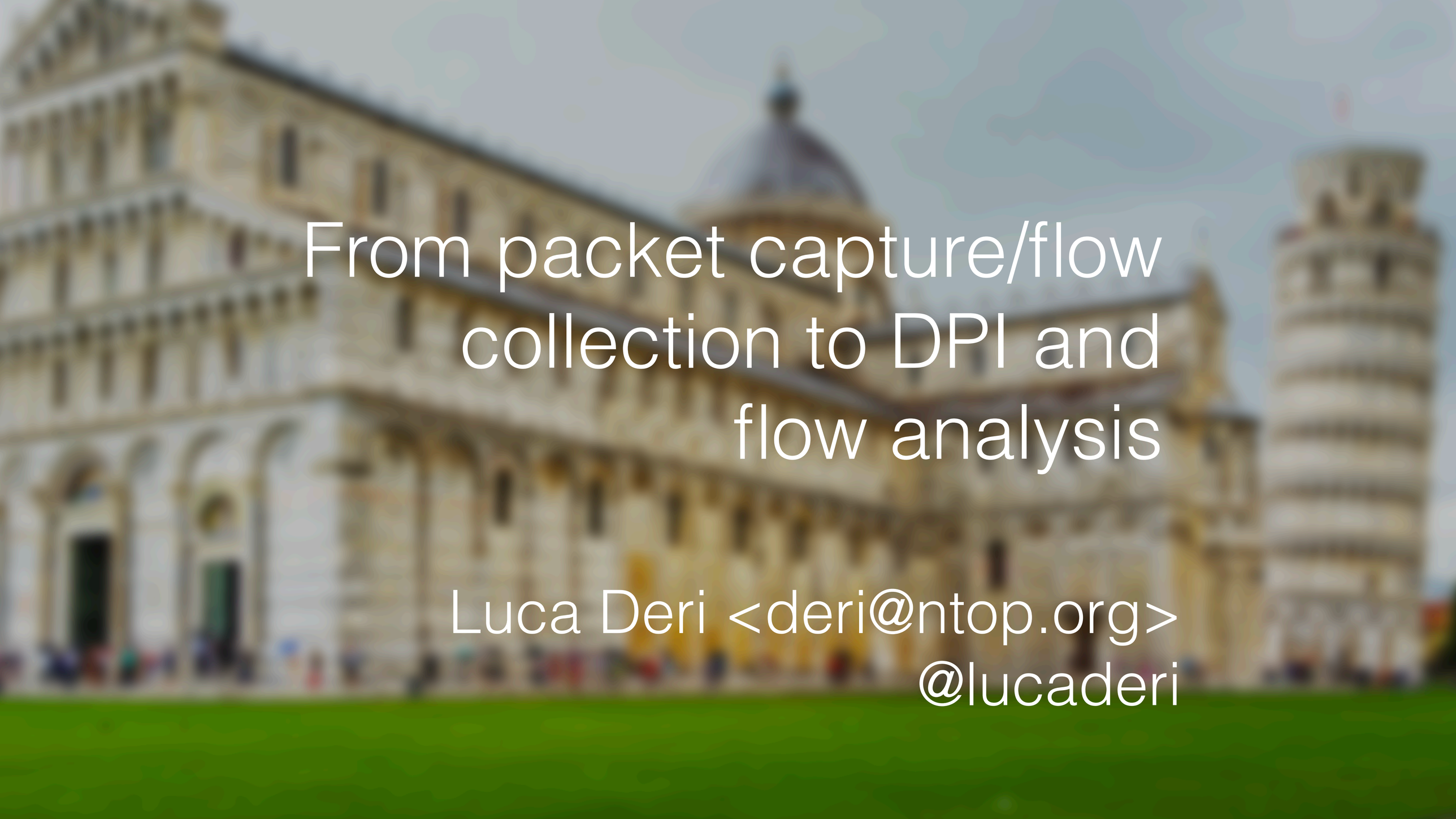


From packet capture/flow collection to DPI and flow analysis

Luca Deri <deri@ntop.org>
@lucaderi

Topics of This Session

- Open Source
  - ntopng: Web-based monitoring application
  - PF_RING: Accelerated RX/TX on Linux
  - nDPI: Deep Packet Inspection Toolkit for Cybersecurity
- Proprietary
  - PF_RING ZC: 1/10/40/100 Gbit Line rate.
  - **nProbe**: 10G NetFlow/IPFIX Probe
  - **nProbe Cento**: flows+packets+security at 40/100 Gbit
  - n2disk/disk2n Network-to-disk and disk-to-network
  - nScrub: Software DDoS Mitigation
  - **nTap**: virtual tap for hosts, containers, and VMs

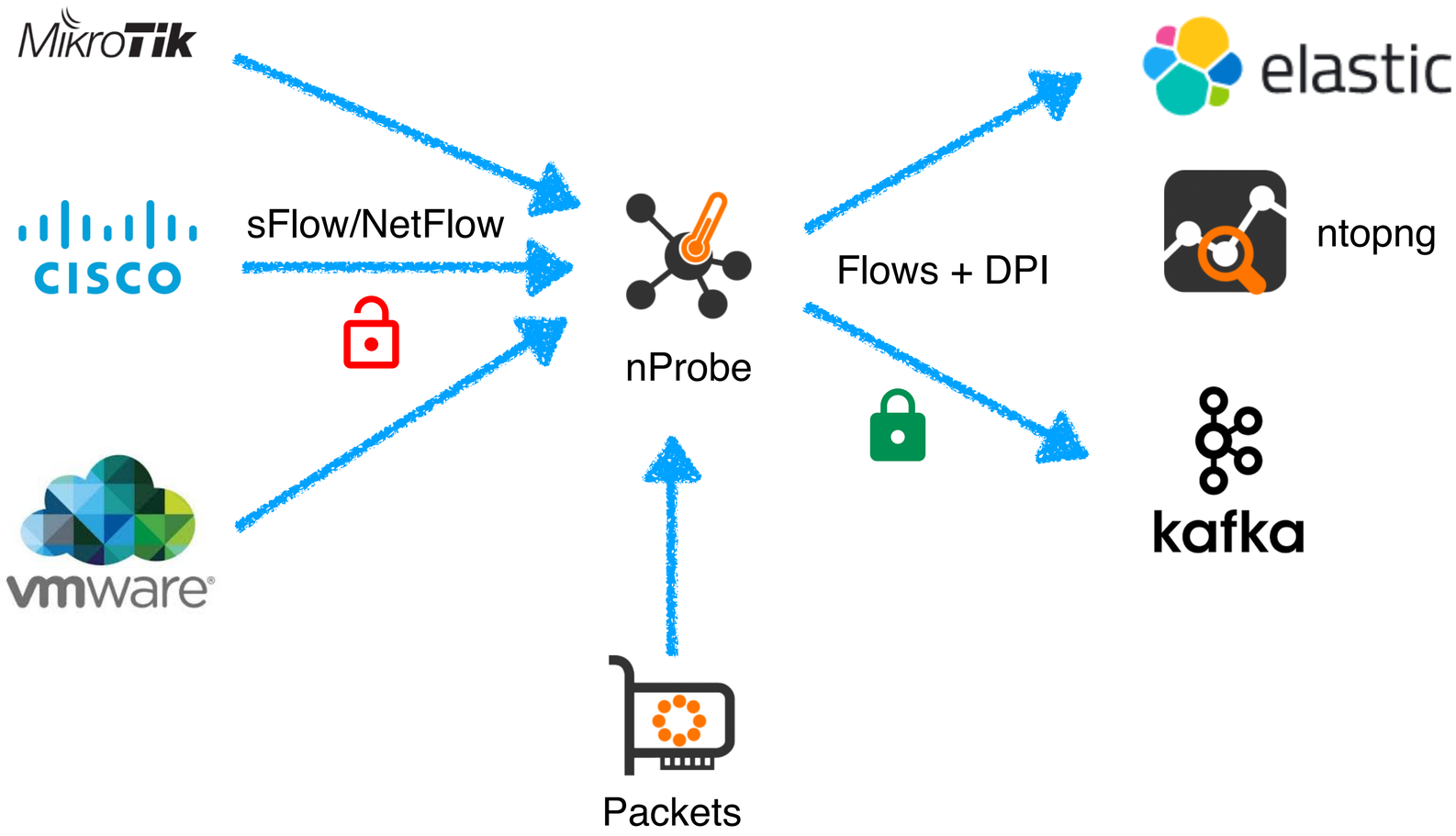
What is nProbe? [1/2]

- nProbe is an extensible NetFlow/IPFIX/sFlow application able to:
  - Capture packets and turn them into flows that are exported in NetFlow/IPFIX format to external collectors, or JSON to ntopng via ZMQ .
  - Collect flows and re-export them (proxy mode).
  - Collect flows and dump them to disk, external consumers (Kafka, ElasticSearch, Syslog, TCP streaming).
  - Fully nDPI-based and extensible via plugin architecture (e.g. VoIP, Email, DNS, HTTP, 3G/4G, Radius…)

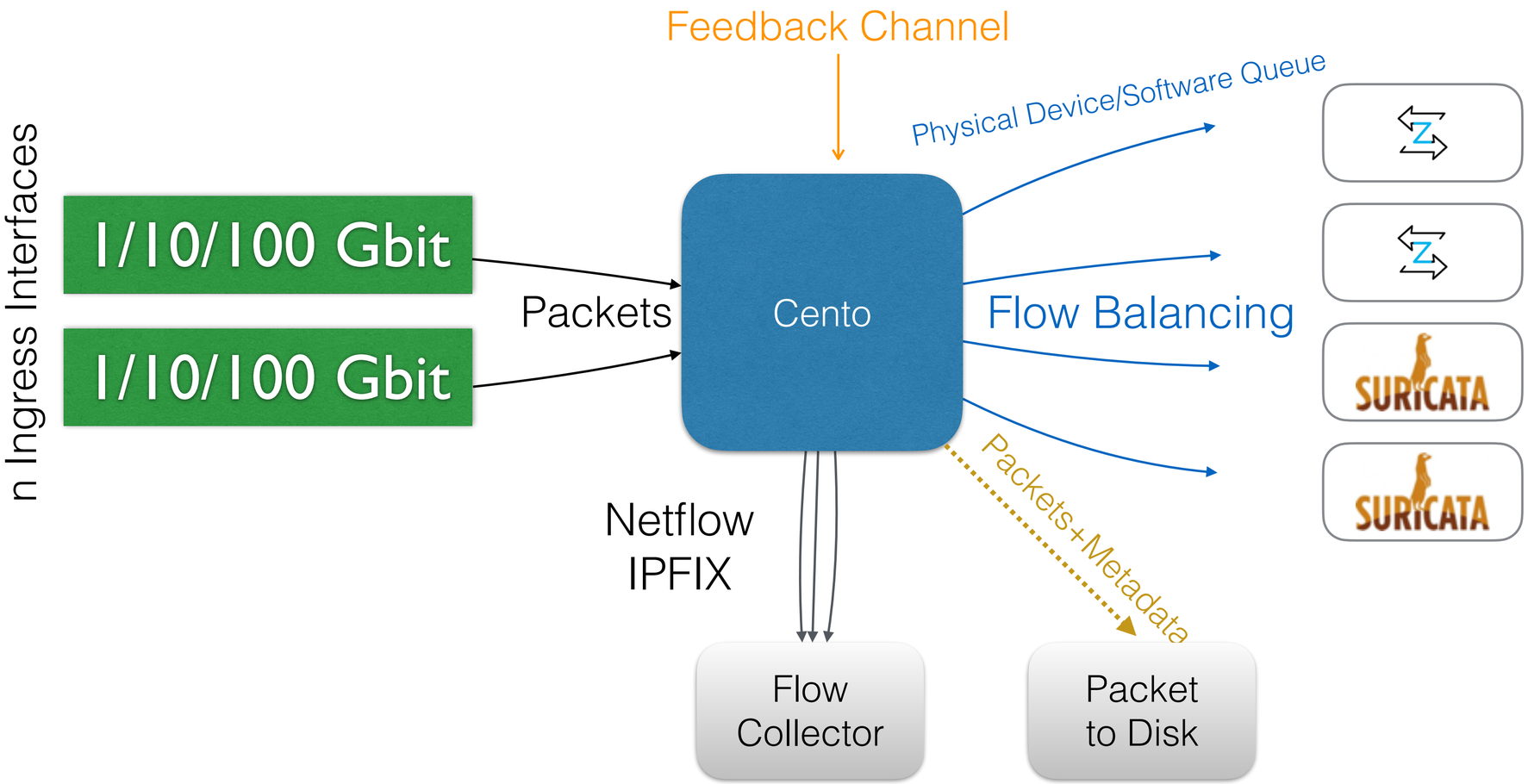
What is nProbe? [2/2]

- nProbe is the oldest product ntop develops (since 2002).
- Initially conceived as a drop-in replacements for Cisco NetFlow analysers it is now a versatile tool for generating and emitting flow-based information.
- It is available in the “classic” nProbe (extremely versatile but not able to go above 10 Gbit) or “cento” designed for 40/100 Gbit networks (less versatile but speed native).
- Both products sit on top of nDPI.
- Support for Linux, Windows, MacOS, BSD (OPNsense, pfSense)

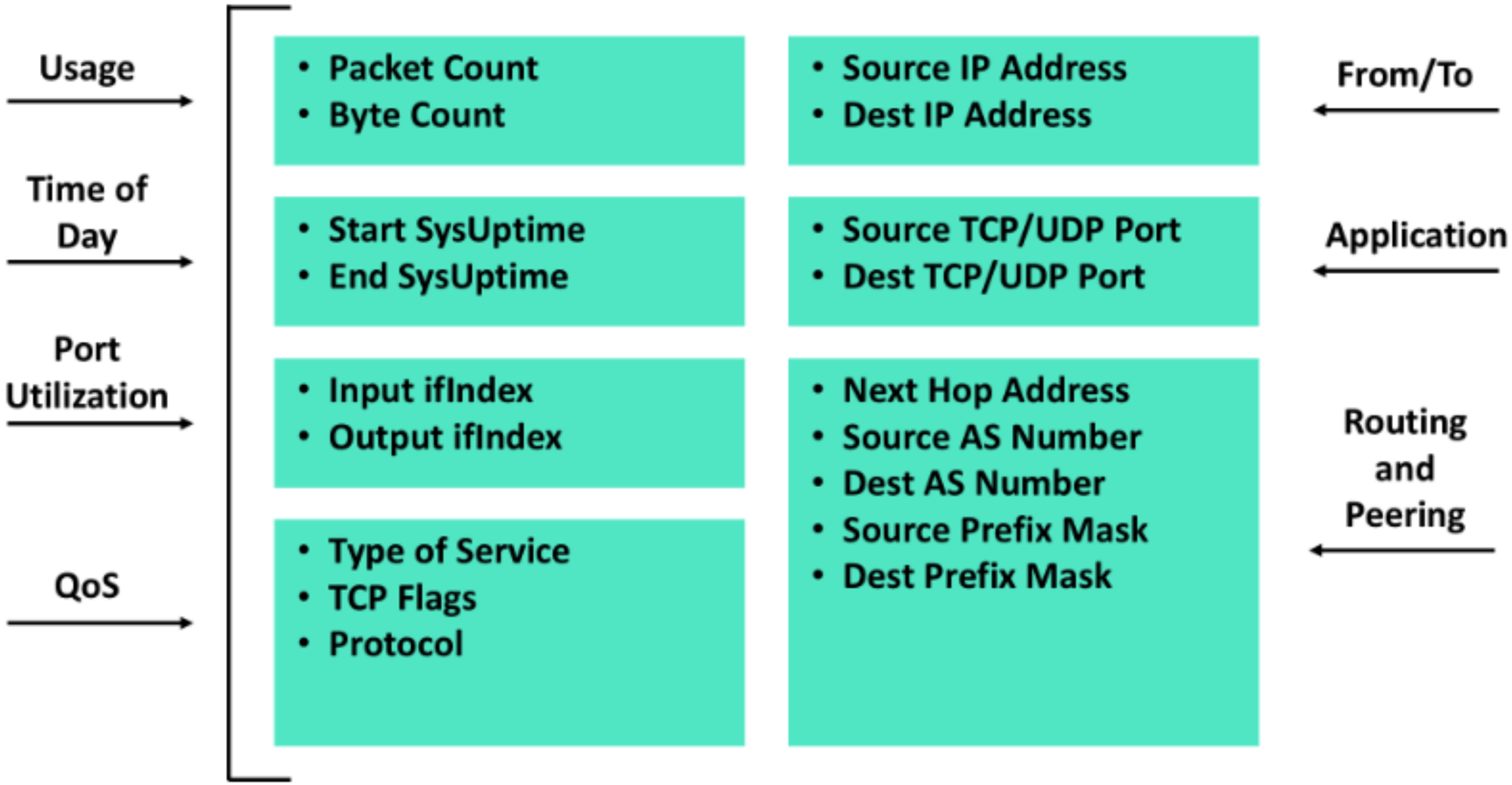
nProbe Architecture



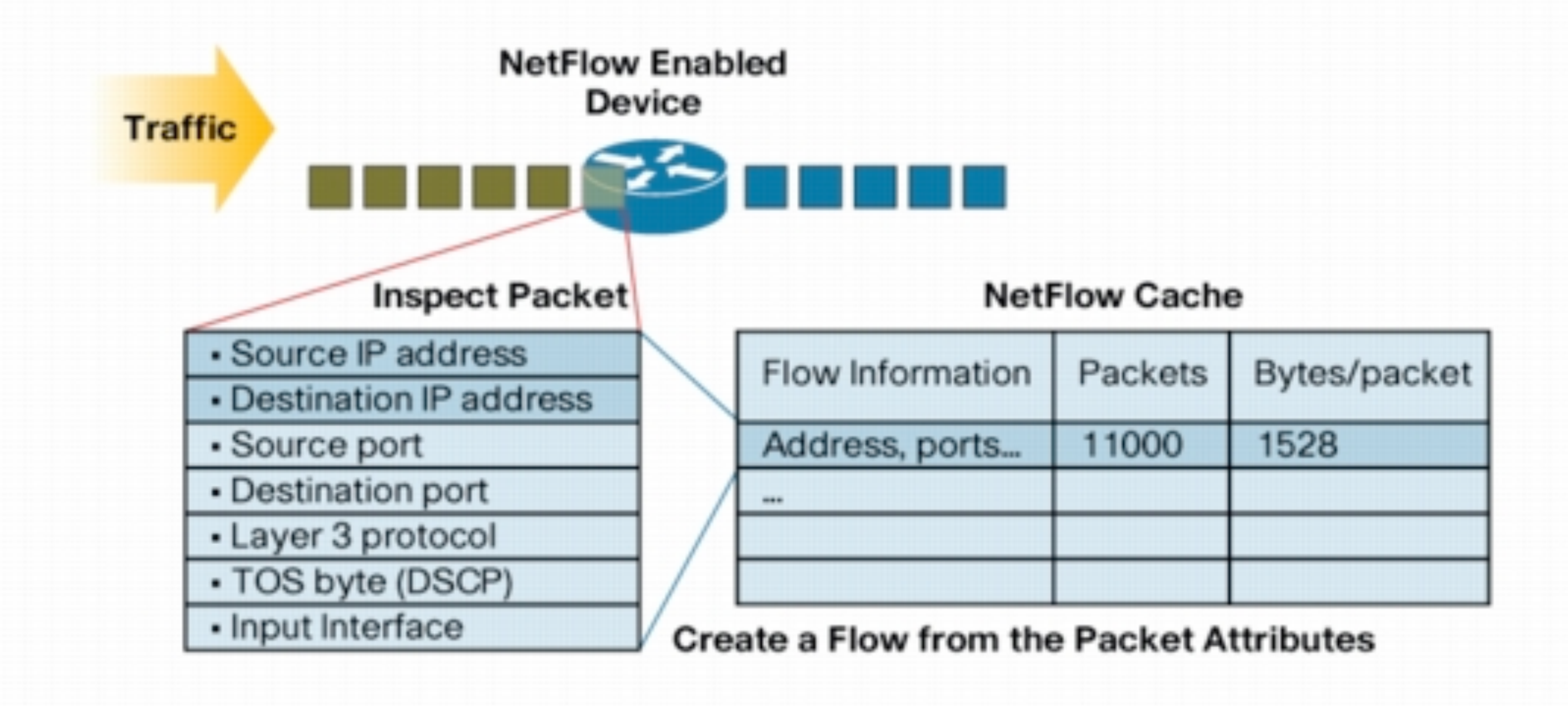
nProbe Cento Architecture



What's Inside a Flow ?



Flow Lifecycle [1/2]

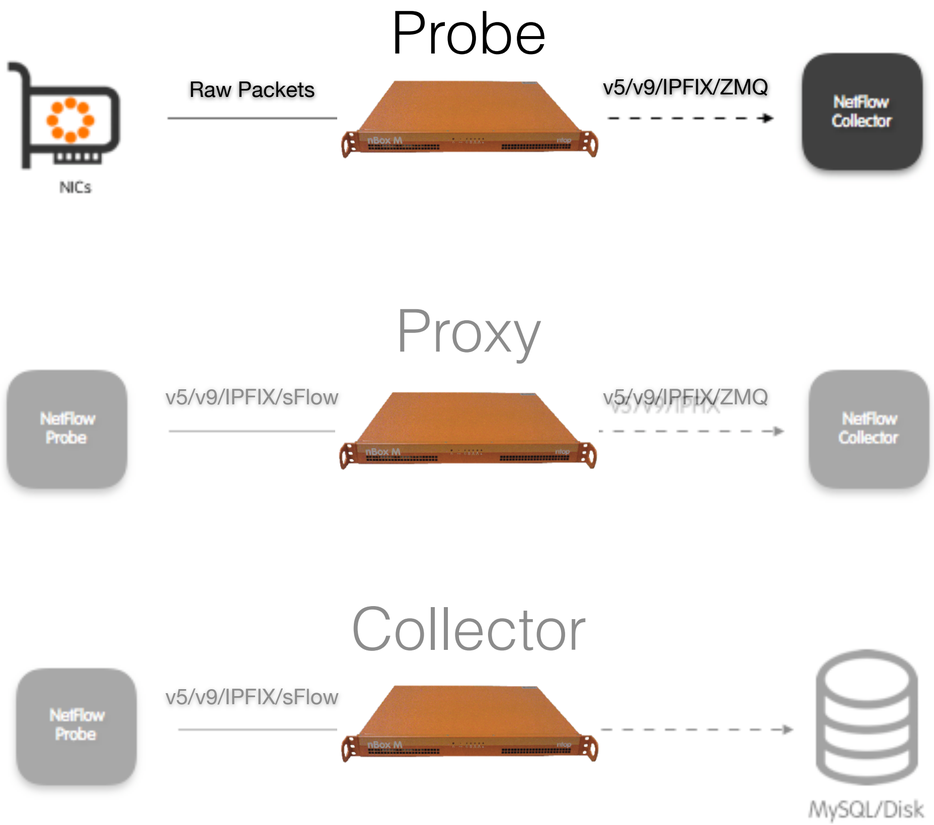


Flow Lifecycle [2/2]

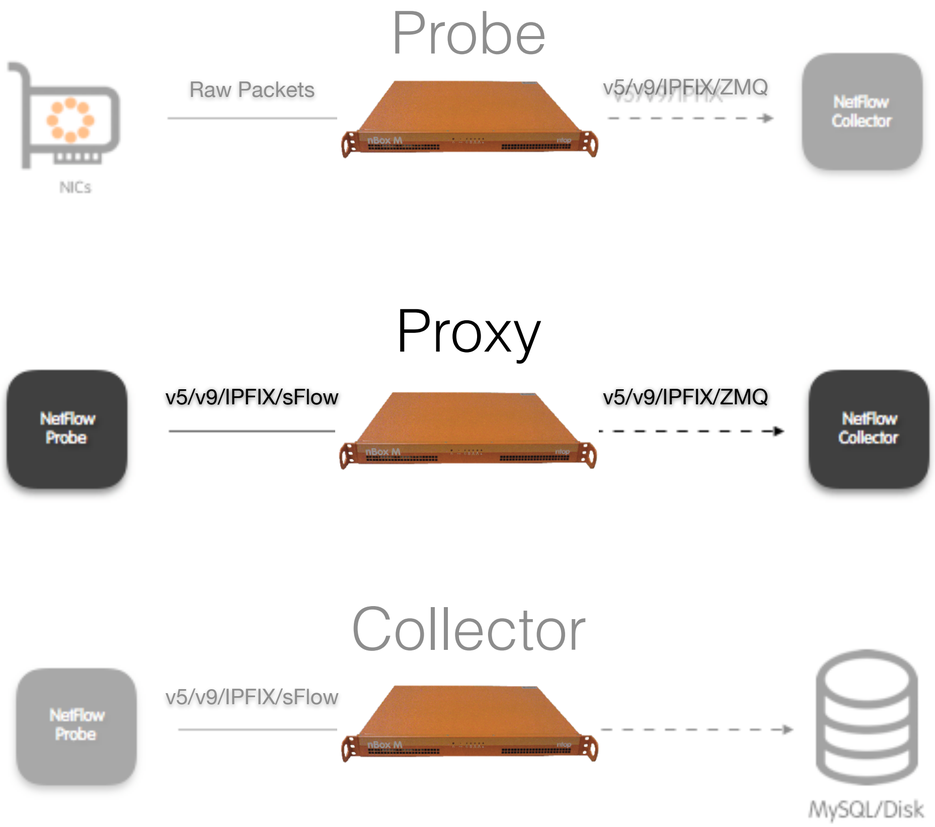
- Flows are terminated when one of these conditions are met:
  - The network communication has ended (e.g. a packet contains the TCP FIN flag).
  - The flow lasted too long (default 30 min).
  - The flow has been not active (i.e. no new packets have been received) for too long (default 15 sec).

```
[--lifetime-timeout|-t] <timeout>   | It specifies the maximum (seconds) flow
                                    | lifetime [default=120]
[--idle-timeout|-d] <timeout>       | It specifies the maximum (seconds) flow
                                    | idle lifetime [default=60]
[--queue-timeout|-l] <timeout>      | It specifies how long expired flows
                                    | (queued before delivery) are emitted
                                    | [default=30]
```

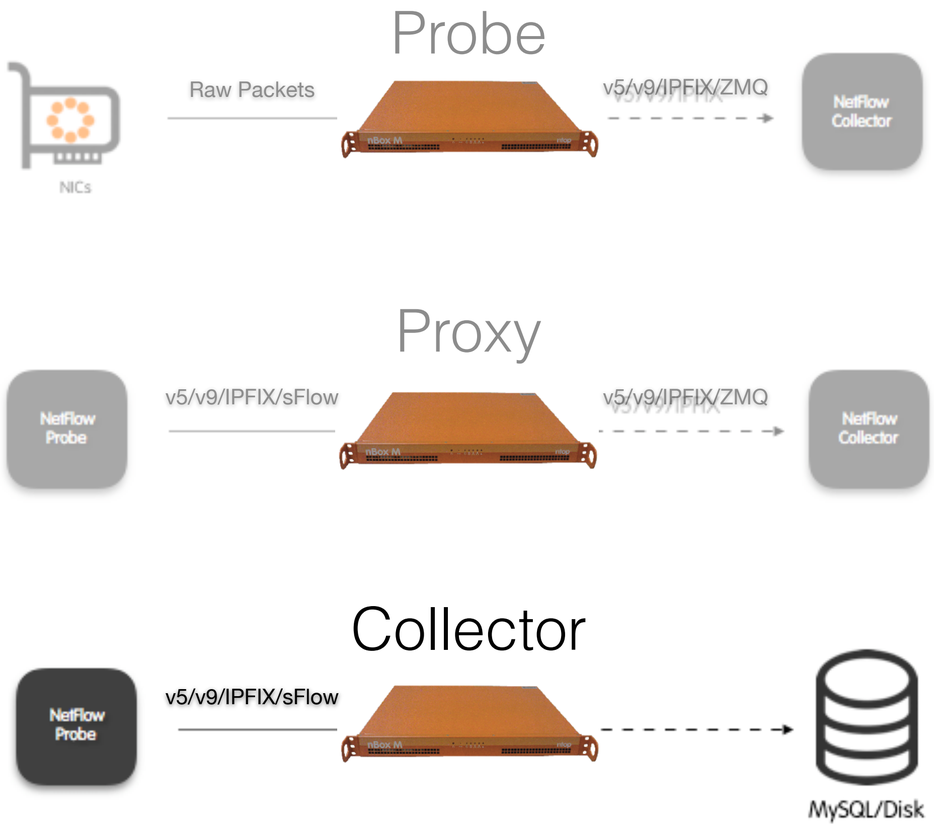
Operational Modes [1/3]



Operational Modes [2/3]



Operational Modes [3/3]

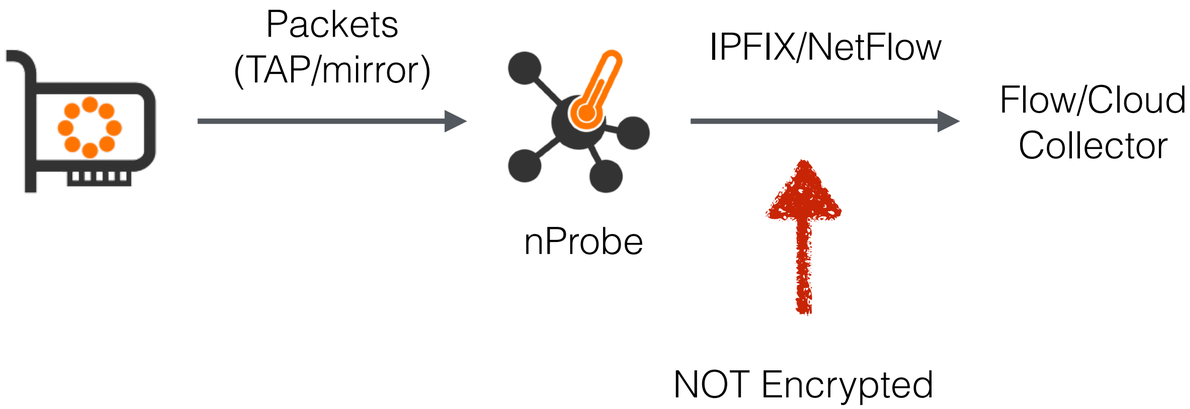


nProbe: Probe Mode [1/3]

- Use cases:
  - You need to monitor traffic on a remote network and send flow to a (few) central location(s).
  - Your router does not do DPI and cybersecurity analysis.
  - Your router has flaws or severe limitations. Typical examples: firewall (Cisco ASA), flaws/limitations (Fortinet).
  - Your router is working fine but it does not analyse the whole network stream but only a subset of it, or it uses packet sampling.
  - You need to compute advanced statistics (e.g. voice quality), or basic stats (TCP retransmissions or packet drops).

nProbe: Probe Mode [2/3]

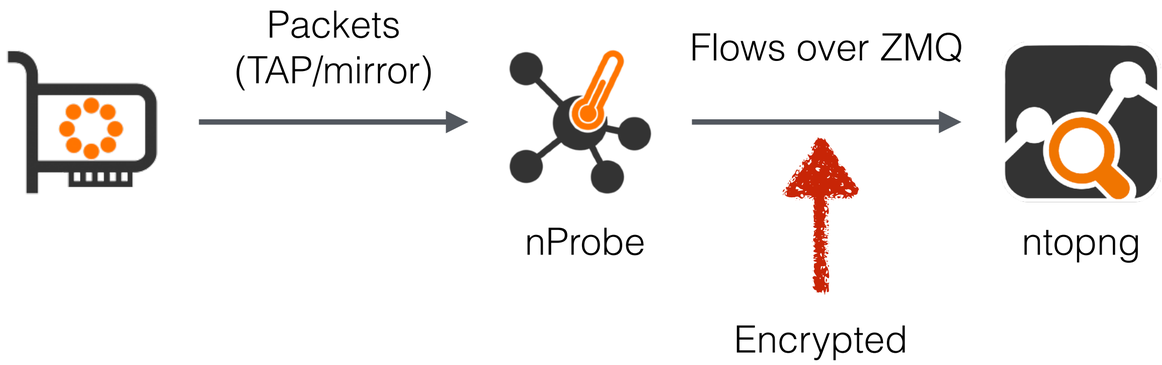
- nProbe to flow collector




```
$ nprobe -i eth1 -n <collector IP>:<collector port>
```

nProbe: Probe Mode [3/3]

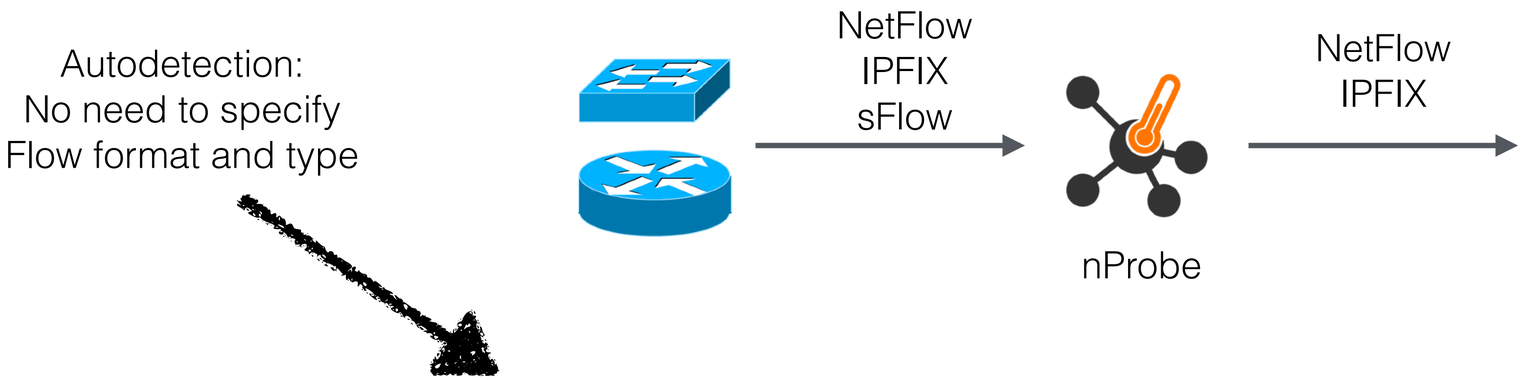
- nProbe to ntopng:
  - nProbe processes live traffic
  - ntopng collects and analyses flows




```
$ nprobe -i eth1 —zmq tcp://*:5556 -T "@NTOPNG@"

$ ntopng -i tcp://127.0.0.1:5556
```

nProbe: Proxy Mode

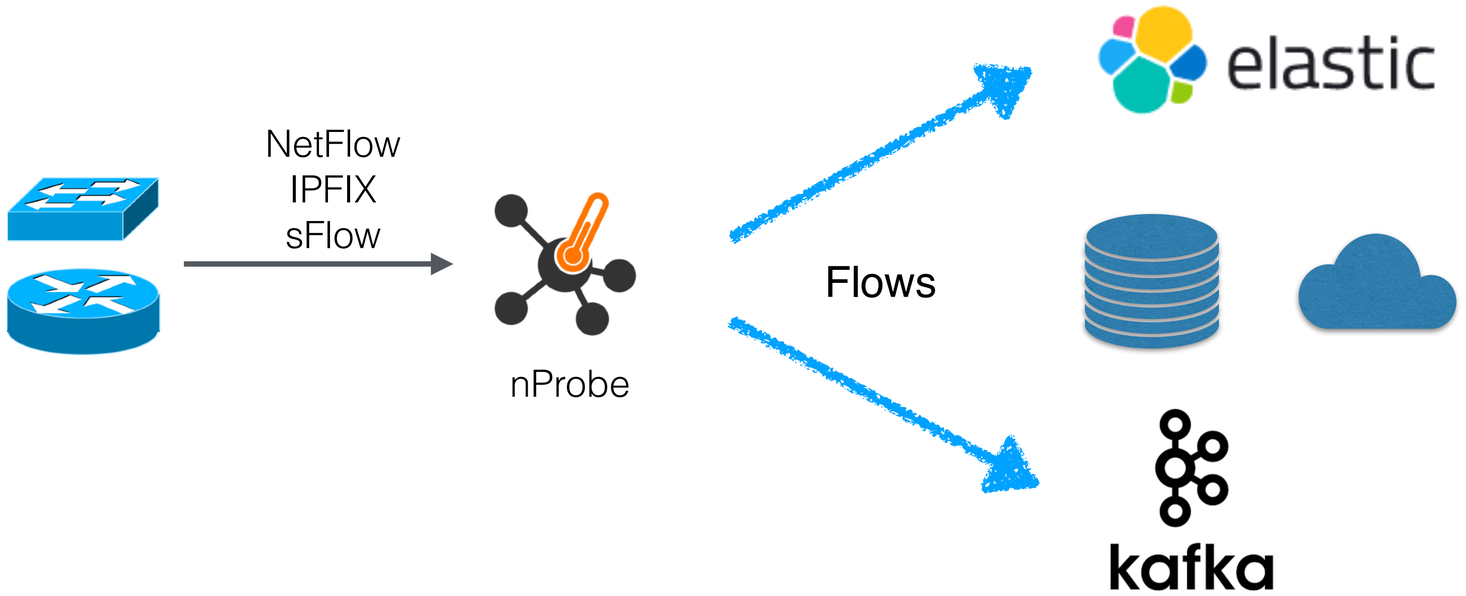
- Use case:
  - Your router exports flows in a format that a remote (non ntop) flow collector cannot understand.
  - You need to translate sFlow to NetFlow/IPFIX




```
$ nprobe -3 <local coll. port> -n <collector IP>:<collector port>
```

nProbe: Collector Mode [1/2]

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



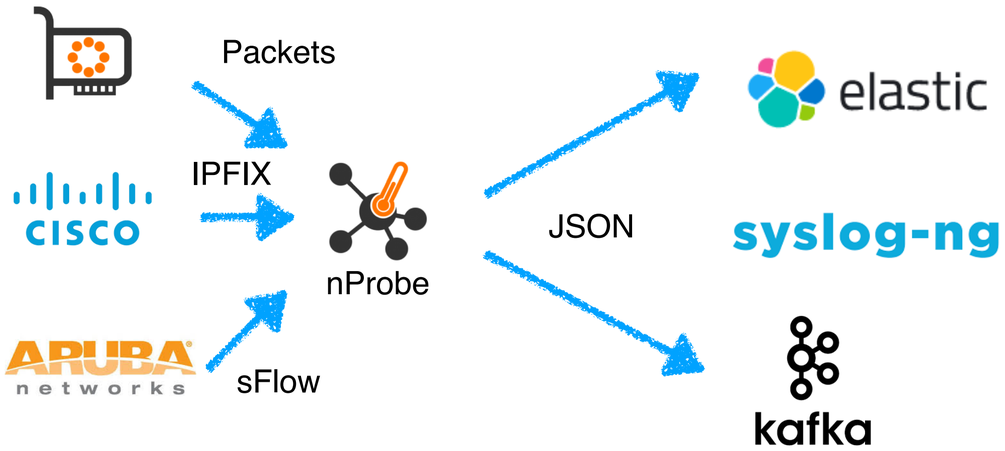

```
$ nprobe -i none --collector-port 2055 --kafka <brokers>;<ftopic>;[<otopic>;<ack>;<comp>]
--elastic <format> --mysql=<host[@port]|unix socket>:<dbname>:<prefix>:<user>:<pw>
--clickhouse=<host[@port]>:<dbname>:<prefix>:<user>:<pw>
```

nProbe: Collector Mode [2/2]

```
[--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 collectior 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
```

Using nProbe to Feed Datalakes

nprobe -i <device> …. --elastic <index type>;<index name>;<es URL>;<es user>:<es pwd>



nprobe -i <device> …. --kafka <brokers>;<topic>;[<opt topic>;<ack>;<comp>]

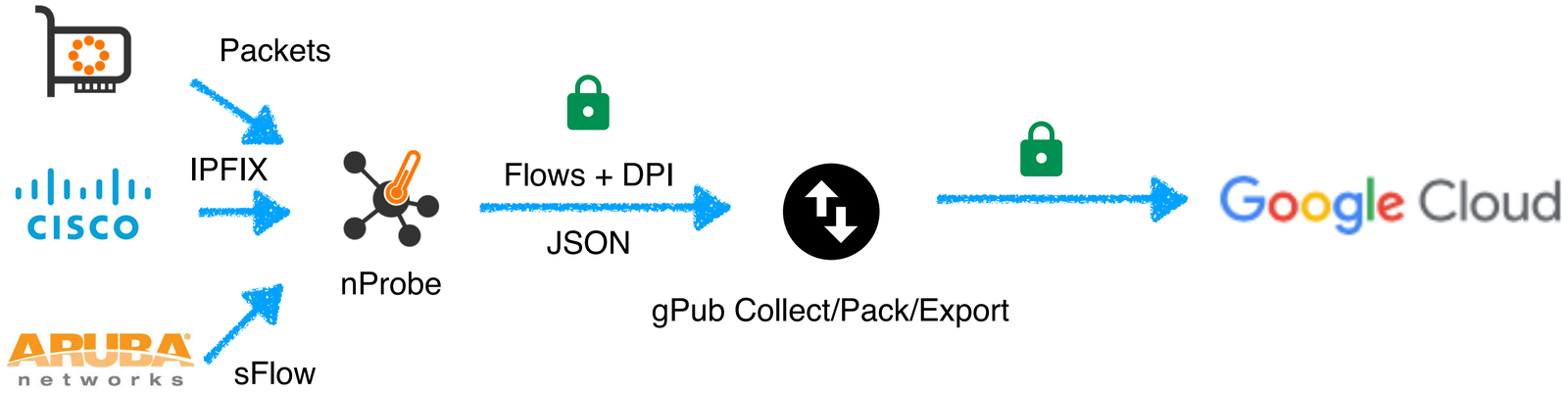
nProbe: AWS Cloud

- Most cloud providers offer log-based monitoring tools for supervising hosted network activities.
- nProbe can now be used to turn AWS logs into standard IPFIX flows.
- AWS logs are stored on S3 buckets that nProbe consumes by looking at files specified with -i <directory>

```
account-id action az-id bytes dstaddr dstport end flow-direction instance-id interface-id log-status packets pkt-dst-aws-service pkt-dstaddr pkt-src-aws-service pkt-srcaddr protocol region srcaddr srcport start sublocation-id
sublocation-type subnet-id tcp-flags traffic-path type version vpc-id
421717577885 ACCEPT use1-az6 396 10.113.39.219 80 1640154903 ingress - eni-0afec37a7c4be140d OK 5 - 10.113.39.219 - 10.113.39.208 6 us-east-1 10.113.39.208 7652 1640154859 - - subnet-048dbd0af4e64ae1f 3 - IPv4 5 vpc-0f4cdb08d3b1bcdf6
421717577885 ACCEPT use1-az6 1895 10.113.39.208 7652 1640154903 egress - eni-0afec37a7c4be140d OK 5 - 10.113.39.208 - 10.113.39.219 6 us-east-1 10.113.39.219 80 1640154859 - - subnet-048dbd0af4e64ae1f 19 1 IPv4 5 vpc-0f4cdb08d3b1bcdf6
421717577885 ACCEPT use1-az6 158 10.113.39.219 53540 1640154903 ingress - eni-0afec37a7c4be140d OK 1 - 10.113.39.219 - 10.112.84.16 17 us-east-1 10.112.84.16 53 1640154859 - - subnet-048dbd0af4e64ae1f 0 - IPv4 5 vpc-0f4cdb08d3b1bcdf6
421717577885 ACCEPT use1-az6 74 10.112.84.16 53 1640154903 egress - eni-0afec37a7c4be140d OK 1 - 10.112.84.16 - 10.113.39.219 17 us-east-1 10.113.39.219 53540 1640154859 - - subnet-048dbd0af4e64ae1f 0 1 IPv4 5 vpc-0f4cdb08d3b1bcdf6
421717577885 ACCEPT use1-az6 396 10.113.39.219 80 1640154903 ingress - eni-0afec37a7c4be140d OK 5 - 10.113.39.219 - 10.113.39.208 6 us-east-1 10.113.39.208 7568 1640154859 - - subnet-048dbd0af4e64ae1f 3 - IPv4 5 vpc-0f4cdb08d3b1bcdf6
421717577885 ACCEPT use1-az6 1895 10.113.39.208 7568 1640154903 egress - eni-0afec37a7c4be140d OK 5 - 10.113.39.208 - 10.113.39.219 6 us-east-1 10.113.39.219 80 1640154859 - - subnet-048dbd0af4e64ae1f 19 1 IPv4 5 vpc-0f4cdb08d3b1bcdf6
```

nProbe: Google Cloud

- For Google cloud users, we now offer the ability to push flows to Google Pub/Sub for creating cloud-based datalakes.
- nProbe can collect (or generate from packet capture) flows, convert them to JSON and push them to Google Cloud in addition to the typical consumers (e.g. ntopng).



nProbe: Data Dump

- At any time you can combine further egress operators for
  - Dumping traffic to disk (-P)
  - Exporting data to Splunk or LogStash (--json-to-syslog)

```
[--dump-path|-P] <path>                 | Directory where dump files will
--dont-nest-dump-dirs                   | Dump files (-P) won't be saved on nested dirs.
                                        | effect without -P.
--csv-separator <separator>             | Specify the text files separator (see -P)
--json-to-syslog                        | Export flows in JSON format to syslog
--json-labels                           | In case JSON label is used (e.g. with ZMQ)
```

ntopng with nProbe

- Specify the ZMQ endpoint:

```
# ntopng -i tcp://127.0.0.1:5556 --zmq-encryption

# nprobe -i enp0s3 -n none --zmq tcp://*:5556 --zmq-encryption-
key xxx
```

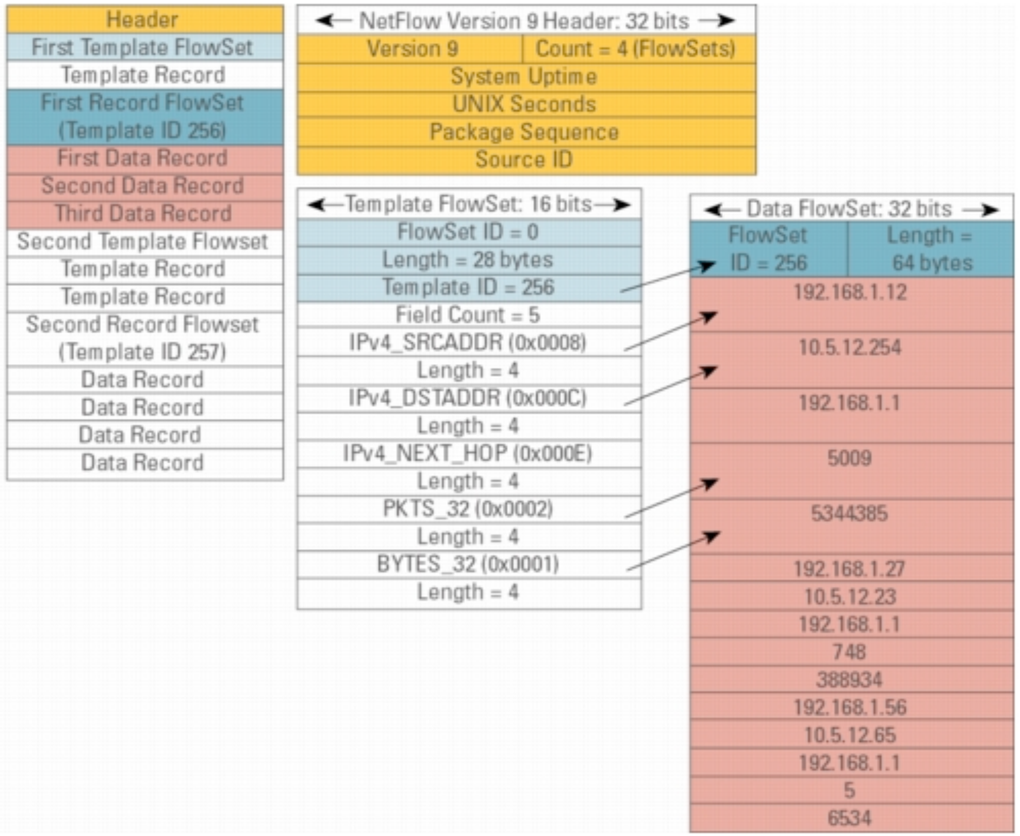
- or using the configuration file:

```
# cat /etc/ntopng/ntopng.conf
-i=tcp://127.0.0.1:5556

# cat /etc/nprobe/nprobe.conf
-i=enp0s3
-n=none
--zmq=tcp://127.0.0.1:5556

# systemctl restart ntopng
# systemctl restart nprobe
```

nProbe: Templates [1/2]

In flow-based analysis the flow template defines how data is exported and what it contains.

nProne is no exception and via -T <template> you can define how traffic analysis is made and what information elements are exported.



nProbe: Templates [2/2]

```
NetFlow v9/IPFIX format [-T]
---------------
The following options can be used to specify the format:

 ID   NetFlow Label                IPFIX Label                    Description
-------------------------------------------------------------------------------
[  1][Len 4] %IN_BYTES                    %octetDeltaCount          Incoming flow bytes (src->dst)
[  2][Len 4] %IN_PKTS                     %packetDeltaCount         Incoming flow packets (src->dst)
[  4][Len 1] %PROTOCOL                    %protocolIdentifier        IP protocol byte
[NFv9 58500][IPFIX 35632.1028][Len 16]    %PROTOCOL_MAP              IP protocol name
[  5][Len 1] %SRC_TOS                     %ipClassOfService          TOS/DSCP (src->dst)
[  6][Len 1] %TCP_FLAGS                   %tcpControlBits            Cumulative of all flow TCP flags
[  7][Len 2] %L4_SRC_PORT                 %sourceTransportPort       IPv4 source port
[NFv9 58503][IPFIX 35632.1031][Len 16]    %L4_SRC_PORT_MAP           Layer 4 source port symbolic name
[  8][Len 4] %IPV4_SRC_ADDR               %sourceIPv4Address         IPv4 source address
[  9][Len 1] %IPV4_SRC_MASK               %sourceIPv4PrefixLength    IPv4 source subnet mask (/<bits>)
[ 10][Len 4] %INPUT_SNMP                  %ingressInterface          Input interface SNMP idx
[ 11][Len 2] %L4_DST_PORT                 %destinationTransportPort  IPv4 destination port
[NFv9 58507][IPFIX 35632.1035][Len 16]    %L4_DST_PORT_MAP           Layer 4 destination port symbolic name
[NFv9 58508][IPFIX 35632.1036][Len 2]     %L4_SRV_PORT               Layer 4 server port
[NFv9 58509][IPFIX 35632.1037][Len 16]    %L4_SRV_PORT_MAP           Layer 4 server port symbolic name
[ 12][Len 4] %IPV4_DST_ADDR               %destinationIPv4Address    IPv4 destination address
[ 13][Len 1] %IPV4_DST_MASK               %destinationIPv4PrefixLengthIPv4 dest subnet mask (/<bits>)
```

Templates and Data Dump

- Configure nProbe to generate flows capturing traffic and dumping them to disk in text format

```
# nprobe -i eth1 -n none —P /tmp
```

- Now export TCP metrics into the dumped flows

```
# nprobe -i eth1 -n none —P /tmp -T "%IPV4_SRC_ADDR %IPV4_DST_ADDR
%INPUT_SNMP %OUTPUT_SNMP %IN_PKTS %IN_BYTES %FIRST_SWITCHED
%LAST_SWITCHED %L4_SRC_PORT %L4_DST_PORT %TCP_FLAGS %PROTOCOL
%RETRANSMITTED_OUT_BYTES %RETRANSMITTED_OUT_PKTS "
```

- Now also export network latency (Hint: use nprobe -H |grep LATENCY)
- Now enable DPI traffic analysis (Hint: use nprobe -H |grep L7)
-

Flow Aggregation

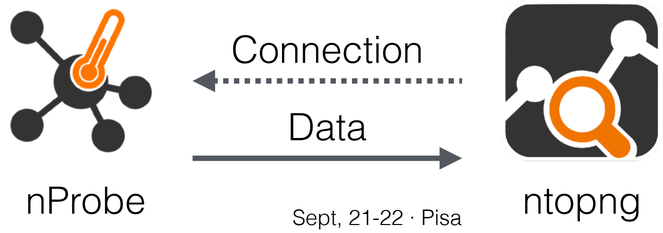
- Flows can be aggregated using “external” criteria and not just based on raw flow fields.
- Usually these external criteria are applied on “key” (not “value”) fields such as port, IP address, protocol etc. and are used to group values together.

```
[--aggregation|-p] <aggregation>    | It specifies the flow aggregation level:
                                    | <OuterVLAN>.<InnerVLAN>/<proto>/<IP>/<port>/<TOS>/<SCTPStreamId>/<exporterIP>/<ICMP>/<OUT Bytes/Pkts>
                                    | where each element can be set to 0=ignore
                                    | or 1=take care. Example '-p 1.0/0/1/1/1/1/0/1'
                                    | ignores the protocol, whereas
                                    | '-p 0.0/0/1/0/0/0/0/0/0' ignores everything
                                    | but the IP. Default: 1.1/1/1/1/0/1/1/0/1
                                    | Note: [ETH][Outer VLAN][Inner VLAN][IP...]
```

Flow Collection: Poll Mode

- Contrary to what happens in NetFlow/sFlow, by default ntopng (collector) connects to nProbe (probe) using ZMQ, and fetches the emitted flows.
- Multiple collectors can connect to the same probe.
- ntopng connects to nProbe:

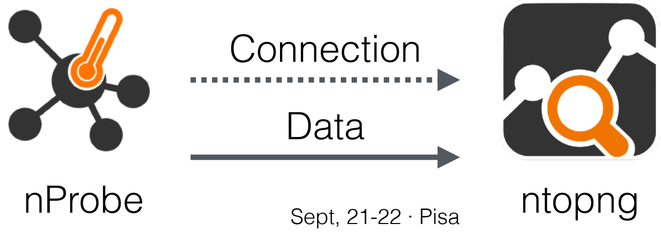
```
host X# ntopng -i "tcp://10.0.0.1:5556"
host Y# nprobe -n none --zmq "tcp://*:5556"
```



Flow Collection: Push Mode

- Optionally, nProbe can be configured to connect to ntopng
- Useful when nProbe is behind a NAT or firewall
- Multiple probes can connect to the same collector.
- nProbe connects to ntopng:

```
host X# ntopng -i “tcp://*:1234c”
host Y# nprobe -n none --zmq “tcp://10.0.0.2:1234” --zmq-probe-mode
```



ntopng with nProbe: Poll vs Push

- Test both modes

```
# ntopng -i tcp://127.0.0.1:5556

# nprobe -i enp0s3 -n none --zmq tcp://*:5556
```

vs

```
# ntopng -i tcp://*:5556c

# nprobe -i enp0s2 -n none --zmq tcp://127.0.0.1:5556 --zmq-
probe-mode
# nprobe -i enp0s1 -n none --zmq tcp://127.0.0.1:5556 --zmq-
probe-mode
```

nProbe Plugins

- nProbe plugins (requires Enterprise Edition) are dissectors for specific protocols that allow rich protocol information to be exported.
- They can be enabled by specifying their name in the template. Example for HTTP:

```
Plugin HTTP Protocol templates:
[NFv9 57652][IPFIX 35632.180][Len 128 varlen] %HTTP_URL                 HTTP URL (IXIA URI)
[NFv9 57832][IPFIX 35632.360][Len 4 varlen] %HTTP_METHOD                HTTP METHOD
[NFv9 57653][IPFIX 35632.181][Len 2] %HTTP_RET_CODE                     HTTP return code (e.g. 200, 304...)
[NFv9 57654][IPFIX 35632.182][Len 128 varlen] %HTTP_REFERER             HTTP Referer
[NFv9 57655][IPFIX 35632.183][Len 256 varlen] %HTTP_USER_AGENT          HTTP User Agent
[NFv9 57656][IPFIX 35632.184][Len 256 varlen] %HTTP_MIME                HTTP Mime Type
[NFv9 57659][IPFIX 35632.187][Len 64 varlen] %HTTP_HOST                 HTTP(S) Host Name (IXIA Host Name)
[NFv9 57833][IPFIX 35632.361][Len 64 varlen] %HTTP_SITE                 HTTP server without host name
[NFv9 57932][IPFIX 35632.460][Len 256 varlen] %HTTP_X_FORWARDED_FOR     HTTP X-Forwarded-For
[NFv9 57933][IPFIX 35632.461][Len 256 varlen] %HTTP_VIA                 HTTP Via
```

nProbe Plugins: Zoom/MS Teams Monitoring [1/2]

- nDPI has been enhanced…

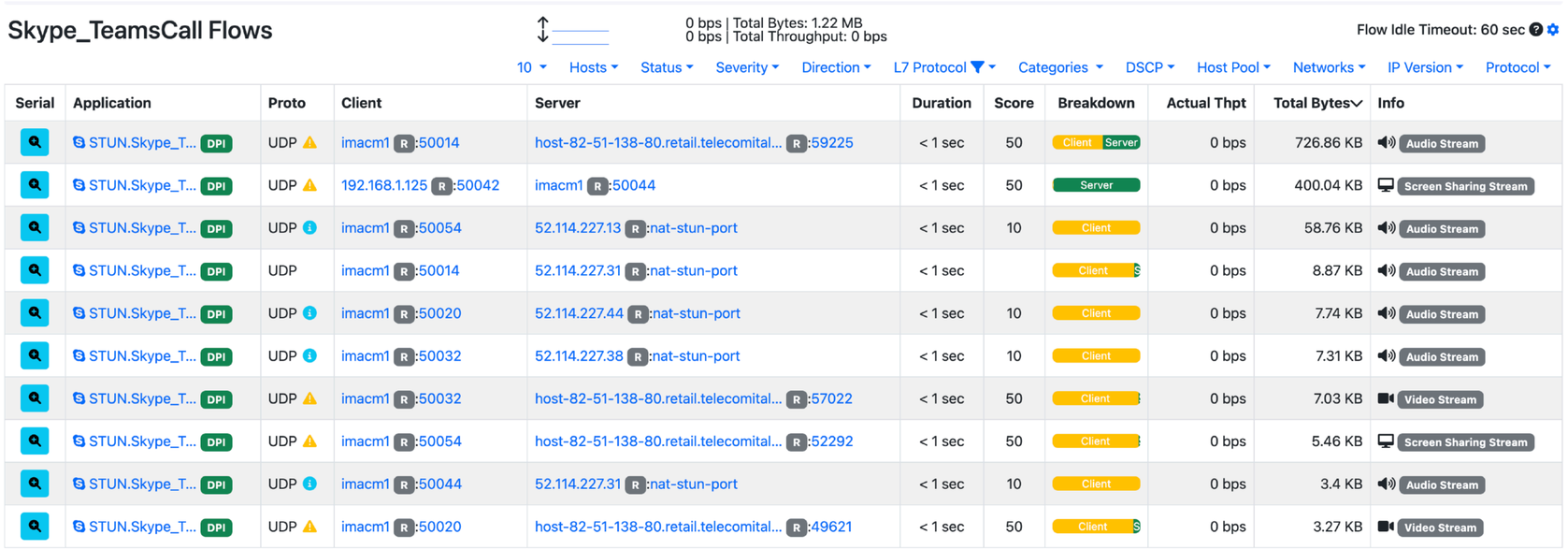
```
 38 Skype_TeamsCall        TCP                 Acceptable   VoIP
125 Skype_Teams            UDP                 Acceptable   VoIP
189 Zoom                   TCP                 Acceptable   Video
250 Teams                  TCP                 Safe         Collaborative
```

- nProbe has been Enhanced to handle STUN/RTP flows with “non-standard”

```
[NFv9 57626][IPFIX 35632.154][Len 4] %RTP_IN_JITTER            RTP jitter (ms * 1000)
[NFv9 57627][IPFIX 35632.155][Len 4] %RTP_OUT_JITTER           RTP jitter (ms * 1000)
[NFv9 57628][IPFIX 35632.156][Len 4] %RTP_IN_PKT_LOST          Packet lost in stream (src->dst)
[NFv9 57629][IPFIX 35632.157][Len 4] %RTP_OUT_PKT_LOST         Packet lost in stream (dst->src)
[NFv9 57902][IPFIX 35632.430][Len 4] %RTP_IN_PKT_DROP          Packet discarded by Jitter Buffer (src->dst)
[NFv9 57903][IPFIX 35632.431][Len 4] %RTP_OUT_PKT_DROP         Packet discarded by Jitter Buffer (dst->src)
[NFv9 57633][IPFIX 35632.161][Len 1] %RTP_IN_PAYLOAD_TYPE      RTP payload type
[NFv9 57630][IPFIX 35632.158][Len 1] %RTP_OUT_PAYLOAD_TYPE     RTP payload type
[NFv9 57631][IPFIX 35632.159][Len 4] %RTP_IN_MAX_DELTA         Max delta (ms*100) between consecutive pkts (src->dst)
[NFv9 57632][IPFIX 35632.160][Len 4] %RTP_OUT_MAX_DELTA        Max delta (ms*100) between consecutive pkts (dst->src)
[NFv9 57820][IPFIX 35632.348][Len 64 varlen] %RTP_SIP_CALL_ID          SIP call-id corresponding to this RTP stream
[NFv9 57906][IPFIX 35632.434][Len 4] %RTP_MOS                  RTP pseudo-MOS (value * 100) (average both directions)
[NFv9 57842][IPFIX 35632.370][Len 4] %RTP_IN_MOS               RTP pseudo-MOS (value * 100) (src->dst)
[NFv9 57904][IPFIX 35632.432][Len 4] %RTP_OUT_MOS              RTP pseudo-MOS (value * 100) (dst->src)
[NFv9 57908][IPFIX 35632.436][Len 4] %RTP_R_FACTOR             RTP pseudo-R_FACTOR (value * 100) (average both directions)
[NFv9 57843][IPFIX 35632.371][Len 4] %RTP_IN_R_FACTOR          RTP pseudo-R_FACTOR (value * 100) (src->dst)
[NFv9 57905][IPFIX 35632.433][Len 4] %RTP_OUT_R_FACTOR         RTP pseudo-R_FACTOR (value * 100) (dst->src)
[NFv9 57853][IPFIX 35632.381][Len 4] %RTP_IN_TRANSIT           RTP Transit (value * 100) (src->dst)
[NFv9 57854][IPFIX 35632.382][Len 4] %RTP_OUT_TRANSIT          RTP Transit (value * 100) (dst->src)
[NFv9 57852][IPFIX 35632.380][Len 4] %RTP_RTT                  RTP Round Trip Time (ms)
```

nProbe Plugins: Zoom/MS Teams Monitoring [2/2]

- And ntopng too…



Skype_TeamsCall Flows

0 bps | Total Bytes: 1.22 MB
0 bps | Total Throughput: 0 bps

Flow Idle Timeout: 60 sec

10 | Hosts | Status | Severity | Direction | L7 Protocol | Categories | DSCP | Host Pool | Networks | IP Version | Protocol

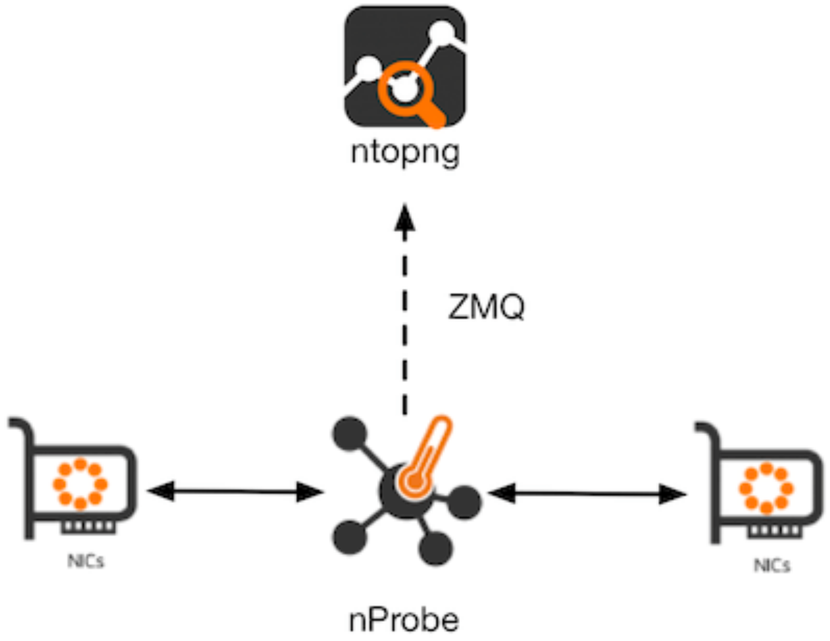
Serial	Application	Proto	Client	Server	Duration	Score	Breakdown	Actual Thpt	Total Bytes	Info
	STUN.Skype_T... DPI	UDP	imacm1 R :50014	host-82-51-138-80.retail.telecomital... R :59225	< 1 sec	50	Client Server	0 bps	726.86 KB	Audio Stream
	STUN.Skype_T... DPI	UDP	192.168.1.125 R :50042	imacm1 R :50044	< 1 sec	50	Server	0 bps	400.04 KB	Screen Sharing Stream
	STUN.Skype_T... DPI	UDP	imacm1 R :50054	52.114.227.13 R :nat-stun-port	< 1 sec	10	Client	0 bps	58.76 KB	Audio Stream
	STUN.Skype_T... DPI	UDP	imacm1 R :50014	52.114.227.31 R :nat-stun-port	< 1 sec		Client	0 bps	8.87 KB	Audio Stream
	STUN.Skype_T... DPI	UDP	imacm1 R :50020	52.114.227.44 R :nat-stun-port	< 1 sec	10	Client	0 bps	7.74 KB	Audio Stream
	STUN.Skype_T... DPI	UDP	imacm1 R :50032	52.114.227.38 R :nat-stun-port	< 1 sec	10	Client	0 bps	7.31 KB	Audio Stream
	STUN.Skype_T... DPI	UDP	imacm1 R :50032	host-82-51-138-80.retail.telecomital... R :57022	< 1 sec	50	Client	0 bps	7.03 KB	Video Stream
	STUN.Skype_T... DPI	UDP	imacm1 R :50054	host-82-51-138-80.retail.telecomital... R :52292	< 1 sec	50	Client	0 bps	5.46 KB	Screen Sharing Stream
	STUN.Skype_T... DPI	UDP	imacm1 R :50044	52.114.227.31 R :nat-stun-port	< 1 sec	10	Client	0 bps	3.4 KB	Audio Stream
	STUN.Skype_T... DPI	UDP	imacm1 R :50020	host-82-51-138-80.retail.telecomital... R :49621	< 1 sec	50	Client	0 bps	3.27 KB	Video Stream

Flow: 192.168.1.29:50014 ⇄ 82.51.138.80:59225 | **Overview**

Flow Peers [Client / Server]	imacm1 R P :50014 [9C:58:3C:A7:EE:CC] ⇄ host-82-51-138-80.retail.telecomitalia.it R :59225 [10:13:31:F1:39:76]
Protocol / Application	UDP / STUN.Skype_TeamsCall (VoIP) [Confidence: DPI] [Audio Stream]

nProbe as an IPS [1/5]

- nProbe can both operate as a passive probe or be used as a bump-in-the-wire on Linux and FreeBSD/OPNsense



nProbe as an IPS [2/5]

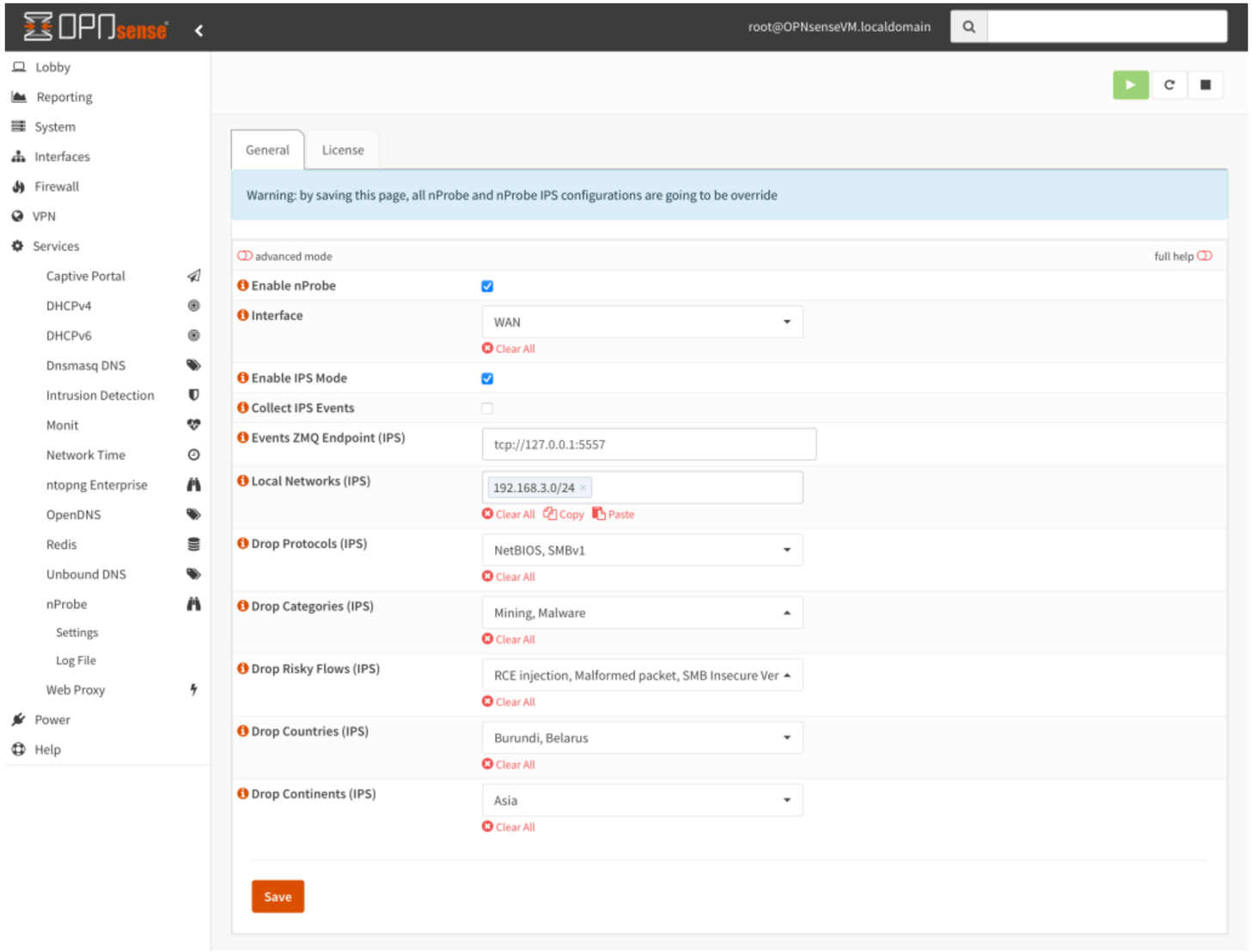
- The configuration is performed by means of a configuration file in JSON format.
- In FreeBSD/OPNsense nProbe leverages on netmap (no configuration but slow) whereas on Linux is uses NFQ queue (faster but it requires confirmation)

```
--ips-mode="/etc/nprobe/nprobe-ips.conf"
-i="nf:0"
-n="none"
```

nProbe as an IPS [3/5]

```
# Pool definition
{"pool":{"id":1,"name":"my pool 1","ip": [ "131.114.0.0/16"], "mac": [ ]},"policy": {"id": 1} }
{"pool":{"id":2,"name":"my pool 2","ip": [ "192.168.1.0/24"], "mac": [ ]},"policy": {"id": 2} }

# Continents: Africa / Asia-Pacific / Europe / North America / South America

# Policy definition
{"policy":{"id":0,"name":"root policy rule", "default_marker": "pass", "markers": { "countries": { "IT": "pass", "CN": "drop",
"US": "pass" } } } }
{"policy":{"id":1,"root":0,"name":"my rule 1", "markers": { "categories": { "Network": 7, "Download-FileTransfer-FileSharing": 8,
"DataTransfer": 8, "VPN": 8, "Video": 9, "Music": 9, "Streaming": 9, "Media": 9 }, "protocols": { "DNS": "drop" }, "countries":
{ "IT": "drop", "CN": "drop", "US": "pass" }, "asn" : { }, "continents" : { "Asia" : "drop"} } , "default_marker": "pass"  } }
{"policy":{"id":2,"root":0,"name":"my rule 2", "markers": { "categories": { "Video": "pass" }, "flow_risk": { "risks": [12],
"marker": "drop" }, "protocols": { "DNS": "drop" }, "countries": { "IT": "pass", "US": "pass" }, "asn" : { "34984" : "drop" } },
"default_marker": "pass"  } }
#{"policy":{"id":2,"root":0,"name":"my rule 2", "markers": { "categories": { "Video": "drop" }, "protocols": { "DNS": "drop" },
"countries": { "IT": "pass", "US": "pass" } }, "default_marker": "pass"  } }

### GeoIP ###

{ "geoip": { "asn": ".GeoLite2-ASN.mmdb", "city": "GeoLite2-City.mmdb" }}
```

nProbe as an IPS [4/5]



nProbe as an IPS [5/5]

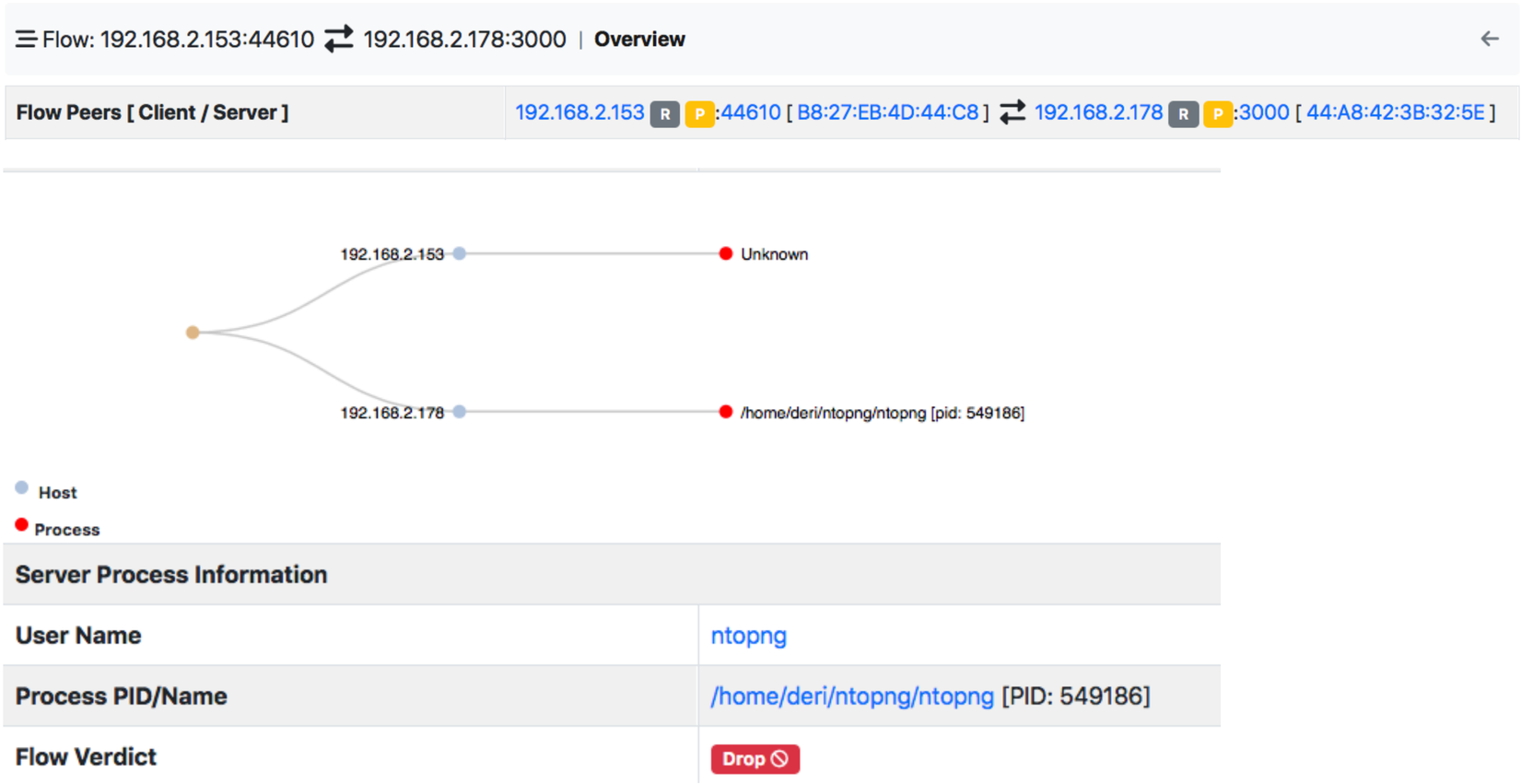
Device	Vanilla Linux Bridge Only	Linux nProbe IPS	Vanilla FreeBSD Bridge Only	FreeBSD nProbe IPS
PC Engines APU2	550 Mbps	600 Mbps	1 Gbps	120 Mbps
Intel E3	10 Gbps / 1.8 Mpps	10 Gbps / 2.4 Mpps		

nProbe as an Agent [1/3]

- nProbe can operate as an agent by exporting flows containing metadata coming from the host it is sitting on.
- Thanks for eBPF (Linux) or Windows APIs, for local flows (i.e. those originate or terminated on the host nProbe runs on) nProbe can export contextual information such as process, memory etc.
- All you need to do is to add --agent-mode to the command line.

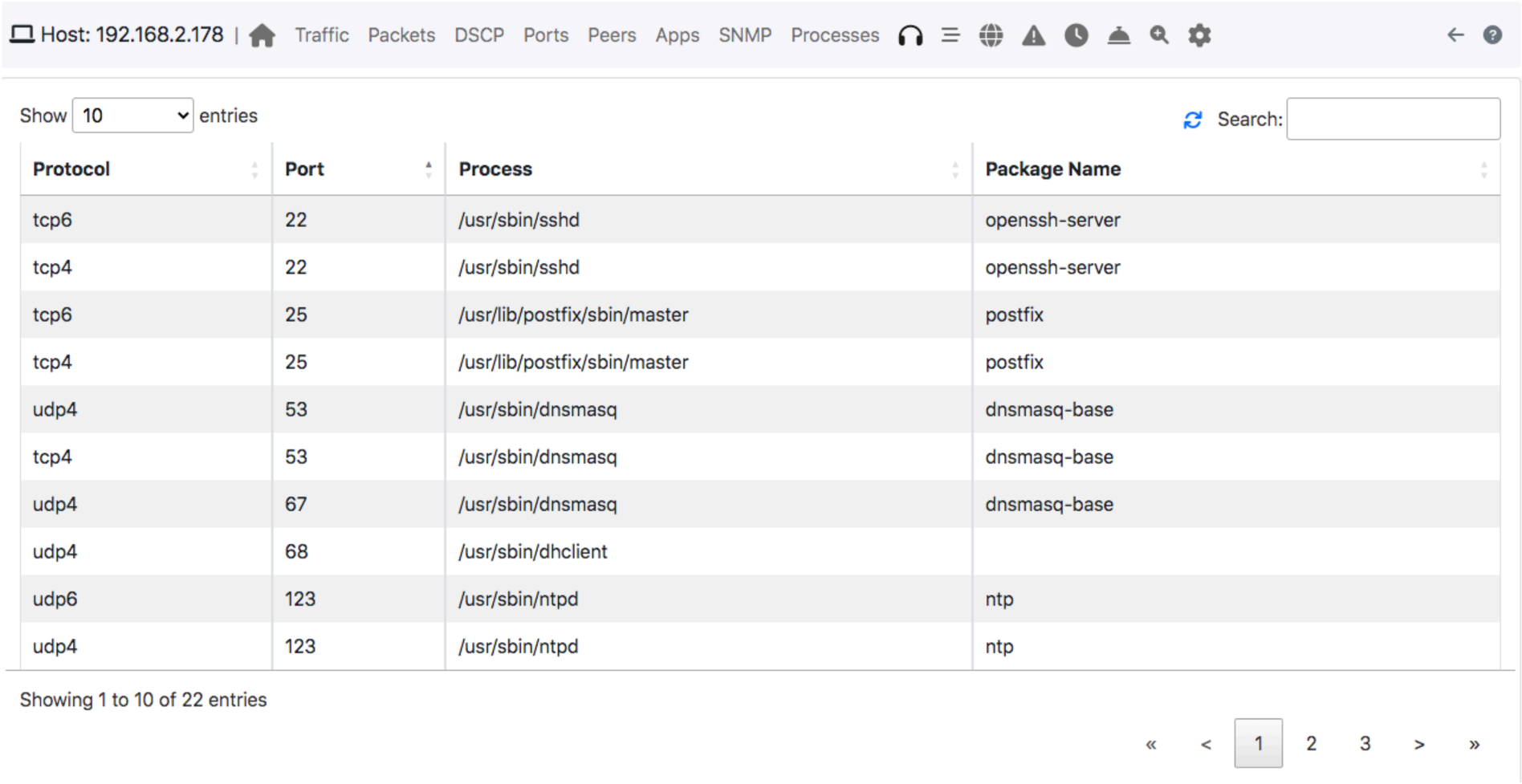
```
# nprobe -i eno1 --zmq tcp://127.0.0.1:1234 --agent-mode
```

nProbe as an Agent [2/3]



Hint: add "-b 2" to display what nProbe reports

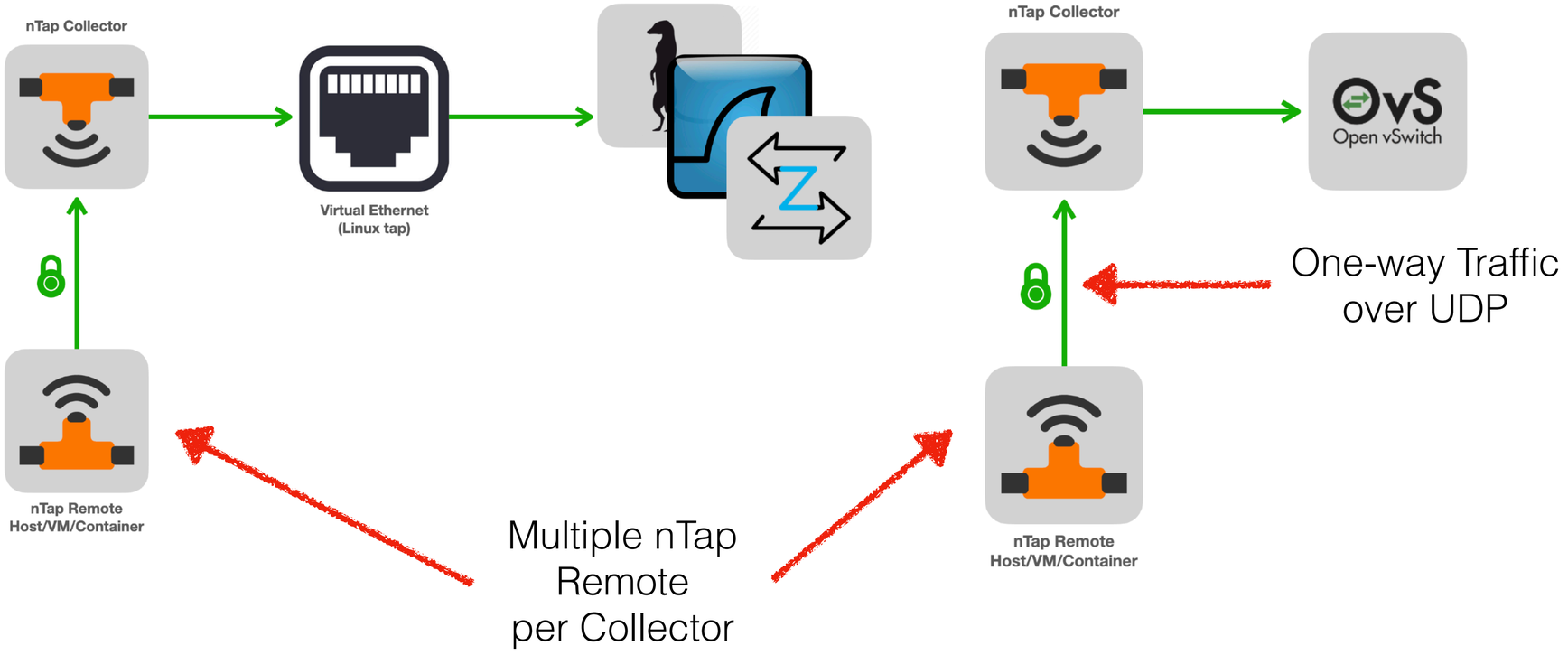
nProbe as an Agent [3/3]



nTap [1/3]

- nTap is a *software* network tap that can be used to provide access to network packets from a remote location.
- Use Cases:
  - Need to collect selected traffic from a remote location and send it in a secure fashion to a monitoring center.
  - Troubleshooting: something is not working as expected and you need to temporarily send traffic for inspection.
  - Distributed topology that does not allow mirror/tap and so you need to grab traffic from remote devices.

nTap [2/3]


nTap Collector
Virtual Ethernet
(Linux tap)
nTap Remote
Host/VM/Container
Multiple nTap
Remote
per Collector
nTap Collector
OvS
Open vSwitch
One-way Traffic
over UDP
nTap Remote
Host/VM/Container

nTap [3/3]

Availability

- nTap Remote: MacOS, Linux, FreeBSD, Windows
- nTap Collector: Linux.

No license required

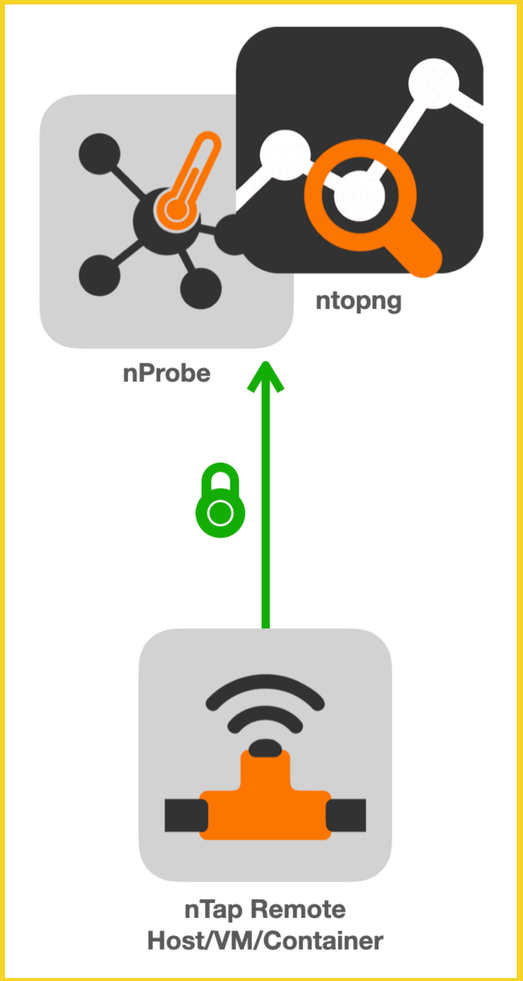
Performance (1 Gbit)

- With encryption: no loss up to 400 Mbps.
- Without encryption: no loss up to 450 Mbps.

Using nTap with nProbe

Native remote traffic collection built-in in nProbe and ntopng

with no nTap license required.

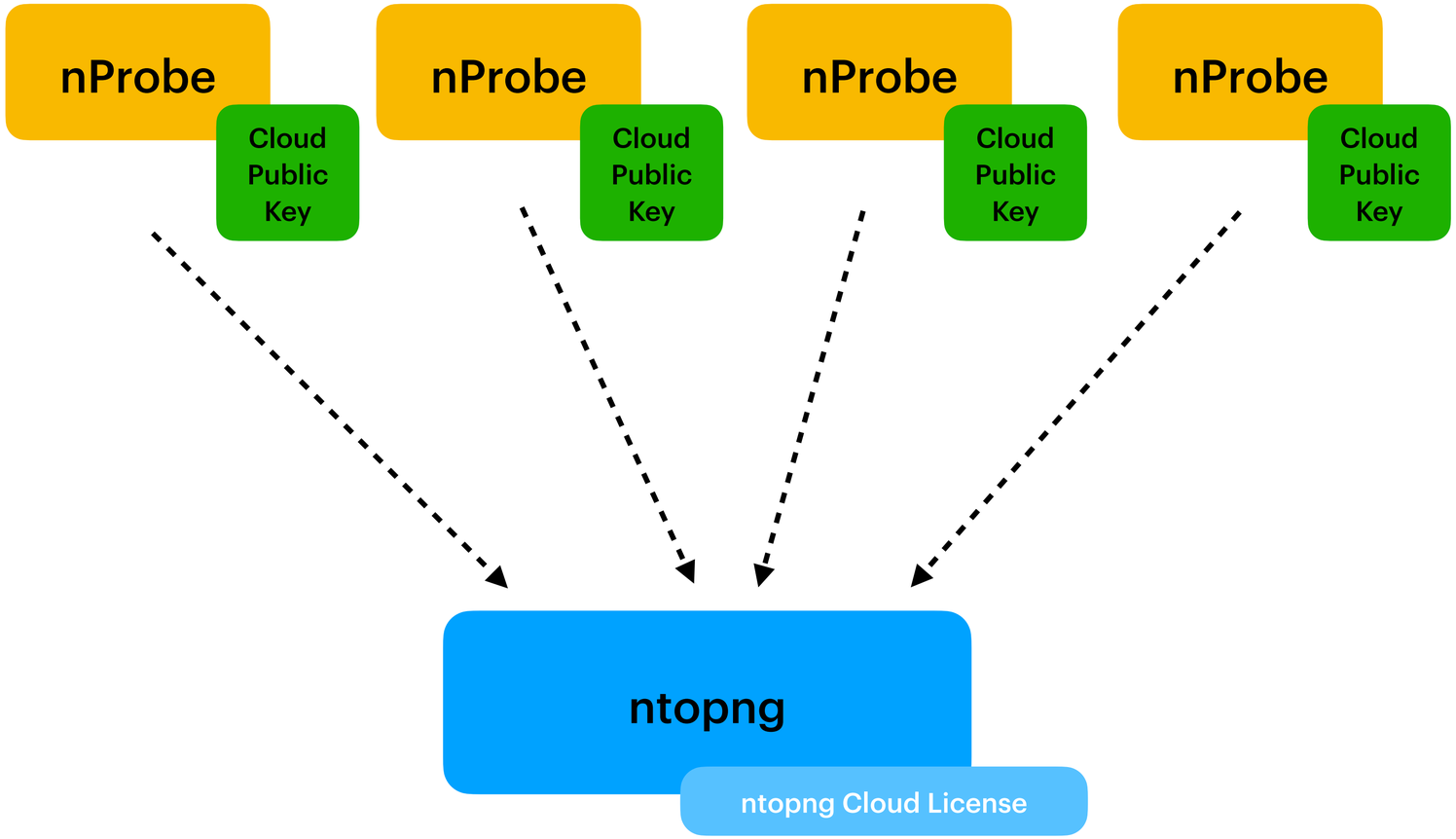


Example:

- [remote host]
  ntap_remote -i eth0 -c 1.2.3.4:5678 **-k hello**
- [local host]
  nprobe -3 5678 -n none **--ntap hello**

What's Next?

- Tomorrow we will present ntopng cloud based on nProbe…



рахмат
danke
謝謝
ngiyabonga
spas
Баярлалаа
mersi
kia ora
barka
welalin
tack
teşekkür ederim
mahalo
tapadh leat
спасибо
faafetai lava
vinaka
спасибі
blagodaram
dank je
misaotra
matondo
paldies
grazzi
thank you
gracias
хвала
nanni
nandri
kiitos
dankie
dhanyavad
mauruuru
köszönöm
asante
manana
obrigada
tenki
bedankt
bayarlalaa
gracie
hvala
dankon
aciu
akun
djiere dieuf
tau
mochchakkeram
murakoze
enkosi
dziękuję
дякую
mamnun
chnorakaloutioun
gratias ago
gracies
sulpay
taiku
go raibh maith agat
chokrane
sobodi
dėkuji
sukriya
kop khun krap
obrigado
mèsi
sagolun
arigatō
takk
dakujem
trugarez
ありがとう
tanemirt
grazie
didi madloba
najis tuke
rahmet
terima kasih
diolch
dhanyavadagalu
shukriya
merce
мерси
kam sah hamnida
rahmat
ευχαριστώ
감사합니다
xièxie
merci