

Subredes.

Bibliografía Utilizada

IP Fundamentals; T.Maufer; Prentice Hall PTR

PARTE 1 CLÁSICO

La **RFC 950*** define los procedimientos estándar de división de las clases A,B y C en partes mas pequeñas llamadas subredes.

Cada red necesita su propia dirección IP lo que lleva a un rápido consumo de las direcciones disponibles.

Dada la cantidad fija de bit que la dirección IP tiene (32), se debe recurrir a algún método que obtenga bits "prestados" y de donde los obtiene es de la parte de Host.

** Como ejercicio interesante resulta interesante leer la RFC 950*

Alternativas La solución que se implemente deberá ser compatible con lo que se tenía hasta ese momento

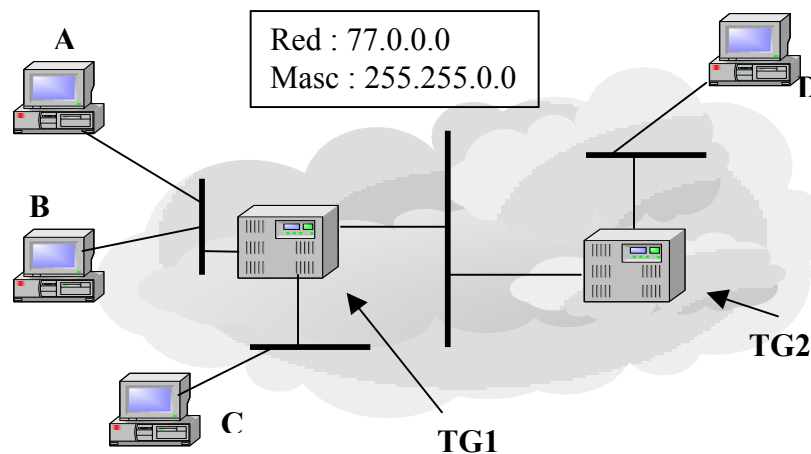
Subredes implícitas Este método es de lectura optativa

Veremos algunas ideas que ya son antiguas pero que tienen interés por representar alternativas que fueron usadas en su momento aunque ya eclipsadas por el método explícito.

1) Método indicado en la RFC - 925

Se tienen aquí "cajas" (TG1, TG2) que participan en un ARP mas allá de las LAN.

A: 77.8.13.99
B: 77.56.3.181
C: 77.0.1.4
D: 77.8.197.6



Este es un procedimiento sencillo que permite que las máquinas direccionen de acuerdo a las reglas normales de IP que dicen que una estación debe ARP'ar otra estación en su misma red y usar el default Gateway para salir de su red.

Dado que la máscara es 255.0.0.0 se hace ARP para cualquier máquina dentro de la red 77. Este permite a una Empresa usar completamente su dirección antes de necesitar otra.

Ejemplo 1

Supongamos que la estación A (77.8.13.99) desea comunicarse con la estación B (77.56.3.181) de su misma red.

1. "A" verifica direcciones y máscara, notando que se encuentra en la misma red que el destino
2. "A" envía un ARP consultando por "B"
3. "B" responde al ARP
4. TG1 cachea toda la info que escucha.

Ejemplo 2

Supongamos que "B" desea comunicarse con "C"

1. "B" Genera un ARP consultando por "C"
2. TG1 debe hacer creer a "B" que "C" esta en su misma red. (Supongamos que TG1 tiene ya cacheadas todas las direcciones)
 - a. Arma un nuevo paquete Broadcast reemplazando la MAC de "B" por su propia MAC. La IP destino continua siendo la de "C" y la IP origen continua siendo la de "B"
 - b. La estación "C" Cachea la info recibida y piensa que la MAC que le llegó es la de "B" aunque realmente es la del TG1
 - c. Simultáneamente TG1 responde a "B" como si fuera "C" pero con la IP origen de "C"
 - d. Todo paquete futuro entre "A" y "B" es direccionado a la MAC de TG1
 - e. El único cambio que TG1 hace en el paquete IP es reducir el TTL en 1
3. En caso que TG1 no tuviera cacheada las direcciones pondría un broadcast en todas las salidas menos en la que dio origen al paquete.

Ejemplo 3

Supongamos que la estación "D" desea comunicarse con "A"

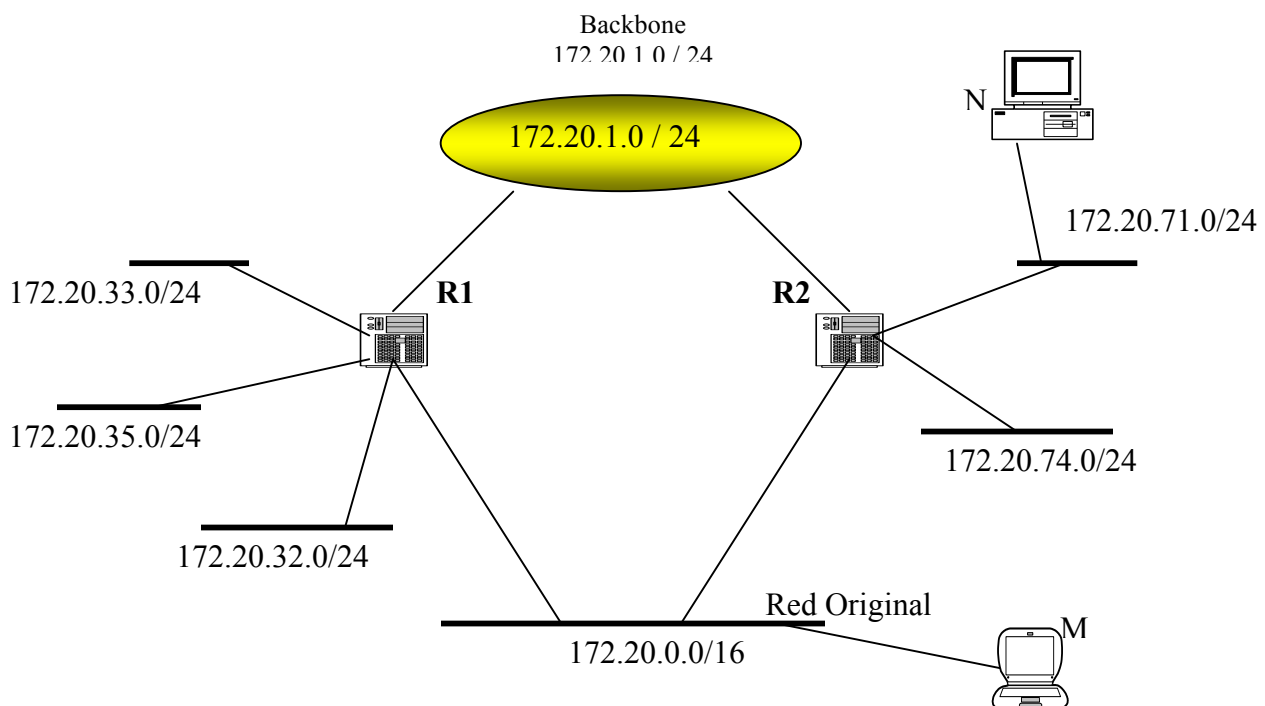
1. "D" genera un ARP
2. TG2 lo pasara a la otra interface y lo recibirá TG1 (lo cachea)
3. TG1 conoce donde está "A" y le envia el ARP (modificado)
4. TG1 responde a TG2
5. TG2 responde a "D"

2) Método Proxi ARP

Este es un método que aún se utiliza, especialmente cuando se está en medio de una conversión a subredes.

Considere que una red se está "subnetando". Durante algún tiempo parte de la red pertenecerá a una subred (con la máscara que le corresponda) y parte de la red aún estará en la red original (con la máscara original)

Ejemplo



En la red original es donde aún permanecen la mayoría de las máquinas, para estas máquinas la red no está subnetada (notar /16). Los routers R1 y R2 aíslan las subredes de la red original y permiten que los host de las subredes se comuniquen entre sí por el backbone de alta velocidad.

Supongamos que la máquina "M" ($172.20.97.101 / 16$) quiere comunicarse con la máquina "N" ($172.20.71.76 / 24$).

En virtud de su máscara, "M" considera que "N" está en su misma subred, por lo cual generará un ARP. Normalmente los routers no permiten el paso de los broadcast por lo cual "N" no se enteraría nunca que "M" desea comunicarse.

Sin embargo R2 está recibiendo un pedido de ARP de una máquina que está buscando una dirección IP que corresponde a una red a la que R2 esta directamente conectado. Si R2 tiene habilitada la opción ARP contestará a "M".

R2 "mentirá" que es N . Cada vez que R2 reciba un paquete dirigido a 172.20.71.76 se dará cuenta que lo debe reenviar a su destino real, para ello arma una trama en la que mantiene la IP original pero reemplaza la dirección destino (que era la propia - la de R2-) por la de N y pone como MAC origen la propia.

El proceso inverso, es decir que N contesta a M, es un procedimiento normal.

172.20.71.76/24 sabe que está en una subred y que el destino está en una subred distinta, por lo que el paquete es enviado al Default Gateway (R2 - 172.20.71.1).

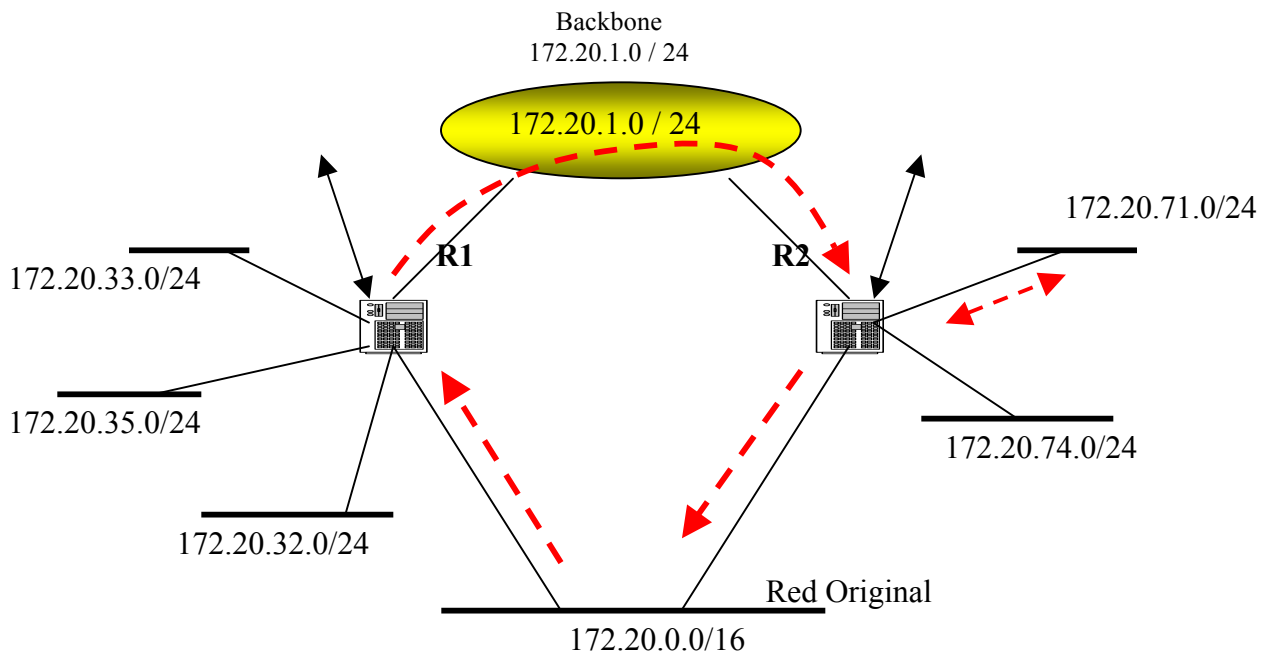
Para ello N hace un ARP y averigua la MAC de R2 y luego envia a la trama. A su vez R2 consulta su tabla para ver si tiene la MAC de M.

Si R2 esta conectado directamente con la red de M continua el proceso normal de delivery mediante un nuevo ARP para averiguar la MAC de M.

El procedimiento se limita a los ruteadores directamente conectados en trew los destinos

Peligros del Proxi ARP

Una "mejora" del método anterior es que el ruteador responda aunque NO este conectado directamente con el destino pero que SI tiene una ruta al destino.



Consideremos que ocurriría si un host X (172.20.97.101) quisiera comunicarse con N (172.20.71.76).

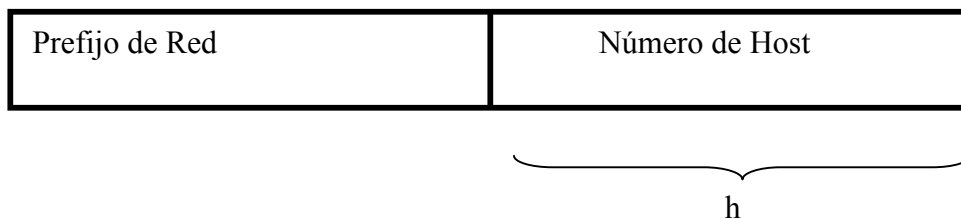
Analicemos si tanto R1 como R2 escuchan el ARP de "X"

1. En la version clásica solo responde R2 que es el ruteador que está directamente conectado a Host destino.
2. En la versión mejorada cualquier ruteador puede responder. Si respondiera R1 el host "X" enviaría sus paquetes a R1 el que los reenrutaria a R2 a través del backbone, sin embargo el camino de regreso sería a través de R2 que es la única puerta de salida conectada a la subred correspondiente. Si el Backbone tuviese problemas la carga se transportaría por la LAN

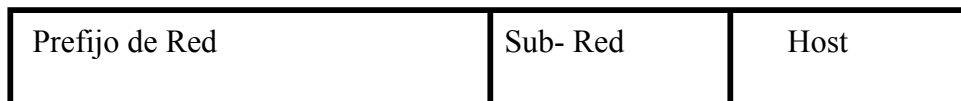
Subredes Explícitas Este método DEBE ser bien conocido por el estudiante y será sobre el que se trabajara posteriormente.

Como se sabe las direcciones se dividen en clases (A, B y C son las que nos interesan en este momento). Se propone aquí dividir los octetos de Host en Subred y Host.

Dos niveles



Tres niveles



La cantidad de bit de subred puede ir de 1 a h-2 (es decir debe dejar al menos 2 bits para los host) . Esta configuración es visible solo dentro de la red, para los usuarios externo no existe. Para que el host conozca la configuración se utiliza una máscara.

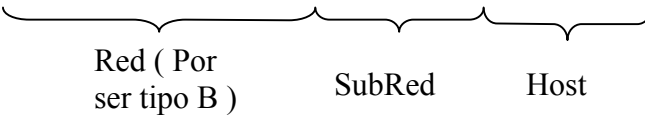
La máscara tiene 1's en las direcciones de red y 0's en la de host.

Al usuario externo le interesa únicamente como llegar a la red, la distribución en subredes es solo de interés local.

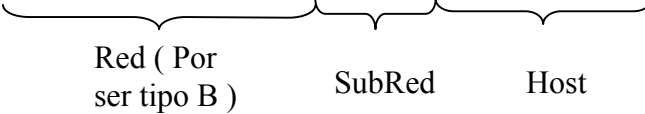
Máscaras

Vimos ya una presentación de utilización de las máscaras, es decir una palabra de 32 bits donde los 1's significan red o subred y los ceros significan host. *Aunque no lo pide la RFC es muy conveniente por comodidad que los 1's y los 0's sean contiguos.*

Ejemplo

Direcc.	130.5.5.25	1000 0010 . 0000 0101 . 0000 0101 . 0001 1001
Mascara	255.255.255.0	1111 1111 . 1111 1111 . 1111 1111 . 0000 0000
		

No hay necesidad que la máscara respete los octetos

Direcc.	130.5.5.25	1000 0010 . 0000 0101 . 0000 0101 . 0001 1001
Mascara	255.255.248.0	1111 1111 . 1111 1111 . 1111 1000 . 0000 0000
		

Existe una forma mas sencilla de tratar con las máscaras. Para el ejemplo anterior se dice que se tiene una red 130.5.5.25 con una máscara 255.255.248.0 ó lo que es lo mismo que se tiene una dirección de red 130.5.5.25 / 21. con /21 se indica que existe 21 unos desde el primer bit de la izquierda.

Ejemplo de configuración

Consideremos la red tipo **B** 172.27.0.0 . Se decide que se desean tener 5 subredes por lo cual es necesario 3 bits para direccionarlas ($2^3 = 8$), lo que nos deja disponibles 5 bits del tercer octeto para ser usado como Host. La máscara deberá ser entonces :

Máscara : **1111 1111 . 1111 1111 . 1110 0000 . 0000 0000**

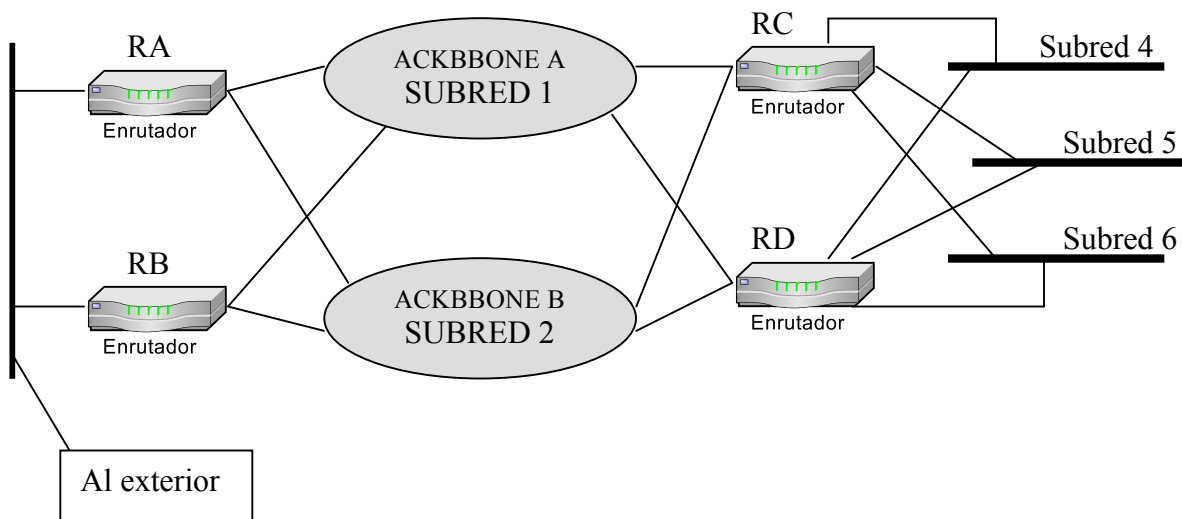
o lo que es mas fácil de leer : **255 . 255 . 224 . 0** ó mas fácil aún : **/19**

La subredes serán : (se subrayan los bit de subred)

#	Dirección	Dos últimos octetos	host desde ...hasta (dos Últimos octetos)	
0	172.27.0.0	<u>000</u> 00000.00000000	0.1	31.254
1	172.27.32.0	<u>001</u> 00000.00000000	31.1	63.254
2	172.27.64.0	<u>010</u> 00000.00000000	64.1	95.254
3	172.27.96.0	<u>011</u> 00000.00000000	96.1	127.254
4	172.27.160.0	<u>100</u> 00000.00000000	128.1	159.254
5	172.27.160.0	<u>101</u> 00000.00000000	160.1	191.254
6	172.27.192.0	<u>110</u> 00000.00000000	192.1	223.254
7	172.27.224.0	<u>111</u> 00000.00000000	224.1	255.254

Notar que no está permitida una dirección de hot en que todos los bits sean 0 (se la reserva para indicar la red), como tampoco está permitida una dirección de Hot en que todos los bits sean 1 (se la reserva para indicar difusión)
 Para el ejemplo usaremos las subredes 1, 2, 4, 5 y 6

Un esquema de la red podría ser:



Los Router A y B informaran la dirección de red (172.27.0.0) a Internet. Si por ejemplo RA recibiera un paquete dirigido a 172,27,182,133 (subred 5 host 5765), debería enviarlo a RC o RD a través de cualquiera de los dos backbone. La elección se hace de acuerdo al menor costo (en caso de igualdad de costos se utilizaran los caminos sucesivamente).

Ventaja de uso de subredes

- No incrementa el crecimiento de las tablas de ruteo si un administrador necesita armar subredes para tener mas espacio
- Los administradores tienen flexibilidad para decidir dentro de las subredes
- Los problemas internos no se reflejan fuera de la subred

Problema 1

Una organización tiene una dirección de red asignada : 193.1.1.0 /24 y necesita definir seis subredes. Cada subred debe soportar 25 Host.

1) Determinar la cantidad de bit necesarias para definir las 6 subredes

Se ve que se necesitan 3 bit ya que con dos solo se pueden definir 4 subredes, por lo que la máscara será : 255.255.255.224. Nos quedan aún 5 bit para host es decir hasta 30 host en cada subred ($2^5 - 2$). se restan 2 por las direcciones de host todas 1 y todas cero.
En notación simplificada la máscara es /27

2) Definiendo cada subred.

El esquema queda como se indica

	3 bits	5 bits
Red tipo C, 3 octetos	SRed	Host

Los bit de la subred pueden tomar 8 valores distintos generando de esta manera las subredes 0 a 7

La subred #3 será entonces aquella que tenga 011 en la parte de subred y cero en la parte de hosts (recordar que cero en host define las subredes)

#3 ➔ 11000001 . 00000001 . 00000001 . 01100000 ➔ 193.1.1.96 / 27

3) Direcciones de Host de cada subred

Todos Cero - Todos Unos <ul style="list-style-type: none">• Se evitan todos ceros y todos unos en las direcciones de Host• Los protocolos modernos permite todos unos y todos ceros en las direcciones de subred. (RIP 1 no lo soporta)

Vamos a ver que direcciones de host se permiten en las subredes. Por ejemplo en la #2

#2 ➔ 193.1.1.64 /27

Dirección de host mas baja ➔ 193.1.1.65/27

Dirección de host mas alta ➔ 193.1.1.94/27

Dirección de Broadcast ➔ 193.1.1.95

Problema 2

Se asigna a una organización la dirección 140.25.0.0 /16 y se quiere crear tantas subredes como sean posibles que soporten al menos 60 host cada una.

Se deja al lector la resolución del problema pero se dan algunas respuesta de guía

Mascara :	255.255.255.128
Subred #9 :	140.25.4.128 /25
Host #6 de subred #3 :	140.25.1.134 /25
Broadcast subred #3	140.25.1.255

Subredes.

Bibliografía Utilizada

IP Fundamentals; T.Maufer; Prentice Hall PTR

PARTE 1 General

Para hacer mas eficiente el aprovechamiento del espacio de direcciones es conveniente crear subredes de tamaño mas a justado en vez de un mismo tamaño para todas.

Evidentemente los ruteadores deberán ser informados para poder rutear adecuadamente estas direcciones.

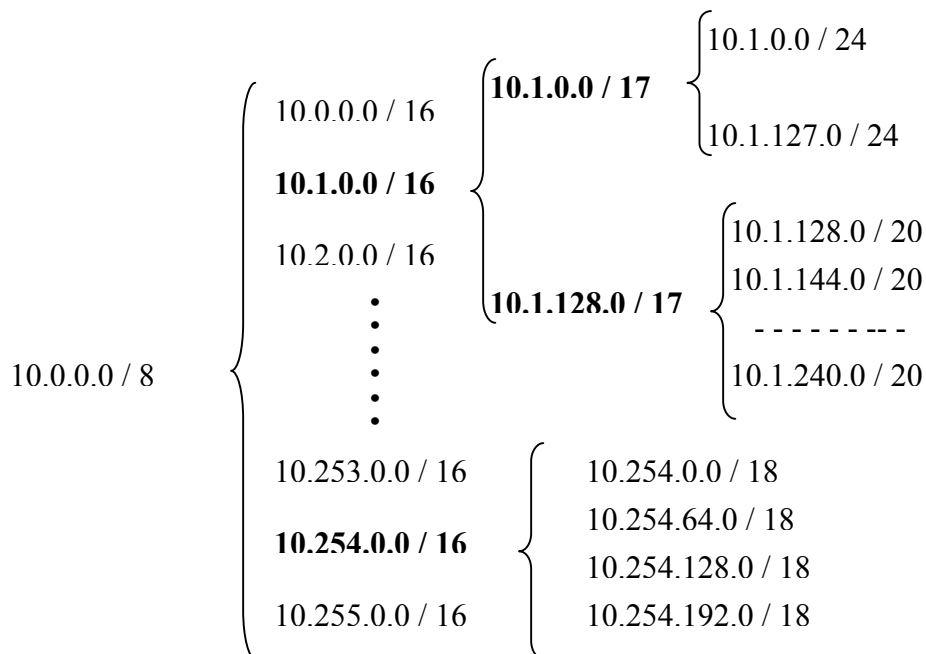
Para poder lograr esto se debe trabajar con máscaras distintas conociéndose la operación como **VLSM** (**V**ariable **L**ength **S**ubnet **M**asks)

VSLM

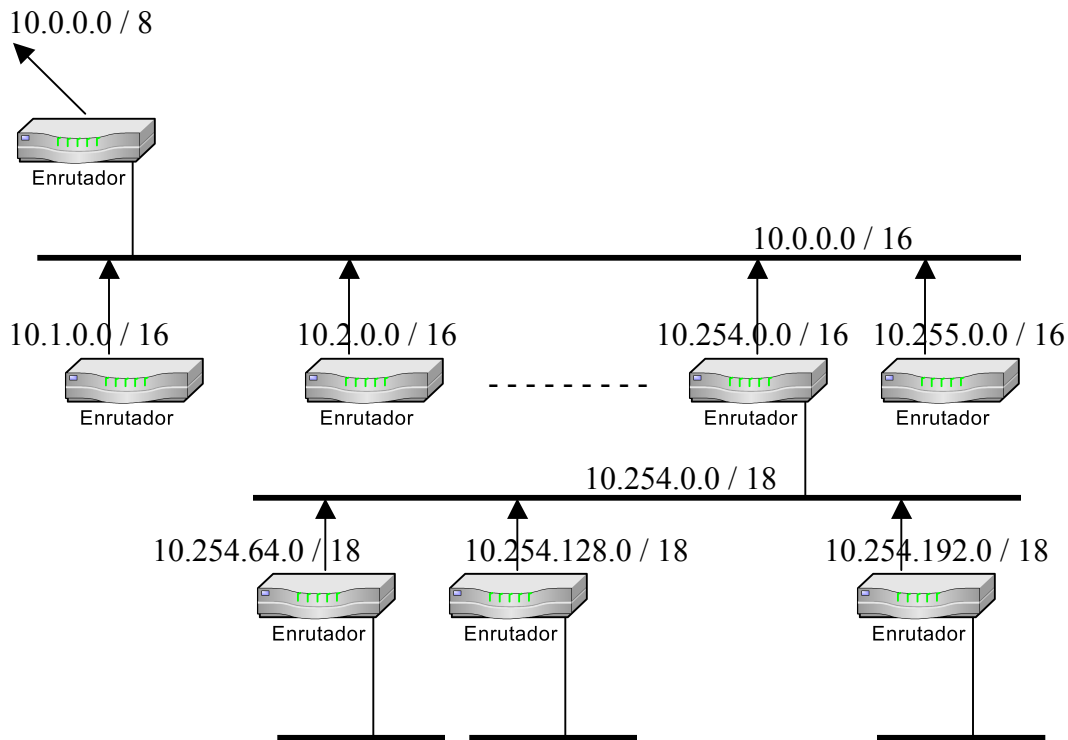
Mediante VSLM logramos que el tamaño de la subred refleje exactamente lo deseado.

Analicemos un ejemplo.

Se tiene la red 10.0.0.0 / 8 que la "subneteamos" con la máscara 255.255.0.0 es decir el tercer octeto se toma como subred. Podemos ahora tomar una de las subredes, por ejemplo la 10.1.0.0 / 16 , y la volvemos a "subnetear " con otra mascara y así siguiendo.



Una posible topología lógica de lo anterior sería : se dibuja solo parcialmente.



Consideraciones de diseño:

Antes de decidir la configuración de las subredes se deben tomar en cuenta

1. Cuantas subredes se necesitan hoy y cuanto se necesitara en el futuro
2. Cuantos Host hay en cada nivel hoy y cuantos habra en el futuro.

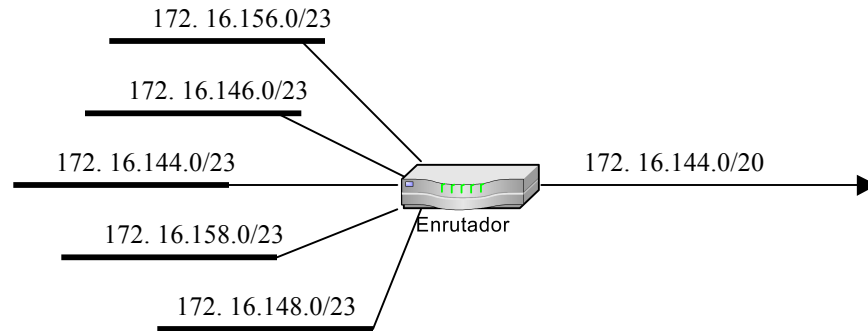
Ejemplo de direccionamiento jerárquico.

Prerrequisitos de un desarrollo de VLSM

1. Verificar si los ruteadores con los que se cuenta soportan protocolos que permiten VLSM. Prácticamente todos los protocolos modernos lo permiten (OSPF, PRIPv2, I-IS-IS, E-IGRP)
2. Los ruteadores deben basar su enrutamiento en "la mayor coincidencia".
Note que cuanto mayor es el número de bits significativos compartidos menor es el grupo de direcciones IP al que se dirige .La mejor ruta es entonces aquella con mayor /n pues es mas específica.

Si para llegar a **10.1.2.5** → **00001010.00000001.00000010.00000101**
 Ruta #1 10.1.0.0/24 → 00001010.00000001.00000000.00000000
 Ruta #2 10.1.2.0/24 → **00001010.00000001.00000010.00000000**
 Ruta #3 10.1.0.0/16 → 00001010.00000001.00000000.00000000

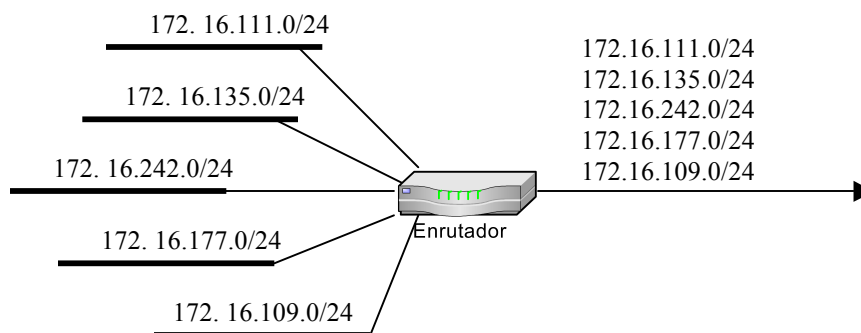
3. Debe ocurrir un estado de agregación de forma tal de tener significación topológica, es decir que grupos de subredes deben compartir los mismos bits mas significativos



Notar que

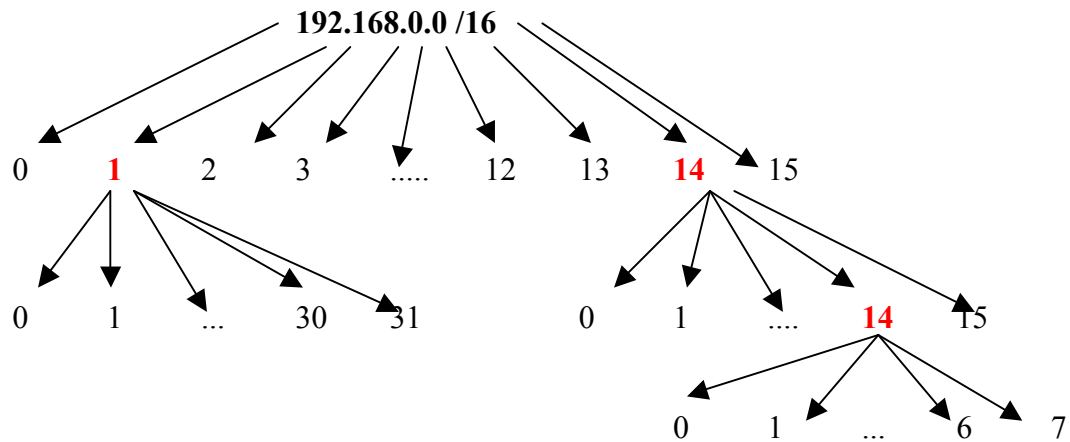
156	1001 1100
146	1001 0010
144	1001 0000
148	1001 0100

En el ejemplo anterior existe una agrupamiento en base a los bits mas significativos, en cambio en este otro ejemplo no.



111	01101111
135	10000111
242	11110000

Supongamos que una organización decide su red **192.168.0.0/16** mediante **VLSM** según el siguiente diseño.



Primero tomamos la **192.168.0.0/16** y la dividimos en 16 bloques iguales, luego el bloque 1 se divide nuevamente en 32 y así siguiendo.

Creación de 16 subredes

- Se requieren 4 bits, que sumados a los /16 del prefijo original se llega a /20 es decir un bloque continuo de $8+4 = 12$ direcciones de red (4096).

Subred #0	192.168.0.0 / 20
#1	192.168.16.0 / 20
#2	192.168.32.0 / 20
.....	
#15	192.168.240.0 / 20

Determinación de direcciones de host de #3₄, El sub índice indica que pertenece a una subred de 4 bits

La subred #3 es 192.168.48.0 / 20. Es decir contiene 12 bits de host → 4094 host posibles.

Los host válidos de la subred #3₄ son

Host #1	192.168.48.1
Host #2 1	192.168.48.2
.....	
Host #4096	192.168.63.254
Broadcast	192.168.63.255

VLSM en un ejemplo simple

Ud es el administrador de redes de una empresa con tres sucursales y una casa central unidas mediante enlaces **WAN**. Ver la figura.



Se le otorga la direccion **192.168.10.0/24**

Se dan a continuación las sucursales y la cantidad de host necesarios en cada una.

<i>Londres:</i>	<i>28 Host</i>
<i>Nueva York</i>	<i>12 Host</i>
<i>Tokio</i>	<i>12 Host</i>

La casa central en:

Ing Berenga (Pcia de Buenos Aires) : 60 Host

Su trabajo es asignar las subredes correspondientes de forma tal que se desperdicie la menor cantidad de direcciones posible.

1) Asigne primero el sitio con **mayor cantidad** de host.

Ing. Berenga 60 Host ==> 6 bit de host , 2 de direccionamiento

Con 2 bit de direccionamiento tenemos 4 subredes

# 0	192.168.10.0 / 26	<i>Ultimo octeto ==></i> 00 00 0000
# 1	192.168.10.64 / 26	01 00 0000
# 2	192.168.10.128 / 26	10 00 0000
# 3	192.168.10.192 / 26	11 00 0000

En negrita (y color rojo) está indicada la parte de subred del último octeto

Ing Berenga. toma entonces la **subred #0 192.168.10.0 / 26**

Host mas bajo 192.168.10.1
Host mas alto : _____
Broadcast: _____

Complete !

2) El siguiente lugar en cantidad de host es **Londres** con **28 máquina** para lo cual hacen falta 5 bits : $2^5 - 2 = 30$. Esto nos dejan 3 bit para direccionamiento (/ 27)

Tomamos las #1 192.168.10.64 / 26 considerada ahora como /27

1,0 192.168.10.64 / 27 *Ultimo octeto ==>* **0100** 0000
1,1 192.168.10.96 / 27 **0110** 0000

En negrita se indican los bits de direccionamiento

Asignamos # 1,0 a **Londres 192.168.10.64 / 27**

Host mas bajo: _____
Host mas alto: _____
Broadcast: 192.168.10.95

3) Nos queda por asignar Nueva York y Tokio
Nueva York necesita **12 Host** ==> 4 bits, esto nos deja 4 bits para direccionamiento.

Tomamos # 1,1pero con /28

1,1,0 192.168.10.96 / 28 *Ultimo octeto =>* **0110** 0000
1,1,1 192.168.10.112 / 28 **0111** 0000

Asignamos a **Nueva York**

1,1,0 **192.168.10.96 / 28**

Host mas bajo: _____
Host mas alto : _____
Bradcast: _____

4) **Tokio** necesita tambien **12 Host**, al igual que Nueva York, Podemos asignar la otra subred, la #1,1,1

Host mas bajo: _____
Host mas alto: _____
Broadcast: _____

5) Recordando que los enlaces **WAN** necesitan cada uno dos direcciones IP, tenemos entonces que es suficiente con 2 bits para cada una.

Para los enlaces **WAN** empleamos entonces la subred # 2 mediante una configuracion /30 (*Complete el ultimo octeto*)

2,0 192.168.10.128 / 30 ==> IP mas baja:

==> IP mas alta:

2,1 192.168.10.____ / 30 ==> IP mas baja:

==> IP mas alta:

2,2 192.168.10. ____ / 30 ==> IP mas baja

==> IP mas alta

Dibuje un esquema de la distribución de los router que permiten la formacion de subredes como desarrollada, indicando en cada punto la direccion IP correspondiente.



Explique cual es la ventaja de emplear direcciones de dirección variable en lugar del "subneteo" de direcciones fijas.

