

Trabajo de promoción teórica - IP Tipo A

Grupo M

Integrantes:

Esteban Débole 13581/7

Juan Cruz Chebreli 13736/7

Nahuel Di Criscio 13013/7

Direccionamiento

Dado el bloque IP **72.168.152.0/22** se tiene que :

La red A tiene 70 hosts y se espera un crecimiento máximo de 20 hosts.

La red X tiene 150 hosts.

La red B cuenta con 20 hosts

La red Y tiene 35 hosts y se espera un crecimiento máximo de 30 hosts.

Los bloques IP asignados en los enlaces entre routers deberán desperdiciar la menor cantidad de direcciones posibles.

Nota: Utilice VLSM.

Con el bloque dado, sabemos que **22** bits son utilizados para identificar la red, la red más grande pedida es de 150, por lo cual necesitamos **8** bits para representarla. Por lo tanto, necesitamos una máscara /24 para dicha red. Lo que haremos es subnetear nuestro bloque IP 72.168.152.0/22 en cuatro subredes con mascara 24 para disminuir el desperdicio. Obteniendo 4 subredes con una capacidad de 254 host c/u.

Las cuales son:

1. 72.168.1001100**00**.00000000/24 ----> 72.168.152.0/24
2. 72.168.1001100**01**.00000000/24 ----> 72.168.153.0/24
3. 72.168.100110**10**.00000000/24 ----> 72.168.154.0/24
4. 72.168.100110**11**.00000000/24 ----> 72.168.155.0/24

La asignación de bloque IP se va realizando a la red de mayor cantidad de host.

RED X

Se requieren 150 host, para ello requerimos 8 bits, ya que $2^8=256 -2 = 254 > 150$.

- 72.168.152.0/24 = ASIGNADA A **RED X**
- 72.168.153.0/24 = Sigo subneteando para generar dos /25
- 72.168.154.0/24 = LIBRE
- 72.168.155.0/24 = LIBRE

RED A

Se requieren 70 host y espera un crecimiento máximo de 20 host. Entonces se necesita $2^7=128 -2 = 126 > 90$.

Tomamos el bloque 72.168.153.0/24 => 72.168.10011001. 00000000/24

- 72.168.10011001.**0**0000000/25 -----> 72.168.153.0/25 **ASIGNADA A RED A**
- 72.168.10011001.**1**0000000/25 -----> 72.168.153.128/25 **SIGO SUBNETEANDO**

RED Y

Se requieren 35 host y se espera un crecimiento máximo de 30 host. Entonces se necesita $2^7=128 -2 = 126 > 65$.

En este caso se podría llegar a tomar el bloque con máscara /25 que quedo en el punto anterior, pero es más conveniente retomar con los 2 bloques IP que quedaron libre en un principio (/24), y así hacer un mayor aprovechamiento de los bloques.

Recordemos que a este punto los bloques libres son los siguientes:

- 72.168.153.128/25 = LIBRE
- 72.168.154.0/24 = LIBRE
- 72.168.155.0/24 = LIBRE

Tomamos el bloque 72.168.154.0/24 => 72.168.154.00000000/24

- 72.168.154.**0**0000000/25 -----> 72.168.154.0/25 **ASIGNADA RED Y**
- 72.168.154.**1**0000000/25 -----> 72.168.154.128/25 **SIGO SUBNETEANDO**

RED B

Se requieren 20 host. Entonces se necesitan $2^5=32 -2 = 30 > 20$.

Para esta red se necesita un bloque con máscara /27, entonces tomaremos uno de los bloques y lo llevaremos a la máscara deseada.

Recordemos que a este punto los bloques libres son los siguientes:

- 72.168.153.128/25 = LIBRE
- 72.168.154.128/25 = LIBRE
- 72.168.155.0/24 = LIBRE

Tomamos el bloque 72.168.153.128/25 => 72.168.153.10000000/25

Debemos desplazar nuestra máscara en 2 bits. Como resultado tendremos 4 subredes con máscara /27

- 72.168.153.**100**00000/27 -----> 72.168.153.128/27 **ASIGNADA A RED B**
- 72.168.153.**101**00000/27 -----> 72.168.153.160/27 **LIBRE**
- 72.168.153.**110**00000/27 -----> 72.168.153.192/27 **LIBRE**
- 72.168.153.**111**00000/27 -----> 72.168.153.224/27 **LIBRE**

Una vez finalizada la asignación de bloque IP para las redes pedidas, se procede con la asignación de redes para los encales punto a punto. Vamos a necesitar 5 redes punto a punto /30.

Subneteo el bloque **72.168.153.160/27** a /30.

El resultado de esto son 8 subredes.

- 72.168.153.160/30 n1-> n11
- 72.168.153.164/30 n1-> n2
- 72.168.153.168/30 n11->n6
- 72.168.153.172/30 n2->n6
- 72.168.153.176/30 n6-> n3
- 72.168.153.180/30 LIBRE
- 72.168.153.184/30 LIBRE
- 72.168.153.188/30 LIBRE

REDES LIBRES

- 72.168.155.0/24
 - 72.168.154.128/25
 - 72.168.153.192/27
 - 72.168.153.224/27
 - 72.168.153.180/30
 - 72.168.153.184/30
 - 72.168.153.188/30
-
- Asigne direcciones IP en los equipos de la topología según el plan anterior.
 - N8= 72.168.153.2/25
 - N12= 72.168.152.3/24
 - N10= 72.168.152.2/24
 - N7= 72.168.154.2/25
 - N15= 72.168.153.131/27
 - Dhcp-server= 72.168.153.130/27
 - N1 eth0= 72.168.153.1/25
 - N1 eth1= 72.168.153.165/30
 - N1 eth2= 72.168.153.161/30
 - N11 eth0= 72.168.153.162/30
 - N11 eth1= 72.168.153.169/30
 - N2 eth0= 72.168.153.166/30

- N2 eht1= 72.168.153.173/30
 - N6 eht0= 72.168.153.174/30
 - N6 eth1= 72.168.153.177/30
 - N6 eht2= 72.168.153.170/30
 - N3 eth0= 72.168.153.178/30
 - N3 eth1= 72.168.154.1/25
 - N3 eth2= 72.168.152.1/24
 - N3 eth3= 72.168.153.129/27
- Configure las tablas de rutas teniendo en cuenta las siguientes restricciones:
 - n1 deberá optar por el enlace verde solamente para rutear el tráfico dirigido a la Red X.

Tabla de rutas

Tabla de n1

Red destino	Gateway	Mask	Interface
72.168.153.0 (Red A)	0.0.0.0	/25	eth0
72.168.152.0 (Red X)	72.168.153.166	/24	eth1
72.168.154.0 (Red Y)	72.168.153.162	/25	eth2
72.168.153.128 (Red B)	72.168.153.162	/27	eth2
72.168.153.160 (n11)	0.0.0.0	/30	eth2
72.168.153.164 (n2)	0.0.0.0	/30	eth1

Tabla de n11

Red destino	Gateway	Mask	Interface
72.168.153.0 (Red A)	72.168.153.161	/25	eth0
72.168.154.0 (Red Y)	72.168.153.170	/25	eth1
72.168.153.128 (Red B)	72.168.153.170	/27	eth1
72.168.153.160 (n1)	0.0.0.0	/30	eth0
72.168.153.164 (n6)	0.0.0.0	/30	eth1

Tabla n2

Red destino	Gateway	Mask	Interface
72.168.152.0 (Red X)	72.168.153.174	/24	eth1
72.168.154.0 (n1)	0.0.0.0	/30	eth0
72.168.153.128 (n6)	0.0.0.0	/30	eth1

Tabla n6

Red destino	Gateway	Mask	Interface
72.168.153.0 (Red A)	72.168.153.169	/25	eth2
72.168.154.0 (Red Y)	72.168.153.178	/25	eth1
72.168.153.128 (Red B)	72.168.153.178	/27	eth1

72.168.152.0 (Red X)	72.168.153.178	/24	eth1
72.168.153.160 (n2)	0.0.0.0	/30	eth0
72.168.153.164 (n11)	0.0.0.0	/30	eth2
72.168.153.176 (n3)	0.0.0.0	/30	eth1

Tabla n3

Red destino	Gateway	Mask	Interface
72.168.153.0 (Red A)	72.168.153.177	/25	eth0
72.168.154.0 (Red Y)	0.0.0.0	/25	eth1
72.168.153.128 (Red B)	0.0.0.0	/27	eth3
72.168.153.160 (n6)	0.0.0.0	/30	eth0
72.168.152.0 (Red X)	0.0.0.0	/24	eth2

- Utilizando la herramienta ping, verifique conectividad entre los hosts pertenecientes a las diferentes redes de usuarios.

Ping de red A – red X

```

Terminal
Archivo  Editar  Ver  Buscar  Terminal  Ayuda
root@n8:/tmp/pycore.49727/n8.conf# ping 72.168.152.2
PING 72.168.152.2 (72.168.152.2) 56(84) bytes of data.
64 bytes from 72.168.152.2: icmp_seq=1 ttl=60 time=0.341 ms
64 bytes from 72.168.152.2: icmp_seq=2 ttl=60 time=0.145 ms
64 bytes from 72.168.152.2: icmp_seq=3 ttl=60 time=0.142 ms
64 bytes from 72.168.152.2: icmp_seq=4 ttl=60 time=0.150 ms
64 bytes from 72.168.152.2: icmp_seq=5 ttl=60 time=0.159 ms
64 bytes from 72.168.152.2: icmp_seq=6 ttl=60 time=0.132 ms
64 bytes from 72.168.152.2: icmp_seq=7 ttl=60 time=0.146 ms
64 bytes from 72.168.152.2: icmp_seq=8 ttl=60 time=0.118 ms
64 bytes from 72.168.152.2: icmp_seq=9 ttl=60 time=0.149 ms

```

Ping red A – red Y

```

Terminal
Archivo  Editar  Ver  Buscar  Terminal  Ayuda
root@n8:/tmp/pycore.49727/n8.conf# ping 72.168.154.2
PING 72.168.154.2 (72.168.154.2) 56(84) bytes of data.
64 bytes from 72.168.154.2: icmp_seq=1 ttl=60 time=1.01 ms
64 bytes from 72.168.154.2: icmp_seq=2 ttl=60 time=0.503 ms
64 bytes from 72.168.154.2: icmp_seq=3 ttl=60 time=0.505 ms
64 bytes from 72.168.154.2: icmp_seq=4 ttl=60 time=0.572 ms
64 bytes from 72.168.154.2: icmp_seq=5 ttl=60 time=0.132 ms
64 bytes from 72.168.154.2: icmp_seq=6 ttl=60 time=0.082 ms
64 bytes from 72.168.154.2: icmp_seq=7 ttl=60 time=0.240 ms
64 bytes from 72.168.154.2: icmp_seq=8 ttl=60 time=0.137 ms
64 bytes from 72.168.154.2: icmp_seq=9 ttl=60 time=0.197 ms

```

Ping red A – red B

```

Terminal
Archivo  Editar  Ver  Buscar  Terminal  Ayuda
root@n8:/tmp/pycore.49727/n8.conf# ping 72.168.153.130
PING 72.168.153.130 (72.168.153.130) 56(84) bytes of data.
64 bytes from 72.168.153.130: icmp_seq=1 ttl=60 time=0.165 ms
64 bytes from 72.168.153.130: icmp_seq=2 ttl=60 time=0.119 ms
64 bytes from 72.168.153.130: icmp_seq=3 ttl=60 time=0.142 ms
64 bytes from 72.168.153.130: icmp_seq=4 ttl=60 time=0.147 ms
64 bytes from 72.168.153.130: icmp_seq=5 ttl=60 time=0.157 ms
64 bytes from 72.168.153.130: icmp_seq=6 ttl=60 time=0.119 ms
64 bytes from 72.168.153.130: icmp_seq=7 ttl=60 time=0.150 ms
64 bytes from 72.168.153.130: icmp_seq=8 ttl=60 time=0.147 ms
64 bytes from 72.168.153.130: icmp_seq=9 ttl=60 time=0.149 ms
64 bytes from 72.168.153.130: icmp_seq=10 ttl=60 time=0.122 ms

```

TTL

1. Utilizando el comando traceroute, realice una traza entre el host n8 y n10, tanto utilizando UDP como ICMP. ¿Qué diferencias tiene cada método y en qué casos utilizaría cada uno?

La aplicación envía 3 paquetes con TTL valor 1 hacia el host destino; tan pronto llegue al primer router en la trayectoria, TTL decrementa en 1 dando como resultado TTL=0, entonces el router responde con un mensaje ICMP de tipo 11 ("time exceeded") que indica que el datagrama ha caducado y es descartado. Ahora se envían otros 3 paquetes con TTL configurado en 2. Esto hace que el segundo router devuelva un mensaje ICMP tipo 11. Este proceso continua hasta que los paquetes alcancen al host destino.

La diferencia entre estos 2 métodos es, con ICMP el primer paquete que envía es un mensaje ICMP tipo 8 (echo request). Hasta llegar al host destino los saltos intermedios van a notificar con mensaje ICMP tipo 11 (time exceeded) como mencionamos anteriormente. Una vez alcanzado el destino responde con mensaje ICMP tipo 0 (echo reply).

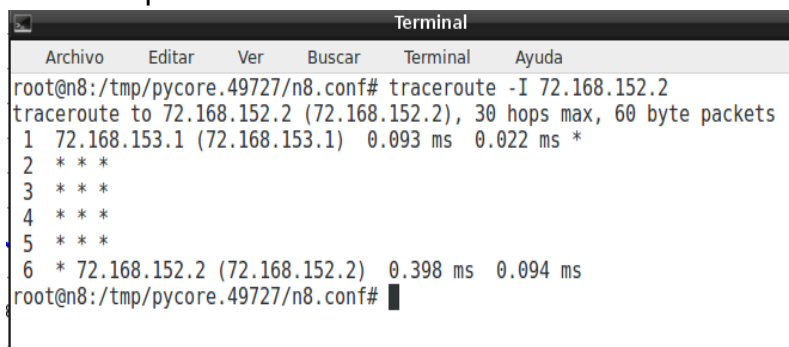
Por otro lado, el método empleado con UDP envía una secuencia de datagramas UDP a un de puerto alto para que sea no válido en el host remoto (por cada envío el puerto cambia). Al igual que el método anterior en los saltos intermedios responde con mensaje ICMP tipo 11 hasta llegar al host destino. Puesto que estos datagramas intentan acceder a un puerto no valido del host destino, los mensajes de puerto inalcanzables indica que se llegó al destino.

El mensaje de puerto inalcanzable es un mensaje ICMP de tipo 3 código 3.

La razón por la que debería elegir a uno u otro método, es que puede suceder que el firewall filtre los paquetes y solo me muestre "* * *", en este caso debería optar por elegir el otro método.

Llevándolo a la práctica se puede ver que al intentar hacer traceroute entre n8 y n10 los resultados son los siguientes.d

Traceroute por ICMP



```
Terminal
Archivo  Editar  Ver  Buscar  Terminal  Ayuda
root@n8:/tmp/pycore.49727/n8.conf# traceroute -I 72.168.152.2
traceroute to 72.168.152.2 (72.168.152.2), 30 hops max, 60 byte packets
 1 72.168.153.1 (72.168.153.1) 0.093 ms 0.022 ms *
 2 * * *
 3 * * *
 4 * * *
 5 * * *
 6 * 72.168.152.2 (72.168.152.2) 0.398 ms 0.094 ms
root@n8:/tmp/pycore.49727/n8.conf#
```

Traceroute por UDP

```

Terminal
Archivo  Editar  Ver  Buscar  Terminal  Ayuda
root@n8:/tmp/pycore.49727/n8.conf# traceroute -U 72.168.152.2
traceroute to 72.168.152.2 (72.168.152.2), 30 hops max, 60 byte packets
 1 72.168.153.1 (72.168.153.1)  0.065 ms  0.015 ms  0.012 ms
 2 72.168.153.166 (72.168.153.166)  0.030 ms  0.019 ms  0.017 ms
 3 72.168.153.170 (72.168.153.170)  0.042 ms  0.026 ms  0.024 ms
 4 72.168.153.178 (72.168.153.178)  0.052 ms  0.033 ms  0.032 ms
 5 72.168.152.2 (72.168.152.2)  0.051 ms  0.042 ms  0.037 ms
root@n8:/tmp/pycore.49727/n8.conf#

```

2. Realice un ping entre n8 y n5 y determine el valor inicial del campo TTL capturando tráfico en la interfaz eth0 del host n8.

No se puede realizar un ping a n5 porque es un switch que es un dispositivo de capa de enlace, por lo tanto no conoce IP, y no tengo una dirección para comunicarme con el. Tomo el host n7, que es un host de la red a la que pertenece el switch.

Entonces el valor de TTL dado por el comando ping entre n8(72.168.153.2/25) y n7(72.168.154.2/25) es de **64**.

1	0.000000000	72.168.153.2	72.168.154.2	ICMP	98 Echo (ping) request	id=0x0052, seq=1/256, ttl=64 (reply in 2)
2	0.000085000	72.168.154.2	72.168.153.2	ICMP	98 Echo (ping) reply	id=0x0052, seq=1/256, ttl=60 (request in 1)

3. A través de la capturas de tráfico, determine en qué momento el router decrementa el valor del TTL.

Por cada salto a un router el valor del TTL disminuye en uno. Tomando como ejemplo la comunicación de origen: n8 y destino: n7. Inicialmente cuando sale de n8, lo hace con un TTL=1, al llegar al próximo router (n1 eth0) le informa a n8 un mensaje ICMP tipo 11 que el TTL es 0. La siguiente vez se enviará con TTL valor 2 (en este caso llegara hasta n11 eth0 y notificará a n8 ICMP tipo 11), posteriormente con valor 3... así hasta llegar a la máquina destino, la cual no contestará con "time exceeded".

En este caso el metodo de traceroute utilizado fue el ICMP, entonces cuando llegue al destino,

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	72.168.153.2	72.168.152.2	ICMP	74	Echo (ping) request id=0x0051, seq=1/256, ttl=1 (no response found!)
2	0.000019	72.168.153.1	72.168.153.2	ICMP	102	Time-to-live exceeded (Time to live exceeded in transit)
3	0.000047	72.168.153.2	72.168.152.2	ICMP	74	Echo (ping) request id=0x0051, seq=2/512, ttl=1 (no response found!)
4	0.000052	72.168.153.1	72.168.153.2	ICMP	102	Time-to-live exceeded (Time to live exceeded in transit)
5	0.000063	72.168.153.2	72.168.152.2	ICMP	74	Echo (ping) request id=0x0051, seq=3/768, ttl=1 (no response found!)
6	0.000067	72.168.153.1	72.168.153.2	ICMP	102	Time-to-live exceeded (Time to live exceeded in transit)
7	0.000100	72.168.153.2	72.168.152.2	ICMP	74	Echo (ping) request id=0x0051, seq=4/1024, ttl=2 (no response found!)
8	0.000125	72.168.153.166	72.168.153.2	ICMP	102	Time-to-live exceeded (Time to live exceeded in transit)
9	0.000136	72.168.153.2	72.168.152.2	ICMP	74	Echo (ping) request id=0x0051, seq=5/1280, ttl=2 (no response found!)
10	0.000147	72.168.153.166	72.168.153.2	ICMP	102	Time-to-live exceeded (Time to live exceeded in transit)
11	0.000156	72.168.153.2	72.168.152.2	ICMP	74	Echo (ping) request id=0x0051, seq=6/1536, ttl=2 (no response found!)
12	0.000166	72.168.153.166	72.168.153.2	ICMP	102	Time-to-live exceeded (Time to live exceeded in transit)
13	0.000175	72.168.153.2	72.168.152.2	ICMP	74	Echo (ping) request id=0x0051, seq=7/1792, ttl=3 (no response found!)
14	0.000208	72.168.153.170	72.168.153.2	ICMP	102	Time-to-live exceeded (Time to live exceeded in transit)
15	0.000217	72.168.153.2	72.168.152.2	ICMP	74	Echo (ping) request id=0x0051, seq=8/2048, ttl=3 (no response found!)
16	0.000233	72.168.153.170	72.168.153.2	ICMP	102	Time-to-live exceeded (Time to live exceeded in transit)
17	0.000242	72.168.153.2	72.168.152.2	ICMP	74	Echo (ping) request id=0x0051, seq=9/2304, ttl=3 (no response found!)
18	0.000258	72.168.153.170	72.168.153.2	ICMP	102	Time-to-live exceeded (Time to live exceeded in transit)
19	0.000267	72.168.153.2	72.168.152.2	ICMP	74	Echo (ping) request id=0x0051, seq=10/2560, ttl=4 (no response found!)
20	0.000297	72.168.153.178	72.168.153.2	ICMP	102	Time-to-live exceeded (Time to live exceeded in transit)
21	0.000307	72.168.153.2	72.168.152.2	ICMP	74	Echo (ping) request id=0x0051, seq=11/2816, ttl=4 (no response found!)
22	0.000327	72.168.153.178	72.168.153.2	ICMP	102	Time-to-live exceeded (Time to live exceeded in transit)
23	0.000337	72.168.153.2	72.168.152.2	ICMP	74	Echo (ping) request id=0x0051, seq=12/3072, ttl=4 (no response found!)
24	0.000356	72.168.153.178	72.168.153.2	ICMP	102	Time-to-live exceeded (Time to live exceeded in transit)
25	0.000366	72.168.153.2	72.168.152.2	ICMP	74	Echo (ping) request id=0x0051, seq=13/3328, ttl=5 (reply in 26)
26	0.000401	72.168.152.2	72.168.153.2	ICMP	74	Echo (ping) reply id=0x0051, seq=13/3328, ttl=60 (request in 25)

4. Utilizando la herramienta hping3 desde n8 envíe un datagrama a n7 con TTL=1. ¿Qué mensaje recibe? ¿Por qué?

El mensaje recibido es "TTL 0 during transit from ip=72.168.153.1". Como el ttl es de valor 1, al llegar al primer salto (n1) decrementa su valor en 1, y con esto llega a 0. Entonces 72.168.153.1 (n1) le informa a n8 no pudo seguir con el envío mediante un mensaje ICMP tipo 11.

Nota: Adjunte capturas de tráfico para cada uno de los incisos

Checksum

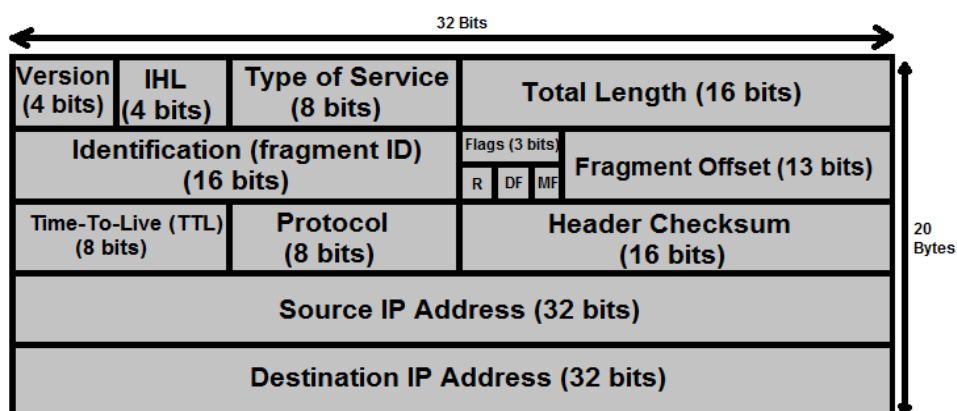
1. ¿Cómo se calcula el checksum en IPv4? ¿Qué campos se utilizan?

El checksum ayuda a los routers a detectar errores de bit en un datagrama IP recibido.

Tiene que volver a calcularse y almacenarse en cada router, ya que el campo TTL y, posiblemente, también el campo de opciones puede cambiar.

Para calcular el checksum es necesario dividir el encabezado IP con palabras de 16 bits y sumamos cada una de ellas y finalmente hacemos un complemento a1 de la suma. Recordar que para realizar el calculo los 16 bits correspondientes al header checksum van en 0.

Los campos utilizados son los primeros 20 bytes del header.



2. ¿Qué acción se lleva a cabo cuando falla dicha comprobación?

Un router calcula el checksum para cada datagrama IP recibido y detecta una condición de error si el valor de checksum incluida en la cabecera del datagrama no coincide con el checksum calculado. Normalmente, los routers descartan los datagramas en los que se ha detectado que existe un error.

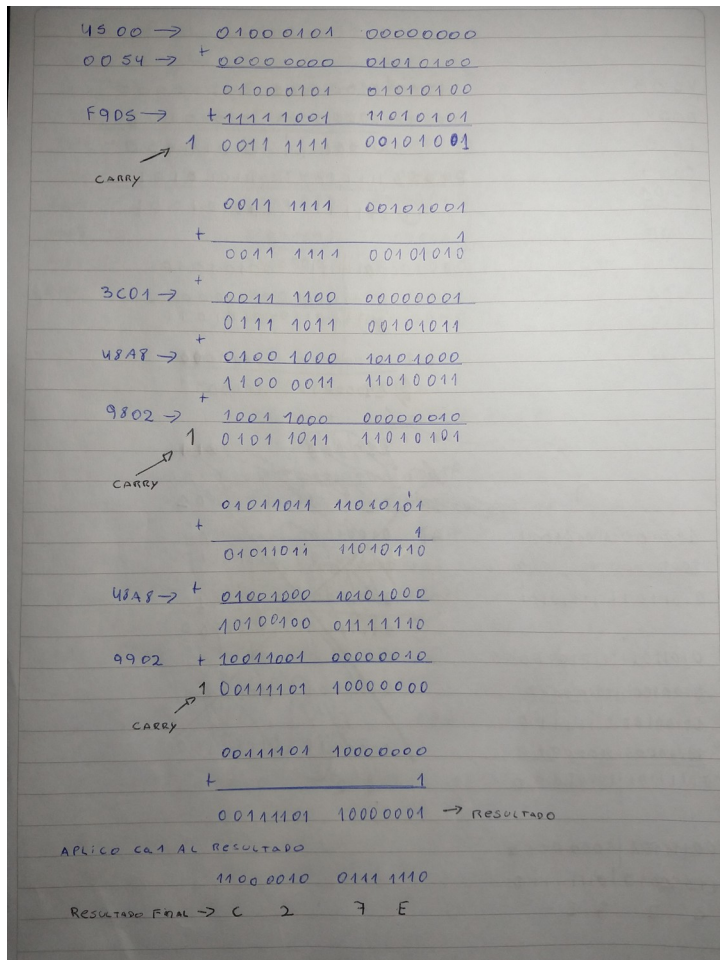
Ademas utiliza un mensaje ICMP tipo 12 para informar que el header IP no es correcto.

3. Utilizando una captura de tráfico tomada de la topología, realice el cálculo de checksum y verifique si dicho valor corresponde con el del campo del header.

Utilizando la captura "ping-n8-n10" y el frame numero 6.

El header IP esta definido por los valores "4500 0054 F9D5 0000 3C01 **C27E** 48A8 9802 48A8 9902"

C27E es el valor del checksum, para realizar el cálculo manual lo dejaremos en valor 0.



Terminado el calculo manual de checksum, tenemos que el resultado es C27E, que es igual al checksum calculado.

4. ¿Cómo se calcula el checksum en IPv6?

En IPv6 no tenemos campo checksum. La tarea de verificar que los datos llegan bien se delegó a la capa de transporte (en UDP va a tener que ser obligatoria). El objetivo es reducir el tiempo de procesamiento; en IPv4, por el campo TTL (el cual cambiaba en cada salto), la suma de comprobación necesitaba ser calculada en cada router. Ahora esto se eliminó, y será mas rápido. El campo TTL de IPv4 fue reemplazado por el campo "límite de saltos".