Redes y Comunicaciones

Práctica 4 - Capa de red

Es la práctica más importante ;)
Puntos que faltan: 26 (no había ganas de terminarlo este)

1. ¿Qué servicios presta la capa de red? ¿Cuál es la PDU en esta capa?

La función de la capa de red es transportar paquetes de un host emisor a un host receptor. En la realización de esta tarea podemos identificar dos importantes funciones de esta capa:

- Reenvío (forwarding): cuando un paquete llega al enlace de entrada de un router este tiene que pasar el paquete al enlace de salida apropiado. Esto es dentro del mismo router.
- Enrutamiento (routing): la capa de red tiene que determinar el camino que deben seguir los paquetes a medida que fluyen de un emisor a un receptor. Los algoritmos que calculan estas rutas se llaman **algoritmos de enrutamiento**.

La PDU de esta capa se llama **datagrama, paquete o célula en ATM**.

2. Compare los siguientes modelos de servicios de red:

	¿Todos los paquetes siguen el mismo camino?	**¿Cuenta con una fase de establecimiento y otra de cierre de circuito?**	**¿Usa mensajes de señalización?**	**¿Usa tablas de enrutamiento?**
Datagrama	No	No	SI	Si
Circuitos virtuales	Si	Si	Si -para establecimiento / mantención / cierre de conexión-	Si

3. ¿Qué dispositivo es considerado sólo de esta capa? Explique las dos funciones principales que

debe realizar.

El dispositivo de capa de Red es el router. Acorde a los dos servicios de la capa de Red, este dispositivo debe:

- Ejecutar algoritmos/protocolos de enrutamiento que seleccionen hacia dónde reenviar un datagrama recibido.
- Encaminar/conmutar los datagramas que llegan a una interfaz o puerto de entrada, a la interfaz o puerto de salida seleccionada por el algoritmo. La conmutación puede hacerse vía memoria -control directo de una CPU-, vía bus compartido en el router o vía crossbar o red de interconexión.

El puerto de salida puede realizar buffering si su tasa de transmisión es inferior a la de llegada de datos desde el entramado de conmutación (produciendo un retraso), si el buffer se llena, pueden perderse paquetes. Del mismo modo si la tasa de llegada de datagramas al puerto de entrada es superior a la velocidad de conmutación, el puerto de entrada utiliza buffering.

4. En las redes IP el ruteo puede hacerse en forma estática como dinámica. Describa conceptualmente como funciona cada uno de ellos e indique ventajas y desventajas de cada método.

Kurose-Ross:

Ruteo estático o dinámico: es una clasificación de los algoritmos de ruteo. En el ruteo **estático**, las rutas cambian muy lentamente a lo largo del tiempo, normalmente como resultado de una intervención humana (porque la persona edita manualmente las tablas de encaminamiento). Los algoritmos de ruteo **dinámico** cambian los caminos de ruteo según cambia la carga del tráfico o la topología de la red. Un algoritmo dinámico podrá ejecutarse bien periódicamente, o bien en respuesta directa a un cambio en la topología o en los costes de los enlaces. Aunque los algoritmos dinámicos responden mejor a los cambios de red, también son más susceptibles a problemas como los bucles de enrutamiento y a la oscilación de rutas.

Diapositivas Clase 8:

Ruteo Estático (Las rutas son establecidas por el administrador manualmente)

PRO:

- Sirve cuando se tiene una red sencilla.
- No tiene problemas de seguridad ni de incompatibilidad.
- No implica costo de procesamiento extra.

CONTR:

- Propenso a errores (error humano)
- Si se cambia la topología requiere cambios manuales en los routers.
- Esquema NO escalable y NO tolerante a fallos.

Ruteo Dinámico

PRO:

- Si se cambia la topología se adapta de forma automática.
- Facilita cuando se tiene una red compleja.
- Esquema escalable y tolerante a fallos.

CONTR:

- Requiere una configuración inicial por el administrador.
- Implica costo de procesamiento extra.

5. Los algoritmos de ruteo dinámico se dividen en estado enlace y vector distancia. Dado el siguiente cuadro compare:

	¿Cada router conoce la topología completa?	**¿Converge rápidamente?**	**Protocolos que lo implementan**
Vector distancia	No	No	RIP, IGRP, EIGRP
Estado de enlace	Si	Si	OSPF, IS-IS

6. ¿Qué son los sistemas autónomos? ¿Qué es necesario para que los distintos sistemas autónomos puedan rutear el tráfico de hosts pertenecientes a diferentes sistemas autónomos?

Un Sistema Autónomo (en inglés, Autonomous System: AS) es un conjunto de redes y dispositivos router IP que se encuentran administrados por una sola entidad (o en algunas ocasiones varias) que cuentan con una política común de definición de trayectorias para Internet. Un ejemplo de un AS es la red de la UNLP que está toda bajo una misma administración.

Los Sistemas Autónomos se comunican entre sí mediante routers BGP y se intercambian el tráfico de Internet que va de una red a la otra. A su vez cada Sistema Autónomo es como una Internet en pequeño, ya que su rol se llevaba a cabo por una sola entidad, típicamente un Proveedor de Servicio de Internet (ISP) o una gran organización con conexiones independientes a múltiples redes, las cuales se apegaban a una sola y clara política de definición de trayectorias definida.

Técnicamente un Sistema Autónomo se define como un grupo de redes IP que poseen una política de rutas propias e independientes. Esta definición hace referencia a la característica fundamental de un Sistema Autónomo: realiza su propia gestión del tráfico que fluye entre él y los restantes Sistemas Autónomos que forman Internet. Aún considerando que el ISP podía soportar múltiples sistemas

autónomos, Internet solo considera la política de definición de trayectorias establecida por el ISP. Por lo tanto, el ISP debería contar con un ASN (AS Number) registrado. Un número de AS o ASN se asigna a cada AS para ser utilizado por el esquema de encaminamiento BGP, este número identifica de manera única a cada red dentro del Internet.

Los protocolos de ruteo que utilizan **internamente** se denominan **IGP** (Interior Gateway Protocol), y pueden ser: RIP, IGRP, EIGRP, OSPF, IS-IS. Los SA permiten el ruteo jerárquico.

Cada SA se conecta a un **router de borde o gateway**, que lo conecta a otros. Se denominan **EGP** (Exterior Gateway Protocols) a los protocolos entre distintos AS (GGP, EGP, BGP).

El ruteo jerárquico permite salvar los problemas que supondría utilizar los mismos protocolos entre diferentes subredes:

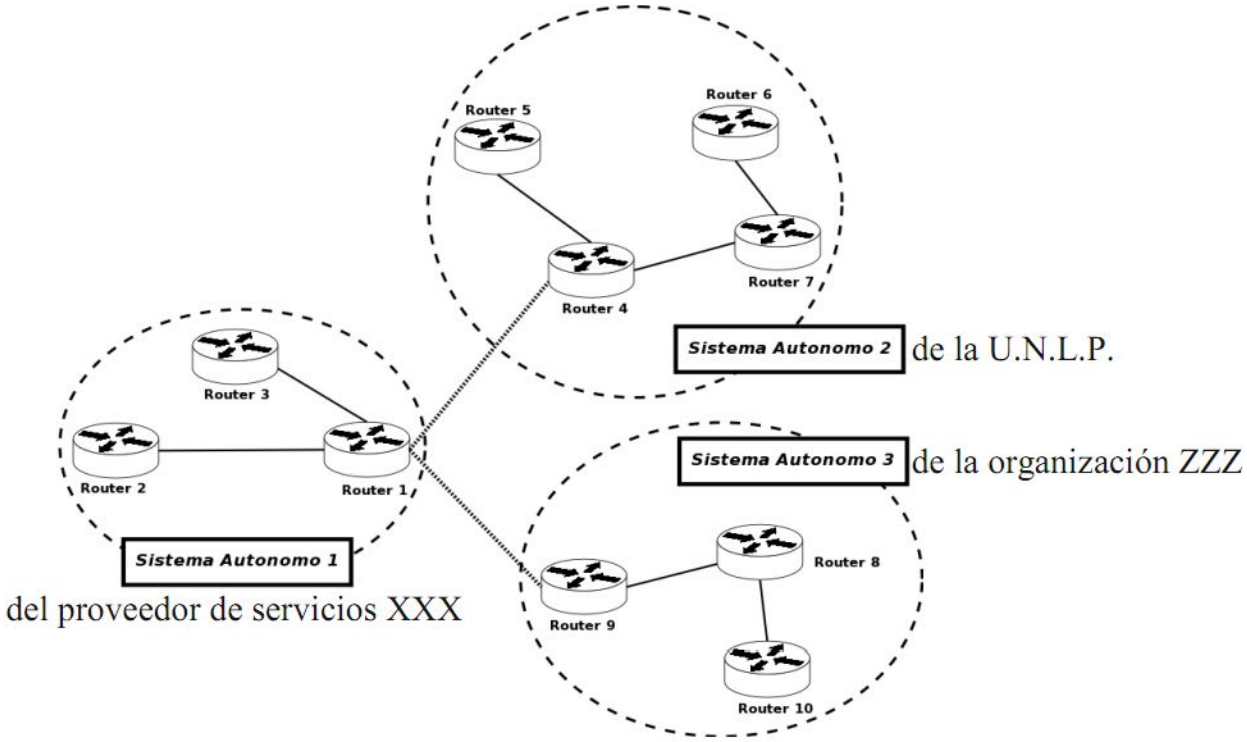
- Escala: El enorme tamaño que alcanzarían las tablas de ruteo, el overhead de intercambiar información entre los routers para una red de grandes dimensiones y el enorme tiempo de convergencia que se tendría.
- Autonomía administrativa: La imposibilidad de elegir un protocolo específico por parte de los administradores de las redes.

Cortito: un AS es un conjunto de redes y routers administrados por una sola entidad. Para comunicar un AS con el exterior se necesita un router de borde y algun protocolo EGP

7. Los algoritmos de ruteo también se pueden clasificar como IGP y EGP. Dado el siguiente cuadro complete:

	¿Implementaciones más conocidas?	**¿Donde se usan?**	**¿Pueden ser sustituidos por configuración manual (Si, Depende, No)? ¿Por qué?**
IGP	RIP, IGRP, EIGRP, OSPF, IS-IS...	Se utilizan dentro de los ASs	Si, con rutas estaticas.
EGP	GGP, EGP, BGP	Entre distintos AS (no dentro)	No.

8. A partir del siguiente gráfico indique:



¿Qué tipo de algoritmo se utiliza para compartir información entre los routers 4 y 7?

IGP, porque están dentro del mismo AS.

¿Qué tipo de algoritmo se utiliza para compartir información entre los routers 1 y 4?

EGP porque están en AS distintos.

¿Qué tipo de algoritmos alimentan las tablas de ruteo de los routers 3 y 10?

IGP

¿Qué tipo de algoritmos alimentan las tablas de ruteo de los routers 1 y 4?

IGP para los hosts internos y EGP para comunicación fuera del AS.

9. ¿Qué es una red clase A? ¿Qué es una red clase B?

¿Qué es una red clase C? ¿Cuántas hay de cada una? ¿Cuántos hosts pueden haber en cada una?

Las direcciones de clase A

La clase A comprende redes desde `1.0.0.0` hasta `127.0.0.0`. El número de red está en el primer octeto, con lo que sólo hay 127 redes de este tipo, pero cada una tiene 24 bits disponibles para identificar a los nodos, lo que se corresponde con poder distinguir en la red unos 1.6 millones de nodos distintos.

Corresponden a redes que pueden direccionar hasta 16.777.214 máquinas cada una.
Las direcciones de red de clase A tienen siempre el primer bit a 0.
0 + Red (7 bits) + Máquina (24 bits)
Solo existen 124 direcciones de red de clase A.

Ejemplo:

	Red	Máquina		
Binario	00001010	00001111	00010000	00001011
Decimal	10	15	16	11

Rangos(notación decimal):
10.xxx.xxx.xxx - 126.xxx.xxx.xxx

Las direcciones de clase B

La clase B comprende redes desde `128.0.0.0` hasta `191.255.0.0`; siendo el número de red de 16 bits (los dos primeros octetos). Esto permite 16320 redes de 65024 nodos cada una.
Las direcciones de red de clase B permiten direccionar 65.534 máquinas cada una.
Los dos primeros bits de una dirección de red de clase B son siempre 10.
10 + Red (14 bits) + Máquina (16 bits)
Existen 16.382 direcciones de red de clase B.
Ejemplo:

	Red		Máquina	
Binario	10 000001	00001010	00000010	00000011
Decimal	129	10	2	3

Rangos(notación decimal):
128.000.xxx.xxx - 191.255.xxx.xxx

Las direcciones de clase C

Las redes de clase C tienen el rango de direcciones desde `192.0.0.0` hasta `223.255.255.0`, contando

con tres octetos para identificar la red. Por lo tanto, hay cerca de 2 millones de redes de este tipo con un máximo de 254 nodos cada una.
Las direcciones de clase C permiten direccionar 254 máquinas.
Las direcciones de clase C empiezan con los bits 110
110 + Red (21 bits) + Máquina (8 bits)
Existen 2.097.152 direcciones de red de clase C.
Ejemplo:

	Red			Máquina
Binario	110 01010	00001111	00010111	00001011
Decimal	202	15	23	11

Rangos(notación decimal):
192.000.000.xxx - 223.255.254..xxx

Las direcciones de clase D

Las direcciones de clase D son un grupo especial que se utiliza para dirigirse a grupos de máquinas. Estas direcciones son muy poco utilizadas. **Los cuatro primeros bits de una dirección de clase D son 1110.** Comprenden las direcciones entre 224.0.0.0 y 254.0.0.0, y están reservadas para uso futuro, o con fines experimentales. No especifican, pues, ninguna red de Internet.

10.¿Qué son la subredes? ¿Por qué es importante siempre especificar la máscara de subred?

La división de subredes es la obtención de otras direcciones de red basadas en una sola dirección con el uso de la máscara de subred, "pidiendo bits prestados" a la parte del host/interface (dependiendo de la cantidad de bits que se pidan, será la cantidad de subredes que se creen a partir de la original).

Una subred es un rango de direcciones lógicas. Cuando una red se vuelve muy grande, conviene dividirla en subredes, para reducir el tamaño de los dominios de broadcast, y hacer la red más manejable.

Típicamente los routers constituyen los límites entre las subredes. La comunicación desde/hasta otras subredes es hecha mediante un router específico. Sin embargo, las subredes permiten dividir lógicamente una red a pesar de su diseño físico, pudiéndose dividir en varias subredes configurando diferentes host que utilicen diferentes routers.

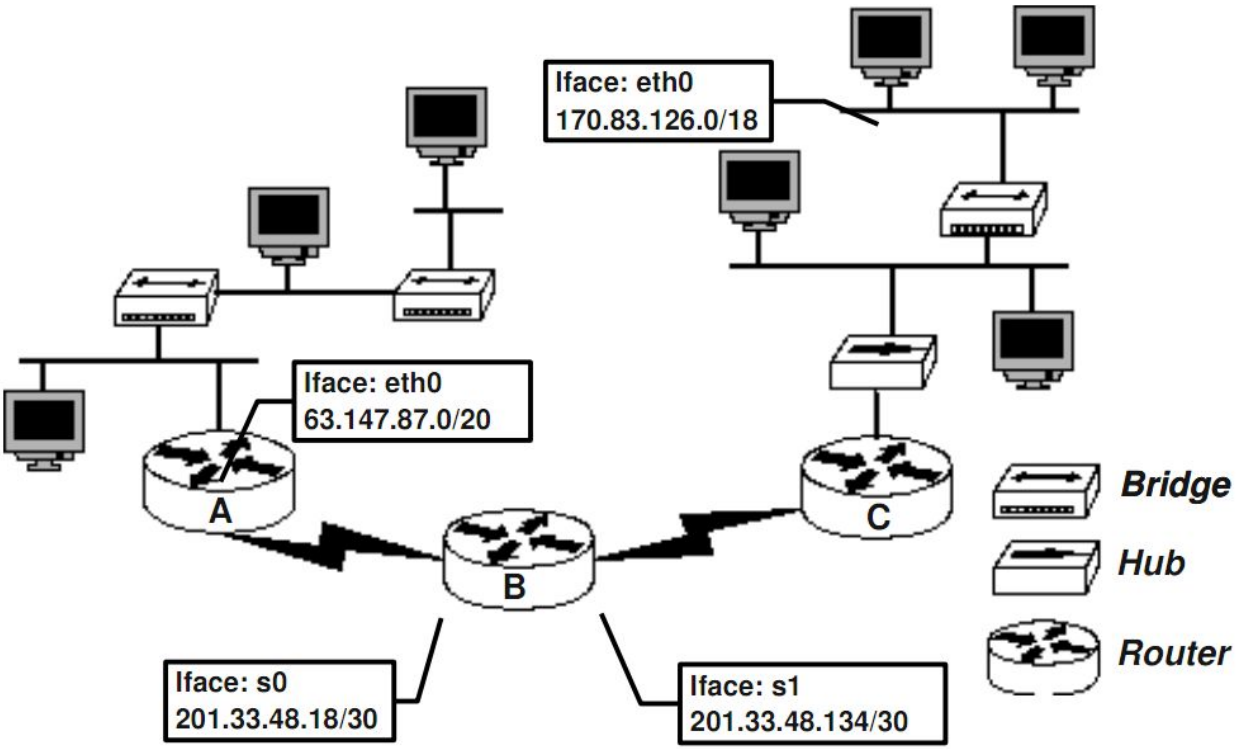
La dirección de todos los nodos en una subred comienzan con la misma secuencia binaria, que es su ID de red e ID de subred (en IPv4, las subredes deben ser identificadas por la base de la dirección y una máscara de subred).

Las subredes simplifican el enrutamiento, representándose típicamente cada una como una fila en las tablas de ruteo en cada router conectado. Fueron usadas antes de IPv4 para permitir a una red grande tener un número importante de redes más pequeñas dentro, controladas por varios routers.

Permiten el Enrutamiento Interdominio Sin Clases (CIDR). Los últimos dos bits del último octeto (los menos significativos) nunca se asignan a la subred, sea cual sea la clase de dirección IP. Como consecuencia, una dirección de subred jamás terminará en un número impar. Por otra parte, el uso de todos los bits disponibles para crear subredes dará como resultado subredes con sólo dos Hosts utilizables

(un método práctico de conservación de direcciones para el direccionamiento de enlace punto a punto, donde no existe otro direccionamiento más que los dos enlaces conectado entre sí).

11.Dado el siguiente gráfico complete:



1. Con los datos dados y para cada segmento de red indique:

http://informatica.iessansebastian.com/portal/index.php/calculadora-ip

- Dirección de Red y clase
- Dirección de Subred y máscara
- Dirección de broadcast
- Cantidad de direcciones utilizables en la subred dada

	170.83.126.0/18	63.147.87.0/20	201.33.48.18/30	201.33.48.134/30
Red	170.83.0.0	63.0.0.0	201.33.48.0	201.33.48.0
Clase	B	A	C	C
Subred	170.83.64.0	63.147.80.0	201.33.48.16	201.33.48.132
Máscara	255.255.192.0	255.255.240.0	255.255.255.252	255.255.255.252
Broadcast	170.83.127.255	63.147.95.255	201.33.48.19	201.33.48.135
# Hosts	2^{14} - 2	2^{12} - 2	2^{2} - 2	2^{2} - 2

2. Escoja una dirección IP adecuada para cada una de las interfaces de cada uno de los routers.

(por lo general para los routers se usa el primer o el último host)

Router A:

- eth0: 63.147.87.0 **/20**
- s0: 201.33.48.17 **/30**

Router B:

- s0: 201.33.48.18 **/30**
- s1: 201.33.48.134 **/30**

Router C:

- eth0: 170.83.64.1 **/18**
- s0: 201.33.48.133 **/30**

3. Defina una tabla de ruteo para cada router en el gráfico, de forma tal de que todos los dispositivos en la red puedan comunicarse.

(**red destino**: subredes a las que se puede llegar; **gateway**: router al cual tiene q mandar; **máscara**: la máscara de red destino; **interface**: ethN para redes directamente conectadas y sN para routers)

Router A

Red Destino	Gateway	Máscara de Red	Interface
63.147.80.0	0.0.0.0	20	eth0
201.33.48.16	0.0.0.0	30	s0
0.0.0.0	201.33.48.18	0	s0

Router B

Red Destino	Gateway	Máscara de Red	Interface
201.33.48.16	0.0.0.0	30	s0
201.33.48.132	0.0.0.0	30	s1
63.147.80.0	201.33.48.17	20	s0
170.83.64.0	201.33.48.133	18	s1

Router C

Red Destino	Gateway	Máscara de Red	Interface
170.83.64.0	0.0.0.0	18	eth0

201.33.48.132	0.0.0.0	30	s0
0.0.0.0	201.33.48.134	0	S0

angras-proyectos.dyndns.org

12.Para cada una de las siguientes direcciones IP (172.16.58.223/26, 163.10.5.49/27, 128.10.1.0/23, 10.1.0.0/24, 8.40.11.179/12) determine:

1. De qué tipo de dirección se trata (A, B o C).(**a** [1 - 127], **b** [128 - 191.255], **c** [192 - 233.255.255])

172.16.58.223/26 => **Clase B**
163.10.5.49/27 => **Clase B**
128.10.1.0/23 => **Clase B**
10.1.0.0/24 => **Clase A**
8.40.11.179/12 => **Clase A**

2. Cuál es la dirección de subred. (aplicar un AND con la máscara / poner en cero los bis fuera d la máscara)

172.16.58.223/26 => **172.16.58.192**

```
RED ORIGINAL:   10101100.00010000.00111010.11|011111
MÁSCARA /26:    11111111.11111111.11111111.11|000000
DIR SUBRED:     10101100.00010000.00111010.11|000000 = 172.16.58.192
```

163.10.5.49/27 => **163.10.5.32**

```
RED ORIGINAL:   XXX.XXX.XXX.001|10001
MÁSCARA /27:    255.255.255.111|00000
DIR SUBRED:     XXX.XXX.XXX.001|00000 = 163.10.5.32
```

128.10.1.0/23 => **128.10.0.0**

```
RED ORIGINAL:   XXX.XXX.0000000|1.00000000
MÁSCARA /23:    255.255.1111111|0.00000000
DIR SUBRED:     XXX.XXX.0000000|0.00000000 = 128.10.0.0
```

10.1.0.0/24 => **10.1.0.0**

```
RED ORIGINAL:   XXX.XXX.XXX.00000000
MÁSCARA /24:    255.255.255.00000000
DIR SUBRED:     XXX.XXX.XXX.00000000 = 10.1.0.0
```

8.40.11.179/12 => **8.32.0.0**

```
RED ORIGINAL:   XXX.0010|1000.Y.Y
MÁSCARA /12:    255.1111|0000.0.0
DIR SUBRED:     XXX.0010|0000.0.0 = 8.32.0.0
```

3. Cuál es la cantidad máxima de hosts que pueden estar en esa subred.

(# de bits libres de la máscara, menos primera dirección q es la dirección de red y la última que es de broadcast)

172.16.58.223/26 => cantidad maxima de host **(2^6)-2 = 62**

163.10.5.49/27 => cantidad maxima de host **(2^5)-2 = 30**

128.10.1.0/23 => cantidad maxima de host **(2^9)-2 = 510**

10.1.0.0/24 => cantidad maxima de host **(2^8)-2 = 254** *aca no se da

8.40.11.179/12 => cantidad maxima de host **(2^20)-2 = 1.048.574**

Forma de verificacion: GENERALMENTE los de clase A tienen mas host que los de clase B y sucesivamente con C.

* Pero estamos hablando de subredes, o sea, redes ya divididas, podría no darse eso

4. Cuál es la dirección de broadcast de la subred.

(última host posible / le haces un OR con el complemento de la máscara)

172.16.58.223/26 => **172.16.58.255**

```
RED ORIGINAL:   10101100.00010000.00111010.11|011111
COMPLEMENTO/26: 00000000.00000000.00000000.00|111111
DIR SUBRED:     10101100.00010000.00111010.11|111111 = 172.16.58.255
```

163.10.5.49/27 => **163.10.5.63**

```
RED ORIGINAL:   X.X.X.001|10001
COMPLEMENTO/27: 0.0.0.000|11111
DIR SUBRED:     X.X.X.001|11111 = 163.10.5.63
```

128.10.1.0/23 => **128.10.1.255**

```
RED ORIGINAL:   X.X.0000000|1.00000000
COMPLEMENTO/23: 0.0.0000000|1.11111111
DIR SUBRED:     x.x.0000000|1.11111111 = 128.10.1.255
```

10.1.0.0/24 => **10.1.0.255**

```
RED ORIGINAL:   X.X.X.00000000
COMPLEMENTO/24: 0.0.0.11111111
DIR SUBRED:     X.X.X.11111111 = 10.1.0.255
```

8.40.11.179/12 => **8.47.255.255**

```
RED ORIGINAL:   X.0010|1000.Y.Y
COMPLEMENTO/12: 0.0000|1111.1.1
DIR SUBRED:     X.0010|1111.1.1 = 8.47.255.255
```

5. Cuál es el rango de hosts válidos dentro de la subred.

(**primer host:** direcciónDeSubred + 1 ; **último host:** direcciónDeBroadcast - 1)

IP	Primer Host	Último Host
172.16.58.223/26	172.16.58.193	172.16.58.254
163.10.5.49/27	163.10.5.33	163.10.5.62
128.10.1.0/23	128.10.0.1	128.10.1.254
10.1.0.0/24	10.1.0.1	10.1.0.254
8.40.11.179/12	8.32.0.1	8.47.255.254

13.Dada la IP 65.0.0.0/8. Se necesitan definir 934 subredes. Indique que máscara debería ser utilizada. Indique cuál sería la subred número 817 indicando el rango de direcciones asignables, dirección de red y broadcast.

Para direccionar 934 subredes necesito 10 bits (2^{10}=1024). Los /8 bits de la máscara original más los 10 necesarios para las 934 subredes da un total de 18 bits. La **mascara** es **/18.**

La red nro 817 tiene 816 en binario que es igual a 1100110000. La subred 817 y todos sus hosts van a tener la forma **X.**11001100.00**YYYYYY.Y** donde las X son los bits reservados y las Y son los hosts.

Dirección	Dirección en binario	Dirección en decimal
Dirección de Red	**01000001.**11001100.00|**000000.00000000**	**65.204.0.0**
Broadcast	**01000001.**11001100.00|**111111.11111111**	**65.204.63.255**
Min Asignable	**01000001.**11001100.00|**000000.00000001**	**65.204.0.1**
Max Asignable	**01000001.**11001100.00|**111111.11111110**	**65.204.63.254**

14.Dada la red 195.200.45.0/24. Se necesitan definir 9 subredes. Indique la máscara utilizada y las nueve primeras subredes. Luego tome una de ellas e indique el rango de direcciones asignables en esa subred, dirección de red y broadcast.

Máscara original /24 + 4 bits para direccionar 9 subredes: Mascara = **/28**

Red original: X.X.X.00000000

Red con máscara: X.X.X.YYYY0000 (X = máscara original; Y = subredes)

```
Subred-1 = X.X.X.0000|0000 = 195.200.45.0
Subred-2 = X.X.X.0001|0000 = 195.200.45.16
Subred-3 = X.X.X.0010|0000 = 195.200.45.32
Subred-4 = X.X.X.0011|0000 = 195.200.45.48
Subred-5 = X.X.X.0100|0000 = 195.200.45.64
Subred-6 = X.X.X.0101|0000 = 195.200.45.80
Subred-7 = X.X.X.0110|0000 = 195.200.45.96
Subred-8 = X.X.X.0111|0000 = 195.200.45.112
Subred-9 = X.X.X.1000|0000 = 195.200.45.128
```

Subred-7
Subred = 195.200.45.96
Min Asignable = 195.200.45.97
Max Asignable = 195.200.45.110
Broadcast = 195.200.45.111

15.¿Que es CIDR (Classless Interdomain routing)? ¿Por qué resulta útil?

Classless Inter-Domain Routing (CIDR) es un método para reducir el tamaño de las tablas de ruteo agrupando direcciones. Si un router puede acceder de la dirección 193.168.0.0/24 a la 193.168.127.0/24, CIDR propone resumir las 128 direcciones a 193.168.0.0/**17**. Para que esto funcione se deben elegir las IPs de manera cuidadosa y ordenarlas de forma jerárquica y las direcciones deben compartir la misma cantidad de bits de mayor peso. Se denomina classless porque en estas direcciones la clase (A, B, C o D) no es de importancia y las direcciones siempre llevan máscara.

Resulta útil justamente porque permite achicar las tablas de ruteo considerablemente al resumir en una dirección classles varias direciones.

16. Dado un router “A” que tiene las siguientes entradas en su tabla de rutas.

Red Destino	**Gateway**	**Mascara**	**interface**
202.58.128.0	170.10.11.1	255.255.255.0	eth0
202.58.129.0	170.10.11.1	255.255.255.0	eth0
202.58.130.0	170.10.11.1	255.255.255.0	eth0
202.58.131.0	170.10.11.1	255.255.255.0	eth0

Realizar la máxima agregación CIDR posible del siguiente conjunto de 4 redes clase C para reducir la cantidad de entradas de la tabla de enrutamiento lo maximo posible (Nota: la agregación no debería incluir rutas que no fueron listadas)

Las direciones son:
202.58.128.0 = 11001010.00111000.10000000.00000000

202.58.129.0 = 11001010.00111000.10000**01**.00000000
202.58.130.0 = 11001010.00111000.10000**10**.00000000
202.58.131.0 = 11001010.00111000.10000**11**.00000000
La máscara original es /24 (clase C), los números más significativos que no se repiten son 22 por lo tanto la máxima agregación CIDR sería 202.58.128.0/22. Otra forma de pensarlo es que para direccionar 4 direcciones se necesitan 2 bits, por lo tanto 24 - 2 = 22.

17.Listar las redes involucradas por los siguientes bloques CIDR:

1. 200.56.168.0/21

200.56.168.0 = **11001000.00111000.10101**000.00000000
La dirección es clase C que usa máscara /24
Primera red: 11001000.00111000.10101**000**.00000000 = 200.56.168.0
Última red: 11001000.00111000.10101**111**.00000000 = 200.56.175.0

2. 195.24.0.0/13 o 195.24/13

La dirección es clase C por ser 195 el primer octeto, las redes de clase C usan 3 octetos para subredes o sea que es /24. Al ser /13 quiere decir que esta agrupando 24 - 13 = 11 bits de direcciones empezando del bit 13 o sea del segundo octeto, tomando 3 bits de este para subredes y dejando el resto para hosts.

IP: **11000011.00011**000.00000000.00000000 = **195.24.0.0**
Primera red: 11000011.00011**000.00000000**.00000000 = **195.24.0.0**
Última red: 11000011.00011**111.00000000**.00000000 = **195.31.0.0**

18.El bloque CIDR 128.0.0.0/2 o 128/2, ¿Equivale a listar todas las direcciones de red de clase B?

Para el bloque 128/2, solo los dos primeros bits (**10**) estarían fijos, es decir que abarca desde la dirección 128.0.0.0 hasta la 191.255.255.255, que son las direcciones clase B.

19.Una máquina que se conecta a Internet, ¿tiene tabla de ruteo?

Sí, todo nodo en Internet tiene una tabla de ruteo.

20.Describa qué es y para qué sirve ICMP. Qué hacen los comandos `ping` y `traceroute` (`tracert` en Windows).

ICMP

Wikipedia:

El **Protocolo de Mensajes de Control de Internet** o **ICMP** (por sus siglas en inglés de ***I****nternet****C****ontrol* ***M****essage* ***P****rotocol*) es el sub protocolo de control y notificación de errores del Protocolo de Internet (IP). Como tal, se usa para enviar mensajes de error, indicando por ejemplo que un servicio determinado no está disponible o que un router o host no puede ser localizado.

ICMP difiere del propósito de TCP y UDP ya que generalmente no se utiliza directamente por las aplicaciones de usuario en la red. La única excepción es la herramienta ping y traceroute, que envían mensajes de petición Echo ICMP (y recibe mensajes de respuesta Echo) para determinar si un host está disponible, el tiempo que le toma a los paquetes en ir y regresar a ese host y cantidad de hosts por los que pasa.

Kurose-Ross:

ICMP es un protocolo utilizado para intercambiar información acerca de la red. El uso más típico es la generación de informes de error (ej. host, protocolo, puerto, red inalcanzables, etc.). Aunque a menudo se considera parte de IP, se encuentra justo encima de IP ya que los mensajes de ICMP son transmitidos dentro de los datagramas IP.

El programa bien conocido ping envía un mensaje ICMP al host especificado. El host de destino, al ver la solicitud de eco, devuelve una respuesta de eco ICMP. Observe que el programa cliente necesita poder instruir al sistema operativo para generar un mensaje ping.

PING

Wikipedia:

Como programa, **ping** es una utilidad diagnóstica en redes de computadoras que comprueba el estado de la conexión del host local con uno o varios equipos remotos de una red TCP/IP por medio del envío de paquetes ICMP de solicitud y de respuesta. Mediante esta utilidad puede diagnosticarse el estado, velocidad y calidad de una red determinada.

Kurose-Ross:

Se envía un mensaje ICMP Tipo 8 y código 0 a un host especificado, este al recibir la solicitud devuelve con una respuesta de eco. En caso de no llegar el paquete, el último host al que llega devuelve otro paquete con información indicando porqué no se pudo entregar.

TRACEROUTE

Wikipedia:

Traceroute es una consola de diagnóstico que permite seguir la pista de los paquetes que vienen desde un host (punto de red). Se obtiene además una estadística del RTT o latencia de red de esos paquetes, lo que viene a ser una estimación de la distancia a la que están los extremos de la comunicación. Esta herramienta se llama traceroute en UNIX, Mac y GNU/linux, mientras que en Windows se llama tracert.

Kurose-Ross:
Traceroute envia un paquete IP con TTL 1 al primer host en la ruta, al llegar este detecta que se le termina el TTL y envia una respuesta ICMP (tipo 11 código 0) al enviador donde indica su nombre e IP. Luego Traceroute envia otro paquete IP para que lo reciba el siguiente host de la ruta con un TTL 2 y espera la respuesta de este y repite esto hasta llegar al destino armando así una lista de nodos para el camino.

1. Indique el tipo y el código ICMP de un ping.

Tipo 8, Código 0 (solicitud de eco)

2. Indique el tipo y el código ICMP de la respuesta de un ping.

Tipo 0, Código 0 (respuesta de eco)

3. Indique el tipo y el código ICMP del cuál se vale el comando traceroute para funcionar.

Tipo 11, Código 0 (TTL caducado)

21.¿Qué es y para qué sirve la dirección 127.0.0.1? ¿Qué PC responde al siguiente comando: ping 127.0.0.1?

La dirección 127.0.0.1 es por convención la dirección de loopback, es decir, la dirección por la que la máquina se responde a sí misma. Al hacer ping a esta dirección la máquina se responderá a sí misma (siempre y cuando no tenga algún problema de hardware).

22.¿Qué es NAT y para qué sirve? De un ejemplo de su uso y analice cómo funcionaría en ese entorno. Ayuda: analizar el servicio de Internet hogareño y como es posible que para una conexión a Internet existan varias computadoras que usan la conexión.

Wikipedia:
NAT (**Network Address Translation** - **Traducción de Dirección de Red**) es un mecanismo utilizado por enrutadores IP para intercambiar paquetes entre dos redes que se asignan mutuamente direcciones incompatibles. Consiste en convertir, en tiempo real, las direcciones utilizadas en los paquetes transportados. También es necesario editar los paquetes para permitir la operación de protocolos que incluyen información de direcciones dentro de la conversación del protocolo.

Un ejemplo de uso de NAT es para las redes hogareñas comunes. El ISP proporciona solo una IP a un hogar. En este hogar se tiene una red con varios hosts conectados. Dentro de la red hogareña se puede usar cualquier IP independientemente de la que haya asignado el ISP. El router mantiene una tabla NAT donde asocia una IP interna con un número de puerto, luego cuando un host quiere conectarse con un equipo externo, el router envía los paquetes al router, que abre un puerto y lo asocia con la IP de ese host.

23.¿Qué especifica la RFC 1918 y cómo se relaciona con NAT?

Wikipedia:

La rfc 1918 especifica la forma de alocamiento, utilizacion y definicion de redes privadas (http://www.rfc-es.org/rfc/rfc1918-es.txt). NAT fue explicado en el punto anterior y dimos un ejemplo de su uso más común que es permitir utilizar direcciones privadas (definidas en el RFC 1918) para acceder a Internet. Existen rangos de direcciones privadas que pueden usarse libremente y en la cantidad que se quiera dentro de una red privada. Si el número de direcciones privadas es muy grande puede usarse solo una parte de direcciones públicas para salir a Internet desde la red privada. De esta manera simultáneamente sólo pueden salir a Internet con una dirección IP tantos equipos como direcciones públicas se hayan contratado. Esto es necesario debido al progresivo agotamiento de las direcciones IPv4. Se espera que con el advenimiento de IPv6 no sea necesario continuar con esta práctica.

La "Autoridad de Números Asignados en Internet", Internet Assigned Numbers Authority (IANA), ha reservado los tres siguientes bloques de direcciones IP para el uso en internets privadas:

```
10.0.0.0        -   10.255.255.255  (prefijo 10/8)
172.16.0.0      -   172.31.255.255  (prefijo 172.16/12)
192.168.0.0     -   192.168.255.255 (prefijo 192.168/16)
```

24.¿Qué es Ipv6? Enumere diferencias existentes en el formato de datagramas respecto de IPv4.

El **Internet Protocol version 6 (IPv6)** (en español: *Protocolo de Internet versión 6*) es una versión del protocolo Internet Protocol (IP), definida en el RFC 2460 y diseñada para reemplazar a Internet Protocol version 4 (IPv4) RFC 791, que actualmente está implementado en la gran mayoría de dispositivos que acceden a Internet.

IPv4 es un protocolo empleado en la nube para la comunicación de datos, en un inicio se creo para utilizarlo en el presente y futuro de las comunicaciones pero al solo tener 4.294.967.296 direcciones únicas por utilizar 32 bits ya se consumieron estas oficialmente el 3 de Febrero del 2011 y los Registros Regionales de Internet deben, desde ahora, manejarse con sus propias reservas, que se estima, alcanzaran hasta Septiembre de 2011, por lo cual se estableció como futuro sucesor a IPv6 la cual utiliza 128 bits dando cabida a 340.282.366.920.938.463.463.374.607.431.768.211.456 (340 sextillones de direcciones), ustedes pensaran ¿Eso con que se come?, pues en un lenguaje más entendible quiere decir que el "internet" ya se lleno y no cabe un alma mas dentro de esta por lo cual se creó un mundo más grande para que quepan muchas más almas.

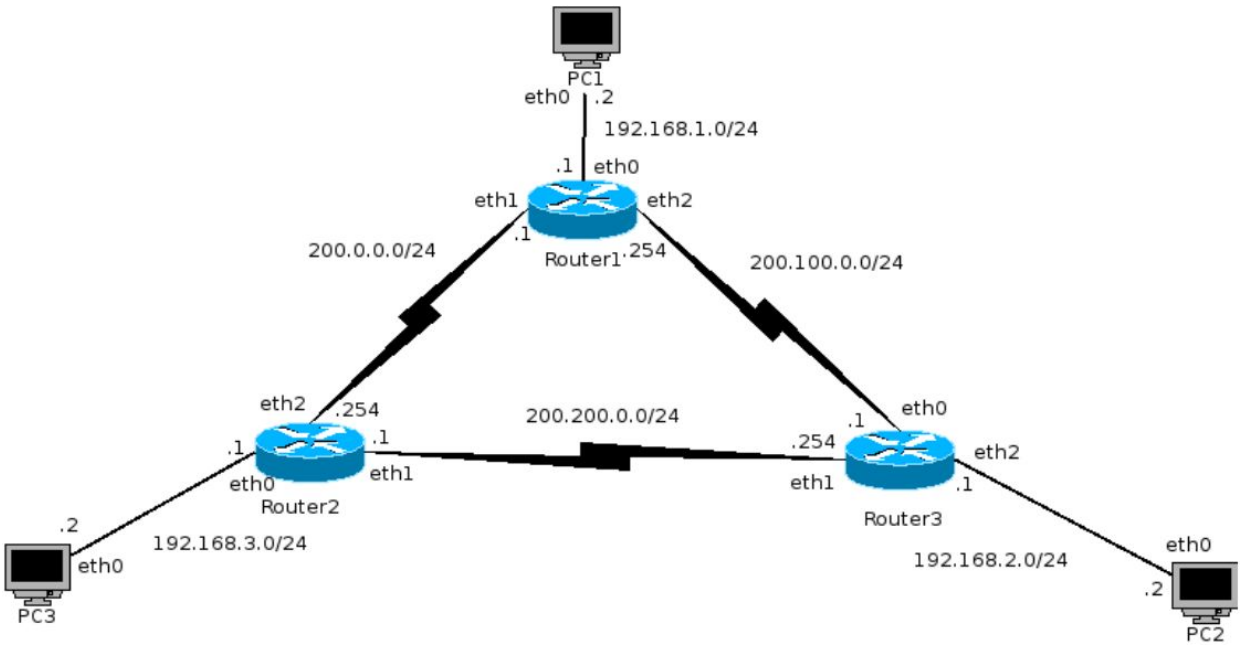
Las diferencias además del tamaño son:

IPv4	IPv6

(ToS) Tipo de servicio	(QoS) Etiqueta de flujo y clase de tráfico
Seguridad opcional	Seguridad extremo a extremo de forma nativa (IPSec)
Configuración Manual o Dinamica (DHCP)	Configuración Plug & Play

De forma más entendible en IPv4 se relacionaba "www.addit.com.mx.A 200.1.17.1" mientras que en IPv6 es "www.addit.com.mx. A6 0 2001:1310:d131:1::1"

25.Utilizando el LiveCD provisto por la cátedra, se simulará la red que muestra el siguiente gráfico:



1. Abra una consola de comandos y ejecute el comando:

topologia capa-red-estatico start

2. Aguarde a que aparezcan cada una de las máquinas involucradas en el gráfico. Cada máquina se representa por una ventana xterminal cuyo título se corresponden con los nombres que muestra el gráfico: PC1, PC2, PC3, Router1, Router2 y Router3

3. Configure cada uno de los equipos considerando:

1. Para iniciar sesión en cada equipo, debe utilizar como nombre de usuario root y contraseña xxxx

2. Utilice el comando ifconfig para configurar las direcciones IP de equipo según las interfaces indicadas en el gráfico. Por ejemplo, en PC3 debe configurar la interfaz eth0 con la IP 192.1683.2, en Router2 eth0 con la IP 192.168.3.1, eth1 con 200.200.0.1 y eth2 con 200.0.0.25

```
router1:~# ifconfig eth0 192.168.1.1
router1:~# ifconfig eth1 200.0.0.1
router1:~# ifconfig eth2 200.100.0.254

pc1:~# ifconfig eth0 192.168.1.2

router2:~# ifconfig eth0 192.168.3.1
router2:~# ifconfig eth1 200.200.0.1
router2:~# ifconfig eth2 200.200.0.254

pc3:~# ifconfig eth0 192.168.3.2

router3:~# ifconfig eth0 200.100.0.1
router3:~# ifconfig eth1 200.200.0.254
router3:~# ifconfig eth2 192.168.2.1

pc2:~# ifconfig eth0 192.168.2.2
```

3. Cada vez que configure los extremos de un enlace, por ejemplo la interfaz eth0 de PC3 y la interfaz eth0 de Router2, compruebe conectividad utilizando el comando ping.

4. Utilice el comando route para configurar las rutas estáticas necesarias en cada equipo. En el caso de los routers debe considerar:

1. **Router1 envía todo el tráfico desconocido a Router2**
2. **Router2 envía todo el tráfico desconocido a Router3**
3. **Router3 envía todo el tráfico desconocido a Router1**

```
router1:~# route add default gw 200.0.0.254
router2:~# route add default gw 200.200.0.254
router3:~# route add default gw 200.100.0.254
pc1:~# route add default gw 192.168.1.1
pc2:~# route add default gw 192.168.2.1
pc3:~# route add default gw 192.168.3.1
```

```
router3:~# route add default gw HostName
```

5. Si un router no intercambia paquetes entre placas, verifique que los siguientes valores del kernel sean los siguientes:

1. Verificar IP_FORWARD, este parámetro admite el intercambio de paquetes entre placas:
   - Para obtener el valor:
     cat /proc/sys/net/ipv4/ip_forward
     El valor en 0 deshabilita su funcionalidad. Un 1 lo habilita.
   - Para cambiar el valor:
     echo 1 > /proc/sys/net/ipv4/ip_forward

2. Verificar RP_FILTER, este parámetro es de seguridad y evita la recepción de paquetes por una interfaz que tengan una IP de origen que pertenezca a una red que el router no rutearía a través de esa interfaz. Este valor debe deshabilitarse en routers:
   - Para obtener el valor:
     cat /proc/sys/net/ipv4/conf/all/rp_filter
     El valor en 0 deshabilita su funcionalidad. Un 1 lo habilita

   - Para cambiar el valor:
     echo 0 >/proc/sys/net/ipv4/conf/all/rp_filter

4. Verifique conectividad entre PC1, PC2 y PC3:

1. Utilizando el comando ping

```
pc1:~# ping 192.168.2.2 -c 1
PING 192.168.2.2 (192.168.2.2) 56(84) bytes of data.
64 bytes from 192.168.2.2: icmp_seq=1 ttl=62 time=0.666 ms

--- 192.168.2.2 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 0.666/0.666/0.666/0.000 ms


pc2:~# ping 192.168.3.2 -c 1
PING 192.168.3.2 (192.168.3.2) 56(84) bytes of data.
64 bytes from 192.168.3.2: icmp_seq=1 ttl=62 time=0.327 ms

--- 192.168.3.2 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 0.327/0.327/0.327/0.000 ms


pc3:~# ping 192.168.1.2 -c1
PING 192.168.1.2 (192.168.1.2) 56(84) bytes of data.
64 bytes from 192.168.1.2: icmp_seq=1 ttl=62 time=0.536 ms

--- 192.168.1.2 ping statistics ---
```

```
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 0.536/0.536/0.536/0.000 ms
```

2. Utilizando el comando traceroute

```
pc2:~# traceroute 192.168.3.2
traceroute to 192.168.3.2 (192.168.3.2), 30 hops max, 40 byte packets
 1  192.168.2.1 (192.168.2.1)  0.254 ms  0.577 ms  0.214 ms
 2  200.0.0.1 (200.0.0.1)  0.547 ms  1.068 ms  0.360 ms
 3  200.200.0.1 (200.200.0.1)  0.402 ms  1.864 ms  1.057 ms
 4  192.168.3.2 (192.168.3.2)  1.560 ms  1.494 ms  0.481 ms

pc1:~# traceroute 192.168.2.2
traceroute to 192.168.2.2 (192.168.2.2), 30 hops max, 40 byte packets
 1  192.168.1.1 (192.168.1.1)  0.182 ms  0.796 ms  0.204 ms
 2  200.200.0.1 (200.200.0.1)  0.478 ms  0.714 ms  0.365 ms
 3  200.100.0.1 (200.100.0.1)  0.438 ms  0.719 ms  0.340 ms
 4  192.168.2.2 (192.168.2.2)  0.629 ms  0.827 ms  0.484 ms

pc3:~# traceroute 192.168.1.2
traceroute to 192.168.1.2 (192.168.1.2), 30 hops max, 40 byte packets
 1  192.168.3.1 (192.168.3.1)  9.177 ms  0.689 ms  0.207 ms
 2  200.100.0.1 (200.100.0.1)  0.481 ms  0.658 ms  0.577 ms
 3  200.0.0.1 (200.0.0.1)  0.446 ms  0.750 ms  0.372 ms
 4  192.168.1.2 (192.168.1.2)  0.588 ms  0.602 ms  0.374 ms
```

3. Utilizando el comando ping -nR

```
pc2:~# ping 192.168.3.2 -c 1 -nR
PING 192.168.3.2 (192.168.3.2) 56(124) bytes of data.
64 bytes from 192.168.3.2: icmp_seq=1 ttl=62 time=0.672 ms
RR:     192.168.2.2
        200.100.0.1
        200.0.0.1
        192.168.3.1
        192.168.3.2
        192.168.3.2
        200.200.0.1
        192.168.2.1
        192.168.2.2


--- 192.168.3.2 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 0.672/0.672/0.672/0.000 ms

pc1:~# ping 192.168.2.2 -c 1 -nR
PING 192.168.2.2 (192.168.2.2) 56(124) bytes of data.
64 bytes from 192.168.2.2: icmp_seq=1 ttl=62 time=0.388 ms
RR:     192.168.1.2
        200.0.0.1
        200.200.0.1
        192.168.2.1
        192.168.2.2
        192.168.2.2
        200.100.0.1
```

```
        192.168.1.1
        192.168.1.2


--- 192.168.2.2 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 0.388/0.388/0.388/0.000 ms

pc3:~# ping 192.168.1.2 -c1 -nR
PING 192.168.1.2 (192.168.1.2) 56(124) bytes of data.
64 bytes from 192.168.1.2: icmp_seq=1 ttl=62 time=0.660 ms
RR:     192.168.3.2
        200.200.0.1
        200.100.0.1
        192.168.1.1
        192.168.1.2
        192.168.1.2
        200.0.0.1
        192.168.3.1
        192.168.3.2


--- 192.168.1.2 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 0.660/0.660/0.660/0.000 ms
```

4. Mientras realiza ping desde una PC, capture paquetes en un router intermedio y verifique qué paquetes pasan por la interfaz. Por ejemplo, mientras PC1 corre el comando ping a la IP de PC2, analice los paquetes que se visualizan en eth0 y en eth1 de Router3. La captura de paquetes puede hacerse con el comando

tcpdump -i *interfaz*

Por ejemplo:

tcpdump -i eth0

```
lihuen@lihuen:~/Desktop$ cat redes4.25.5
router3:~# tcpdump -i eth1 > tcpdump.1
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth1, link-type EN10MB (Ethernet), capture size 96 bytes
^C509 packets captured
793 packets received by filter
284 packets dropped by kernel
router3:~# head tcpdump.1
19:01:18.463140 IP 192.168.1.2 > 192.168.2.2: ICMP echo request, id 20741, seq 608, length 64
19:01:18.470052 IP 200.100.0.1.60676 > 80.58.61.250.domain:  41213+ PTR? 2.2.168.192.in-addr.arpa.
(42)
19:01:18.470061 IP 200.0.0.1.35322 > 80.58.61.250.domain:  39274+ PTR? 1.0.100.200.in-addr.arpa.
(42)
19:01:18.982442 IP 192.168.1.2 > 192.168.2.2: ICMP echo reply, id 20485, seq 530, length 64
19:01:19.470261 IP 192.168.1.2 > 192.168.2.2: ICMP echo request, id 20741, seq 609, length 64
19:01:19.990510 IP 192.168.1.2 > 192.168.2.2: ICMP echo reply, id 20485, seq 531, length 64
```

```
19:01:20.482191 IP 192.168.1.2 > 192.168.2.2: ICMP echo request, id 20741, seq 610, length 64
19:01:21.002321 IP 192.168.1.2 > 192.168.2.2: ICMP echo reply, id 20485, seq 532, length 64
19:01:21.490124 IP 192.168.1.2 > 192.168.2.2: ICMP echo request, id 20741, seq 611, length 64
19:01:22.010312 IP 192.168.1.2 > 192.168.2.2: ICMP echo reply, id 20485, seq 533, length 6


router1:~# tcpdump -i eth1 >  tcpdump.1
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth1, link-type EN10MB (Ethernet), capture size 96 bytes
^C530 packets captured
785 packets received by filter
255 packets dropped by kernel
router1:~# head tcpdump.1
19:01:53.297598 IP 192.168.1.2 > 192.168.2.2: ICMP echo reply, id 20485, seq 564, length 64
19:01:53.301985 IP 200.0.0.1.51513 > 80.58.61.250.domain:  37536+ PTR? 2.2.168.192.in-addr.arpa.
(42)
19:01:53.541331 IP 200.100.0.1.36074 > 80.58.61.254.domain:  32580+ PTR? 2.1.168.192.in-addr.arpa.
(42)
19:01:53.797309 IP 192.168.1.2 > 192.168.2.2: ICMP echo request, id 20741, seq 643, length 64
19:01:54.305358 IP 192.168.1.2 > 192.168.2.2: ICMP echo reply, id 20485, seq 565, length 64
19:01:54.805204 IP 192.168.1.2 > 192.168.2.2: ICMP echo request, id 20741, seq 644, length 64
19:01:55.264929 arp who-has 200.0.0.254 tell 200.0.0.1
19:01:55.265185 arp reply 200.0.0.254 is-at fe:fd:00:00:02:02 (oui Unknown)
19:01:55.317382 IP 192.168.1.2 > 192.168.2.2: ICMP echo reply, id 20485, seq 566, length 64
19:01:55.817083 IP 192.168.1.2 > 192.168.2.2: ICMP echo request, id 20741, seq 645, length 64

router1:~# tcpdump -i eth0 >  tcpdump.0
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 96 bytes
^C2137 packets captured
2264 packets received by filter
127 packets dropped by kernel
router1:~# head tcpdump.0
19:09:20.646500 IP 192.168.2.2 > 192.168.1.2: ICMP echo request, id 20485, seq 1007, length 64
19:10:00.727586 IP 192.168.1.2 > 192.168.2.2: ICMP echo reply, id 20485, seq 1007, length 64
19:09:20.966081 IP 192.168.1.2 > 192.168.2.2: ICMP echo request, id 20741, seq 1086, length 64
19:09:20.966445 IP 192.168.2.2 > 192.168.1.2: ICMP echo reply, id 20741, seq 1086, length 64
19:09:21.658023 IP 192.168.2.2 > 192.168.1.2: ICMP echo request, id 20485, seq 1008, length 64
19:09:21.658180 IP 192.168.1.2 > 192.168.2.2: ICMP echo reply, id 20485, seq 1008, length 64
19:09:21.977999 IP 192.168.1.2 > 192.168.2.2: ICMP echo request, id 20741, seq 1087, length 64
19:09:21.978357 IP 192.168.2.2 > 192.168.1.2: ICMP echo reply, id 20741, seq 1087, length 64
19:09:22.666009 IP 192.168.2.2 > 192.168.1.2: ICMP echo request, id 20485, seq 1009, length 64
19:09:22.666106 IP 192.168.1.2 > 192.168.2.2: ICMP echo reply, id 20485, seq 1009, length 64


router3:~# tcpdump -i eth0 > tcpdump.0
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 96 bytes
^C1040 packets captured
1262 packets received by filter
222 packets dropped by kernel
router3:~# head tcpdump.0
19:09:11.550553 IP 192.168.2.2 > 192.168.1.2: ICMP echo request, id 20485, seq 998, length 64
19:09:11.554135 IP 200.100.0.1.58368 > 80.58.61.250.domain:  63946+ PTR? 2.1.168.192.in-addr.arpa.
(42)
19:09:11.881444 IP 192.168.2.2 > 192.168.1.2: ICMP echo reply, id 20741, seq 1077, length 64
19:09:12.561125 IP 192.168.2.2 > 192.168.1.2: ICMP echo request, id 20485, seq 999, length 64
```

```
19:09:12.893398 IP 192.168.2.2 > 192.168.1.2: ICMP echo reply, id 20741, seq 1078, length 64
19:09:13.569037 IP 192.168.2.2 > 192.168.1.2: ICMP echo request, id 20485, seq 1000, length 64
19:09:13.901137 IP 192.168.2.2 > 192.168.1.2: ICMP echo reply, id 20741, seq 1079, length 64
19:09:14.580996 IP 192.168.2.2 > 192.168.1.2: ICMP echo request, id 20485, seq 1001, length 64
19:09:14.901257 IP 192.168.2.2 > 192.168.1.2: ICMP echo reply, id 20741, seq 1080, length 64
19:09:15.589068 IP 192.168.2.2 > 192.168.1.2: ICMP echo request, id 20485, seq 1002, length 64
```

5. Relevamiento: Utilizando el comando “route -n” o “netstat -nr” indique la configuración de las tablas de rutas tanto de los routers como la de las PCs especificando para cada dispositivo:

Red Destino	Gateway	Máscara de Red	Interface

```
router1:~# route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
200.100.0.0     0.0.0.0         255.255.255.0   U     0      0        0 eth2
192.168.1.0     0.0.0.0         255.255.255.0   U     0      0        0 eth0
200.0.0.0       0.0.0.0         255.255.255.0   U     0      0        0 eth1
0.0.0.0         200.0.0.254     0.0.0.0         UG    0      0        0 eth1

router3:~# route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
192.168.2.0     0.0.0.0         255.255.255.0   U     0      0        0 eth2
200.100.0.0     0.0.0.0         255.255.255.0   U     0      0        0 eth0
200.200.0.0     0.0.0.0         255.255.255.0   U     0      0        0 eth1
0.0.0.0         200.100.0.254   0.0.0.0         UG    0      0        0 eth0

router2:~# route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
192.168.3.0     0.0.0.0         255.255.255.0   U     0      0        0 eth0
200.0.0.0       0.0.0.0         255.255.255.0   U     0      0        0 eth2
200.200.0.0     0.0.0.0         255.255.255.0   U     0      0        0 eth1
0.0.0.0         200.200.0.254   0.0.0.0         UG    0      0        0 eth1
```

1. Si la estación PC1 le envía un ping a la estación PC2:

```
pc1:~# ping 192.168.2.2 -nR
PING 192.168.2.2 (192.168.2.2) 56(124) bytes of data.
64 bytes from 192.168.2.2: icmp_seq=1 ttl=62 time=0.593 ms
RR:     192.168.1.2
        200.0.0.1
        200.200.0.1
```

```
        192.168.2.1
        192.168.2.2
        192.168.2.2
        200.100.0.1
        192.168.1.1
        192.168.1.2
```

1. ¿Cúal es el camino por el que viaja el requerimiento?
   PC1->Router1->Router2->Router3->PC2

2. ¿Cúal es el camino por el que viaja la respuesta?
   PC2->Router3->Router1->PC1

2. Evalúe lo mismo para comunicaciones entre la PC1 con la PC3 y entre la PC2 con la PC3.

```
pc1:~# ping 192.168.3.2 -nR
PING 192.168.3.2 (192.168.3.2) 56(124) bytes of data.
64 bytes from 192.168.3.2: icmp_seq=1 ttl=61 time=11.6 ms
RR:     192.168.1.2
        200.0.0.1
        192.168.3.1
        192.168.3.2
        192.168.3.2
        200.200.0.1
        200.100.0.1
        192.168.1.1
        192.168.1.2
```

PC1->Router1->Router2->PC3->PC3->Router2->Router3->Router1->PC1

```
pc2:~# ping 192.168.3.2 -nR
PING 192.168.3.2 (192.168.3.2) 56(124) bytes of data.
64 bytes from 192.168.3.2: icmp_seq=1 ttl=62 time=0.846 ms
RR:     192.168.2.2
        200.100.0.1
        200.0.0.1
        192.168.3.1
        192.168.3.2
        192.168.3.2
        200.200.0.1
        192.168.2.1
        192.168.2.2
```

PC2->Router3->Router1->Router3->PC3->PC3->Router2->Router3->PC2

6. Mantenimiento: Suponiendo que en Router1 se agregó una interfaz de red con la dirección IP 163.10.10.1/24:

1. Especifique los cambios necesarios de modo que tanto PC1, PC2 como PC3 puedan comunicarse exitosamente con los hosts de esta nueva red.

No hace falta hacer cambio alguno porque las PCs se comunicarán por el gateway por defecto.

2. En base a lo anterior, ¿qué puede decir respecto del mantenimiento del ruteo de la red?

A medida que crece la red crece la complejidad del mantenimiento.

7. ICMP y RUTEO 1: Desde la PC1, realice un ping a la dirección IP 5.5.5.5 ¿Qué indica el mensaje de error recibido?, ¿Quién lo envía?

```
pc1:~# ping 5.5.5.5 -c1
PING 5.5.5.5 (5.5.5.5) 56(84) bytes of data.
From 192.168.1.1 icmp_seq=1 Time to live exceeded

--- 5.5.5.5 ping statistics ---
1 packets transmitted, 0 received, +1 errors, 100% packet loss, time 0ms
```

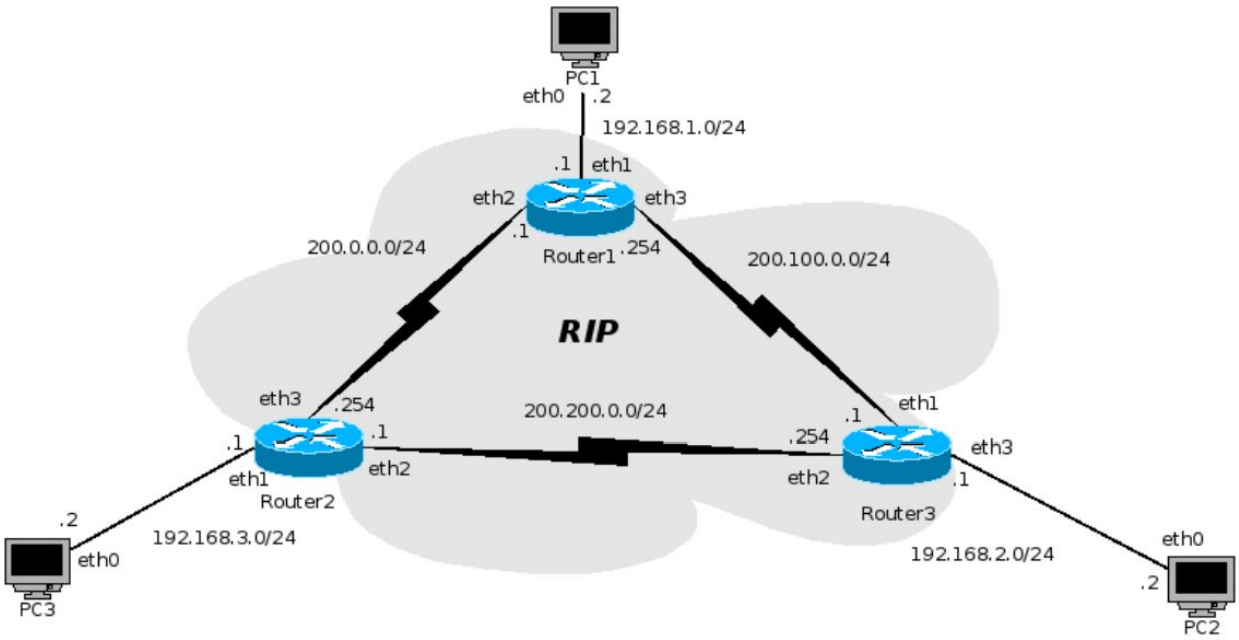
8. ICMP y RUTEO 2: Desde la PC1, realice un ping a la dirección IP 192.168.2.5 ¿Qué indica el mensaje de error recibido?, ¿Quién lo envía?

```
pc1:~# ping 192.168.2.5 -c 1
PING 192.168.2.5 (192.168.2.5) 56(84) bytes of data.
From 200.100.0.1 icmp_seq=1 Destination Host Unreachable

--- 192.168.2.5 ping statistics ---
1 packets transmitted, 0 received, +1 errors, 100% packet loss, time 0ms
```

9. ICMP y RUTEO 3: Provoque un loop de enrutamiento entre los routers con la red 180.20.10.0/24 y luego desde la PC1, realice un ping a la dirección 180.20.10.1. ¿Qué indica el mensaje de error recibido?, ¿Quién lo envía?

```
router1:~#  route add -net 180.20.10.0 gw 200.100.0.1 netmask 255.255.255.0
router2:~#  route add -net 180.20.10.0 gw 200.0.0.1 netmask 255.255.255.0
router3:~#  route add -net 180.20.10.0 gw 200.200.0.1 netmask 255.255.255.0

pc1:~# ping 192.168.2.5 -c 1
PING 192.168.2.5 (192.168.2.5) 56(84) bytes of data.
From 200.100.0.1 icmp_seq=1 Destination Host Unreachable

--- 192.168.2.5 ping statistics ---
1 packets transmitted, 0 received, +1 errors, 100% packet loss, time 0ms

PING 180.20.10.1 (180.20.10.1) 56(84) bytes of data.
From 192.168.1.1 icmp_seq=1 Time to live exceeded

--- 180.20.10.1 ping statistics ---
1 packets transmitted, 0 received, +1 errors, 100% packet loss, time 0ms
```

26.Utilizando el LiveCD provisto por la cátedra, se simulará la red que muestra el siguiente gráfico:



1. Abra una consola de comandos y ejecute el comando:

topologia capa-red-dinamico start

2. Aguarde a que aparezcan cada una de las máquinas involucradas en el gráfico. Cada máquina se representa por una ventana xterminal cuyo titulo se corresponden con los nombres que muestra el gráfico: PC1, PC2, PC3, Router1, Router2 y Router3

3. Cada equipo de la de red ya se encuentra configurado y el ruteo es dinámico utilizando RIP. El nombre de usuario de las máquinas es root y su contraseña xxxx

4. Verifique conectividad entre PC1, PC2 y PC3:

1. **Utilizando el comando ping**
2. **Utilizando el comando traceroute**
3. **Utilizando el comando ping -nR**

```
pc1:~# ping 192.168.2.2 -nR -c1
PING 192.168.2.2 (192.168.2.2) 56(124) bytes of data.
64 bytes from 192.168.2.2: icmp_seq=1 ttl=62 time=32.7 ms
RR:     192.168.1.2
        200.100.0.254
        192.168.2.1
        192.168.2.2
        192.168.2.2
        200.100.0.1
        192.168.1.1
        192.168.1.2


--- 192.168.2.2 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 32.768/32.768/32.768/0.000 ms


router1:~# route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
192.168.3.0     200.0.0.254     255.255.255.0   UG    2      0        0 eth2
192.168.2.0     200.100.0.1     255.255.255.0   UG    2      0        0 eth3
```

```
200.100.0.0     0.0.0.0         255.255.255.0   U     0      0        0 eth3

192.168.1.0     0.0.0.0         255.255.255.0   U     0      0        0 eth1

10.250.0.0      0.0.0.0         255.255.255.0   U     0      0        0 eth0

200.0.0.0       0.0.0.0         255.255.255.0   U     0      0        0 eth2

200.200.0.0     200.0.0.254     255.255.255.0   UG    2      0        0 eth2
```

5. Relevamiento: Utilizando el comando “route -n” o “netstat -nr” indique la configuración de las tablas de rutas tanto de los routers como la de las PCs especificando para cada dispositivo:

Red Destino	Gateway	Máscara de Red	Interface

```
router1:~# route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
192.168.3.0     200.0.0.254     255.255.255.0   UG    2      0        0 eth2
192.168.2.0     200.100.0.1     255.255.255.0   UG    2      0        0 eth3
200.100.0.0     0.0.0.0         255.255.255.0   U     0      0        0 eth3
192.168.1.0     0.0.0.0         255.255.255.0   U     0      0        0 eth1
10.250.0.0      0.0.0.0         255.255.255.0   U     0      0        0 eth0
200.0.0.0       0.0.0.0         255.255.255.0   U     0      0        0 eth2
200.200.0.0     200.0.0.254     255.255.255.0   UG    2      0        0 eth2

pc1:~# route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
192.168.1.0     0.0.0.0         255.255.255.0   U     0      0        0 eth0
0.0.0.0         192.168.1.1     0.0.0.0         UG    0      0        0 eth0
```

1. Si la estación PC1 le envía un ping a la estación PC2:

3. ¿Cúal es el camino por el que viaja el requerimiento?
4. ¿Cúal es el camino por el que viaja la respuesta?

2. Evalúe lo mismo para comunicaciones entre la PC1 con la PC3 y entre la PC2 con la PC3.

6. Mantenimiento: Dada las rutas obtenidas en el punto anterior, daremos de baja uno de los enlaces verificando el funcionamiento del ruteo dinámico. Para ello, debe utilizar el comando

```
ifconfig INTERFACE down
```

Por ejemplo, supongamos que el enlace que une Router1 con Router2 sufre una caída. Simular esto dando de baja la interfaz eth2 de Router1 (ifconfig eth2 down) y la interfaz eth3 de Router2 (ifconfig eth3 down).

7. Verifique el correcto funcionamiento desde cada una de las PCs al resto.