Práctica 1.1 Diseño y gestión de escenarios IPv4. Configuración básica. Diseño y administración de redes

Héctor Lacueva Sacristán

Fecha de 9/25

Índice

Cuestiones previas

Cuestión 1: Una vez configurado el sistema, que resultados se obtienen al ejecutar ping desde PCA1 Y PCA2 hacia PCB1

Desde PCA1 (centos)

En este caso el comando a ejecutar es `ping -r` y el resultado esperado es el siguiente:

1. Dirección IP de PCA3 (LAN C, ej. 192.168.7.x)
2. Dirección IP de PCB3 (LAN B, ej. 192.168.20.1)
3. Dirección IP de PCB1 (192.168.20.x)

Obtienes las direcciones IP de las interfaces de salida por los que va pasando la comunicación.

Desde PCA2 (tinycore)

En este caso el comando a ejecutar es `traceroute` y el resultado esperado es el siguiente:

1. Dirección IP de PCA3 (LAN A, ej. 192.168.10.1)
2. Dirección IP de PCB3 (LAN C, ej. 192.168.7.x)
3. Dirección IP de PCB1 (192.168.20.x)

Obtienes las direcciones IP por las que pasa, pero siempre mirando hacia el origen.

Cuestión 5: Indicar los distintos casos de entrega directa e indirecta observados especificando su relación con el tráfico ARP capturado. Pon un ejemplo de un paquete IP encaminado mediante entrega directa especificando la IP destino y la MAC destino. Pon un ejemplo de un paquete IP encaminado mediante entrega indirecta especificando la IP destino y MAC destino.

Si tenemos en cuenta un `ping` de PCA1 a PCB1, los distintos casos de entrega directa en indirecta que habrá serán:

- **Directa** (cuando te comunicas en tu misma red):
  - PCA1 → PCA3 (red A).
  - PCA3 → PCB3 (red C).
  - PCB3 → PCB1 (red B).
- **Indirecta** (cuando vas a otra red):
  - PCA1 → PCB3 (red A → red C).
  - PCA1 → PCB1 (red A → red B).
  - PCA3 → PCB1 (red C → red B).

El tráfico ARP capturado debe coincidir con las comunicaciones directas que se han mencionado, y solo se puede detectar desde dentro de las propias redes.

Ejemplo de paquete IP encaminado mediante entrega directa:

- Paquete que va de PCA1 a PCA3 (en la red A) por un `ping`:
  - **IP destino**: IP de PCA3 en la red A.
  - **MAC destino**: dirección MAC de PCA3 en la LAN A.

Ejemplo de paquete IP encaminado mediante entrega indirecta:

- Paquete que va de PCA1 a PCB3 (de red A a red C) por un `ping`:
  - **IP destino**: IP de PCB3 en la red C.
  - **MAC destino**: dirección MAC de PCA3 en LAN A.

Cuestión 6: ¿A qué se deben las preguntas ARP durante la transmisión del primer ICMP Echo Request?

`Echo Request` se utiliza cuando no se conocen las direcciones MAC de sus siguientes saltos. Por eso, antes de poder enviar el paquete IP, cada dispositivo debe resolver mediante ARP la dirección MAC asociada a la IP del próximo salto.

Cuestión 7: ¿Por qué no hay preguntas ARP durante la transmisión del primer ICMP Echo Reply?

Porque cuando haces un `Echo Reply`, antes has recibido un `Echo Request` que contenía la dirección MAC a la que tienes que devolver el resultado por lo que no necesitas volver a preguntar la MAC de destino.

Cuestión 8: ¿Qué ocurre con los siguientes paquetes ICMP?

Los siguientes paquetes ICMP se envían directamente usando las direcciones MAC almacenadas en las cachés ARP de cada dispositivo. No habrá más tráfico ARP relacionado, salvo que las entradas caduquen o cambie la topología.

Cuestión 9: Sin dejar de capturar, parar el anterior ping e inmediatamente después iniciar un ping desde un host LAN C → PCA1. ¿Aparecen mensajes ARP en LAN interna y en LAN externa?¿Por qué?

El `ping` anterior era de PCA1 a PCB1, si se hiciera ahora un ping desde un host en LAN C a PCA1, aparecería un mensaje en la LAN interna C, ya que PCC1 (host de la LAN C) no conoce la dirección MAC de PCA3. Una vez obtenida la respuesta no debería de haber ningún ARP más porque PCA3 conoce la MAC de PCA1 (siempre y cuando no caduque ni cambie la topología).

Cuestión 10: Indicar el número y tamaño de los paquetes capturados en las redes LAN A, LAN C, LAN B. Para ello, rellena la tabla adjunta y justifica teóricamente los tamaños indicados.

Primero repaso los tamaños básicos (cálculo digit-a-digit):

- ICMP payload = 1400 bytes.
- ICMP header = 8 bytes → **ICMP total = 1400 + 8 = 1408 bytes**.
- IP header = 20 bytes → **IP total original = 20 + 1408 = 1428 bytes**.

Ahora paso por paso con MTU de cada enlace (según tu escenario actualizado):

- **LAN A PCA3 (eth0)**: MTU = **800** → disponible para datos por fragmento = `800 - 20 = 780` bytes. Pero el *Fragment Offset* obliga a que todos los fragmentos **excepto el último** tengan datos en múltiplos de 8 bytes. 780 mod 8 = 4, por tanto el máximo útil múltiplo de 8 ≤ 780 es **776** bytes (porque 97×8 = 776).
- **PCA3 (eth1) → LAN C**: MTU = **1100** → disponible para datos = `1100 - 20 = 1080` (ya múltiplo de 8).
- **PCB3 (eth0) → LAN B**: MTU = **700** → disponible para datos = `700 - 20 = 680` (múltiplo de 8).

Fragmentación en LAN A (MTU 800)

Tenemos 1408 bytes de datos a fragmentar:

1. Primer fragmento en LAN A: máximo usable = **776** bytes de datos.

- Datos en F1 = 776.
- IP total F1 = 776 + 20 = **796 bytes**.
- Offset (unidades 8 bytes) = 0.
- MF = 1 (porque quedan más datos).

2. Resta de datos: 1408 - 776 = **632** bytes.

- Como es el último pedazo, puede tener cualquier tamaño ≤ 776.
- Datos en F2 = 632.
- IP total F2 = 632 + 20 = **652 bytes**.
- Offset (unidades 8) = 776 / 8 = **97** → en bytes: 97 x 8 = **776 bytes**.
- MF = 0 (último fragmento del paquete original).

Por tanto, **en LAN A** se capturan **2 paquetes IP**:

- 796 bytes (offset 0, MF=1).
- 652 bytes (offset 776 bytes / unidad 97, MF=0).

PCA3 reenvía hacia eth1 con MTU 1100

PCA3 recibe los dos fragmentos (796 y 652). Al enviar por `eth1` con MTU 1100, ambos fragmentos **caben tal cual** (1100 > 796 y 1100 > 652), por lo que **no es necesaria reensamblación** ni nueva fragmentación. RFC permite reenviar fragmentos sin reensamblar. Entonces en **LAN C** verás exactamente los **mismos dos fragmentos** que en LAN A:

- 796 bytes (offset 0, MF=1).
- 652 bytes (offset 776 bytes / unidad 97, MF=0).

(Nota: estos tamaños son <1100, por eso pasan sin cambio.)

Fragmentación en PCB3 (hacia LAN B con MTU 700)

PCB3 recibe los dos fragmentos desde LAN C y debe enviarlos por su `eth0` con MTU 700 → disponible para datos = **680** bytes.

Analicemos cada fragmento recibido:

Re-fragmentación del fragmento grande (796 bytes)

- Datos en ese fragmento = 796 - 20 = **776 bytes** (coincide con el 1er fragmento original).
- Con MTU 700, cada subfragmento puede llevar hasta **680 bytes** de datos (múltiplo de 8).
- Dividimos 776 en: 680 + 96.
  - Subfrag 1a: datos 680 → IP total = 680 + 20 = **700 bytes**, offset = 0, MF = 1 (aún vienen más).
  - Subfrag 1b: datos 96 → IP total = 96 + 20 = **116 bytes**, offset = 680/8 = **85** (en bytes = 680), MF = 1 (porque aún queda el segundo fragmento original que sigue).

El segundo fragmento (652 bytes)

- Datos = 652 - 20 = **632 bytes**.
- 632 ≤ 680 → cabe entero; por tanto se envía como un solo fragmento: IP total = **652 bytes**, offset = sigue siendo **97** (unidad) → bytes 776.
- MF = 0 (es el último fragmento del paquete original).

Por tanto, **en LAN B** verás **3 paquetes** (resultado final tras re-fragmentación):

1. **700 bytes** (subfragmento del primero), offset 0, MF = 1.
2. **116 bytes** (resto del primer fragmento), offset 85 (85×8 = 680 bytes), MF = 1.
3. **652 bytes** (el fragmento original pequeño, sin cambiar), offset 97 (97×8 = 776 bytes), MF = 0.

(Orden físico en la captura puede variar; offsets en bytes: 0, 680, 776.)

Tabla (usando ID = 1)

Red (captura)	Nº paquete	Tamaño (bytes) IP total	ID	DF	MF	Offset (unidades=8)	Offset (bytes, Wireshark)
LAN A	1.a	**796** (20 + 776)	1	0	1	0	-
	1.b	**652** (20 + 632)	1	0	0	**97**	-
LAN C	1.a	**796** (igual que LAN A)	1	0	1	0	-
	1.b	**652** (igual que LAN A)	1	0	0	**97**	-
LAN B	2.a (subfrag)	**700** (20 + 680)	1	0	1	0	-
	2.b (subfrag)	**116** (20 + 96)	1	0	1	**85**	-
	2.c (desde 1.b)	**652** (20 + 632)	1	0	0	**97**	-