

## Práctica 11 – Rutas asimétricas. Fragmentación y reensamblado

### 1- Objetivos

En esta práctica se modificará el tamaño de la MTU de los interfaces WAN serie de los routers Cisco. Primero se analizará una configuración simple de los mismos y a continuación se creará una topología de red en anillo empleando los interfaces de los routers Cisco disponibles.

### 2- Material

- 3 routers Cisco con al menos 1 interfaz Ethernet y 2 interfaces WAN serie
- 2 PCs
- Conmutador(es) Ethernet

### 3- Fragmentación en origen



Figura 1.- Topología en serie

- Conecten uno de los puertos serie de router1 con uno de router2
- Entren en modo de configuración de cada interfaz serie que hayan conectado y asignen una dirección IP y una máscara de red a ese interfaz (empleen a su gusto la red asignada a su armario: 10.3.armario.0/24)
- Recuerden especificar, en el interfaz serie adecuado, la velocidad a la que se empleará la línea serie (comando `clock rate`). Y por supuesto activen los interfaces (`no shutdown`)

Con esto deberían poder hacer ping desde un router al otro a través de la línea serie. La línea serie estará empleando como nivel de enlace el encapsulado HDLC. Cambie el tipo de encapsulamiento a PPP escribiendo `encapsulation ppp` en el modo de configuración de las interfaces serie en ambos routers. Utilice `show interface serial<numero>`, válido tanto desde el modo usuario como desde el modo administrador, para verificar el cambio de encapsulado. ¿Qué tamaño tiene la MTU de estos interfaces serie? ¿Entre qué rango de valores, en bytes, puede configurarse esta MTU? Para averiguar el comando (**lo necesitará para realizar el siguiente apartado**) con el que configurar dicha MTU, entre en el modo de configuración del interfaz serie y consulte la ayuda del IOS de Cisco.

- ¿Cuál es el tamaño de la MTU de una interfaz Ethernet? Puede comprobarlo en cualquiera de sus PCs mediante el comando `ifconfig`.

A continuación conecten uno de los interfaces Ethernet de cada router a un conmutador diferente del switch0 (consulte la documentación de los armarios):

- Conecten un PC a cada una de esas LANs
- Configuren 3 redes: una para cada LAN y una tercera para el enlace serie, empleando el espacio de direcciones que tienen asignado
- Configuren en cada router una ruta por defecto al otro router por el interfaz serie
- Configuren en cada PC la dirección IP y el router por defecto
- Ponga Wireshark a capturar tramas Ethernet en PCA y en PCC
- Prueben a hacer ping entre los PCs
- Detengan Wireshark y utilicen sus capturas para:
  1. Averiguar el tamaño (en bytes) de datos que envía, por defecto, el comando ping.
  2. ¿Cuál es el tamaño de cada trama capturada? ¿A qué se debe esta diferencia de bytes entre los datos enviados por ping y el tamaño de la trama ethernet?
  3. Realice un esquema de una de las tramas Ethernet capturadas en el que se aprecie el desencapsulado en cada uno de los niveles de la pila de protocolos TCP/IP, indicando el tamaño de los encabezados correspondientes a cada nivel de la pila.
- Consulte el manual de ping (man ping) para hacer que ping envíe un solo paquete de 2945bytes de datos. Analice mediante la captura correspondiente de Wireshark las tramas resultantes:
  1. ¿Cuántas tramas son necesarias para que ping envíe 2945bytes de datos? ¿Por qué?
  2. ¿En cada una de estas tramas, intervienen los mismos protocolos? ¿Por qué?
  3. Para las tramas capturadas, correspondientes al ping de 2945bytes, complete los siguientes valores:

**Fragmento1:** Long. total = ? bytes |Desplazamiento = ? |MF= ? (contiene los primeros ?bytes de datos del datagrama original)

**Fragmento2:** Long. total = ? bytes |Desplazamiento = ? |MF= ? (contiene los siguientes ?bytes de datos del datagrama original)

**Fragmento3:** Long. total = ? bytes |Desplazamiento = ? |MF= ? (contiene los últimos ?bytes de datos del datagrama original)

MF: More Fragments

Utilice ping para conseguir una captura de Wireshark con sólo dos tramas Ethernet (cuatro en realidad; dos de ida y dos de vuelta). La segunda trama, sólo tendrá 1 byte ocupado en el campo de datos ICMP.

#### 4- Fragmentación en tránsito (ruta simétrica)

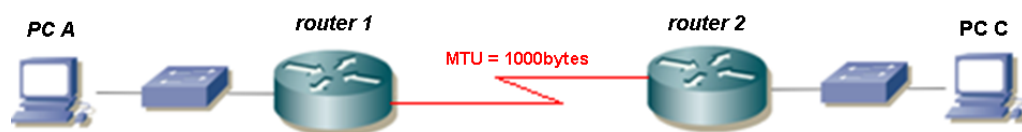


Figura 2.- Topología en serie. MTU modificada

Un encaminador (router) fragmenta un datagrama (IPv4) en varios si el enlace del siguiente salto por el que tiene que viajar el datagrama (IPv4) tiene una MTU inferior a la longitud del datagrama (IPv4).

Por defecto su distribución de linux tiene activado el PMTUD (Path MTU Discovery) lo que implica que todo datagrama (IPv4) que salga de su PC tendrá el bit de No Fragmentar (“Don’t fragment”) a 1. Si no lo desactivásemos, el datagrama (IPv4) no conseguiría atravesar el router1 y, por tanto, no tendría forma de llegar hasta el PC C (al menos en un primer intento), ya que el router1 descartaría el datagrama, pero indicaría a PC A, mediante un mensaje ICMP (tipo 3; “destino inalcanzable”, código de error 4; “se necesita fragmentación”), el tamaño de la MTU del enlace del siguiente salto. De esta forma, PC A, en la siguiente comunicación, fragmentaría (si fuera necesario) sus datagramas según dicha MTU. Como no es objeto de esta práctica el análisis del PMTUD, lo desactivamos ejecutando el siguiente script, tanto en PCA como en PCC: `sudo pmtud no`

*Nota:* Si quieren saber más sobre PMTUD y su motivación, pueden la RFC1191

Modifique la MTU de los interfaces serie asociados al enlace serie que conecta router1 con router2 de manera que el mayor datagrama que pueda atravesar dicho enlace, sin ser fragmentado, sea como máximo de 1000bytes.

Punto de control: Analicen las tramas capturadas mediante Wireshark, tanto en PCA como en PCC para un solo ping de 2549bytes de datos desde PCA hacia PCC. Justifique la fragmentación que se produce en los datagramas de ida y vuelta.

## 5- Fragmentación en tránsito (ruta asimétrica)

Ahora intentaremos crear la topología de la figura 3.

### Destino a un salto

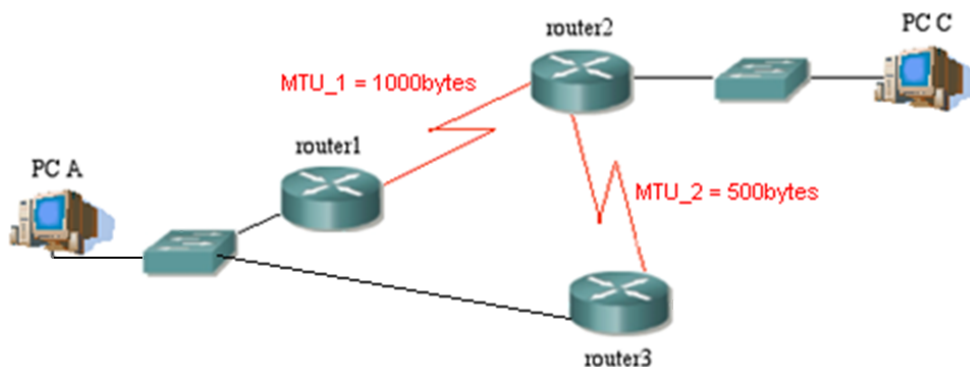


Figura 3.- Topología asimétrica.

- En su espacio de direcciones (10.3.armario.0/24) creen subredes para la topología de la figura 3, escogiendo una única máscara de longitud fija (lo que sería subnetting clásico si esa red estuviera en el rango de redes C).
- Configuren los enlaces serie entre routers como en la figura 3. Modifique la MTU (MTU\_2) de los interfaces serie asociados al enlace serie que conecta router2 con router3 de manera que el mayor datagrama que pueda atravesar dicho enlace, sin ser fragmentado, sea como máximo de 500bytes. La MTU\_1 ya la tendrá configurada del apartado anterior.

- Configuren un interfaz Ethernet de cada router para conectarse a una LAN en la que conectarán un PC (empleen switch0)
- Cada PC tendrá como router por defecto al router conectado en su LAN. El de PC A será router 1.
- En cada router configuren una ruta por defecto de forma que los paquetes circulen en el sentido router1 -> router2 -> router3
- Prueben a hacer ping entre los PCs

Analice las tramas capturadas mediante Wireshark, tanto en PCA como en PCC para un solo ping de 2549bytes de datos desde PCA hacia PCC. Justifique la fragmentación que se produce en los datagramas de ida y vuelta.

### Destino a dos saltos

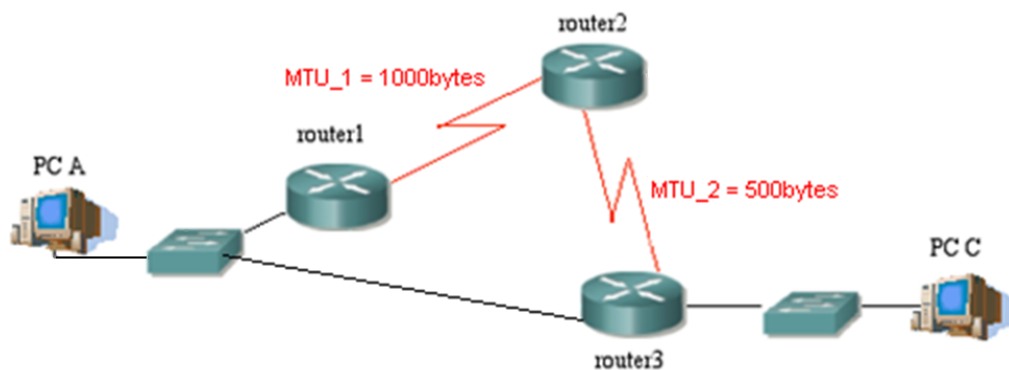
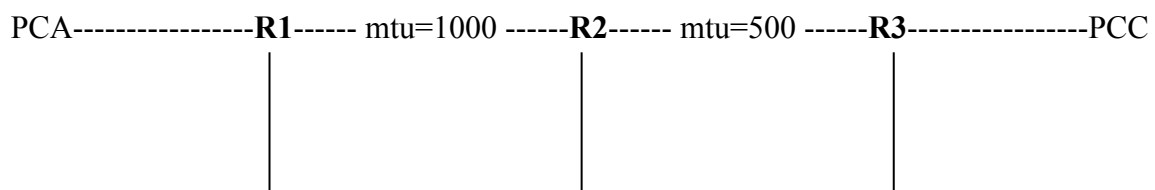


Figura 4.- Topología asimétrica II

- Modifique la localización de PC C, manteniendo el enrutamiento en el anillo
- Analice las tramas capturadas mediante Wireshark, tanto en PCA como en PCC para un solo ping de 2549bytes de datos desde PCA hacia PCC. Justifique la fragmentación que se produce en los datagramas de ida y vuelta.
- Debería ser capaz de responder a las siguientes cuestiones:
  - ¿Qué fragmentos en destino (PC C) se corresponden con el datagrama original (PC A)
  - ¿Dónde se produjo la fragmentación del datagrama original? ¿Por qué?

*Nota: Para analizar la fragmentación de los datagramas en cada enlace entre routers, le resultará útil apoyarse en un esquema de este tipo:*



## **Guardando la ruta**

Con el mismo escenario del apartado anterior, utilice la opción adecuada del comando ping (man ping) que le permita guardar la ruta que siguen los paquetes.

Punto de control: Analicen las tramas capturadas mediante Wireshark, tanto en PCA como en PCC para un solo ping de 2549bytes de datos desde PCA hacia PCC. Justifique la fragmentación que se produce en los datagramas de ida y vuelta.