

Fundamentos de Redes



Seminario 1



2017

Fundamentos de Redes

Una vez instaladas las 2 máquinas, procedemos a realizar la configuración de la red, teniendo en cuenta que estas máquinas están conectadas a través de una red interna.

Máquina 1:

Nombre de la conexión: **Conexión cableada 2**

☒ Conectar automáticamente

Cableada Seguridad 802.1x Ajustes de IPv4 Ajustes de IPv6

Método: **Manual**

Dirección

Dirección	Máscara de red	Puerta de enlace	Añadir	Eliminar
10.0.0.1	255.255.255.0	0.0.0.0		

Servidores DNS:

Domínios de búsqueda:

ID del cliente DHCP:

☐ Requiere dirección IPv4 para que esta conexión se complete

Rutas...

☒ Disponible para todos los usuarios

Máquina 2:

Nombre de la conexión: **Conexión cableada 2**

☒ Conectar automáticamente

Cableada Seguridad 802.1x Ajustes de IPv4 Ajustes de IPv6

Método: **Manual**

Dirección

Dirección	Máscara de red	Puerta de enlace	Añadir	Eliminar
10.0.0.2	255.255.255.0	0.0.0.0		

Servidores DNS:

Domínios de búsqueda:

ID del cliente DHCP:

☐ Requiere dirección IPv4 para que esta conexión se complete

Rutas...

☒ Disponible para todos los usuarios

Ahora, pasamos a realizar un ping entre ellas para comprobar su conexión:

1

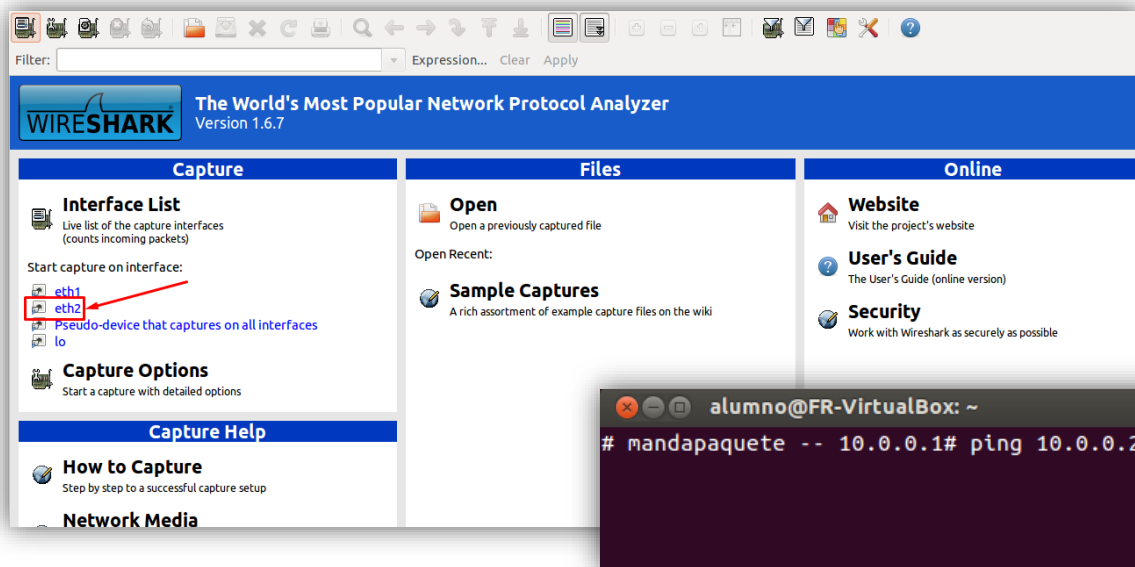
```
alumno@FR-VirtualBox: ~  
# mandapaquete -- 10.0.0.1# ping 10.0.0.2 -c 7  
PING 10.0.0.2 (10.0.0.2) 56(84) bytes of data.  
64 bytes from 10.0.0.2: icmp_req=1 ttl=64 time=0.289 ms  
64 bytes from 10.0.0.2: icmp_req=2 ttl=64 time=0.451 ms  
64 bytes from 10.0.0.2: icmp_req=3 ttl=64 time=0.791 ms  
64 bytes from 10.0.0.2: icmp_req=4 ttl=64 time=0.652 ms  
64 bytes from 10.0.0.2: icmp_req=5 ttl=64 time=0.263 ms  
64 bytes from 10.0.0.2: icmp_req=6 ttl=64 time=0.685 ms  
64 bytes from 10.0.0.2: icmp_req=7 ttl=64 time=0.835 ms  
--- 10.0.0.2 ping statistics ---  
7 packets transmitted, 7 received, 0% packet loss, time 5997ms  
rtt min/avg/max/mdev = 0.263/0.566/0.835/0.217 ms  
# mandapaquete -- 10.0.0.1#
```

```
alumno@FR-VirtualBox: ~  
# recibepaquete -- 10.0.0.2# ping 10.0.0.1 -c 7  
PING 10.0.0.1 (10.0.0.1) 56(84) bytes of data.  
64 bytes from 10.0.0.1: icmp_req=1 ttl=64 time=0.270 ms  
64 bytes from 10.0.0.1: icmp_req=2 ttl=64 time=0.809 ms  
64 bytes from 10.0.0.1: icmp_req=3 ttl=64 time=0.242 ms  
64 bytes from 10.0.0.1: icmp_req=4 ttl=64 time=0.285 ms  
64 bytes from 10.0.0.1: icmp_req=5 ttl=64 time=0.260 ms  
64 bytes from 10.0.0.1: icmp_req=6 ttl=64 time=0.291 ms  
64 bytes from 10.0.0.1: icmp_req=7 ttl=64 time=0.312 ms  
--- 10.0.0.1 ping statistics ---  
7 packets transmitted, 7 received, 0% packet loss, time 5996ms  
rtt min/avg/max/mdev = 0.242/0.352/0.809/0.188 ms  
# recibepaquete -- 10.0.0.2#
```

Podemos ver como efectivamente hay conexión entre las dos máquinas que tenemos instaladas y conectadas a través de una red interna.

Fundamentos de Redes

Procedemos ahora a la ejecución de Wireshark para leer estos paquetes ping:



Una vez empezado el escaneo, mandamos el ping a la máquina que esta escaneando la red y podemos ver a continuación como ésta recibe los paquetes:

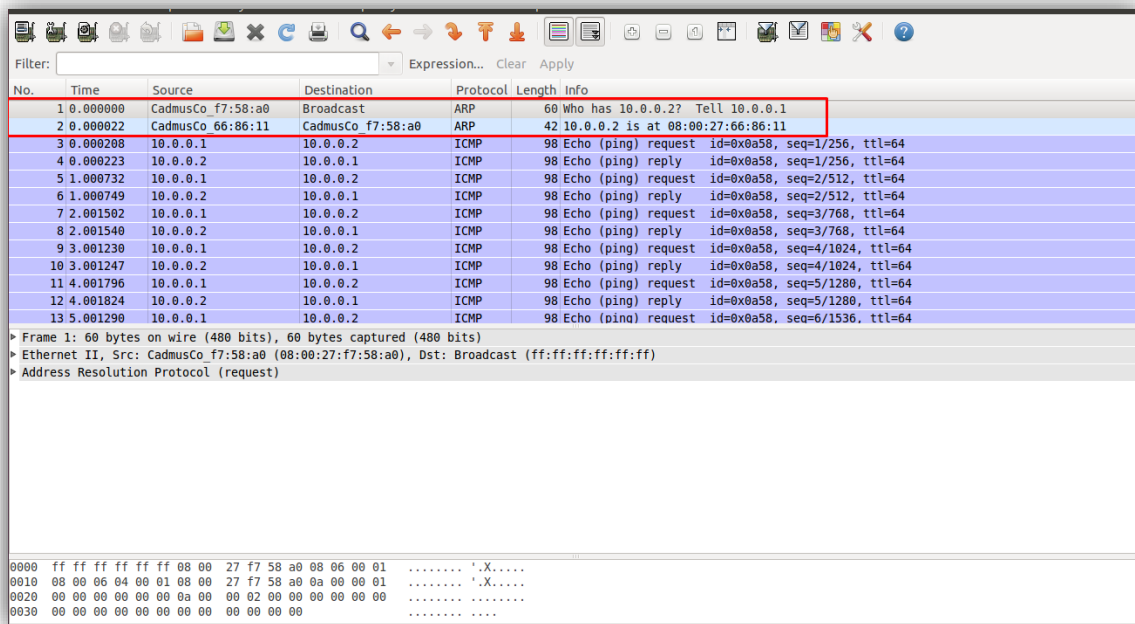
2

The screenshot shows the Wireshark 1.6.7 interface with a list of captured packets. A red box highlights the first 14 packets, which are ICMP Echo (ping) requests and replies. The table below shows the details of these packets.

No.	Time	Source	Destination	Protocol	Length	Info
12	4.001824	10.0.0.2	10.0.0.1	ICMP	98	Echo (ping) reply id=0x0a58, seq=5/1280, ttl=64
13	5.001290	10.0.0.1	10.0.0.2	ICMP	98	Echo (ping) request id=0x0a58, seq=6/1536, ttl=64
14	5.001329	10.0.0.2	10.0.0.1	ICMP	98	Echo (ping) reply id=0x0a58, seq=6/1536, ttl=64
15	5.010804	CadmusCo_66:86:11	CadmusCo_f7:58:a0	ARP	42	Who has 10.0.0.1? Tell 10.0.0.2
16	5.011717	CadmusCo_f7:58:a0	CadmusCo_66:86:11	ARP	60	10.0.0.1 is at 08:00:27:f7:58:a0
17	6.000620	10.0.0.1	10.0.0.2	ICMP	98	Echo (ping) request id=0x0a58, seq=7/1792, ttl=64
18	6.000638	10.0.0.2	10.0.0.1	ICMP	98	Echo (ping) reply id=0x0a58, seq=7/1792, ttl=64
19	7.001420	10.0.0.1	10.0.0.2	ICMP	98	Echo (ping) request id=0x0a58, seq=8/2048, ttl=64
20	7.001437	10.0.0.2	10.0.0.1	ICMP	98	Echo (ping) reply id=0x0a58, seq=8/2048, ttl=64
21	8.001821	10.0.0.1	10.0.0.2	ICMP	98	Echo (ping) request id=0x0a58, seq=9/2304, ttl=64
22	8.001865	10.0.0.2	10.0.0.1	ICMP	98	Echo (ping) reply id=0x0a58, seq=9/2304, ttl=64
23	9.003155	10.0.0.1	10.0.0.2	ICMP	98	Echo (ping) request id=0x0a58, seq=10/2560, ttl=64
24	9.003177	10.0.0.2	10.0.0.1	ICMP	98	Echo (ping) reply id=0x0a58, seq=10/2560, ttl=64

Frame 1: 60 bytes on wire (480 bits), 60 bytes captured (480 bits)
Ethernet II, Src: CadmusCo_f7:58:a0 (08:00:27:f7:58:a0), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
Address Resolution Protocol (request)

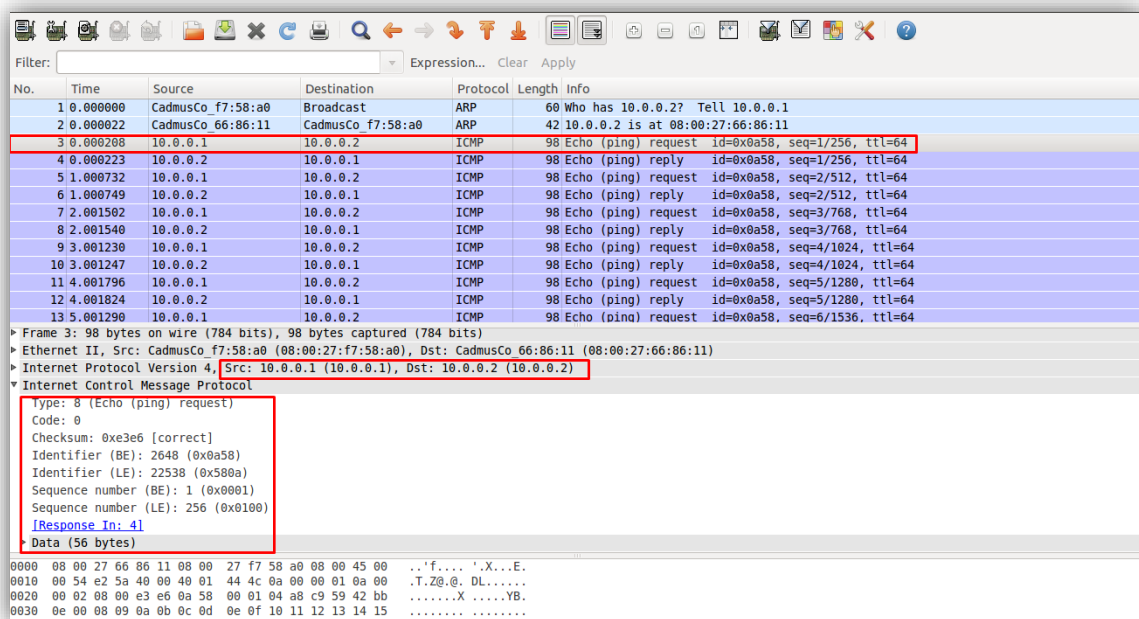
Podemos ver varios tipos de paquetes, como pueden ser:



The image shows a Wireshark packet capture. The top toolbar includes icons for file operations, network analysis, and search. Below the toolbar is a filter bar with 'Filter:' and 'Expression...' fields. The main pane displays a list of captured packets. The first two packets are ARP requests from CadmusCo_f7:58:a0 to Broadcast. The subsequent packets are ICMP Echo (ping) requests and replies between 10.0.0.1 and 10.0.0.2. The bottom pane shows the raw packet data in hexadecimal and ASCII.

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	CadmusCo_f7:58:a0	Broadcast	ARP	60	Who has 10.0.0.2? Tell 10.0.0.1
2	0.000022	CadmusCo_66:86:11	CadmusCo_f7:58:a0	ARP	42	10.0.0.2 is at 08:00:27:66:86:11
3	0.000208	10.0.0.1	10.0.0.2	ICMP	98	Echo (ping) request id=0xa58, seq=1/256, ttl=64
4	0.000223	10.0.0.2	10.0.0.1	ICMP	98	Echo (ping) reply id=0xa58, seq=1/256, ttl=64
5	1.000732	10.0.0.1	10.0.0.2	ICMP	98	Echo (ping) request id=0xa58, seq=2/512, ttl=64
6	1.000749	10.0.0.2	10.0.0.1	ICMP	98	Echo (ping) reply id=0xa58, seq=2/512, ttl=64
7	2.001502	10.0.0.1	10.0.0.2	ICMP	98	Echo (ping) request id=0xa58, seq=3/768, ttl=64
8	2.001540	10.0.0.2	10.0.0.1	ICMP	98	Echo (ping) reply id=0xa58, seq=3/768, ttl=64
9	3.001230	10.0.0.1	10.0.0.2	ICMP	98	Echo (ping) request id=0xa58, seq=4/1024, ttl=64
10	3.001247	10.0.0.2	10.0.0.1	ICMP	98	Echo (ping) reply id=0xa58, seq=4/1024, ttl=64
11	4.001796	10.0.0.1	10.0.0.2	ICMP	98	Echo (ping) request id=0xa58, seq=5/1280, ttl=64
12	4.001824	10.0.0.2	10.0.0.1	ICMP	98	Echo (ping) reply id=0xa58, seq=5/1280, ttl=64
13	5.001290	10.0.0.1	10.0.0.2	ICMP	98	Echo (ping) request id=0xa58, seq=6/1536, ttl=64

En esta imagen vemos como se envían paquetes para que la maquina emisora pueda saber quién es la máquina receptora, preguntando a través del protocolo ARP qué dirección MAC corresponde a una determinada IP (en este caso la 10.0.0.2)



The image shows a Wireshark packet capture with the details pane expanded for the third packet (ICMP Echo (ping) request). The details pane shows the packet structure: Type: 8 (Echo (ping) request), Code: 0, Checksum: 0xe3e6 [correct], Identifier (BE): 2648 (0xa58), Identifier (LE): 22538 (0x580a), Sequence number (BE): 1 (0x0001), Sequence number (LE): 256 (0x0100), and Data (56 bytes). The packet list pane shows the same packet with the details pane expanded.

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	CadmusCo_f7:58:a0	Broadcast	ARP	60	Who has 10.0.0.2? Tell 10.0.0.1
2	0.000022	CadmusCo_66:86:11	CadmusCo_f7:58:a0	ARP	42	10.0.0.2 is at 08:00:27:66:86:11
3	0.000208	10.0.0.1	10.0.0.2	ICMP	98	Echo (ping) request id=0xa58, seq=1/256, ttl=64
4	0.000223	10.0.0.2	10.0.0.1	ICMP	98	Echo (ping) reply id=0xa58, seq=1/256, ttl=64
5	1.000732	10.0.0.1	10.0.0.2	ICMP	98	Echo (ping) request id=0xa58, seq=2/512, ttl=64
6	1.000749	10.0.0.2	10.0.0.1	ICMP	98	Echo (ping) reply id=0xa58, seq=2/512, ttl=64
7	2.001502	10.0.0.1	10.0.0.2	ICMP	98	Echo (ping) request id=0xa58, seq=3/768, ttl=64
8	2.001540	10.0.0.2	10.0.0.1	ICMP	98	Echo (ping) reply id=0xa58, seq=3/768, ttl=64
9	3.001230	10.0.0.1	10.0.0.2	ICMP	98	Echo (ping) request id=0xa58, seq=4/1024, ttl=64
10	3.001247	10.0.0.2	10.0.0.1	ICMP	98	Echo (ping) reply id=0xa58, seq=4/1024, ttl=64
11	4.001796	10.0.0.1	10.0.0.2	ICMP	98	Echo (ping) request id=0xa58, seq=5/1280, ttl=64
12	4.001824	10.0.0.2	10.0.0.1	ICMP	98	Echo (ping) reply id=0xa58, seq=5/1280, ttl=64
13	5.001290	10.0.0.1	10.0.0.2	ICMP	98	Echo (ping) request id=0xa58, seq=6/1536, ttl=64

Por último, en esta imagen podemos ver tanto como llegan los paquetes del ping a la maquina receptora tanto como esta misma maquina receptora devuelve la respuesta a la que ha mandado el ping.

En la información del paquete, podemos ver como efectivamente lo que se recibe es un ping que no es más que una conexión por la que se mandan 56 bytes de datos al receptor y así poder comprobar la conectividad con otra máquina a la que se está conectado.