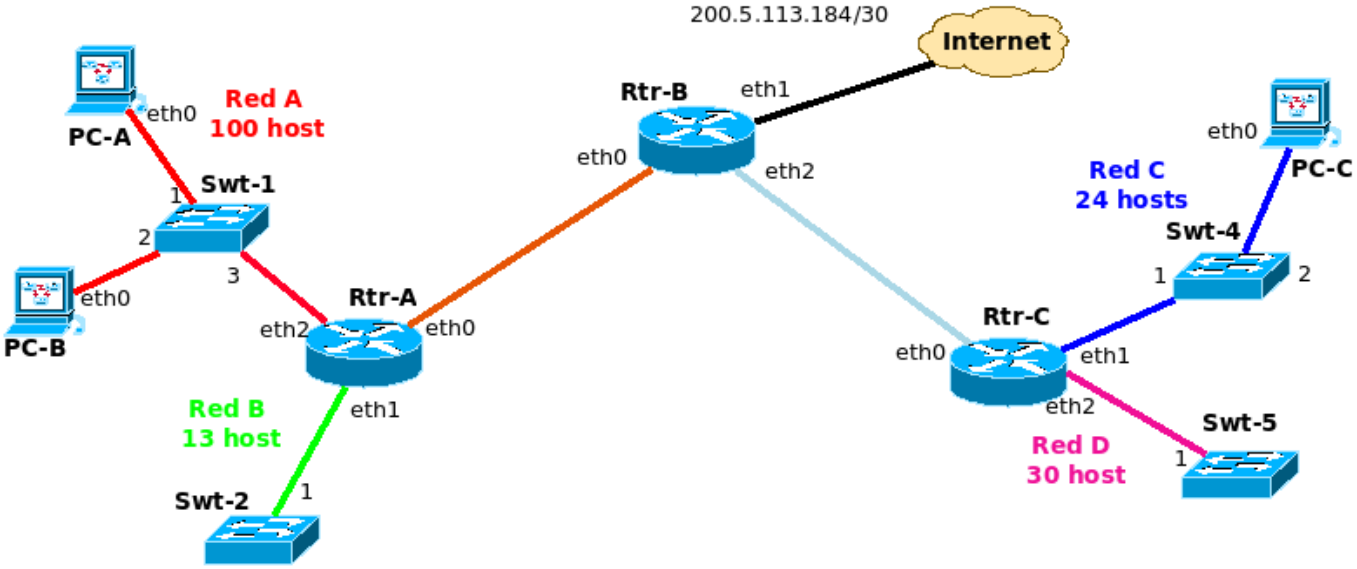
VLSM - Ruteo Estático - ARP



1) **Tenemos la red 192.168.10.0/24 para subnetear. ¿Es posible cubrir todas las redes con subnetting de máscara de longitud fija?**

    a) **Tomando la red de mayor tamaño, Red A.**
        i) **No sirve. Para cubrir Red A necesito un /25 ya que tiene 100 hosts. Esto provoca que queden 7 bits para la parte de hosts, pero al tomar un solo bit para hacer subnetting solo podemos cubrir 2 redes y, claramente, no nos alcanza para cubrir el total de redes que tenemos, que son 6 (no tenemos en cuenta la que va de Rtr-B a Internet)**

    b) **Tomando la cantidad de redes**
        i) **Tampoco sirve. Para cubrir 6 redes se necesitan 3 bits que debemos tomar de la parte de hosts (el último byte). Pero, al tomar 3 bits para sumar a la parte de red, quedan 5 bits para la parte de hosts lo que nos da que se pueden cubrir redes de hasta un tamaño de 30 hosts. Como se observa,**

en este caso es posible cubrir una mayor cantidad de redes, hasta 8 redes, pero cada una de menor tamaño.

2) ¿Qué deberíamos aplicar?

Aplicando VLSM:

192.168.10.0/24 la dividimos en dos:
Último byte: 00000000 -> 192.168.10.0/25 (Red A)
10000000 -> 192.168.10.128/25 (Libre, para seguir subnetenado)

192.168.10.128/25
Último byte: 10000000 -> 192.168.10.128/26 (62 direcciones disponibles, puedo seguir subneteando)
11000000 -> 192.168.10.192/26 (62 direcciones disponibles, puedo seguir subneteando)

192.168.10.128/26
Último byte: 10000000 -> 192.168.10.128/27 (Red C)
10100000 -> 192.168.10.160/27 (Red D)

192.168.10.192/26
Último byte: 11000000 -> 192.168.10.192/27 (30 direcciones disponibles, puedo seguir subneteando)
11100000 -> 192.168.10.224/27 (30 direcciones disponibles, puedo seguir subneteando) - Libre

192.168.10.192/27
Último byte: 11000000 -> 192.168.10.192/28 (Red B)
11010000 -> 192.168.10.208/28

192.168.10.208/28

Último byte: 11010000 -> 192.168.10.208/29

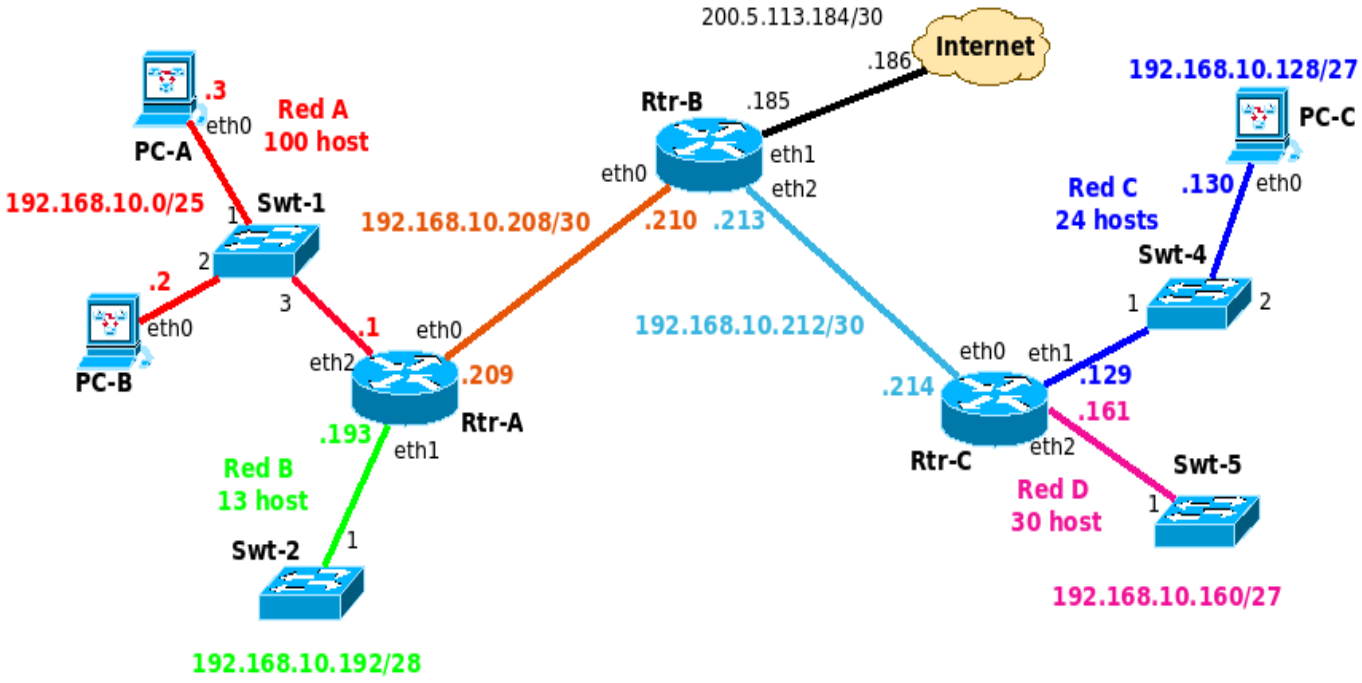
11011000 -> 192.168.10.216/29 (Libre)

192.168.10.208/29

Último byte: 11010000 -> 192.168.10.208/30 (Enlace entre RtrA-RtrB)

11010100 -> 192.168.10.212/30 (Enlace entre RtrB-RtrC)

--- Las direcciones IPs que dicen Libre pueden ser utilizadas en futuras redes



Rtr-A:

Red Destino	Máscara	Next-Hop	Iface
192.168.10.0	/25	-	eth2
192.168.10.192	/28	-	eth1
192.168.10.208	/30	-	eth0
192.168.10.212	/30	192.168.10.210	eth0
192.168.10.128	/27	192.168.10.210	eth0
192.168.10.160	/27	192.160.10.210	eth0
0.0.0.0	/0	192.168.10.210	eth0

Como las últimas 4 redes tienen el mismo gateway, o next-hop, se las puede reemplazar por la ruta default y, de esta manera, achicar el tamaño de la tabla de ruteo:

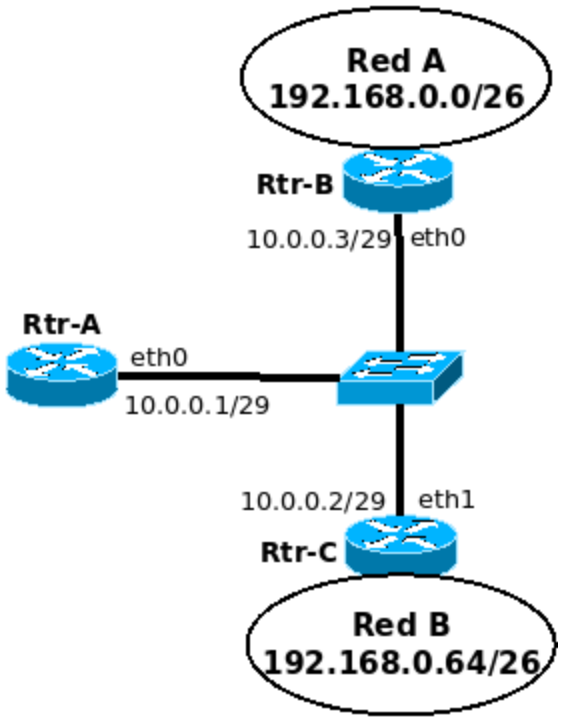
Red Destino	Máscara	Next-Hop	Iface
192.168.10.0	/25	-	eth2
192.168.10.192	/28	-	eth1
192.168.10.208	/30	-	eth0
0.0.0.0	/0	192.168.10.210	eth0

Rtr-B

Red Destino	Máscara	Next-Hop	Iface
192.168.10.208	/30	-	eth0
192.168.10.212	/30	-	eth2
200.5.113.184	/30	-	eth1
192.168.10.0	/25	192.168.10.209	eth0
192.168.10.192	/28	192.168.10.209	eth0
192.168.10.128	/27	192.168.10.214	eth2
192.168.10.160	/27	192.168.10.214	eth2
0.0.0.0	/0	200.5.113.186	eth1

¿Podemos sumarizar las redes en rojo?

Al momento de sumarizar es importante tener en cuenta que los valores de los campos Next-Hop e Interface deben ser iguales. Si esto no sucede, no debería aplicarse sumarización por más que las redes sean posibles de sumarizar. Podría suceder que la interface de salida sea la misma pero no el Next-Hop. Esto último lo podemos ver en el siguiente gráfico en el cual, si bien las dos redes son sumarizables en una red 192.168.0.0/25, Rtr-A no podrá sumarizarlas debido a que no se cumplen todas las condiciones para poder hacerlo. En este caso, si bien es igual la interface de salida de Rtr-A, eth0, para llegar a las dos redes no son iguales los Next-Hop. Ante esta situación no queda otra solución que agregar una entrada para cada red en la tabla de ruteo.



Siguiendo con la pregunta vemos que están dadas todas las condiciones para que esas dos redes sean sumarizadas.

192.168.10.128 -> 11000000.10101000.00001010.10|000000
192.168.10.160 -> 11000000.10101000.00001010.10|100000

Esta sumarización también se podría haber aplicado en RtrA, pero en nuestro caso decidimos reemplazar todo por la ruta default!!

Hasta el | son iguales -> 192.168.10.128/26

Red Destino	Máscara	Next-Hop	Iface
192.168.10.208	/30	-	eth0
192.168.10.212	/30	-	eth2
200.5.113.184	/30	-	eth1
192.168.10.0	/25	192.168.10.209	eth0
192.168.10.192	/28	192.168.10.209	eth0
192.168.10.128	/26	192.168.10.214	eth2
0.0.0.0	/0	200.5.113.186	eth1

Tarea:

- **Realizar la tabla de ruteo de RtrC**

Pregunta:

- **Viendo que RtrC conoce las dos redes, 192.168.10.128/27 y 192.168.10.160/27, es posible sumarizarlas como hicimos en RtrB?**

La respuesta es que no es posible ya que RtrC las tiene a ambas redes directamente conectadas y el tiene que discriminar claramente donde se encuentra cada red para poder entregar los paquetes de manera correcta.

Si Rtr-B recibe dos paquetes que tienen como IP destino 192.168.10.140 y 192.168.10.165, a él no le interesa saber si corresponden a distintas redes, solo sabe que a esos paquetes se los debe reenviar a Rtr-C de acuerdo a lo que dice la tabla de ruteo. Pero cuando esos paquetes lleguen a Rtr-C, éste tiene que saber que el paquete con IP destino 192.168.10.140 debe enviarlo por la eth1 ya que va a RedC y, en cambio, el paquete que tiene la IP Destino 192.168.10.165 lo debe reenviar por la eth2 ya que corresponde a la RedD.

ARP (Address Resolution Protocol)

2 Casos:

a) **Origen y Destino de la conexión en la misma red**
b) **Origen y Destino de la conexión en distintas redes**

i) Suponemos que en ambos casos las tablas ARP están vacías!!!

Caso a)

Supongamos que en Red A, PC-A le hace un ping a PC-B

pc-a# ping 192.168.10.2

En base a esto podemos armar el paquete IP:

IP Origen	IP Destino	ICMP Packet
192.168.10.3	192.168.10.2	Echo Request

Cuando este paquete IP pase a la capa inferior, capa 2, que en nuestro caso es Ethernet, es necesario armar la trama Ethernet que, entre otra información, agregará las direcciones MAC origen y destino.

Teniendo en cuenta lo indicado en i), ¿es PC-A capaz de armar la trama Ethernet? (tener en cuenta que en el gráfico no están todos los campos de las headers de Ethernet y/o IP y tampoco el trailer de Ethernet)

MAC Destino	MAC Origen	IP Origen	IP Destino	ICMP Packet
?????	PC-A_Eth0	192.168.10.3	192.168.10.2	Echo Request

En este punto, PC-A conoce sus propias direcciones, tanto IP como MAC, y la dirección IP del destino (en este caso está indicada en el comando ping), por lo tanto, para terminar de formar la trama Ethernet debe averiguar la dirección MAC del destino. Es en este momento donde se recurre a ARP para que solucione este problema.

Como PC-A y PC-B están en la misma red (esto lo deduce PC-A mirando su propia tabla de ruteo), PC-A enviará un paquete ARP Request preguntando cuál es la dirección MAC asociada a la IP 192.168.10.2 (que es la IP de PC-B).

ARP Request:

MAC Destino	MAC Origen	ARP Packet
FF:FF:FF:FF:FF:FF	PC-A_Eth0	IP Src: 192.168.10.3 IP Dst: 192.168.10.2 MAC Src: PC-A_Eth0 MAC Dst: ??????

Tener en cuenta:

1) El paquete ARP, lo que está en rojo, se encapsula dentro de una trama Ethernet. Las 2 direcciones MAC en azul corresponden a la trama Ethernet. Recordar que faltan campos en la cabecera y el correspondiente trailer.
2) MAC Destino en la trama Ethernet es la dirección broadcast. El paquete es recibido por todos los dispositivos dentro del dominio de broadcast, inclusive la interface del router que está conectada a esa red, que en este caso es la eth2 de RtrA.

3) **MAC Destino en el paquete ARP (a las direcciones destino la RFC 826 que define ARP las llama Target Address) se deja en blanco o todos 0s (o indicarla con ???), pero no poner FF:FF:FF:FF:FF:FF (ésta es la dirección de broadcast)**
4) **El router no retransmite los mensajes de tipo broadcast por defecto, por lo cual, el ARP Request no será retransmitido por Rtr-A a ninguna de sus otras redes conectadas. Esto implica que no es posible utilizar ARP para determinar la dirección MAC de un dispositivo que se encuentra en otra red!!!!!**

Obviamente, aunque todos los dispositivos van a recibir y procesar el paquete ARP, el único que va a contestar es el que tenga asignada la dirección IP 192.168.10.2 (que es la dirección IP indicada como IP Destino (o Target) en el paquete ARP), que en nuestro caso es PC-B. Los demás descartarán silenciosamente el ARP Request recibido.

PC-B le responderá con un ARP Reply:

ARP Reply:

MAC Destino	MAC Origen	ARP Packet
PC-A_Eth0	PC-B_Eth0	IP Src: 192.168.10.2 IP Dst: 192.168.10.3 MAC Src: PC-B_Eth0 MAC Dst: PC-A_Eth0

1) **El ARP Reply, a diferencia del ARP Request que es broadcast, es de tipo unicast**
2) **En el paquete ARP se invierten los valores del origen/destino**
3) **PC-B, en nuestro caso, almacenará en su tabla ARP los valores de PC-A (es altamente probable que cuando PC-B reciba el paquete de PC-A le tenga que responder y con**

esto se evita de tener que volver a ejecutar el proceso ARP nuevamente)

Con toda la información disponible ahora sí podemos armar la trama Ethernet y enviarla al destino.

MAC Destino	MAC Origen	IP Origen	IP Destino	ICMP Packet
PC-B_Eth0	PC-A_Eth0	192.168.10.3	192.168.10.2	Echo Request

Caso b)

Supongamos que PC-A, en la Red-A, le hace un ping a PC-C en la Red-C

pc-a# ping 192.168.10.130

En base a esto podemos armar el paquete IP:

IP Origen	IP Destino	ICMP Packet
192.168.10.3	192.168.10.130	Echo Request

Como se indicó en el punto anterior este paquete IP deberá pasar a la capa inferior, Ethernet, para que se pueda formar la trama y, obviamente nos encontramos con el mismo problema, no tenemos la MAC Destino para incluir en la trama

MAC Destino	MAC Origen	IP Origen	IP Destino	ICMP Packet
?????	PC-A_Eth0	192.168.10.3	192.168.10.130	Echo Request

Pero, de acuerdo a lo que dijimos en el punto 4) del ARP Request, PC-A no podría enviar un ARP Request de tipo broadcast para averiguar la dirección MAC de PC-C. Rtr-A no dejaría pasar ese paquete broadcast y, por lo tanto, la consulta nunca llegaría a PC-C.

Para solucionar esto, lo que va a hacer PC-A, mediante ARP, es averiguar la dirección MAC de su default-gateway, es decir, la MAC del dispositivo que le permitirá salir de su propia red. Como PC-A no le puede enviar el paquete directamente a PC-C (como sí pasaba en el caso anterior con PC-B) lo que va a hacer es pasarlo a un dispositivo intermedio que se encargará de reenviarlo hacia el destino. En base a esto, el ARP Request que envíe PC-A no consultará por la MAC asociada a la IP de PC-C sino por la MAC asociada a la IP de su default-gateway (tener en cuenta que PC-A ya tiene esta información configurada, o manualmente o por DHCP). En nuestro ejemplo, el default-gateway de PC-A es la IP que tiene asignada el router Rtr-A en su interface eth2, 192.168.10.1.

En base a lo dicho el ARP Request enviado por PC-A será el siguiente:

ARP Request:

MAC Destino	MAC Origen	ARP Packet
FF:FF:FF:FF:FF:FF	PC-A_Eth0	IP Src: 192.168.10.3 IP Dst: 192.168.10.1 MAC Src: PC-A_Eth0 MAC Dst: ??????

Y el ARP Reply quedará de la siguiente manera:

ARP Reply:

MAC Destino	MAC Origen	ARP Packet
PC-A_Eth0	Rtr-A_Eth2	IP Src: 192.168.10.1 IP Dst: 192.168.10.3 MAC Src: Rtr-A_Eth2 MAC Dst: PC-A_Eth0

Con esta información ahora PC-A es capaz de completar la trama para enviarla:

MAC Destino	MAC Origen	IP Origen	IP Destino	ICMP Packet
Rtr-A_Eth2	PC-A_Eth0	192.168.10.3	192.168.10.130	Echo Request

Una vez que la trama llegue a Rtr-A por la interface Eth2, el router controla que esté correcta y, si es así, desarma la trama y pasa el paquete IP a la capa de red (proceso de desencapsulamiento) para que el router haga su trabajo: rutear. Usando la IP destino del paquete y comparándola contra su tabla de ruteo decide por cuál interface debe reenviar el paquete. Como la dirección IP destino es 192.168.10.130 al compararla contra su tabla de ruteo, Rtr-A encuentra que existe un matching con la entrada a la red 192.168.10.128/27 y que el next-hop es la IP 192.168.10.209, la IP de la interface Eth0 de Rtr-B (esto si miramos la primer tabla de ruteo en la que están todas las redes especificadas, en la segunda tabla el match sería con la ruta default).

En este ejemplo debe enviarla por la Eth0 por lo que baja el paquete a la capa 2, nuevamente Ethernet (proceso de encapsulamiento), y debe iniciarse el proceso de ARP Request/ARP Reply nuevamente. Pero ahora las MACs que nos interesan son las asociadas a la interface Eth0 de RtrA y

Eth0 de RtrB. La trama ahora debe viajar de Rtr-A a Rtr-B por lo que los paquetes ARP Request y ARP Reply quedarían de la siguiente manera:

ARP Request:

MAC Destino	MAC Origen	ARP Packet
FF:FF:FF:FF:FF:FF	Rtr-A_Eth0	IP Src: 192.168.10.209 IP Dst: 192.168.10.210 MAC Src: Rtr-A_Eth0 MAC Dst: ??????

Y el ARP Reply quedará de la siguiente manera:

ARP Reply:

MAC Destino	MAC Origen	ARP Packet
Rtr-A_Eth0	Rtr-B_Eth0	IP Src: 192.168.10.210 IP Dst: 192.168.10.209 MAC Src: Rtr-B_Eth0 MAC Dst: Rtr-A_Eth0

Con esta información ahora Rtr-A es capaz de completar la trama para reenviarla:

MAC Destino	MAC Origen	IP Origen	IP Destino	ICMP Packet
Rtr-B_Eth0	Rtr-A_Eth0	192.168.10.3	192.168.10.130	Echo Request

Como se puede observar, y comparándolo con el paquete que viajó en la Red A entre PC-A y Rtr-A, las direcciones IP se mantienen iguales pero si se modificaron las direcciones MAC.

Este proceso se irá repitiendo por cada una de las redes por la que pase el paquete hasta llegar al destino. En cada nueva red por donde pase

el paquete las direcciones MACs se modificarán y no las direcciones IPs (salvo que se aplique algún proceso tipo NAT).

Tener en cuenta que si algún dispositivo ya conoce la MAC Destino no es necesario ejecutar ARP!! Además, como todos los dispositivos fueron aprendiendo las MACs de sus vecinos, cuando PC-C le responda a PC-A no será necesario ejecutar el proceso ARP nuevamente.

Los dispositivos mantienen las entradas en su tabla ARP por un tiempo determinado y lo van refrescando cada vez que esa entrada es utilizada. Si ese tiempo se vence sin que la entrada sea utilizada, se la elimina de la tabla. En caso de volverse a necesitar, el proceso ARP deberá ser ejecutado nuevamente!