**Capa de Red - Direccionamiento**

**Redes y comunicaciones - 2022**

**Práctica 7**

**Capa de Red – Direccionamiento**

**17, 18, 19, 20, 21**

**Introducción**

**1. ¿Qué servicios presta la capa de red?**

Transporta paquetes desde un host emisor a un host receptor. Para esto tiene dos importantes funciones:

● Reenvío (forwarding). Cuando un paquete llega al enlace de entrada de un router, éste tiene que pasar el paquete al enlace de salida apropiado.

● Enrutamiento (routing): Determina la ruta o camino que deben seguir los paquetes a medida que fluyen de un emisor a un receptor. Los algoritmos que calculan estas rutas se conocen como algoritmos de enrutamiento.

Los routers tienen una tabla de reenvío (Routeo). Este reenvía un paquete examinando el valor de un campo de la cabecera del mismo y utilizando después ese valor para indexarlo dentro de la tabla de reenvío. Esto indica a cuál de las interfaces de salida del router será reenviado el paquete. El algoritmo de enrutamiento determina los valores que se introducen en las tablas de reenvío de los routers.

Servicios:

* Direccionamiento
* Ruteo
* (De)multiplexación, de los protocolos de las capas superiores
* Evitar tener paquetes “dando vueltas por la red”

**¿Cuál es la PDU en esta capa?**

Al PDU asociado a esta capa se lo conoce como datagrama o paquete.

**¿Qué dispositivo es considerado sólo de la capa de red?**

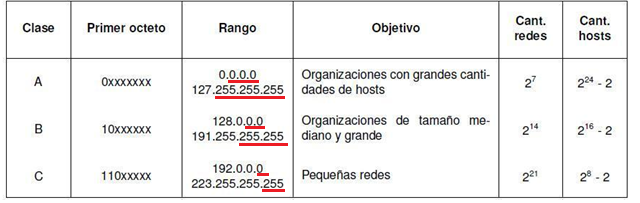
El router, que es un conmutador de paquetes que no implementa las capas superiores a la capa de red.

Por otro lado, el switch, se basa en la cabecera de la capa de enlace ya que pertenece a esta.

**2. ¿Por qué se lo considera un protocolo de mejor esfuerzo?**

Porque todos los usuarios reciben el mejor servicio posible en ese momento, según el tráfico en la red. No garantiza que los datos lleguen efectivamente a su destino (problema de la capa de transporte), ni una calidad de servicio mínima (No están garantizados la temporización relativa entre paquetes, que los paquetes se reciban en el orden que fueron emitidos ni la entrega de los paquetes transmitidos).

**3. ¿Cuántas redes clase A, B y C hay? ¿Cuántos hosts como máximo pueden tener cada una?**

****

En rojo lo que está reservado para host.

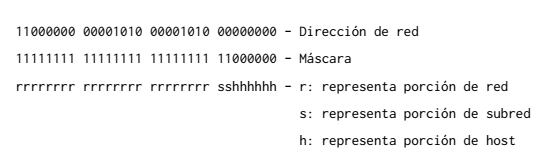
A la cantidad de hosts le resta 2 porque la primera es la dirección de subred, y la última es la dirección de broadcast. Para ver de qué clase es una IP de red hay que ver en el rango de cual está.

**4. ¿Qué son las subredes?**

Si una red desperdicia muchas direcciones IP entonces es dividida en N subredes que aprovechen mejor el espacio de direccionamiento. La forma más sencilla de entender esto es con un ejemplo:

Suponiendo un caso en que el que una red clase C es muy chica y una B muy grande, se puede dividir la B en subredes que se ajusten a este caso. Por ejemplo, dividir una red de clase B en 64 subredes de 1024 hosts cada una (en realidad 1022, pues la primera y última dirección no pueden utilizarse para hosts). De esta forma, la organización que antes desperdiciaba el 98,47% de sus direcciones IP ahora desperdiciará sólo el 2,34% y quedará la posibilidad de tener direcciones para otras 63 organizaciones de similar tamaño.

**¿Por qué es importante siempre especificar la máscara de subred asociada?**

La máscara indica qué bits son de red y de host. En las redes con clases, la máscara estaba implícita en la dirección de clase, pues se conocía a priori los bits para red y los bits para host. Con las subredes también se les asoció una máscara de subred, que resultó de utilizar algunos “bits de hosts” para crear subredes y de esta manera obtener varias subredes con menos hosts cada una (por haber usado bits de host para redes).

**5. ¿Cuál es la finalidad del campo Protocol en la cabecera IP? ¿A qué campos de la capa de transporte se asemeja en su funcionalidad?**

Se usa para multiplexar/demultiplexar la comunicación entre la red y los distintos protocolos de las capas superiores

Se asemeja al número de puerto, ya que este también se usa para multiplexar/demultiplexar la comunicación entre procesos y la red

En el campo Protocol en la cabecera IP basicamente se señala el protocolo a utilizar en la capa de transporte ya sea udp o tcp, etc.

**División en subredes**

**6. 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:**

a. ¿De qué clase de red es la dirección dada (Clase A, B o C)?

* 172.16.58.223/26 → B
* 163.10.5.49/27 → B
* 128.10.1.0/23 → B
* 10.1.0.0/24 → A
* 8.40.11.179/12 → A

**b. ¿Cuál es la dirección de subred?**

* 172.16.58.223/26 → 172.16.58.192/26

255.255.255.192 16 bits de red, 10 de subred, 6 de host

AND= 1+1=1, TODO LO DEMAS DA 0, se debe hacer el AND entre la mascara de subred y la ip que nos dan

* 172.16.58.223

10101100.00010000.00111010.11011111

11111111.11111111.11111111.11000000 máscara de subred

10101100.00010000.00111010.11000000= 172.16.58.192/26

* 163.10.5.49/27 →

11111111.11111111.11111111.11100000 máscara de subred

00110001 ->49

00100000 ->32

163.10.5.32/27 es la direccion de subred

* 128.10.1.0/23 →

11111111.11111111.11111110.00000000 máscara de subred

128.10. 00000001.00000000

128.10.0.0/23 es la direccion de subred

* 10.1.0.0/24 →

11111111.11111111.11111111.00000000 máscara de subred

10.1.0. 00000000

10.1.0.0/24 es la direccion de subred

* 8.40.11.179/12 →

11111111.11110000.00000000.00000000

8. 00101000

00100000

8.32.0.0/12 es la direccion de subred

**c. ¿Cuál es la cantidad máxima de hosts que pueden estar en esa subred?**

Clase B: tenes 16 bits para host, pero se ocuparon 10 con subneting, quedando 6 para direccionar host. Con 6 bits quedan 2^6= 64 -2 direcciones. La primera es de subred y la última de broadcast

32 bits son toda la dirección-> 232-mascara-2

* 172.16.58.223/26 → 232-mascara-2 -> 2^6 - 2 = 62
* 163.10.5.49/27 → 2^5 - 2 = 30
* 128.10.1.0/23 → 2^9 - 2 = 510
* 10.1.0.0/24 → 2^8 -2 = 254
* 8.40.11.179/12 → 2^20 -2 = 1.048.574

**d. ¿Cuál es la dirección de broadcast de esa subred?**

La primera forma de hacer la cuenta es sumando la cantidad de host a la dirección de subred, más 1, lo cual nos da la dirección de broadcast. Si no podemos calcular la dirección de la segunda subred y restarle 1 y así vamos a obtener la dirección de broadcast.

* 172.16.58.223/26 → 172.16.58.255/26

223= 1101 1111 -> host. Todo host en 1-> broadcast -> 1111 1111 =255

* 163.10.5.49/27 → 163.10.5.63/27

49= 0011 0001 Todo host en 1 para broadcast -> 0011 1111 = 63

* 128.10.1.0/23 → 128.10.0.0 + 511= 128.10.1.255

0000 0001.0000 0000 -> 1111 1111 ->255

10000000.00001010.00000000.00000000 +

00000000.00000000.00000001.11111111

----------------------------------------------------------

10000000.00001010.00000001.11111111 =   
128. 10. 1. 255

* 10.1.0.0/24 → 10.1.0.255
* 8.40.11.179/12 → 8.32.0.0/12 dir. de subred

00001000 00100000 00000000 00000000 dir de subred

00000000 00001111 11111111 11111111 hosts= 1.048.575

—-----------------------------------------------------

00001000.00101111.11111111.11111111

8.47.255.255

**e. ¿Cuál es el rango de direcciones IP válidas dentro de la subred?**

* 172.16.58.223/26 → 172.16.58.193/26 hasta 172.16.58.254/26

223= 1101 1111

Primera valida: 1100 0001 -> 193

Ultima valida: 1111 1110 -> 254

* 163.10.5.49/27 → 163.10.5.33/27 hasta 163.10.5.62/27

Ese 33 + 30 (cant de host)-> 62

* 128.10.1.0/23 → 128.10.0.1/23 hasta 128.10.1.254/23
* 10.1.0.0/24 → 10.1.0.1/24 hasta 10.1.0.254/24
* 8.40.11.179/12 → 8.32.0.1/12 hasta 8.47.255.254/12

**7. Su organización cuenta con la dirección de red 128.50.10.0. Indique:**

**a. ¿Es una dirección de red o de host?**

Es una dirección IP clase B (Por qué empieza con 128, por lo que está en su rango), lo cual quiere decir que los primeros dos octetos (128.50) corresponden a la dirección de red, y los últimos dos (10.0), a la dirección de host. Como no especifica la máscara, es de host. Para ser dirección de red debería ser 128.50.0.0, por ejemplo. (no se usarían los bits que no forman parte de la máscara de red de una clase B).

Para determinar si una dirección IP es una dirección de red o una dirección de host:

1. Representa la dirección IP y la máscara de subred en binario.
2. Aplica la operación lógica AND a la dirección IP y la máscara de subred.
3. Si el resultado de la operación AND coincide con la dirección de red correspondiente a la máscara de subred, entonces la dirección IP es una dirección de red.

192.168.0.0 = 1100 0000.1010 1000.0000 0000.**0000 0000**

/24 = 1111 1111.1111 1111.1111 1111.**0000 0000**

AND= 1100 0000.1010 1000.0000 0000.0000 0000

Si la dirección “De subred” tuviera bits en el sector de host, al hacer and con la máscara desaparecerían, dando un valor diferente, por lo que, en este caso, es dirección de subred

1. Si el resultado de la operación AND es diferente de la dirección de red, entonces la dirección IP es una dirección de host dentro de la red.

**b. Clase a la que pertenece y máscara de clase.**

Pertenece a una clase B (rango de 128-191). Por ende, la máscara es 255.255.0.0 (Si la máscara tiene un 1 indica que ese bit es de ip de red y si es 0 indica que es de ip host, 255 = 11111111 y 0 = 00000000) 🡺 11111111.11111111.00000000.00000000 = 255.255.0.0

**c. Cantidad de hosts posibles.**

Dado que la dirección de host son los últimos 2 octetos (es decir, 16 bits), la cantidad de host posibles será 2^16 – 2= 65534(También se saca de que es una red de clase B). Se resta 2 porque la primera dirección de host está reservada para referenciar a la red, y la última para el broadcast.

**d. Se necesitan crear, al menos, 513 subredes. Indique:**

Aclaración: Al ser una dirección de host, este punto NO se puede hacer con la dirección dada. No se pueden asignar subredes cuando la dirección base no es de red. Se resuelve tomando la dirección de red 128.50.0.0.

**i. Máscara necesaria.**

Se necesitan 10 bits porque con 9 no alcanza ya que 2^9 = 512<**513**. Con 2^**10** = 1024 sobran direcciones.

La máscara en binario sería: **11111111 11111111 11111111 11000000** donde los primeros 2 octetos refieren a la dirección de red y los siguientes 10 bits a la subred especificada. Por lo que la mascara será 255.255.255.192 o /26 (Por que son **26 unos**)

**ii. Cantidad de redes asignables.**

Puede haber hasta 1024 subredes. Los bits en naranja reflejan que los primeros 10 bits en 1 son para referenciar a subredes, y los **6 restantes** a host. Por ende, 2^10 = 1024.

**iii. Cantidad de hosts por subred.**

2^6 - 2= 62 (**bits en 0** en el inciso i)

**iv. Dirección de la subred 710.**

Se debe representar el número de subred que desea obtenerse menos uno en la parte que corresponde a los bits asignados para subred. El motivo por el cual se resta uno al número de subred es porque la primera

subred válida es la red 0.

709 en binario: **1011000101**.

Ubicar el número obtenido ocupando la posición de los bits de subred.

Dirección 128.50.10.0 = 10000000.00110010.00001010.00000000

Subred 710 = 10000000.00110010.**10110001.01**000000 = 128.50.177.**64**

**v. Dirección de broadcast de la subred 710.**

La dirección de broadcast es la máxima dirección de host de la subred (Con su propia máscara). En este caso: 10000000.00110010.**10110001.01111111** = 128.50.177.**127** Donde los últimos 6 bits corresponden a dirección de host.

**127**-**64**+1 (por que ambos extremos van) =64 direcciones que tiene la subred, utilizables 62 ya que se reservan la primera y la última.

**SE LE PONE LA /26 A LA DIRECCIÓN ACA?**

**8. Si usted estuviese a cargo de la administración del bloque IP 195.200.45.0/24**

**a. ¿Qué máscara utilizaría si necesita definir al menos 9 subredes?**

Es clase C, los primeros 3 octetos no se usan para subred, esta y los hosts quedan solo con el último. Se necesitan 4 bits para referenciar a 9 subredes: 2^**4** = 16.

Por lo que la máscara en binario sería 11111111 11111111 11111111 **1111**0000, lo que equivale a: 255.255.255.240 = /28 (Por que lleva 28 unos)

**b. Indique la dirección de subred de las primeras 9 subredes.**

Es una red de clase C, así que lo único que nos importa para subredes y host es el último octeto. Los 4 primeros bits del último octeto referencian a red, y los últimos 4 a host.

195.200.45.0 /28 0000 0000 = 0

195.200.45.16 /28 0001 0000 = 16

195.200.45.32 /28 0010 0000 = 32

195.200.45.48 /28 0011 0000 = 48

195.200.45.64 /28 0100 0000 = 64…

195.200.45.80 /28 0101

195.200.45.96 /28 0110

195.200.45.112 /28 0111

195.200.45.128 /28 1000

**c. Seleccione una e indique dirección de broadcast y rango de direcciones asignables en esa subred.**

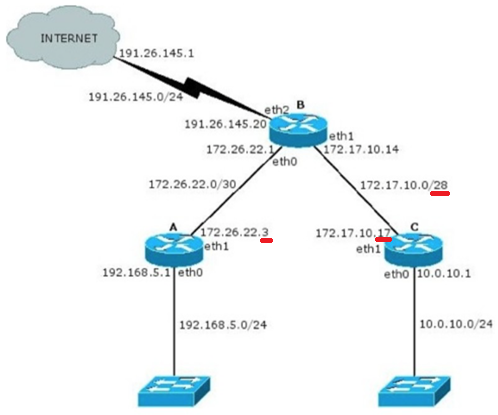
Subred: 195.200.45.16 /28

* Rango: 195.200.45.**17** /28 (**17**=0001 0001)

195.200.45. 0001 1110 /28 🡪 195.200.45.30 /28

* Broadcast: 195.200.45.31 /28 (31=0001 1111)

**9. Dado el siguiente gráfico:**

****

**a. Verifique si es correcta la asignación de direcciones IP y, en caso de no serlo, modifique la misma**

**para que lo sea.**

* 172.17.10.0/28 No está optimizado. Por ser red entre routers alcanzaría con una /30 para no desperdiciar direccionamiento.
* 172.17.10.17 no es correcta ya que, con /28 quedan 4 bits para host, con los que no alcanzan para hacer .17
* 17. 172.26.22.3 noes correcta porque es de broadcast

**b. ¿Cuántos bits se tomaron para hacer subredes en la red 10.0.10.0/24? ¿Cuántas subredes se podrían generar?**

24(Mascara)-8(Red)=16 bits para subredes

2^16 = 65536 subredes

**c. Para cada una de las redes utilizadas indique si son públicas o privadas.**

Las redes privadas no son ruteables

Espacio de direcciones privado

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)

Nos referiremos al primer bloque como "bloque de 24 bits", al segundo como "bloque de 20 bits" y al tercero como "bloque de 16 bits".

Hay un rango de redes privadas por clase.

* 172.16.58.223/26 → Privada
* 163.10.5.49/27 → Publica
* 128.10.1.0/23 → Publica
* 10.1.0.0/24 → Privada
* 8.40.11.179/12 → Publica

**CIDR**

**10. ¿Qué es CIDR (Class Interdomain routing)? ¿Por qué resulta útil?**

Sirve para frenar algunos problemas del crecimiento de Internet:

● Agotamiento del espacio de direcciones de clase B.

● Crecimiento de las tablas de enrutamiento más allá de la capacidad del software y hardware disponibles.

● Eventual agotamiento de las direcciones IP en general.

Consiste en permitir máscaras de subred de longitud variable (VLSM) para optimizar la asignación de direcciones IP y utilizar resumen de rutas para disminuir el tamaño de las tablas de enrutamiento. Con resumen se refiere a agrupar rutas, es decir de 4 rutas en 1 sola (hablando de la base de CIDR).

La dirección IP de 32 bits se divide en dos partes y se expresa como **a.b.c.d/x**, donde x indica el número de bits de la primera parte de la dirección. Los x bits más significativos de una dirección en el formato a.b.c.d/x constituyen la parte de red de la dirección IP y a menudo se hace referencia a ellos como el prefijo (o prefijo de red) de la dirección.

Cuando un router externo a la organización reenvía un datagrama cuya dirección de destino está dentro de la organización, considera solo los primeros x bits de la dirección. Esto reduce considerablemente el tamaño de la tabla, ya que una única entrada con el formato a.b.c.d/x bastará para reenviar paquetes a cualquier destino dentro de la organización.

Los 32-x bits restantes de una dirección pueden emplearse para diferenciar los dispositivos internos de la organización, teniendo todos ellos el mismo prefijo de red.

**11. ¿Cómo publicaría un router las siguientes redes si se aplica CIDR?**

**a. 198.10.1.0/24**

**b. 198.10.0.0/24**

**c. 198.10.3.0/24**

**d. 198.10.2.0/24**

198.10.0.0/22

Para hacer esto las redes a resumir deben ser consecutivas. Se escriben las direcciones IP de estas y se arma la máscara con unos en todos los lugares donde coinciden los bits de las direcciones.

1100 0110.0000 1010.0000 00**01**.0000 0000 🡪 198.10.**1**.0/24

1100 0110.0000 1010.0000 00**00**.0000 0000 🡪 198.10.**0**.0/24

1100 0110.0000 1010.0000 00**11**.0000 0000 🡪 198.10.**3**.0/24

1100 0110.0000 1010.0000 00**10**.0000 0000 🡪 198.10.**2**.0/24

22 unos🡪**11111111 11111111 11111100** **11111111** 🡪 Esos ceros representan donde son distintas (0, 1, 2, 3, 4)

Se hace un AND (1&1=1, todo lo demás da cero) con cualquiera de las direcciones IP y la máscara obtenida, y el resultado junto con la máscara obtenida, será la ruta de resumen.

11000110 00001010 00000011 00000000 → 198.10.3.0

AND **11111111 11111111 11111100 11111111**

-------------------------------------------------------------------------------------------------------------

11000110 00001010 00000000 00000000 → **198.10.0.0**

**198.10.0.0/22**. Las cuatro redes son agrupadas todas juntas por el proceso denominado agregación o resumen de rutas, el cual nos permite utilizar el mismo prefijo para anunciar múltiples redes.

**/22:** La original era /24, pero se agruparon 4 redes en una única dirección de red. Al agrupar 4 redes, con una sola dirección es posible referenciar a esas 4. 4 direcciones equivalen a 2^2, entonces hay dos bits que se voy a utilizan

**12. Listar las redes involucradas en los siguientes bloques CIDR:**

**200.56.168.0/21**

La máscara tiene 21 bits. Es decir que se utilizan los primeros 2 octetos y parte del tercero (5 de 8 bits): **11111111 11111111 11111000** 00000000🡪 Host

168= 10101000

175= 10101111

Como la dirección es de clase C, se utilizan los 3 primeros octetos para representar la dirección de red. De esos 24 bits solo se usan 21. Es decir que quedan 3 bits para referenciar redes. Es decir 2^3 = 8 redes:

200.56.168.0/24

200.56.169.0/24

200.56.170.0/24

200.56.171.0/24

200.56.172.0/24

200.56.173.0/24

200.56.174.0/24

200.56.175.0/24

**195.24.0.0/13 o 195.24/13 (Es la misma)**

Sabiendo que la ip es de clase C, se utilizan sólo **13** de los 24 bits. Es decir que se pueden usar 24-13= **11** bits para referenciar a redes.

**11111111 11111000 0000000** 00000000🡪 Host

24= 0001 1000

199=0001 1111

Es decir 2^**11**= **2048** redes:

195.**24**.**0**.0/24

195.24.1.0/24

195.24.2.0/24

…

195.**31**.**255**.0/24

**0\_\_255** =256 redes (Contar los extremos)

**24\_31**= 8 redes 🡪 256\*8=**2048**

En 195.24/13 (o sea, 195.24.0.0/13):

Notación equivalente al de 12.B

* /13 significa que los primeros 13 lugares son los que tienen valores que comparten todas las redes listadas: XXXX XXXX.XXXX X|XXX.XXXX XXXX.XXXX XXXX
* 195.24.0.0/13: 11000011.0001 1|000.0000 0000.0000 0000
* al ser una red de tipo C los 3 primeros octetos van a ser para redes y el ultimo para host, por lo tanto 195.24.0.0/24 es la primera
* la última: 11000011.0001 1|111.1111 1111.0000 0000: 195.31.255.0/24

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

128.0.0.0/**2**:

**11000000 00000000** 0000000 00000000

Primer octeto, min 0 max XX111111 (63), por lo tanto, es de 0 a 63 (64)

Segundo octeto, min 0, max 11111111 (255), por lo tanto, es de 0 a 255 (256)

64\*256= 16384 = **2^14**

Clase B: 16 - **2** = **14**

**2^14** redes:

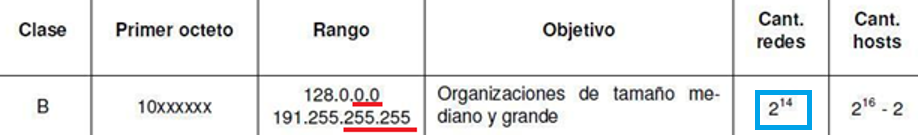
128.0.0.0/16

128.1.0.0/16

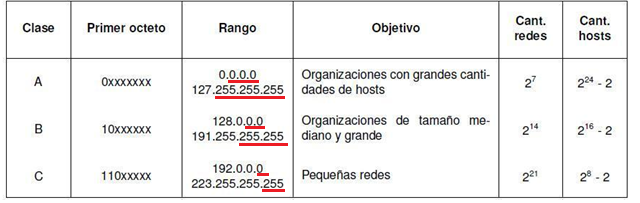
128.2.0.0/16

…

191.255.0.0/16



**¿Cuál sería el bloque CIDR que agrupa todas las redes de clase A?**

****

0.0.0.0 /1 Equivaldría al bloque CIDR que agrupa todas las redes de clase A. Aunque no hay que dejar de considerar la IP 0.0.0.0 (que estaría incluida en el bloque anterior), que es una dirección especial (no IP asignada).

CHATGPT: El bloque CIDR que agrupa todas las redes de clase A es 0.0.0.0/1 o 0/1. Este bloque cubre todas las direcciones IP desde 0.0.0.0 hasta 127.255.255.255, lo que incluye todas las redes de clase A. En CIDR, la longitud del prefijo (representada después de la barra) especifica la cantidad de bits que se utilizan para la parte de la red de la dirección IP. En el caso de 0.0.0.0/1, se utilizan los primeros 1 bit, lo que permite abarcar todas las redes de clase A.

**VLSM**

**14. ¿Qué es y para qué se usa VLSM?**

VLSM (variable-length subnet masking) consiste en realizar varios niveles de divisiones en subredes con máscaras de longitud variable para lograr mascaras más óptimas para cada subred necesaria. Se usa para frenar el agotamiento de direcciones IPv4.

Si hay 512 subredes con la capacidad de contener 126 hosts cada una y suponiendo que, excepto la única área que tiene 126 hosts, las demás áreas no tienen más de 30 hosts se estarían desperdiciando, entonces, alrededor de 90 direcciones IP por subred.

**Pasos** para dividir en subredes utilizando VLSM:

● Subnetear para la red con mayor cantidad de hosts.

● De las subredes obtenidas, asignar todas las que se puedan con el menor desperdicio posible.

● Si aún quedan segmentos de red sin una subred asignada volver al paso 1.

Otra función de VLSM es descentralizar las redes y de esta forma conseguir redes más seguras y jerárquicas.

**15. Describa, con sus palabras, el mecanismo para dividir subredes utilizando VLSM.**

Se toma una red y se divide en subredes fijas, luego se toma una de esas subredes y se vuelve a dividir, tomando bits de la porción de hosts, ajustándose a la cantidad de hosts requeridos por cada segmento de nuestra red.

Por ejemplo, si se toma la dirección de red 192.168.1.0/24 y se subdivide usando una máscara /26 tendremos 4 subredes (192.168.1.0/26, 192.168.1.64/26, 192.168.1.128/26 y 192.168.1.192/26). Suponga que se construye un enlace serie entre dos routers y tomamos para ello una de las subredes (la 192.168.1.0/26): con esta máscara de subred sin aplicar vlsm se desperdiciarían 60 direcciones utilizables (64-2=62 menos las 2 direcciones aplicadas a las interfaces de los routers, da 62 hosts, [64-2=62] una dirección para el nombre de la red o dirección de red y la otra para la dirección de difusión o broadcast).

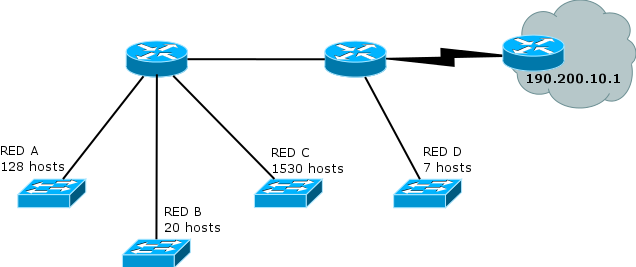
Ahora, si se aplica vlsm a la subred anterior (la 192.168.1.0/26) y se toman "prestados" 4 bits de la porción de host tendríamos otras 16 subredes /30 (192.168.1.0/30, 192.168.1.4/30, 192.168.1.8/30, 192.168.1.12/30, 192.168.1.16/30 y así sucesivamente hasta la 192.168.1.60/30) cada una con un total de 4 direcciones totales, pero solamente dos direcciones utilizables y no se genera desperdicio. Finalmente podemos tomar cualquiera de ellas, por ejemplo, la 192.168.1.4/30 y aplicar las direcciones 192.168.1.5/30 y 192.168.1.6/30 a las interfaces de los routers.

Video explicativo: <https://www.youtube.com/watch?v=0LyltIZejvA>

**16. Suponga que trabaja en una organización que tiene la red que se ve en el gráfico y debe armar el**

**direccionamiento para la misma, minimizando el desperdicio de direcciones IP. Dicha organización posee**

**la red 205.10.192.0/19, que es la que usted deberá utilizar.**



**a. ¿Es posible asignar las subredes correspondientes a la topología utilizando subnetting sin vlsm?**

**Indique la cantidad de hosts que se desperdicia en cada subred.**

La Red C es la que más hosts necesita. Para 1530 hosts se necesitan 11 bits: 2^11 - 2 = 2046.

Se están usando 19 bits para la dirección de red. 11 son usados para los hosts por lo que quedarían 2 bits para subredes.

11111111 11111111 111sshhh hhhhhhhh

2^2= 4. Se podrían referenciar 4 redes, pero como son 5, contando la entre routers, no es posible y hay que utilizar vlsm para realizar el subnetting.

**b. Asigne direcciones a todas las redes de la topología. Tome siempre en cada paso la primera dirección**

**de red posible.**

1) Calcular la máscara para la/s subred/es de mayor cantidad de hosts:

1530 hosts -> 2^**11** - 2 = 2046 hosts

Máscara: 11111111 11111111 11111000 00000000 /19

2) Escribir la red original en forma binaria

**205.10.192.0/19**

11001101 00001010 11000000 00000000

3) Dibujar una línea vertical por el límite original de la subred

11001101 00001010 110|00000 00000000 (/19)

4) Dibujar una línea vertical por el segundo límite de subred

11001101 00001010 110|00|**000 00000000** (/21)

5) Calcular las direcciones de las subredes (4 en este caso). Se hace con lo que quedó entre las ||

* 11001101 00001010 110|00|000 00000000

205.10.192.0/21 → la dejo para la **Red C**

* 11001101 00001010 110|01|000 00000000

205.10.200.0/21

* 11001101 00001010 110|10|000 00000000

205.10.208.0/21 ⇒ *Estas van para el punto c)*

* 11001101 00001010 110|11|000 00000000

205.10.216.0/21 ⇒ *Estas van para el punto c)*

Ahora tengo que definir la red para 128 hosts (Red A). Necesito 8 bits → 2^8 - 2 = 254

Máscara: 11111111 11111111 11111111 00000000 /24

Red original: 205.10.200.0/21

11001101 00001010 11001|000 |00000000 (Queda la /21 anterior y la /24 como |)

Direcciones de las subredes:

* 11001101 00001010 11001|000 |00000000

205.10.200.0/24 → Asignada a la **Red A**

* 11001101 00001010 11001|001 |00000000

205.10.201.0/24

* 11001101 00001010 11001|010 |00000000

205.10.202.0/24 ⇒ *De acá para abajo van para el punto c)*

* 11001101 00001010 11001|011 |00000000

205.10.203.0/24

* 11001101 00001010 11001|100 |00000000

205.10.204.0/24

* 11001101 00001010 11001|101 |00000000

205.10.205.0/24

* 11001101 00001010 11001|110 |00000000

205.10.206.0/24

* 11001101 00001010 11001|111 |00000000

205.10.207.0/24

Ahora tomamos la red B que tiene 20 hosts. Para 20 hosts son 5 bits → 2^5 - 2 = 30

Máscara: 11111111 11111111 11111111 11100000 /27

11001101 00001010 11001000 |000|00000

205.10.200.0/27

* 11001101 00001010 11001000 |000|00000

205.10.201.0/27 → asignada para la **Red B**

* 11001101 00001010 11001000 |001|00000

205.10.201.32/27

* 11001101 00001010 11001000 |010|00000

205.10.201.64/27 ⇒ *De acá para abajo van para el punto c)*

* 11001101 00001010 11001000 |011|00000

205.10.201.96/27

* 11001101 00001010 11001000 |100|00000

205.10.201.128/27

* 11001101 00001010 11001000 |101|00000

205.10.201.160/27

* 11001101 00001010 11001000 |110|00000

205.10.201.192/27

* 11001101 00001010 11001000 |111|00000

205.10.201.224/27

Ahora debemos asignar la red D que tiene 7 hosts. Necesito 4 bits → 2^4 - 2 = 14

Máscara: 11111111 11111111 11111111 11110000 /28

11001101 00001010 11001000 001|0|0000

205.10.201.32/28

* 11001101 00001010 11001000 001|0|0000

205.10.201.32/28 → Asigno a la red D

* 11001101 00001010 11001000 001|1|0000

205.10.201.48/28

Ahora queda la red E (red entre routers). Se necesita una única referencia, serían 2 bit. 2^2 - 2

**siempre son 2 bits para conexión entre routers**

11001101 00001010 11001000 0011|00|00

205.10.201.48/30

* 11001101 00001010 11001000 0011|00|00

205.10.201.48/30 → asignada a la red E

* 11001101 00001010 11001000 0011|01|00

205.10.201.52/30 ⇒ *De acá para abajo van para el punto c)*

* 11001101 00001010 11001000 0011|10|00

205.10.201.56/30

* 11001101 00001010 11001000 0011|11|00

205.10.201.60/30

Router 1: 205.10.201.49/30

Router 2: 205.10.201.50/30

**c. Para mantener el orden y el inventario de direcciones disponibles, haga un listado de todas las**

**direcciones libres que le quedaron, agrupándolas utilizando CIDR.**

Para la primera división quedaron libres las subredes:

● 205.10.208.0/21

● 205.10.216.0/21

Se hace la máscara con todos los bits donde coincide

11001101.00001010.1101|0000.00000000 → 205.10.208.0

11001101.00001010.1101|1000.00000000 → 205.10.216.0

----------------------------------------------

11111111.11111111.11110000.00000000 → 255.255.240.0 (mascara resultado)

(se cuentan los bits que coinciden y con eso se hace la máscara, si coinciden 20 bits, la máscara es de 20)

* Acá se hace un and con cualquiera de las direcciones IP y la máscara obtenida.

11001101.00001010.11010000→ 205.10.208.0

11111111 11111111. 11110000

------------------------------------------

11001101.00001010.11010000 → Resultado (205.10.208.0/20)

-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*

Para la segunda subdivisión quedaron libres:

● 205.10.202.0/24

● 205.10.203.0/24

● 205.10.204.0/24

● 205.10.205.0/24

● 205.10.206.0/24

● 205.10.207.0/24

11001101.00001010.11001|010 → 205.10.202

11001101.00001010.11001|011 → 205.10.203

11001101.00001010.11001|100 → 205.10.204

11001101.00001010.11001|101 → 205.10.205

11001101.00001010.11001|110 → 205.10.206

11001101.00001010.11001|111 → 205.10.207

-----------------------------------------

11111111.11111111.11111000.00000000 → 255.255.248.0 (mascara)

* Acá se hace un and con cualquiera de las direcciones IP y la máscara obtenida.

11001101.00001010.11001110 00000000→ 205.10.206.0

11111111.11111111. 11111000 00000000→ máscara

------------------------------------------

11001101.00001010.11001000 → Resultado (205.10.200.0/21)

-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*

Para la tercera subdivisión me quedan libres:

● 205.10.201.64/27

● 205.10.201.96/27

● 205.10.201.128/27

● 205.10.201.160/27

● 205.10.201.192/27

● 205.10.201.224/27

01000000 → 64

01100000 → 96

10000000 → 128

10100000 → 160

11000000 → 192

11100000 → 224

---------------

00000000 → 205.10.201.0 (mascara)

aca podemos preguntar el que pasa si incluimos el .32 y el .0 porque esas ya estan usadas, pero la mascara que conseguimos puede darnos como resultado, ips que ya estamos usando

* Acá se hace un and con cualquiera de las direcciones IP y la máscara obtenida.

10100000

00000000

------------

00000000 → Resultado (205.10.201.0/24)

-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*

Para la última subdivisión:

● 205.10.201.52/31

● 205.10.201.56/31

● 205.10.201.60/31

00110100 → 52

00111000 → 56

00111100 → 60

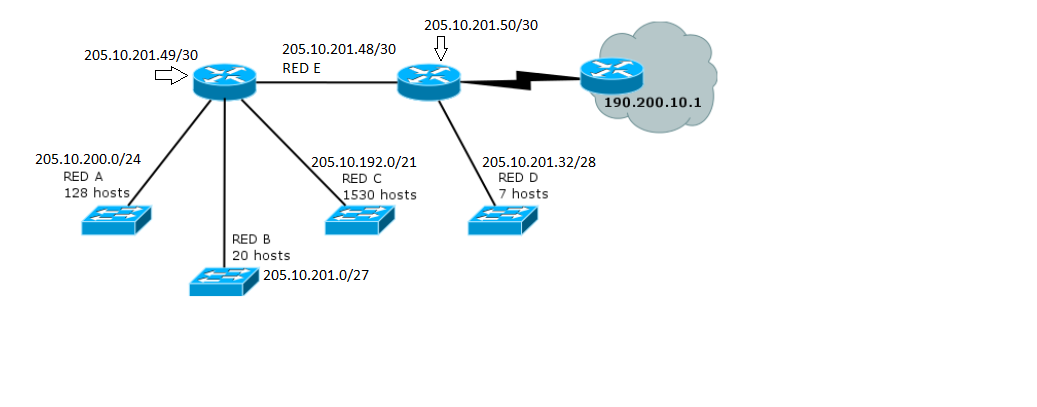
------------

11110000 → 255.255.255.240 (mascara)

Hago el AND con cualquiera de las anteriores y da: 00110000 → Resultado

(205.10.201.48/28)

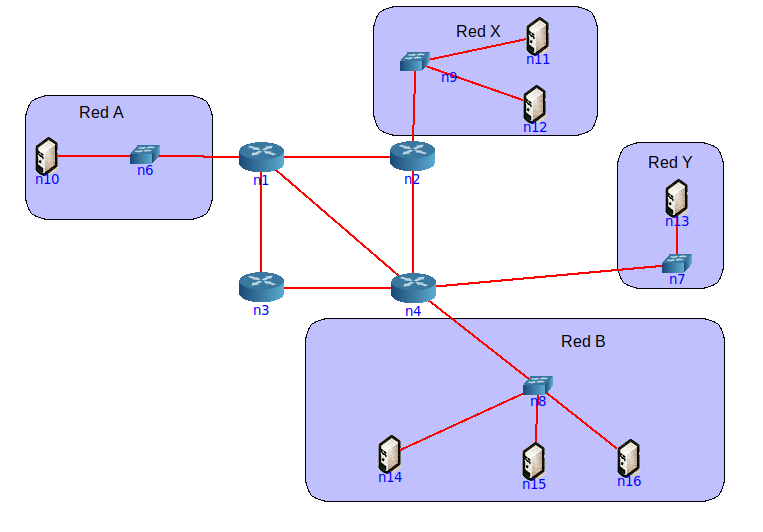
**d. Asigne direcciones IP a todas las interfaces de la topología que sea posible.**



**IDEM 16**

**17. Utilizando la siguiente topología y el bloque asignado, arme el plan de direccionamiento IPv4 teniendo en cuenta las siguientes restricciones:**

**Utilizar el bloque IPv4 200.100.8.0/22.**



**a. La red A tiene 125 hosts y se espera un crecimiento máximo de 20 hosts.**

**b. La red X tiene 63 hosts.**

**c. La red B cuenta con 60 hosts**

**d. La red Y tiene 46 hosts y se espera un crecimiento máximo de 18 hosts.**

**e. En cada red, se debe desperciciar la menor cantidad de direcciones IP posibles. En este sentido,**

**las redes utilizadas para conectar los routers deberán utilizar segmentos de red /30 de modo de**

**desperdiciar la menor cantidad posible de direcciones IP.**

**18. Asigne direcciones IP en los equipos de la topología según el plan anterior.**

**nota de Mati: estos ejercicios ya estan hechos, mira la carpeta del drive, el pdf se llama Punto17y18\_p7.pdf**

**ICMP y Configuraciones IP**

**19. Describa qué es y para qué sirve el protocolo ICMP.**

Es un protocolo parte de IP, que brinda mensajes de control y estado (errores, etc) a protocolos que nativamente no les brindan soporte, como UDP e IP.

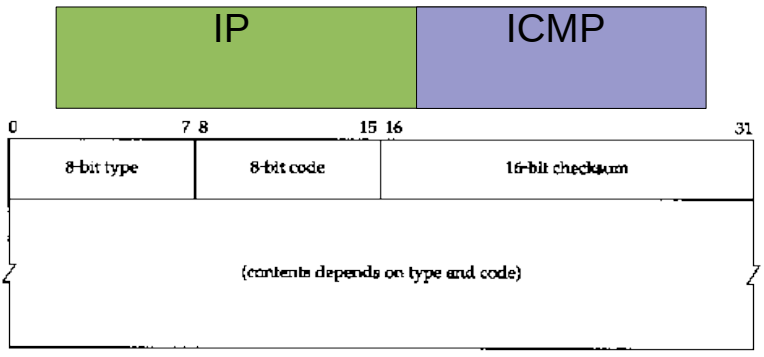
La capa de red tiene tres componentes principales: el protocolo IP, los protocolos de enrutamiento (incluyendo RIP, OSPF y BGP) e ICMP.

Los hosts y los routers utilizan ICMP, para intercambiarse información acerca de la capa de red. El uso más típico de ICMP es la generación de informes de error. Por ejemplo, al ejecutar una sesión Telnet, FTP o HTTP, puede encontrarse con un mensaje de error como “Red de destino inalcanzable”. Este mensaje

tiene su origen en ICMP

* Protocolo de L3, “Helper” de IP. ip carece de control, el mismo es dado por un protocolo auxiliar.
* No le agrega confiabilidad a IP, solo brinda un “feedback” para poder resolver problemas en la red.
* Podría ser prescindible de en IPv4, aunque en la RFC se indica que se debe implementar en cada módulo IP.
* Se encapsula en IP.
* ICMP no es un protocolo de transporte, ya que no fue concebido para llevar datos de usuario.

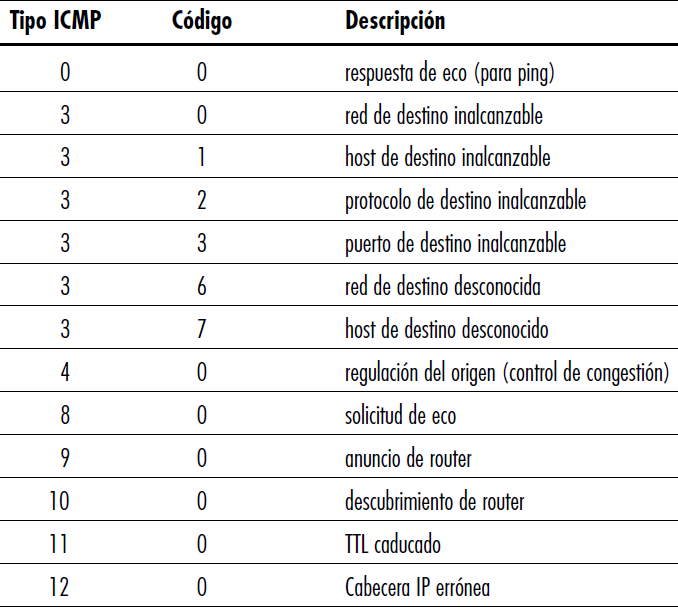
Formato de Mensaje:



**Algunos Tipos de Mensajes ICMP:**

Los mensajes ICMP tienen un campo de tipo y un campo de código, y contienen la cabecera y los 8 primeros bytes del datagrama IP que ha dado lugar a la generación del mensaje ICMP en primer lugar (de modo que el emisor puede determinar qué datagrama ha producido el error).

* Echo Request/Echo Reply (PING).
* Destino Inalcanzable.
* TTL expirado.
* Source Quench (Control de Congestión).
* Redirección de Ruta.
* Address Mask y Timestamp.



**ICMP PING (Echo):**

* Pensado para probar conectividad IP entre dos hosts.
* Sirve para medir el RTT min/avg/max/dev y loss, de esta forma poder diagnosticar problemas.
* Packet INternet Gopher, el nombre basado en el sonido de un sonar de un submarino al escanear.
* Si un nodo recibe un Echo Request, debe responder copiando el contenido con un Echo Reply (PONG). RFC-1122.
* Actualmente, muy desprestigiado ;-) Se filtra.

**a. Analice cómo funciona el comando ping.**

**i. Indique el tipo y código ICMP que usa el ping.**

Ping envía un mensaje ICMP de tipo 8 y código 0 al host especificado (solicitud de eco).

Sentido Valor campo “Tipo” Valor campo “Código”

Solicitud 8 0

**ii. Indique el tipo y código ICMP que usa la respuesta de un ping.**

El host de destino, al ver la solicitud de eco, devuelve una respuesta de eco ICMP de tipo 0 y código 0 (respuesta de eco).

Sentido Valor campo “Tipo” Valor campo “Código”

Respuesta 0 0

**b. Analice cómo funcionan comandos como traceroute/tracert de Linux/Windows y cómo manipulan el**

**campo TTL de los paquetes IP.**

**Traceroute** permite trazar una ruta desde un host a cualquier otro host enviando datagramas IP desde el origen al destino, para determinar los nombres y las direcciones de los routers entre ambos puntos. Cada datagrama transporta un segmento UDP con un número de puerto poco probable. El primero de estos datagramas tiene un TTL de 1, el segundo de 2, el tercero de 3, etc. El origen también inicia los temporizadores para cada uno de los datagramas. Cuando el datagrama n-ésimo llega al router n-ésimo, éste observa que el TTL acaba de caducar, entonces el router descarta el datagrama y envía al origen un mensaje ICMP de TTL caducado (tipo 11, código 0). Este incluye el nombre del router y su dirección IP. Cuando dicho mensaje llega al origen, éste obtiene el tiempo de ida y vuelta del temporizador, y el nombre y la dirección IP del router n-ésimo gracias al propio mensaje ICMP. Uno de los datagramas terminará recorriendo el camino completo hasta el host de destino. Dado que ese datagrama contiene un segmento UDP con un número de puerto improbable, el host de destino devuelve al origen un mensaje ICMP de puerto inalcanzable (tipo 3, código 3). Cuando el host de origen recibe este mensaje ICMP, sabe que no tiene que enviar más paquetes de prueba.

* Al inicio, el TTL actual valdrá 1
* Envía un datagrama IP dirigido al destino, enviando el TTL actual
* El valor se irá decrementando con cada salto, y cuando llega a 0 en un router, éste devuelve un paquete ICMP Time Exceeded, incluyendo su IP en el header de campos adicionales
* El comando recibe este datagrama y muestra la IP en pantalla
* Si llegó al destino, finaliza la ejecución
* Sino, incrementa el valor de TTL actual y repite

**c. Indique la cantidad de saltos realizados desde su computadora hasta el sitio www.nasa.gov. Analice:**

*sudo traceroute "www.nasa.gov" --icmp*

traceroute to www.nasa.gov (18.65.48.62), 30 hops max, 60 byte packets

1 192.168.122.1 (192.168.122.1) 0.211 ms 0.187 ms 0.109 ms

2 192.168.0.1 (192.168.0.1) 1.388 ms 1.888 ms 2.360 ms

3 \* \* \*

4 \* \* \*

5 \* \* \*

6 \* \* \*

7 181.96.103.168 (181.96.103.168) 47.231 ms 46.865 ms 46.704 ms

8 99.82.181.106 (99.82.181.106) 19.780 ms 15.035 ms 12.411 ms

9 \* \* \*

10 \* \* \*

11 \* \* \*

12 \* \* \*

13 \* \* \*

14 150.222.28.142 (150.222.28.142) 19.700 ms 14.937 ms 15.331 ms

15 server-18-65-48-62.eze50.r.cloudfront.net (18.65.48.62)

14.173 ms 14.308 ms 14.428 ms

*traceroute "www.nasa.gov" --icmp -n*

traceroute to www.nasa.gov (18.65.48.62), 30 hops max, 60 byte packets

1 192.168.122.1 0.210 ms 0.123 ms 0.195 ms

2 192.168.0.1 1.859 ms 2.392 ms 2.809 ms

3 \* \* \*

4 \* \* \*

5 \* \* \*

6 \* \* \*

7 181.96.103.168 22.801 ms 21.608 ms 26.619 ms

8 99.82.181.106 25.675 ms 24.783 ms 24.799 ms

9 \* \* \*

10 \* \* \*

11 \* \* \*

12 \* \* \*

13 \* \* \*

14 150.222.28.142 11.544 ms 12.085 ms 12.344 ms

15 18.65.48.62 11.216 ms 11.155 ms 11.246 ms

Esto significa que ese router intermedio no respondió al datagrama con un mensaje ICMP

**i. Cómo hacer para que no muestre el nombre del dominio asociado a la IP de cada salta.**

**ii. La razón de la aparición de \* en parte o toda la respuesta de un salto.**

**d. Verifique el recorrido hacia los servidores de nombre del dominio unlp.edu.ar. En base al recorrido**

**realizado, ¿podría confirmar cuál de ellos toma un camino distinto?**

*dig "unlp.edu.ar" ns*

; <<>> DiG 9.16.33-Debian <<>> unlp.edu.ar ns

;; global options: +cmd

;; Got answer:

;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 27462

;; flags: qr rd ra ad; QUERY: 1, ANSWER: 4, AUTHORITY: 0, ADDITIONAL: 7

;; OPT PSEUDOSECTION:

; EDNS: version: 0, flags:; udp: 1232

; COOKIE: d6ddb4dfda606f3401000000635b17f5e1fbedd84f73b8ce (good)

;; QUESTION SECTION:

;unlp.edu.ar. IN NS

;; ANSWER SECTION:

unlp.edu.ar. 3600 IN NS unlp.unlp.edu.ar.

unlp.edu.ar. 3600 IN NS anubis.unlp.edu.ar.

unlp.edu.ar. 3600 IN NS hostex.unlp.edu.ar.

unlp.edu.ar. 3600 IN NS ns1.riu.edu.ar.

;; ADDITIONAL SECTION:

ns1.riu.edu.ar. 3600 IN A 170.210.0.18

unlp.unlp.edu.ar. 3600 IN A 163.10.0.67

anubis.unlp.edu.ar. 3600 IN A 163.10.0.65

ns1.riu.edu.ar. 3600 IN AAAA 2800:110:fe::aad2:12

unlp.unlp.edu.ar. 3600 IN AAAA 2800:340:0:64::67

anubis.unlp.edu.ar. 3600 IN AAAA 2800:340:0:64::65

;; Query time: 228 msec

;; SERVER: 172.28.0.29#53(172.28.0.29)

;; WHEN: Thu Oct 27 20:44:53 -03 2022

;; MSG SIZE rcvd: 289

*sudo traceroute "unlp.unlp.edu.ar." --icmp*

traceroute to unlp.unlp.edu.ar. (163.10.0.67), 30 hops max, 60 byte packets

1 192.168.122.1 (192.168.122.1) 0.651 ms 0.565 ms 0.521 ms

2 192.168.0.1 (192.168.0.1) 2.538 ms 3.048 ms 3.750 ms

3 \* \* \*

4 \* \* \*

5 \* \* \*

6 \* \* \*

7 181.96.113.234 (181.96.113.234) 26.215 ms 24.985 ms 24.994 ms

8 200.0.17.12 (200.0.17.12) 180.863 ms 182.753 ms 183.245 ms

9 200.115.81.1 (200.115.81.1) 21.418 ms 14.748 ms 20.826 ms

10 163.10.199.205 (163.10.199.205) 15.931 ms 15.763 ms 15.984 ms

11 \* \* \*

12 \* \* \*

...

30 \* \* \*

*sudo traceroute "anubis.unlp.edu.ar." --icmp*

*sudo traceroute "hostex.unlp.edu.ar." --icmp*

*sudo traceroute "ns1.unlp.edu.ar." --icmp*

^ En teoria esta

**20. ¿Para que se usa el bloque 127.0.0.0/8? ¿Qué PC responde a los siguientes comandos?**

Las direcciones del rango ' 127.0.0.0/8 ' son direcciones de loopback, de las cuales se utiliza, de forma mayoritaria, la ' 127.0.0.1 '

loopback: direcciones del rango '127.0.0.0/8'. Se suele utilizar cuando una transmisión de datos tiene como destino el propio host. También se suele usar en tareas de diagnóstico de conectividad y validez del protocolo de comunicación. Crea un método de acceso directo para las aplicaciones y servicios TCP/IP que se ejecutan en el mismo dispositivo para comunicarse entre sí.

[**https://es.wikipedia.org/wiki/Anexo:Direcciones\_IP\_reservadas**](https://es.wikipedia.org/wiki/Anexo:Direcciones_IP_reservadas)

La dirección de loopback es una dirección especial que los hosts utilizan para dirigir el tráfico hacia ellos mismos.

**a. ping 127.0.0.1**

**b. ping 127.0.54.43**

A pesar de que sólo se usa la dirección única '127.0.0.1', se reservan las direcciones desde la '127.0.0.0' hasta la '127.255.255.255'. Cualquier dirección dentro de este bloque producirá un loopback dentro del host local.

**21. Investigue para qué sirven los comandos ifconfig y route.**

* **ifconfig**: Se utiliza en interfaces en Linux para configurarlas manualmente, ver su estado, asignarles direcciones IP, habilitar o deshabilitarlas, administrar cachés ARP, rutas, etc. Además, las secuencias de comandos de inicio de Oracle Solaris ejecutan ifconfig para configurar pseudo-interfaces, como los puntos finales de túnel 6to4.
* **Route**: Muestra la tabla de enrutamiento que reside en el kernel y también se usa para modificarla.

**¿Qué comandos podría utilizar en su reemplazo?**

Como reemplazo de ifconfig y route está el paquete de utilidades de red iproute2 para versiones posteriores a Linux 2.2.x.

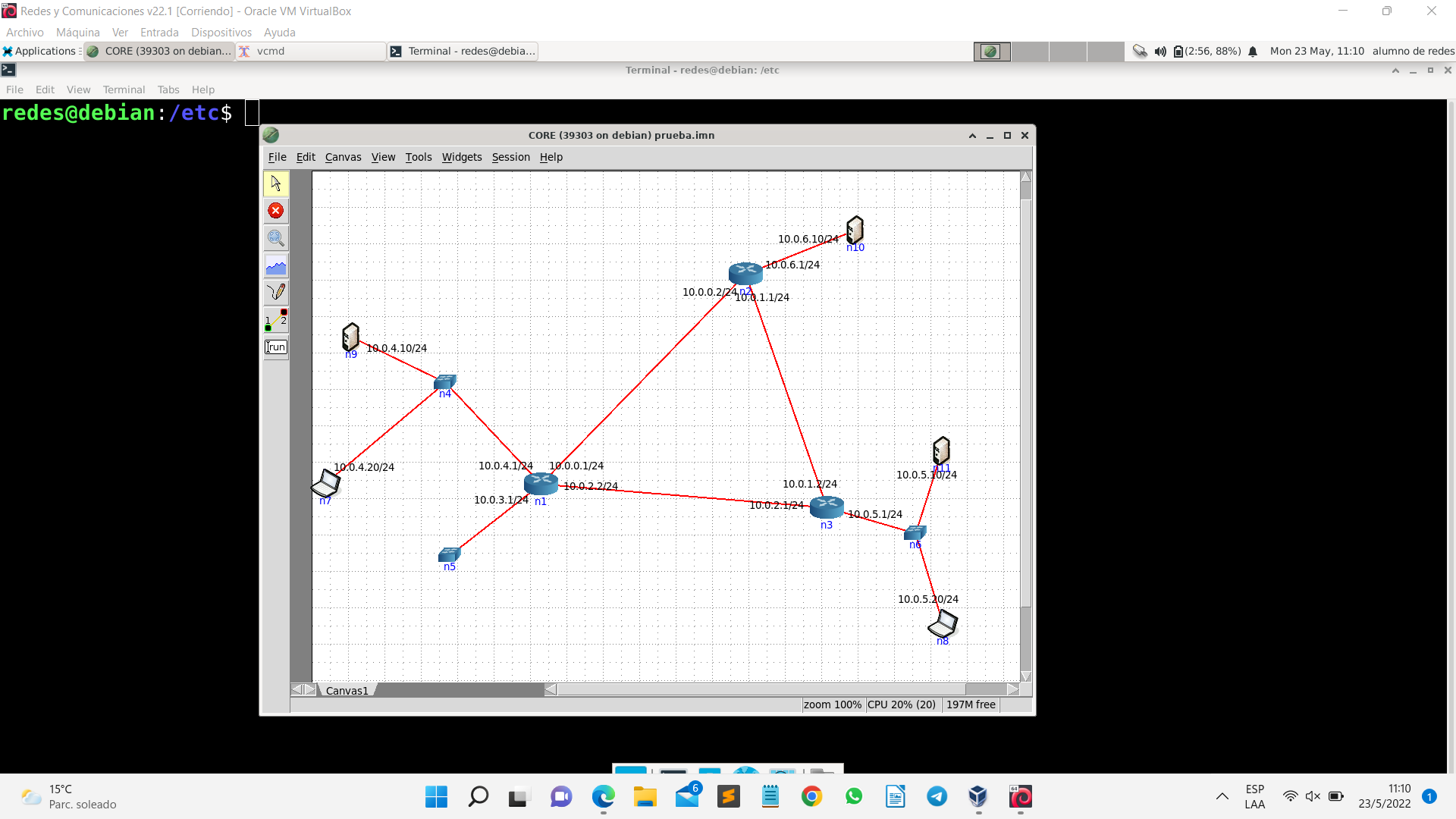
La herramienta principal de iproute2 es el comando «ip», con el que podremos ver y configurar direcciones IP, ver y configurar tablas de enrutamiento, ver y configurar túneles IP, y también ver y configurar la interfaz física. La sintaxis que debemos utilizar con este comando es la siguiente: **ip [ OPCIONES ] OBJETO [ COMANDO ]**

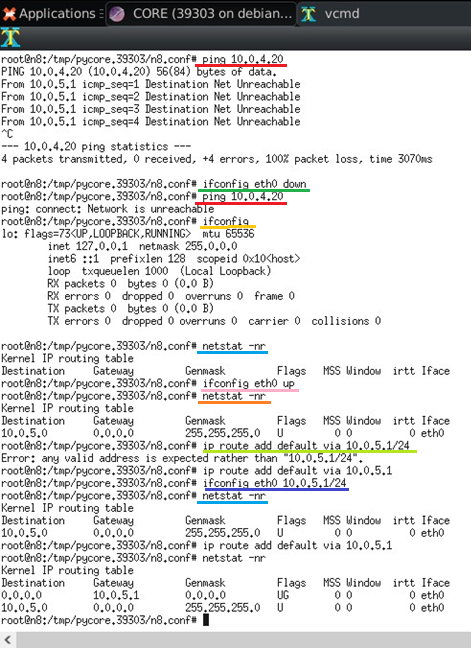
**Inicie una topología con CORE, cree una máquina y utilice en ella los comandos anteriores para practicar sus diferentes opciones, mínimamente:**

**Configurar y quitar una dirección IP en una interfaz.**

**Ver la tabla de ruteo de la máquina.**

**DEBE ESTAR BIEN, PERO HAY QUE REVISAR**





Comandos en la captura anterior:

**-ping 10.0.4.20:** para enviar una consulta desde n8 a n7, vemos que se envía pero no llega el mensaje

**-ifconfig eth0 down**: para deshabilitar la interfaz de eth0.

**-ping 10.0.4.20**: de nuevo pero ya la red es inalcanzable.

-**ifconfig:** nos mostrará las conexiones habilitadas en este caso eth0 no lo esta-

**-netstat -nr:** también podemos observar que en la tabla de ruteo tampoco aparece la conexión

**-ifconfig eth0 up**: se habilita de nuevo la conexión.

-**netstat -nr:** volvemos a ver la tabla de ruteo y vemos que el gateway es incorrecto.

-**ip add route default via 10.0.5.1**: agregamos la dir. ip por defecto.

-**ifconfig eth0 10.0.5.1/24**: modificamos la dir. ip para incluya a 10.0.5.1, esto no iría, lo que hace es modificar la ip de eth0, lo que está mal

-**netstat -nr**: volvemos a ver la tabla y ahora los datos son correctos.

Si hacemos un ping veremos los mismos datos que al principio.