Redes: Capa de enlace

Grupo: 96

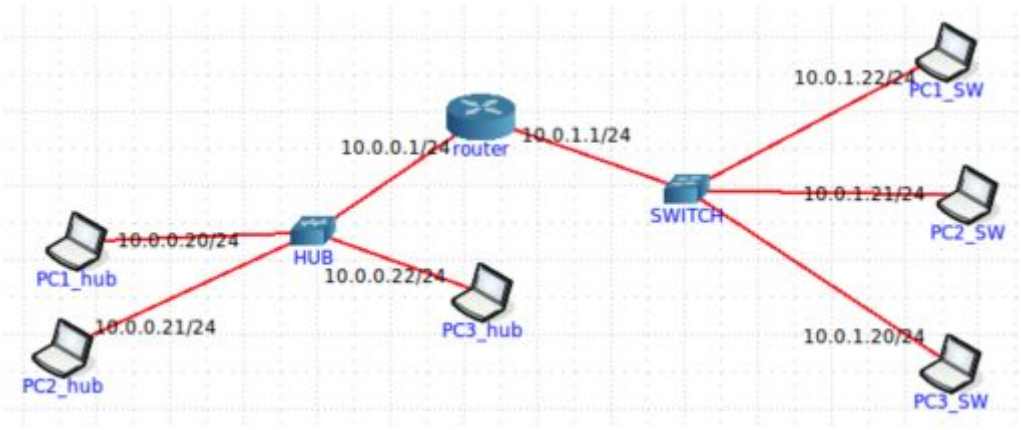
Autores

Valentín Colato 15655/7
Nicolas Cesar Champane Peñalva 15974/9
Nahuel Bigurrarena 15782/3

Practica 11

1. (Ejercicio de promoción) Utilizando la máquina virtual provista por la cátedra, arme una red como la siguiente, con un segmento de LAN usando un HUB y otro segmento de LAN usando un SWITCH.

NOTA: para quienes hagan la promoción, este será un ejercicio entregable. En la entrega deberán estar todas las preguntas respondidas y debidamente justificadas. En los puntos donde es necesario ejecutar comandos, los mismos deberán adjuntarse a la entrega.



a. Antes de empezar el ejercicio ejecute en una terminal el siguiente comando:

sudo iptables -P FORWARD ACCEPT

b. Analizar el funcionamiento de ARP.

I. Indique para PC1_SW, PC2_SW y PC3_SW la IP y la dirección MAC de cada una.

Creamos la tabla con todas las IPS y MAC de las PC para usarla como referencia para los siguientes ejercicios.

Las direcciones MAC fueron asignadas automáticamente.

Dispositivos	IP	MAC
PC1_hub	10.0.0.20/24	00:00:00:aa:00:00
PC2_hub	10.0.0.21/24	00:00:00:aa:00:01
PC3_hub	10.0.0.22/24	00:00:00:aa:00:02

PC1_SW	10.0.1.22/24	00:00:00:aa:00:05
PC2_SW	10.0.1.21/24	00:00:00:aa:00:06
PC3_SW	10.0.1.20/24	00:00:00:aa:00:07
Router con HUB	10.0.0.1/24	00:00:00:aa:00:03
Router con SW	10.0.1.1/24	00:00:00:aa:00:04

ii. Verifique el contenido de la tabla ARP de cada una de ellas.

Utilizamos el comando **"*arp -an*"** para ver la tabla de arp, donde no nos devolvio nada ya que no existe ninguna dirección ip y mac conocida, debido a que no se hizo un intento de conexión con algún dispositivo de la red.

iii. Inicie Wireshark en PC2_SW y luego envíe un ping desde la PC1_SW a la PC2_SW. Analice los paquetes ARP e ICMP capturados e indique:

Para ARP: tipo de paquete, direcciones de capa 2 y datos específicos del protocolo.

1-

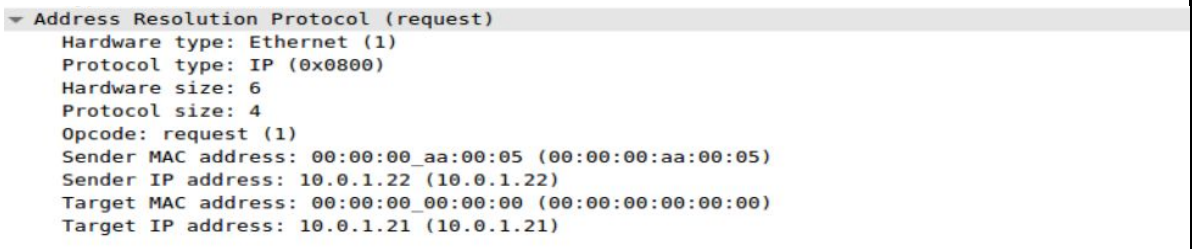
Tipo de paquete: **ARP Request**

Direcciones de capa 2:

Src: 00:00:00:aa:00:05

Dst: ff:ff:ff:ff:ff:ff

Datos:



El emisor envía una solicitud ARP (ARP Request) con la dirección IP de destino a todos los hosts de la red. Para tal fin, el emisor utiliza la dirección de broadcast de ARP FF:FF:FF:FF:FF:FF como dirección del destinatario.

Dentro de los datos de ARP se encuentran una tabla con la IP y MAC del emisor y la del destinatario(la cual se quiere saber su MAC). Se completan las del emisor con sus datos y para el destinatario se pone la IP destino y la MAC en 00:00:00:00:00:00 indicando que está incompleta, la cual debe ser completada por el mismo.

2-

```
10 39.11904000( 00:00:00_aa:00:06     00:00:00_aa:00:05     ARP          42 10.0.1.21 is at 00:00:00:aa:00:06
```

Tipo de paquete: **ARP Reply**

Direcciones de capa 2:

Src: 00:00:00:aa:00:06

Dst: 00:00:00:aa:00:05

Datos:

```
▾ Address Resolution Protocol (reply)
    Hardware type: Ethernet (1)
    Protocol type: IP (0x0800)
    Hardware size: 6
    Protocol size: 4
    Opcode: reply (2)
    Sender MAC address: 00:00:00_aa:00:06 (00:00:00:aa:00:06)
    Sender IP address: 10.0.1.21 (10.0.1.21)
    Target MAC address: 00:00:00_aa:00:05 (00:00:00:aa:00:05)
    Target IP address: 10.0.1.22 (10.0.1.22)
```

El receptor al recibir el mensaje de ARP (ARP Request) revisa si está dirigida hacia él, si así fuera el caso, completará el campo de Target MAC con su propia MAC. Y se lo envía al emisor con el mensaje ARP Reply.

Para ICMP: tipo de paquete, direcciones de capa 2, de capa 3, tipo y código ICMP.

1-

```
11 39.11904200( 10.0.1.22          10.0.1.21          ICMP         98 Echo (ping) request  id=0x0012, seq=1/256, ttl=64 (reply in 12)
```

Tipo de paquete: **Echo (ping) request**

Direcciones capa 2:

Src: 00:00:00:aa:00:05

Dst: 00:00:00:aa:00:06

Direcciones capa 3:

Src: 10.0.1.22

Dst: 10.0.1.21

Tipo: 8

Código ICMP: 0

```
▸ Internet Control Message Protocol
    Type: 8 (Echo (ping) request)
    Code: 0
    Checksum: 0xebfc [correct]
    Identifier (BE): 18 (0x0012)
    Identifier (LE): 4608 (0x1200)
    Sequence number (BE): 1 (0x0001)
    Sequence number (LE): 256 (0x0100)
    [Response frame: 12]
    Timestamp from icmp data: Dec  6, 2020 03:01:34.000000000 GMT
    [Timestamp from icmp data (relative): 0.226343000 seconds]
```

2-

```
12 39.11906300( 10.0.1.21        10.0.1.22        ICMP     98 Echo (ping) reply    id=0x0012, seq=1/256, ttl=64 (request in 11)
```

Tipo de paquete: **Echo (ping) reply**

Direcciones capa 2:

Src: 00:00:00:aa:00:06

Dst: 00:00:00:aa:00:05

Direcciones capa 3:

Src: 10.0.1.21

Dst: 10.0.1.22

Tipo: 0

Código ICMP: 0

```
▾ Internet Control Message Protocol
    Type: 0 (Echo (ping) reply)
    Code: 0
    Checksum: 0xf3fc [correct]
    Identifier (BE): 18 (0x0012)
    Identifier (LE): 4608 (0x1200)
    Sequence number (BE): 1 (0x0001)
    Sequence number (LE): 256 (0x0100)
    [Request frame: 11]
    [Response time: 0,021 ms]
    Timestamp from icmp data: Dec  6, 2020 03:01:34.000000000 GMT
    [Timestamp from icmp data (relative): 0.226364000 seconds]
```

El comando ping funciona con ICMP y mensaje Echo.

El **Echo Request** (Petición eco) es un mensaje de control que se envía a un host con la expectativa de recibir de él un **Echo Reply** (Respuesta eco).

iv. Verifique nuevamente el contenido de la tabla ARP de las PCs ni bien termine de ejecutar el comando ping. ¿Qué entradas aparecen en cada tabla y por qué? ¿Qué estado tienen (ip neigh ls)?

PC2sw:

```
root@PC2sw:/tmp/pycore.36767/PC2sw.conf# arp -an
? (10.0.1.22) at 00:00:00:aa:00:05 [ether] on eth0
root@PC2sw:/tmp/pycore.36767/PC2sw.conf# ip neigh ls
10.0.1.22 dev eth0 lladdr 00:00:00:aa:00:05 REACHABLE
```

PC1sw:

```
root@PC1sw:/tmp/pycore.36767/PC1sw.conf# arp -an
? (10.0.1.21) at 00:00:00:aa:00:06 [ether] on eth0
root@PC1sw:/tmp/pycore.36767/PC1sw.conf# ip neigh ls
10.0.1.21 dev eth0 lladdr 00:00:00:aa:00:06 REACHABLE
root@PC1sw:/tmp/pycore.36767/PC1sw.conf#
```

Para cada tabla ARP de las PC 1 y 2 , se guarda la dirección ip de la otra PC con su dirección MAC . Esto sucede al recibir el ARP dónde guarda en su tabla ARP personal el valor de mac recibido, con la dirección IP correspondiente.

Con el comando “ip negih ls” nos marca que la entrada de la tabla ARP tiene el estado de REACHABLE que significa que la entrada es válida y que hay conexión establecida, pasado unos minutos el estado pasa a ser STALE lo cual quiere decir que la entrada de la tabla es válida pero no se sabe si es alcanzable, por lo tanto tendrá que volver a consultarla. En caso de usarse la entrada la PC intentará comprobar su validez.

v. Borre las entradas de las tablas ARP de ambas PC y agregue de forma estática en PC1_SW la entrada que corresponde a PC2_SW y en PC2_SW la que corresponde a PC1_SW. Si hiciera un ping de PC1_SW a PC2_SW, ¿se verían paquetes de ARP? Verifíquelo en la máquina virtual iniciando una captura de tráfico en PC2_SW. ¿Qué estado tienen ahora las entradas ARP?

Para borrar las entradas de las tablas utilizamos los siguientes comandos:

1. arp -d 10.0.1.22
2. arp -d 10.0.1.21

Luego para agregar de forma estática en PC1_SW:

```
root@PC2sw:/tmp/pycore.36767/PC2sw.conf# arp -s 10.0.1.22 00:00:00:aa:00:05
root@PC2sw:/tmp/pycore.36767/PC2sw.conf# arp -an
? (10.0.1.22) at 00:00:00:aa:00:05 [ether] PERM on eth0
```

En PC2_SW:

```
root@PC1sw:/tmp/pycore.36767/PC1sw.conf# arp -s 10.0.1.21 00:00:00:aa:00:06
root@PC1sw:/tmp/pycore.36767/PC1sw.conf# arp -an
? (10.0.1.21) at 00:00:00:aa:00:06 [ether] PERM on eth0
```

En PC1sw : Ping 10.0.1.21
Captura PC2sw:

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	10.0.1.1	224.0.0.5	OSPF	78	Hello Packet
2	0.620579000	fe80::200:ff:feaa:4	ff02::5	OSPF	90	Hello Packet
3	7.468306000	10.0.1.22	10.0.1.21	ICMP	98	Echo (ping) request id=0x0023, seq=1/256, ttl=64 (reply in 4)
4	7.468320000	10.0.1.21	10.0.1.22	ICMP	98	Echo (ping) reply id=0x0023, seq=1/256, ttl=64 (request in 3)
5	8.467300000	10.0.1.22	10.0.1.21	ICMP	98	Echo (ping) request id=0x0023, seq=2/512, ttl=64 (reply in 6)
6	8.467315000	10.0.1.21	10.0.1.22	ICMP	98	Echo (ping) reply id=0x0023, seq=2/512, ttl=64 (request in 5)

Al momento de realizar el ping de PC1_SW a PC2_SW realizamos la captura no aparece el protocolo de ARP debido a que ya conoce la MAC destino, la cual fue agregada de forma estática anteriormente.

El estado en la tabla ARP de las PCs es PERM,se le asigna este estado al agregarlo de forma estática, indicando su condición de permanencia.

vi. En PC1_SW modifique la entrada ARP que agregó en el punto anterior poniendo una MAC que no exista en la red. Vuelva a intentar hacer el ping. ¿Qué ocurre y por qué?

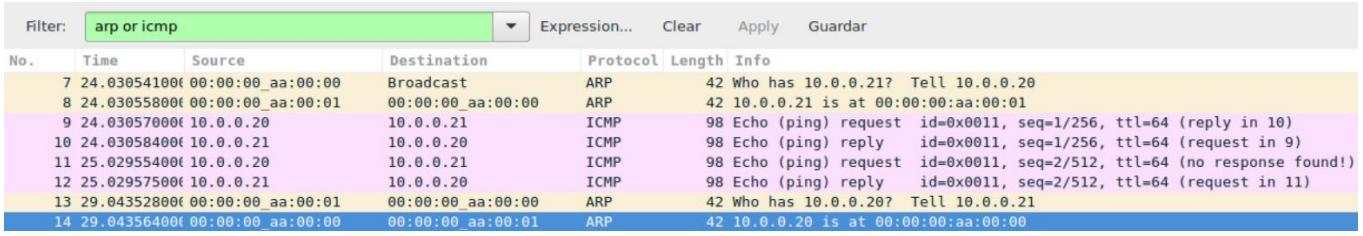
```
root@PC1sw:/tmp/pycore.36767/PC1sw.conf# arp -s 10.0.1.21 aa:aa:aa:aa:aa:aa
root@PC1sw:/tmp/pycore.36767/PC1sw.conf# arp -an
? (10.0.1.21) at aa:aa:aa:aa:aa:aa [ether] PERM on eth0
root@PC1sw:/tmp/pycore.36767/PC1sw.conf# ping 10.0.1.21
PING 10.0.1.21 (10.0.1.21) 56(84) bytes of data.
^C
--- 10.0.1.21 ping statistics ---
60 packets transmitted, 0 received, 100% packet loss, time 59192ms
```

La PC1sw va a hacer un ping a 10.0.1.21, no necesita obtener la dirección MAC ya que la “conoce” porque se encuentra en la tabla ARP. Al realizar el ping lo realizara con ip destino 10.0.1.21 y con MAC destino aa:aa:aa:aa:aa:aa la cual no existe en la topología, esto va a hacer que se envíen mensajes ECHO constantemente pero sin recibir respuestas ya que el switch no conoce la mac y realizará floading y aun así ninguna red conectada a este tendrá la dirección MAC destino, por ende nadie le responderá.

c. Analizar y comparar el funcionamiento de un HUB y de un SWITCH.

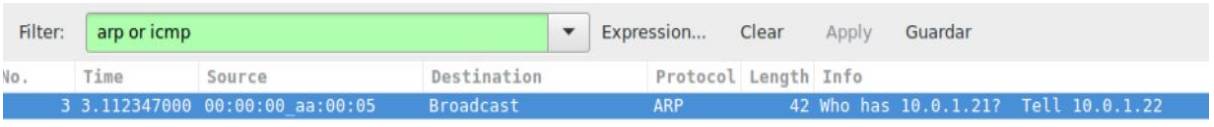
i. Antes de empezar asegúrese que todas las tablas estén vacías. Puede hacerlo deteniendo e iniciando la topología nuevamente.

ii. Inicie Wireshark en PC3_HUB y luego envíe un ping desde la PC1_HUB a PC2_HUB. Analice el origen y destino de cada uno de los paquetes ARP e ICMP capturados. ¿Alguno se origina en o va destinado a PC3_HUB? ¿Por qué observa cada uno de esos paquetes?



El único que está destinado al PC3_HUB es el ARP de la PC1_HUB preguntando a todos (broadcast) ¿Quién es 10.0.0.21?. Pero también recibe los demás mensajes que no están dirigidos a él. Esto sucede por el funcionamiento del HUB que lo único que hace es recibir datos y enviarlos a todas los demás dispositivos que estén conectados, unificando los dominios de colisión.

iii. Inicie Wireshark en PC3_SW y luego envíe un ping desde la PC1_SW a PC2_SW. Analice el origen y destino de cada uno de los paquetes ARP e ICMP capturados. ¿Alguno se origina en o va destinado a PC3_SW? ¿Por qué observa cada uno de esos paquetes?



El único que está destinado al PC3_SW es el ARP de la PC1_SW preguntando a todos (broadcast) ¿Quien es 10.0.1.21?. Se observa esto solo ya que con el SWITCH solo va a recibir los mensajes que estén destinados a él ya sea de forma directa o por broadcast.

iv. ¿Qué diferencia observa entre los dos casos anteriores? Explique por qué ocurre así.

La diferencia es cómo se envían las tramas en cada caso, en un caso se estarían enviando a todos los dispositivos en la red con un único destino(HUB) y en el otro caso se envía solamente a la dirección destino(SWITCH).

v. Indique cómo queda la tabla CAM del SWITCH una vez realizado el ping. ¿Cómo se arma y en qué orden?

PC1_SW al realizar el ping, como no conoce la dirección MAC de la PC2_SW envía un ARP a la dirección Broadcast para saber la MAC de la PC2_SW. Este mensaje pasaría por el SWITCH el cual guarda la MAC de origen(00:00:00:aa:00:05) y envía a todos los dispositivos el mensaje ARP.

Tabla CAM:	
00:00:00:aa:00:05	e1

Cuando recibe el mensaje de broadcast la PC2_SW le responde a la PC1_SW (de forma directa ya que tiene la MAC de este por el mensaje ARP) con su dirección MAC, este mensaje pasa a través del SWITCH el cual almacena el valor de la dirección MAC de esta interfaz ya que no la tenía. Y el SW lo envía directo a la PC1_SW ya que tiene su dirección MAC en la tabla CAM.

Tabla CAM:	
00:00:00:aa:00:05	e1
00:00:00:aa:00:06	e2

Tabla Cam Final ↑