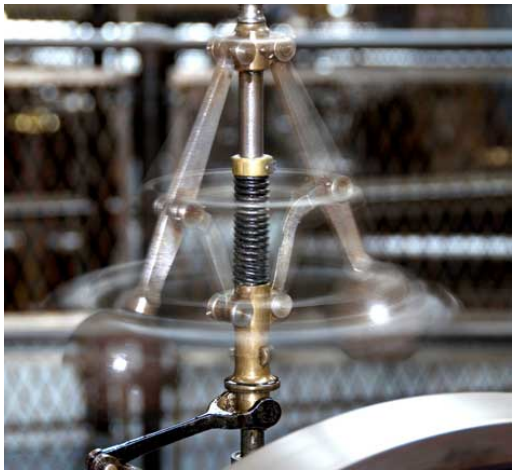
inMon sFlow

sFlow®

Data network visibility and control



"You can't control what you can't measure"
Tom DeMarco

Reason 1: Widely supported industry standard

Reason 1: Widely supported industry standard

inMon sFlow

hp

BROCADE

IBM

DELL

H3C ITolP Solutions Expert

Juniper NETWORKS

BLADE NETWORK TECHNOLOGIES

3COM

ARISTA

FORCE10

HITACHI Inspire the Next

NEC Empowered by Innovation

AlaxalA

extreme networks GO BEYOND

Alcatel·Lucent

D-Link Building Networks for People

Allied Telesis

enterasys Secure Networks

FORTINET

VYATTA Open Networking

CITRIX | XenServer

Xen Cloud Platform

Reason 1: Widely supported industry standard

inMon sFlow

hp

BROCADE

IBM

DELL

H3C IToIP Solutions Expert

Juniper NETWORKS

BLADE NETWORK TECHNOLOGIES

3COM

ARISTA

FORCE10

HITACHI Inspire the Next

NEC Empowered by Innovation

AlaxalA

extreme networks GO BEYOND

Alcatel·Lucent

D-Link Building Networks for People

Allied Telesis

enterasys Secure Networks

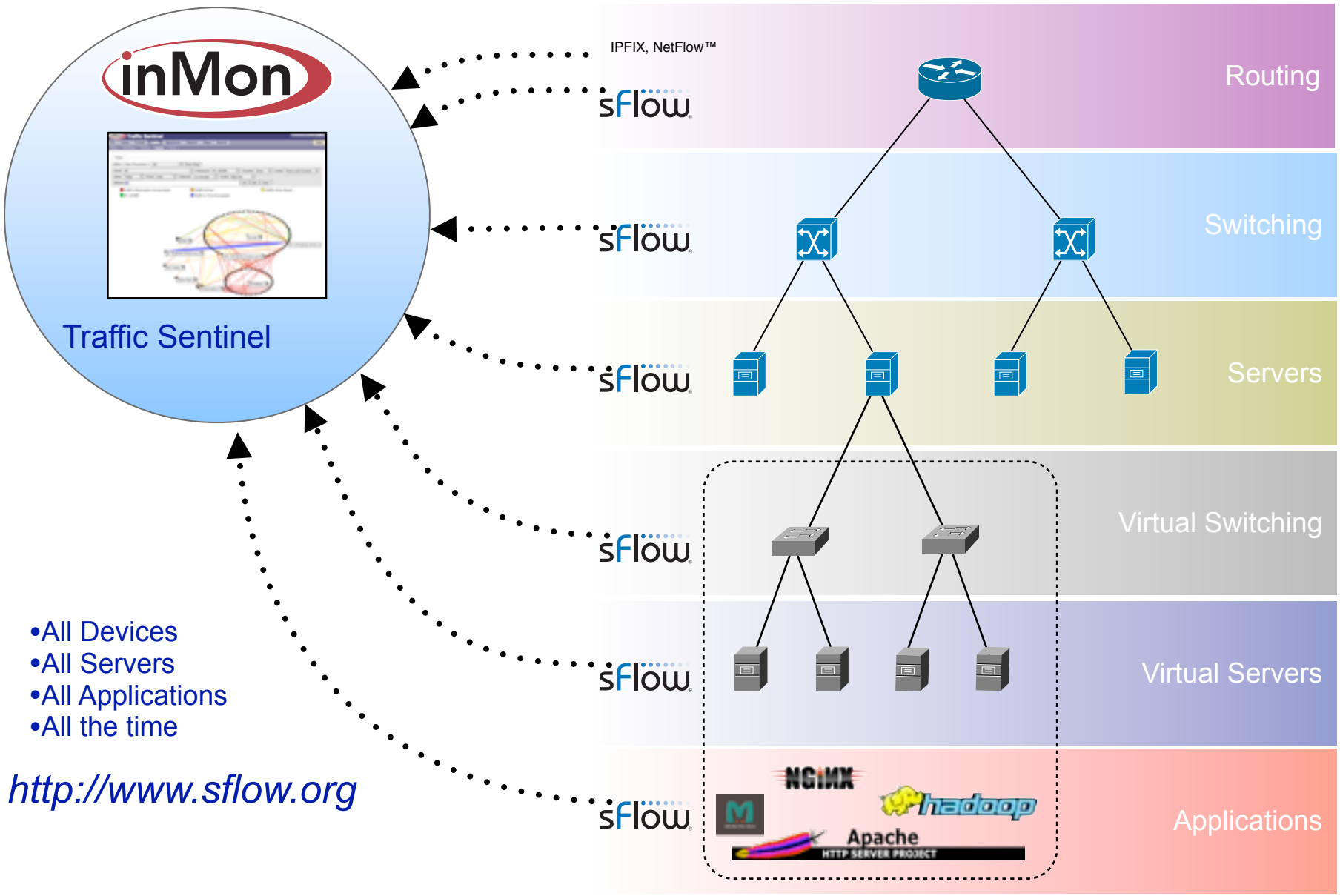
FORTINET

VYATTA Open Networking

CITRIX | XenServer

Xen Cloud Platform

Reason 2: Comprehensive



- All Devices
- All Servers
- All Applications
- All the time

http://www.sflow.org

sFlow Overview: replaces counter polling

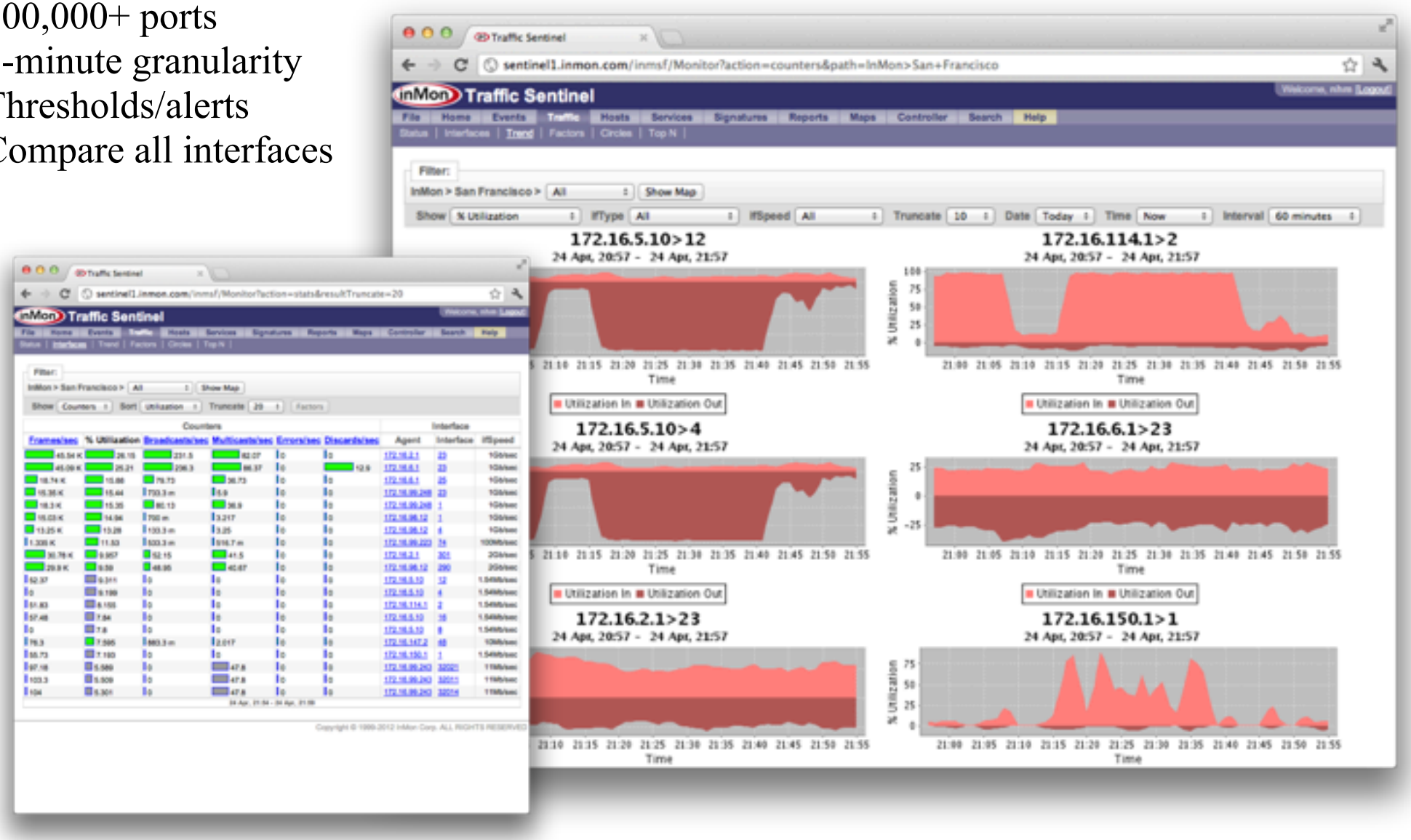
"De-synchronized, Parallel Push"

- sFlow agent automatically pushes full set of SNMP ifTable counters[1]
- Compared to SNMP polling, counter push results in 10-20x fewer packets on network, reduces CPU load on switch and on network management software (XDR is easier to encode/decode than SNMP)
- Single sFlow collector can easily monitor 200,000 switch ports with 1 minute granularity. SNMP polling with 5 minute granularity requires 5-10 collectors.

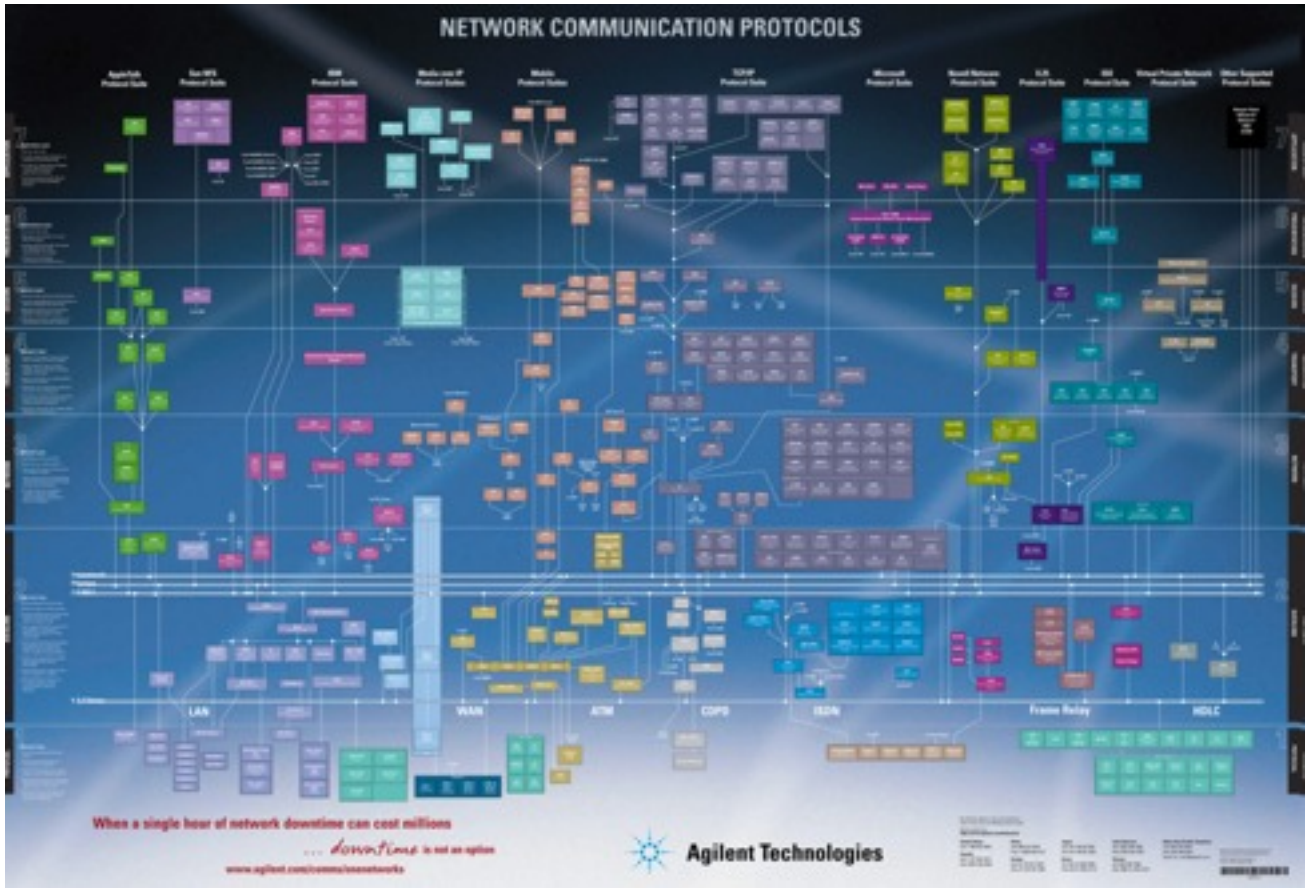
1. ifIndex, ifType, ifType, ifSpeed, ifDirection, ifAdminStatus, ifOperStatus, ifInOctets, ifInUcastPkts, ifInMulticastPkts, ifInBroadcastPkts, ifInDiscards, ifInErrors, ifInUnknownProtos, ifOutOctets, ifOutUcastPkts, ifOutMulticastPkts, ifOutBroadcastPkts, ifOutDiscards, ifOutErrors, ifPromiscuousMode

Traffic Sentinel: Interface counters

- 200,000+ ports
- 1-minute granularity
- Thresholds/alerts
- Compare all interfaces



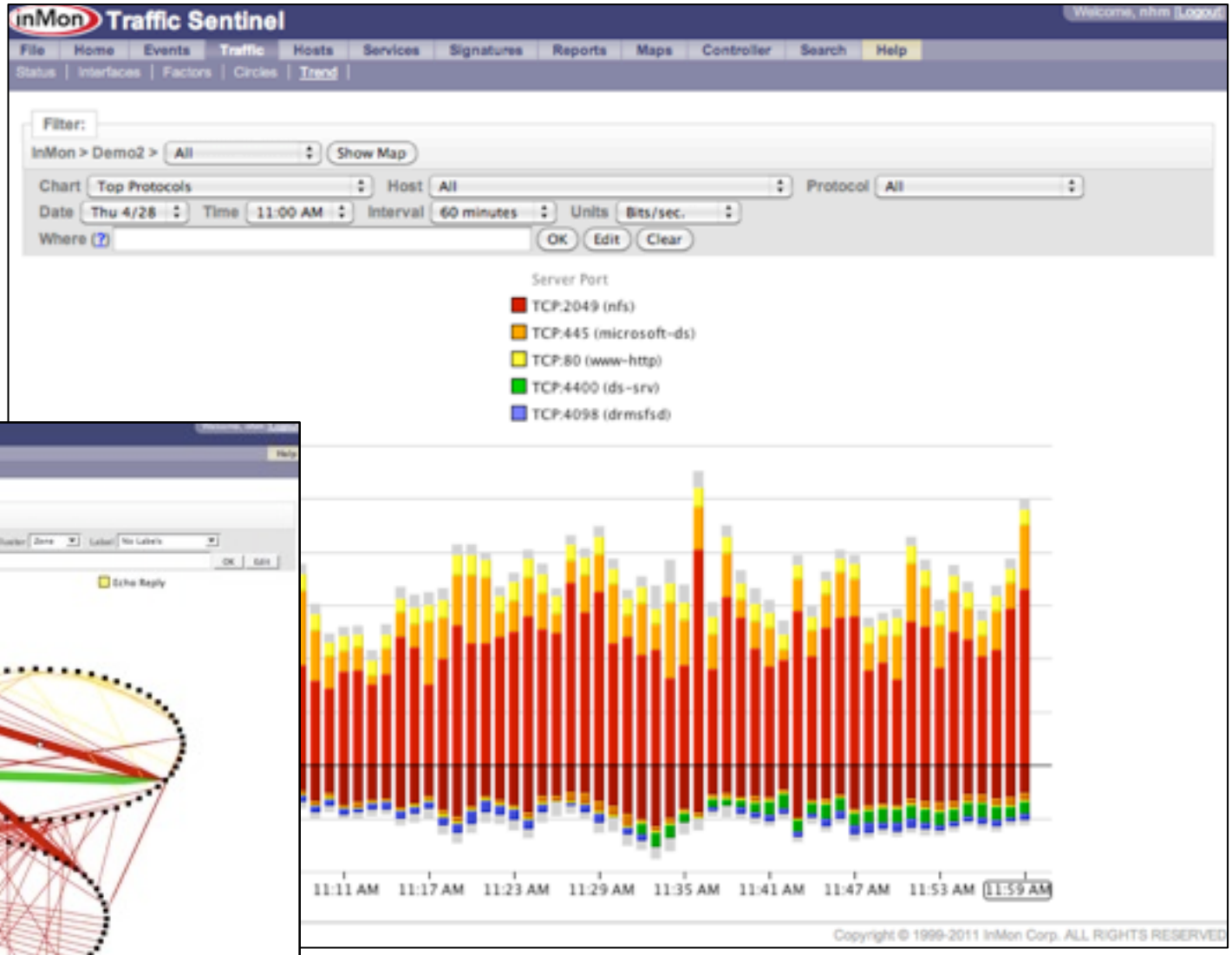
sFlow Overview: monitors all protocols

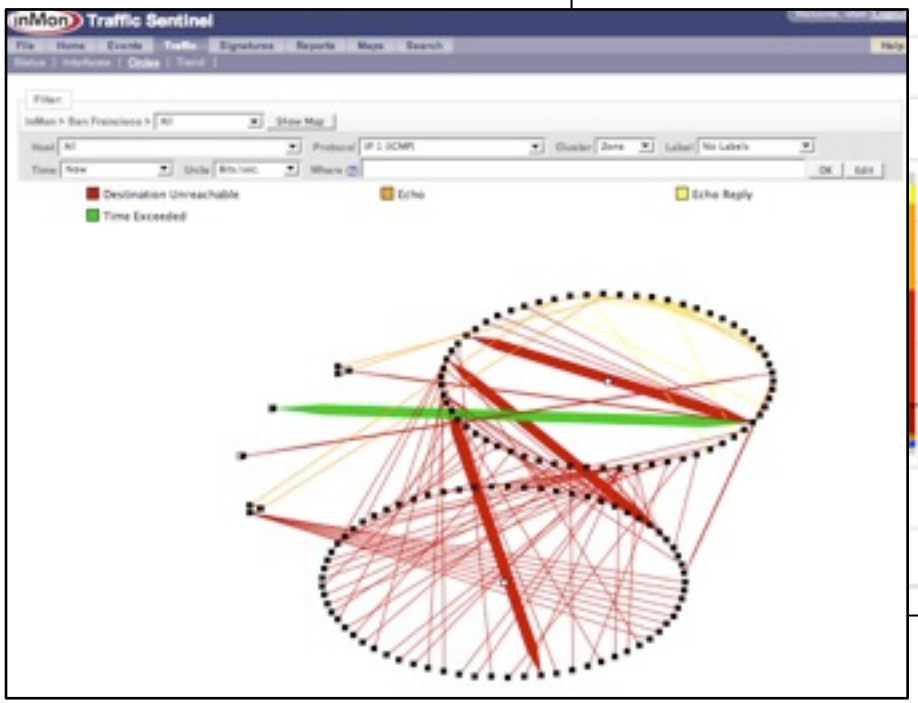


- Simple agents: packet headers sent to sFlow collector for decoding.
- Easier to add decodes to central collector than to every device in a multi-vendor network (e.g. IPv6, FCoE etc.)
- Captures complex layering (e.g. MAC/VLAN/MPLS/IPv4/IPv6): critical for tracing packet paths through network.

Traffic Sentinel: Traffic Breakdown

- MAC, VLAN, IP, IPv6, TCP, UDP, MPLS, TRILL, RTP etc. (over 100 fields)
- 1-minute granularity
- Thresholds/alerts
- Automatic de-duplication
- Subnet rollups





TRILL Fabrics

```
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|               Outer Destination MAC Address                   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Outer Destination MAC Address | Outer Source MAC Address      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                   Outer Source MAC Address                    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Ethertype = IEEE 802.1Q       | UP  |C|   Outer VID           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Ethertype = TRILL             | V | Hop Limit |M|  Reserved   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Egress RBridge Address       |  Ingress RBridge Address      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|               Inner Destination MAC Address                   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Inner Destination MAC Address | InnerSource MAC Address       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                    Inner Source MAC Address                   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Ethertype = IEEE 802.1Q       | UP  |C|   Inner VID           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
|                 Original Ethernet Payload                     |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                           New FCS                             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

- sFlow on existing Ethernet switches captures the following TRILL fields
  - TRILL RBridge Addresses
  - Forwarding path
  - Hop count
  - Broadcast bit
- As well as inner/outer MAC addresses and encapsulated TCP/IP etc. data
- sFlow monitoring provides information about path utilization, applications using a path etc. Critical for load balancing and troubleshooting TRILL deployments
-

TRILL Fabrics

Traffic Sentinel

10.0.0.111/inmsf/Monitor?action=trend

inMon Traffic Sentinel

Welcome, pp [Logout]

File | Home | Events | Traffic | Hosts | Services | Signatures | Reports | Maps | Controller | Search | Help

Status | Interfaces | Trend | Factors | Circles | Top N |

Filter:

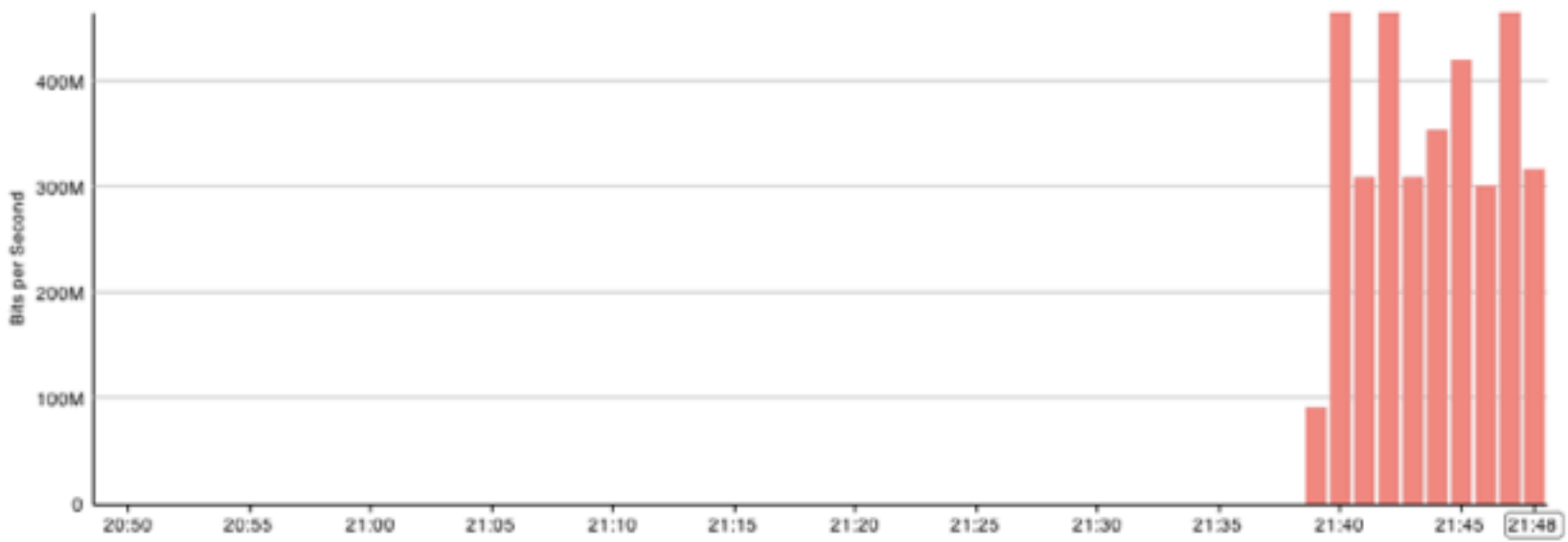
InMon > HQ > All | Show Map

Chart TRILL Flows | Edit Columns | Host All | Protocol All

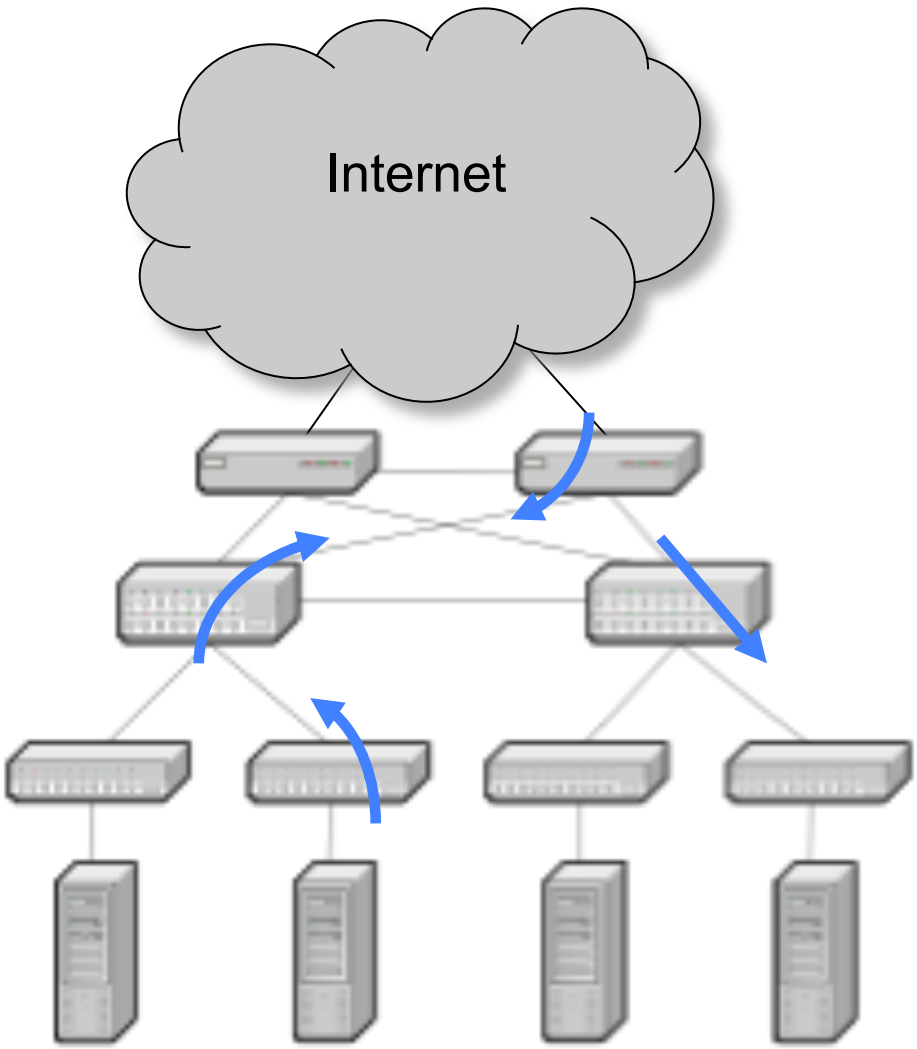
Date Today | Time Now | Interval 60 minutes | Units Bits/sec.

Where (?) | OK | Edit | Clear

	TRILL RBridge Ingress	TRILL RBridge Egress	MAC Source	MAC Destination	Source Address	Destination Address	IP DSCP	Value
	2	1	000533540CCC	0005335C8757	192.85.1.2	192.0.0.1	be (0)	315.97M



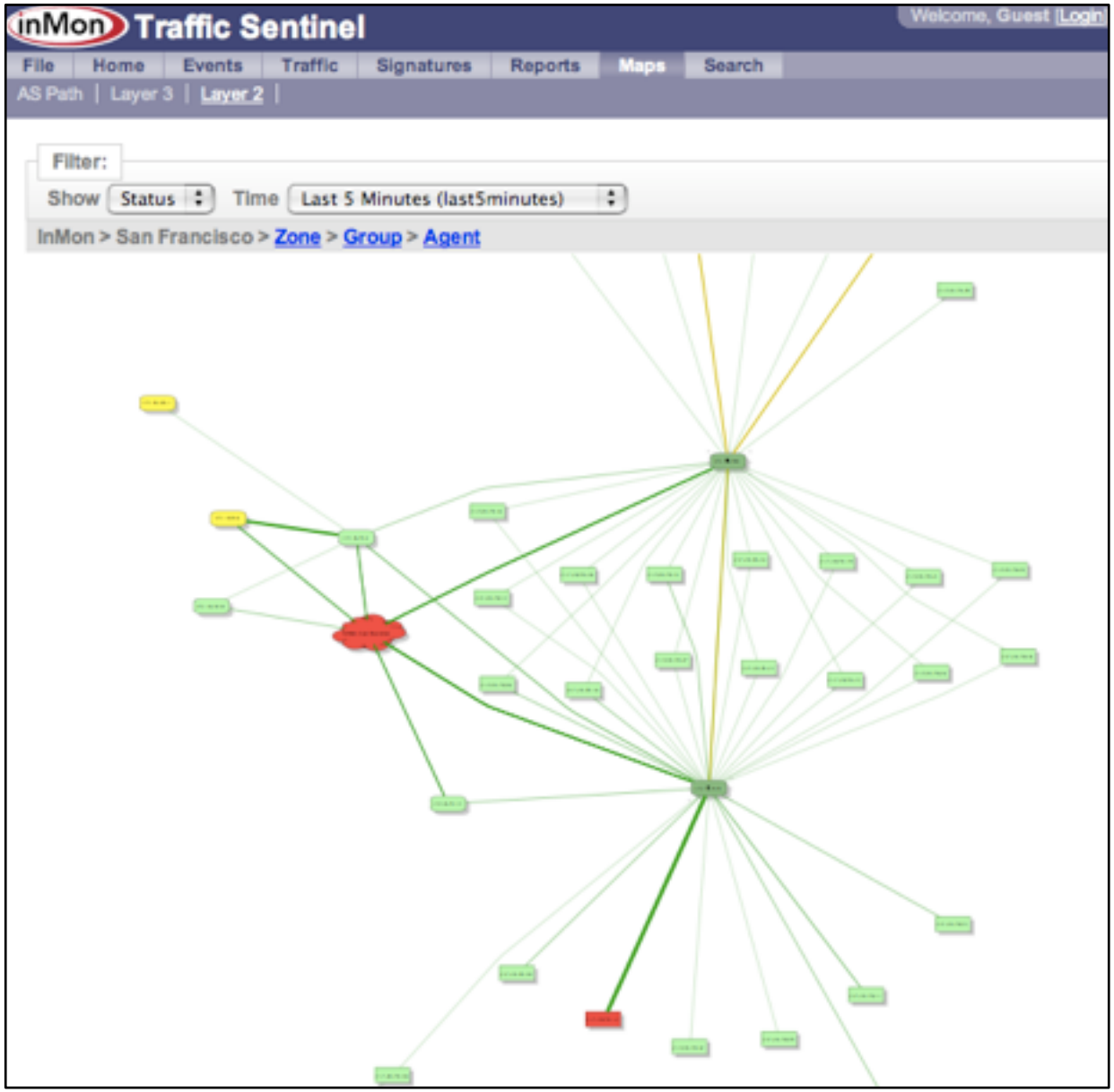
sFlow Overview: captures packet path



- Each packet sample captures the forwarding path for the packet
- Threading together the paths provides a constantly updating picture of network topology and host locations
- The combination of forwarding table data and packet headers provides an integrated view of traffic. E.g. you can filter on forwarding attributes (VLAN, MPLS, route) and see traffic, or filter on traffic and identify forwarding paths.

Traffic Sentinel: Multivendor topology discovery

Uses:
- sFlow
- CDP
- FDP
- LLDP
- Spanning-tree
- Bridge-tables
- and more...

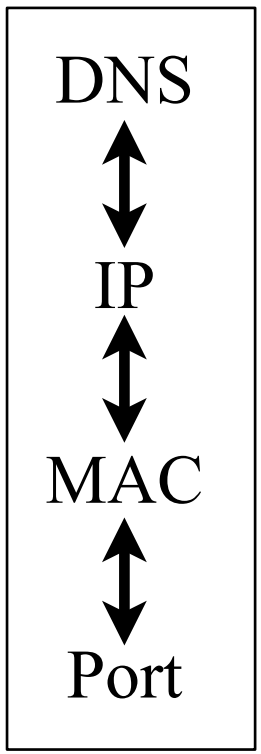


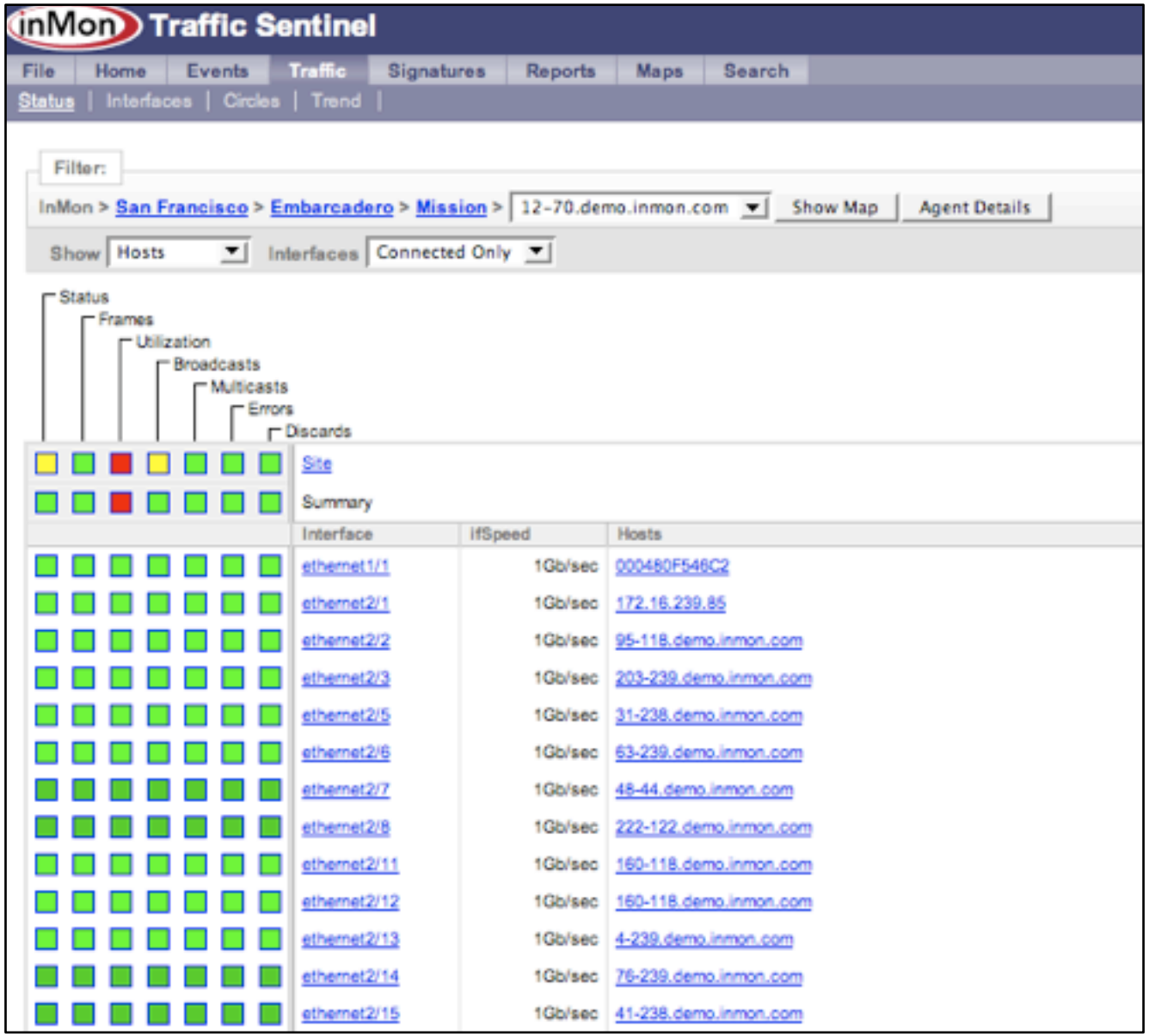
- Auto-layout
- Mouse-wheel zoom
- Show Status,Traffic

(refreshed every minute)

Traffic Sentinel: End-host location

Uses:
sFlow
SNMP





With sFlow, host locations can be updated within 60 seconds

sFlow Architecture

sFlow Architecture

Simple agents

sFlow Architecture

Simple agents

sFlow Architecture

Simple agents

Smart collector

sFlow Architecture

Simple agents

Easy to implement

Smart collector

sFlow Architecture

Simple agents

Smart collector

Easy to implement
Embedded, wire-speed

sFlow Architecture

Simple agents

Smart collector

- Easy to implement
- Embedded, wire-speed
- Low cost

sFlow Architecture

Simple agents

Smart collector

- Easy to implement
- Embedded, wire-speed
- Low cost
- Counter-push

sFlow Architecture

Simple agents → sFlow → Smart collector

Easy to implement
Embedded, wire-speed
Low cost
Counter-push
+ packet/transaction sampling

sFlow Architecture

Simple agents

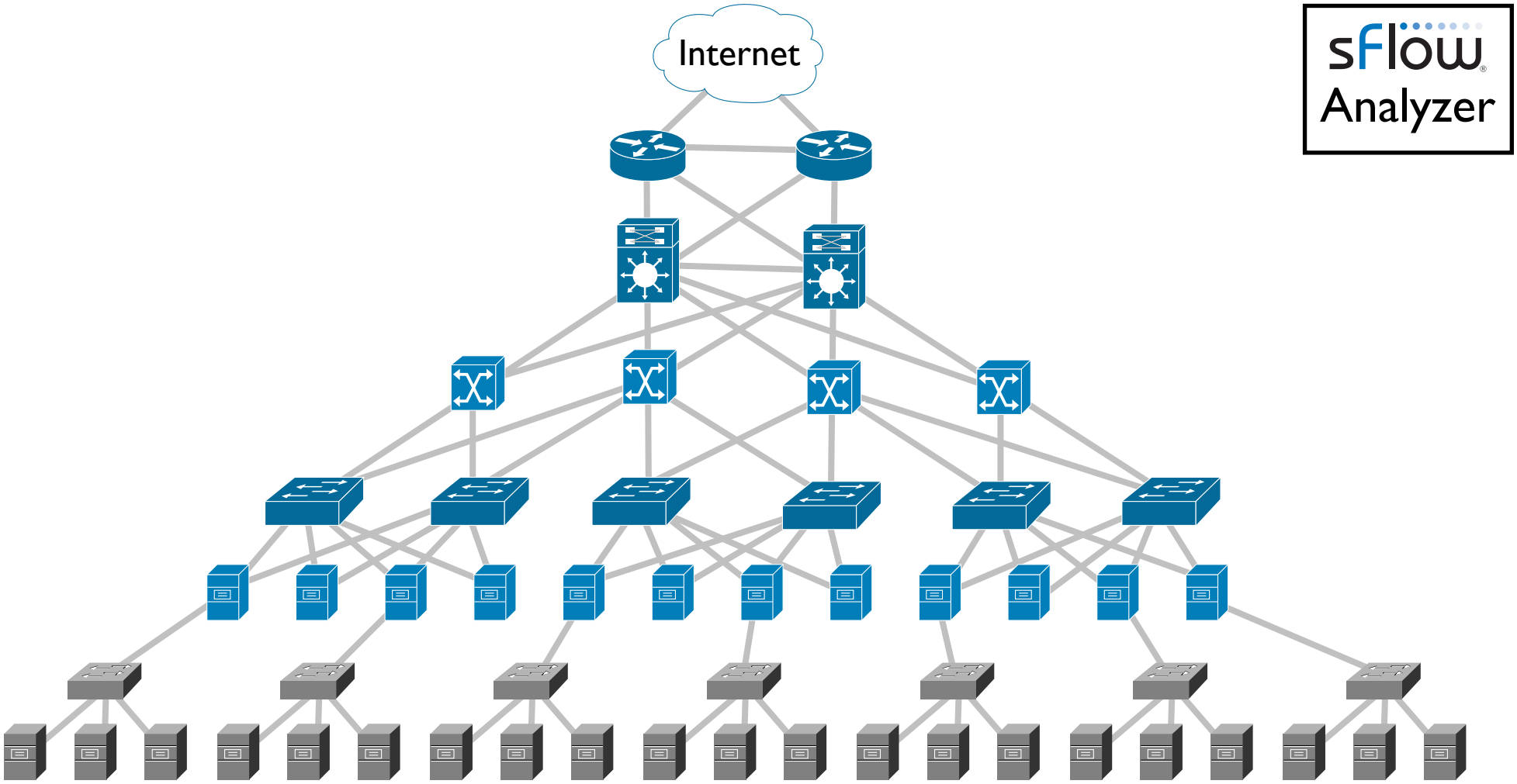
Easy to implement
Embedded, wire-speed
Low cost
Counter-push
+ packet/transaction sampling

Smart collector

Network-wide,
integrated,
visibility and control

sFlow Architecture

Simple agents → sFlow → Smart collector

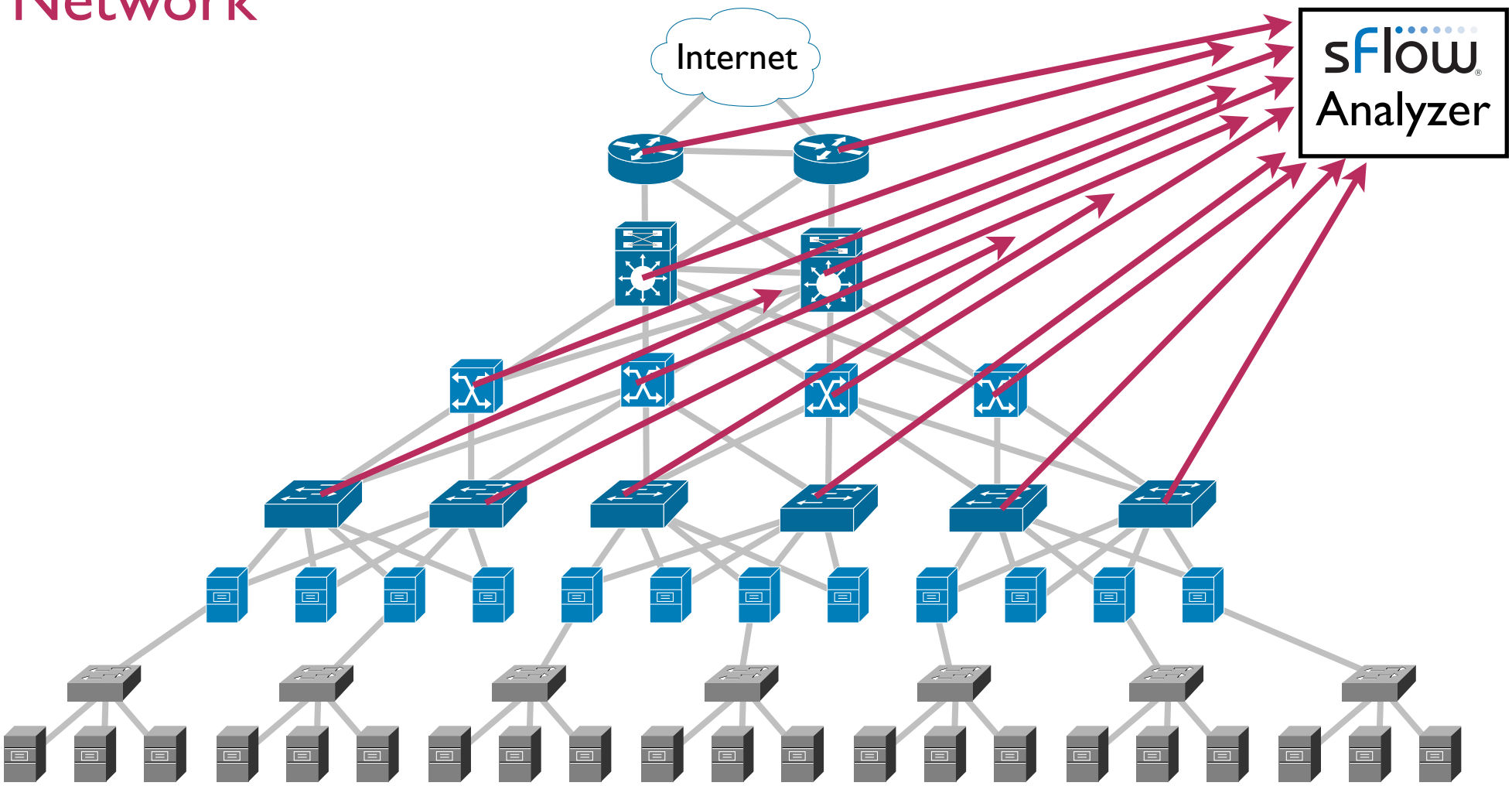


sFlow Architecture

Simple agents

Network

Smart collector



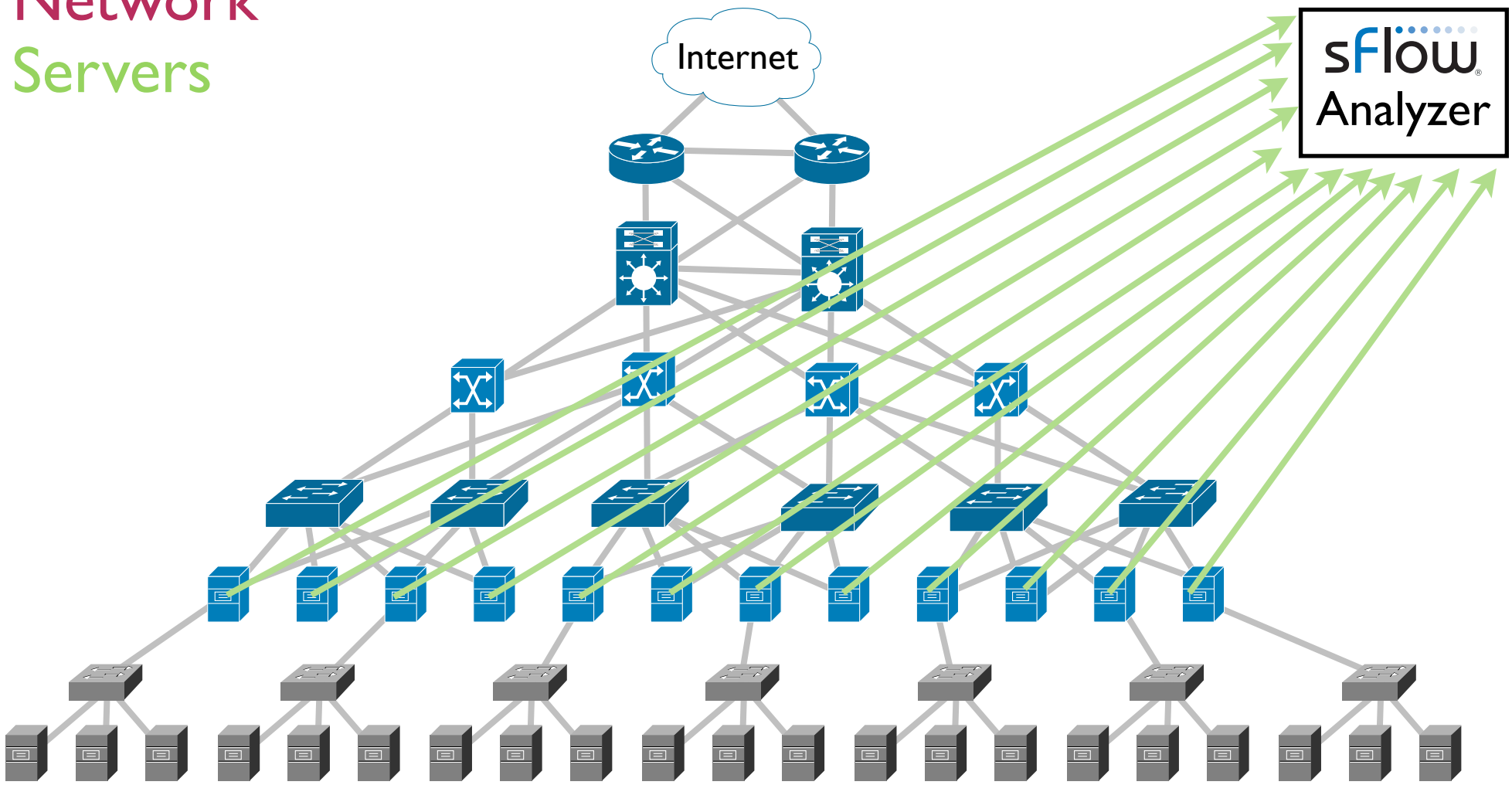
sFlow Architecture

Simple agents

Network

Servers

Smart collector



sFlow Architecture

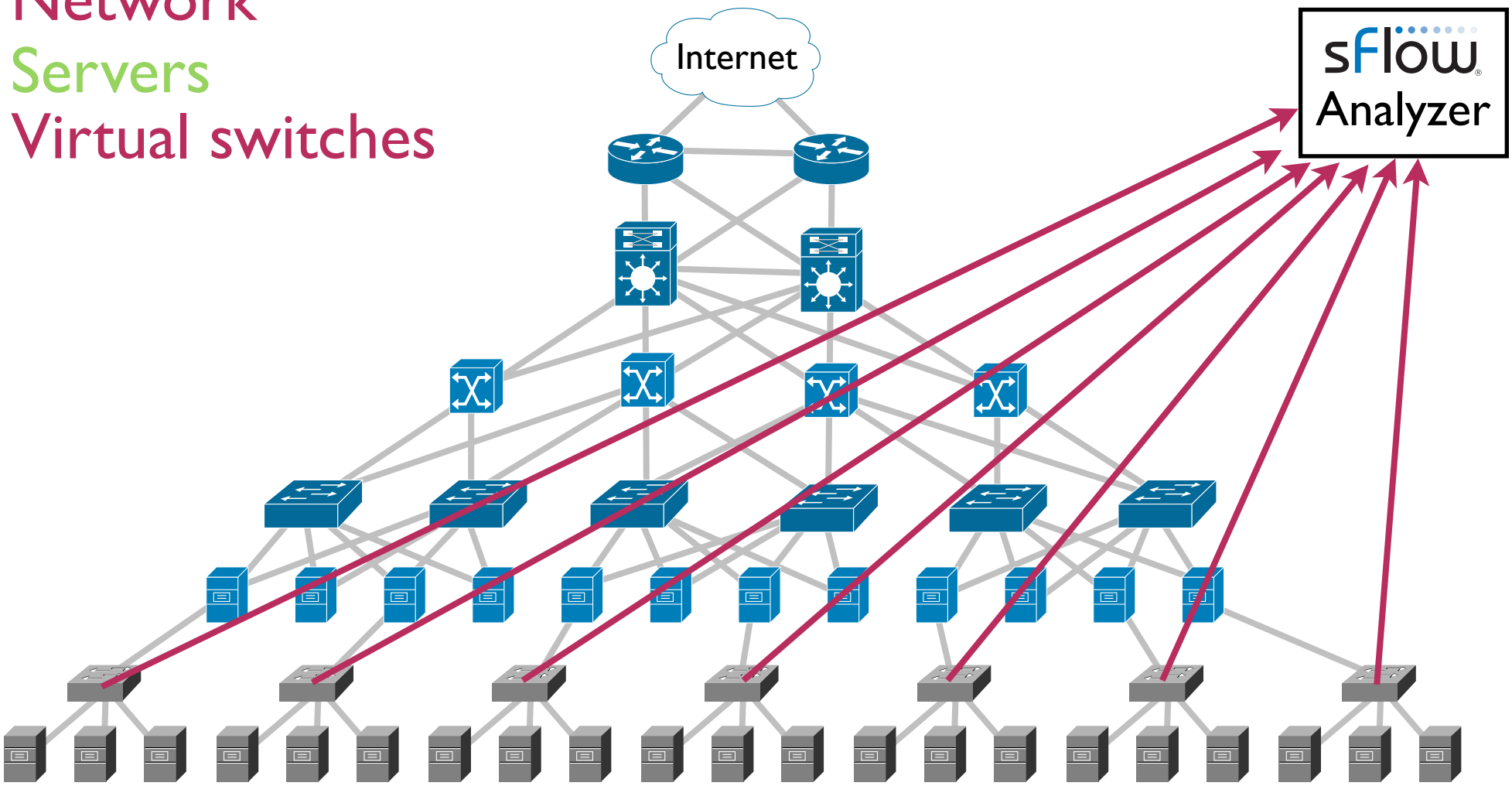
Simple agents

Network

Servers

Virtual switches

Smart collector



sFlow Architecture

Simple agents

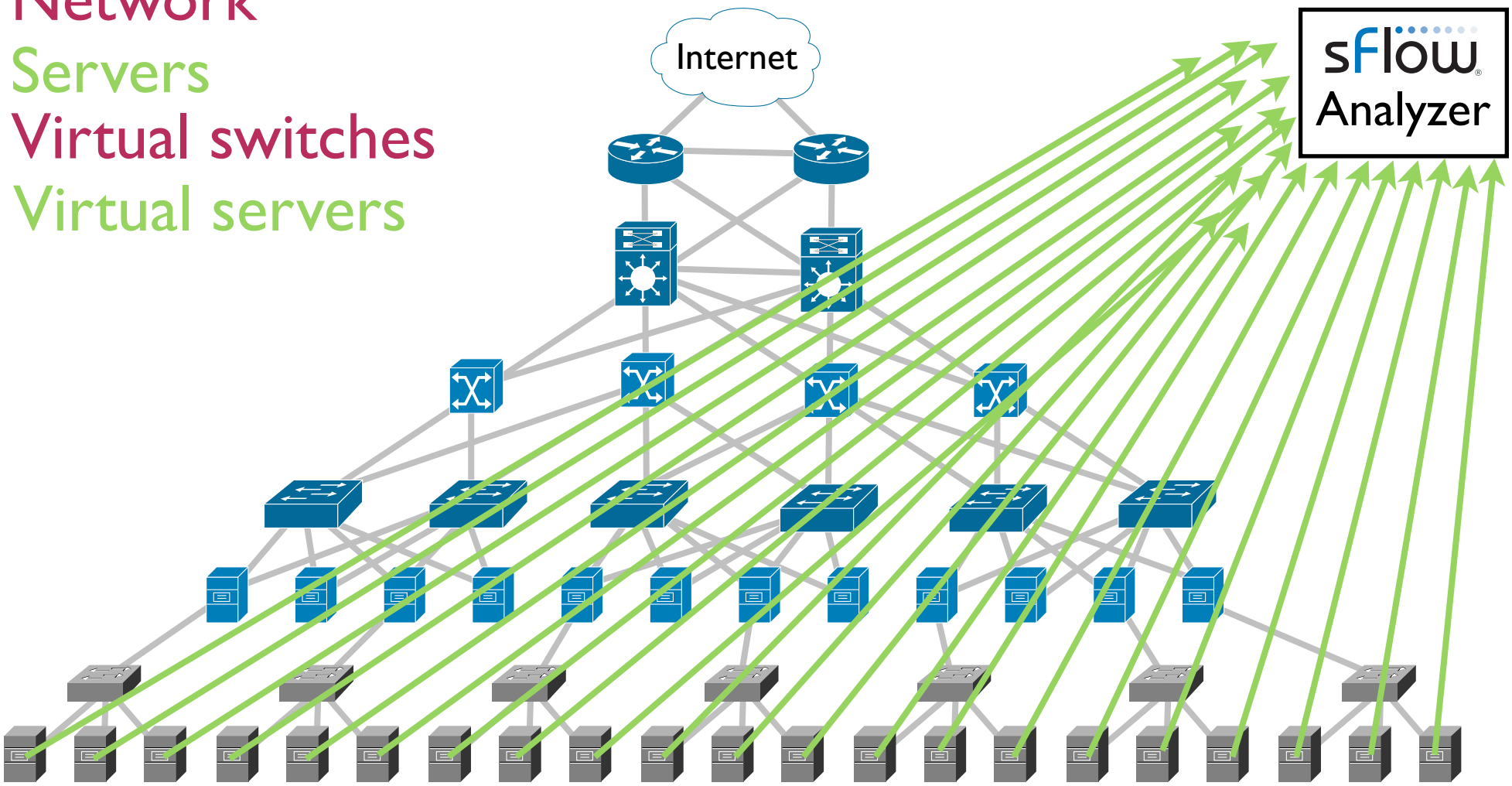
Network

Servers

Virtual switches

Virtual servers

Smart collector



sFlow Architecture

Simple agents

Network

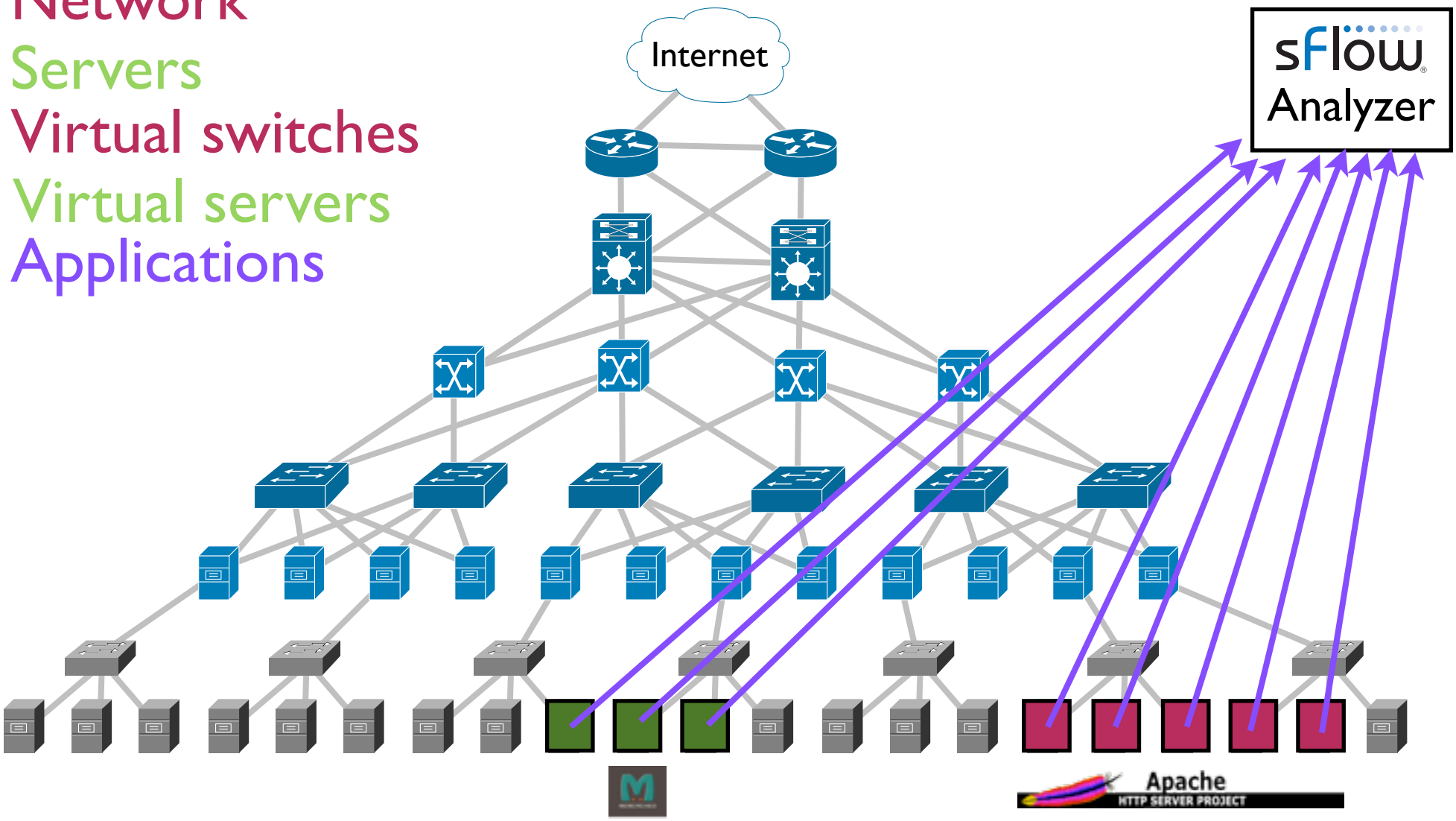
Servers

Virtual switches

Virtual servers

Applications

Smart collector



sFlow Architecture

Simple agents

Network

Servers

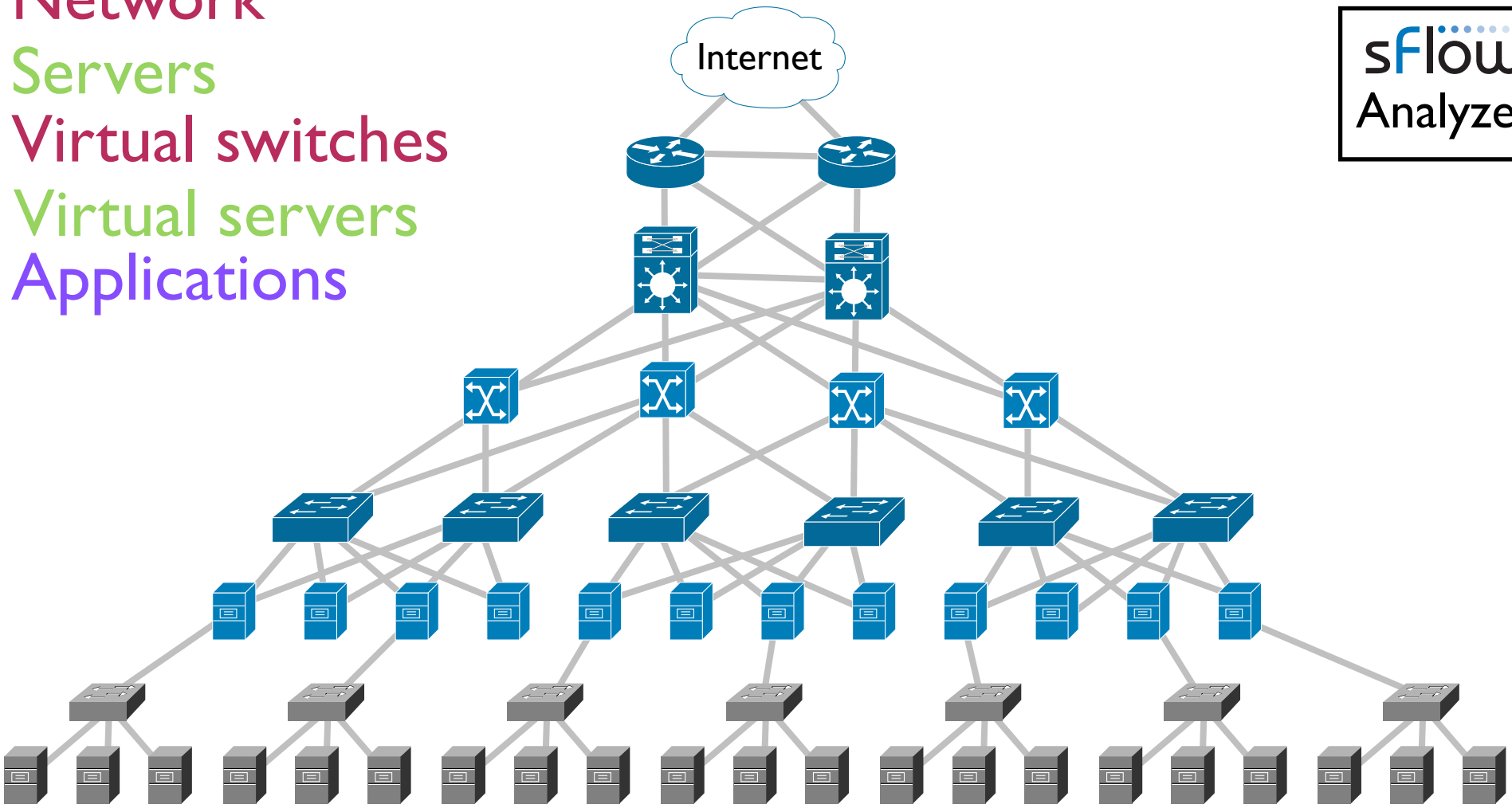
Virtual switches

Virtual servers

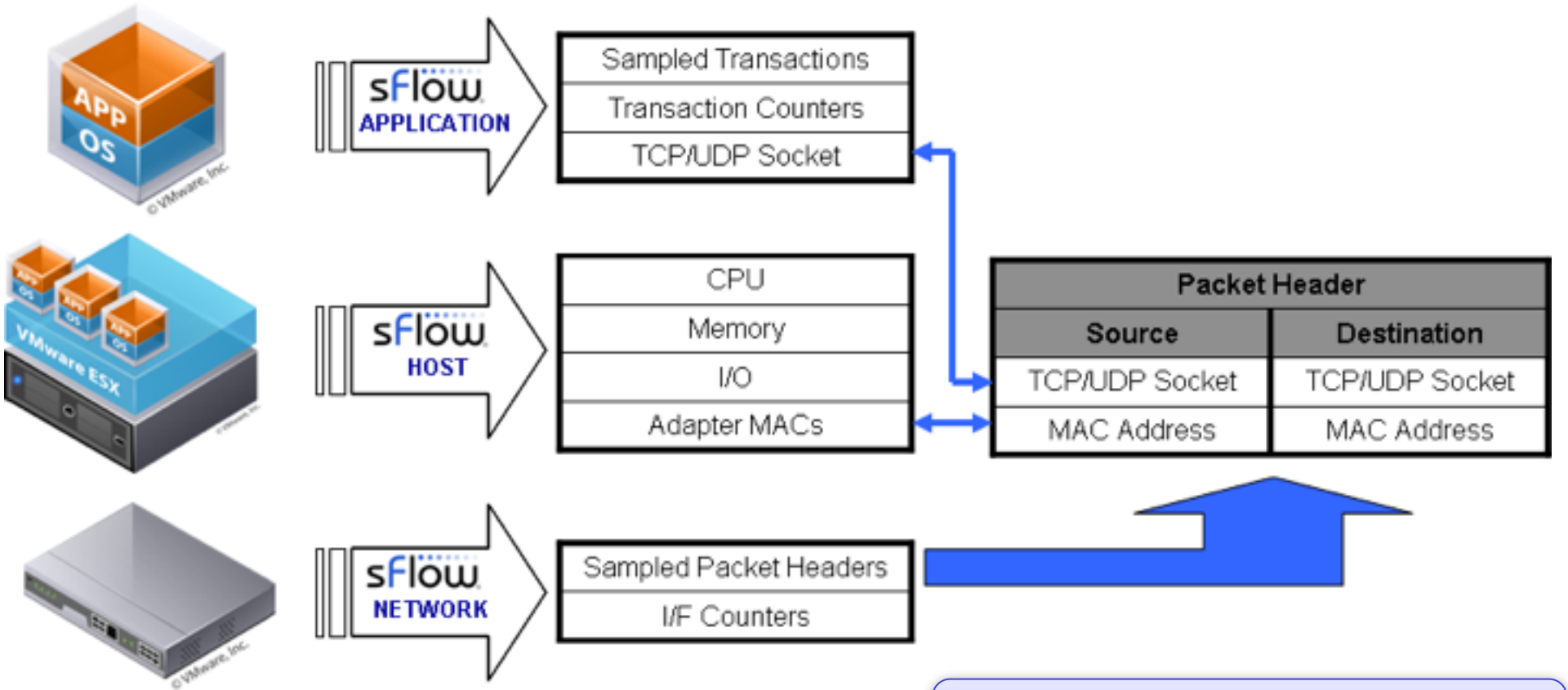
Applications

Smart collector

sFlow Analyzer



Cross-layer correlation: Application, Host and Network



Standard measurements from different sources designed to be joinable at the collector

e.g. application response time increase correlated directly to congestion on network path

host-sflow.sourceforge.net

host sFlow

Data Center Wide Server Performance Monitoring

Home | About Host sFlow | Documentation | Related Links | Participate

The **Host sFlow** agent exports physical and virtual server performance metrics using the sFlow protocol. The agent provides scalable, multi-vendor, multi-OS performance monitoring with minimal impact on the systems being monitored.

NEWS

July 7, 2011 - Ganglia 3.2 supports Host sFlow

March 4, 2011 - FreeBSD support implemented

February 22, 2011 - Windows load averages added

December 17, 2010 - iptables/ULOG traffic monitoring added

October 15, 2010 - KVM/libvirt support implemented

October 8, 2010 - Host-sFlow version 1.06 released (XCP 0.5 /XenServer 5.6 FP1)

September 23, 2010 - Windows installer implemented

September 1, 2010 - Host-sFlow 1.3 released adding Open vSwitch integration

July 29, 2010 - New project web site launched

July 23, 2010 - Host-sFlow version 1.0 released

June 16, 2010 - DNS service discovery implemented (DNS-SD)

May 1, 2010 - Host sFlow project started

DOWNLOAD NOW

The latest release of the host-sflow sources can be downloaded at http://sourceforge.net/projects/host-sflow

Supported operating systems:

- FreeBSD
- Linux
- Windows

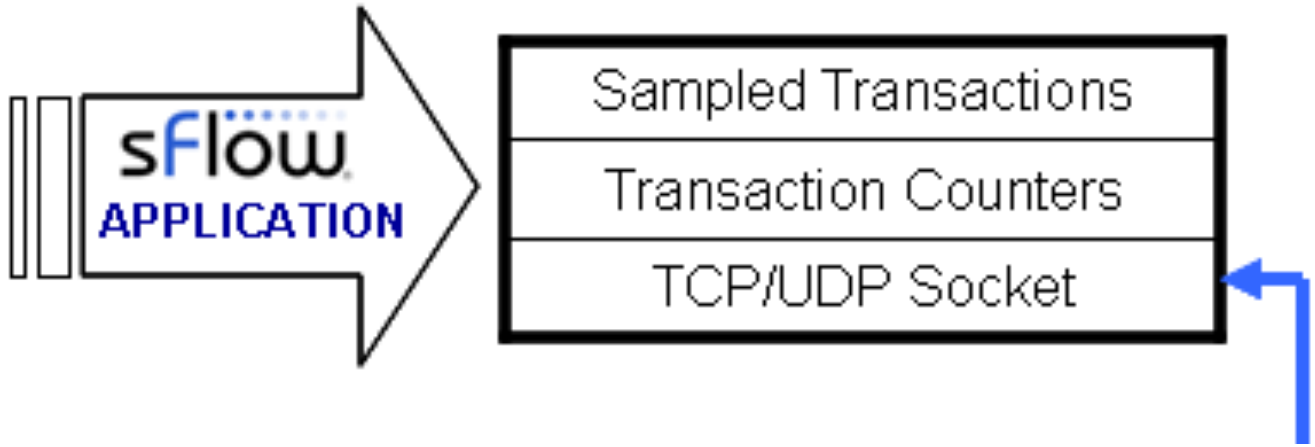
Supported hypervisors:

- KVM/libvirt
- Xen/XCP/XenServer

Home | About Host sFlow | Documentation | Participate

For more information, join the host-sflow-discuss mailing list

sFlow-APPLICATION



Examples:

NFS/CIFS transactions (file path, bytes, response-time, socket)

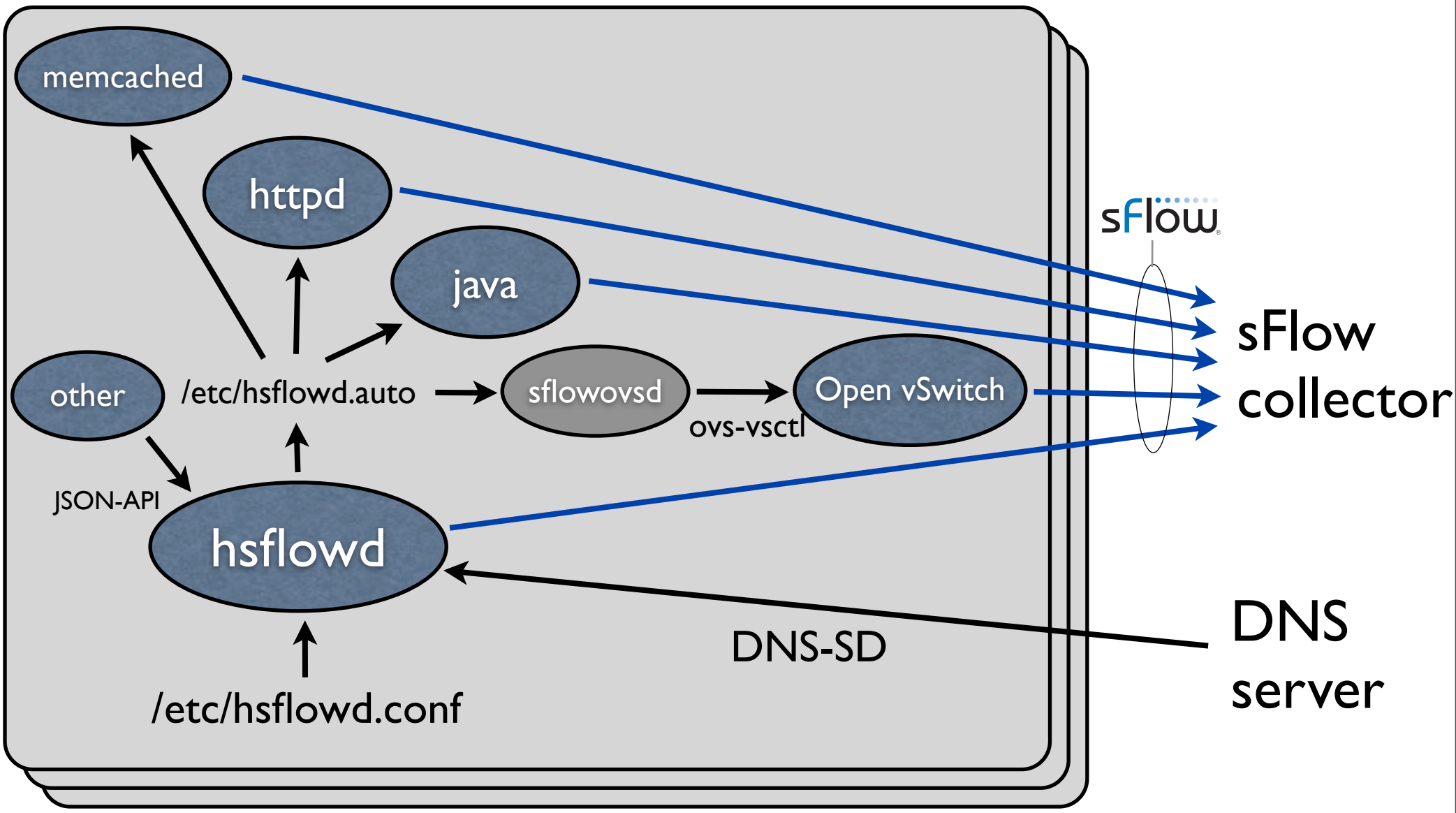
HTTP requests (URL, user-agent, mime-type, bytes, response-time, socket)

Memcached lookups (key, value-bytes, hit/miss, socket)

Database queries (query#, response-time, socket)

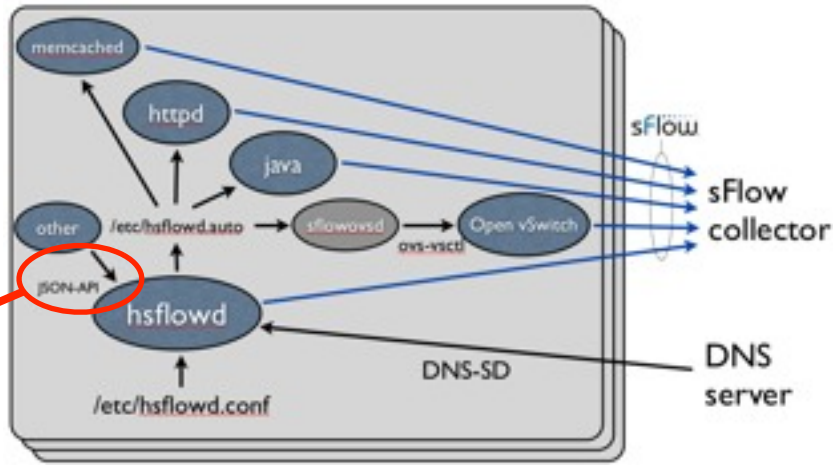
Application layer measurements much more valuable when correlated with performance of every component in underlying infrastructure!

Host sFlow distributed agent



Host sFlow - JSON API

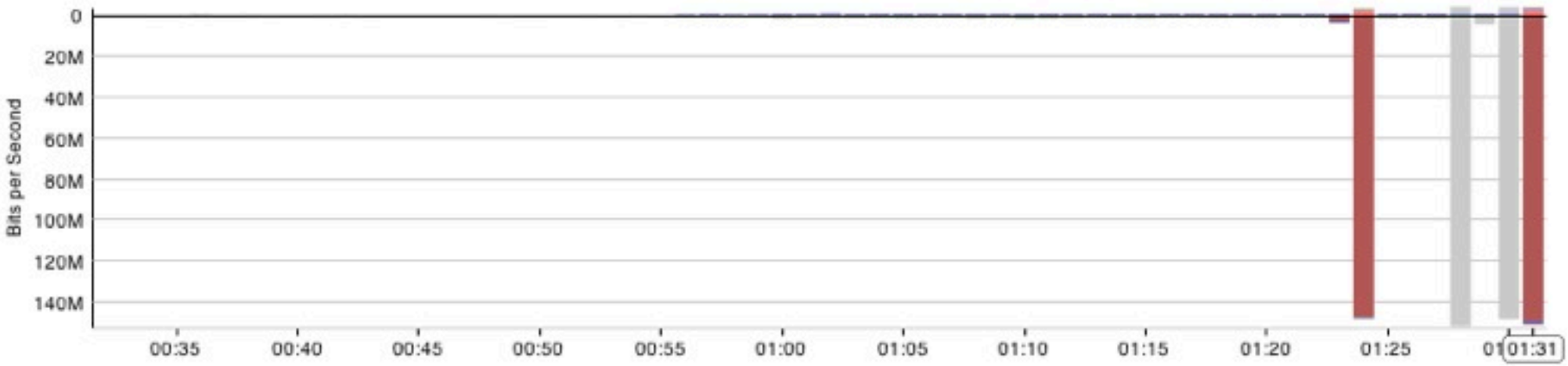

```
{
 "flow_sample:{
  "app_name":"myapp",
  "sampling_rate": 100
  "app_operation": {
   "operation":"task.start",
   "attributes":"id=123&user=root",
   "status_descr":"OK",
   "status":0,
   "req_bytes":43,
   "resp_bytes":234,
   "uS":2000
  },
  "app_initiator": {"actor":"123"},
  "app_target": {"actor":"231"},
  "extended_socket_ipv4": {
   "protocol":6,
   "local_ip":"10.0.0.1",
   "remote_ip":"10.0.0.23",
   "local_port":123,
   "remote_port":43032
  }
 }
}
```

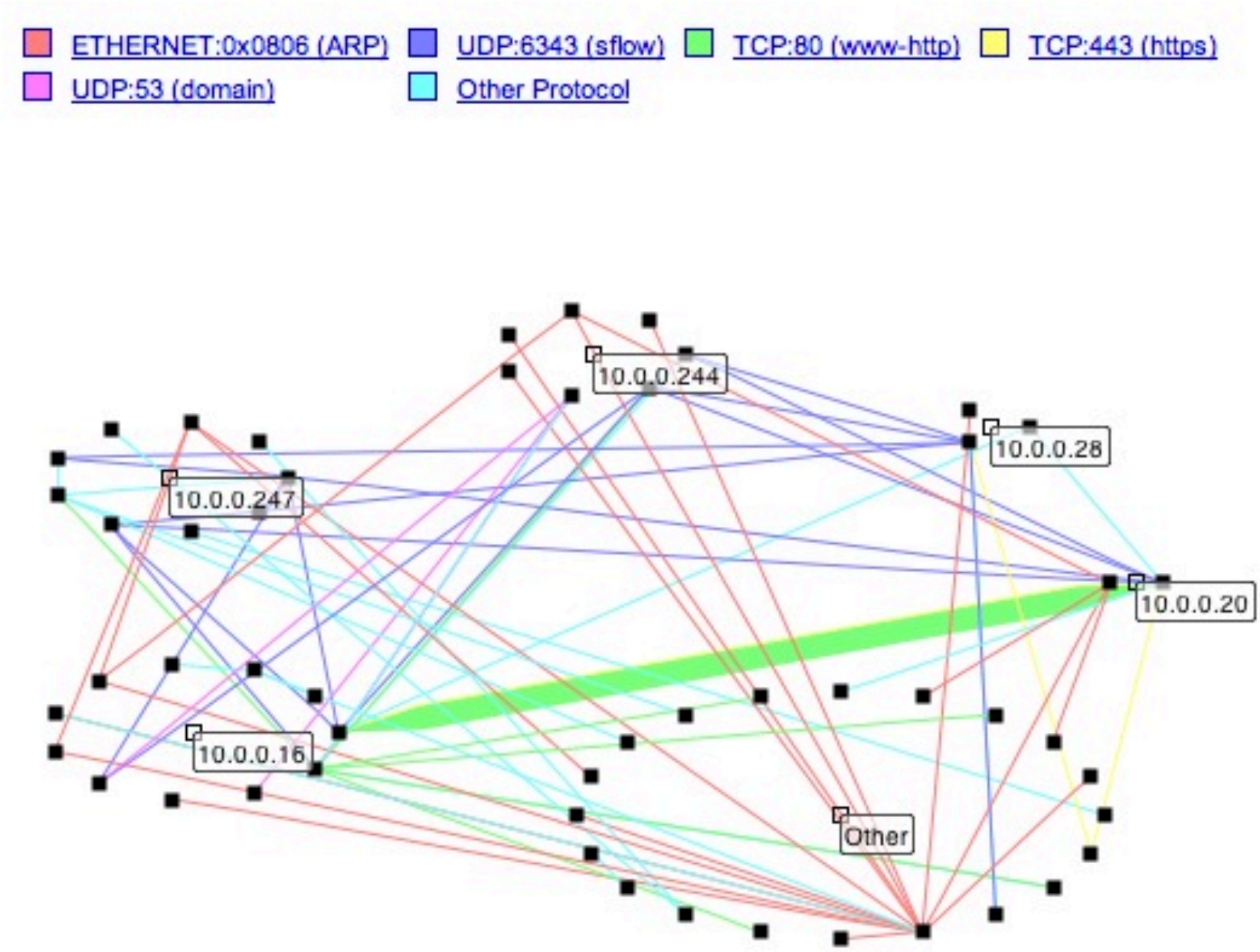
- UDP msg to hsflowd on localhost:36343
- most fields optional
- sampling in app. or hsflowd
- counters in app. or hsflowd
- hsflowd sends binary sFlow-APPLICATION feed to configured collectors.

XenMotion bandwidth

	Server Address	Server Port	Client Address	Bytes From Server	Bytes To Server	Bytes
	xenserver2 (10.0.0.16)	TCP:80 (www-http)	xenserver1 (10.0.0.20)	2.73M	149.54M	152.26M
	ganglia (10.0.0.112)	TCP:11211 (memcache)	xenvm1 (10.0.0.150)	773.75K	1.07M	1.85M
	openfiler (10.0.0.18)	TCP:3260 (iscsi-target)	xenserver1 (10.0.0.20)	251.93K	21.21K	273.14K
	openfiler (10.0.0.18)	TCP:3260 (iscsi-target)	xenserver2 (10.0.0.16)	167.92K	72.85K	240.77K
	openfiler (10.0.0.18)	TCP:2049 (nfs)	xenserver1 (10.0.0.20)	0	211.39K	211.39K

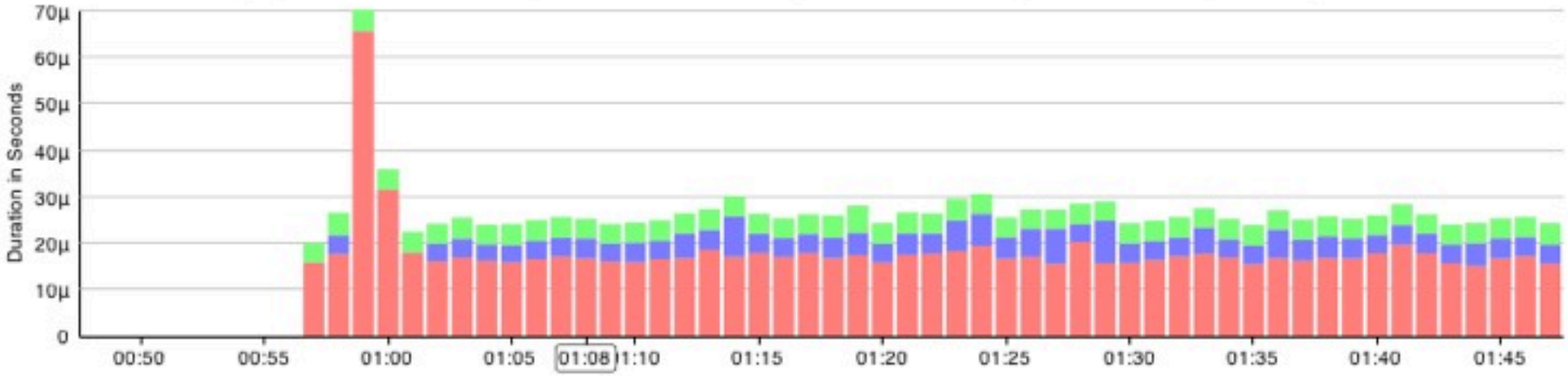


XenMotion bandwidth



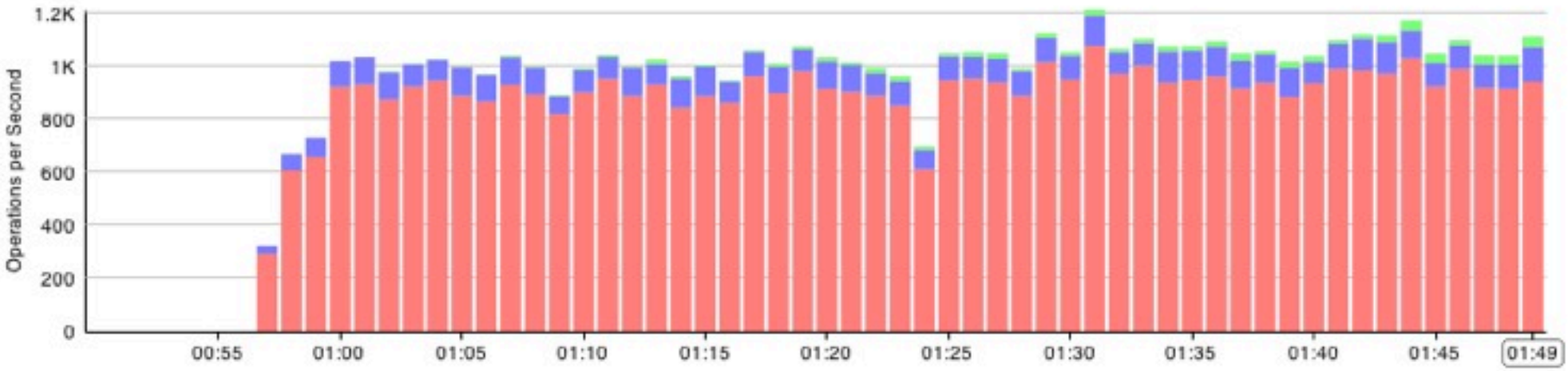
Application throughput and response time

	Memcache Protocol	Memcache Command	Memcache Status	Memcache Key	Duration
	ASCII	SET	STORED	brutis	16.70µ
	ASCII	GET	EXISTS	brutis	4.25µ
	ASCII	GET	NOT_FOUND	brutis	4.21µ



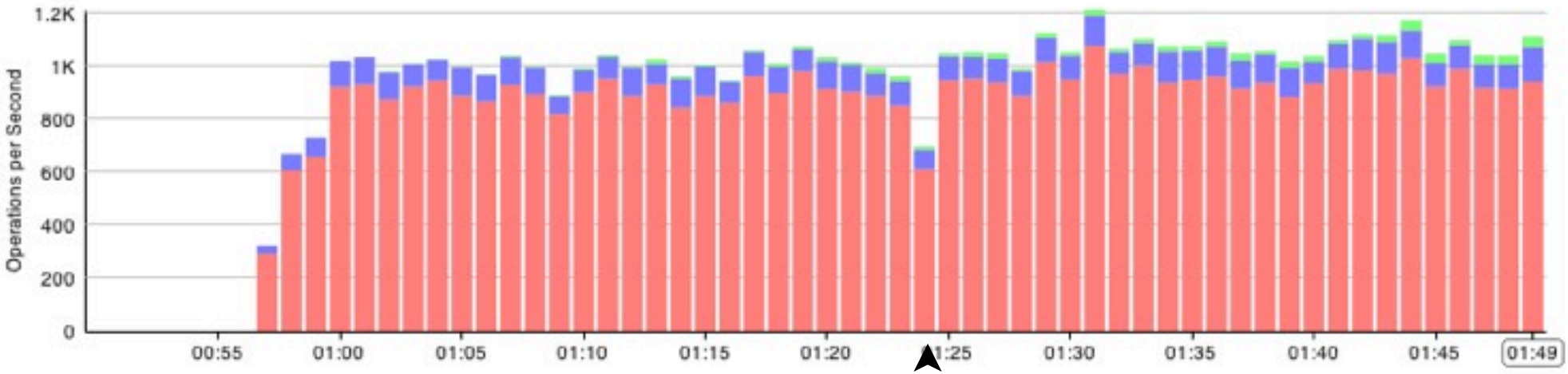
Application throughput and response time

	Memcache Protocol	Memcache Command	Memcache Status	Memcache Key	Operations
	ASCII	GET	NOT_FOUND	brutis	938.33
	ASCII	SET	STORED	brutis	130.00
	ASCII	GET	EXISTS	brutis	37.50



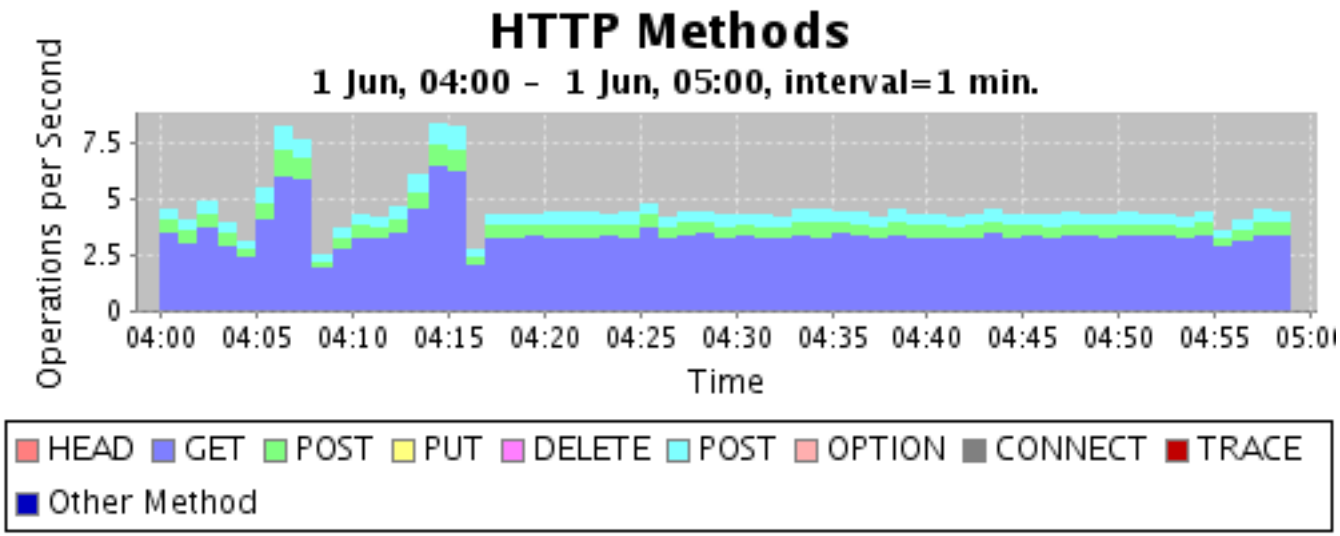
Application throughput and response time

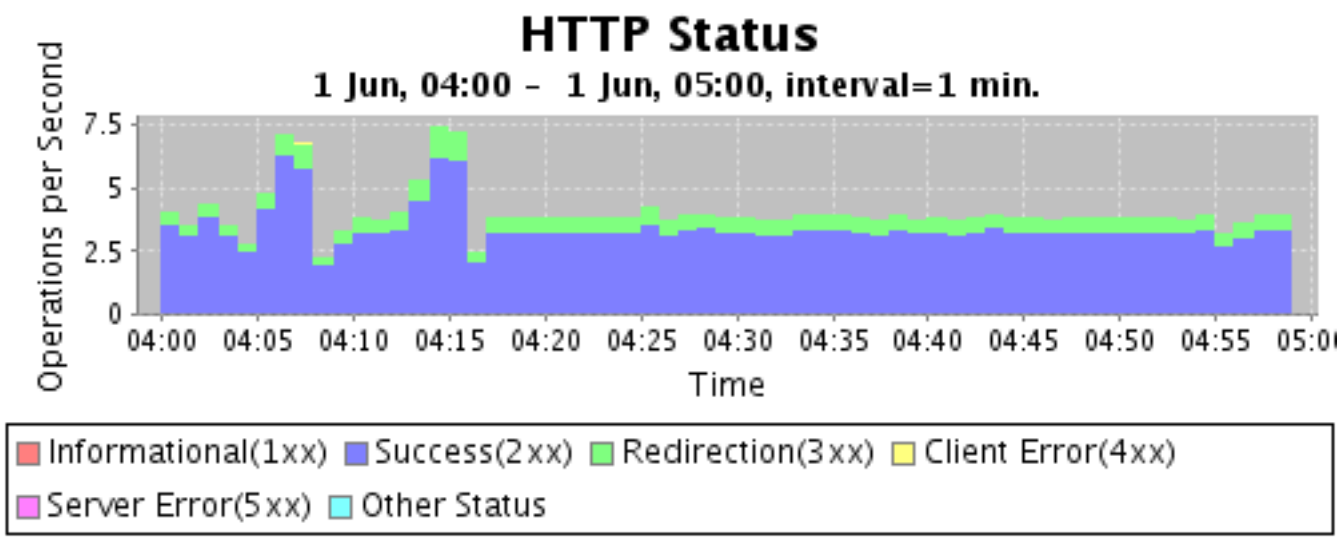
	Memcache Protocol	Memcache Command	Memcache Status	Memcache Key	Operations
	ASCII	GET	NOT_FOUND	brutis	938.33
	ASCII	SET	STORED	brutis	130.00
	ASCII	GET	EXISTS	brutis	37.50



Drop in throughput during XenMotion of Memcache client

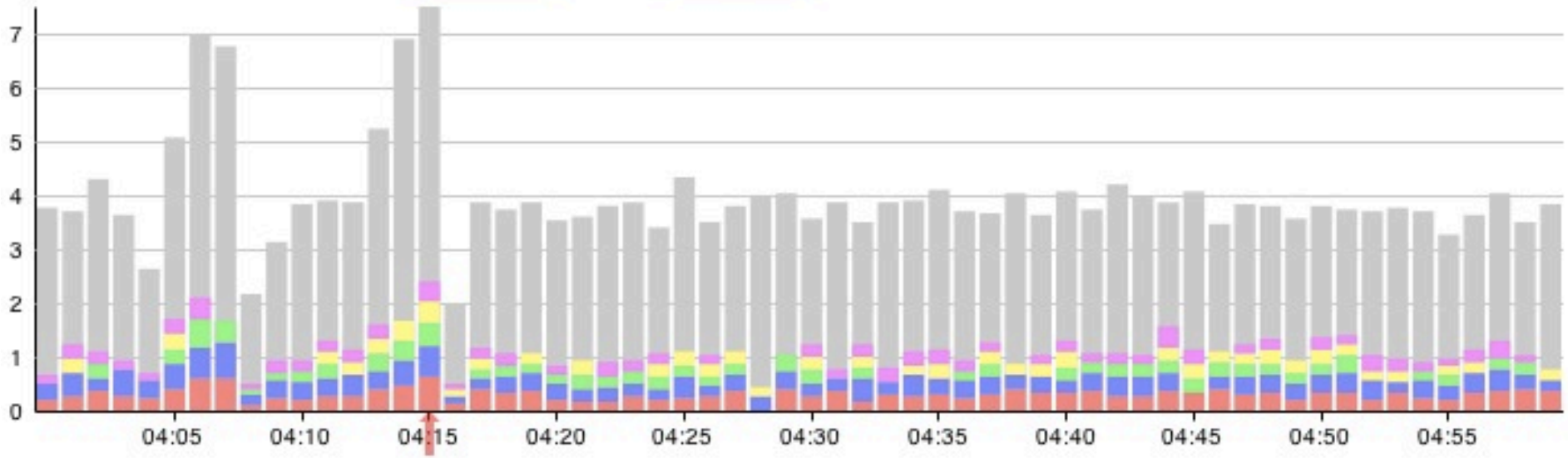
sFlow-APPLICATION example: transactions





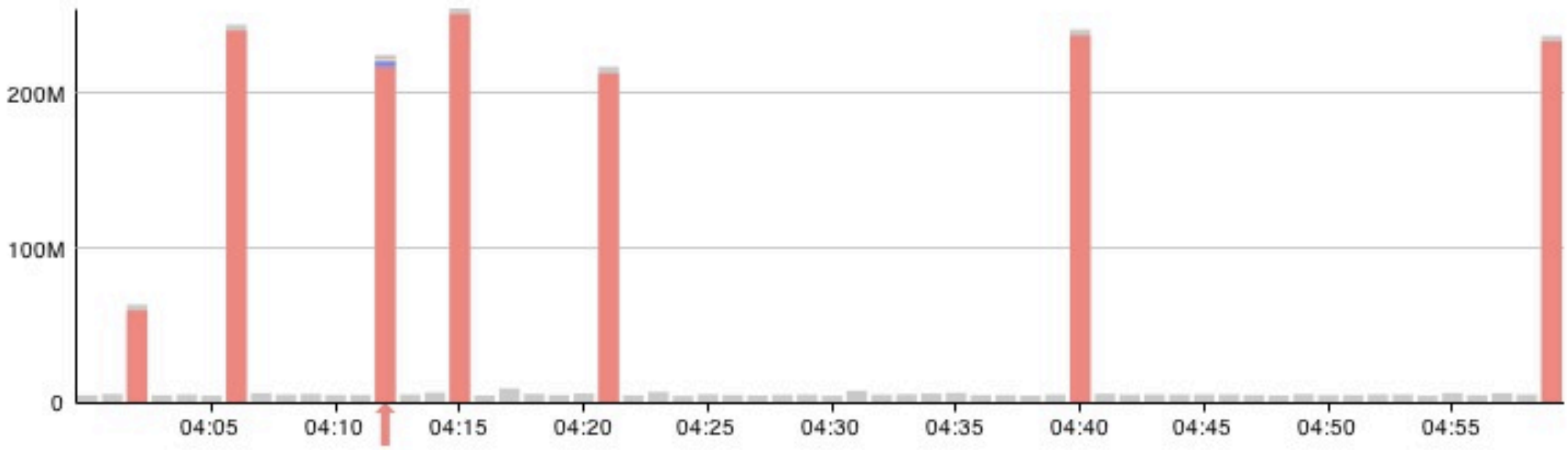
sFlow-APPLICATION example: transaction detail

	URI Host	URI Path	Operations
	goku.brightcove.com	/1pix.gif	0.63
	www.miitv.com	/	0.57
	www.miitv.com	/module/setup.php	0.43
	c.brightcove.com	/services/messagebroker/amf	0.40
	www.miitv.com	/adspace.php	0.37



sFlow-APPLICATION example: latency

	URI Host	URI Path	Duration
	talkgadget.google.com	/talkgadget/channel/bind	216.39M
	ds.addthis.com	/red/psi/sites/www.miitv.com/p.json	3.47M
	c1142172.cdn.cloudfiles.rackspacecloud.com	/100x75/j0c08ui84r.jpg	1.13M
	c1142172.cdn.cloudfiles.rackspacecloud.com	/100x75/eghxushmko.jpg	1.12M
	c1142172.cdn.cloudfiles.rackspacecloud.com	/100x75/gdsussbdf4.jpg	1.12M



Why Monitor Everything?

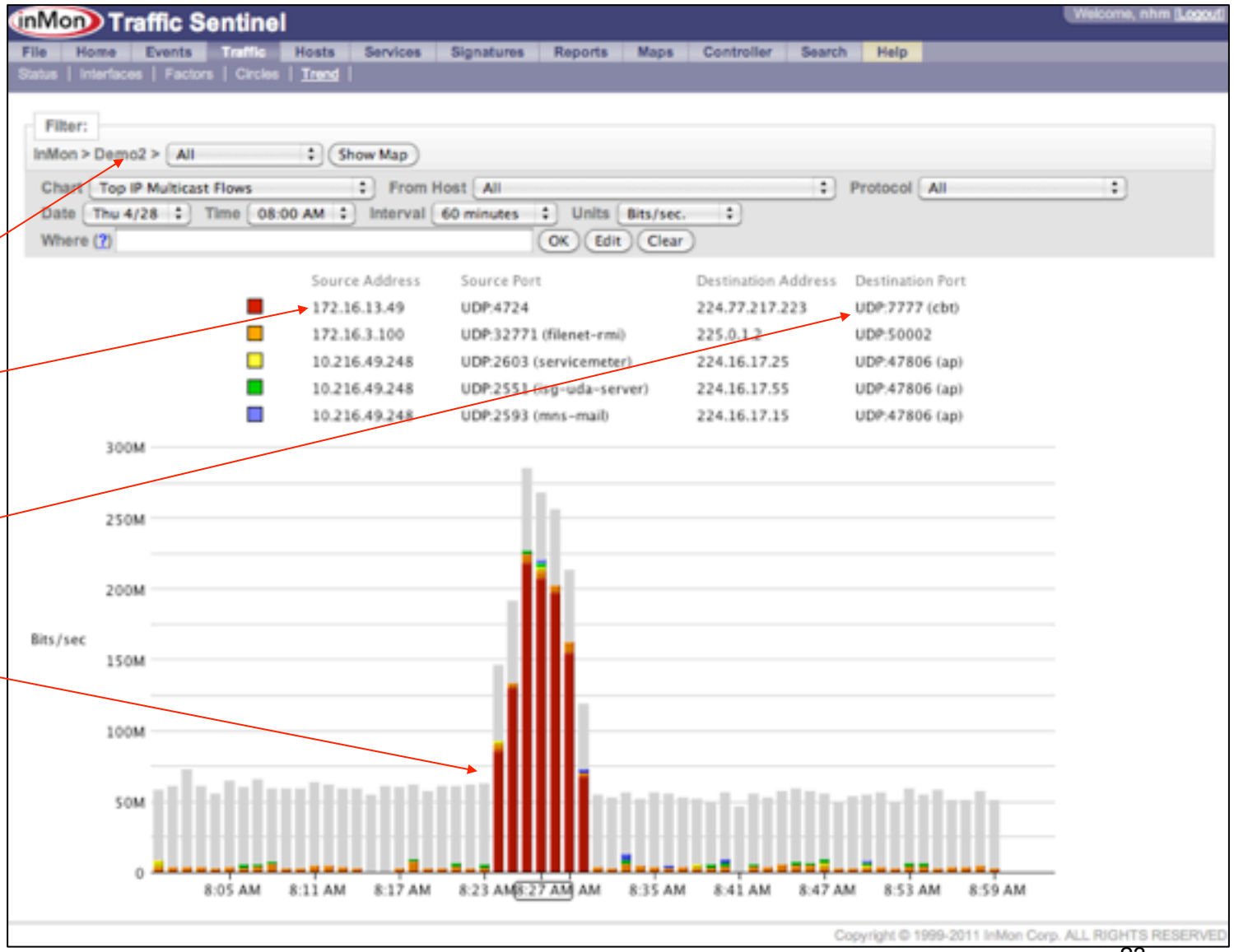
1. Troubleshooting - always have context

where

who

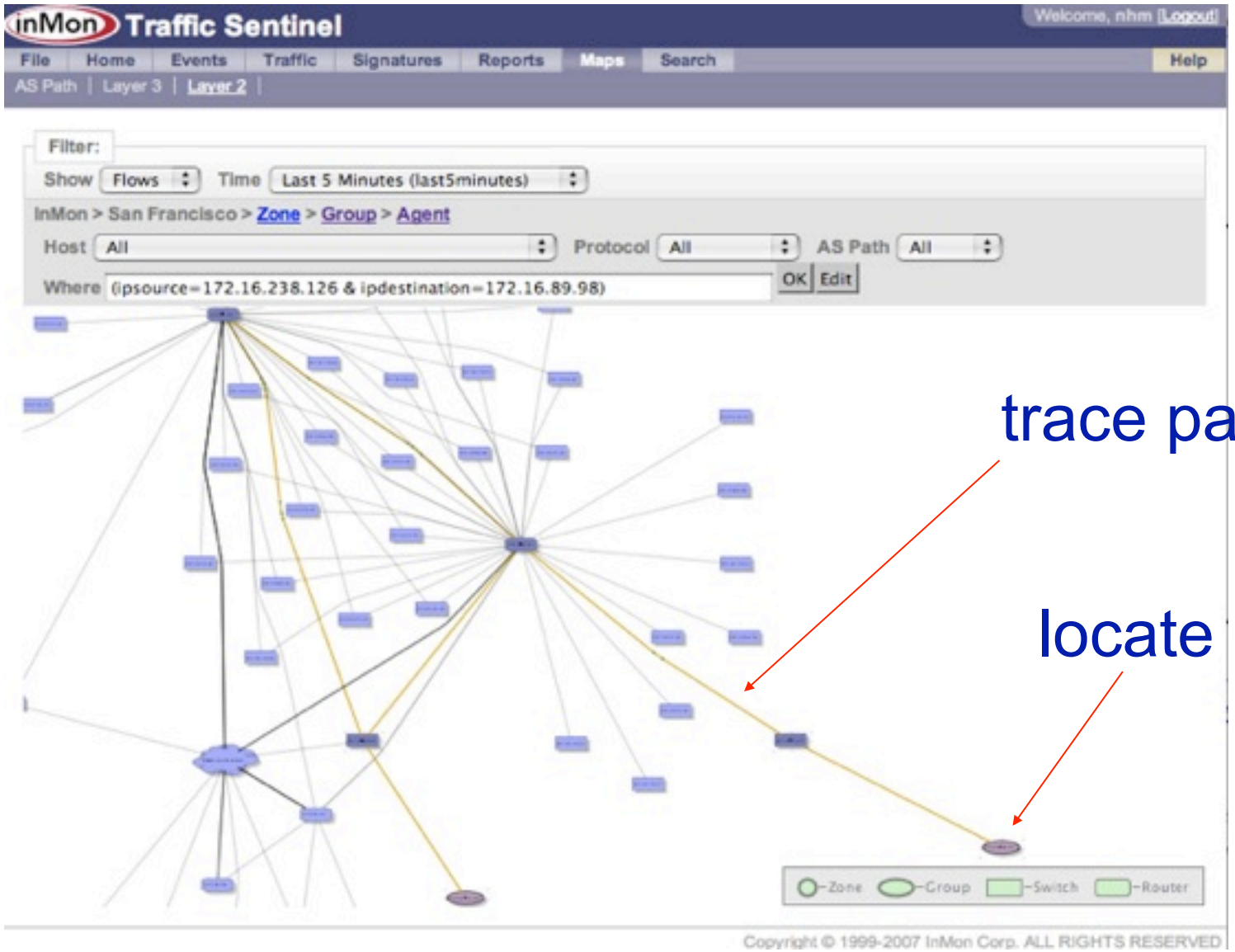
what

when



Why Monitor Everything?

1. Troubleshooting - always have context

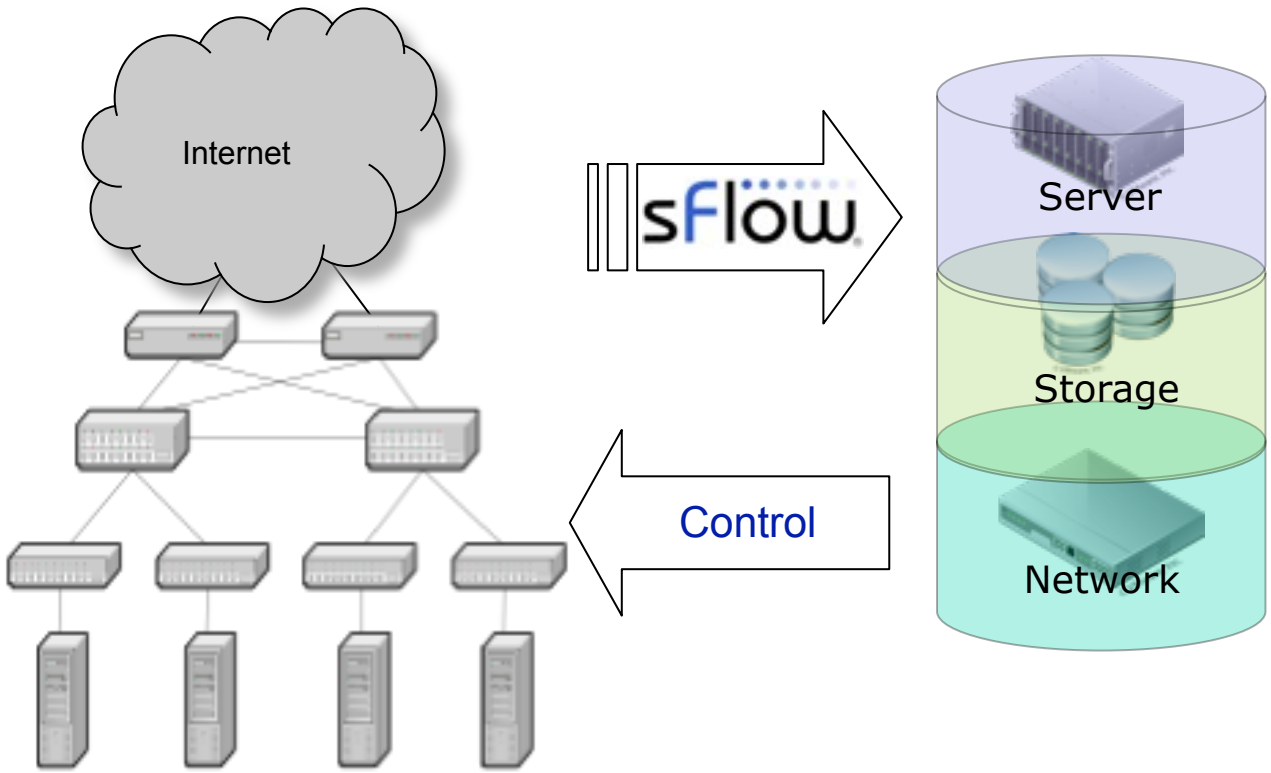


trace path

locate hosts

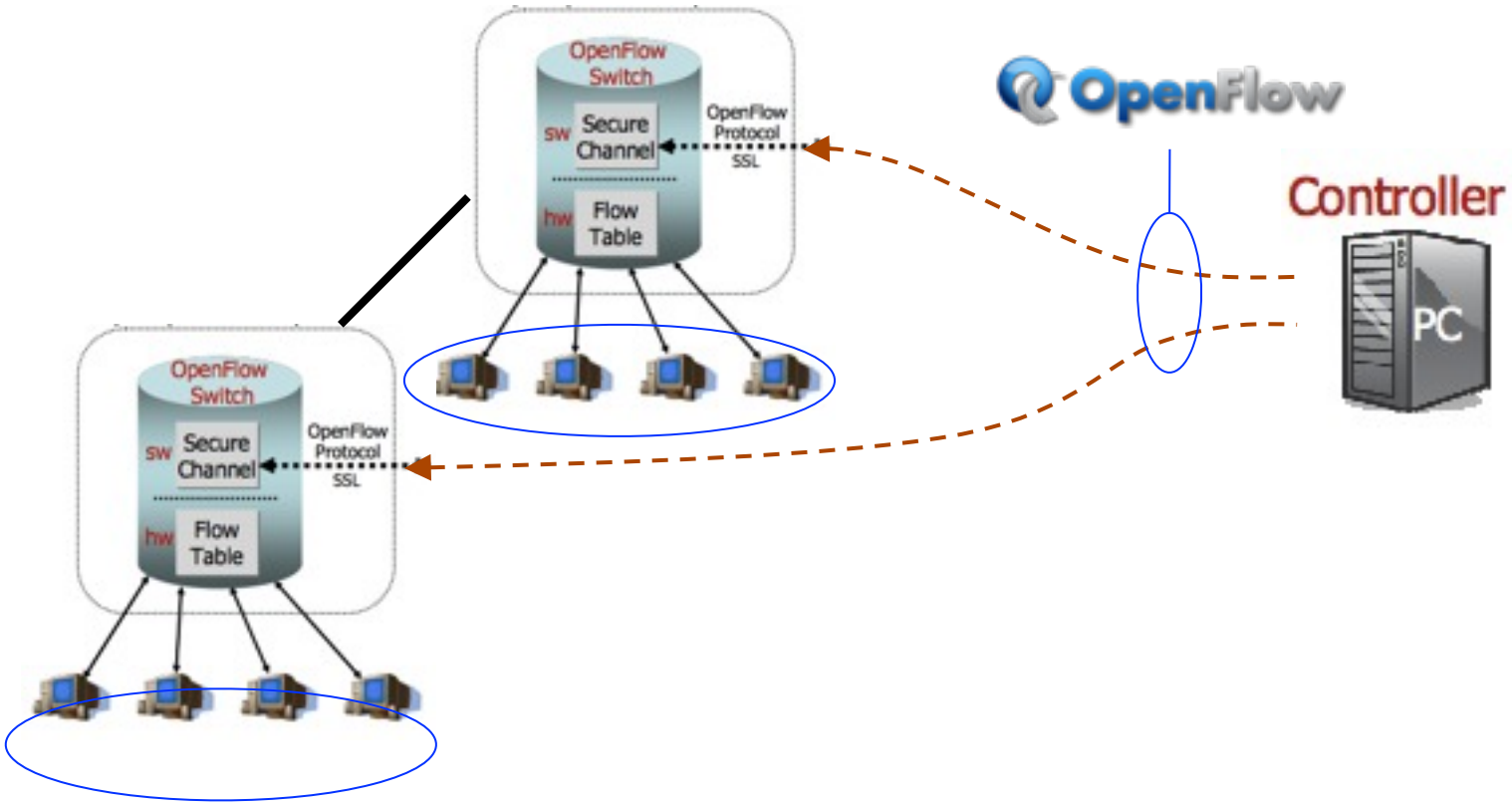
Why Monitor Everything?

2. Put Network and Server teams on same page

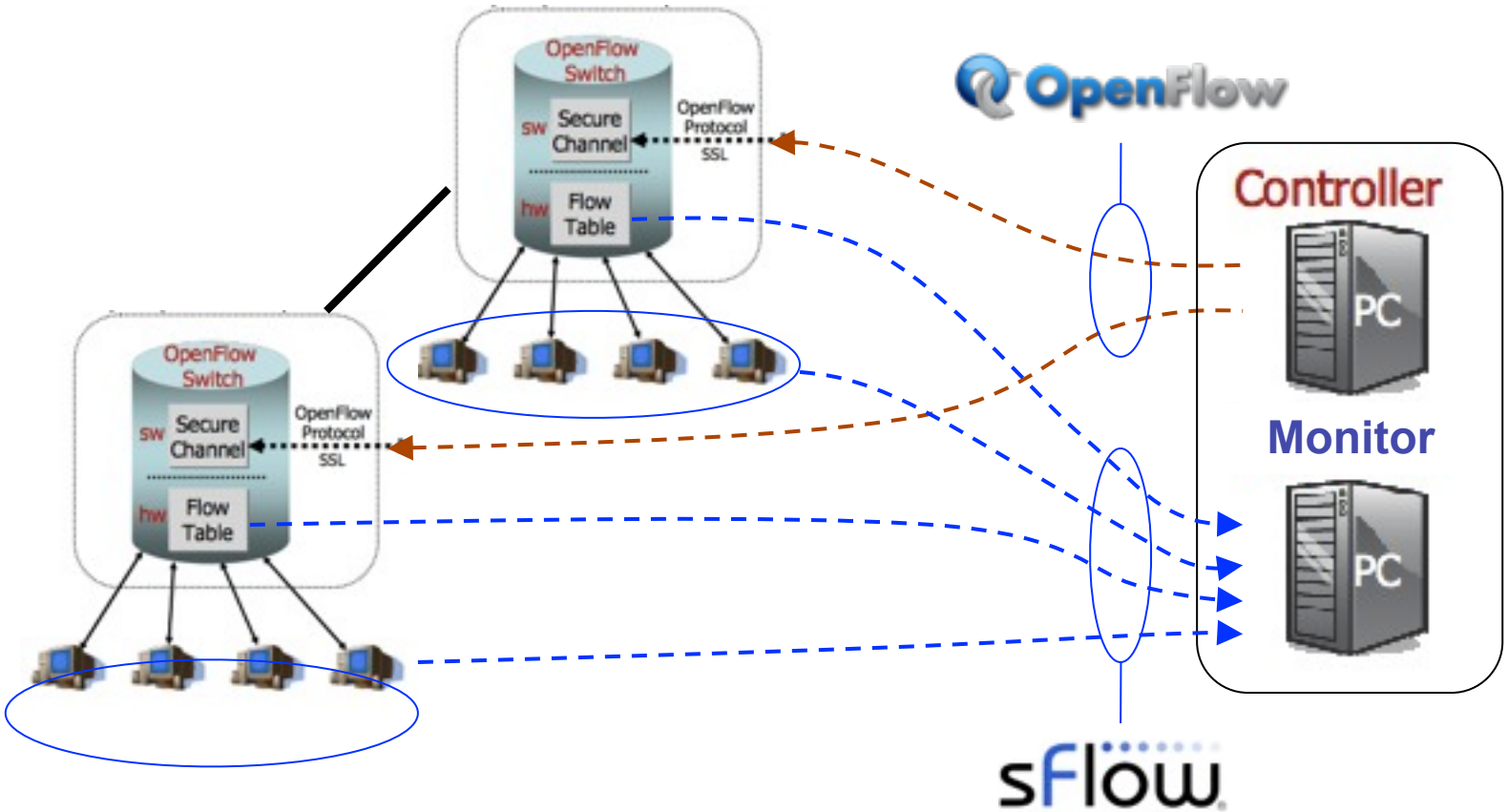


3. Full "Observability" required for automated control

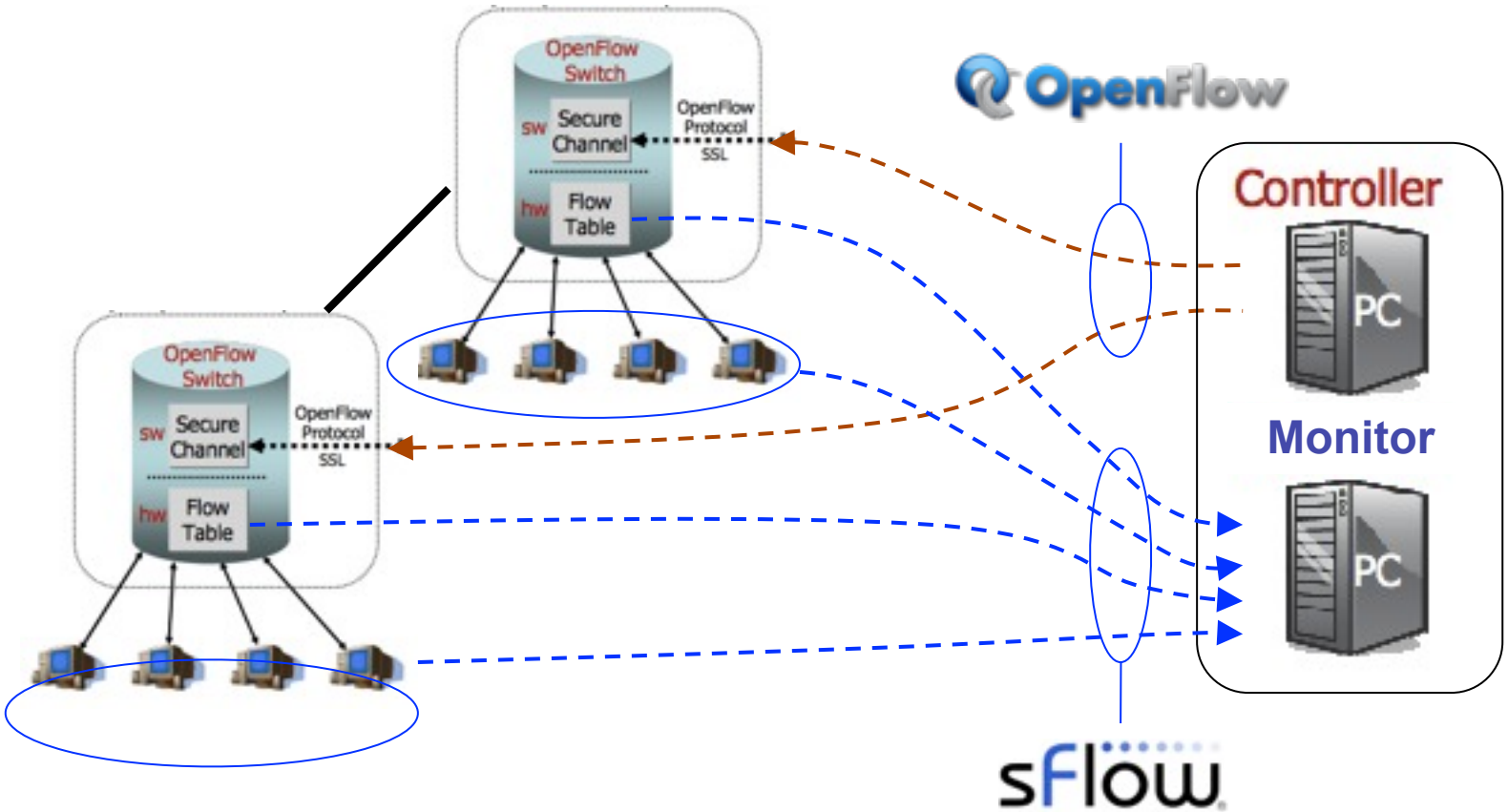
OpenFlow + sFlow = Adaptive Control



OpenFlow + sFlow = Adaptive Control

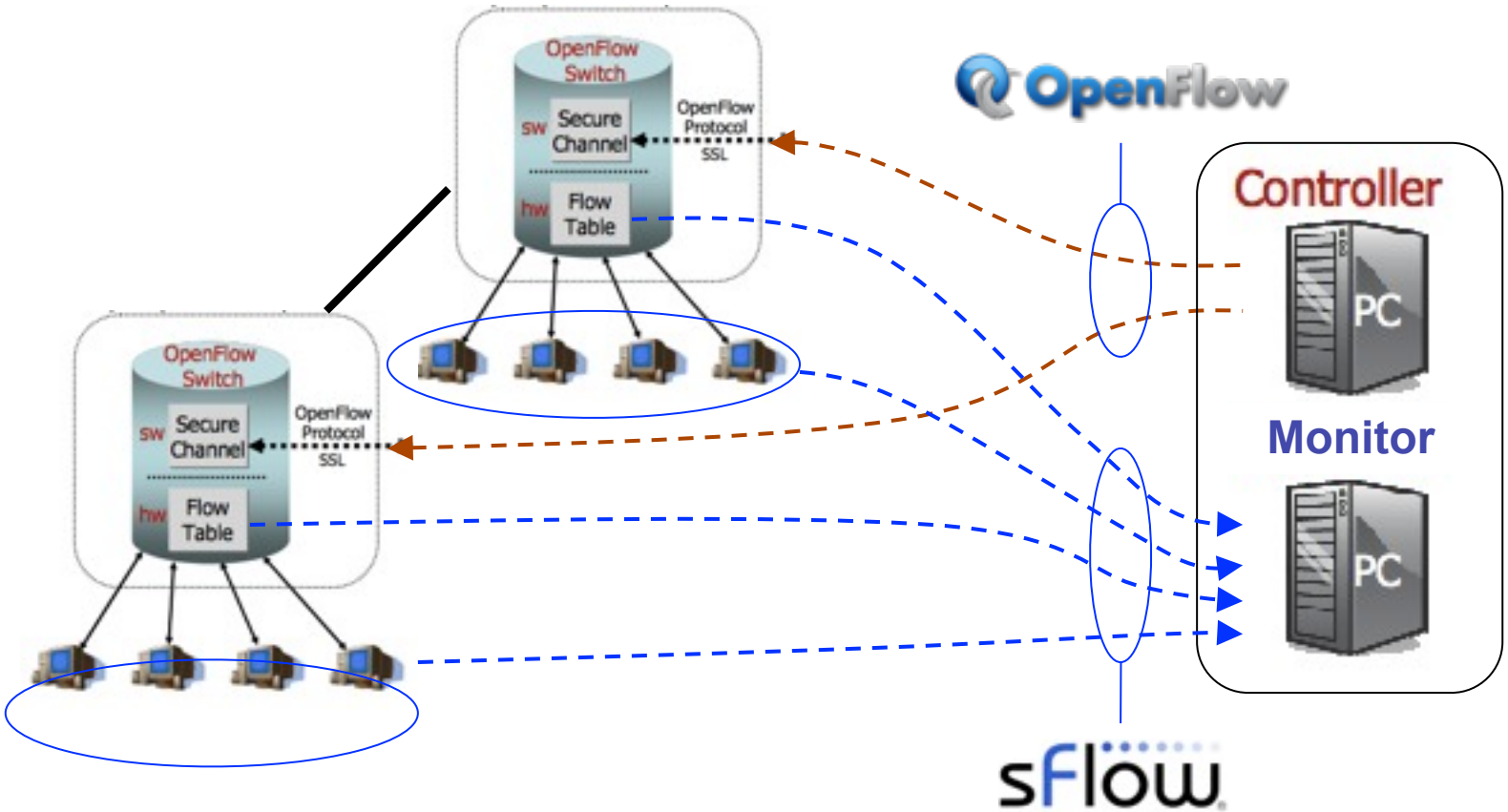


OpenFlow + sFlow = Adaptive Control



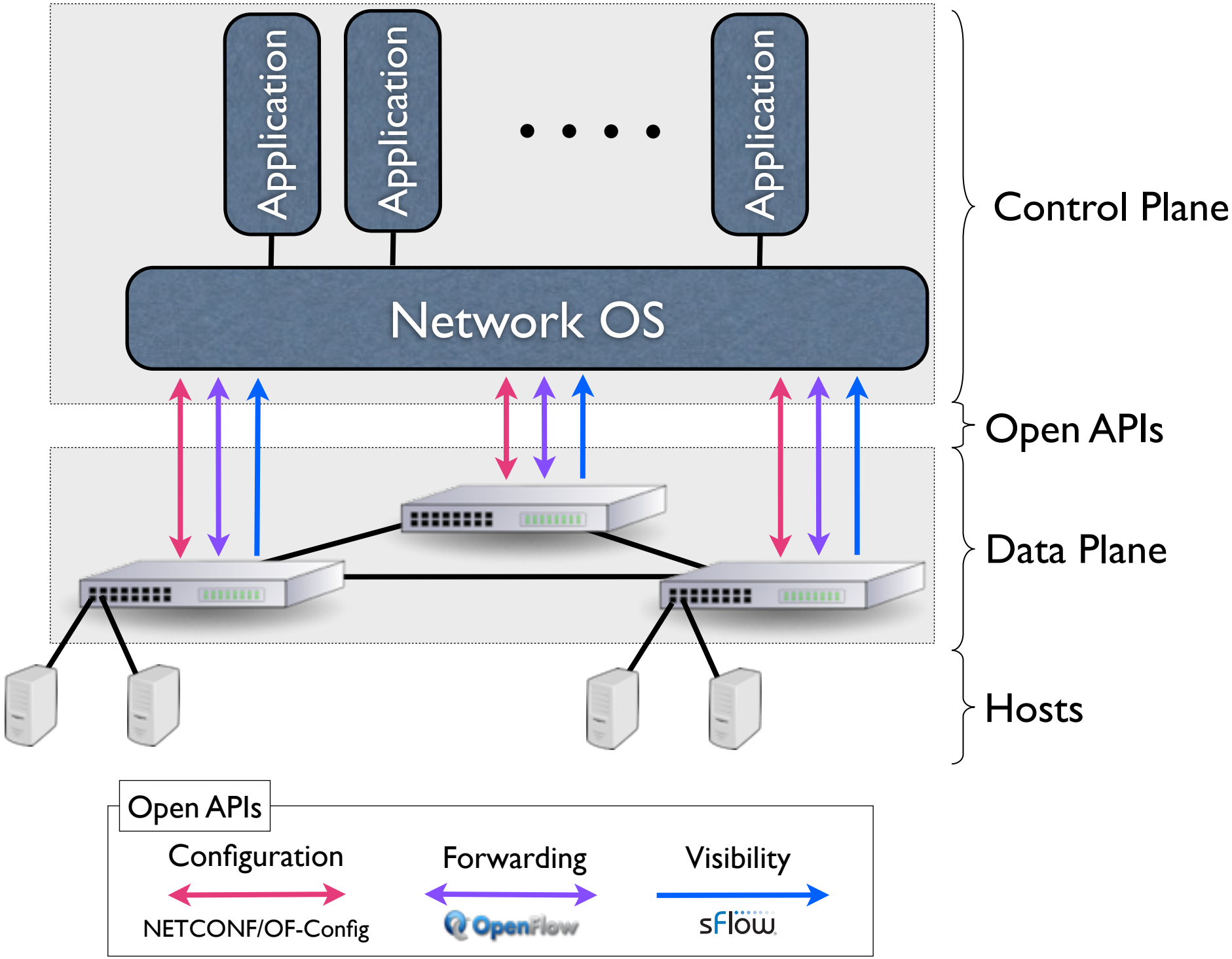
- Detailed, low-latency measurements from sFlow allows OpenFlow controller to adapt network to changing traffic patterns (load balancing, DDoS mitigation etc.).

OpenFlow + sFlow = Adaptive Control

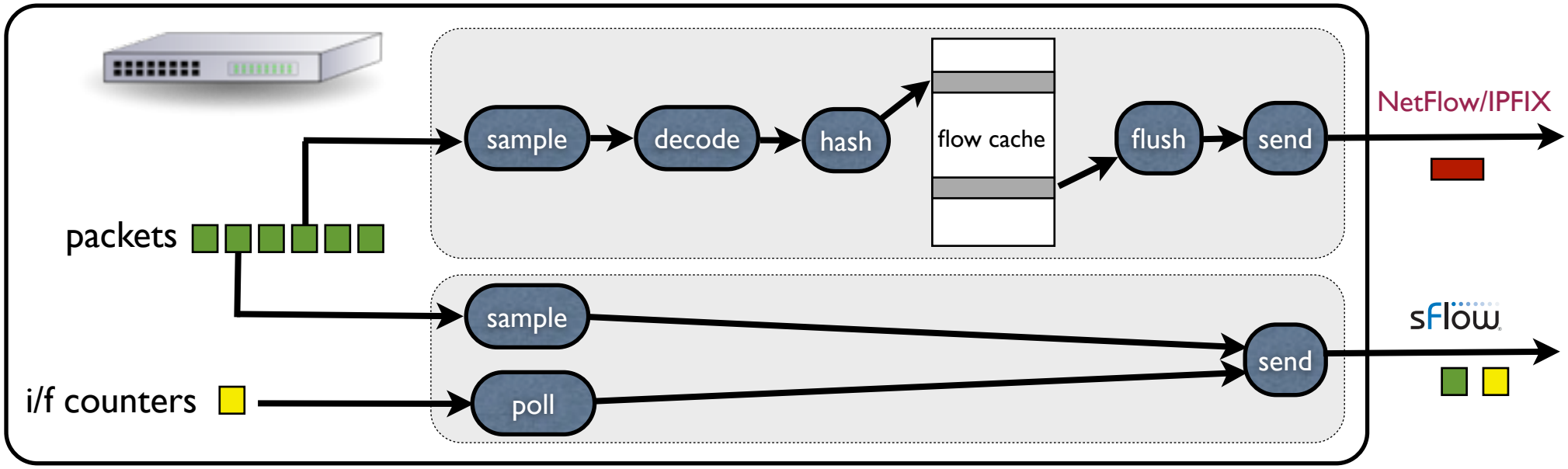


- Detailed, low-latency measurements from sFlow allows OpenFlow controller to adapt network to changing traffic patterns (load balancing, DDoS mitigation etc.).

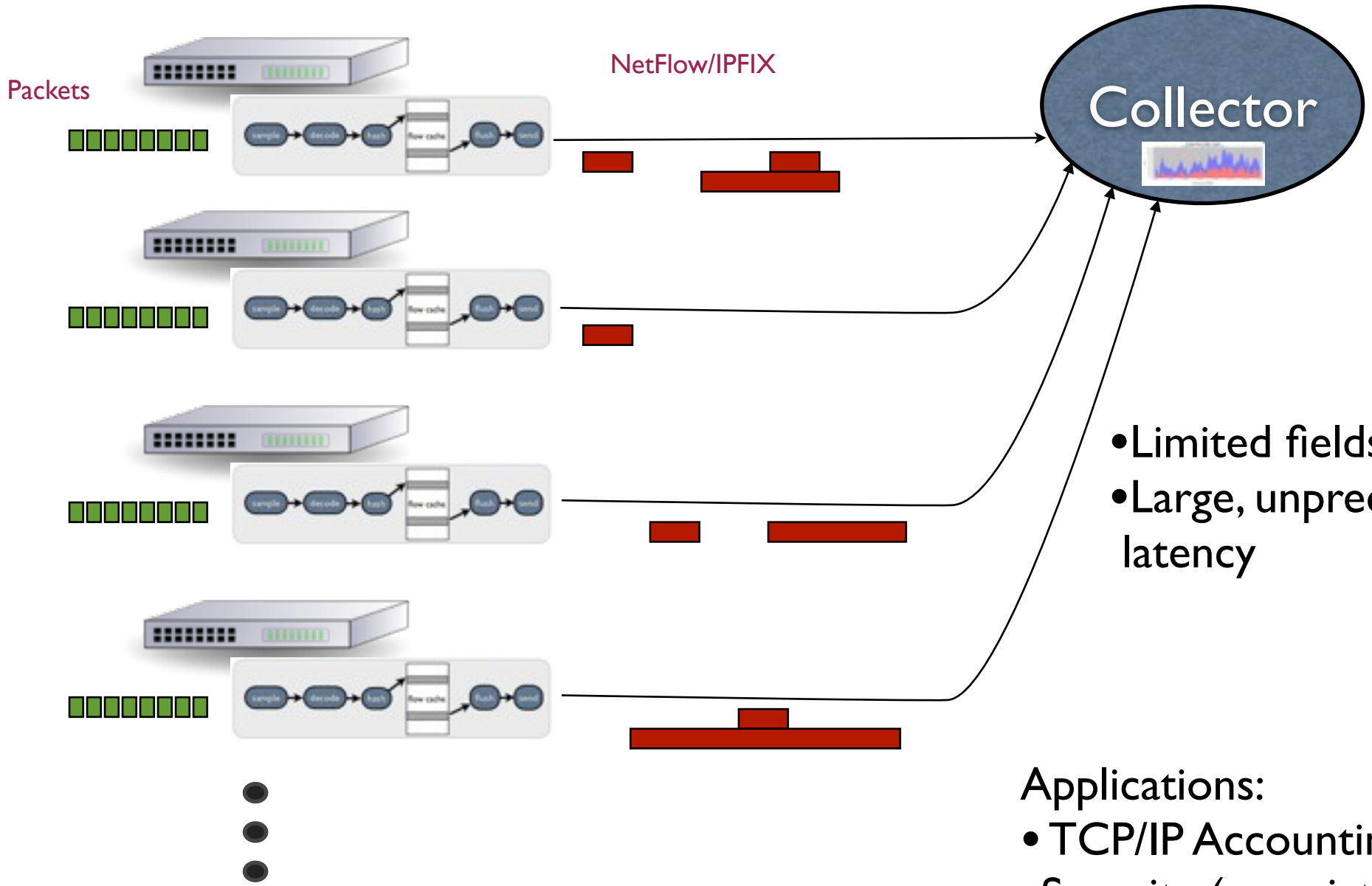
- OpenFlow can be optimized for efficiency (e.g. by using wildcards), sFlow provides visibility to detect and manage large flows.


Application
Application
Application
Network OS
Control Plane
Open APIs
Data Plane
Hosts
Open APIs
Configuration
NETCONF/OF-Config
Forwarding
OpenFlow
Visibility
sFlow

sFlow vs NetFlow/IPFIX



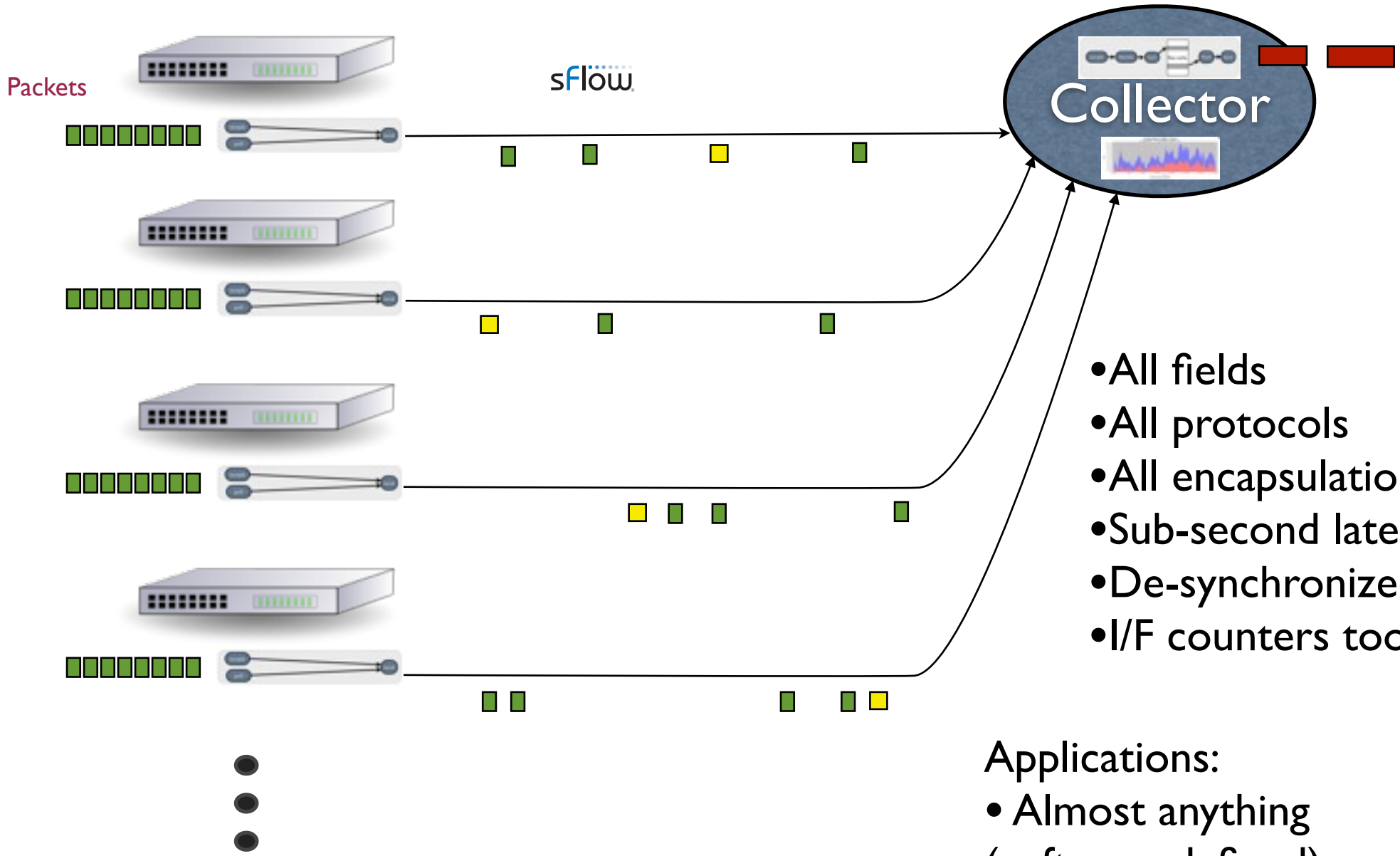
sFlow vs NetFlow/IPFIX : system-wide NetFlow/IPFIX



- Limited fields
- Large, unpredictable latency

Applications:
- TCP/IP Accounting
- Security (associations)

sFlow vs NetFlow/IPFIX : system-wide sFlow

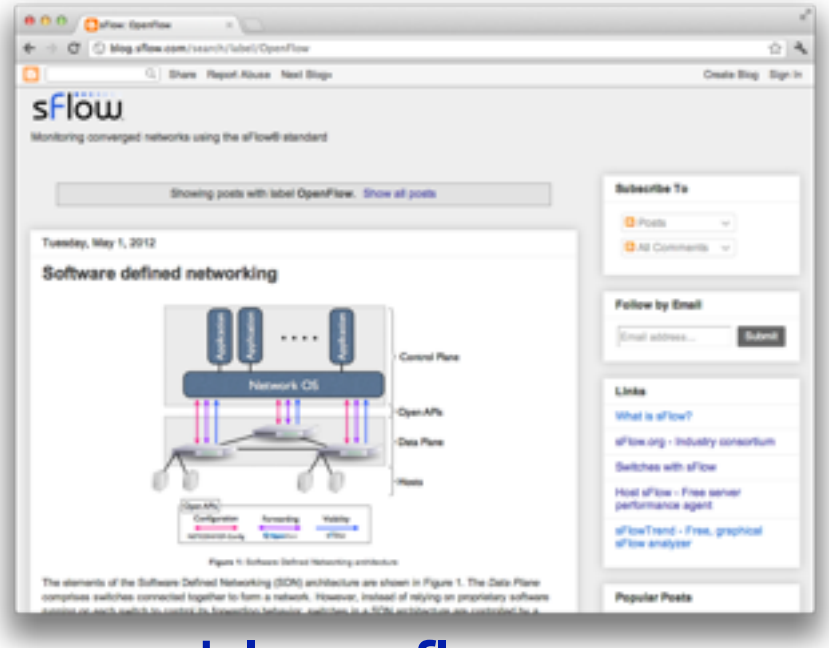


- All fields
- All protocols
- All encapsulations
- Sub-second latency
- De-synchronized
- I/F counters too

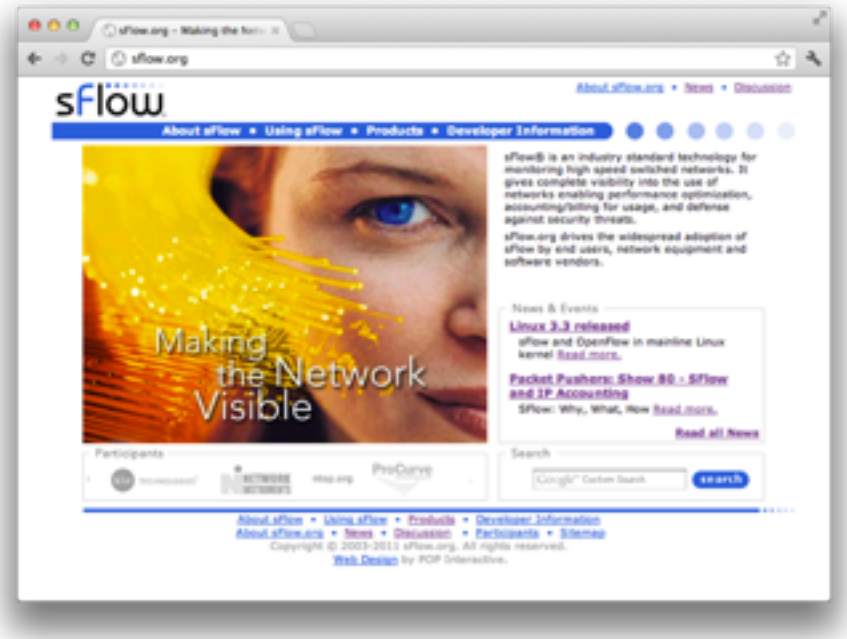
Applications:
- Almost anything (software defined)

More Information

host-sflow.sourceforge.net



blog.sflow.com



sflow.org