



Network Visibility: The Heart of Modern Monitoring

Jeff Gehlbach • Technical Product Manager • The OpenNMS Group, Inc. • jeffg@opennms.com

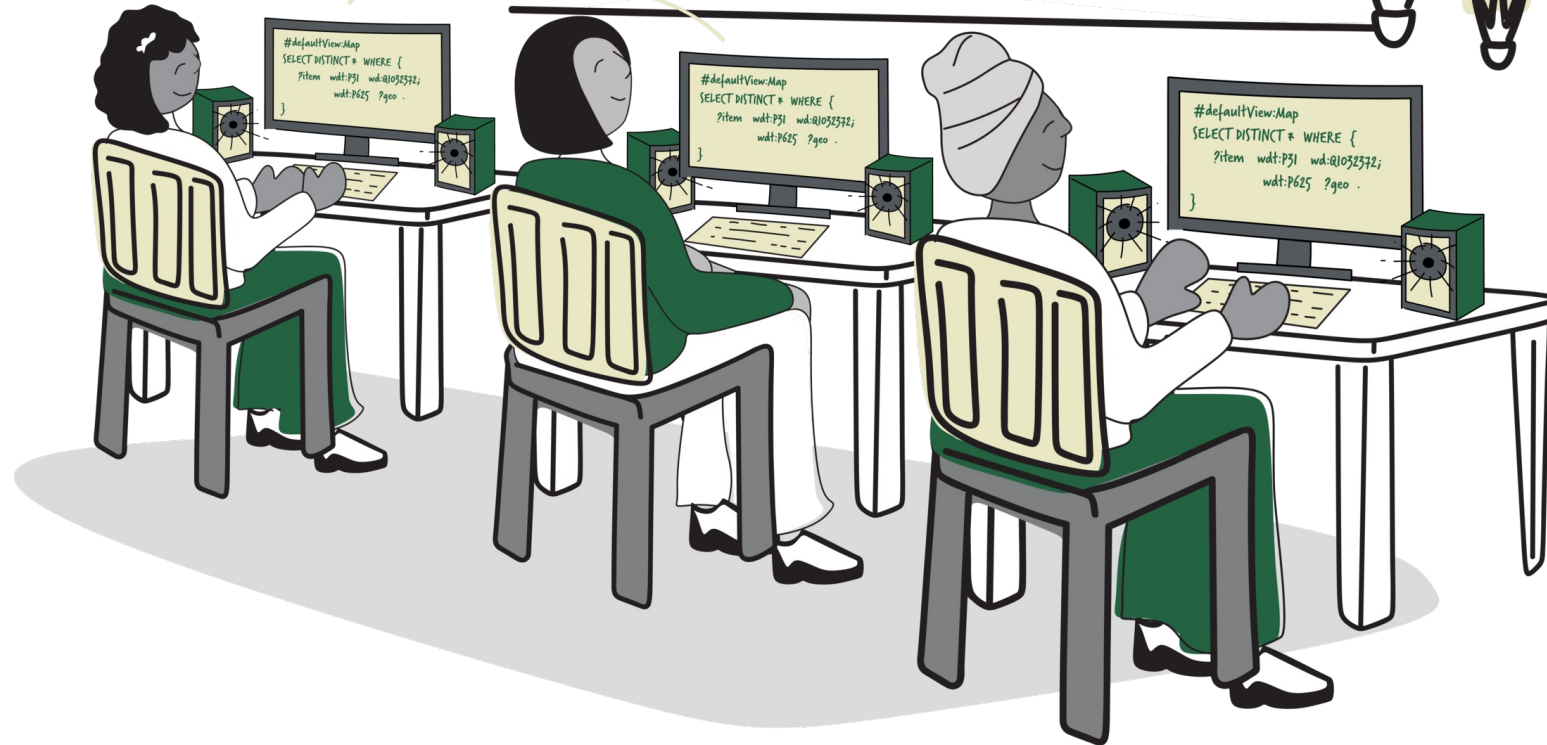
Your Speaker

- Doing monitoring since before it was called that
- Joined OpenNMS Group in 2007
- Attended first SCaLE in 2008
- Became a Product Manager in 2021
- Attached to OpenNMS Horizon ecosystem



"It's nice to see you."

```
#defaultView:Map  
SELECT DISTINCT * WHERE {  
  ?item wdt:P31 wd:Q1032372;  
  wdt:P625 ?geo .  
}
```



Agenda

- The Network Matters
- Monitoring and Network Visibility
- Spotlight on NetFlow (and friends)
- Building a 100% FLOSS Flow Subsystem
- The Future



The Network Matters

Data center



Physical office



Last mile

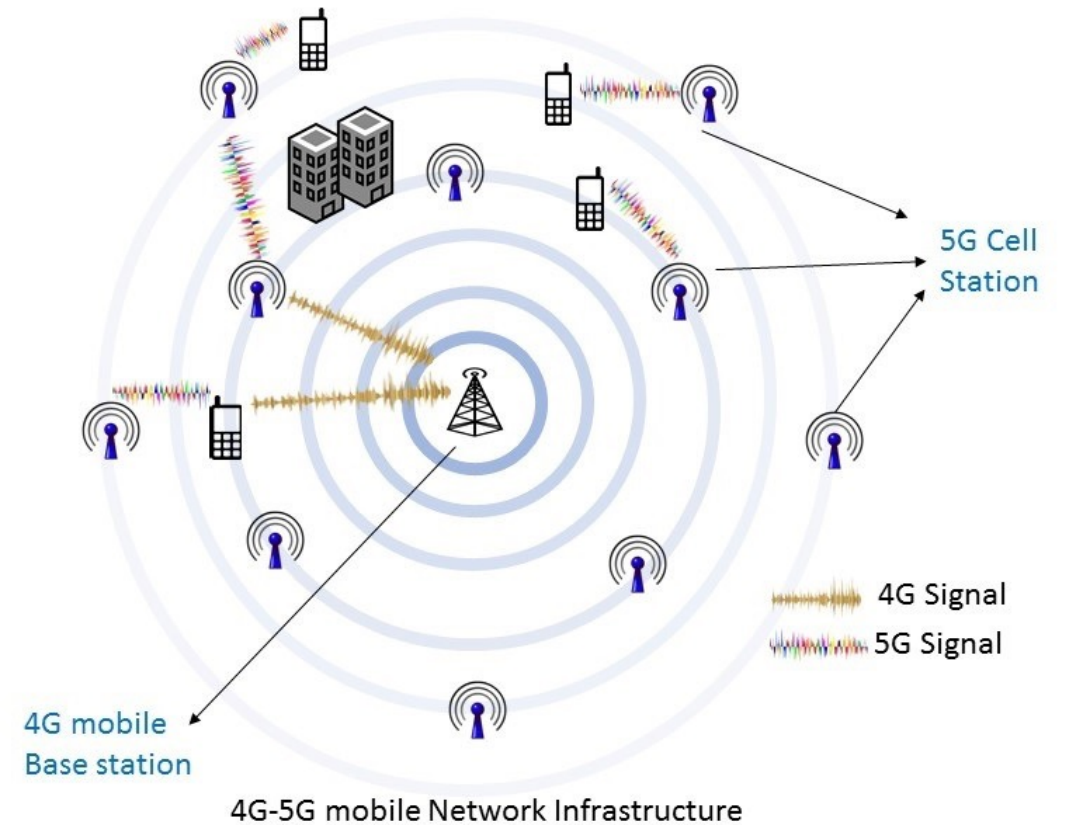
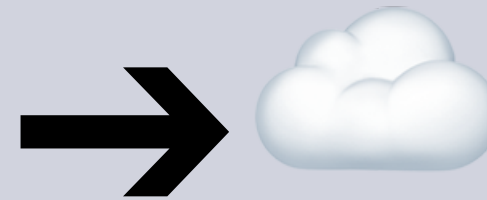
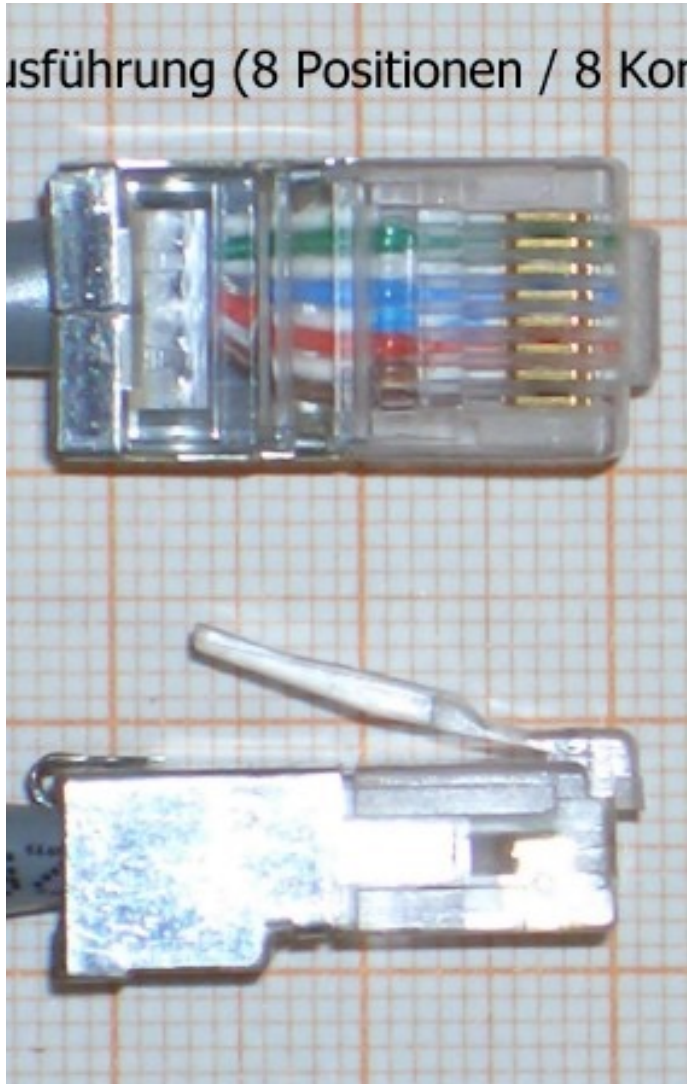
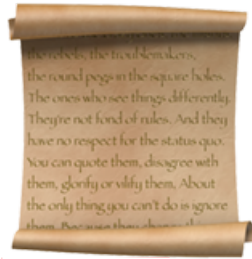


Image: Wikimedia Commons / Goodtiming8871, CC-BY-SA 4.0

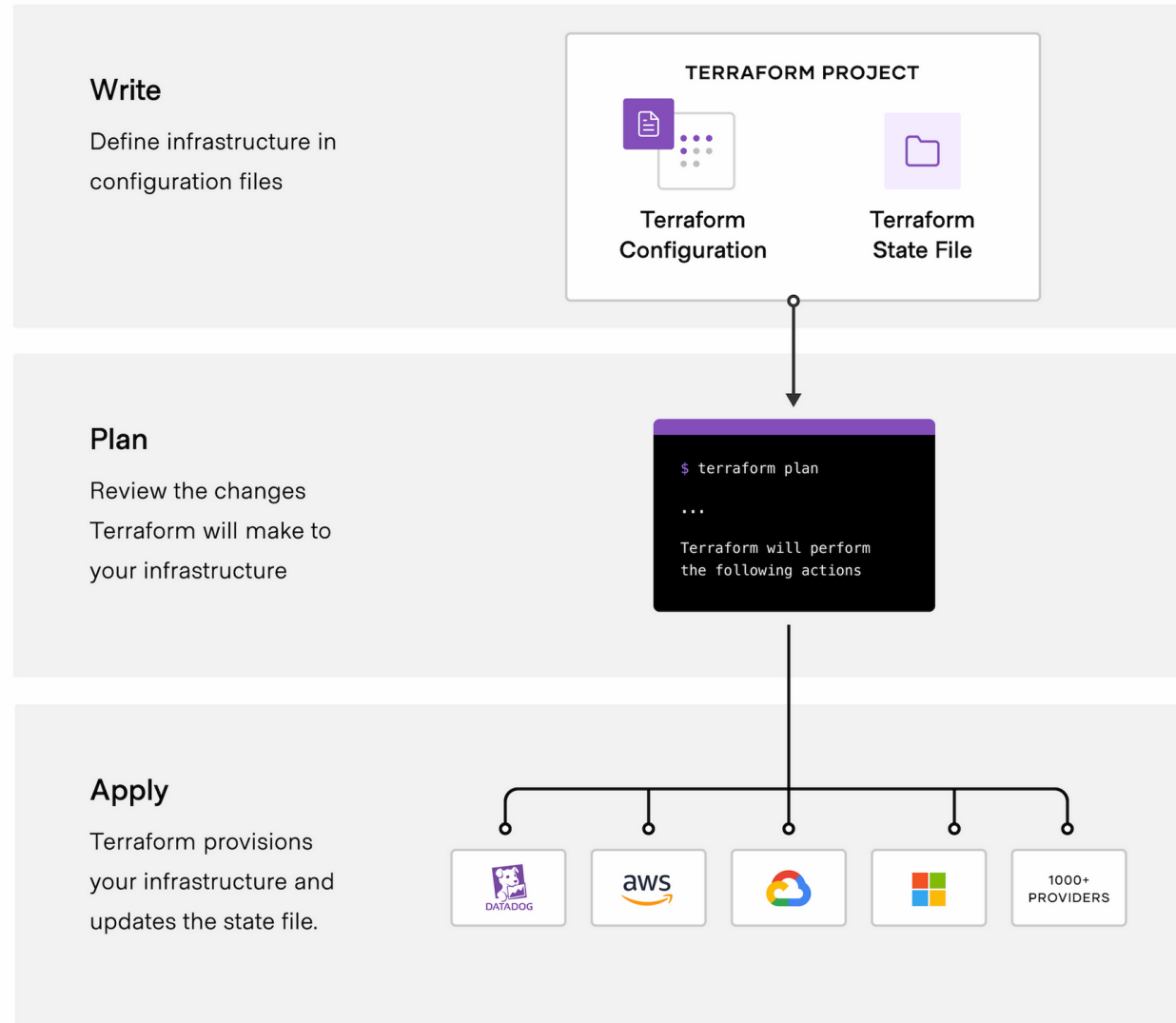
Cloud-native deployment



Cloud-native deployment



Magical as this feels...



...something is moving the bits around.



“Any sufficiently advanced technology is indistinguishable from magic.”

– Arthur C. Clarke

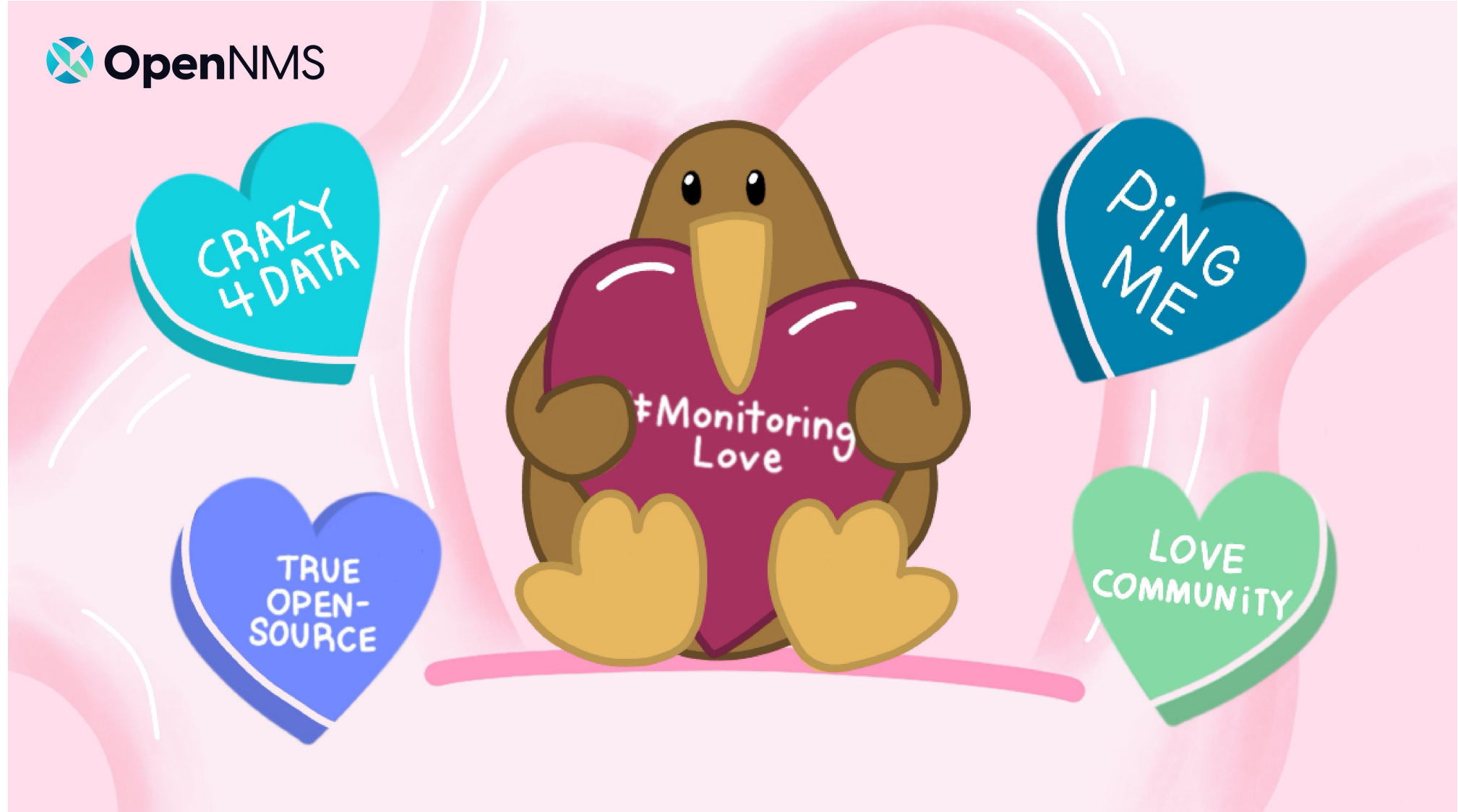
“How’s that going?”

This is why we do monitoring.



Monitoring and Network Visibility

The ♥ of modern monitoring



Visibility is the point of monitoring



FCAPS and FAB, for example

OSI FCAPS

<u>F</u>	Fault management
<u>C</u>	Configuration management
<u>A</u>	Accounting management
<u>P</u>	Performance management
<u>S</u>	Security management

eTOM FAB (TM Forum)

<u>F</u>	Fulfillment
<u>A</u>	Assurance
<u>B</u>	Billing

Fault management



Configuration management

OpenNMS

Nodes

Map

External Requisitions and Thread Pools

File Editor

Logs

Endpoints

Resource Graphs

Device Configs

Secure Credentials Vault

Device Configuration

Devices: 1

2022-07-30T14:48:53-04:002022-07-30T14:51:28-04:00

440ip mtu 1700

440full-duplex

441!

442interface GigabitEthernet0/0/0.15

443interface FastEthernet0/2/6

444bandwidth 100000

445no ip address

446half-duplex

447!

448interface FastEthernet0/2/7

449bandwidth 100000

450no ip address

451half-duplex

452!

453interface GigabitEthernet0/0/0

454description to P2-AMH-US

455bandwidth 1000000

456ip address 172.16.33.14 255.255.255.0

457ip mtu 1700

458full-duplex

459!

460interface GigabitEthernet0/0/0.15

461bandwidth 1000000

440ip mtu 1700

440full-duplex

441!

442interface GigabitEthernet0/0/0.15

443interface FastEthernet0/2/6

444+description to Azure

445bandwidth 100000

446+ip address 62.9.108.42 255.255.255.252

447+full-duplex

448!

449interface FastEthernet0/2/7

450bandwidth 100000

451no ip address

452half-duplex

453!

454interface GigabitEthernet0/0/0

455description to P2-AMH-US

456bandwidth 1000000

457ip address 172.16.33.14 255.255.255.0

458ip mtu 1700

459full-duplex

460!

461interface GigabitEthernet0/0/0.15

462bandwidth 1000000

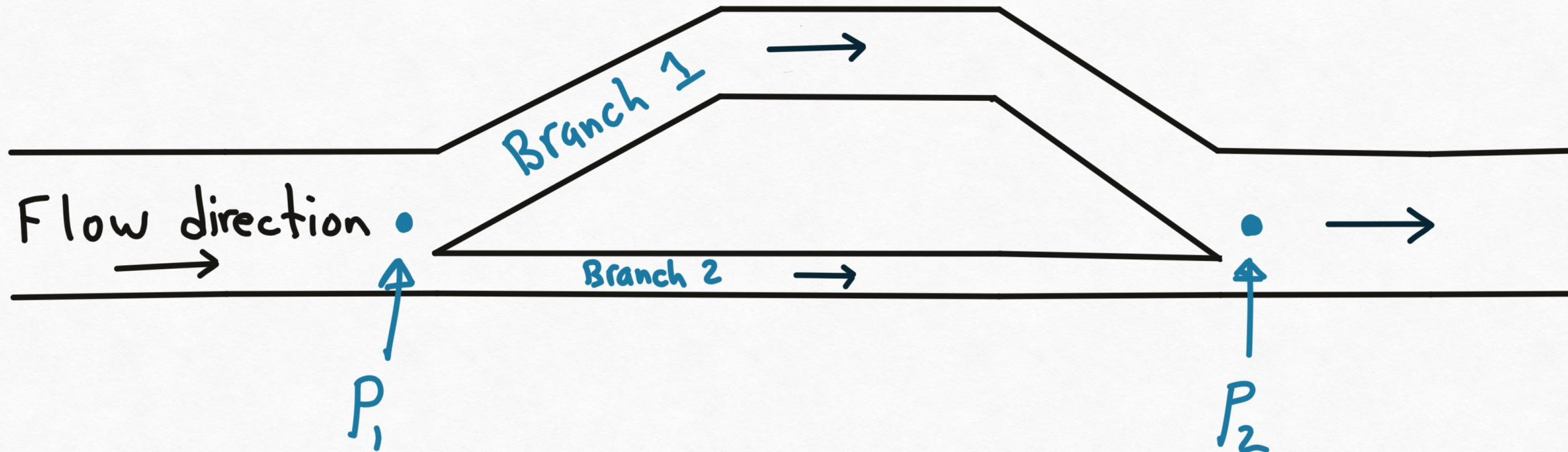
Search device

Schedule DateSchedule Interval

Performance management



Flows



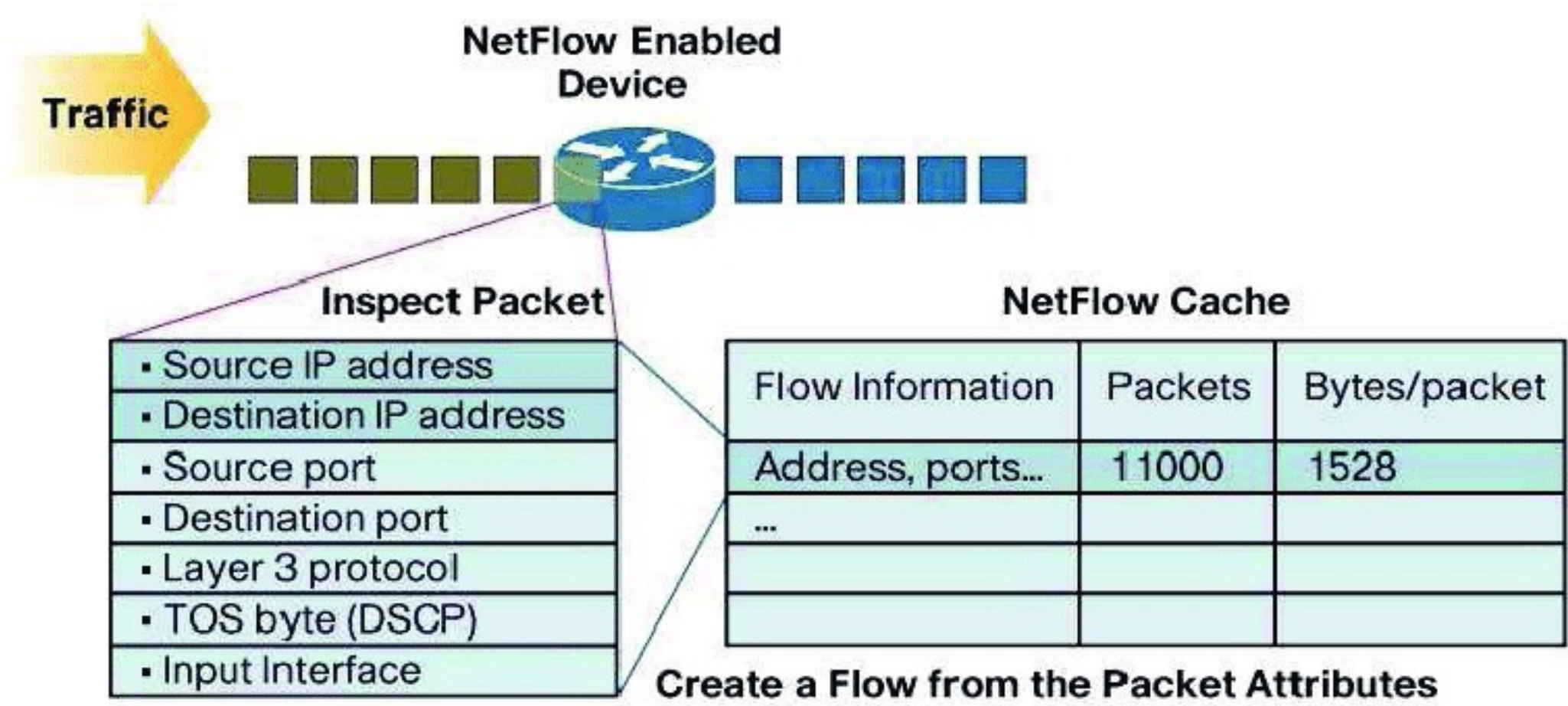


Spotlight on NetFlow and Friends

NetFlow

- Many flow export protocols are available
- Cisco-proprietary NetFlow v5 started the party (almost by accident) in 1996
- Successors and equivalents include
 - NetFlow v9 (proprietary)
 - JFlow (Juniper), NetStream (Huawei, HPE)
 - IPFIX (standards-track)
 - sFlow (standards-track, multi-vendor)
- SDN and SD-WAN may have their own mechanisms, which we will not cover in depth

Anatomy of a flow



Source: Ominike, Akpovi. (2016). Generating Netflow Traces for Network Configurations.

Flow protocols

	NetFlow v5	NetFlow v9 ¹	IPFIX ²
Commercial Release	2002	2004	2013
Standards Based	Proprietary	Proprietary	Standardised (RFC-7011)
Template Elements	18	79 (Cisco: 105)	386
Template Extensibility	Fixed Template	Yes	RFC-5610
Information Elements	No	Single-vendor ID	Multi-vendor ID RFC-7012, RFC-7013
Enterprise Elements	No	No	RFC-5610
Variable Length Fields	No	No	RFC-6313
Structured Data	Yes	Yes	RFC-6313
Transport Protocol	UDP	UDP	TCP, UDP SCTP (RFC-6526)
Flow Security	None	None	Encryption, Integrity

Source: Graham, Mark. (2017). An IPFIX Primer. 10.13140/RG.2.2.33426.35523.

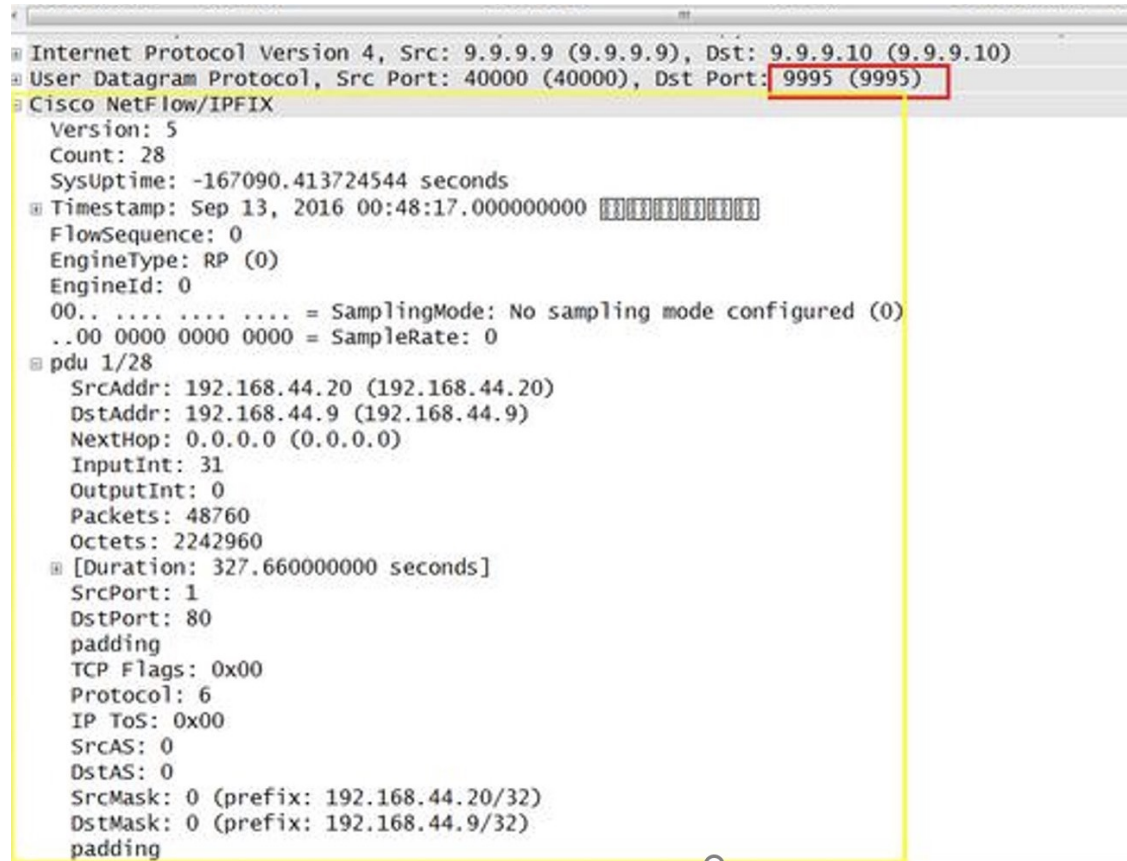
Also, sFlow

- Sampled flows
- Also a streaming-telemetry protocol for generic, non-flow metrics
- First described in 2001 (RFC 3176)

NetFlow v5 export packet

Dissected example

Important fields for our purposes



```
Internet Protocol Version 4, Src: 9.9.9.9 (9.9.9.9), Dst: 9.9.9.10 (9.9.9.10)
User Datagram Protocol, Src Port: 40000 (40000), Dst Port: 9995 (9995)
Cisco NetFlow/IPFIX
  Version: 5
  Count: 28
  SysUptime: -167090.413724544 seconds
  Timestamp: Sep 13, 2016 00:48:17.000000000
  FlowSequence: 0
  EngineType: RP (0)
  EngineId: 0
  00.. .... = SamplingMode: No sampling mode configured (0)
  ..00 0000 0000 0000 = SampleRate: 0
  pdu 1/28
    SrcAddr: 192.168.44.20 (192.168.44.20)
    DstAddr: 192.168.44.9 (192.168.44.9)
    NextHop: 0.0.0.0 (0.0.0.0)
    InputInt: 31
    OutputInt: 0
    Packets: 48760
    Octets: 2242960
  [Duration: 327.660000000 seconds]
    SrcPort: 1
    DstPort: 80
    padding
    TCP Flags: 0x00
    Protocol: 6
    IP ToS: 0x00
    SrcAS: 0
    DstAS: 0
    SrcMask: 0 (prefix: 192.168.44.20/32)
    DstMask: 0 (prefix: 192.168.44.9/32)
    padding
```

Image: Jesse White

NetFlow v5 export packet

Dissected example

```
▣ pdu 1/28
  SrcAddr: 192.168.44.20 (192.168.44.20)
  DstAddr: 192.168.44.9 (192.168.44.9)
  NextHop: 0.0.0.0 (0.0.0.0)
  InputInt: 31
  OutputInt: 0
```

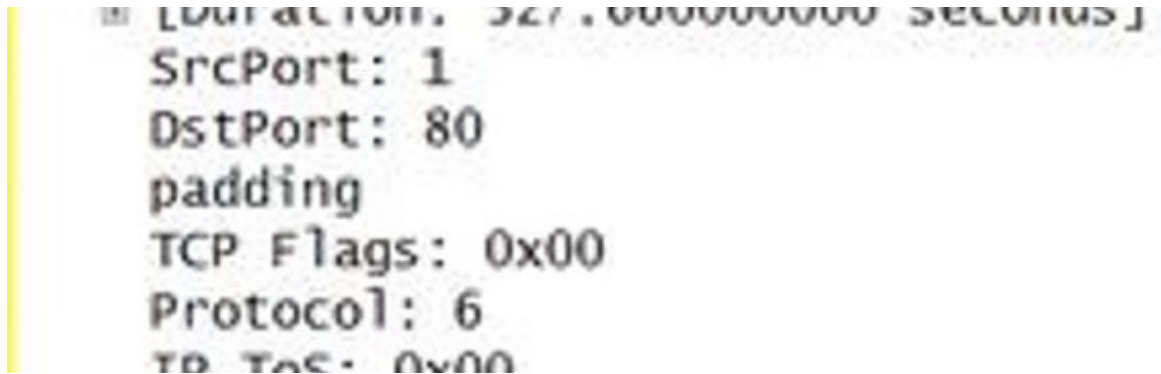
Important fields for our purposes

- *Source IP address*
- *Destination IP address*
- *Input ifIndex*

Image: Jesse White

NetFlow v5 export packet

Dissected example



Important fields for our purposes

- Source IP address
- Destination IP address
- Input ifIndex
- *Source port*
- *Destination port*
- *IP Protocol*

Image: Jesse White

NetFlow v5 export packet

Dissected example

```
Packets: 48760  
Octets: 2242960  
[Duration: 327.660000000 seconds]  
SrcPort: 1
```

```
DstMask: 0 (preFix: 192.168.44.9/32)  
padding
```

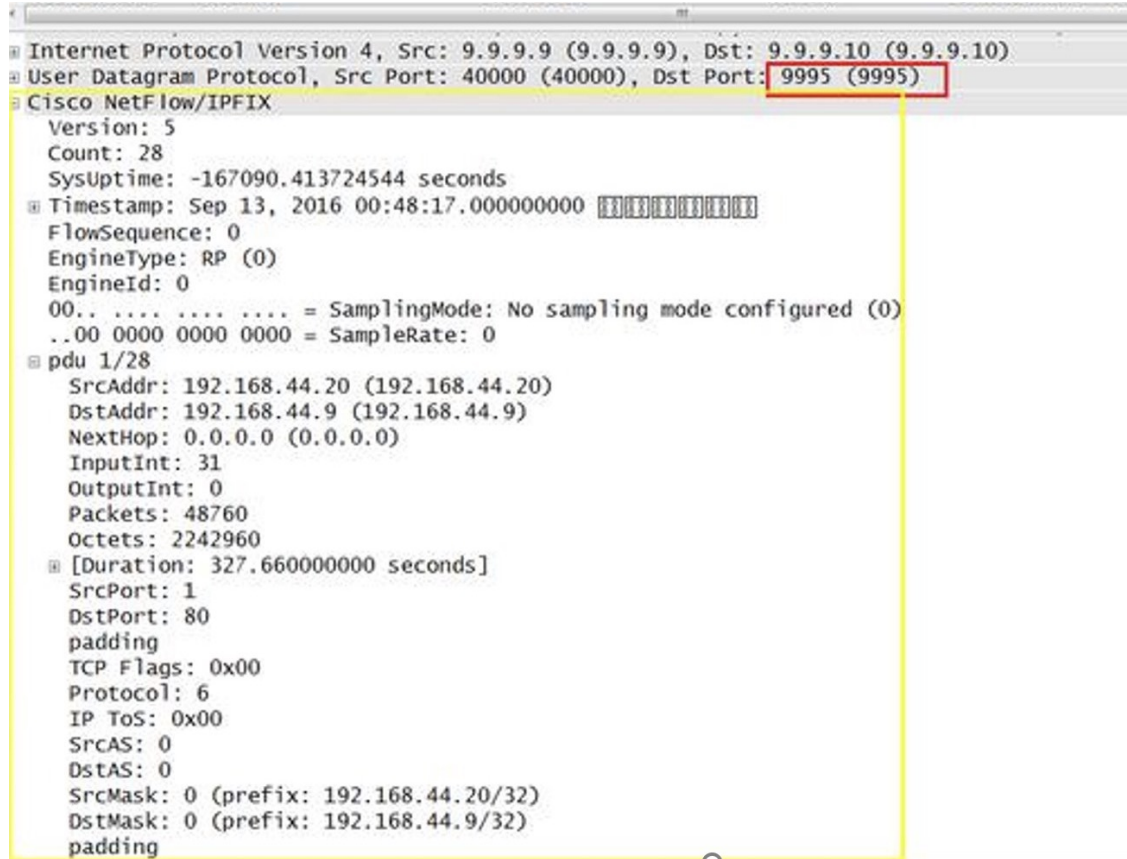
Important fields for our purposes

- Source IP address
- Destination IP address
- Input ifIndex
- Source port
- Destination port
- IP Protocol
- *Octets*
- *Duration*
- *Padding (ingress vs. egress)*

Image: Jesse White

NetFlow v5 export packet

Dissected example



Important fields for our purposes

- Source IP address
- Destination IP address
- Input ifIndex
- Source port
- Destination port
- IP Protocol
- Octets
- Duration
- Padding (ingress vs. egress)

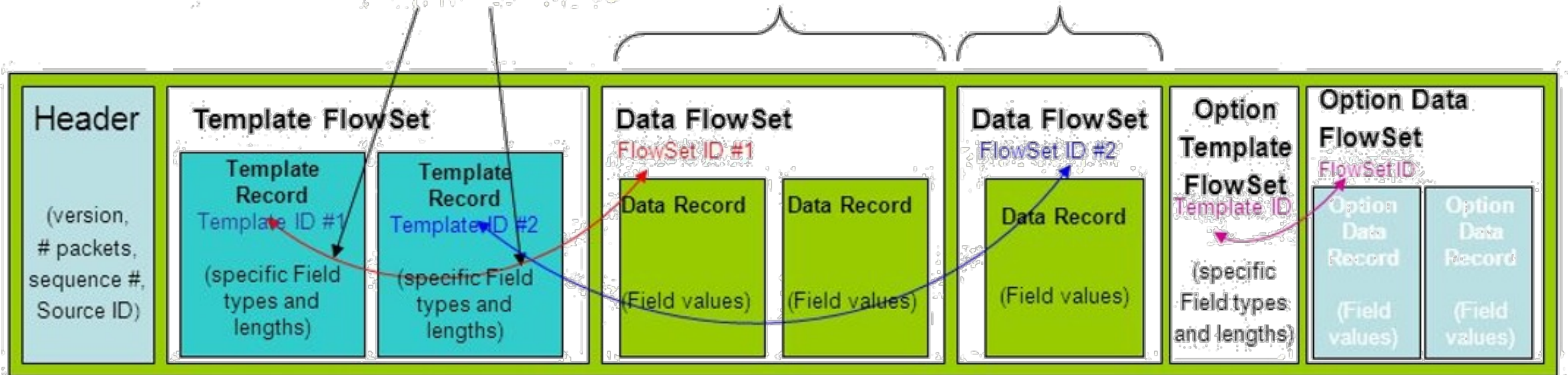
Image: Jesse White

NetFlow v9 export packet

To support technologies such as MPLS or Multicast, this export format can be leveraged to easily *insert new fields*

Flows from
Interface A

Flows from
Interface B



Source: Cisco Systems

SNMP vs. sFlow vs. NetFlow

Protocol	SNMP	sFlow	NetFlow (v9)	IPFIX
Type of information	MIB counter	Partial packets chosen by sampling	Flow	
Amount of data	Small	Large (depending on sampling rate)	Between SNMP and sFlow (depending on sampling rate and flow creation conditions)	
Collectable information	Amount of data across interface	Data from data-link layer (containing packet header and data of partial packet payload)	Data from data-link layer to transport layer	
			Data other than the above is collected by vendor extensions	
Status of standardization	RFC3411, RFC3418, etc. (standard)	RFC3176 (informational by InMon)	RFC3954 (informational by Cisco)	Stage immediately before publication as an RFC (standard)

Source: Irino, Katayama, Chaki. (2007). Flow-based Network Measurement— NetFlow & IPFIX; NTT Technical Review



Building a 100% FLOSS Flow Subsystem

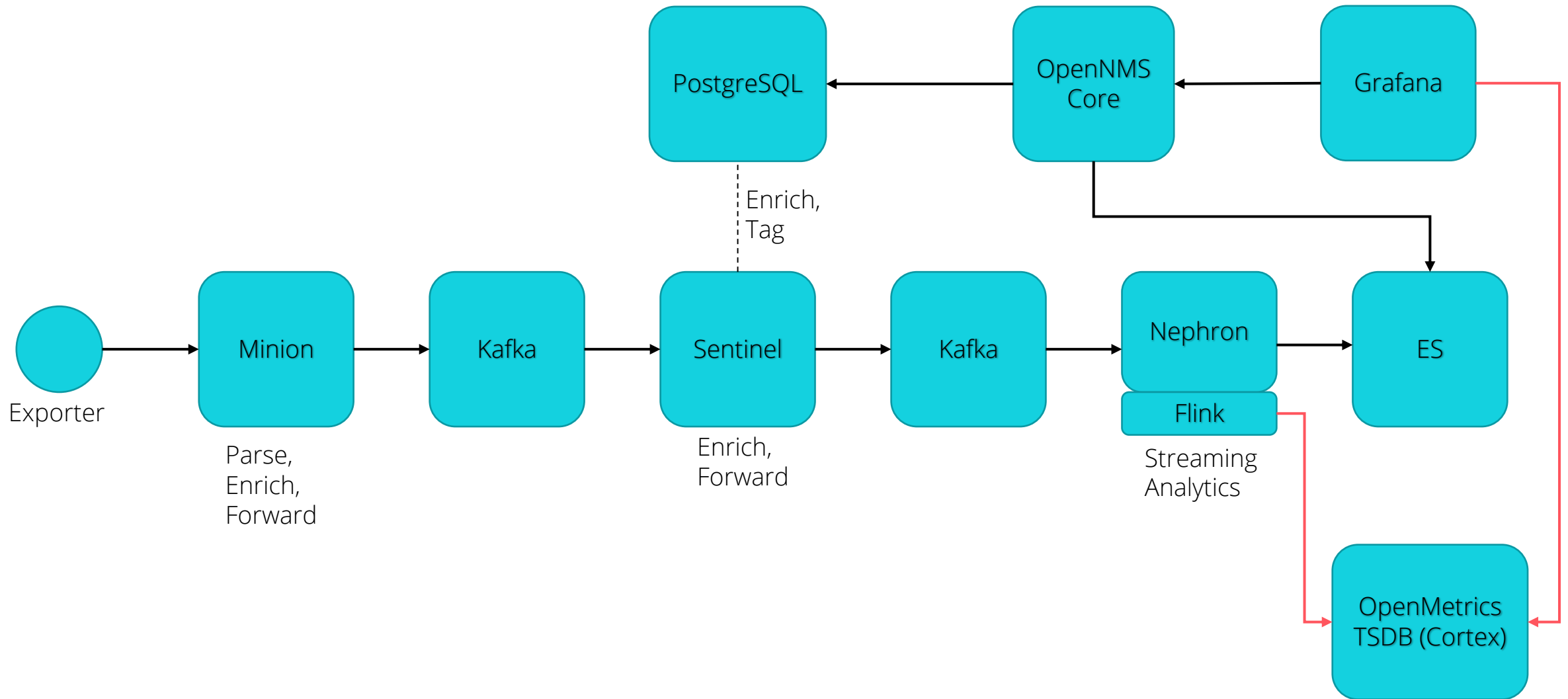
OpenNMS provides

- A platform to collect, persist, and visualize flows, with support for:
 - NetFlow v5 and v9
 - IPFIX
 - sFlow
- Inventory enrichment (mapping of flows to OpenNMS nodes)
- Application classification
- Horizontal scale (proven out at 300K+ flows/sec)
- Flow thresholds
- Enterprise reporting (push reports via PDF)
- Top K stats by interface, application, host, conversation, w/QOS

Flow visualization



OpenNMS Horizon flow pipeline (full scale, self-hosted)



Scaling flows in the real world

- 800 routers generating flows
- 1 interface on most routers
 - 2-4 interfaces on some
- 6 million+ flows per interface per hour

Name ↑	Status	Document Count	Data
netflow-2019-12-24-16	Deleted / Closed	128.3m	71.3 GB
netflow-2019-12-24-17	Green	139.6m	77.4 GB
netflow-2019-12-24-18	Green	145.2m	80.3 GB
netflow-2019-12-24-19	Green	133.6m	74.0 GB
netflow-2019-12-24-20	Green	122.4m	68.0 GB
netflow-2019-12-24-21	Green	121.7m	67.6 GB
netflow-2019-12-24-22	Green	129.2m	72.0 GB
netflow-2019-12-24-23	Green	116.9m	65.3 GB
netflow-2019-12-25-00	Green	117.5m	65.5 GB
netflow-2019-12-25-01	Green	133m	73.6 GB
netflow-2019-12-25-02	Green	151.8m	83.2 GB
netflow-2019-12-25-03	Green	154.3m	84.0 GB
netflow-2019-12-25-04	Green	146.3m	79.7 GB
netflow-2019-12-25-05	Green	153.3m	83.8 GB
netflow-2019-12-25-06	Green	164.7m	90.0 GB
netflow-2019-12-25-07	Green	165.8m	90.7 GB
netflow-2019-12-25-08	Green	153.6m	84.0 GB

Image: Jesse White

Challenges with just-in-time flow statistics

- Elasticsearch query performance
- Long dashboard render times when zoomed to long time periods
- The stats:
 - Top N over 4 billion documents
 - 120000 unique hosts
 - 6000 unique applications

Challenges with streaming flow analysis

- Time-domain problems
 - Elements (flow data) must be grouped by time window
 - Elements may arrive early, on time, or late
 - Even with perfect clock sync, processing time may introduce time lag
- Apache Beam tries to help with this
 - It's still a big engineering effort to get it right



The Future

The future of flows in OpenNMS

- Integrate with SDN and SD-WAN platforms that support IPFIX, or some proprietary flow-export mechanism
 - Velocloud integration now WIP
- Reduce complexity of the flow architecture
 - Eliminate the need for a Flink cluster to run Nephron
 - Help improve Cortex support for high-cardinality data
 - Eliminate need for Elasticsearch, ideally
- Easier deployment with Kubernetes operator

Be part of the future of OpenNMS!

[Your ideas here] – come hack with us

opennms.discourse.group



chat.opennms.com



github.com/opennms



issues.opennms.org





Conclusion

Thank you!