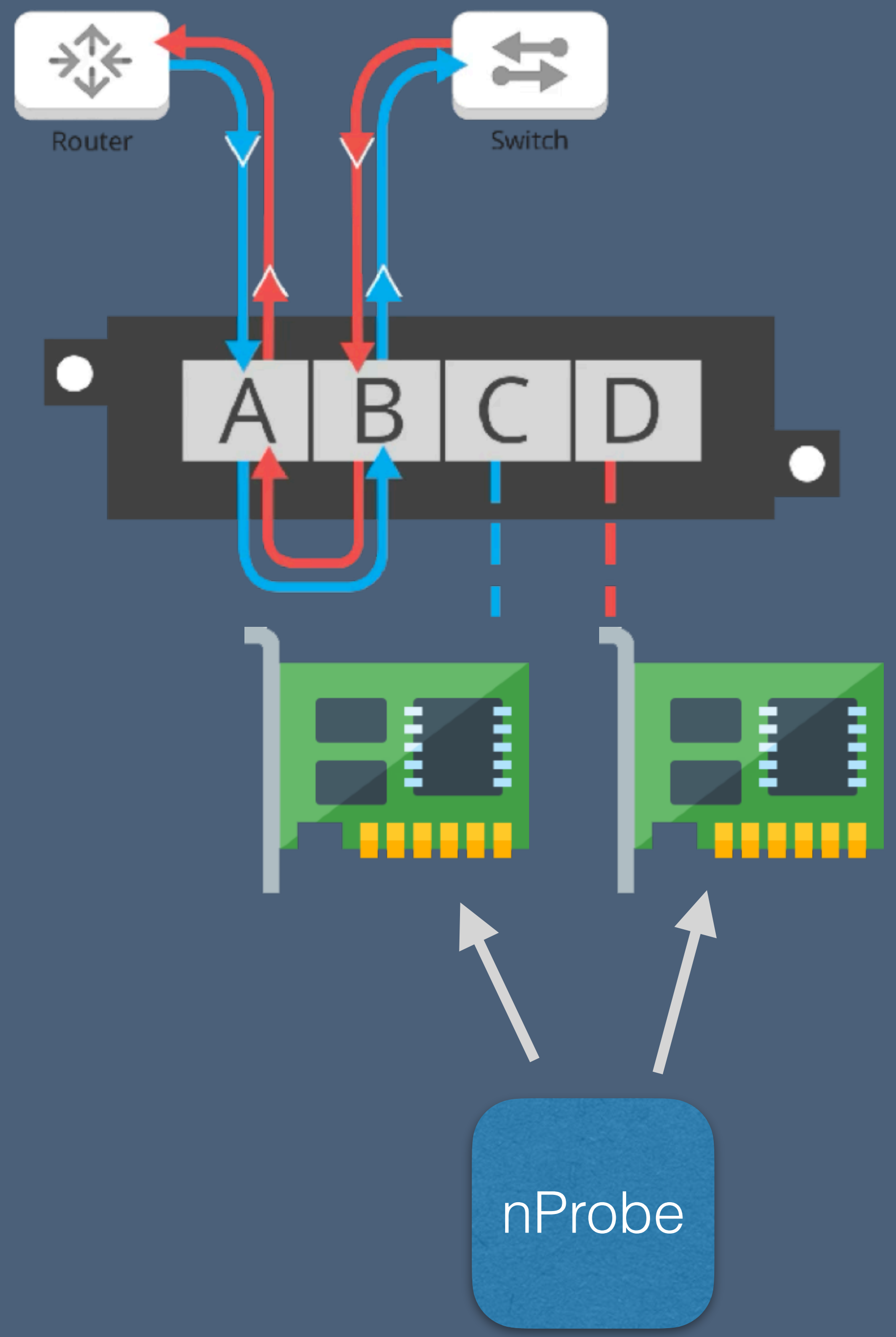
- RSS (Receive Side Scale) is a technique for balancing traffic across virtual network adapter queues. Great for scaling up.




```
$ nprobe -i eth1@0 —zmq tcp://*:5556 --cpu-affinity 1
$ nprobe -i eth1@1 —zmq tcp://*:5556 --cpu-affinity 3
$ nprobe -i eth1@2 —zmq tcp://*:5556 --cpu-affinity 5
$ nprobe -i eth1@3 —zmq tcp://*:5556 --cpu-affinity 7
```

nProbe Probe Mode [3/3]

- Network taps: RX and TX need to be reconciled in order DPI to work.



Single queue (over PF_RING)

```
$ nprobe -i eth1,eth2 —zmq tcp://*:5556 --cpu-affinity 1
```

Multiqueue (over PF_RING)

```
$ nprobe -i eth1@0,eth2@0 —zmq tcp://*:5556 --cpu-affinity 1
$ nprobe -i eth1@1,eth2@1 —zmq tcp://*:5556 --cpu-affinity 3
$ nprobe -i eth1@2,eth2@2 —zmq tcp://*:5556 --cpu-affinity 5
$ nprobe -i eth1@3,eth2@3 —zmq tcp://*:5556 --cpu-affinity 7
```

nDPI Data Analysis

ETA
(Encrypted Traffic Analysis)

Cybersecurity

TLS Certificate	Client Requested: 89.31.75.170 +	
TLS issuerDN	O=Webmin Webserver on finanziamentinews, CN=*	
JA4	t13i6412ht_0f757fa8abd0_1da015a32102	
TCP Flags and Connection State	Client ➡ Server: S A F P	Client ⬅ Server: S A F P
	Flow is active.	
Total Flow Score / Score Category Breakdown	460	Network / Cybersecurity
Issues		

Description	Score	Info	Mitre Att&ck	Remediation	Actions
TLS Cert Expired nDPI	100	07/Feb/2014 17:18:59 - 06/Feb/2019 17:18:59	T1078 Initial Access		
TLS Cert Self-signed nDPI	100	O=Webmin Webserver on finanziamentinews, CN=*	T1557 Credential Access		
Weak TLS Ciphers nDPI	100	Cipher TLS_RSA_WITH_3DES_EDE_CBC_SHA	T1573 Command and Control		
Blacklisted Client Contact ntopng	50	Blacklisted Client [Blacklist: "Stratosphere Lab"]			
Known Proto on Non Std Port nDPI	50	Expected on port 443	T1571 Command and Control		
TLS Uncommon ALPN nDPI	50	h	T1018 Discovery		
HTTP/TLS/QUIC Numeric Hostname/SNI nDPI	10	89.31.75.170	T1070 Defense Evasion		

Fingerprinting

nDPI Flow Risks

- HTTP suspicious user-agent
- HTTP numeric IP host contacted
- HTTP suspicious URL
- HTTP suspicious protocol header
- TLS connections not carrying HTTPS (e.g. a VPN over TLS)
- Suspicious DGA domain contacted
- Malformed packet
- SSH/SMB obsolete protocol or application version
- TLS suspicious ESNI usage
- Unsafe Protocol used
- Suspicious DNS traffic
- TLS with no SNI
- XSS (Cross Site Scripting)
- SQL Injection
- Arbitrary Code Injection/Execution
- Binary/.exe application transfer (e.g. in HTTP)
- Known protocol on non standard port
- TLS self-signed certificate
- TLS obsolete version
- TLS weak cipher
- TLS certificate expired
- TLS certificate mismatch
- DNS suspicious traffic
- HTTP suspicious content
- Risky ASN
- Risky Domain Name
- Malicious JA3 Fingerprint
- Malicious SHA1 Certificate
- Desktop of File Sharing Session
- TLS Uncommon ALPN
- TLS Certificate Validity Too Long
- Suspicious TLS Extension
- TLS Fatal Alert
- Suspicious Protocol traffic Entropy
- Clear-text Credentials Exchanged
- DNS Large Packet
- DNS Fragmented Traffic
- Invalid Characters Detected
- Possible Exploit Detected
- TLS Certificate Close to Expire
- Punycode/IDN Domain
- Error Code Detected
- Crawler/Bot Detected
- Anonymous Subscriber
- Unidirectional Traffic
- HTTP Obsolete Server
- ALPN/SNI Mismatch
- Client Contacted A Malware Host
- Binary File/Data Transfer (Attempt)
- Probing Attempt
- Obfuscated Traffic

Legenda: Clear Text Only, Encrypted/Plain Text, Encrypted Only

nDPI Fingerprinting [1/2]

- Fingerprinting is a technique for labelling data regardless of its format (plain text or encrypted).
- nDPI supports various fingerprinting methods:
  - TCP and DHCP are used to identify the operating system.
  - TLS/QUIC (JA3/JA4) and Web Browser Fingerprint
  - SSH,OpenVPNs (and dialects)
  - Obfuscated TLS (encrypted tunnels based on a TLS dialect)
  - Fully Encrypted Protocols (ShadowSocks, VMess, Trojan,...)

Router/AccessPoint MAC Address	TechnicolorD_60:ED:80
Host MAC Address	Apple _A7:EE:CC
IP Address	192.168.1.29 [192.168.1.0/24]
OS	macOS
Name	imacm1

Flow Peers [Client / Server]	imacm1 L P :60381 [9C:58:3C:A7:EE:CC] 140.82.114.25 R :443 [GitHub, Inc.]
Protocol / Application	TCP / TLS.Github (Collaborative) [Network: Github] [Confidence: DPI] TCP Fingerprint: 2_64_65535_d29295416479 [TLSv1.3]

nDPI Fingerprinting [2/2]

- Browser fingerprinting
  Collects information about a web browser and device where it's running on including browser type, version, operating system, screen resolution, installed plugins. This creates a unique "fingerprint" that can be used to track the user across different sessions and websites.

- Policy Enforcement (OS/Device Fencing)
  Restrict to specific VLANs/block old/specific devices/OSs by looking at the device MAC address or initial DHCP request. This technique plays an important role in securing OT (Operational Technology) networks.

- Hidden Device Detection
  Spot NAT devices or hotspots

Final Remarks

- Flows are a smart way to preprocess packets and create a distributed observability platform.
- nDPI adds contextual information that is useful in traffic analysis:
  - Host characterization (OS and application usage, e.g. browser)
  - Cybersecurity (detect communication risks, including ETA)
- Depending how data is collected, full/limited visibility is permitted.

Thank you!

https://forms.gle/2jGgoRSgGuUj1U1E7