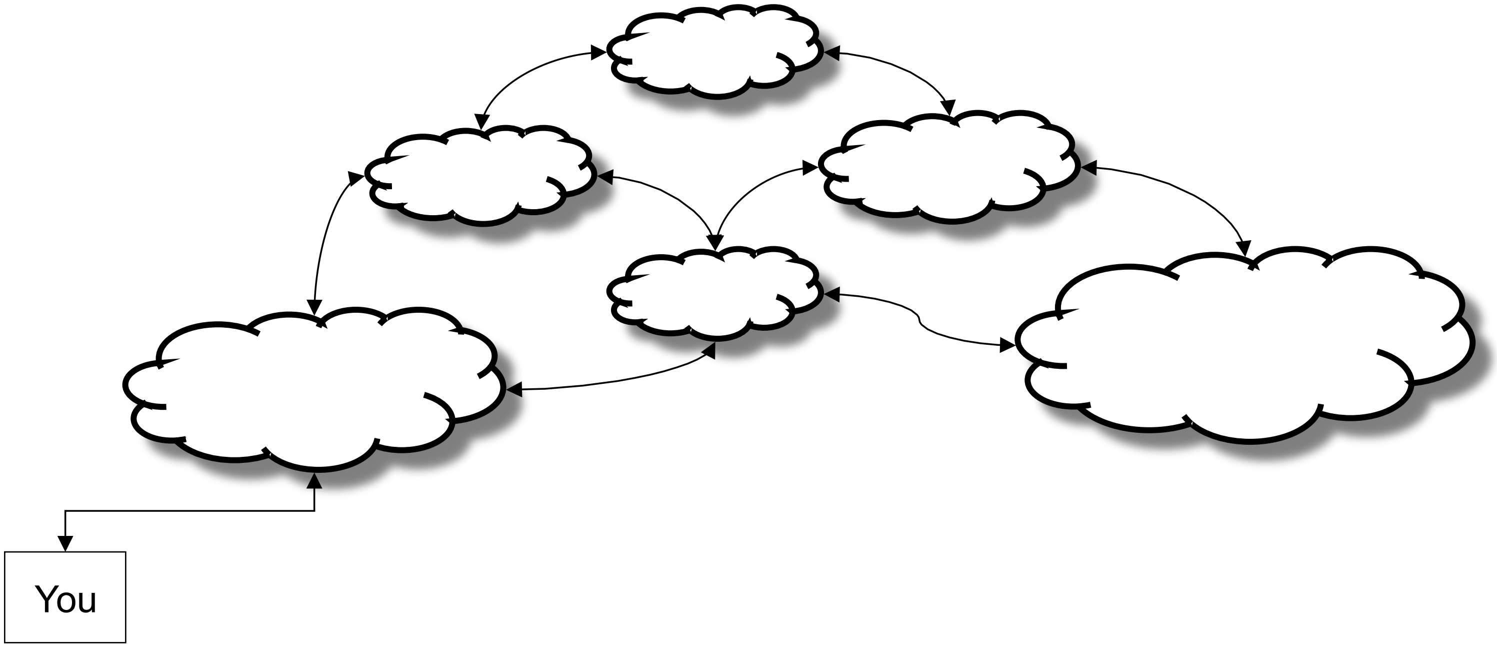
How the Internet sees you

Demonstrating what activities most ISPs see you doing on the Internet

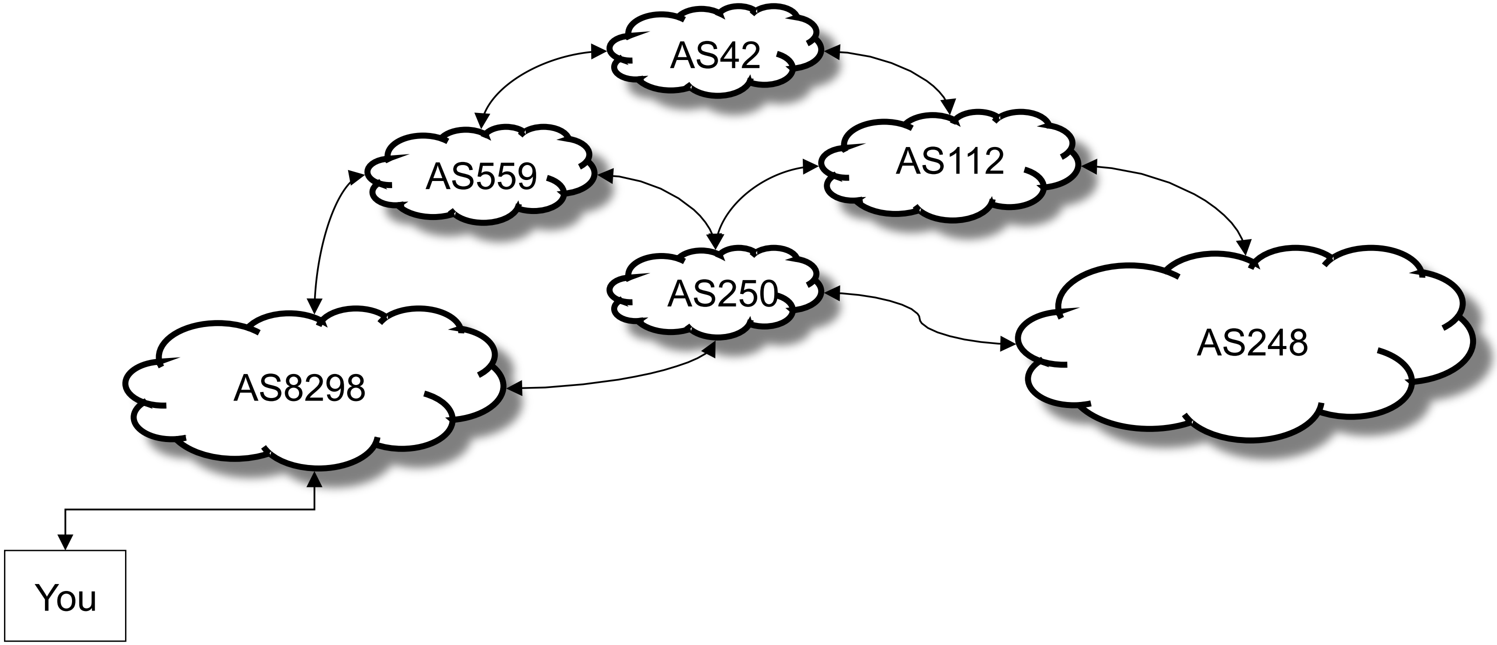
Jeroen Massar <jma@zurich.ibm.com>

Network of networks

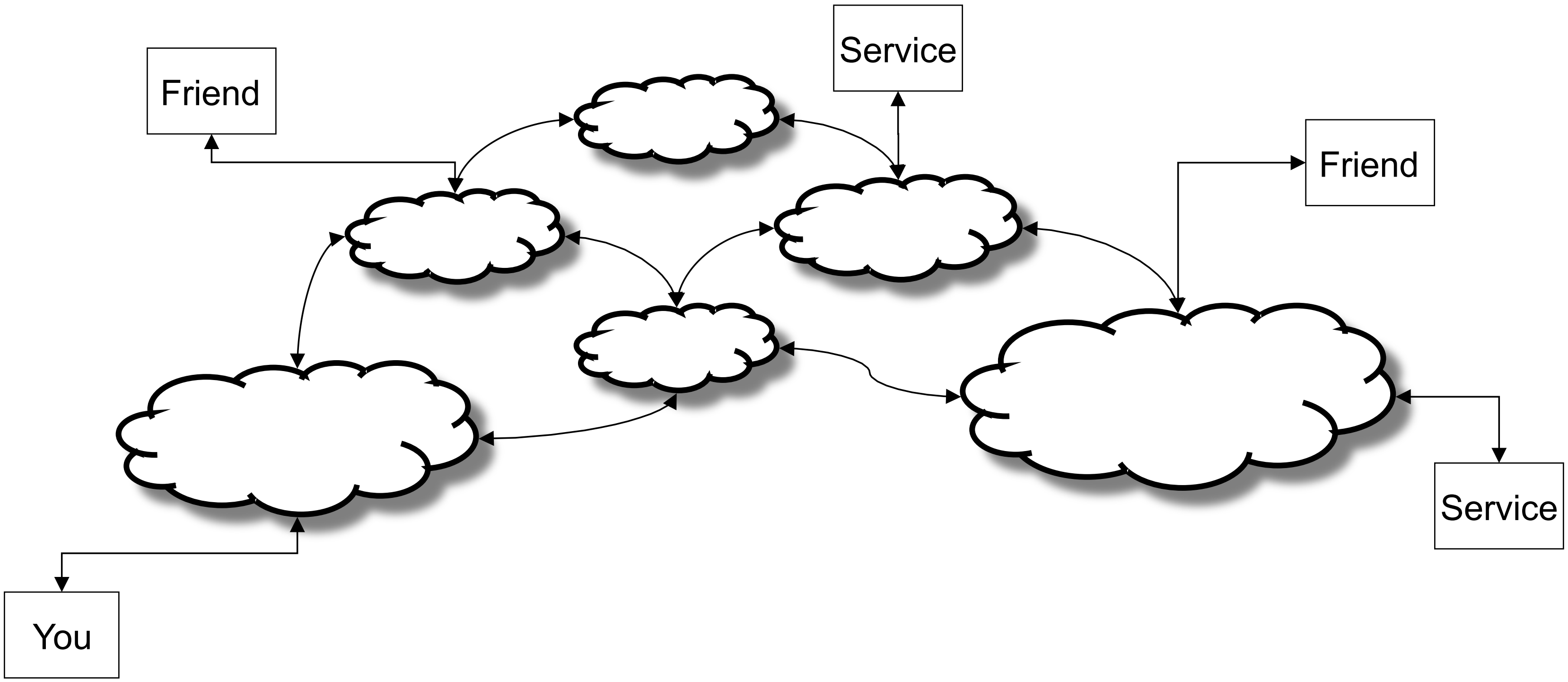


Autonomous Systems

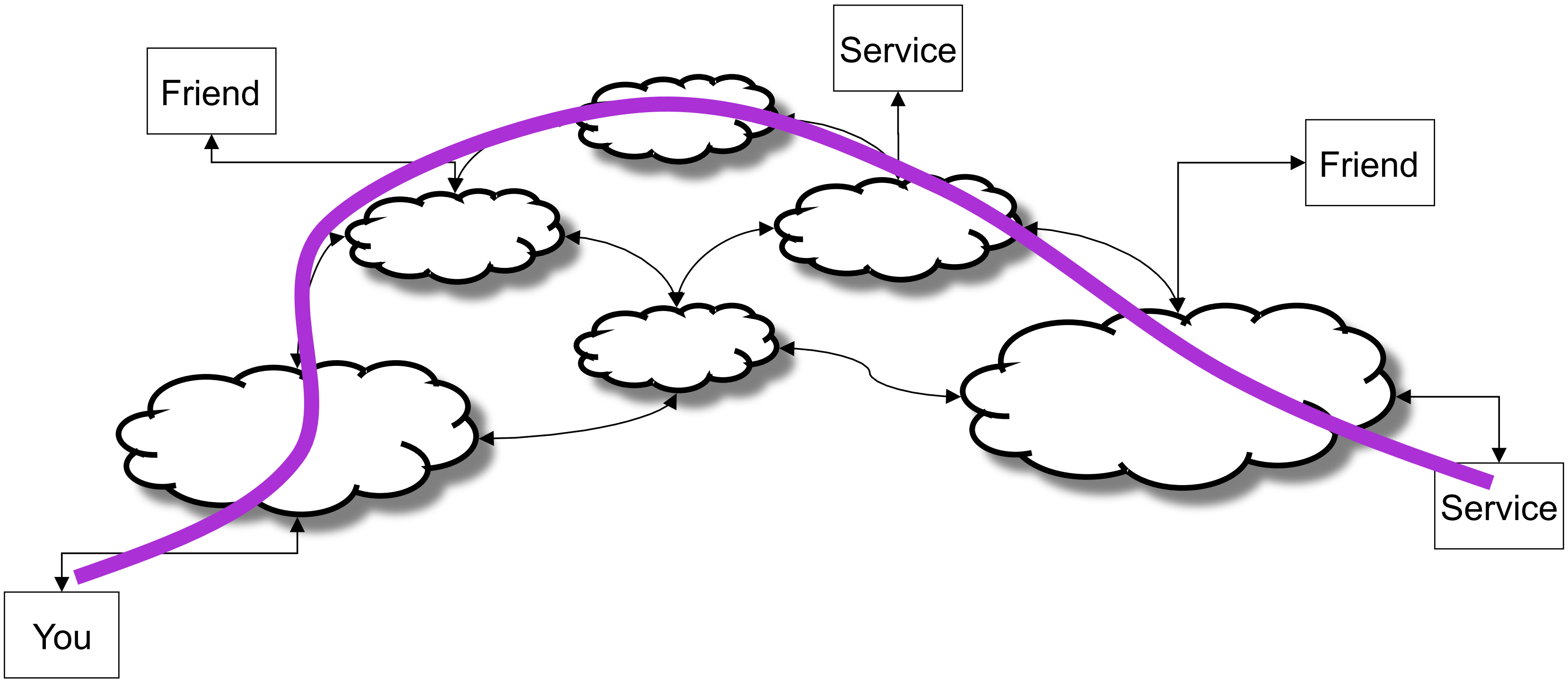
- AS = network operated under a single policy



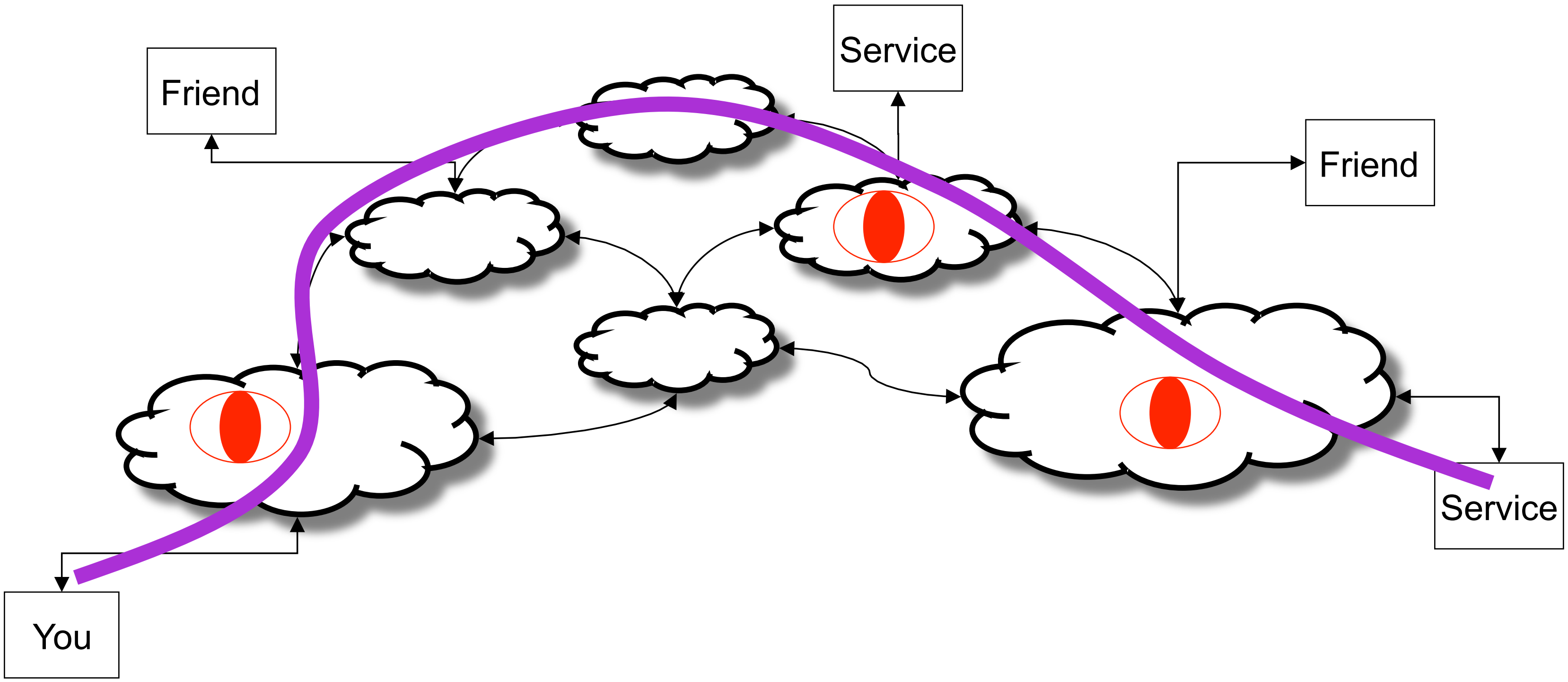
Services and friends are all over the place



When you communicate you pass those networks



They keep their eyes open…



Some quick notes

- Networks can see what is in their network
- They can’t see what happens in another network
  - … though if packets cross their network they do
  - … unless they cooperate
  - … or some organization requires them to share
- Forward and reverse path for packets might be asymmetric

TAP / Mirror port

- Optical splitters on fibers or implemented in the switch/router to copy all traffic to another port

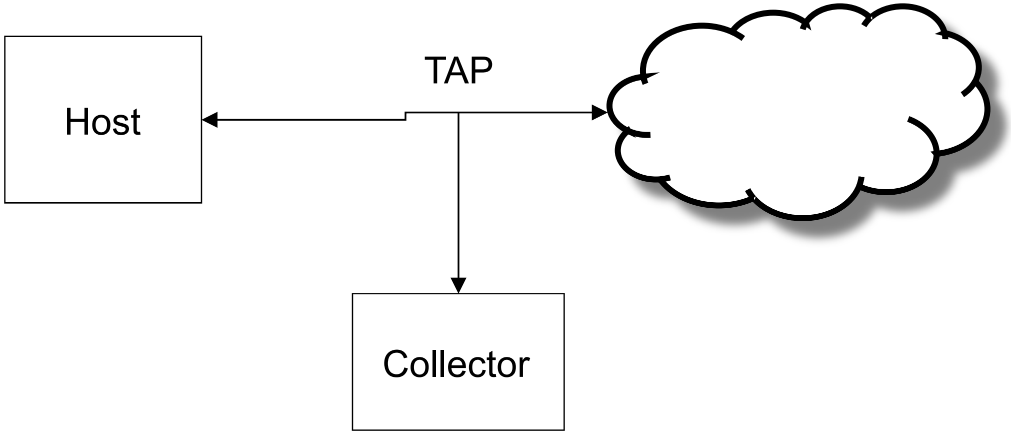
Pro:

- See everything

Con:

- Store and analyze it all

  (unless you filter what you (don’t) want)



A Flow

“A Flow is defined as a set of IP packets passing an Observation Point in the network during a certain time interval” (RFC5101)

Effectively: ip_src : port_src -> ip_dst : port_src

NetFlow

- Originally intended as a way to make routing faster
- Versions v1, v5, v6, v7, v8, v9, IPFIX (IETF)
- Up to version 8 static templates
- Version 9 + IPFIX (v10) have variable templates
- IPFIX has ‘enterprise’ information elements allowing any kind of data

Pro:

- Much lower data rate and thus also analysis and storage requirements

Con:

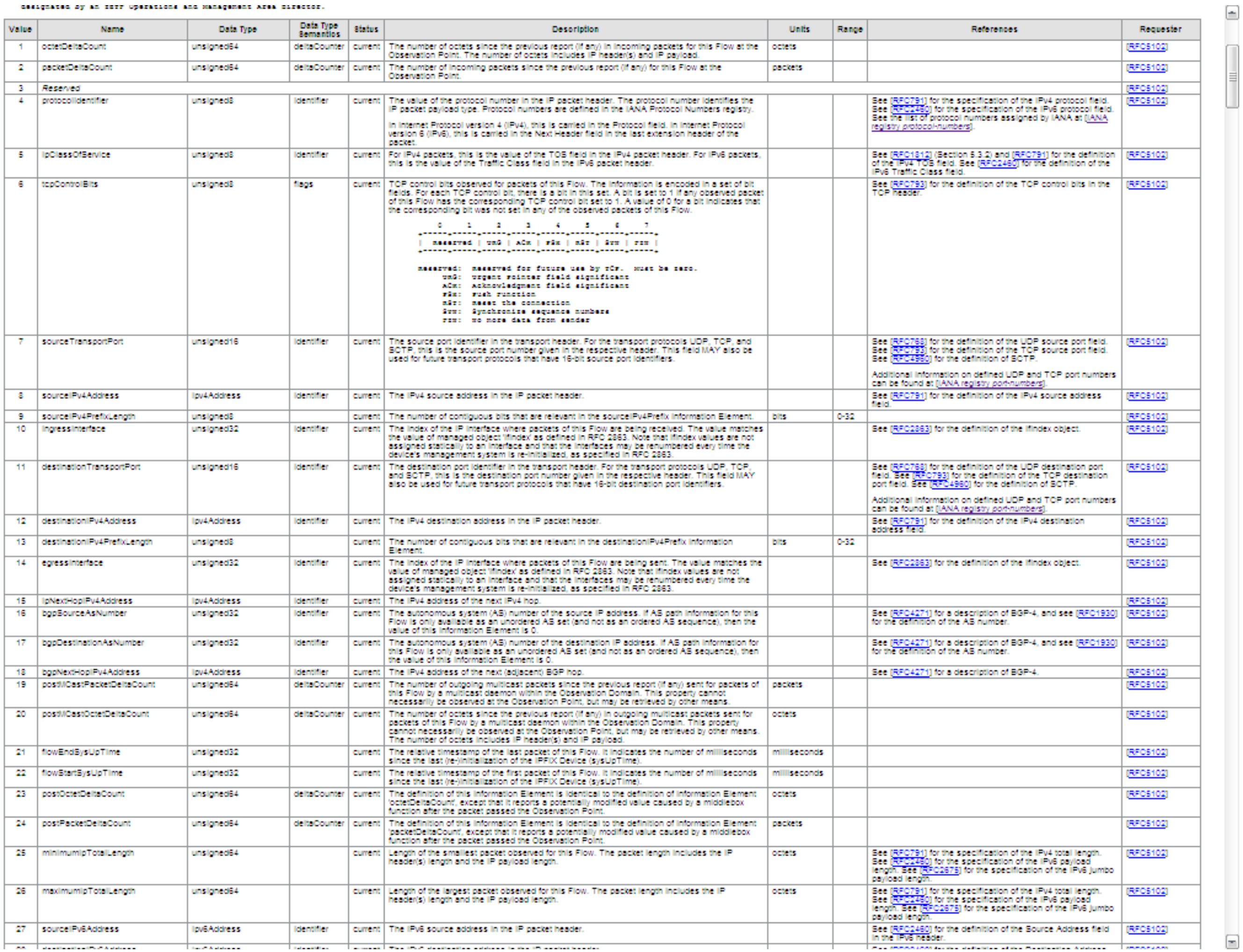
- No packet contents, just header summary or fields that are selected which then generally are summaries
- Higher overhead on the collector as it needs to keep big flow tables

Could do sampling, but not nicely supported.

NetFlow v5

```
/* NetFlow Version 5 Record Format */
struct NFv5R
{
        uint32_t        ip_src;         /* Source IP address */
        uint32_t        ip_dst;         /* Destination IP address */
        uint32_t        ip_nxt;         /* IP address of the next hop router */
        uint16_t        iface_in;       /* SNMP index of the input interface */
        uint16_t        iface_out;      /* SNMP index of the output interface */
        uint32_t        packets;        /* Packets in the flow */
        uint32_t        octets;         /* Total number of Layer 3 bytes */
        uint32_t        first;          /* SysUptime at start of flow */
        uint32_t        last;           /* SysUptime when the last packet was rcvd */
        uint16_t        port_src;       /* TCP/UDP source port number */
        uint16_t        port_dst;       /* TCP/UDP destination port number */
        uint8_t         pad1;           /* Unused */
        uint8_t         tcp_flags;      /* Cumulative OR of TCP flags */
        uint8_t         protocol;       /* IP protocol */
        uint8_t         tos;            /* IP ToS */
        uint16_t        asn_src;        /* AS of the source address */
        uint16_t        asn_dst;        /* AS of the destination address */
        uint8_t         ip_src_mask;    /* Source address prefix mask bits */
        uint8_t         ip_dst_mask;    /* Destination address prefix mask bits */
        uint16_t        pad2;
} PACKED;
```

NetFlow v9 / IPFIX uses “Information Elements”



designated by an IETF Operations and Management Area director.

Value	Name	Data Type	Data Type Semantics	Status	Description	Units	Range	References	Requester
1	octetDeltaCount	unsigned64	deltaCounter	current	The number of octets since the previous report (if any) in incoming packets for this Flow at the Observation Point. The number of octets includes IP header(s) and IP payload.	octets			[RFC5102]
2	packetDeltaCount	unsigned64	deltaCounter	current	The number of incoming packets since the previous report (if any) for this Flow at the Observation Point.	packets			[RFC5102]
3	*Reserved*								[RFC5102]
4	protocolIdentifier	unsigned8	identifier	current	The value of the protocol number in the IP packet header. The protocol number identifies the IP packet payload type. Protocol numbers are defined in the IANA Protocol Numbers registry. In Internet Protocol version 4 (IPv4), this is carried in the Protocol field. In Internet Protocol version 6 (IPv6), this is carried in the Next Header field in the last extension header of the packet.			See [RFC791] for the specification of the IPv4 protocol field. See [RFC2460] for the specification of the IPv6 protocol field. See the list of protocol numbers assigned by IANA at [IANA registry protocol-numbers].	[RFC5102]
5	ipClassOfService	unsigned8	identifier	current	For IPv4 packets, this is the value of the TOS field in the IPv4 packet header. For IPv6 packets, this is the value of the Traffic Class field in the IPv6 packet header.			See [RFC1812] (Section 5.3.2) and [RFC791] for the definition of the IPv4 TOS field. See [RFC2460] for the definition of the IPv6 Traffic Class field.	[RFC5102]

6	tcpControlBits	unsigned8	flags	current	TCP control bits observed for packets of this Flow. The information is encoded in a set of bit fields. For each TCP control bit, there is a bit in this set. A bit is set to 1 if any observed packet of this Flow has the corresponding TCP control bit set to 1. A value of 0 for a bit indicates that the corresponding bit was not set in any of the observed packets of this Flow.
					`0 1 2 3 4 5 6 7`
					`+-----+-----+-----+-----+-----+-----+-----+-----+`
					`| Reserved | URG | ACK | PSH | RST | SYN | FIN |`
					`+-----+-----+-----+-----+-----+-----+-----+-----+`
					`Reserved: Reserved for future use by TCP. Must be zero.`
					`URG: Urgent Pointer field significant`
					`ACK: Acknowledgment field significant`
					`PSH: Push Function`
					`RST: Reset the connection`
					`SYN: Synchronize sequence numbers`
					`FIN: No more data from sender`
								See [RFC793] for the definition of the TCP control bits in the TCP header.	[RFC5102]

7	sourceTransportPort	unsigned16	identifier	current	The source port identifier in the transport header. For the transport protocols UDP, TCP, and SCTP, this is the source port number given in the respective header. This field MAY also be used for future transport protocols that have 16-bit source port identifiers.			See [RFC768] for the definition of the UDP source port field. See [RFC793] for the definition of the TCP source port field. See [RFC4960] for the definition of SCTP. Additional information on defined UDP and TCP port numbers can be found at [IANA registry port-numbers].	[RFC5102]
8	sourceIPv4Address	ipv4Address	identifier	current	The IPv4 source address in the IP packet header.			See [RFC791] for the definition of the IPv4 source address field.	[RFC5102]
9	sourceIPv4PrefixLength	unsigned8		current	The number of contiguous bits that are relevant in the sourceIPv4Prefix Information Element.	bits	0-32		[RFC5102]
10	ingressInterface	unsigned32	identifier	current	The index of the IP interface where packets of this Flow are being received. The value matches the value of managed object 'ifIndex' as defined in RFC 2863. Note that ifIndex values are not assigned statically to an interface and that the interfaces may be renumbered every time the device's management system is re-initialized, as specified in RFC 2863.			See [RFC2863] for the definition of the ifIndex object.	[RFC5102]
11	destinationTransportPort	unsigned16	identifier	current	The destination port identifier in the transport header. For the transport protocols UDP, TCP, and SCTP, this is the destination port number given in the respective header. This field MAY also be used for future transport protocols that have 16-bit destination port identifiers.			See [RFC768] for the definition of the UDP destination port field. See [RFC793] for the definition of the TCP destination port field. See [RFC4960] for the definition of SCTP. Additional information on defined UDP and TCP port numbers can be found at [IANA registry port-numbers].	[RFC5102]
12	destinationIPv4Address	ipv4Address	identifier	current	The IPv4 destination address in the IP packet header.			See [RFC791] for the definition of the IPv4 destination address field.	[RFC5102]
13	destinationIPv4PrefixLength	unsigned8		current	The number of contiguous bits that are relevant in the destinationIPv4Prefix Information Element.	bits	0-32		[RFC5102]
14	egressInterface	unsigned32	identifier	current	The index of the IP interface where packets of this Flow are being sent. The value matches the value of managed object 'ifIndex' as defined in RFC 2863. Note that ifIndex values are not assigned statically to an interface and that the interfaces may be renumbered every time the device's management system is re-initialized, as specified in RFC 2863.			See [RFC2863] for the definition of the ifIndex object.	[RFC5102]
15	ipNextHopIPv4Address	ipv4Address	identifier	current	The IPv4 address of the next IPv4 hop.				[RFC5102]
16	bgpSourceAsNumber	unsigned32	identifier	current	The autonomous system (AS) number of the source IP address. If AS path information for this Flow is only available as an unordered AS set (and not as an ordered AS sequence), then the value of this Information Element is 0.			See [RFC4271] for a description of BGP-4, and see [RFC1930] for the definition of the AS number.	[RFC5102]
17	bgpDestinationAsNumber	unsigned32	identifier	current	The autonomous system (AS) number of the destination IP address. If AS path information for this Flow is only available as an unordered AS set (and not as an ordered AS sequence), then the value of this Information Element is 0.			See [RFC4271] for a description of BGP-4, and see [RFC1930] for the definition of the AS number.	[RFC5102]
18	bgpNextHopIPv4Address	ipv4Address	identifier	current	The IPv4 address of the next (adjacent) BGP hop.			See [RFC4271] for a description of BGP-4.	[RFC5102]
19	postMCastPacketDeltaCount	unsigned64	deltaCounter	current	The number of outgoing multicast packets since the previous report (if any) sent for packets of this Flow by a multicast daemon within the Observation Domain. This property cannot necessarily be observed at the Observation Point, but may be retrieved by other means.	packets			[RFC5102]
20	postMCastOctetDeltaCount	unsigned64	deltaCounter	current	The number of octets since the previous report (if any) in outgoing multicast packets sent for packets of this Flow by a multicast daemon within the Observation Domain. This property cannot necessarily be observed at the Observation Point, but may be retrieved by other means. The number of octets includes IP header(s) and IP payload.	octets			[RFC5102]
21	flowEndSysUpTime	unsigned32		current	The relative timestamp of the last packet of this Flow. It indicates the number of milliseconds since the last (re-)initialization of the IPFIX Device (sysUpTime).	milliseconds			[RFC5102]
22	flowStartSysUpTime	unsigned32		current	The relative timestamp of the first packet of this Flow. It indicates the number of milliseconds since the last (re-)initialization of the IPFIX Device (sysUpTime).	milliseconds			[RFC5102]
23	postOctetDeltaCount	unsigned64	deltaCounter	current	The definition of this Information Element is identical to the definition of Information Element 'octetDeltaCount', except that it reports a potentially modified value caused by a middlebox function after the packet passed the Observation Point.	octets			[RFC5102]
24	postPacketDeltaCount	unsigned64	deltaCounter	current	The definition of this Information Element is identical to the definition of Information Element 'packetDeltaCount', except that it reports a potentially modified value caused by a middlebox function after the packet passed the Observation Point.	packets			[RFC5102]
25	minimumIpTotalLength	unsigned64		current	Length of the smallest packet observed for this Flow. The packet length includes the IP header(s) length and the IP payload length.	octets		See [RFC791] for the specification of the IPv4 total length. See [RFC2460] for the specification of the IPv6 payload length. See [RFC2675] for the specification of the IPv6 jumbo payload length.	[RFC5102]
26	maximumIpTotalLength	unsigned64		current	Length of the largest packet observed for this Flow. The packet length includes the IP header(s) length and the IP payload length.	octets		See [RFC791] for the specification of the IPv4 total length. See [RFC2460] for the specification of the IPv6 payload length. See [RFC2675] for the specification of the IPv6 jumbo payload length.	[RFC5102]
27	sourceIPv6Address	ipv6Address	identifier	current	The IPv6 source address in the IP packet header.			See [RFC2460] for the definition of the Source Address field in the IPv6 header.	[RFC5102]

http://www.iana.org/assignments/ipfix/ipfix.xhtml

NetFlow v9 / IPFIX

Bits 0..15	**Bits 16..31**
Version = 0x000a	Message Length = 64 Bytes
Export Timestamp = 2005-12-31 23:59:60	
Sequence Number = 0	
Source ID = 12345678	
Set ID = 2 (Template)	Set Length = 20 Bytes
Template ID = 256	Number of Fields = 3
Typ = sourceIPv4Address	Field Length = 4 Bytes
Typ = destinationIPv4Address	Field Length = 4 Bytes
Typ = packetDeltaCount	Field Length = 4 Bytes
Set ID = 256 (Data Set using Template 256)	Set Length = 28 Bytes
Record 1, Field 1 = 192.168.0.201	
Record 1, Field 2 = 192.168.0.1	
Record 1, Field 3 = 235 Packets	
Record 2, Field 1 = 192.168.0.202	
Record 2, Field 2 = 192.168.0.1	
Record 2, Field 3 = 42 Packets	

Storage requirements for NetFlow / IPFIX

	Flow Rate	**NetFlow Volume**	**Data Volume**
Small Network	<100 flows/s	<260 MiB/d	<260 MiB/d
300 People Site	300 flows/s	800 MiB/d	200 GiB/d
Single Core Router	20000 flows/s	100 GiB/d	8 TiB/d
Large ISP	2 M flows/s	4 TiB/d	2 PiB/d

sFlow

- InMon Corporation standard
- Makes “samples” of the network traffic, thus eg 1 out of 4000 packets
- Carries the first portion of the Ethernet/IPv4/IPv6 packet
- Not accurate for perfect account, but a pretty good guess
- Supported by Foundry, Extreme, Force10
- Primarily targeted as a replacement of RMON/NetFlow v5
- Can be used for counters

Pro:

- Sampled thus much smaller portion of data
- Low overhead in the implementation on the router

Con:

- Higher overhead on the collector (and quite a bloated protocol)
- Might just miss what you wanted to see due to sampling

Passive DNS

- Idea by Florian Weimar
- Log DNS queries and answers (as they are not crypted)
- Get a very good overview of what DNS questions are being asked
- Can detect previously undetected DNS labels, don’t need to AXFR a domain for this

Normally.,.

… these tools are used for accounting/billing based on traffic volumes
… or tracing abuse.

But they can also be abused for other things

Putting it all together

Using one of or a combination of TAP, NetFlow or sFlow.

Add to that Passive DNS as then we get a better overview of what ‘name’ that corresponds to the IP address one is talking to

We now have:

- Knowledge of what IP address talks to what IP address
- What port numbers and protocols are being used
- In most cases what hostname belongs to the IP address

Digital Fingerprint

The browser identity:

- Cookies
- Plugin lists
- and way more: https://panopticlick.eff.org/

An ISP would have to look inside the packets and reconstruct TCP to be able to see the details in there and of course when it is crypted (TLS etc) then they won't be able to get to it.

Digital Profiling

People tend to use a restricted set of services

- The common set: Twitter, Facebook, Gmail, etc

But the bigger issue is that one has auto-update services:

- These connect every day, week, other period to their services

Because of that and the combination with Passive DNS, one can thus derive from the NetFlow data who you are talking to, and thus there is a very nice profile of who you are, even if you move around through the world…

Our little 27C3 experiment

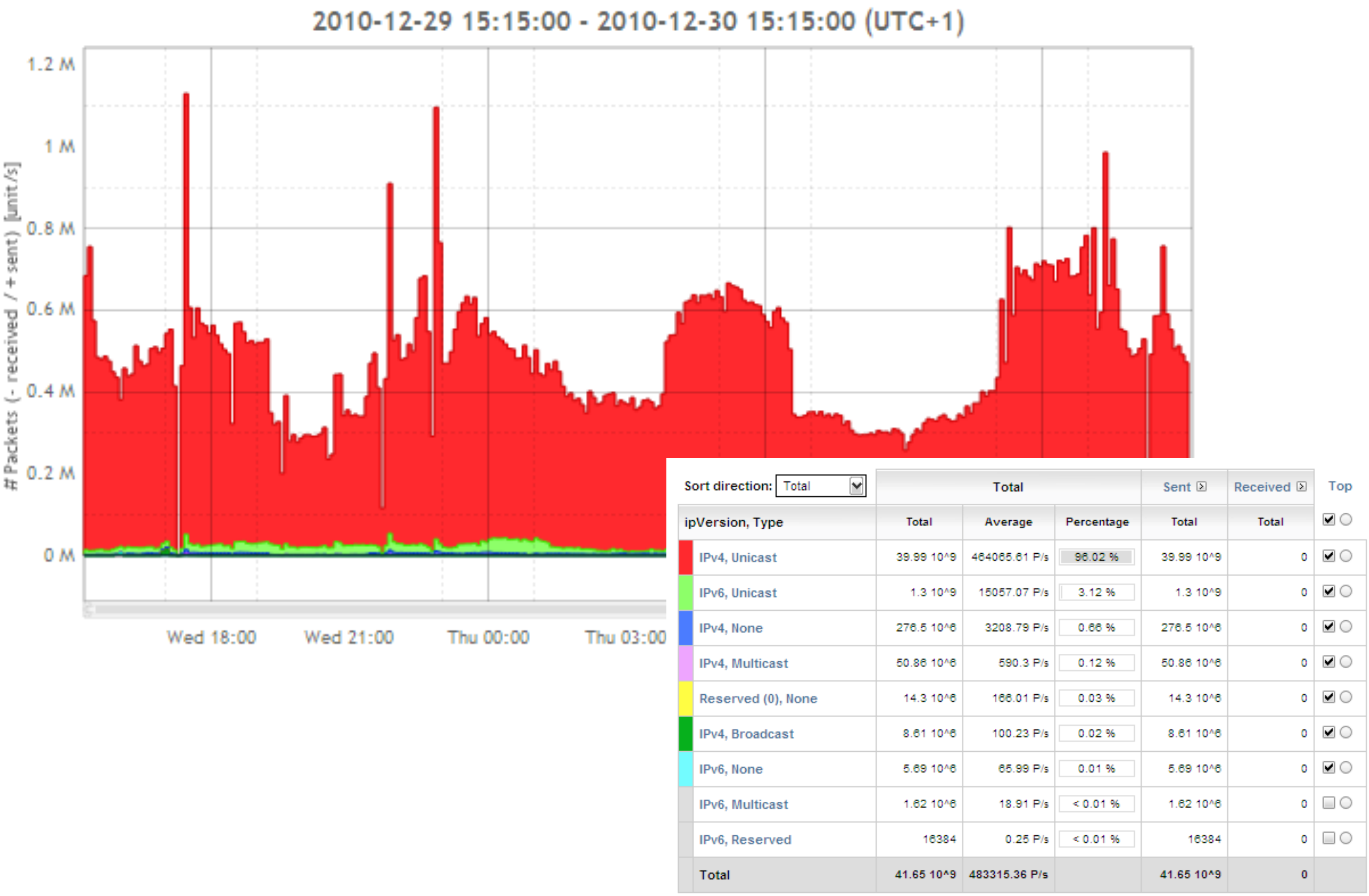
- Set up our Anaphera tool (an NetFlow / IPFIX / sFlow collector & analyzer)
- Send sFlow from the router which connects the 27C3 congress network to the Internet

The restrictions:

- Anonymize IP addresses
- sFlow… we only get 1/4000 packets
- Don’t store anything (well, we keep the graphs)

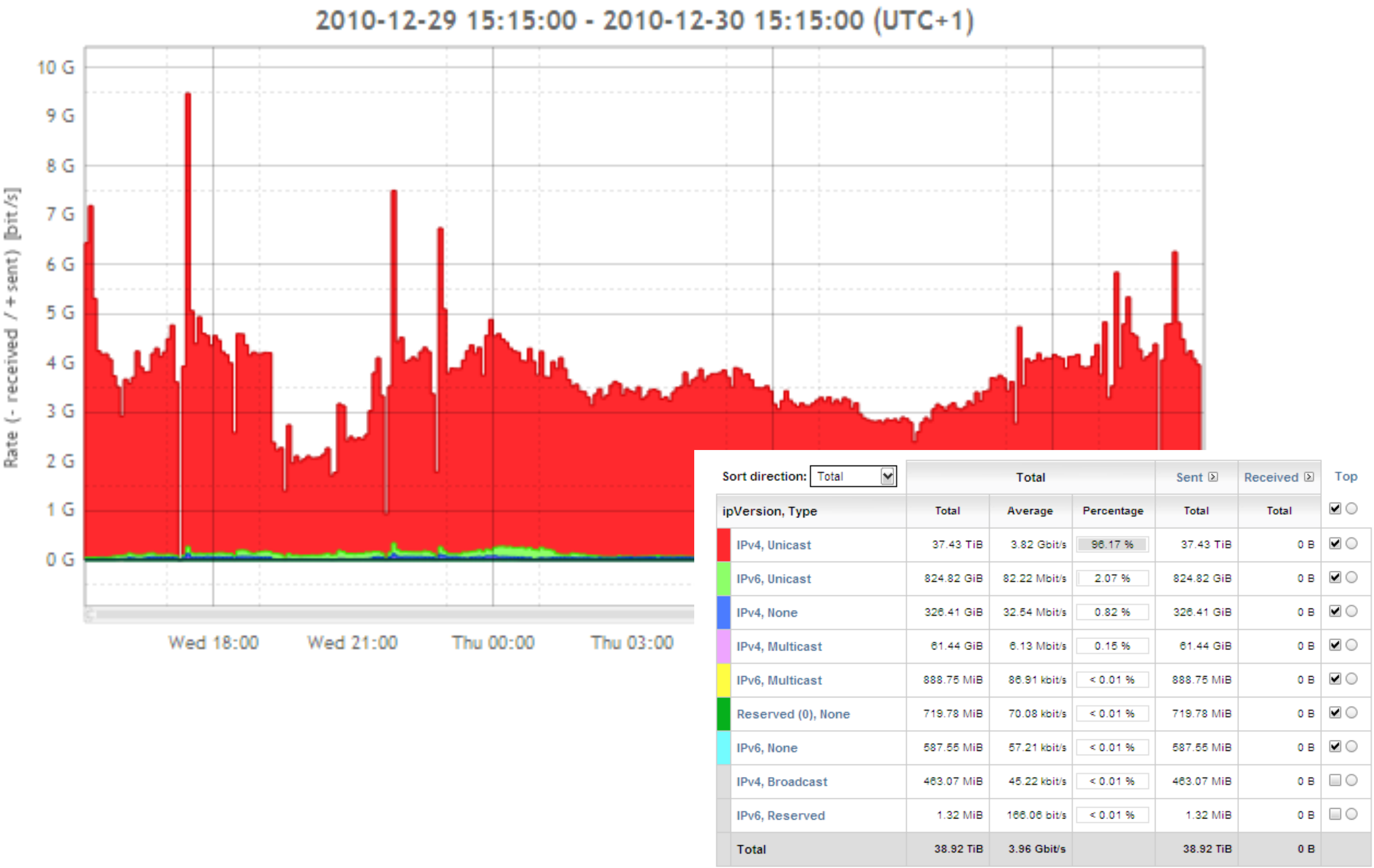
as such we could not perform the nice tricks that we just discussed, be happy ;)

Packets



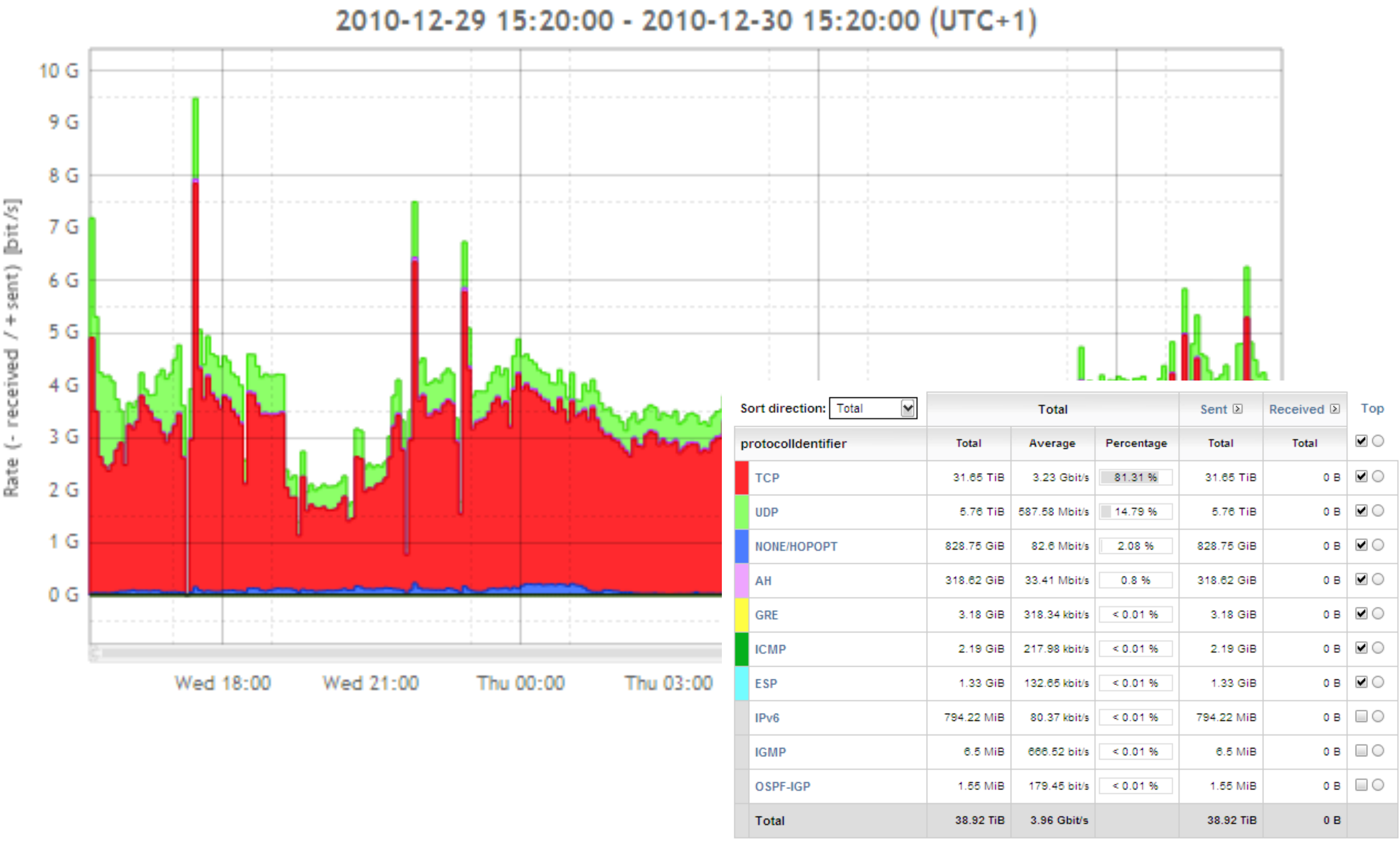
Sort direction: Total		Total			Sent ⊡	Received ⊡	Top
ipVersion, Type		Total	Average	Percentage	Total	Total	☑ ○
🟥	IPv4, Unicast	39.99 10^9	464065.61 P/s	96.02 %	39.99 10^9	0	☑ ○
🟩	IPv6, Unicast	1.3 10^9	15057.07 P/s	3.12 %	1.3 10^9	0	☑ ○
🟦	IPv4, None	276.5 10^6	3208.79 P/s	0.66 %	276.5 10^6	0	☑ ○
🟪	IPv4, Multicast	50.86 10^6	590.3 P/s	0.12 %	50.86 10^6	0	☑ ○
🟨	Reserved (0), None	14.3 10^6	166.01 P/s	0.03 %	14.3 10^6	0	☑ ○
🟩	IPv4, Broadcast	8.61 10^6	100.23 P/s	0.02 %	8.61 10^6	0	☑ ○
🟦	IPv6, None	5.69 10^6	65.99 P/s	0.01 %	5.69 10^6	0	☑ ○
	IPv6, Multicast	1.62 10^6	18.91 P/s	< 0.01 %	1.62 10^6	0	□ ○
	IPv6, Reserved	16384	0.25 P/s	< 0.01 %	16384	0	□ ○
	Total	41.65 10^9	483315.36 P/s		41.65 10^9	0	

Octets

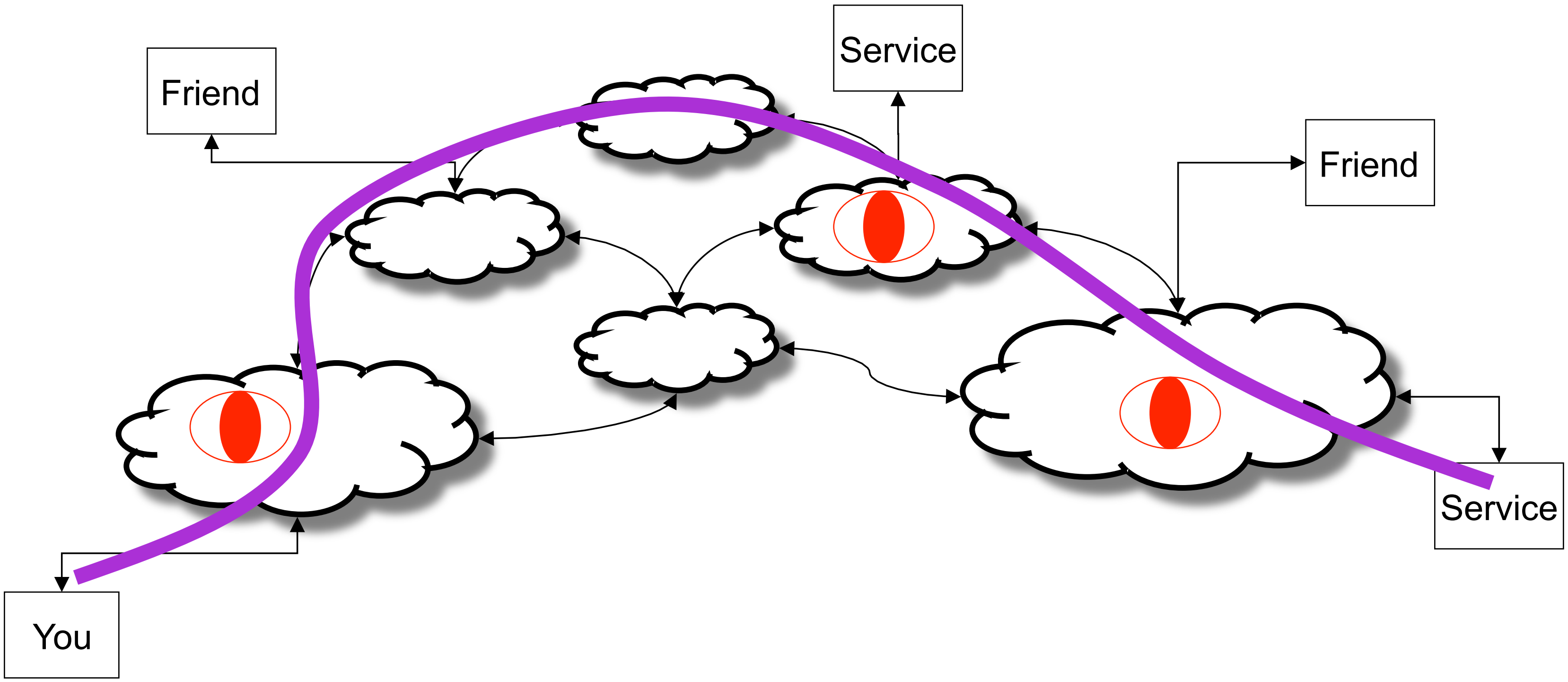

2010-12-29 15:15:00 - 2010-12-30 15:15:00 (UTC+1)
Rate (- received / + sent) [bit/s]
10 G
9 G
8 G
7 G
6 G
5 G
4 G
3 G
2 G
1 G
0 G
Wed 18:00
Wed 21:00
Thu 00:00
Thu 03:00

Sort direction: Total		Total			Sent	Received	Top
	ipVersion, Type	Total	Average	Percentage	Total	Total	☑ ○
🟥	IPv4, Unicast	37.43 TiB	3.82 Gbit/s	96.17 %	37.43 TiB	0 B	☑ ○
🟩	IPv6, Unicast	824.82 GiB	82.22 Mbit/s	2.07 %	824.82 GiB	0 B	☑ ○
🟦	IPv4, None	326.41 GiB	32.54 Mbit/s	0.82 %	326.41 GiB	0 B	☑ ○
🟪	IPv4, Multicast	61.44 GiB	6.13 Mbit/s	0.15 %	61.44 GiB	0 B	☑ ○
🟨	IPv6, Multicast	888.75 MiB	86.91 kbit/s	< 0.01 %	888.75 MiB	0 B	☑ ○
🟩	Reserved (0), None	719.78 MiB	70.08 kbit/s	< 0.01 %	719.78 MiB	0 B	☑ ○
🟦	IPv6, None	587.55 MiB	57.21 kbit/s	< 0.01 %	587.55 MiB	0 B	☑ ○
	IPv4, Broadcast	463.07 MiB	45.22 kbit/s	< 0.01 %	463.07 MiB	0 B	□ ○
	IPv6, Reserved	1.32 MiB	166.06 bit/s	< 0.01 %	1.32 MiB	0 B	□ ○
	Total	38.92 TiB	3.96 Gbit/s		38.92 TiB	0 B	

Protocols


2010-12-29 15:20:00 - 2010-12-30 15:20:00 (UTC+1)
Rate (- received / + sent) [bit/s]
10 G
9 G
8 G
7 G
6 G
5 G
4 G
3 G
2 G
1 G
0 G
Wed 18:00
Wed 21:00
Thu 00:00
Thu 03:00

Sort direction: Total	Total			Sent	Received	Top
protocolIdentifier	Total	Average	Percentage	Total	Total	☑ ○
🟥 TCP	31.65 TiB	3.23 Gbit/s	81.31 %	31.65 TiB	0 B	☑ ○
🟩 UDP	5.76 TiB	587.58 Mbit/s	14.79 %	5.76 TiB	0 B	☑ ○
🟦 NONE/HOPOPT	828.75 GiB	82.6 Mbit/s	2.08 %	828.75 GiB	0 B	☑ ○
🟪 AH	318.62 GiB	33.41 Mbit/s	0.8 %	318.62 GiB	0 B	☑ ○
🟨 GRE	3.18 GiB	318.34 kbit/s	< 0.01 %	3.18 GiB	0 B	☑ ○
🟩 ICMP	2.19 GiB	217.98 kbit/s	< 0.01 %	2.19 GiB	0 B	☑ ○
🟦 ESP	1.33 GiB	132.65 kbit/s	< 0.01 %	1.33 GiB	0 B	☑ ○
IPv6	794.22 MiB	80.37 kbit/s	< 0.01 %	794.22 MiB	0 B	□ ○
IGMP	6.5 MiB	666.52 bit/s	< 0.01 %	6.5 MiB	0 B	□ ○
OSPF-IGP	1.55 MiB	179.45 bit/s	< 0.01 %	1.55 MiB	0 B	□ ○
Total	38.92 TiB	3.96 Gbit/s		38.92 TiB	0 B	


Service
Friend
Friend
Service
You

Questions?

Jeroen Massar <jma@zurich.ibm.com>