Capa De Red

1. ¿Qué servicios presta la capa de red? ¿Cuál es la PDU en esta capa?
La capa de Red permite la conexión desde un host origen a un host destino. En TCP/IP está implementada en el protocolo **IP**, e interviene en cada host y encaminador intermedio -**router**-.
El PDU es el **datagrama**, que encapsula los segmentos de **Transporte** agregándole las direcciones IP origen y destino.
Ofrece dos tipos de servicios:

- **Retransmisión (forwarding):** Los datagramas que llegan a un router son dirigidos a la interfaz de salida apropiada.
- **Encaminamiento (routing):** Determina el camino que toman los paquetes desde el origen al destino (según diferentes algoritmos de routeo).

Otro tipo de servicio, aunque ausente en Internet, es el **establecimiento de conexión** (en redes tipo ATM -de conmutación de circuitos-): los routers intermedios del camino establecen la conexión virtual, reservando recursos -bandwidth, buffers...- y "*recordando*" el camino elegido antes que los paquetes se comiencen a transmitir (agregan entrada a tabla de ruteo).

2. Compare los siguientes modelos de servicios de red:

	Datagrama	**Circuitos Virtuales**
¿Todos los paquetes siguen el mismo camino?	**NO**	**SI**
¿Cuenta con una fase de establecimiento y otra de cierre de circuito?	**NO**	**SI**
¿Usa mensajes de señalización?	**NO**	**SI** -para establecimiento/mantención/cierre de conexión-
¿Usa tablas de enrutamiento?	**SI** (rango/prefijo IPs destino, interface OUT)	**SI** (interface IN, #CV IN, interface OUT, #CV OUT)

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 tanto estática como dinámica. Describa conceptualmente como funciona cada uno de ellos e indique ventajas y desventajas de cada método.

- **Ruteo estático:** En redes en las que las rutas cambian muy lentamente en el tiempo, por ejemplo porque un administrador edita manualmente la tabla de ruteo de un router. Se basan en tablas de ruteo.
    - **V:** Algoritmos de ruteo más simples. Apropiado si el tráfico de red es predecible. Los cambios se reflejan ni bien el administrador modifica las tablas. No hay sobrecarga de la red -no hay intercambio de información entre routers-. No hay problemas de seguridad ni compatibilidad. No hay procesamiento extra.

    - **D:** Precisa configuración manual. Se vuelve inmanejable en redes grandes y cambiantes. Propensa a errores -por ejemplo, si no se actualiza una tabla un router puede enviar información por un camino que no llegará nunca al destino; eventualmente el datagrama será descartado por vencer su *TTL* o se perderá por *buffer overflow*-.

- **Ruteo dinámico:** Las rutas cambian frecuentemente de acuerdo a la carga de la red y las esperas que se producen, o bien por cambios en la topología. Puede ejecutarse el algoritmo de ruteo dinámico periódicamente o por cambios en la topología o en los costos de los enlaces. Si un router detecta un cambio, recalcula su tabla y propaga el cambio a los vecinos para que se recalculen.
    - **V:** Reacciona automáticamente a cambios en la red. Precisa poca configuración.
    - **D:** Susceptible a *ciclos* y a *oscilaciones* entre routers. Sobrecarga de red por intercambio de información entre routers. Procesamiento extra por el algoritmo. Los cambios detectados por un router pueden tardar en informarse al resto.

5. Los algoritmos de ruteo dinámico se dividen en Estado Enlace y Vector Distancia. Dado el siguiente cuadro compare:

	Vector distancia	**Estado de enlace**
¿Cada router conoce la topología completa?	**NO** (decentralizado)	**SI** (global)

¿Converge rápidamente? Tiempo de convergencia: Velocidad con la que los routers comparten información. La convergencia ocurre cuando todos los routers del dominio están de acuerdo en las rutas que se encuentran disponibles.	**NO**	**SI**
Protocolos que lo implementan	**RIP v1 y v2** **IGRP** **GGP**	**OSPF** **IS-IS** (de ISO)

Vector distancia: Algoritmo decentralizado, iterativo, asíncrono y distribuido. Cada nodo conoce sólo a sus vecinos, y ninguno conoce el camino completo para un paquete. Cada nodo realiza sus cálculos para mantener su tabla de distancias (con distancia por vecino/destino) e informa los resultados **completos** a sus vecinos directos; el proceso se repite hasta que ningún nodo intercambia información. **Propenso a bucles**. Protocolos:

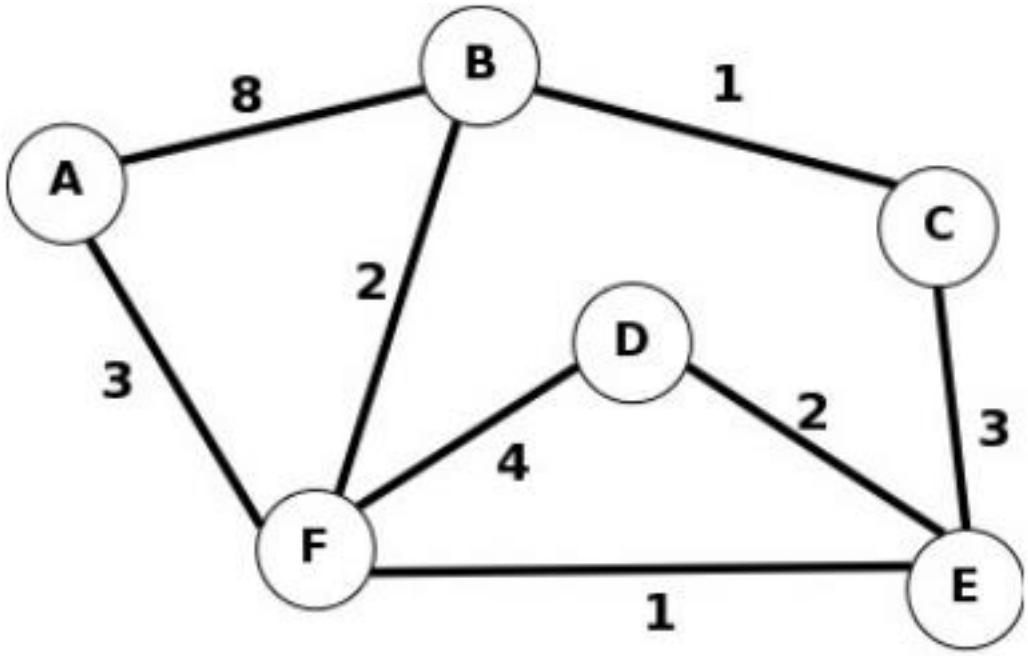
- **RIP:** comparte toda la tabla cada 30 segundos. La versión 1 considera sólo clases completas transmitiendo por *broadcast*, la 2 envía en los paquetes la máscara de subred, por *multicast*. Métrica: cantidad de "saltos" (limitada hasta 15, valores mayores se ignoran para evitar bucles).

- **IGRP:** comparte toda la tabla por broadcast cada 90 segundos, y las actualizaciones al ser detectadas. Métrica compuesta por ancho de banda, retardo, carga, fiabilidad y MTU (Maximum Transfer Unit, la cantidad máxima de información que pueda transportar un paquete de la capa de enlace con el protocolo que se use en ella).

Estado de Enlace: Cada nodo conoce la imagen completa del "grafo" que representa la red y costo **sólo de los routers vecinos**, pudiendo todos conocer así la estructura completa). Aplican **Dijkstra** para calcular el camino de costo mínimo desde el origen al destino, obteniendo el camino completo para el paquete. **Libre de bucles**.

Lo primero que hace un router es detectar a sus vecinos enviando un paquete **HELO** por *broadcast*; si alguien responde le envía un **ECHO** para medir el tiempo de respuesta. Luego construye un paquete con la información recogida -puede hacerlo periódicamente o sólo al detectar cambios-, y lo informa a la red por medio de *inundación o flooding.* Cada paquete tiene un #secuencia, de modo que sólo se procesan y retransmiten los de #secuencia mayor al último recibido. Estos paquetes a su vez, mantienen su *edad* -TTL- que se decrementa en cada router, eliminándose los paquetes de edad 0 (evitando problemas por bucles).

6. Dado el apunte que explica en detalle los algoritmos de ruteo dinámicos de estado enlace y vector distancia, y dado el siguiente grafo, indique el procedimiento para calcular el camino de costo mínimo a partir del nodo B según los siguientes cuadros:



ESTADO DE ENLACE

Iteración N D(A),p(A) D(C),p(C) D(D),p(D) D(E),p(E) D(F),p(F)

0	B	8,B	**1,B**		2,B
1	B**C**	8,B		4,C	**2,B**
2	BC**F**	5,F	6,F	**3,F**	
3	BCF**E**	**5,F**	5,E		
4	BCFE**A**		**5,E**		
5	BCFEA**D**				

VECTOR DISTANCIA

$D^B(\ ,\ )$ **A** **F** **C** (< vecinos)

A	8	**5**	7 conviene ir a A pasando por F
C	11	5	**1** conviene ir a C directamente
D	11	**5**	6 conviene ir a D pasando por F
E	9	**3**	4 conviene ir a E pasando por F
F	8	**2**	4 conviene ir a F directamente

(^ destinos)

7. ¿Qué son los sistemas autónomos y por qué resultan necesarios?

Un **Sistema Autónomo (Autonomous System, AS)** es un conjunto de redes bajo la misma administración (podría ser gestionada por más de un operador de red), y utilizando uno o varios protocolos de enrutamiento internamente, independientemente de la red de su proveedor. Cada SA mantiene una clara y única política de ruteo. Cada AS en Internet debe tener un número identificador: **ASN** (AS Number).

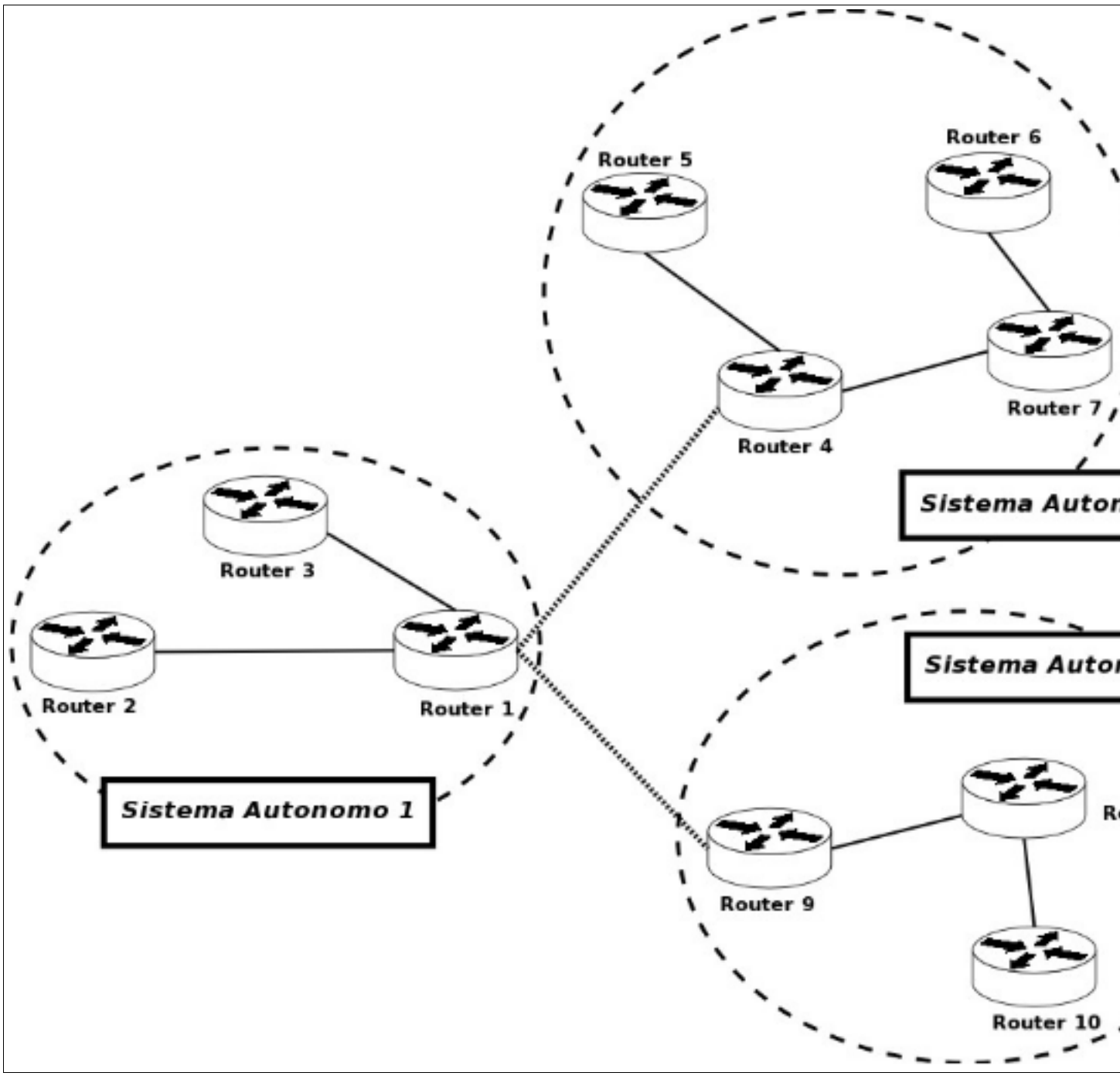
Los protocolos de ruteo que utilizan internamente se denominan **IGP** (Interior Gateway Protocol), y pueden ser: RIP, IGRP, EIGRP, OSPF, IS-IS...

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 supondrían 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.

8. A partir del siguiente gráfico indique



9. Dadas las siguientes direcciones IP, indique su clase y si las mismas corresponden a direcciones de máquina, de red o broadcast:

Dirección IP	Clase	Tipo
203.15.6.87	**C**	**Host/Interface**
67.154.0.0	**A**	**Host/Interface**
171.58.0.0	**B**	**Red**
127.0.0.1	**A --Loopback en realidad--**	**Loopback / Host/Interface**
24.130.56.0	**A**	**Host/Interface**
197.54.66.255	**C**	**Broadcast**

Clase	Rango	Primer Byte	N° de Redes	N° de Host	Máscara de Red	Broad
A	1.0.0.0 - 127.0.0.0	0xxx xxxx	126	16.777.214	255.0.0.0	x.255.255.
B	128.0.0.0 - 191.255.0.0	10xx xxxx	16.384	65.534	255.255.0.0	x.x.255.25
C	192.0.0.0 - 223.255.255.0	110x xxxx	2.097.152	254	255.255.255.0	x.x.x.255
D - *Multicast-*	224.0.0.0 - 239.255.255.255	1110 xxxx				
E - *Reservada-*	240.0.0.0 - 255.255.255.255	1111 xxxx				

- Una dirección IP no identifica un equipo (computadora/router) específico, sino que identifica una **interface** de un host en una red. Un equipo con acceso a múltiples redes (por ejemplo, un router) debe tener asignada una dirección IP por cada una de éstas.
- La dirección 0.0.0.0 es utilizada por las máquinas cuando están arrancando o no se les ha asignado dirección.
- La dirección que tiene su parte de host a cero sirve para definir la red en la que se ubica. Se denomina **dirección de red**.
- La dirección que tiene su parte de host a unos sirve para comunicar con todos los hosts de la red indicada. Se denomina **dirección de broadcast directo**.

- La dirección que tiene su parte de red y de host a unos sirve para comunciar con todos los equipos de la red local (IP 255.255.255.255). Se denomina **dirección de broadcast limitado**.
- Las direcciones 127.x.x.x se reservan para pruebas de retroalimentación. Se denomina **dirección de bucle local** o **loopback**.

- Hay ciertas direcciones en cada clase de dirección IP que no están asignadas y que se denominan **direcciones privadas**; pueden ser utilizadas por los hosts que usan *traducción de dirección de red* (**NAT**) para conectarse a una red pública o por los hosts que no se conectan a Internet. En una misma red no puede existir dos direcciones iguales, pero sí se pueden repetir en dos redes privadas que no tengan conexión entre sí o que sea a través de NAT. Las direcciones privadas son:
- Clase A: 10.x.x.x (8 bits red, 24 bits hosts)
- Clase B: 172.16.x.x a 172.31.x.x (12 bits red, 20 bits hosts)
- Clase C: 192.168.x.x (16 bits red, 16 bits hosts)

10.¿Qué son las subredes?

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.Dada la red 195.200.45.0. 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.

IP: 192.200.45.0, en binario: **110**00000 11001000 00101101 00000000
Como empieza con 110 entonces es **clase C.** La máscara de red es 11111111 11111111 11111111 00000000 o /24.

Para obtener 9 subredes, se piden prestados 4 bytes: $2^4 = 16 > 9$ (si se pedían 3 se tendrían $2^3=8$ subredes solamente, así que se crearán más subredes de las necesarias)
Se usará la máscara de subred **/28** o **255.255.255.240.**
Las direcciones de las subredes son:

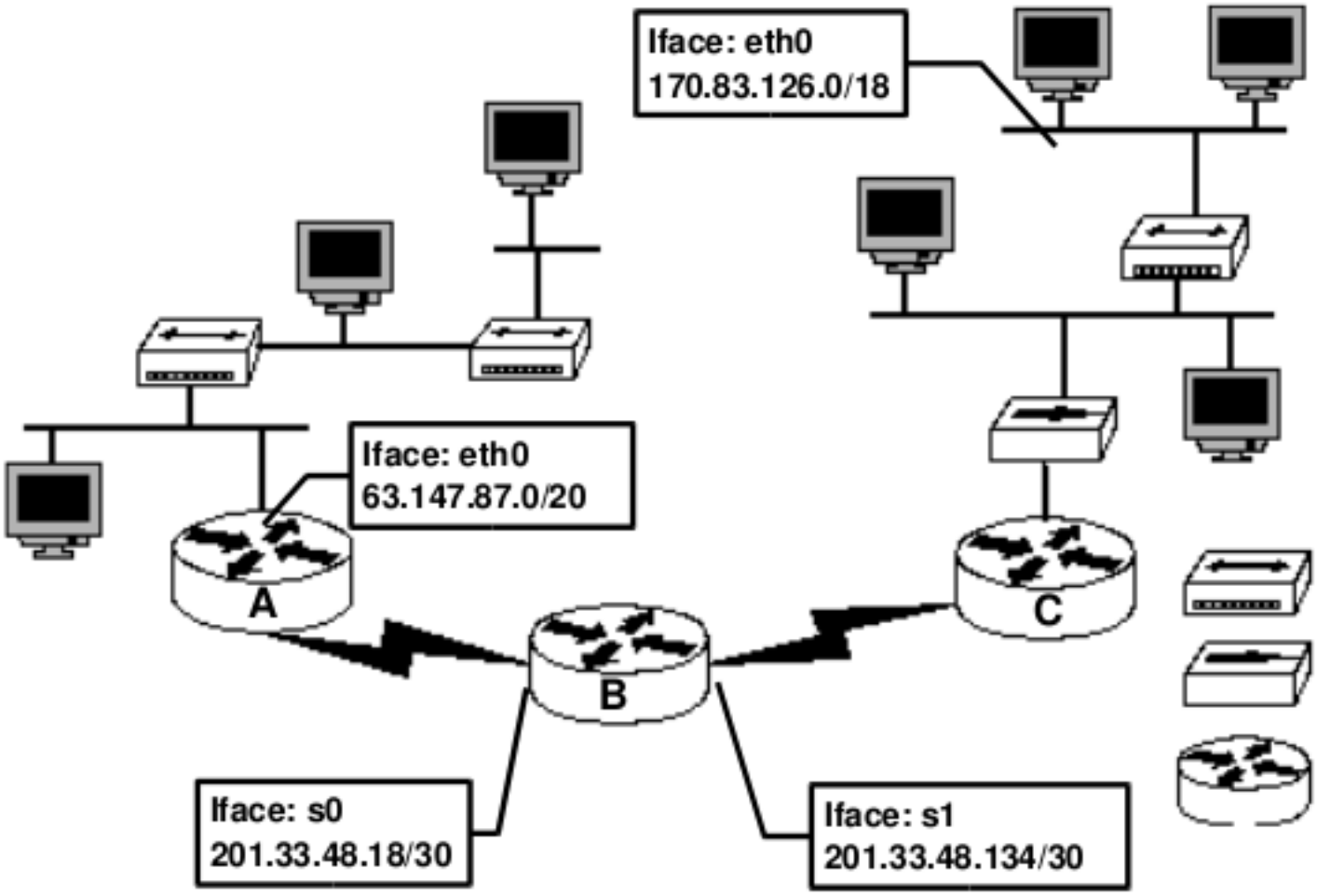
192.200.45.0000 0000 =	0	subred #1
.0001 0000 =	16	subred #2
.0010 0000 =	32	subred #3
.0011 0000 =	48	subred #4
.0100 0000 =	64	subred #5
.0101 0000 =	80	subred #6
.0110 0000 =	96	subred #7
.0111 0000 =	112	subred #8
.1000 0000 =	128	subred #9
.1001 0000 =	144	subred #10 (no se utilizará)
.1010 0000 =	160	subred #11 (no se utilizará)
.1011 0000 =	176	subred #12 (no se utilizará)
.1100 0000 =	192	subred #13 (no se utilizará)
.1101 0000 =	208	subred #14 (no se utilizará)
.1110 0000 =	224	subred #15 (no se utilizará)
.1111 0000 =	240	subred #16 (no se utilizará)

Para la subred #2:

Dirección de Subred:	192.200.45.16/28 (0001 0000)
Dirección de broadcast:	192.200.45.31/28 (0001 1111)
Rango de direcciones asignables:	192.200.45.17/28 - 192.200.45.30/28

Cada subred tendrá 2^4 - 2 = **14 direcciones** IP disponibles (14 interfaces).

12.Dado el siguiente gráfico complete



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

Interface	IP	Dir. Red	Clase	Dir. Subred	Máscara	Broadcast
eth0	**170.83.126.0**	**170.83.0.0**	**B**	**#2: 170.83.64.0**	**/18 = 255.255.192.0**	**170.83.127.255**
eth0'	**63.147.87.0**	**63.0.0.0**	**A**	**#2357: 63.147.80.0**	**/20 = 255.255.240.0**	**63.147.95.255**
s0	**201.33.48.18**	**201.33.48.0**	**C**	**#4: 201.33.48.16**	**/30 = 255.255.255.252**	**201.33.48.19**
s1	**201.33.48.134**	**201.33.48.0**	**C**	**#33: 201.33.48.132**	**/30 = 255.255.255.252**	**201.33.48.135**

Hasta 127 A; hasta 191 B; hasta 223 C; hasta 239 Multicast.

Subred de eth0:
170.83.126.0 en 170.83.0.0/18
2 ^(18-16) = 2^2 = **4 subredes** de 2^(32-18) = 2^14 = **16384 interfaces cada una (16382 útiles)**.
126 = 01111110 subred #2 dirección: 170.83.01000000.0 = **subred 170.83.64.0**
broadcast**:** 170.83.01 111111.255 **= broadcast 170.83.127.255**
Subred eth0:
63.147.87.0 en 63.0.0.0/20
2^(20-8) = 2^12 = **4096 subredes** de 2^(32-20) = 2^12 = **4096 interfaces cada una (4094 útiles)**
63.10010011.01010111 subred #2357 dirección: 63.147.01010000.0 = **subred 63.147.80.0**
broadcast: 63.147.0101 1111.255 = **broadcast 63.147.95.255**
Subred de s0:
201.33.48.18 en 201.33.48.0/30
2^(30-24) = 2^6 = **64 subredes** de 2^(32-30) = **4 interfaces cada una (2 útiles)**
18 = 00010010 subred #4 dirección: 201.33.48.16
broadcast: 201.33.48.000100 11 = **broadcast 201.33.48.19**
Subred de s1:
201.33.48.134 en 201.33.48.0/30
2^(30-24) = 2^6 = **64 subredes** de 2^(32-30) = **4 interfaces cada una (2 útiles)**
134 = 10000110 subred #33 dirección: 201.33.48.132
broadcast: 201.33.48.100001 11 = **broadcast 201.33.48.135**

2. Indique la dirección IP en cada una de las interfaces de cada uno de los routers.

Router A interfaz a B es: 201.33.48.17
Como tiene máscara /30 es para sólo dos IP disponibles 2^(32-30) - 2 = 2 . La interfaz de B tiene 201.33.48.18 así que la restante es la de A.
201.33.48.16 (subred), 201.33.48.17 (A), 201.33.48.18 (B), 201.33.48.19 (broadcast)
Router C interfaz a B: 201.33.48.133
201.33.48.132 (subred), 201.33.48.133 (C), 201.33.48.134 (B), 201.33.48.135 (broadcast)
La interfaz de C hacia 170.83.0.0 no se puede determinar con la información disponible

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

	Tabla Router A			**Tabla Router B**		
Red Destino	**Próximo Router***	**Cant Saltos****	**Red Destino**	**Próximo Router***	**Cant Saltos****	**Red De**
63.0.0.0	**-**	**1**	**63.0.0.0**	**201.33.48.17 (A)**	**2**	**63.0.0.0**
170.83.0.0	**201.33.48.18 (B)**	**3**	**170.83.0.0**	**201.33.48.133 (C)**	**2**	**170.83.**
201.33.48.0	**-**	**1**	**201.33.48.0**	**-**	**1**	**201.33.**

* gateway, ** métrica

13.Dada la IP 65.0.56.34. Se necesitan definir 934 subredes. Indique cuál es la máscara utilizada, y luego describa la subred 817 indicando el rango de direcciones asignables, dirección de red y broadcast.

IP 65.0.56.34 es **Clase A** (65 < 127); se usaran **8 bits para dirección de red**. Para lograr 934 subredes se precisan: Log2(934) = 9.86 es decir **10 bits para indicar la subred.**

Así, la máscara completa es: 255.1111 1111.1100 0000.0 = **máscara de subred 255.255.192.0** (se definirán 2^10 = 1024 redes, sólo se utilizarán las primeras 934).

Se tendrá disponible 2^(32-18) = 2^14 = **16384 direcciones IP**. Serán utilizables: 16384 - 934*2 = **14516 útiles** (descontando subred y broadcast de cada subred).

Para la subred #817 (817_{10} = 1100110001_2, + 1 por el 0: 1100110010_2):

Dirección de red: 65.1100 1100.1000 0000.0 = **65.204.128.0 (subred)**

Broadcast: 65.1100 1100.1011 1111.255 = **65.204.191.255 (broadcast)**

Rango asignable: **desde 65.204.128.1 hasta 65.204.191.254**.

14.Dada la dirección IP 172.16.58.223/26. ¿Cuál es la dirección de subred? ¿Y la de broadcast?

La IP 172.16.58.233 sería de **clase B** (127 < 172 < 192). Mascara de Red: 255.255.0.0. Dirección de RED: 172.16.0.0. Dirección de broadcast de RED: 172.16.255.255

/26 = 255.255.255.192 es la máscara de subred

2^(26 bits de máscara subred -16 bits por ser clase B) = 2^10 = **1024 subredes** de 2^(32-26) - 2 = 2^6 - 2 = **64 - 2 = 62 interfaces útiles en cada subred**

172.16.58.223 = 172.16.00111010.11011111

Subred: 172.16.58.11000000 = 172.16.58.192 dir. de subred

Broadcast: 172.16.58.11111111 = 172.16.58.255 dir de broadcast

15.Realizar la máxima agregación CIDR (Class Interdomain routing) posible del siguiente conjunto de 4 redes clase C.

Classless Inter-Domain Routing (**CIDR** Encaminamiento Inter-Dominios sin Clases) es una mejora en el modo como se interpretan las direcciones IP. Su introducción permitió una mayor flexibilidad al dividir rangos de direcciones IP en redes separadas, reemplazando la sintaxis previa para nombrar direcciones IP, las **clases** de redes.

CIDR usa la técnica **VLSM** (**Variable-Length Subnet Masking** - Máscara de Subred de Longitud Variable), para hacer posible la asignación de prefijos de longitud arbitraria (la división red/host puede ocurrir en cualquier bit de los 32 que componen la dirección IP).

Un gran beneficio de CIDR es la posibilidad de agregar prefijos de encaminamiento -**supernetting**-. Por ejemplo, dieciséis redes /24 contíguas pueden ser agregadas y publicadas en los routers como una sola ruta /20 (si los primeros 20 bits de sus respectivas redes coinciden). Dos redes /20 contiguas pueden ser agregadas en una /19, etc. Esto permite reducir significativamente el número de rutas que los routers tienen que conocer (reduciendo memoria, recursos, etc.) y previene una explosión de tablas de encaminamiento.

Decimos que una dirección IP **está incluida** en un **bloque CIDR**, y que **encaja** con el prefijo CIDR, si los N bits iniciales de la dirección y el prefijo son iguales. Por tanto, para entender CIDR

es necesario visualizar la dirección IP en binario:

212.56.132.0/24 : 212.56.1000 0100.0
212.56.133.0/24 : 212.56.1000 0101.0
212.56.134.0/24 : 212.56.1000 0110.0
212.56.135.0/24 : 212.56.1000 0111.0

8 + 8 + 6 = 22 bits iguales. Los 2 bits restantes del 3er octeto toman las 2^2=4 combinaciones posibles, así que es posible reducirlas a un solo bloque.

Las cuatro redes clase C pueden agregarse en el bloque CIDR: **212.56.132/22**

16.Listar las redes que involucra el bloque CIDR 200.56.168.0/21.

Se trata de redes clase C (192 < 200 < 223).
200.56.168.0 = 200.56.1010 1000.0
8 + 8 + 5 = 21 bits iguales en el bloque.
El tercer byte queda con 3 bits fuera del bloque: hay 2^3 = 8 redes contenidas en el bloque

#1 200.56.168.0 = 200.56.1010 1000.0
#2 200.56.169.0
#3 200.56.170.0
#4 200.56.171.0
#5 200.56.172.0
#6 200.56.173.0
#7 200.56.174.0
#8 200.56.175.0 = 200.56.1010 1111.0

17. Listar las redes que involucra el bloque CIDR 195.24/13.

Redes clase C (192 < 195 < 223).
195.24 = 1100 0011.0001 1000. 13 bits iniciales iguales en redes de 24 bits de parte de red. 24 - 13 = 11 bits restantes "resumidos", hacen a un total de 2^11 = **2048 redes tipo C resumidas**.
Las redes en el bloque van **desde la 195.24.0.0 hasta la 195.31.255.0**.

18.Una máquina que se conecta a Internet, ¿tiene tabla de ruteo?
Los host también mantienen tablas de ruteo, en el caso más simple conteniendo la red local (la del segmento al que se conecta su interfaz si tiene sólo una tarjeta de red). Si el equipo está conectado a internet, tendrá también una entrada para su gateway por default.

19.El comando netstat presentado en la práctica anterior, al igual que el comando route permite visualizar las tablas de ruteo. Analice su salida.

```
# route
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric
Ref    Use Iface
10.0.2.0        *               255.255.255.0   U     0
0        0 eth0
default         10.0.2.2        0.0.0.0         UG    0
0        0 eth0
```

La primer columna indica *dirección de red* (debe interpretarse de acuerdo a la *máscara* de la tercera columna) y la segunda columna indica el *gateway* -router- a utilizar (0.0.0.0 o * se utiliza para indicar que hay conexión directa a la red en cuestión).

La columna *Flag*s indica las características de la ruta:

U (route is **u**p)
H (target is a **h**ost)
G (use **g**ateway)
R (**r**einstate route for dynamic routing)
D (**d**ynamically installed by daemon or redirect)
M (**m**odified from routing daemon or redirect)
A (installed by **a**ddrconf)
C (**c**ache entry)
! (reject route)

Metric: Valor utilizado para cuantificar la ruta. IP utiliza este valor para seleccionar la mejor de dos o más rutas alternativas a la misma red.

Ref: Cantidad de veces que esta ruta fue utilizada para establecer una conexión.

Use: Cantidad de paquetes trasmitidos a través de esa ruta.

Iface: Interfaz de salida a la ruta.

```
$ netstat -r
Kernel IP routing table
Destination     Gateway         Genmask         Flags   MSS Window  irtt Iface
10.0.2.0        *               255.255.255.0   U         0 0          0 eth0
default         10.0.2.2        0.0.0.0         UG        0 0          0 eth0
```

La salida de netstat agrega las columnas: *MSS* -Maximum Segment Size-, *Window* -tamaño de la ventana de TCP para las conexiones hacia esa ruta- e *irtt* -initial round trip time para conexiones TCP, en milisegundos.

20.Describa qué hacen los comandos ping y traceroute (tracert en windows).

ping -bfr -c *count* -i *interval* -I *interface*|*adress* -s *packetsize* -t *ttl* -w *deadline* -W *timeout* IPDestino

Utiliza el protocolo ICMP para solicitar a un host/gateway una respuesta de *echo*. Envía un datagrama ECHO_REQUEST al IP indicado. Útil para controlar que una interfaz este funcionando o que hay conectividad. Al terminar, muestra algunas estadísticas como cantidad de paquetes perdidos, TTL mínimo/medio/máximo...

-b Permite hacer ping a una dirección de broadcast.
-c *count* Se detiene luego de enviar *count* ECHO_REQUEST.
-f Imprime un . por cada ECHO_REQUEST y un backspace (borra el punto) por cada ECHO_REPLY recibido, mostrando cuantos paquetes se pierden

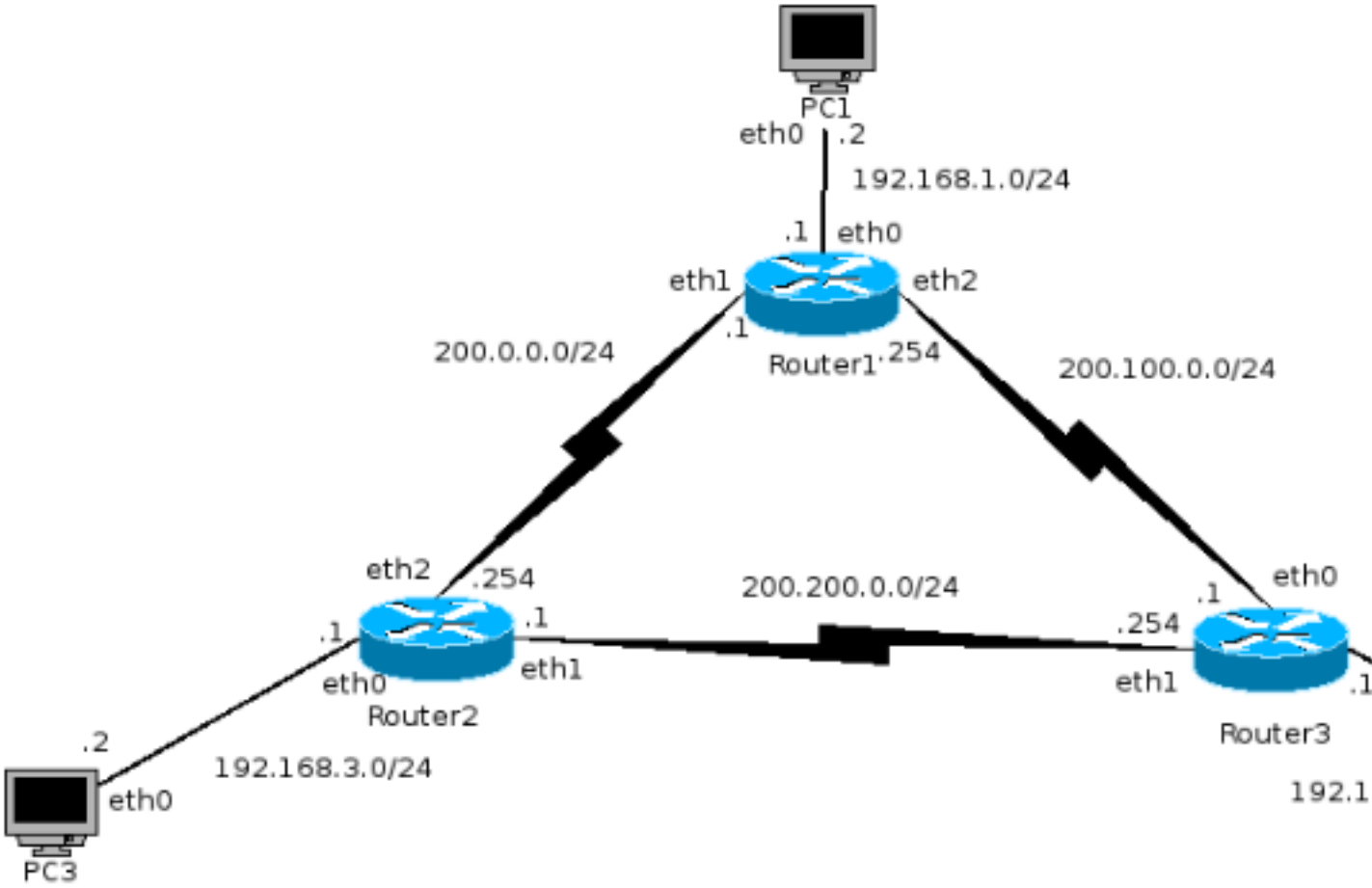
-i *interval* Permite indicar el intervalo a esperar, en segundos, entre cada paquete.
-I *interface|address* Permite indicar la interfaz/IP que se usará como dirección de origen.
-R Incluye la opción Record route en el paquete ECHO_REQUEST y muestra el buffer de ruta de los paquetes (el encabezado IP tiene tamaño suficiente para sólo 9 rutas).
-n Mostrar direcciones númericas, no intentar obtener nombres de host.
-r Permite saltear las tablas de ruteo normal y enviar el paquete a un equipo en una interfaz adjunta.
-s *packetsize* Cantidad de bytes de datos a transmitir.
-t *ttl* Permite indicar el TTL de IP -time to live-.
-w *deadline* Permite indicar un tiempo máximo de funcionamiento.Ping terminará pasado ese tiempo sin importar cuántos paquetes haya enviado o recibido. Si se usa junto a -c, ping terminará al cumplirse el tiempo o al enviar la cantidad indicada de paquetes, lo que ocurra primero.
-W *timeout* Cantidad de segundos queping esperará una respuesta. Por defecto, dos RTTs.

traceroute -4|6 -ITFn -f *firstTTL* -m *maxTTL* -N *squeries* -p *port* -q *nqueries* host
Muestra la ruta que siguen los paquetes por la red hasta llegar a un destino.
Utiliza el campo TTL de IP, enviando paquetes con valores incrementales desde TTL=1, y recibiendo las respuestas ICMP de tipo TIME_EXCEEDED de cada gateway/router en el camino hasta el destino.
Para cada valor de TTL envía tres paquetes, luego imprime el gateway que respondió y el RTT obtenido. Si no obtiene respuesta en 5 segundos (por defecto) para algún paquete, imprime un asterisco. Los paquetes se envían a un puerto UDP que difícilmente se utilice, de modo que al llegar al destino, éste responda un ICMP PORT_UNREACHABLE o un segemento TCP de tipo RESET, denegando la conexión.
-4|6 Forzar el uso de IPv4 o IPv6
-I Utilizar ICMP ECHO para los paquetes de sondeo. Si no se indica -I ni -T, utiliza UDP.
-T Utilizar TCP SYN para los paquetes de sondeo. Si no se indica -I ni -T, utiliza UDP.
-F Marca el bit de "Don't Fragment" para que los routers intermedios no fragementen paquetes mayores al MTU -Maximun Transfer Unit- del enlace.
-f *firstTTL* Indica con qué valor de TTL comenzar (por defecto, 1)
-i *interface* Permite indicar la interfaz a utilizar para los paquetes.
-m *maxTTL* Permite indicar la cantidad maxima de "saltos" que se intentará (por defecto, 30).
-N *squeries* Cantidad de paquetes a enviar simultáneamente (por defecto, 16).
-n Mostrar direcciones númericas, no intentar obtener nombres de host.
-p *port* Para tracing con UDP/TCP indica el número de puerto destino a utilizar. Para ICMP, el número de secuencia inicial.
-q *nqueries* Cantidad de paquetes para cada "salto" posible (cada valor TTL). Por defecto, 3.

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

127.0.0.1/8 es la dirección de **retroalimentación** o **loopback** de todo equipo con placa *Ethernet*. Hace referencia al mismo host que envía el paquete, de modo que éste es procesado por los protocolos del stack TCP/IP pero no llega a salir nunca por la interfaz, sólo se maneja en memoria. Si bien lo más frecuente es utilizar 127.0.0.1, toda la gama 127.x.x.x tiene el mismo efecto, ya que se encuentra reservada para igual propósito.
Se puede usar para probar el funcionamiento de TCP/IP: al hacer **ping 127.0.0.1**, los paquetes se manejan localmente, por lo que responde el mismo host. Así puede asumirse que los componentes asociados al protocolo están bien -sería un primer paso para aislar problemas de red, por ejemplo-.
El equivalente en IPv6 es **::1/128**.

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



1. **Abra una consola como usuario *lihuen* 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 corresponde con el nombre 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 192.168.3.2, en Router2 eth0 con la IP 192.168.3.1, eth1 con 200.200.0.1 y eth2 con 200.0.0.254**
    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***

Comando ifconfig.

```
ifconfig [-v] [-a] [-s] [interface]
ifconfig [-v] interface [aftype] options | address ...
```

Permite configurar una tarjeta de red. Si no se le proporcionan parámetros, muestra el estado de las interfaces actualmente *activas*. Si sólo interesa una interfaz, puede indicarse su nombre. Puede indicarse la familia de protocolos a utilizar para una interfaz con el parámetro **aftype** (**inet**: TCP/IP -default-, **inet6**: IPv6, **ipx**: NOVEL IPX...).

-a Muestra información de todas las interfaces -incluso inactivas-.
-s Lista reducida (igual a **netstat -i**).
-v Verbose. Incrementa el nivel de detalle de la salida del programa.
interface Nombre de una interfaz, usualmente el driver seguido del número de unidad, como eth0. Si el kernel soporta alias se indican como eth0:0 para el primer alias de eth0 (esto permite asignar una segunda dirección). Puede eliminarse un alias con: **ifconfig eth0:0 down** (si se eliminan el primario, se eliminan también todos los alias).
up|down Activa/desactiva la interfaz
[-]arp Habilita/Deshabilita el uso del protocoloARP en la interfaz
[-]promisc Habilita/Deshabilita el modo *promiscuo* para la interfaz (permite que la interfaz reciba todos los paquetes en la red, no sólo los que la tienen por destino)
[-]allmulti Habilita/Deshabilita el modo all-multicast (permite que la interfaz reciba todos los paquetes multicast en la red)
metric N Setea la métrica utilizada por la interfaz
mtu N Setea el Maximum Transfer Unit (MTU) de la interfaz
netmask addr Setea la máscara de red (por defecto utiliza máscara A, B o C según la dirección IP, pero puede indicarse cualquier valor)
add|del addr/prefixlen Agrega/elimina una dirección IPv6 a/de la interfaz
tunnel aa.bb.cc.dd Crea un nuevo dispositivo SIT (IPv6-in-IPv4) hacia el destino indicado.
[-]broadcast [addr] Setea la dirección de broadcast para la interfaz. Si no se indica la dirección setea/quita el flag IFF_BROADCAST para la interfaz.
[-]pointopoint [addr] Habilita/Deshabilita el modo point-to-point para la interfaz (es un link directo entre dos máquinas sin nadie más escuchando entre ellas). Si se indica la dirección considera que es la que está al otro lado, sino setea/quita el flag IFF_POINTTOPOINT para la interfaz.
hw *class address* Setea la dirección de hardware para la interfaz. Se debe indicar el nombre de la clase (ether para Ethernet) de hardware y el equivalente ASCII de la dirección.
multicast Marca el flag *multicast* en la interfaz. Normalmente no se usa -lo maneja el driver-.
address La dirección IP a asignar a la interfaz.

```
pc1:~# ifconfig eth0 192.168.1.2 netmask 255.255.255.0      #no es precisa la
mascara porque 192... ya es C
router1:~# ifconfig eth0 192.168.1.1 netmask 255.255.255.0
router1:~# ping 192.168.1.2                                  #devuelve 0% packet loss
```

```
pc2:~# ifconfig eth0 192.168.2.2
router3:~# ifconfig eth2 192.168.2.1
router3:~# ping 192.168.2.2                                  #devuelve 0% packet loss
```

```
pc3:~# ifconfig eth0 192.168.3.2
router2:~# ifconfig eth0 192.168.3.1

router1:~# ifconfig eth1 200.0.0.1
router1:~# ifconfig eth2 200.100.0.254
router2:~# ifconfig eth1 200.200.0.1
router2:~# ifconfig eth2 200.0.0.254
router3:~# ifconfig eth0 200.100.0.1
router3:~# ifconfig eth1 200.200.0.254
```

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 trádico desconocido a Router1**

Comando route.

Manipula/muestra las tablas IP de ruteo estático, según se utilice o no con las opciones add/del.

-n Utilizar direcciones numéricas (no intenta convertirlas a nombres simbólicos)

-e|**ee** Controla nivel de detalle de la salida, con un formato como el de netstat

del|**add** Elimina/Agrega una ruta

target Red/Host destino (dirección IP o nombre del host)

-net|**host** Indica que el target es un/a red/host

netmask ***NM*** Indica la máscara de red a utilizar si se agrega una ruta de red

gw ***GW*** Permite rutear los paquetes vía el gateway indicado. Debe ser alcanzable (por ejemplo por otra ruta estática). Si se indica el nombre de una interfaz local, se la usará para decidir hacia dónde se deberán rutear los paquetes.

reject Instala una ruta de bloqueo, por lo que fallarán los búsquedas.

dev ***If*** Fuerza que la ruta se asocie a una interfaz, ya que de otro modo el kernel intentará determinar solo por qué interfaz enviar los paquetes.

[If] ***dev*** Indica la interfaz que se configura. Es la última opción.

```
router1:~# route add default gw 200.0.0.254 eth1
router2:~# route add default gw 200.200.0.254 eth1
router3:~# route add default gw 200.100.0.254 eth0
pc1:~# route add default gw 192.168.1.1 eth0
pc2:~# route add default gw 192.168.2.1 eth0
pc3:~# route add default gw 192.168.3.1 eth0
```

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 una red diferente a la IP configurada en 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
```

```
router1:~# cat /proc/sys/net/ipv4/ip_forward
0
router1:~# echo 1 > /proc/sys/net/ipv4/ip_forward
router1:~# cat /proc/sys/net/ipv4/conf/all/rp_filter
0

router2:~# cat /proc/sys/net/ipv4/ip_forward
0
router2:~# echo 1 > /proc/sys/net/ipv4/ip_forward
router2:~# cat /proc/sys/net/ipv4/conf/all/rp_filter
0

router3:~# cat /proc/sys/net/ipv4/ip_forward
0
router3:~# echo 1 > /proc/sys/net/ipv4/ip_forward
router3:~# cat /proc/sys/net/ipv4/conf/all/rp_filter
0
```

6. Verifique conectividad entre PC1, PC2 y PC3:

1. Utilizando el comando *ping*

```
pc1:~# ping 192.168.2.2
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=117 ms
64 bytes from 192.168.2.2: icmp_seq=2 ttl=62 time=0.336 ms
...
```

2. Utilizando el comando *traceroute*

```
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)   0.160 ms   0.445 ms  0.115 ms
2  200.200.0.1 (200.200.0.1)   0.263 ms   0.405 ms  0.200 ms
3  200.100.0.1 (200.100.0.1)   0.205 ms   0.409 ms  0.181 ms
4  192.168.2.2 (192.168.2.2)   0.342 ms   0.229 ms  0.200 ms
```

3. Utilizando el comando *ping -nR*

```
pc1:~# ping -nR 192.168.2.2
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=0.425 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
     192.168.2.2
64 bytes from 192.168.2.2: icmp_seq=2 ttl=62 time=0.291 ms
(same route)
...
```

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:

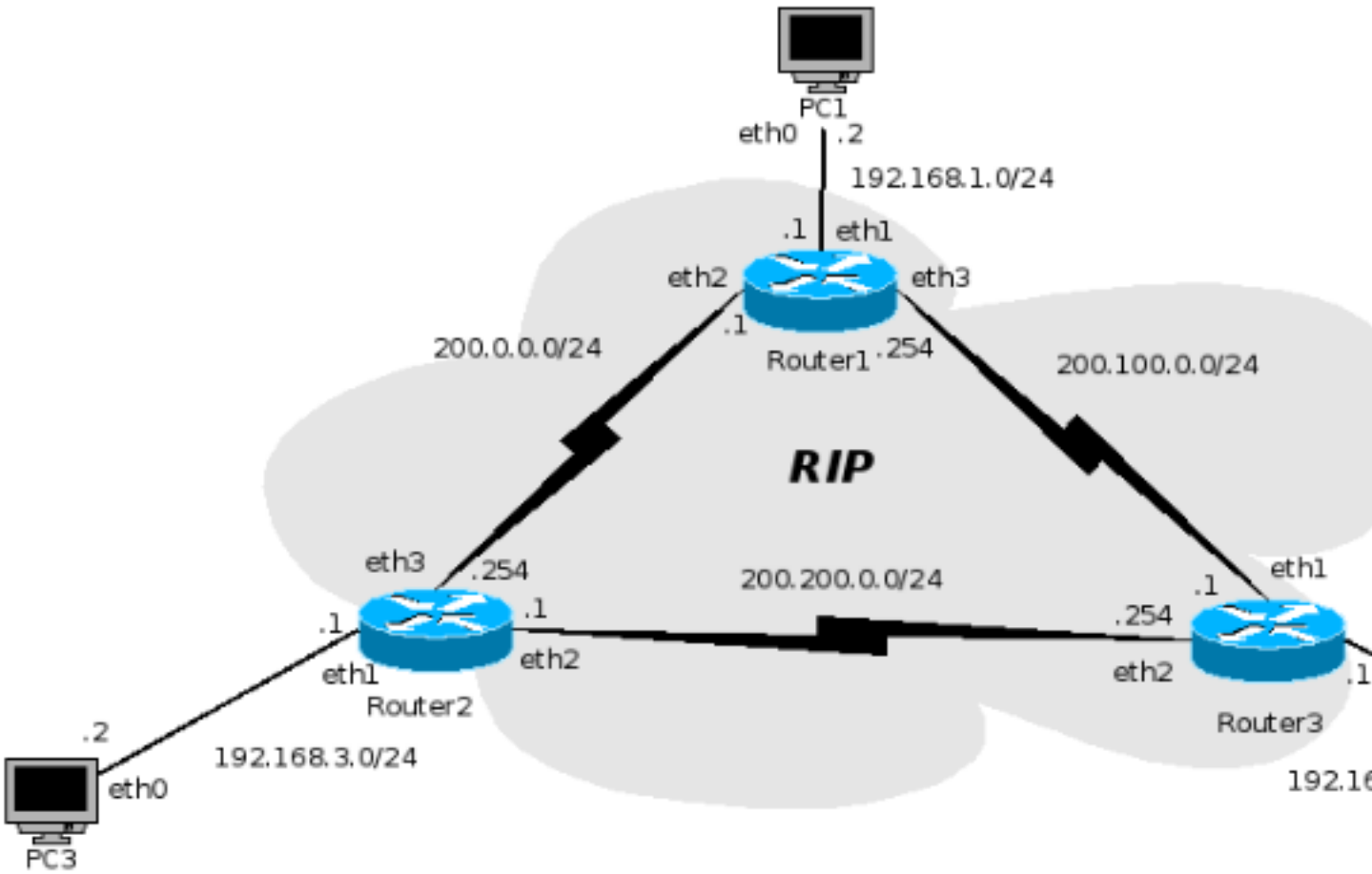
```
tcpdump -i eth0
```

Router 3	**PC1**
# (tcpdump -vv -i eth0 > eth0.pcap) & [1] 1223 **# (tcpdump -vv -i eth1 > eth1.pcap) &** [2] 1226 **# kill 1223** **# kill 1226** **# cat eth1.pcap** .... 192.168.1.2 > 192.168.2.2: ICMP echo **request**... **# cat eth0.pcap** .... 192.168.2.2 > 192.168.1.2: ICMP echo **reply**...	 **# ping 192.168.2.2** 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=117 ms ... ^C

5. En base al punto anterior, ¿qué puede deducir?

...

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



1. **Abra una consola como usuario *lihuen* y ejecute el comando:**

```
topologia capa-red-dinamico start
```

2. **Aguarde a que la consola devuelva el prompt y 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 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. **Desde una de las PC compruebe la ruta que siguen los paquetes al resto de las PC. Para verificar las rutas, puede utilizar el comando ping o traceroute.**

```
pc1:~# ping -nR 192.168.2.2
... RR: 192.168.1.2 / 200.100.0.254 / 192.168.2.1 / 192.168.2.2...
pc1:~# ping -nR 192.168.3.2
... RR: 192.168.1.2 / 200.0.0.1 / 192.168.3.1 / 192.168.3.2...
```

La ruta de PC1->PC2 cambió desde el ejercicio anterior, usando ahora un camino más directo (PC1->Router1->Router3->PC2).

5. Dada la ruta obtenida 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 la ruta desde PC1 a PC3 pasa por la red 200.0.0.0/24. Luego, debe dar de baja la interfaz eth2 de Router1 (`ifconfig eth1 down`) y la interfaz eth3 de Router2 (`ifconfig eth3 down`).

```
router1:~# ifconfig eth3 down
router3:~# ifconfig eth1 down
```

6. Desde la misma PC del punto 4, vuelva comprobar la ruta que siguen los paquetes al resto de las PC.

```
pc1:~# ping -nR 192.168.2.2
... RR: 192.168.1.2 / 200.0.0.1 / 200.200.0.1 / 192.168.2.1 / 192.168.2.2...
pc1:~# ping -nR 192.168.3.2
... RR: 192.168.1.2 / 200.0.0.1 / 192.168.3.1 / 192.168.3.2...
```

Ejercicios Evaluables

1. Dada una dirección IP y la máscara correspondiente, deberá saber responder:
- **De qué tipo de dirección se trata (A, B o C).**
- **Cuál es la dirección de subred.**
- **Cuál es la cantidad máxima de host que pueden estar en esa red.**
- **Cuál es la dirección de broadcast.**
- **Cuál es el rango de hosts válidos dentro de la red.**

2. Con el ejercicio 22 resuelto (ruteo estático), copie el gráfico de la topología de red e indique la tabla de ruteo de cada dispositivo de red, tanto routers como PCs de usuarios. Para la confección de la tabla de enrutamiento puede usar el formato del comando "route -n" o "netstat -nr", aunque alcanza con especificar los siguientes campos: Destino, Gateway, Mascara, Interface.

Sobre este gráfico se le realizarán preguntas de comprensión sobre el tema ruteo, como por ejemplo:

Si la PC1 le envia un ping a la estación PC2, ¿cúal es el camino por el que viaja el requerimiento?, ¿cúal es el camino por el que viaja la respuesta?

El ayudante cambiará alguna de las tablas de ruteo y usted tendrá que evaluar como afecta el cambio realizado.