Trabajo Practico Especial 2018- Parte 1

Comunicación de Datos 1

Autores:

Álvarez Maximiliano (maxi25294@gmail.com)
Talú Bernabé(berni.talu@hotmail.com)

Ayudante designado: Mailen.

27/09/2018

Índice

Introducción

En esta primer parte del trabajo práctico fuimos capaces de crear, desde sus raíces, una red y adentrarnos en el mundo de la comunicación de datos. Lo logramos mediante la representación VLSM y con la herramienta Core. Desde esta etapa logramos dividir dicha red en sub-redes para distinto uso y con distinta cantidad de usuarios o hosts. Pudimos utilizar el comando “ping” entre 2 PCs de la misma sub-red para visualizar su comunicación y luego, con la herramienta Wireshark, poder analizar el envío de paquetes y como se conforma cada uno.

Desarrollo y resolución

Enunciados:

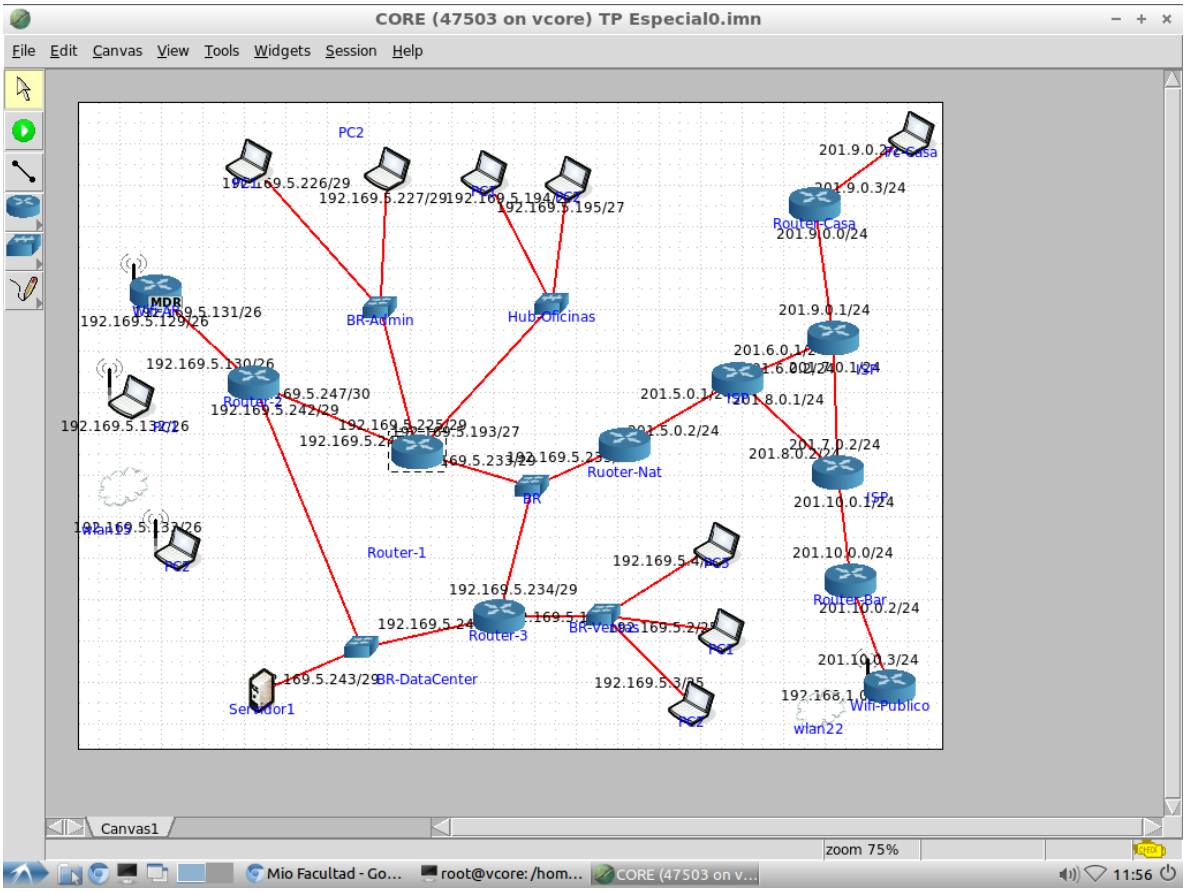
1. *Para la cantidad de conexiones proyectadas para cada una de las redes, realice una asignación de direcciones IP utilizando VLSM. Considere que las direcciones privadas se encuentren en el rango 192.169.X.0 a 192.169.X.255, donde X es el número de grupo que se les asignó.*

   El grupo que se nos asignó es el grupo número 5, por lo tanto, el rango será desde 192.169.5.0 a 192.169.5.255. La asignación la hicimos ordenando de mayor a menor la cantidad de entradas que se requerían y así poder asignar a medida que se va necesitando. Para “subnetear” (dividir la red en subredes), realizamos los siguientes pasos:
   - Identificamos la máscara de red actual (/24)
   - Aplicamos la formula 2^n-2 >= Host; esto se utiliza para medir la cantidad de hosts o usuarios que se necesitan por subred, a lo que el -2 hace referencia a la dirección base y la dirección Broadcast (primera y última dirección de cada bloque, no utilizables).
   - Luego, calculando la n, hallamos la máscara de red (32-n). A medida que menos hosts o usuarios se necesiten, más grande será la máscara de red.
   - Por último, inicia la segunda subred en la dirección de broadcast+1, que será la dirección base de la próxima subred.

2. *Realice una tabla en donde se indiquen cada una de las subredes resultantes, indicando el nombre de cada red, su dirección base, la máscara, y el rango que incluye cada bloque.*

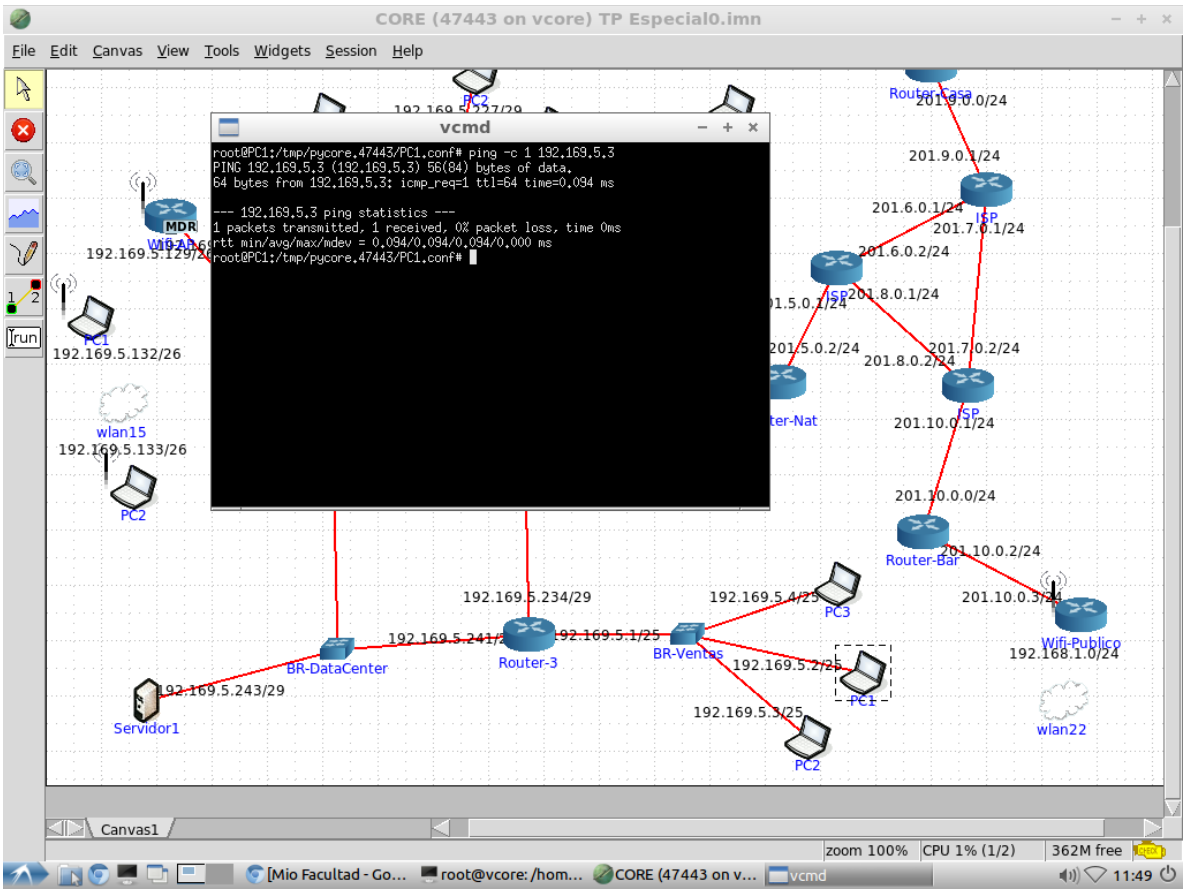
	Sub-red	Dir. Base	Máscara	Rango	Broadcast
1	Ventas	192.169.5.0	/25	[1;126]	127
2	Wifi-AP	192.169.5.128	/26	[129;190]	191
3	Oficina	192.169.5.192	/27	[193;222]	223
4	Admin	192.169.5.224	/29	[225;230]	231
5	BR	192.169.5.232	/29	[233;238]	239
6	BR-DataC	192.169.5.240	/29	[241;246]	247
7	R1 a R2	192.169.5.248	/30	[249;250]	251

3. *Implemente la red propuesta en el emulador CORE con la disposición de equipos que actualmente se tienen conectados. Considere la asignación IP realizada en el ejercicio 1, y la colocación de direcciones públicas en donde corresponda (considere que se debe utilizar el comando ifconfig para configurar cada una de las interfaces de los routers, mientras que en los host se puede realizar la configuración de la interfaz colocando la Ip que corresponda utilizando la opción IPv4 address de la pantalla de configuración del host.*

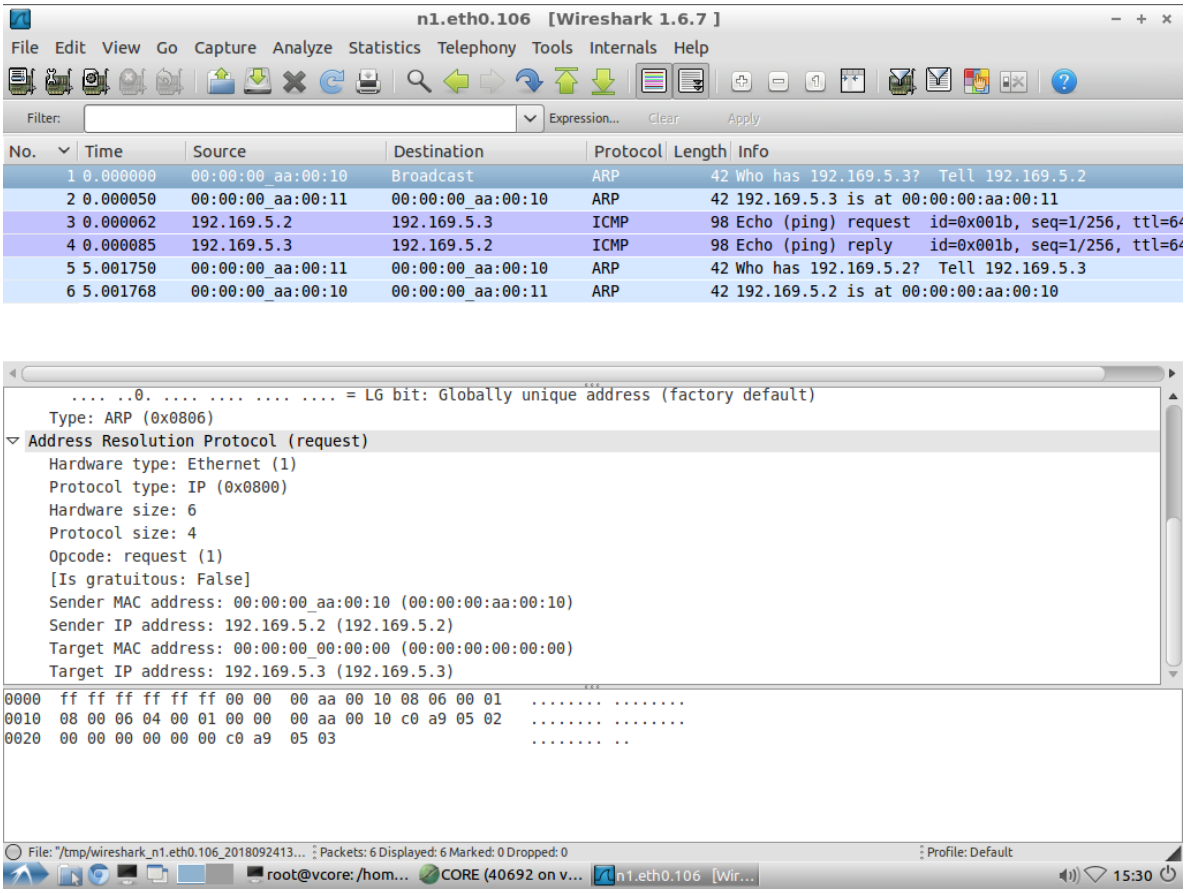


I. Configuramos los ISP con las direcciones públicas dadas y la privada que ingresa al Router-Nat
II. Colocamos los routers requeridos con los switchs y hubs correspondientes realizando las respectivas conexiones.
III. Los routers wifi los conectamos mediante las “Wireless LAN”
IV. Colocamos las PCs necesarias dadas por el trabajo especial.
V. Luego de asignar las IPs correspondientes, en los routers, colocamos el comando ifconfig con sus interfaces en la pestaña de startup. Esto se lleva a cabo con el fin de configurar la interfaz de red de cada uno de ellos, asignándole el comando “up” marcamos la interfaz como disponible para que se pueda utilizar por la capa IP. Se realiza en cada router, con cada interfaz de estos. Ej. ifconfig eth0 192.169.5.3 up.

4. *Ejecute el comando Ping con la opción –c 1 entre dos host de la subred ventas. Utilizando la herramienta Wireshark, inspeccione los paquetes que se generan en el origen y destino (captura de la interfaz origen e interfaz destino). Identifique los campos más importantes de cada paquete (dirección origen, dirección destino, protocolo, ttl, tipo de paquete, etc.)*



- Se utilizó el comando “ping” entre la PC1 y la PC2 de la subred ventas para verificar la comunicación entre ellos dos. Con el comando –c 1, el cual envía sólo un paquete.

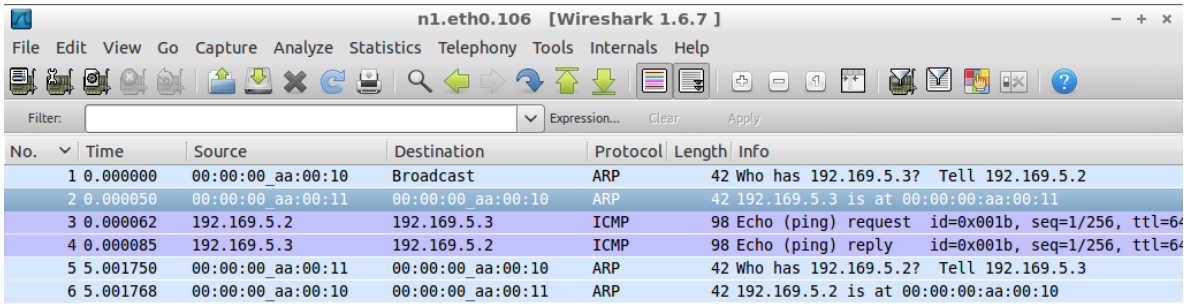


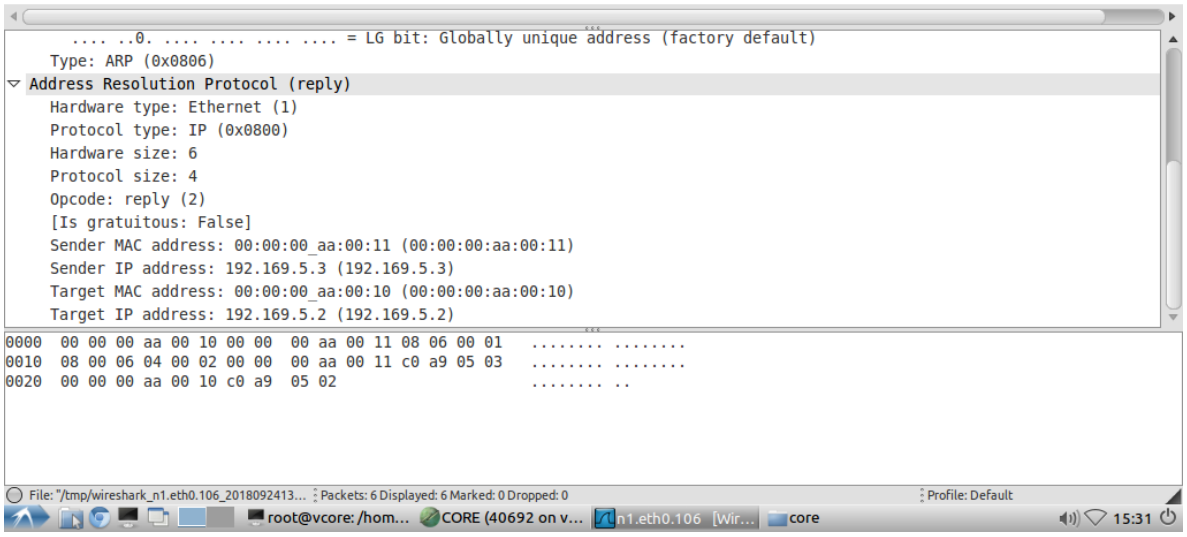
- Utilizando la herramienta Wireshark analizamos el envío de paquetes cuando se utilizó el comando “ping” entre las dos PCs de la subred Ventas. Esta información está “empaquetada” en hexadecimal, y en la interfaz gráfica de dicha herramienta lo traduce para que se pueda entender amigablemente. Analizamos 3 de los 6 paquetes mostrados, sobre todo por la importancia del número 3 y número 4 que son los que llevan “data”.

  Descomponiendo cada paquete hexadecimal, la finalidad de cada conjunto. Por ejemplo, en el paquete número 1 (Broadcast):

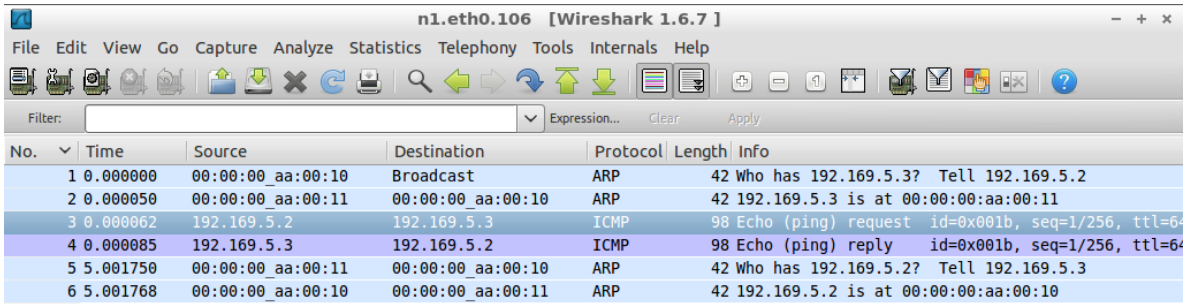
- ff ff ff ff ff ff: Dirección de destino
- 00 00 00 aa 00 10: Dirección Origen
- 08 06: ARP(0x0806) Type (Protocolo)
- 00 01: Hardware type: Ethernet (1)
- 08 00: Protocol type: IP (0x0800)
- 06: Hardware Size
- 04: Protocol Size
- 00 02: Opcode: request (1)
- 00 00 00 aa 00 10: Sender MAC address

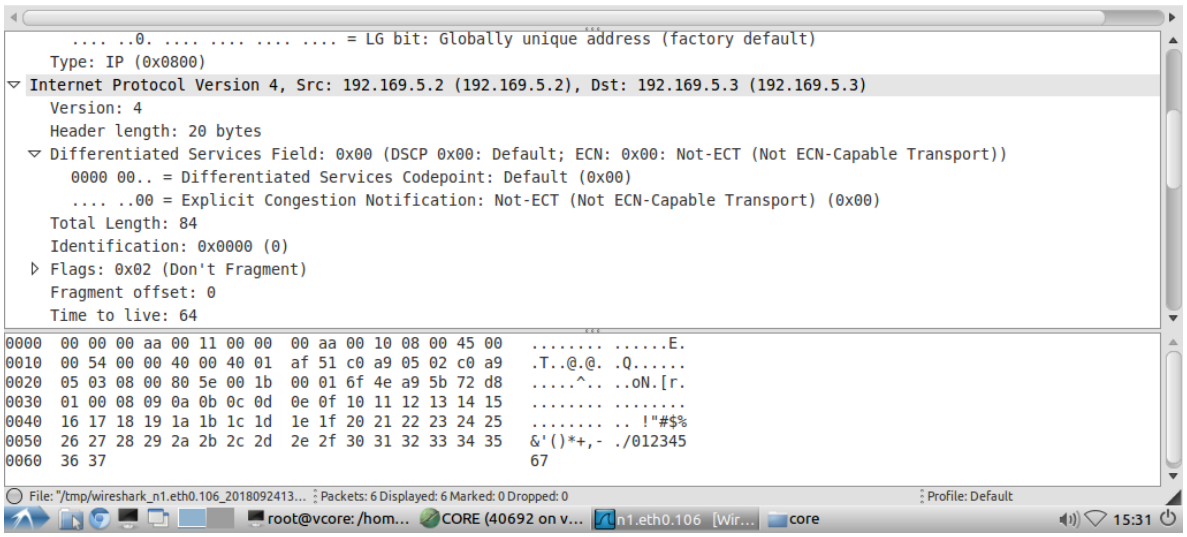
- c0 a9 05 02: Sender IP address
- 00 00 00 00 00 00: Target MAC Address
- c0 a9 05 03: Target IP Address



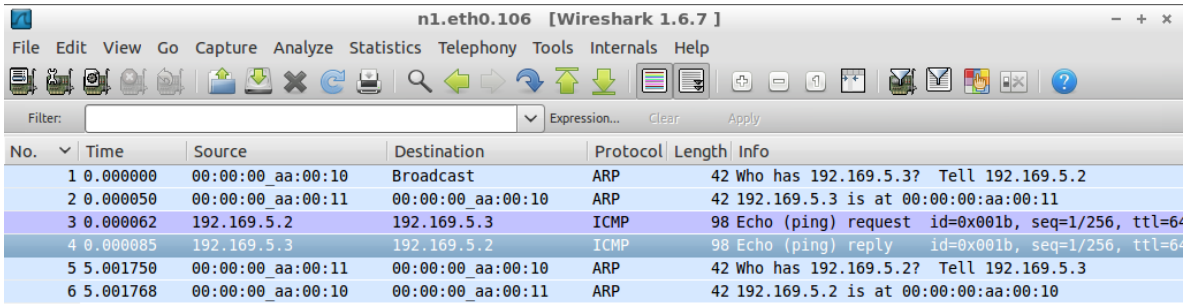


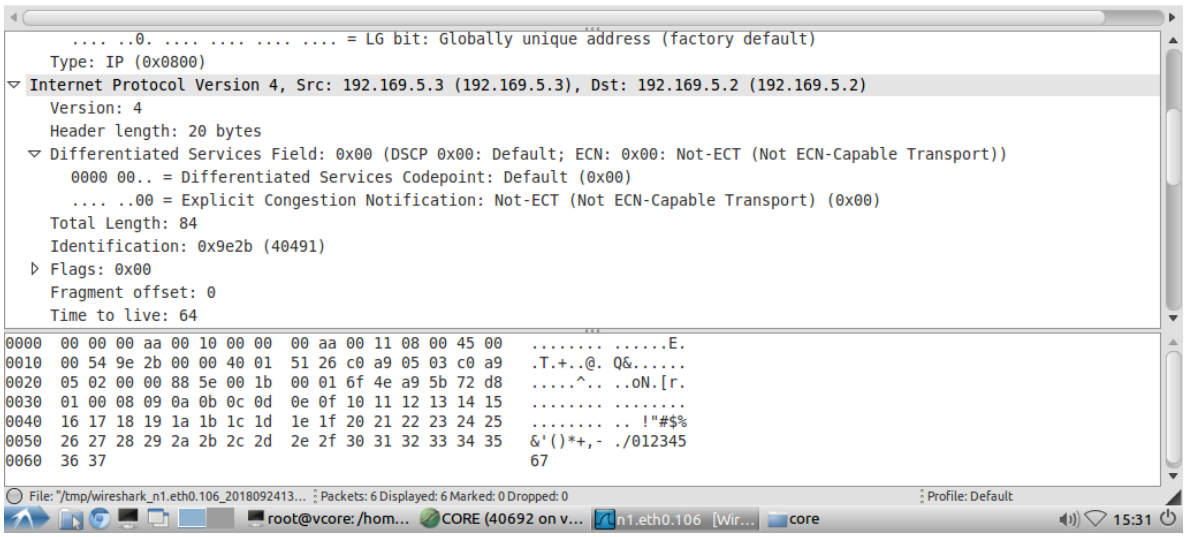
- 00 00 00 aa 00 10: Dirección de destino
- 00 00 00 aa 00 11: Dirección origen
- 08 06: ARP(0x0806) Type (Protocolo)
- 00 01: Hardware type: Ethernet (1)
- 08 00: Protocol type: IP (0x0800)
- 06: Hardware Size
- 04: Protocol Size
- 00 02: Opcode: reply (2)
- 00 00 00 aa 00 11: Sender MAC address
- c0 a9 05 03: Sender IP address
- 00 00 00 aa 00 10: Target MAC Address
- c0 a9 05 02: Target IP Address





- 00 00 00 aa 00 11: Dirección Destino
- 00 00 00 aa 00 10: direccion origen
- 08 00: type ip(0x0800)
- 45: Header Length: 20 bytes
- 00: Not-ECT (0x00)
- 00 54: Total length: 84
- 00 00: Identification: 0x0000 (0)
- 40 00: Fragment Offset: 0
- 40: time to live: 64
- 01: protocol: icmp (1)
- af 51: header checksum 0xaf51 bad:false
- c0 a9 05 02: header checksum source
- c0 a9 05 03: header checksum destination
- 08: type (echo ping request)
- 00: code
- 80 5e: checksum 0x805e correct
- 00 1b: identifier (le): 6912
- 00 01: sequence number (LE): 256 (0x0100)
- El resto es "Data": 56 bytes





- 00 00 00 aa 00 10: Dirección Destino
- 00 00 00 aa 00 11: Dirección Origen
- 08 00: Type IP (0x0800)
- 45: Header Length: 20 bytes
- 00: Not-ECT(0x00)
- 00 54: Total Length: 84
- 9e 2b: Identification
- 00 00: Fragment Offset
- 40: Time to live
- 01: Protocol: ICMP (1)
- 51 26: Header Checksum
- c0 a9 05 03: Header Checksum Source
- c0 a9 05 02: Header Checksum Destination
- 00: Echo “ping” request
- 00: Code
- 88 5e: Checksum
- 00 1b: Identifier (le)
- 00 01: Sequence Number (LE)

Conclusión

Con la realización del trabajo llegamos a diversas conclusiones:

1. El método VLSM si bien fue un gran avance para la comunicación de datos, cuando lo llevamos a grandes escalas, observamos que no es eficaz. Esto se debe a que las subredes se dividen obligatoriamente en una potencia de 2, y por ejemplo, si se necesitan exactamente 256 equipos, sin contar con la dirección base y broadcast , se debería realizar una subred una disponibilidad de 512 equipos, lo que hace que queden demasiadas direcciones sin ocupar.
2. Si se posee una red chica, es improductivo dividirla en muchas subredes debido a que se pierden demasiadas redes en proporción a las obtenidas.
3. Mediante la herramienta Wireshark pudimos ver la composición de cada paquete y al descomponerlo, la disponibilidad de los datos en su envío, las “preguntas” que se hacen entre receptor y origen para reconocerse, y los movimientos que se realizan para que el paquete con su “data” llegue a destino.