

Actividad 6.2 Wireshark

Wuke Zhang

2-ASIR

Protocolos más conocidos:

1. HTTP (Hypertext Transfer Protocol):

- **Descripción:** Es el protocolo utilizado para la comunicación entre los navegadores web y los servidores web. Permite la transferencia de documentos en formato HTML y otros tipos de contenido, como imágenes, videos, etc.
- **Uso:** Cada vez que visitas una página web, tu navegador utiliza HTTP para hacer solicitudes y recibir respuestas del servidor web.

2. TCP (Transmission Control Protocol):

- **Descripción:** Protocolo de transporte confiable que garantiza la entrega ordenada de datos entre dos dispositivos a través de una red. Establece una conexión entre los dispositivos y se asegura de que los paquetes lleguen correctamente.
- **Uso:** Usado en protocolos como HTTP, FTP y muchos otros que requieren comunicación confiable.

3. DNS (Domain Name System):

- **Descripción:** Es el sistema de nombres de dominio de Internet, utilizado para traducir los nombres de dominio legibles (como www.google.com) a direcciones IP numéricas que las máquinas pueden entender.
- **Uso:** Sin DNS, no sería posible acceder a los sitios web utilizando sus nombres, ya que tendríamos que recordar direcciones IP.

Protocolos menos conocidos:

1. mDNS (Multicast DNS):

- **Descripción:** Protocolo utilizado para resolver nombres de host en redes locales sin necesidad de un servidor DNS central. Es útil para dispositivos que necesitan resolver nombres en una red local sin depender de un servidor DNS.
- **Uso:** Comúnmente usado en redes domésticas, dispositivos como impresoras, cámaras y otros dispositivos IoT.

2. NBNS (NetBIOS Name Service):

- **Descripción:** Es un protocolo de red usado para la resolución de nombres NetBIOS en una red local. Es utilizado principalmente en redes basadas en Windows.
- **Uso:** Facilita la identificación y localización de dispositivos y recursos compartidos en redes locales sin necesidad de un servidor DNS.

3. LLMNR (Link-Local Multicast Name Resolution):

- **Descripción:** Un protocolo utilizado para la resolución de nombres en redes locales, similar a mDNS, pero específicamente en redes basadas en Windows. Permite que los dispositivos se encuentren y resuelvan nombres de forma local, sin necesidad de un servidor central.
- **Uso:** Común en entornos de red de Windows para la resolución de nombres en redes locales.

Bueno abajo tengo relleno del curso pasado.

Parte 1: Investigación Teórica

Estructura de las Tramas

1. Ethernet

La trama Ethernet tiene la siguiente estructura:

Preamble (7 bytes): Síncrona a los receptores con una secuencia de 10101010. Start Frame Delimiter (SFD) (1 byte): Marca el final del preámbulo con 10101011. Destination MAC Address (6 bytes): Dirección MAC del destinatario.

Source MAC Address (6 bytes): Dirección MAC del remitente.

EtherType/Length (2 bytes): Indica el tipo de protocolo de la capa superior o la longitud de la trama. Payload/Data (46-1500 bytes): Datos transportados.

Frame Check Sequence (FCS) (4 bytes): CRC para la verificación de errores.

2. Point-to-Point Protocol (PPP)

La trama PPP tiene la siguiente estructura:

Flag (1 byte): Marca el inicio y el final de la trama con 01111110.

Address (1 byte): Valor constante (0xFF) indicando que se envía a todos los destinos. Control (1 byte): Valor constante (0x03) indicando que no hay secuencias.

Protocol (2 bytes): Identifica el protocolo encapsulado.

Payload/Data (variable): Datos transportados (máximo de 1500 bytes).

Frame Check Sequence (FCS) (2 o 4 bytes): CRC para la verificación de errores.

3. Frame Relay

La trama Frame Relay tiene la siguiente estructura:

Flag (1 byte): Marca el inicio y el final de la trama con 01111110.

Address (2-4 bytes): Contiene el DLCI (Data Link Connection Identifier) y otros bits de control.

Control (1 byte): Si está presente, generalmente tiene un valor de 0x03.

Protocol/Data (variable): Contiene los datos o el protocolo de nivel superior. Frame Check Sequence (FCS) (2 bytes): CRC para la verificación de errores.

Parte 2: Captura y Análisis de Tramas

Configuración del entorno de captura:

Ejemplo de eth

The screenshot shows the Wireshark network protocol analyzer interface. The top menu bar includes Archivo, Edición, Visualización, Ir, Captura, Analizar, Estadísticas, Telefonía, Wireless, Herramientas, and Ayuda. The interface is set to capture on the 'eth' interface. The packet list pane displays the following traffic:

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	192.168.77.31	192.168.76.1	DNS	92	Standard query 0x602
2	0.030516	192.168.77.31	192.168.76.1	DNS	92	Standard query 0x602
3	0.880988	192.168.76.1	192.168.77.31	DNS	215	Standard query respo
4	0.880988	192.168.76.1	192.168.77.31	DNS	215	Standard query respo
5	0.881643	192.168.77.31	104.208.16.90	TCP	66	50152 → 443 [SYN] Se
6	1.071236	104.208.16.90	192.168.77.31	TCP	66	443 → 50152 [SYN, AC
7	1.071298	192.168.77.31	104.208.16.90	TCP	54	50152 → 443 [ACK] Se
8	1.071616	192.168.77.31	104.208.16.90	TLSv1.2	285	Client Hello (SNI=mo
9	1.257356	104.208.16.90	192.168.77.31	TCP	1506	443 → 50152 [ACK] Se
10	1.257356	104.208.16.90	192.168.77.31	TCP	1506	443 → 50152 [ACK] Se
11	1.257356	104.208.16.90	192.168.77.31	TCP	1506	443 → 50152 [ACK] Se
12	1.257356	104.208.16.90	192.168.77.31	TCP	1506	443 → 50152 [ACK] Se

The packet details pane for the selected packet (Frame 1) shows the following structure:

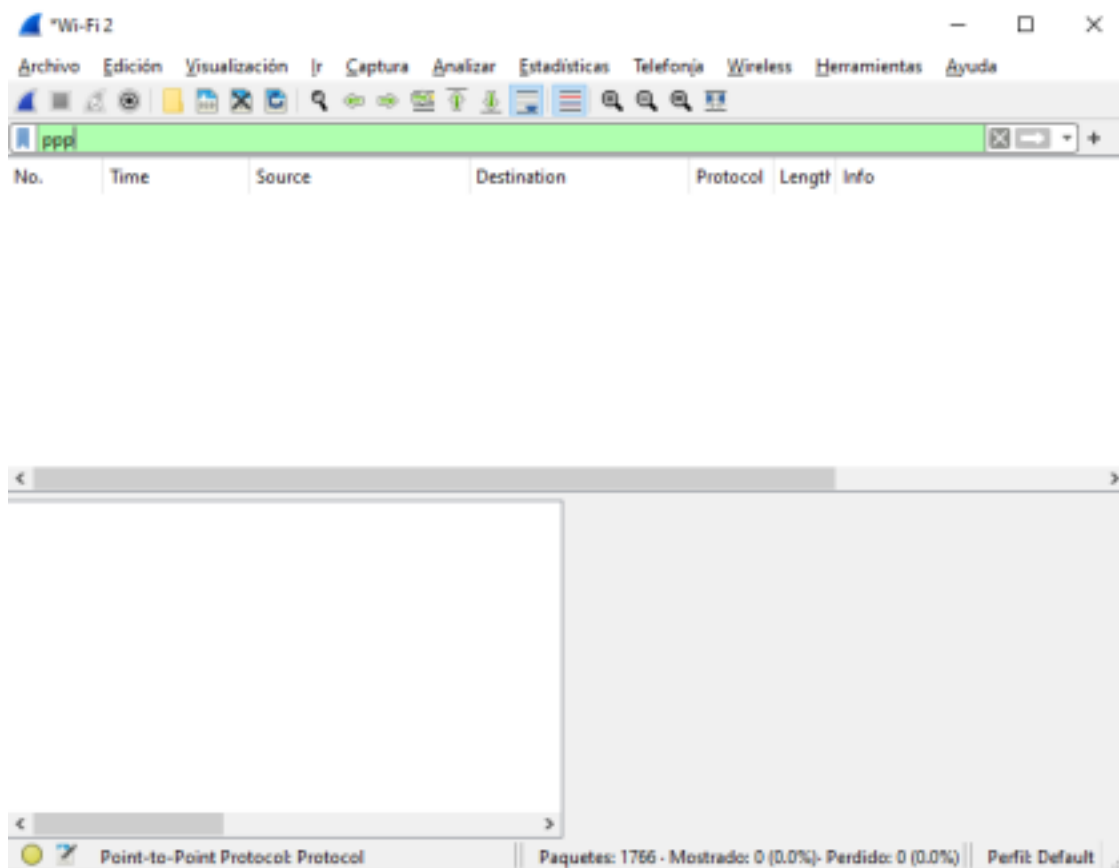
- > Frame 1: 92 bytes on wire (736 bits), 92 bytes captured
- > Ethernet II, Src: LiteonTechno_53:7b:75 (d8:f3:bc:53:7b:75), Dst: 08:00:0c:28:3e:02 (d8:f3:bc:08:00:0c:28:3e:02)
- > Internet Protocol Version 4, Src: 192.168.77.31, Dst: 192.168.76.1
- > User Datagram Protocol, Src Port: 52206, Dst Port: 53
- > Domain Name System (query)

The packet bytes pane shows the raw data in hexadecimal and ASCII format.

Use hasta el modo promiscuo para capturar el trafico de todo para ver si encontraba de PPP o Frame Relay pero es muy improbable debido a ciertas causas.

Ejemplo de PPP:

No aparece debido a que se muestran las capturas la primera vez que nos conectamos a internet pero nosotros estamos conectados siempre a internet es improbable que salga y ademas según lo que he



encontrado el set up de wireshark no captura este tipo o se requiere de otra version vieja de WinCap.

wincap.org/misc/faq.htm#Q-5

Q-2: After the installation, I cannot see WinPcap under the properties of my network adapter in control panel. Did anything go wrong?

A: No, if you have a recent version of WinPcap. As Q-1 says, recent versions appear under "add/remove programs" and not under network properties.

Q-3: How can I see if WinPcap is currently running on my Win2K/XP/2K3 machine?

A: Click on the Start button and then on Run. Type `cmd`. The System Information panel will show up. Choose Software Environment, then System Drivers. The entry `NPF` should appear there. If you launched a WinPcap application previously, the state should be running. Remember that WinPcap should have been run at least one time in order to appear in this list.

Q-4: The XXX WinPcap-based application doesn't run properly on my system. Is it a WinPcap problem?

A: Try `Windump`. In particular, `windump -G` reports the list of valid adapters and shows if WinPcap is able to detect correctly your hardware. If `Windump` works, the problem is in the XXX program and not in WinPcap, so contact the authors of XXX for help.

Q-5: Can I use WinPcap on a PPP connection?

A: Windows NT4. It's not possible to capture on PPP/VPN connections on this operating system.

Windows 2000/XP (x86/x64) (x86), these systems have limitations in the NDIS binding process that prevent a protocol driver from working properly on WAN adapters. WinPcap 3.1 and newer offer limited support for capturing on dial-up adapters using a wrapper over the Microsoft NetMon driver.

NOTES:

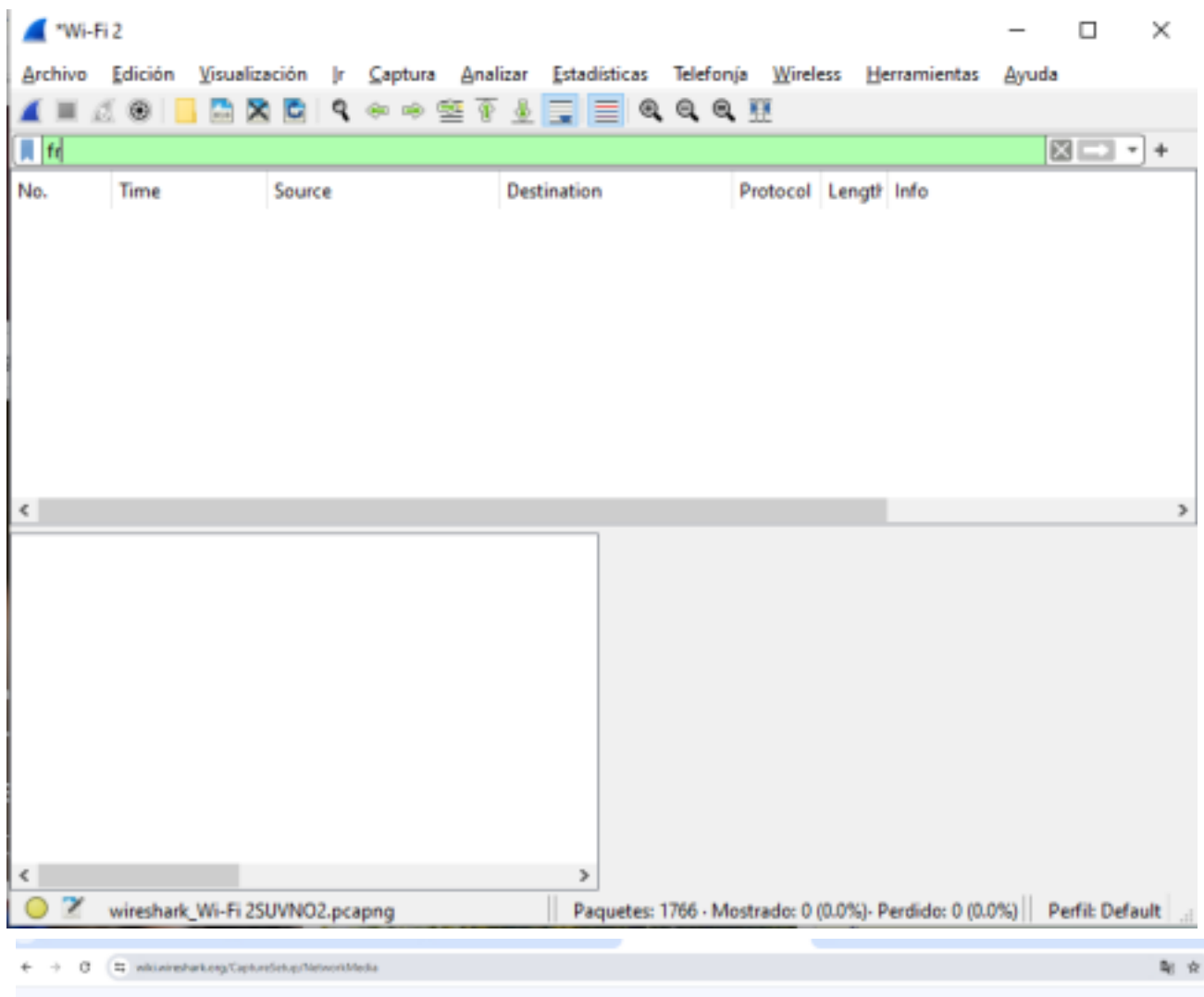
- It is possible to capture control packets (LCP and NCP) using the "Generic Dialup" or "Generic NdisWan" adapter (which is always listed even if no dialup connections are available). Control frames are captured as Ethernet encapsulated PPP frames.
- the PPP protocol is translated by the OS into a fake Ethernet. You'll see Ethernet frames and not PPP frames.
- transmission is not supported.
- filtering and statistics gathering is done at user level.

Windows XP (x86/x64) (x64), it's not possible to capture on PPP/VPN connections on these operating systems.

Windows Vista and more recent, it's not possible to capture on PPP/VPN connections on these operating systems.

Ejemplo de Frame Relay:

De Frame Relay mas de lo mismo debido a que es un protocolo muy viejo es por eso que no usa ya mucho y no se encuentra capturas ademas de que por lo visto no esta para wireshark con el SetUp de windows



Network media specific capturing

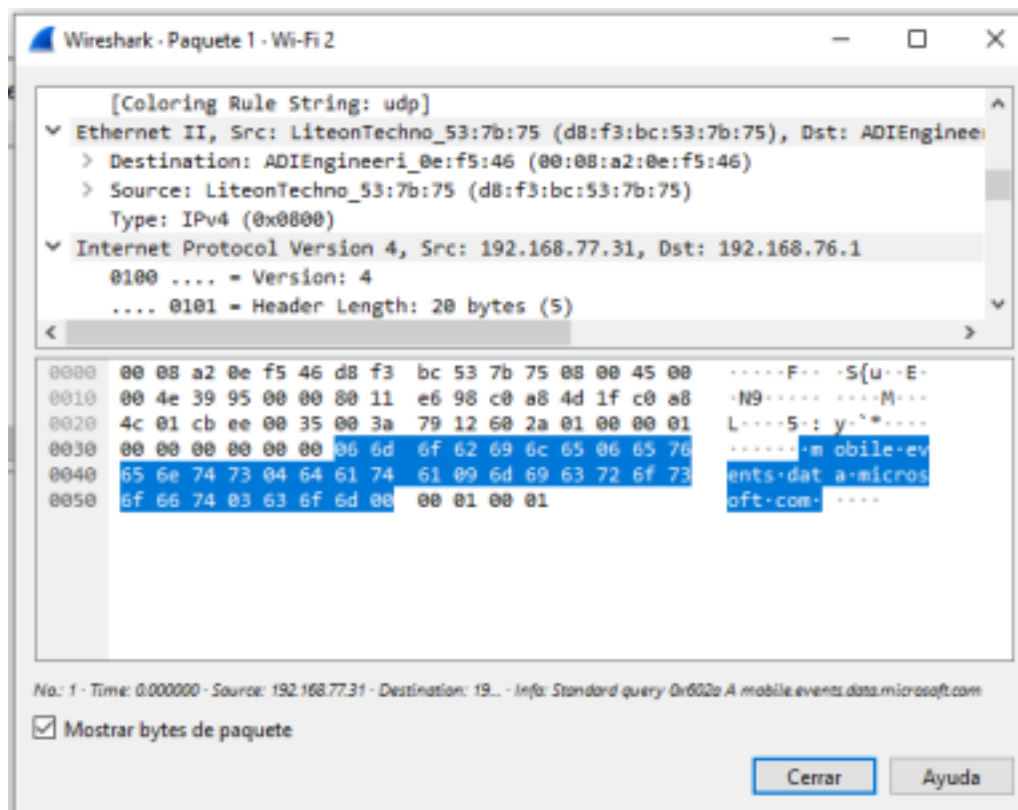
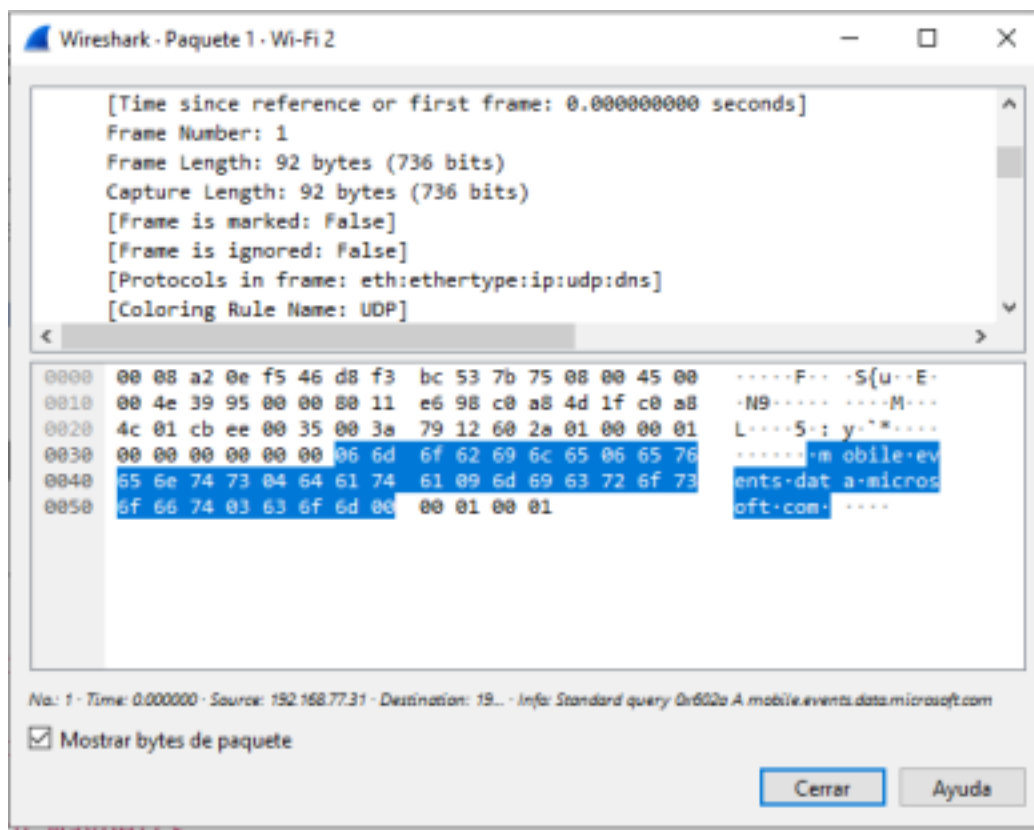
The capture library [libpcap](#) / [WinPcap](#), and the underlying packet capture mechanisms it uses, don't support capturing on all network types on all platforms: Wireshark and TShark use libpcap/WinPcap, and thus have the same limitations it does.

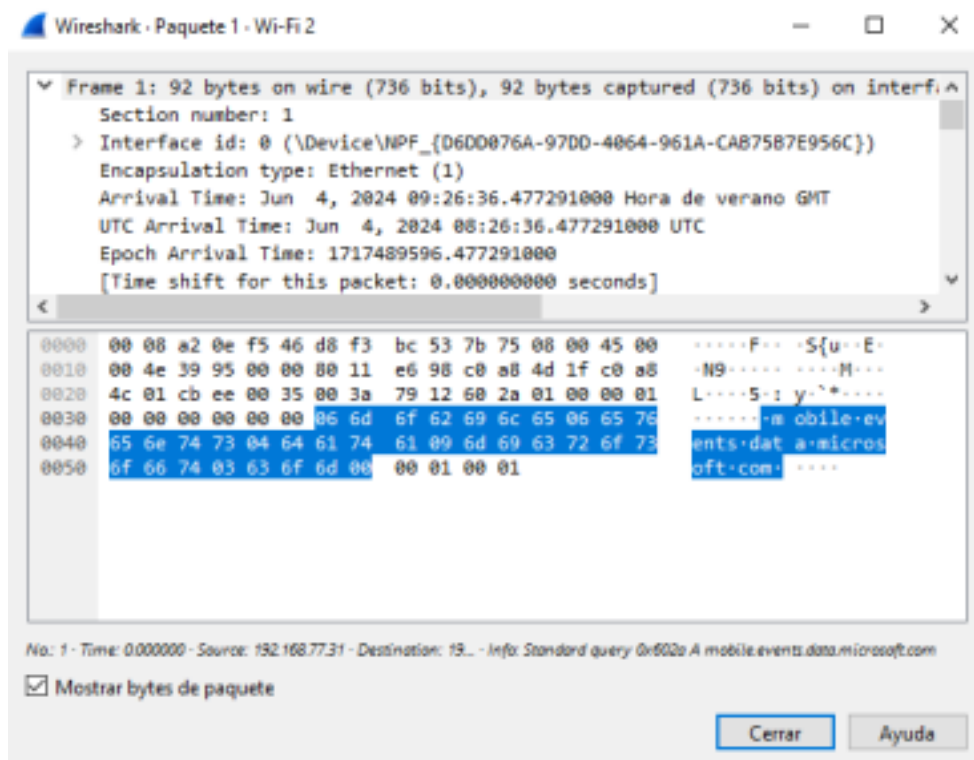
This is a table giving the network types supported on various platforms:

Interface	ADK	FreeBSD	HP-UX	Irix	Linux	macOS	NetBSD	OpenBSD	Solaris	Tru64 UNIX	Windows
ATM	?	?	?	?	✓	✗	?	?	✓	?	?
Bluetooth	✗	✗	✗	✗	✓ ¹	✗	✗	✗	✗	✗	✗
CiscoHDLC	?	✓	?	?	✓	?	✓	✓	?	?	?
Ethernet	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
FDDI	?	?	?	?	✓	✗	?	?	✓	?	?
FrameRelay	?	?	✗	✗	✓	✗	?	?	✗	✗	✗

Análisis de las tramas capturadas:

Dado que solo he encontrado de Ethernet voy a mostrar:





Tipo: IPv4
 Código: 0
 Bytes de Datos: 92 bytes (736 bits)
 Cabeceras IP 4
 Longitud de cabecera: 20 bytes (5)
 Longitud total: 78
 Bytes de datos: 50

Preambulo y SFD: Estos no se muestran explícitamente en Wireshark, ya que son parte del nivel físico y no se registran en las capturas de paquetes de red. La captura comienza típicamente con la dirección MAC de destino.

Destination: ADIEngineeri_0e:f5:46 (00:08:a2:0e:f5:46)
 Address: ADIEngineeri_0e:f5:46 (00:08:a2:0e:f5:46)
0. = LG bit: Globally unique address (factory default)
0. = IG bit: Individual address (unicast)

Source: LiteonTechno_53:7b:75 (d8:f3:bc:53:7b:75)
 Address: LiteonTechno_53:7b:75 (d8:f3:bc:53:7b:75)
0. = LG bit: Globally unique address (factory default)
0. = IG bit: Individual address (unicast)