



Análisis de Flujos de Red. *Netflow*

Gestión de Redes

Análisis de tráfico: proceso de medida y valoración del tráfico de red para la identificación de patrones de interés.

(e.g. Matriz de tráfico de la semana pasada)

- Ejemplo de análisis tradicional:
 - ▣ Uso de SNMP y de la MIB de RMON (o un analizador de paquetes)
 - Examinar los paquetes de los diferentes enlaces y sacar estadísticas
- Problema: el procesamiento se hace por paquete
 - ▣ Se examina un paquete y se actualizan estadísticas (contadores...)
 - ▣ No se pueden detectar relaciones entre paquetes
 - ¿Iba el paquete al mismo destino que el anterior?
 - ¿Qué paquetes forman parte de una misma conversación?
 - 20% del tráfico son emails. Pero ¿cuántos emails se han enviado?
- Sólo permite estadísticas agregadas

Solución: la abstracción de Flujo de Red

3

- Queremos poder valorar *comunicaciones* que vayan más allá del paquete individual
 - ▣ ¿Tamaño máximo de los emails enviados?
 - ▣ ¿Duración media de una llamada VoIP?
 - ▣ ¿Número máximo de conexiones TCP simultáneas en un enlace?
- Claramente, el análisis tradicional no permite responder a estas preguntas
- Solución: análisis basado en flujos de red
 - ▣ Flujo: secuencia de paquetes relacionados
 - ▣ Definición genérica y flexible. Permite al administrador de red emplear la granularidad que crea oportuna
 - Un flujo por cada par de programas comunicándose
 - Un flujo para todos los paquetes que van hacia un servidor web

Posibles usos del Análisis de Flujos

4

- ❑ **Caracterización de tráfico:** entender el tráfico de red, clasificándolo por protocolo, por aplicación, por destino...
- ❑ **Estimación de rendimiento,** asignando flujos específicos a la carga útil respecto del tráfico total
- ❑ **Contabilidad y facturación** de servicios, separando flujos por cliente y servicio.
- ❑ **Comparación de entrada/salida de tráfico** para dimensionamiento de enlaces, por ejemplo.
- ❑ **Evaluación de calidad de servicio (QoS),** confirmando si se cumple lo pactado con proveedores o clientes
- ❑ **Detección de anomalías** como eventos o variaciones inesperadas en los flujos de red

Posibles usos del Análisis de Flujos

5

- **Análisis de seguridad**, por ejemplo, para detectar ataques de denegación de servicio o propagación de malware (virus, *worms*)
- **Diagnóstico y solución de problemas en tiempo real**, permitiendo la detección de problemas en el momento en que ocurren
- **Análisis Forense**, empleando datos históricos de flujos de red para identificar los eventos que dieron lugar a un problema
- **Planificación de capacidad**, usando la información de flujos para construir matrices de tráfico más avanzadas
- **Dimensionamiento y planificación de aplicaciones**, empleando flujos para monitorizar aplicaciones concretas de cara a su gestión específica

Flujos unidireccionales y bidireccionales

6

□ Flujos unidireccionales

- ▣ Se corresponden con tráfico que fluye en una sola dirección
- ▣ Muy populares
 - La mayoría de enlaces son *full-duplex*
 - Es habitual modelarlos como dos enlaces independientes
 - Permite modelar encaminamiento asimétrico
 - Más flexible

□ Flujos bidireccionales

- ▣ Capturan la interacción en ambos sentidos de la comunicación
- ▣ Muy útiles en aplicaciones “simétricas”
 - Misma cantidad de datos en ambos sentidos
 - Mismo camino para ambos sentidos
 - Poco usual

□ En ocasiones la propia aplicación restringe el tipo

Niveles de agregación de flujos

7

- Concepto intuitivo de flujo
 - ▣ Comunicación única entre dos puntos finales
- En la práctica, la definición es más flexible
 - ▣ En general, se basa en campos de las cabeceras de los paquetes
 - e.g. paquetes con una determinada IP de destino y puerto TCP destino
 - ▣ A veces también restricciones temporales
 - e.g. que no estén separados más de 30 segundos entre sí
- Agregación de flujos
 - ▣ Agrupación de comunicaciones individuales en una misma medida
- Nivel de agregación: granularidad

Ejemplos de agregación de flujos

8

- Por dirección IP
 - ▣ origen (e.g. número de hosts contribuyendo a una congestión)
 - ▣ destino (e.g. número de hosts externos con los que interaccionamos)
 - ▣ Origen y destino (pares de máquinas comunicándose)
- Por interfaz de red
 - ▣ e.g. para analizar el tráfico de salida de un router
- Por protocolo de transporte (UDP, TCP, ICMP...)
 - ▣ Para analizar cada protocolo por separado

Ejemplos de agregación de flujos (II)

9

- Por tipo de aplicación
 - ▣ Agregando por número de puerto
 - ▣ E.g. ¿Qué porcentaje de tráfico es Web y cuál email?
- Por el contenido del paquete
 - ▣ *Deep packet inspection*
 - ▣ E.g. Flujos para secciones específicas de una web
- Por extremos finales a nivel de aplicación
 - ▣ Clasificando por tupla (IP origen, IP destino, puerto origen, puerto destino, protocolo)

Ejemplos de agregación de flujos (III)

10

- Flujo unidireccional por IP origen y destino



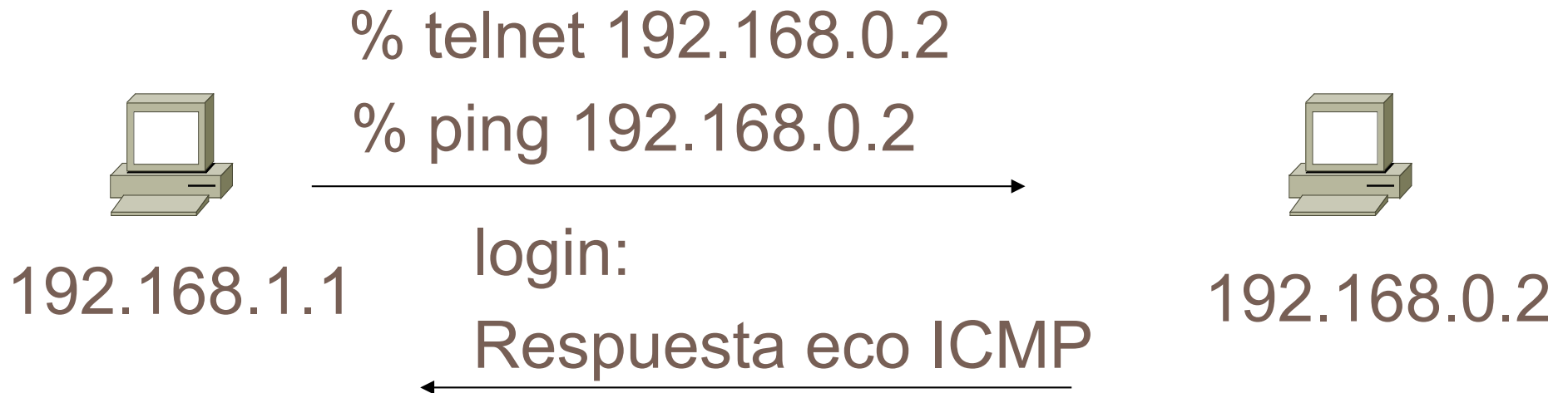
- Flujos activos

	Flujo	IP Origen	IP Destino
	1	192.168.1.1	192.168.0.2
	2	192.168.0.2	192.168.1.1

Ejemplos de agregación de flujos (IV)

11

- Flujo unidireccional por IP origen y destino



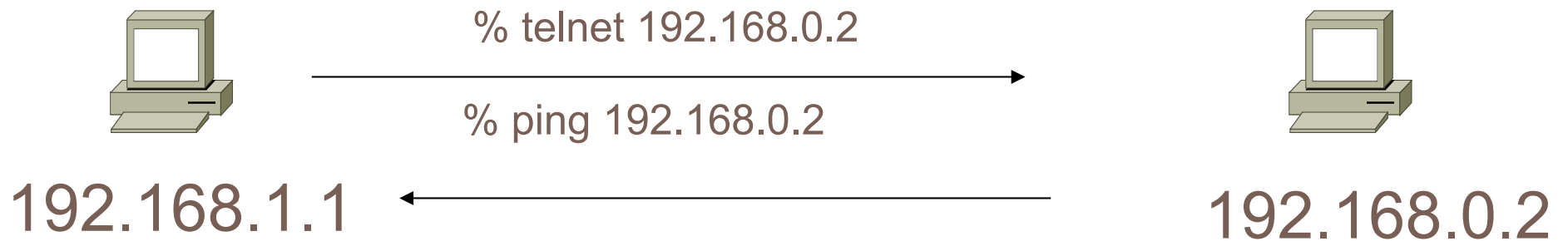
- Flujos activos

	Flujo	IP Origen	IP Destino
	1	192.168.1.1	192.168.0.2
	2	192.168.0.2	192.168.1.1

Ejemplos de agregación de flujos (V)

12

□ Flujo unidireccional por IPs, puertos y protocolo



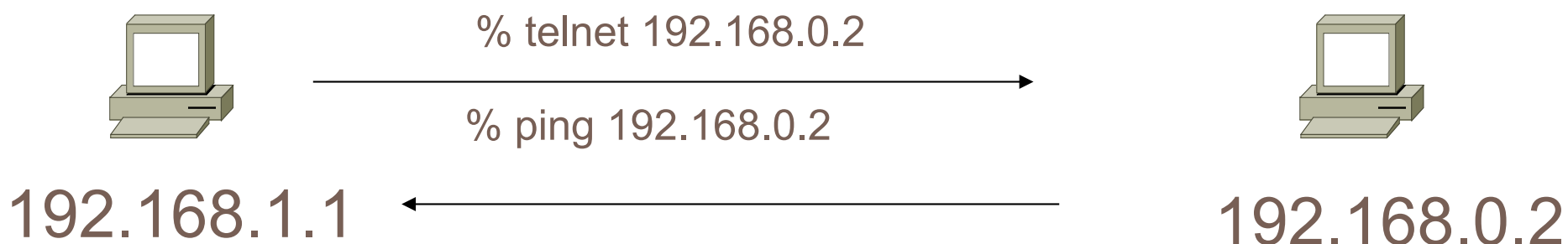
□ Flujos activos

Flujo	IP Origen	IP Destino	prot	srcPort	dstPort
1	192.168.1.1	192.168.0.2	TCP	32000	23
2	192.168.0.2	192.168.1.1	TCP	23	32000
3	192.168.1.1	192.168.0.2	ICMP	0	0
4	192.168.0.2	192.168.1.1	ICMP	0	0

Ejemplos de agregación de flujos (VI)

13

- Flujo **bidireccional** por IPs, puertos y protocolo



- Flujos activos

Flujo	IP	IP	Prot	Port	Port
1	192.168.1.1	192.168.0.2	TCP	32000	23
2	192.168.1.1	192.168.0.2	ICMP	0	0

□ Análisis *online*

- Los flujos se analizan en tiempo real
 - Se envían los datos al analizador en cuanto se recogen
 - Retardos entre el evento y la visualización
 - Por la velocidad del análisis
 - Por el nivel de agregación
 - Por el retardo en el transporte de los datos desde recogida a análisis

□ Análisis *offline*

- Se analizan los flujos después de que los datos se han recogido y almacenado
 - e.g. cada noche se analizan los datos de las últimas 24 horas
- Útil para análisis forense o cuando no se sabe a priori qué analizar
- Ventaja principal: no hay restricciones de tiempo de cómputo

Captura de Datos Activa y Pasiva

15

□ Captura de datos pasiva

- ▣ El hardware que captura información está separado del que procesa o reencamina los paquetes
 - sonda conectada a un hub en una subred
 - Mirror port en un switch
 - optical splitter en redes de fibra
- ▣ No interfiere en el tráfico de la red
- ▣ Normalmente se restringe a un solo enlace o VLAN

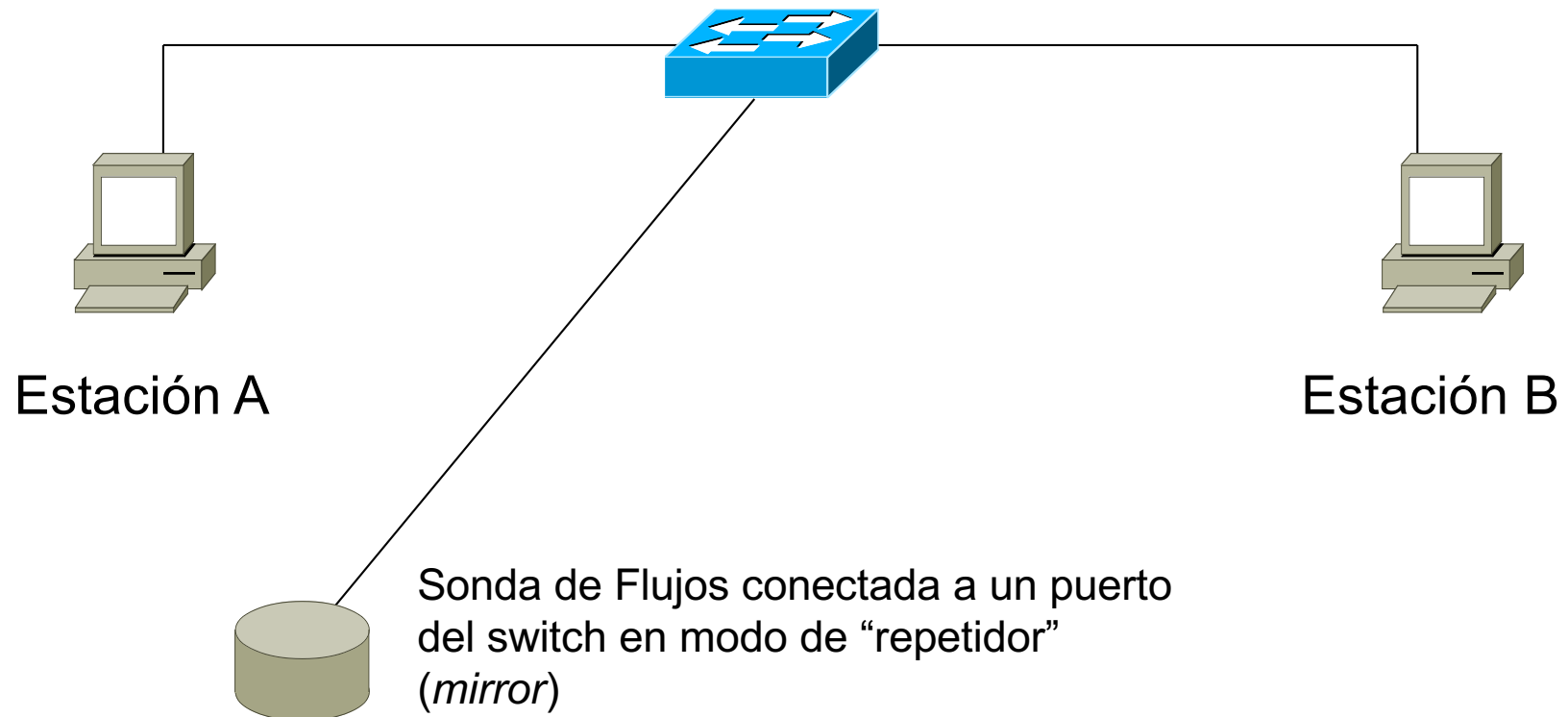
□ Captura de datos activa

- ▣ Un elemento de la red incluye hardware/software para la captura
 - e.g. Añadir una tarjeta de análisis de flujos en un router
- ▣ Permite analizar todo el tráfico que pasa por el elemento de red
- ▣ Permite emplear la información de flujo para la operación
- ▣ Hace los routers más complejos

Captura de Datos Activa y Pasiva

16

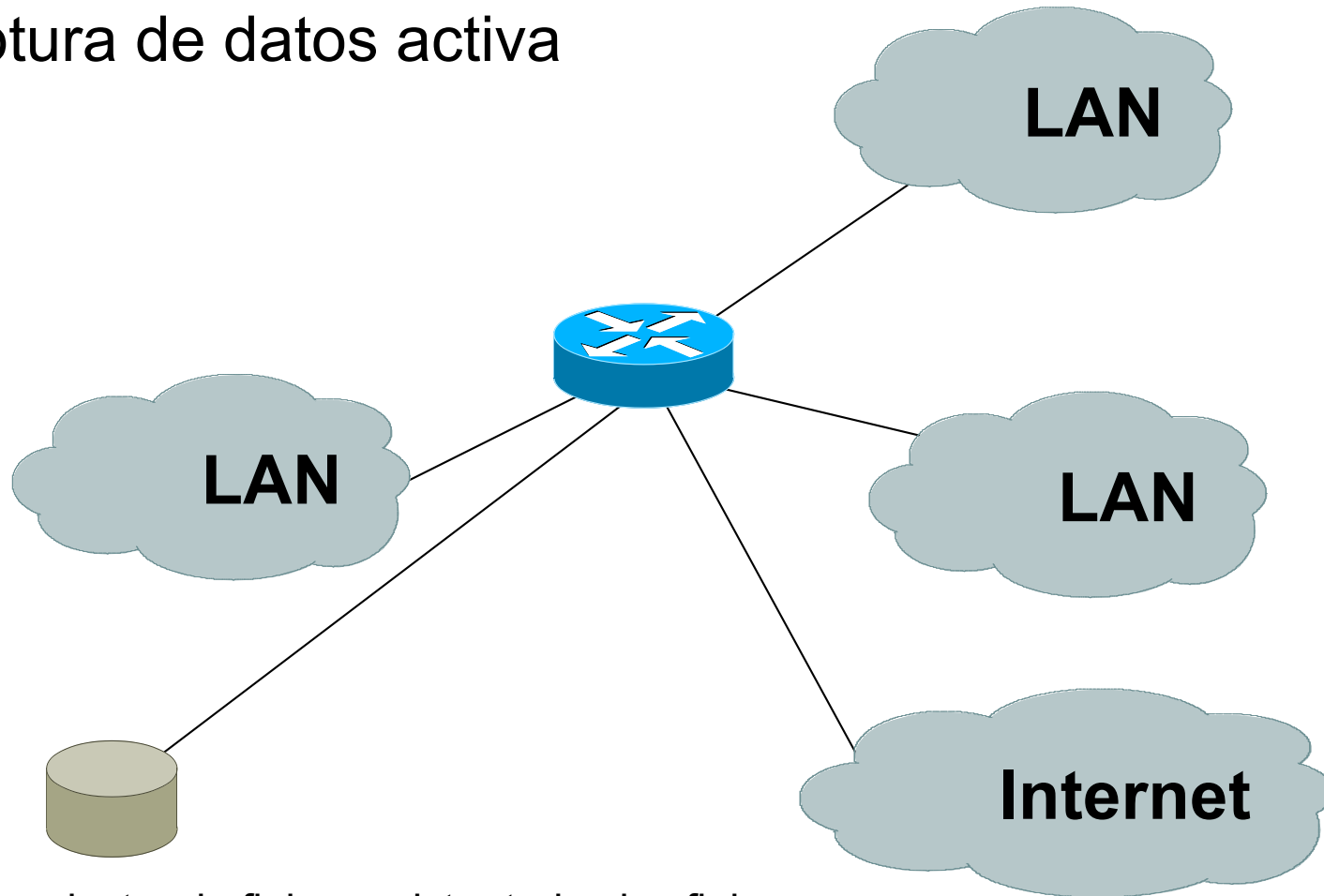
□ Captura de datos pasiva



Captura de Datos Activa y Pasiva

17

□ Captura de datos activa



Módulo colector de flujos registra todos los flujos procesados por el router.

- Imaginemos una paquete que lleva una petición HTTP:



- Cada cabecera tiene un campo que indica de qué tipo es la PDU encapsulada
 - ▣ (0x0800, 6, 80) → IP, TCP, HTTP (probablemente)
 - ▣ Consultando estos tres campos “sabemos” que es un flujo web.
- Clasificación: proceso de chequeo selectivo de campos para ver a qué flujo pertenece un paquete
 - ▣ Mucho más rápido que inspeccionar el paquete completo
 - ▣ Puede optimizarse por hardware
- El “esfuerzo” es proporcional al número de cabeceras

Inspección y clasificación de paquetes (II)

19

- ¿Y si queremos mucho más control?:

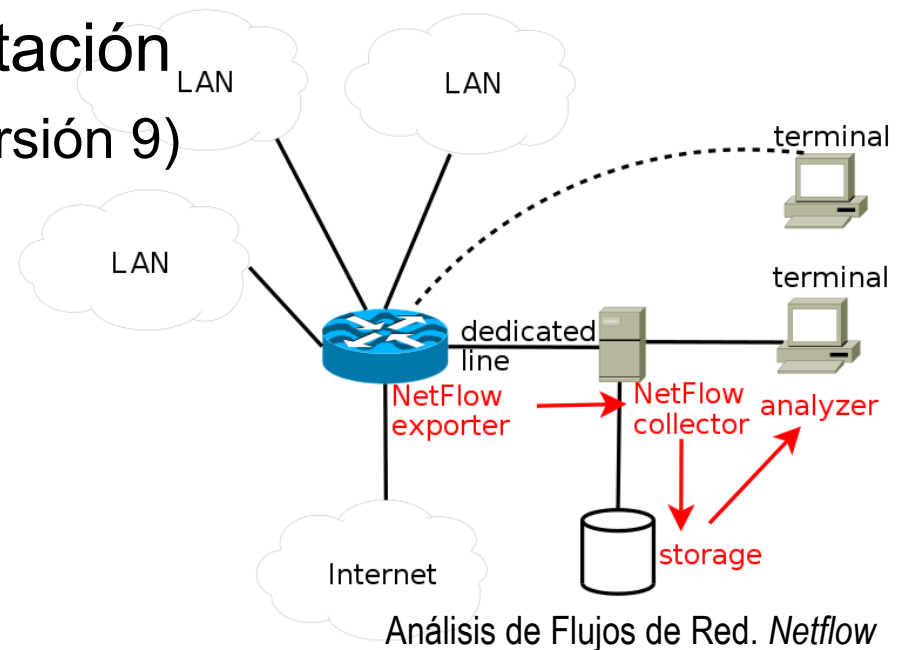


- Podemos utilizar la URL (o parte de ella) para definir flujos para diferentes secciones del sitio web
 - ▣ Desde el punto de vista del administrador, prácticamente igual
 - ▣ Desde el punto de vista del clasificador, muy diferente
 - La URL no se encuentra en una posición fija
 - Requiere procesar el paquete a nivel de aplicación
- *Deep packet inspection*
 - ▣ Requiere muchos más recursos computacionales
 - ▣ El clasificador será más lento (tiempo real → menos paquetes/s)

- Dos funciones diferenciadas en un router “avanzado”
 - ▣ Reenvío de paquetes y Captura de información de flujos
 - ▣ Ambas suponen una clasificación de paquetes
 - Reenvío: cabeceras → siguiente salto
 - Flujos: cabeceras → flujo
- Pueden combinarse para optimizar el reenvío
 - ▣ Creamos una memoria caché para los flujos activos
 - Asociando ID de flujo con siguiente salto
 - Menos flujos que posibles destinos
 - Memoria más pequeña → acceso más rápido
 - ▣ Los paquetes de flujos conocidos se reenvían mirando la caché
 - Sólo se mira la tabla de reenvío para el primer paquete de cada flujo
 - ▣ Útil si es probable que se repitan paquetes de un flujo
 - Localidad temporal de referencia → normalmente elevada

- Comunicación de los datos de flujo desde el punto donde se originan hasta el punto de análisis
 - ▣ Típicamente, router → máquina de gestión
- Formato de datos de exportación
 - ▣ Compromiso entre cantidad de información y eficiencia
 - Más información implica mejores posibilidades de análisis
 - Pero consume ancho de banda y almacenamiento
 - ▣ Metadatos
 - Timestamp de cada registro
 - ID del equipo que capturó los datos
- Protocolo
 - ▣ De nuevo, compromiso entre eficiencia y efectividad
 - e.g. ¿nivel de enlace o de transporte?

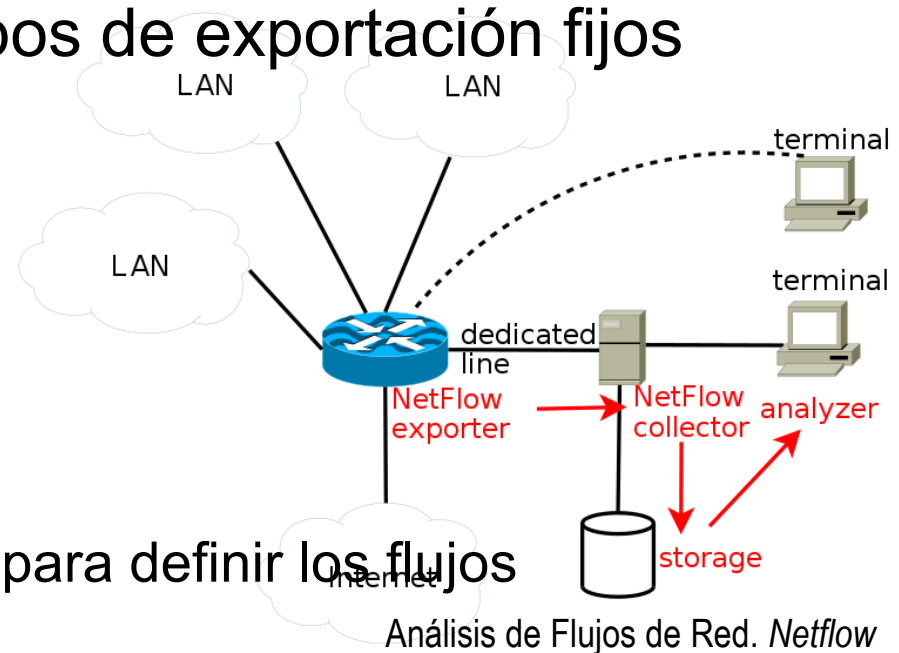
- La tecnología de flujos más usada actualmente
 - ▣ Desarrollada por CISCO para mejorar el rendimiento de los *routers* (con cachés de flujos)
 - ▣ Definieron un formato de exportación de datos de flujo para monitorizar el rendimiento
 - ▣ Fueron los usuarios los que se dieron cuenta del potencial de esos datos y desarrollaron herramientas de análisis
- Formato de datos de exportación
 - ▣ Recogido en el RFC 3954 (versión 9)
- Arquitectura distribuida
 - ▣ Exportadores
 - ▣ Colectores
 - ▣ Analizadores



NetFlow. Características básicas

23

- Paradigma de captura activa
 - ▣ El router registra información sobre los flujos y la exporta a un colector externo
- Flujos de datos unidireccionales
- Permite al administrador controlar la captura de datos
 - ▣ E.g. Frecuencia de muestreo de paquetes
- En versiones previas, campos de exportación fijos
 - ▣ IP origen/destino
 - ▣ Puerto origen/destino
 - ▣ Interfaz de entrada /salida
 - ▣ ...
- Versión 9: más flexible
 - ▣ Permite elegir qué datos usar para definir los flujos



Extensibilidad y plantillas

24

- NetFlow envía información descriptiva con los datos
 - ▣ Campos de datos enviados
 - ▣ Longitud de cada campo
 - ▣ ...
- Con esa información (plantilla), el colector ya sabe cómo procesar los datos
 - ▣ Sólo es necesario configurar los exportadores
- Las plantillas tienen un ID asignado, por lo que sólo se envían la primera vez
- El formato de paquete permite incluso añadir nuevos campos sin necesidad de crear una versión nueva

- Campos que se incluyen en cada plantilla
 - ▣ Pueden definirse plantillas diferentes para cada flujo
- Tasa de muestreo de paquetes
 - ▣ Básicamente, permite muestrear un paquete de cada N
 - No necesariamente muestreo determinista
 - ▣ Permite balancear la precisión y el volumen de datos
- Tamaño de caché
 - ▣ Se exportan flujos completos → necesario almacenamiento
 - ▣ El tamaño de caché limita el número de flujos simultáneos
 - ¡¡¡Si la caché se llena, se exportan los flujos incompletos!!!
- ¿Cómo sabe el *exporter* cuando el flujo está completo?
 - ▣ TCP: muy simple → cuando se libera la conexión
 - ▣ ¿IP, UDP? → temporizador de flujo

Transporte de datos en Netflow

26

- Se emplea **UDP** para enviar datos desde los *exporters* a los *collectors*
 - ▣ Se pueden perder o reordenar paquetes
 - ▣ No hay control de flujo ni de congestión
- A partir de la versión 2, Netflow emplea **números de secuencia** en sus mensajes
 - ▣ Si los mensajes llegan antes que su plantilla, se almacenan hasta que esta llegue
 - ▣ Periódicamente se reenvía la plantilla (por si acaso)
- El tráfico de Netflow puede congestionar la red
 - ▣ ¿Cómo evitarlo? Netflow no lo hace
 - ▣ Soluciones a nivel de administrador
 - Enviar el tráfico Netflow por una red separada
 - Limitar la cantidad de tráfico que genera Netflow

Consideraciones finales

27

- Herramientas de análisis de datos para Netflow
 - ▣ Existen múltiples, y algunas de código abierto
 - ▣ <http://www.splintered.net/sw/flow-tools/>
- Información adicional por parte de CISCO
 - ▣ http://www.cisco.com/en/US/products/ps6601/products_ios_protocol_group_home.html
- IETF Internet Protocol Flow Information eXport (IPFIX)
 - ▣ Protocolo basado en Netflow versión 9
 - ▣ Incluye seguridad y soporte de control de congestiones
 - ▣ <http://tools.ietf.org/wg/ipfix/>

- Douglas E. Comer “Automated Network Managemet Systems”, Capítulo 11
- Recursos adicionales:
 - <http://sourceforge.net/projects/netfltools/>
 - <http://ws.edu.isoc.org/>
- Más recursos adicionales
 - <http://www.splintered.net/sw/flow-tools/docs/flow-tools-examples.html>
 - <http://nfdump.sourceforge.net/>
 - <http://flow-tools.sourceforge.com/>
 - http://www.cisco.com/en/US/docs/ios/12_2/switch/configuration/guide/xcfnfc.html

- Vale, ¿pero qué pinta tiene esto en la práctica?
- Cabecera netflow (versión 5)

Byte 1	Byte 2	Byte 3	Byte 4
Version		Count	
SysUpTime			
UNIX Seconds (seconds since Epoch)			
UNIX Nanoseconds (residual nanoseconds)			
Flow Sequence Number			
Engine Type	Engine ID	Reserved	

□ Registro netflow (versión 5)

Byte 1	Byte 2	Byte 3	Byte 4
Source IP Address			
Destination IP Address			
Next Hop IP Address			
Input ifIndex		Output ifIndex	
Packets			
Bytes			
Start time of flow			
End time of flow			
Source port		Destination port	
Padding	TCP Flags	IP Protocol	TOS
Source AS		Destination AS	
Source Mask Length	Destination Mask Length	Padding	

□ No necesariamente conveniente (e.g. Tiempo de inicio)

□ Configurar dispositivo “exportador” (e.g. router Cisco)

```
configure terminal
interface serial 3/0/0
ip route-cache flow exit
ip flow-export 1.1.15.1 12345 version 5 peer-as
exit
clear ip flow stats
```

□ Caché de Netflow en el router

```
Router# show ip cache flow
```

```
IP packet size distribution (230151 total packets):
```

```
1-32 64 96 128 160 192 224 256 288 320 352 384 416 448 480
```

```
.999 .000 .000 .000 .000 .000 .000 .000 .000 .000 .000 .000 .000 .000 .000
```

```
512 544 576 1024 1536 2048 2560 3072 3584 4096 4608
```

```
.000 .000 .000 .000 .000 .000 .000 .000 .000 .000 .000
```

□ Caché de Netflow en el router (cont.)

```
IP Flow Switching Cache, 4456448 bytes
65509 active, 27 inactive, 820628747 added
955454490 aged polls, 0 flow alloc failures
Exporting flows to 1.1.15.1 (2057)
820563238 flows exported in 34485239 udp datagrams, 0 failed
last clearing of statistics 00:00:03
```

Protocol	Total	Flows	Packets	Bytes	Packets	Active(Sec)	Idle(Sec)
-----	Flows	/Sec	/Flow	/Pkt	/Sec	/Flow	/Flow
TCP-BGP	71	0.0	1	49	0.0	2.5	15.8
UDP-other	17	0.0	1	328	0.0	0.0	15.7
ICMP	18966	6.7	10	28	72.9	0.1	22.9
Total:	19054	6.7	10	28	72.9	0.1	22.9

SrcIf	SrcIPAddress	DstIf	DstIPAddress	Pr	TOS	Flgs	Pkts
Port Msk AS		Port Msk AS	NextHop			B/Pk	Active
Et1/1	52.52.52.1	Fd4/0	42.42.42.1	01	55	10	3748
0000 /8 50		0000 /8 40	202.120.130.2			28	17.8
Et1/2	52.52.52.1	Fd4/0	42.42.42.1	01	CC	10	3568
0000 /8 50		0000 /8 40	202.120.130.2			28	17.8
Et1/2	10.1.3.2	Fd4/0	42.42.42.1	01	C0	10	1124
0000 /0 0		0000 /8 40	202.120.130.2			28	17.8
Et1/2	11.1.3.2	Fd4/0	42.42.42.1	01	C0	10	1157
0000 /0 0		0000 /8 40	202.120.130.2			28	17.7
Et1/2	14.1.3.2	Fd4/0	42.42.42.1	01	C0	10	1149
0000 /0 0		0000 /8 40	202.120.130.2			28	17.8
Et1/2	15.1.3.2	Fd4/0	42.42.42.1	01	C0	10	1127
0000 /0 0		0000 /8 40	202.120.130.2			28	17.7

- Configurar un “colector” (Linux con flow-tools)

```
#flow-capture -p /var/run/flow-capture.pid  
-n 287 -w /var/netflow/sensorXY \  
-S 5 1.1.15.1/1.1.15.2/12345
```

- Examinar datos en un “analizador” (Linux con flow-tools)

- ▣ Definir filtros (no imprescindible, pero conveniente)

```
filter-primitive tcp_only  
type ip-protocol  
permit 6  
filter-definition tcp_only  
match ip-protocol tcp_only
```

□ Examinar datos en un “analizador” (Linux con flow-tools)

```
$ flow-cat ./data | flow-nfilter -f filters -F tcp_only | flow-stat -f9 -S1
```

```
# --- ---- Report Information --- --- ---
```

```
# Fields: Total
```

```
# Symbols: Disabled
```

```
# Sorting: Descending Field 1
```

```
# Name: Source IP
```

```
#
```

```
# Args: flow-stat -f9 -S1
```

```
#
```

```
#
```

```
# IPaddr flows octets packets
```

```
#
```

```
130.49.72.0 30415 6881695 36110
```

```
128.223.216.0 28716 540202269 427566
```

```
130.49.88.0 24267 10770631 32186
```

```
...
```

□ Más posibilidades (flow-report)

```
root@server:~# flow-cat /var/flows/2008/2008-11/2008-11-13/ft-  
v05.2008-11-13.113001-0430 | flow-report -Sst3 | flow-rptfmt -  
fascii
```

ip-destination-port	flows	octets	packets
http	1966	1380162	15390
telnet	11	334734	8016
imaps	12	277522	3284
https	169	219010	1092
48492	2	141779	1357
60280	2	141739	1356
ssh	53	114556	1281
34498	3	75666	1244
43607	3	74334	1243
40282	2	72953	1236
1863	65	57043	607
cfengine	4	17877	182
55641	3	3458	15
55636	2	2598	10

□ Más posibilidades (nfdump)

```
# nfdump -r nfcapd.200508300700 -o extended -s srcip -s  
ip/flows -s dstport/pps/packets/bytes -s record/bytes
```

Top 10 Src IP Addr ordered by flows:

Date first seen	Duration	Src IP Addr	Flows	Packets	Bytes	pps	bps	bpp
2005-08-30 06:45:50.990	1147.332	125.67.123.234	109183	202523	13.1 M	176	96116	68
2005-08-30 06:45:02.928	1192.834	94.180.151.203	62794	219715	25.9 M	184	182294	123
2005-08-30 06:59:24.563	330.110	9.209.28.173	27864	47943	2.2 M	145	55769	48
2005-08-30 06:45:07.728	1190.594	125.248.33.146	17271	41942	5.7 M	35	40438	143
2005-08-30 06:59:16.431	341.892	138.5.122.251	12253	75925	39.2 M	222	962768	541
2005-08-30 06:59:48.111	310.211	130.195.23.210	11742	46928	3.2 M	151	86940	71
2005-08-30 06:59:54.066	304.257	255.93.216.43	11383	56943	4.5 M	187	123968	82
2005-08-30 06:59:53.362	304.894	219.182.16.57	11209	44784	2.0 M	146	54640	46
2005-08-30 06:47:06.503	1068.361	3.15.99.52	9000	16962	3.1 M	15	24415	192
2005-08-30 06:59:52.784	172.102	11.121.123.165	7176	7176	330096	41	15344	46

Top 10 IP Addr ordered by flows:

Date first seen	Duration	IP Addr	Flows	Packets	Bytes	pps	bps	bpp
2005-08-30 06:45:50.990	1147.332	125.67.123.234	234366	458197	30.2 M	399	221164	69
2005-08-30 06:45:02.928	1192.835	94.180.151.203	115841	428885	42.6 M	359	299577	104
2005-08-30 06:45:07.728	1190.594	125.248.33.146	28218	73178	7.8 M	61	55234	112
2005-08-30 06:59:24.563	330.110	9.209.28.173	27916	48086	2.2 M	145	55931	47
2005-08-30 06:59:48.111	310.212	130.195.23.210	23467	105779	7.9 M	340	212311	77
2005-08-30 06:59:53.362	304.960	219.182.16.57	22938	89563	4.0 M	293	109251	46
2005-08-30 06:59:54.066	304.257	255.93.216.43	22769	102496	7.6 M	336	210205	77
2005-08-30 06:59:16.431	341.892	138.5.122.251	21840	132119	45.3 M	386	1.1 M	359
2005-08-30 06:46:33.104	1102.656	129.251.42.241	20305	74505	5.8 M	67	43757	80
2005-08-30 06:47:07.272	1068.360	131.250.225.247	14452	36714	3.1 M	34	24078	87

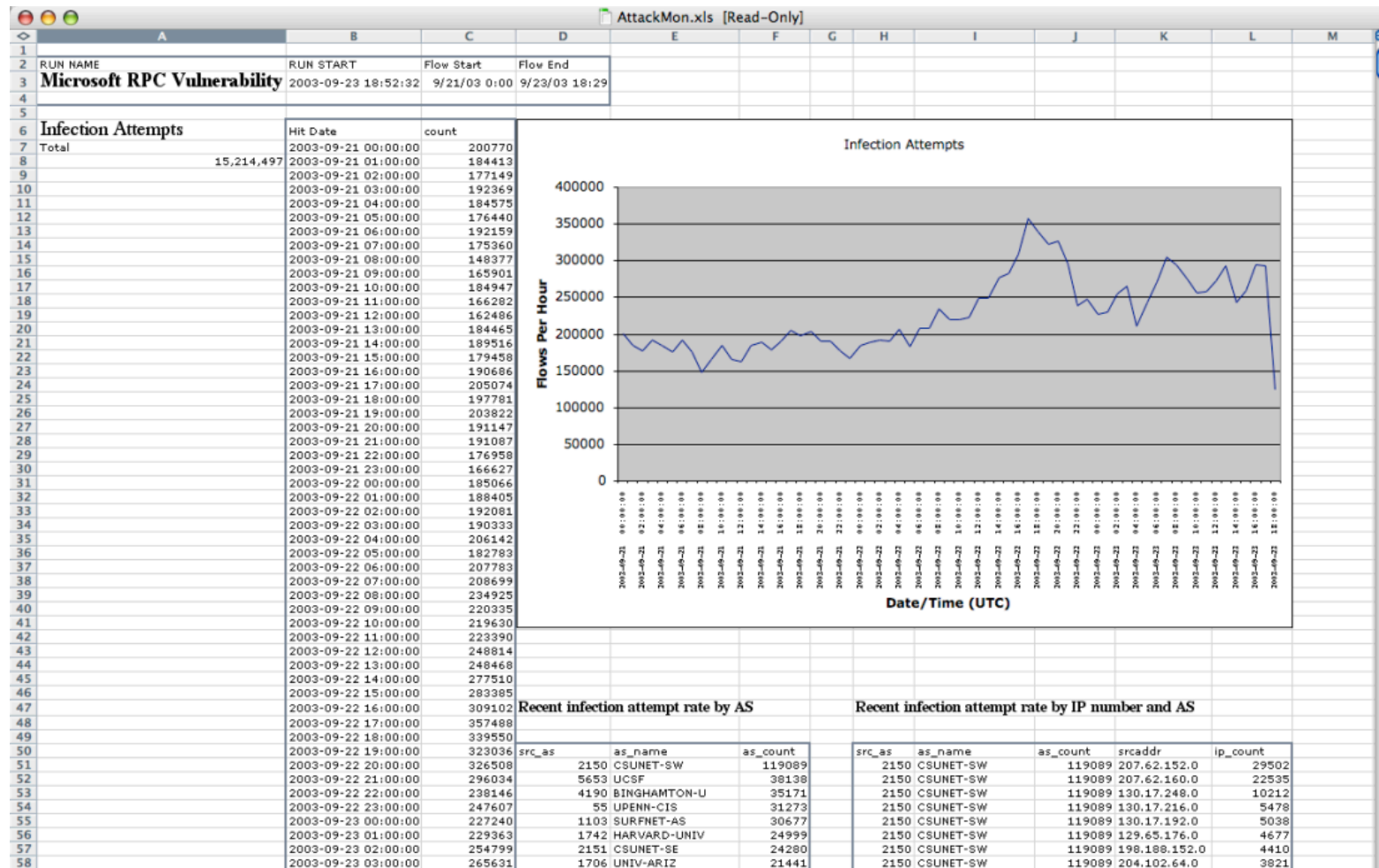
- Más posibilidades
 - ▣ Exportación de los flujos a tablas en un servidor PostgreSQL
 - PervasiveTechnologies at Indiana University
 - ▣ Permite exportarlo luego a Excel para sacar gráficas, estadísticas, etc.

```
netflow=# \d v5_rawnetflow_current
Table "public.v5_rawnetflow_current"
  Column      |  Type  | Modifiers
-----+-----+-----
 unix_secs    | integer |
 unix_nsecs   | integer |
 sysuptime    | bigint  |
 exaddr       | inet    |
 dpkts        | integer |
 doctets       | integer |
 first        | bigint  |
 last         | bigint  |
 engine_type  | integer |
 engine_id    | integer |
 srcaddr      | inet    |
 dstaddr      | inet    |
 nexthop      | inet    |
 input        | integer |
 output       | integer |
 srcport      | integer |
 dstport      | integer |
 prot         | integer |
 tos          | integer |
 tcp_flags    | integer |
 src_mask     | integer |
 dst_mask     | integer |
 src_as       | integer |
 dst_as       | integer |
```

Anexos – Uso de Netflow

38

□ Más posibilidades



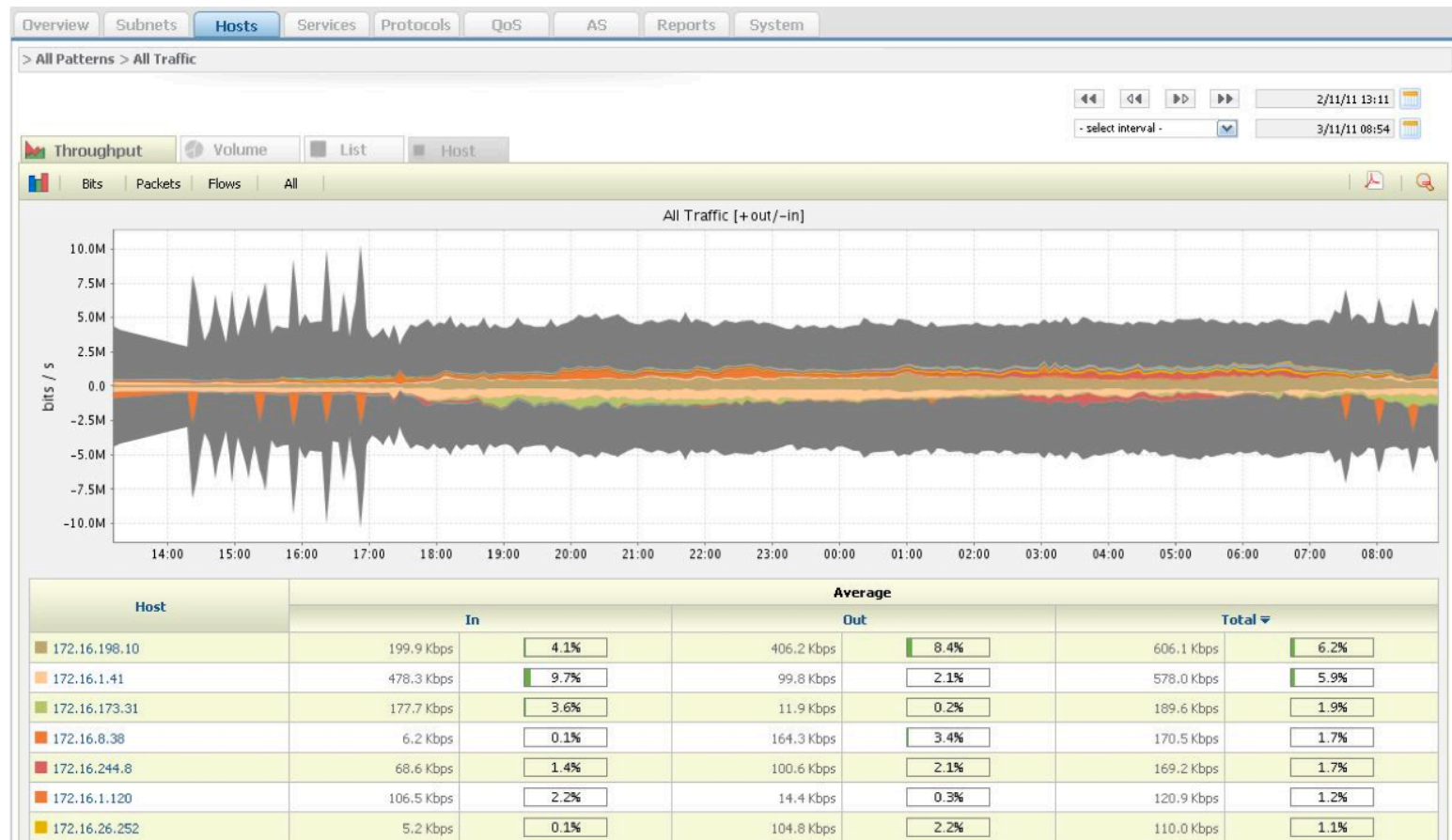
□ Aún más posibilidades (herramientas comerciales)

■ ICmyNet.Flow

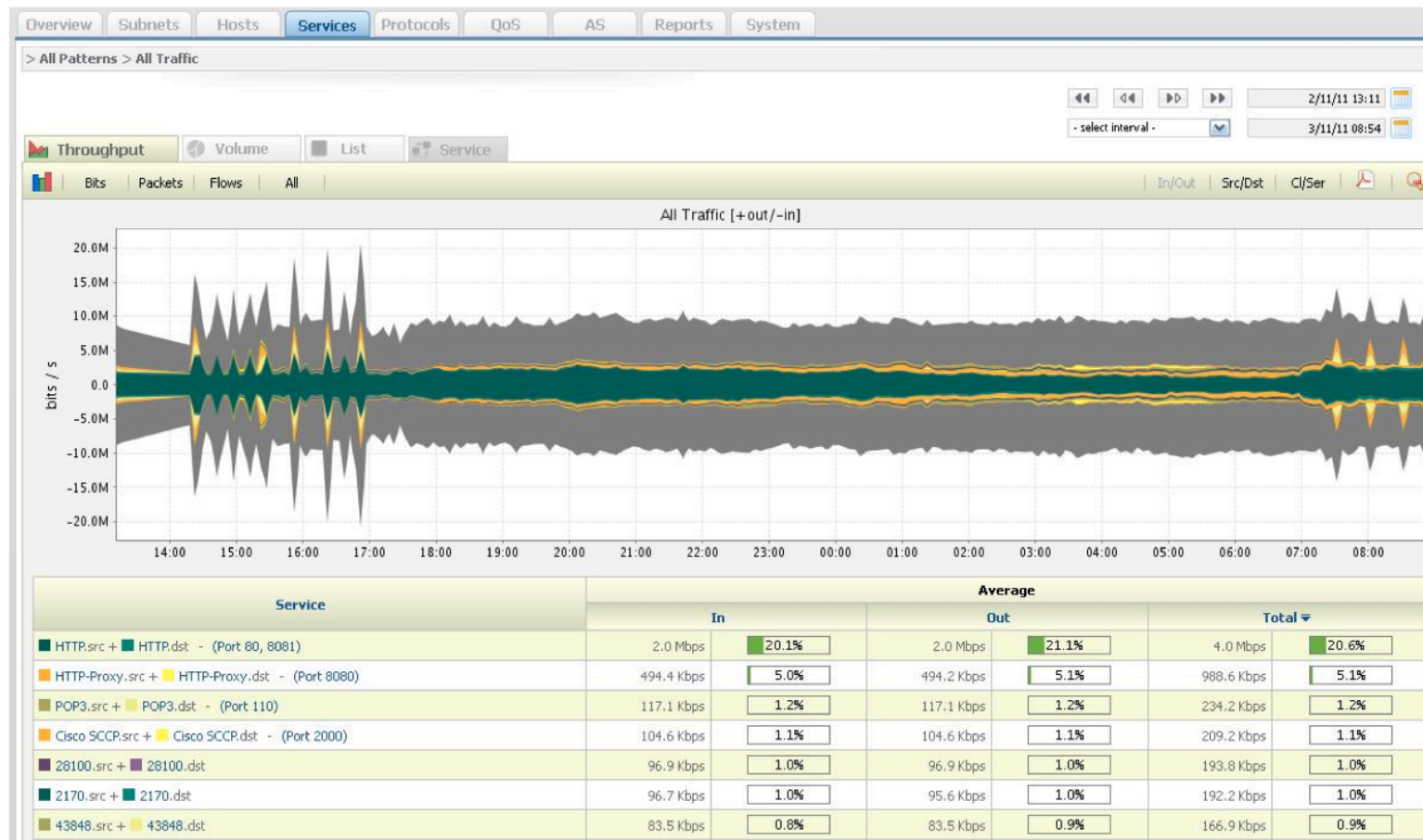
- <http://www.icmynet.com/products/icmynet.flow/overview.html>
- Libre para uso académico (y *tryout* de 30 días)



- Aún más posibilidades (herramientas comerciales)
 - ICmyNet.Flow

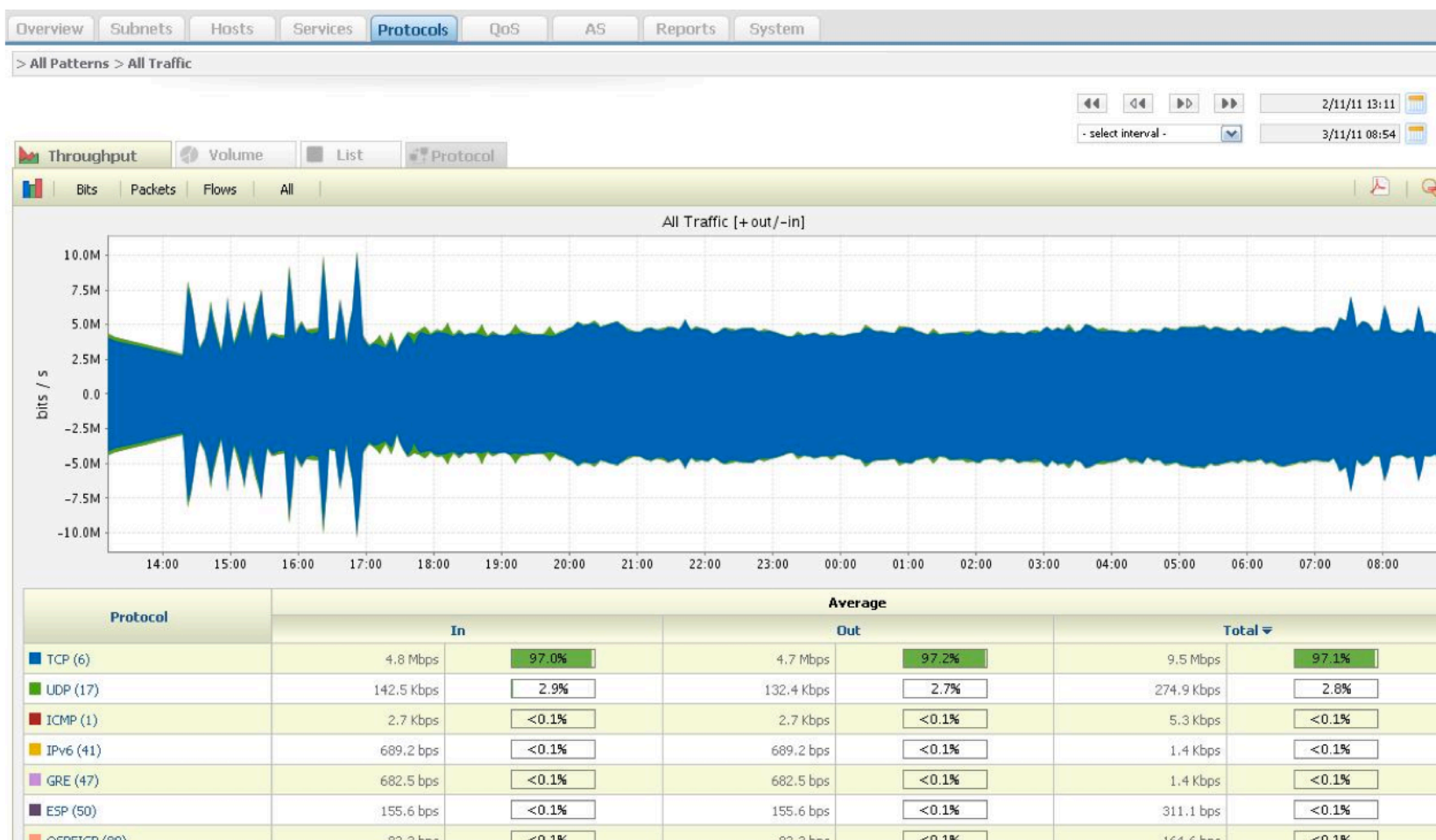


- Aún más posibilidades (herramientas comerciales)
 - ICmyNet.Flow



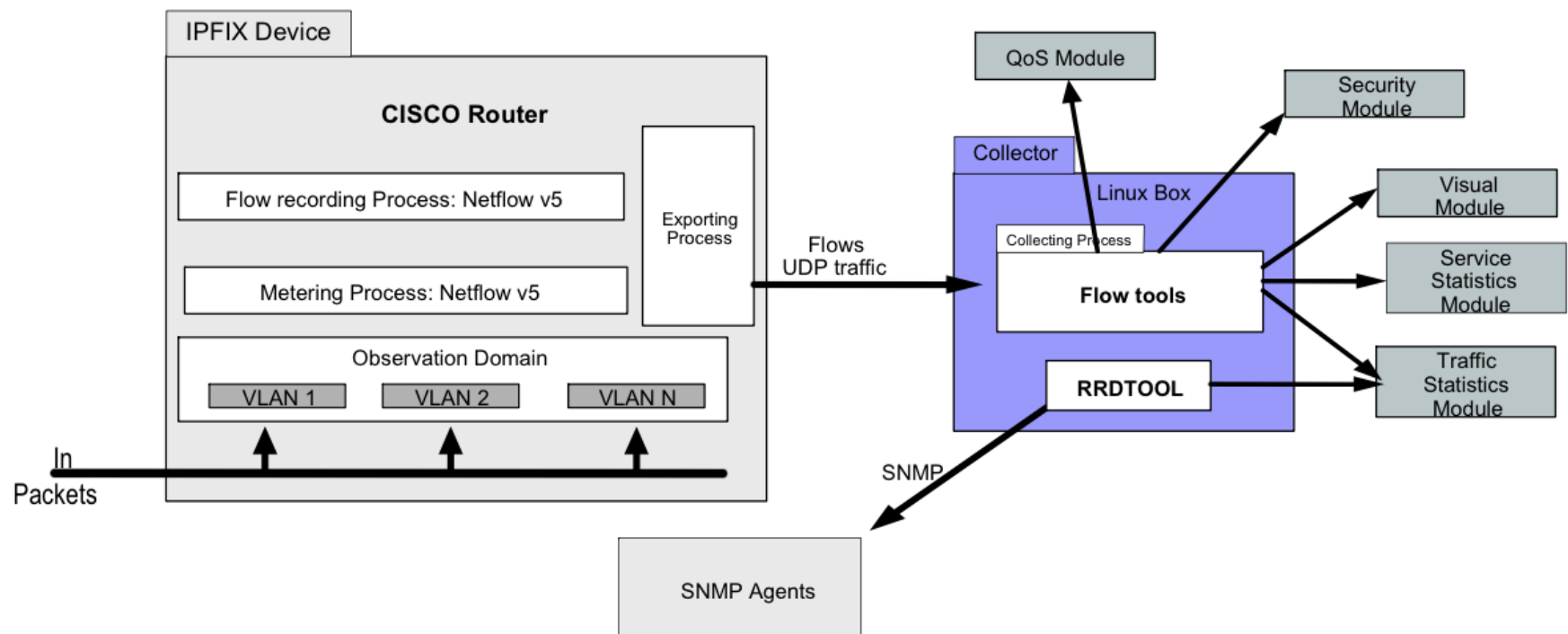
□ Aún más posibilidades (herramientas comerciales)

■ ICmyNet.Flow



Ejemplo práctico

□ Monitorización de una red académica mediante Netflow



Más herramientas

44

- Netflow analyzer

https://www.manageengine.com/products/netflow/?utm_source=capterra&utm_medium=ppc&utm_campaign=NetFlow

- Vigilar los IP SLA

<https://www.manageengine.com/products/netflow/ipsla-monitor.html>

- NFsens

<https://sourceforge.net/projects/nfsen/>