Redes de Datos - Laboratorio

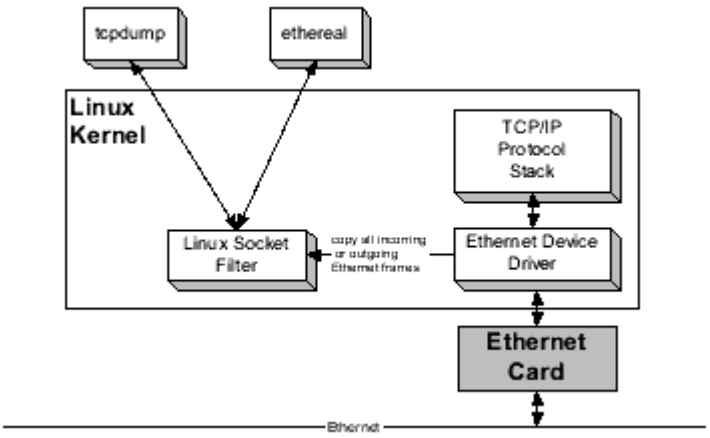
WireShark

Objetivo

Este instructivo describe el uso del programa WireShark (antes llamado Ethereal) para examinar paquetes en una red de datos.

Analizadores de Protocolos de Red

Para observar y analizar el comportamiento de los protocolos de red es preciso disponer de una herramienta capaz de monitorear el tráfico en la red y mostrarlo en una forma legible. Las herramientas que capturan y muestran el tráfico existente en una interfaz de red se denominan *analizadores de protocolos de red*, *analizadores de paquetes*, *"packet sniffers"* o simplemente *"sniffers"* (del inglés sniff, olfatear). Para visualizar el tráfico los analizadores de protocolo colocan la tarjeta de red en *modo promiscuo*, una modalidad en la cual es capturado todo el tráfico visible para la tarjeta de red. En una red Ethernet una interfaz de red en modo promiscuo puede ver todo el tráfico generado por todos los equipos que comparten el mismo conjunto de cables y concentradores (hubs). El modo promiscuo implica riesgos evidentes de seguridad, por lo que su uso suele limitarse al supervisor.



Arquitectura de software para un analizador de protocolo en Linux

La arquitectura de software para un protocolo de red en una máquina Linux con tarjeta Ethernet aparece en la figura. El analizador de protocolo corre como una aplicación, comunicándose con un componente del kernel Linux llamado Linux Socket Filter. El *kernel* de un sistema operativo es la parte central o núcleo del sistema, el *socket* (significa enchufe) es una forma de comunicación entre procesos propia de los sistemas Unix, un *proceso* es un programa en ejecución. La figura muestra dos analizadores de protocolo distintos, tcpdump y WireShark; tcpdump funciona en línea de comando, WireShark es una aplicación gráfica, pero ambos hacen más o menos lo mismo. El Linux Socket Filter actúa como intermediario entre el analizador de protocolo y el controlador de la tarjeta de

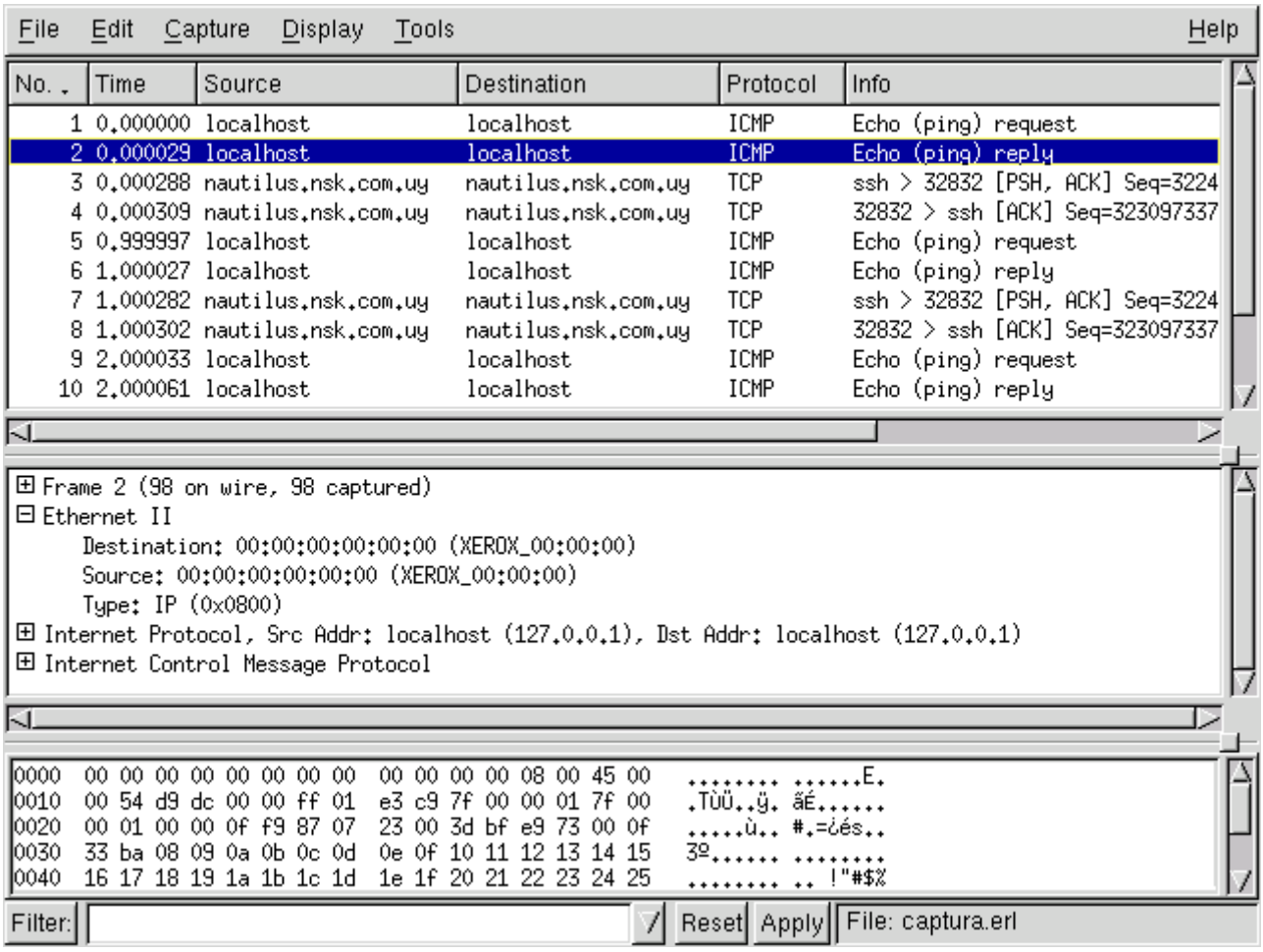
red Ethernet (Ethernet device driver); coloca la tarjeta de red en modo promiscuo y obtiene una copia de todo el tráfico entrante desde la red y todo el tráfico saliente hacia la red. El socket de filtro procesa este tráfico y lo transfiere al analizador de protocolo, que lo presenta al usuario.

WireShark

WireShark es un analizador de protocolos con interfaz gráfica capaz de reconocer muchos protocolos distintos. Permite tanto revisar los paquetes de datos en una red activa como desde un archivo de captura previamente generado; es capaz de comprender diversos formatos de archivo propios de otros programas de captura, en particular el clásico `tcpdump`.

Uso de WireShark

La invocación del programa WireShark puede hacerse a través del menú de invocación del ambiente gráfico o desde una terminal Unix si no existe la opción en el menú. Si se hace a través de una terminal Unix, el comando `wireshark &` o `ethereal &` (en el laboratorio) arranca el programa y devuelve el control de la terminal al usuario para poder continuar ingresando comandos. El símbolo `&` arranca el programa como proceso independiente de la terminal. La figura muestra la ventana principal de WireShark luego de una captura de datos. Inicialmente, esta ventana aparece vacía.

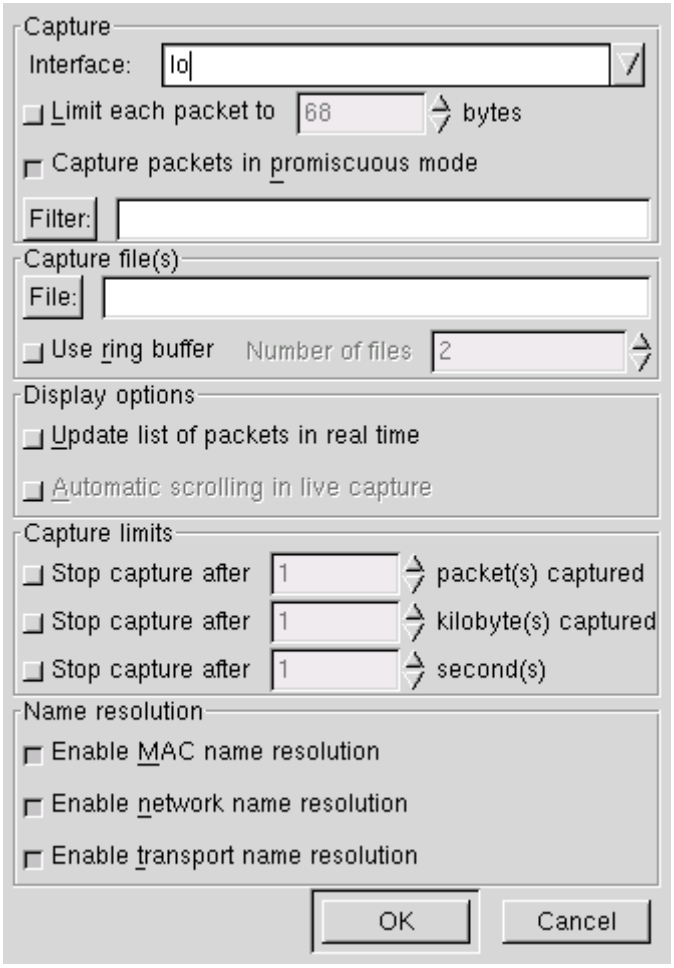


Ventana principal de WireShark luego de una captura.

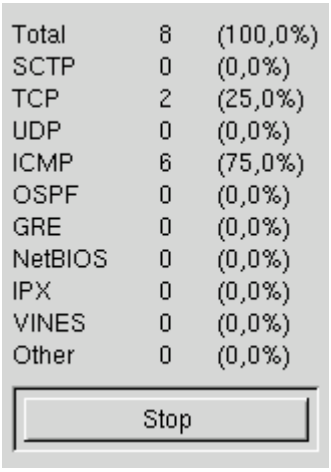
En la ventana principal de WireShark se reconocen tres áreas de despliegue:

- resumen de paquetes capturados, un paquete por línea; uno de ellos ha sido seleccionado como paquete actual (dando clic sobre la línea del paquete). Al desplazarse en la lista y cambiar el paquete actual se actualizan las otras dos ventanas, donde se despliega en dos formatos diferentes el contenido del paquete.
- detalles de encabezado de protocolos para el paquete seleccionado; los encabezados pueden abrirse (clic en +) para ver mayor detalle, o cerrarse (clic en -) para ocupar sólo una línea.
- datos crudos del paquete, representación hexadecimal y ASCII del encabezado del paquete seleccionado en el campo del medio.

Para iniciar la captura de datos, elegir las opciones de menú Capture: Start (Capturar, Comienzo). En la ventana de opciones de captura (ver figura), debe fijarse al menos la interfaz sobre la que se quiere realizar la captura. Los nombres varían según los sistemas operativos; la interfaz lo (loopback) permite enviar y recibir paquetes en la propia máquina.



Opciones de captura



Estado de captura

Para capturar en un archivo debe indicarse su nombre en el cuadro "Capture file(s)" de la ventana de Opciones de Captura (Capture: Start abre esta ventana). Estos archivos pueden ser examinados luego con el propio Ethereal mediante la opción de menú File:

Open. El tráfico ya capturado puede grabarse en un archivo eligiendo File: Print (Archivo: Imprimir); esta opción graba en formato legible (texto).

La ventana de estado muestra en tiempo real la cantidad de paquetes capturados, en total y de algunos tipos corrientes. La situación de captura se mantiene hasta que se presiona el botón Stop. Luego de unos instantes aparecen los paquetes capturados, tal cual se ve en la imagen de la ventana principal. Si se activó la opción de actualizar lista de paquetes en tiempo real ("Update list of packets in real time") estos se visualizan a medida que son capturados.

Ejercicio: captura, reconocimiento, grabar como texto, grabar en formato WireShark, visualizar.

1. Iniciar una captura (Capture: Start), elegir la interfaz lo. Arrancar la captura presionando el botón OK.
2. En ventana aparte, en una terminal de comandos, escribir `ping -c3 localhost`. Aguardar a que termine de ejecutarse el comando.
3. Terminar la captura (Stop en la ventana de captura de WireShark).
4. Examinar la captura. Reconocer los paquetes pedido - respuesta (request - reply) del protocolo ICMP (Internet Control Message Protocol) .
5. Grabar la captura, como texto, en un archivo: File: Print, tic en File, indicar un nombre de archivo (por ejemplo captura1.txt).
6. Visualizar el archivo; verificar que es legible.
7. Grabar la captura en el formato propio de WireShark: File: Save as...; indicar un nombre de archivo (por ejemplo captura1.ether). Este archivo no es legible como texto, debe ser visualizado con WireShark o con tcpdump.
8. Visualizar el archivo de captura en formato WireShark: File, Open, elegir el archivo. Verificar que se visualiza correctamente.

Si se desea examinar los paquetes a medida que van siendo capturados, activar en la ventana de Opciones de Captura "Update list of packets in real time" y "Automatic scrolling in live capture".

Opciones de invocación

En Unix, la página man de WireShark contiene sobre las opciones de invocación del programa cuando se lo arranca de la línea de comando. Algunas de las más usadas son:

`-h`: muestra versión y opciones

`-f <expresión>`: expresión filtro de captura (sintaxis de tcpdump)

`-i <interfaz>`: nombre de interfaz de escucha, tal como es mostrada en la salida de los comandos Unix `ifconfig -a` o `netstat -i`.

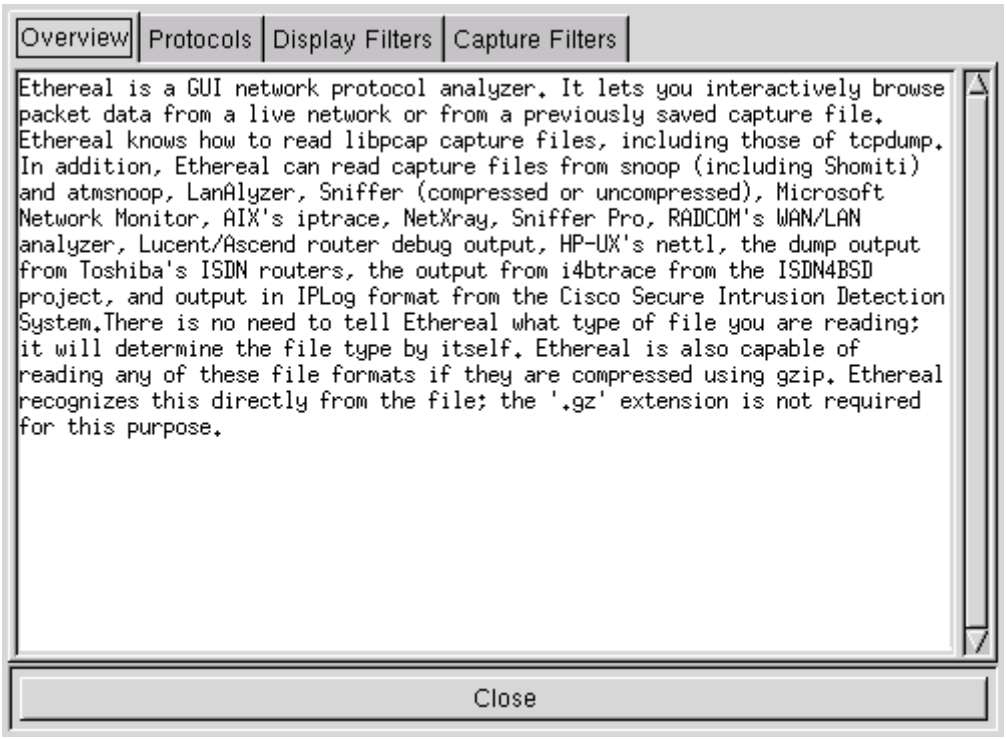
`-n`: deshabilita resolución de nombres (tales como nombres de máquinas y nombres de puertos TCP y UDP). Esta opción es útil para evitar ver el intercambio de paquetes originados en las consultas al servidor DNS para resolver nombres de máquinas.

`-r <archivo>`: lee los paquetes del archivo indicado en lugar de realizar una captura. El archivo debe haber sido generado con WireShark (Ethereal), tcpdump o algún otro analizador que use el mismo formato.

`-w <archivo>`: fija el nombre del archivo de captura.

Ayuda y documentación

La ventana de ayuda da una reseña del programa (Overview), lista los protocolos reconocidos, lista los nombres de los filtros posibles (Display filters) y refiere a la página man de tcpdump para la sintaxis de filtros de captura (Capture filters); la sintaxis de filtrado en la captura es diferente de la sintaxis de filtrado en el despliegue.



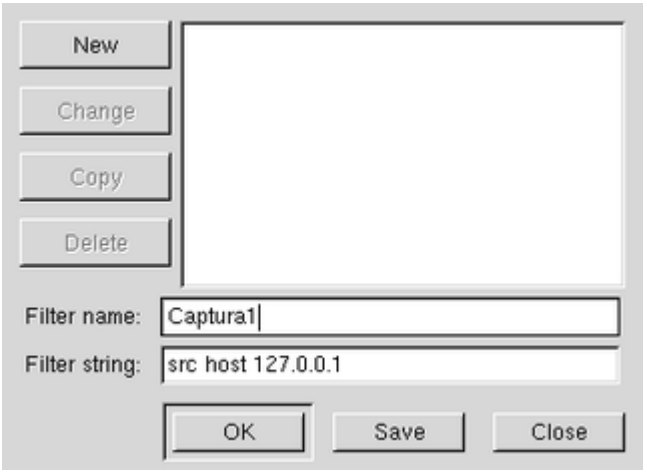
WireShark - Ayuda

Filtrado de paquetes

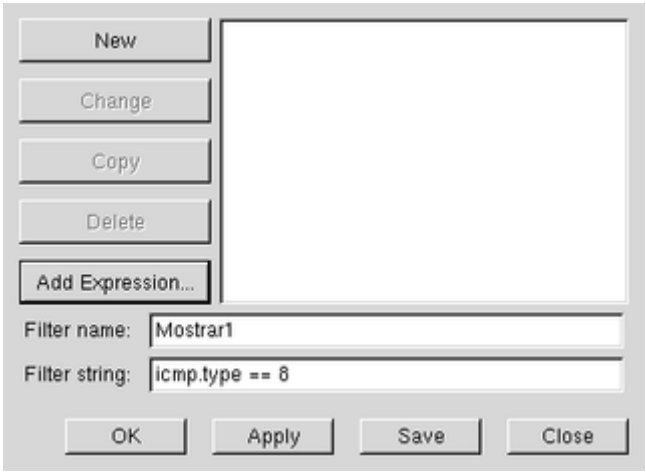
El filtrado de paquetes permite capturar o desplegar sólo aquellos paquetes de interés para el estudio en curso, desconociendo la existencia de otros. WireShark tiene dos modos de filtro distintos:

- filtro de captura: sólo se retienen los paquetes que cumplen la expresión filtro. Define lo que se guarda.
- filtro de despliegue: de los paquetes capturados, sólo se muestran los paquetes que cumplen la expresión filtro. Define lo que se ve de lo que hay guardado.

La sintaxis de escritura de ambos tipos de filtro es diferente. Los filtros de captura siguen la sintaxis del comando tcpdump y deben ser escritos en el cuadro Filter de la ventana de opciones de captura, antes de iniciar la captura. Los filtros de despliegue se fijan en el cuadro File de la ventana principal de WireShark. En ambos casos, presionando este botón File aparece un cuadro de diálogo que permite asignar un nombre a la expresión filtro construida.

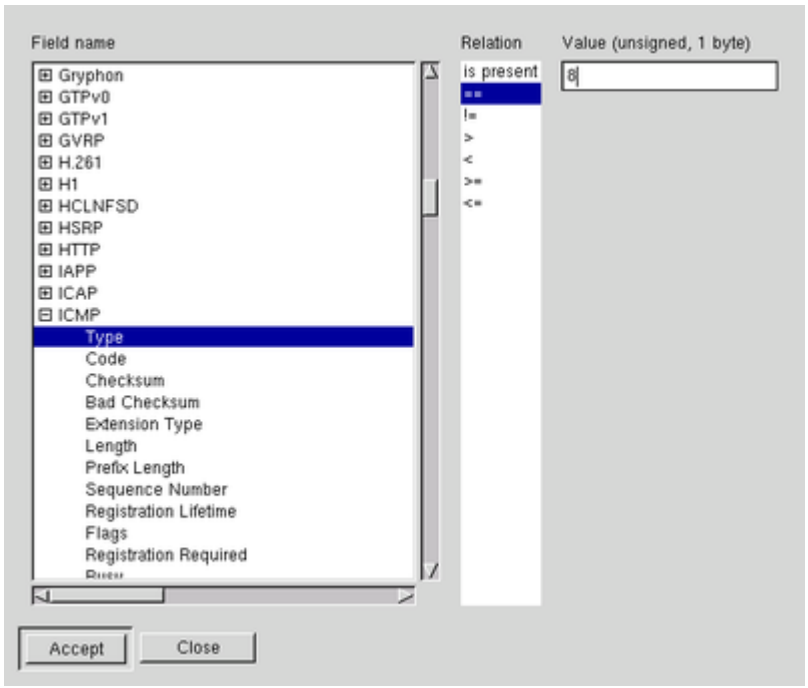


Diálogo para construir filtro de captura.



Diálogo para construir filtro de despliegue.

Para los filtros de despliegue existe una ayuda adicional en el botón Add Expression, que permite construir la expresión eligiendo el protocolo, sus campos y operadores relacionales.



Agregar expresión para filtro de despliegue

Filtro de captura

Un filtrado de paquetes por tipo puede lograrse en forma muy simple escribiendo el tipo en el cuadro Filter de la ventana principal: tcp, icmp, udp, etc. Para construcción de expresiones de filtro de captura se dispone de palabras claves (src, dst, host, net, len, etc.), operadores lógicos (and, or, not) y paréntesis. Los paquetes capturados deberán cumplir con la expresión booleana construida. La página man de tcpdump muestra todas las opciones. La tabla siguiente muestra un conjunto de ejemplos comunes.

dst host 192.167.1.1 src host 192.167.1.2 host 127.0.0.1	IP de destino (dst) IP de origen (src) IP de origen y de destino
dst net 10.137.101.0/24 src net 10.137.101.0/24 net 10.137.101.0/24	IPs de máquinas en la red destino IPs de máquinas en la red origen IPs de máquinas enla red origen y en la red destino
dst port 80 src port 80 port 80 tcp port 80 udp port 80	Puerto destino 80 en segmento TCP o datagrama UDP puerto origen 80 en segmento TCP o datagrama UDP puerto origen o destino 80 en segmento TCP o datagrama UDP puerto origen o destino 80 en segmento TCP puerto origen o destino 80 en datagrama UDP
len <= 200	largo de paquete menor que 200 bytes
icmp tcp udp ospf	el campo del protocolo IP fijado en el número correspondiente a ICMP, TCP, UDP u OSPF
ip proto 17	número de protocolo IP es 17
Broadcast	paquetes de difusión Ethernet
ip broadcast	paquete de difusión IP
Multicast	paquete multidifusión (multicast, destinos múltiples) de Ethernet
ip or arp	carga útil Ethernet es IP o ARP

Filtro de despliegue

La creación de filtros de despliegue es similar a la de captura, pero la sintaxis es diferente. Se construyen más fácilmente usando el constructor invocado por el botón Add Expression, donde pueden elegirse los campos de cada protocolo, los operadores, e ingresar los valores requeridos. Algunos ejemplos pueden verse en la siguiente tabla.

ip.dst==192.167.1.1 ip.src==192.167.1.2 ip.addr==192.167.1.1	campo dirección destino IP de una máquina campo dirección origen IP de una máquina campo dirección origen y campo dirección destino IP de una máquina
ip.dst==192.167.1.0/24 ip.src==192.167.1.0/24	campo dirección destino IP de una red campo dirección origen IP de una red campo dirección origen o campo dirección destino IP de una

ip.addr==192.167.1.0/24	red
tcp.destport==80 or udp.destport==80 tcp.srcport==80 or udp.srcport==80 tcp.port==80 or udp.port==80	Puerto destino es 80 en segmento TCP o datagrama UDP puerto origen es 80 en segmento TCP o datagrama UDP puerto origen y destino es 80 en segmento TCP o datagrama UDP
eth.len<=200	largo de paquete menor o igual a 200 bytes
icmp tcp udp ospf	el campo protocolo de paquete IP fijado en el número correspondiente a cada uno de los protocolos indicados (aquí se usan palabras clave en lugar de verificar el número de protocolo)
ip.proto==17	el campo de protocolo del paquete IP fijado en 17
eth.dst=ff:ff:ff:ff:ff:ff	paquete de difusión Ethernet
eth.dst[0]==1	paquete Ethernet multidifusión (multicast)
ip.dst=224.0.0.0/4	paqueteIP multidifusión (multicast)
ip or arp	carga útil Ethernet es IP o ARP

Referencias

- WireShark - Network protocol analyzer (http://www.wireshark.org).
- tcpdump, página man.
- Internet Lab Manual: http://www.cs.virginia.edu/~itlab/book/pdf/Introduction_v6c.pdf