Trabajo Práctico Especial

**Configuración y Desarrollo de Aplicaciones en Redes**



**Marsico, Cristian -** [**cristianmarsico84@gmail.com**](mailto:cristianmarsico84@gmail.com)

**Van Der Vluet, Julio -** [**juliovdv@gmail.com**](mailto:juliovdv@gmail.com)

Prof.: Gómez Leandro - [lgomez@rec.unicen.edu.ar](mailto:lgomez@rec.unicen.edu.ar)

Octubre de 2020

Tecnicatura Universitaria en Desarrollo de Aplicaciones Informáticas

Facultad de Ciencias Exactas - UNICEN

Indice

**Introducción2**

Objetivo General2

Objetivos Específicos2

**PRIMER PARTE**

**Desarrollo**3

VLSM, Asignaciones de IP y Tablas de red 3

Core 4

Ruteo 5

**Resultados6**

VLSM, Asignaciones de IP y Tablas de red 6

Core 7

Ruteo 9

**Conclusión**11

**SEGUNDA PARTE**

Punto 5 3

Punto 64

Punto 7 5

Punto 8 6

Punto 97

Punto 109

Punto 11 6

Punto 127

Punto 13 9

Punto 14 9

**Conclusión**11

Introducción

En el presente trabajo a partir de una topología de red se quiere crear las conexiones para comunicar un parque industrial, la casa del jefe y las respectivas fábricas.

La infraestructura ISP utiliza las bandas 201.0.1.0/24 y 201.0.2.0/24 para los clientes actuales (casa y parque industrial).

La casa del jefe dispone de un router wifi que maneja la banda 192.168.10.0/24.

En a primer parte del informe se explica la creación de una red para un parque industrial y sus respectivas fábricas, usando las herramientas de VLSM, asignación de IP´s y tablas de ruteo, y plasmarlo en el simulador CORE.

En la segunda parte se agrega la casa del jefe, estructurando el informe según los ítems solicitados por la cátedra.

El parque industrial dispone de un *datacenter* (R2, R3 y R4) que puede alojar hasta 120 servidores, de los cuales actualmente hay dos conectados. Este parque cuenta con 11 fábricas conectadas a una red troncal que necesita disponer de 300 direcciones IP. En el presente trabajo se configuran dos fábricas (A y B).

La fábrica A tiene conectadas 2PC con un *switch* (SW3) que soporta 45 dispositivos y un router (R7) que debe soportar como mínimo 80 equipos. Además, cuenta con un centro de cómputo conectado a través de otro *switch* (SW4) que tiene un servidor conectado y se prevé tener hasta 10.

La fábrica tiene dos PC conectadas por un *HUB* que cuenta con 8 interfases. El *switch* (SW5) de la misma soporta hasta 65 dispositivos. También cuenta con un centro de cómputos conectador por el SW6 que tiene conectar un equipo y se prevé tener hasta 24.

En la topología de red utilizada el ISP utiliza las bandas 201.0.1.0/24 y 201.0.2.0/24 para los clientes actuales. Las direcciones privadas deben encontrarse en la red 172.2.0.0/20

Dentro del desarrollo primero se crean los VLSM realizando las asignaciones de la IP y las tablas de red. Luego se muestra la carga en el emulador CORE y la ejecución de este. Para finalizar el desarrollo se describe de qué manera se realizaron las tablas de ruteo y las simplificaciones de rutas.

En la sección de resultados siguiendo el orden del desarrollo se muestran los resultados de cada una de las acciones llevadas a cabo para cumplir con los objetivos planteados.

Objetivo General

* Creación de una red para un parque industrial y dos de sus fábricas (A y B).

Objetivos Específicos

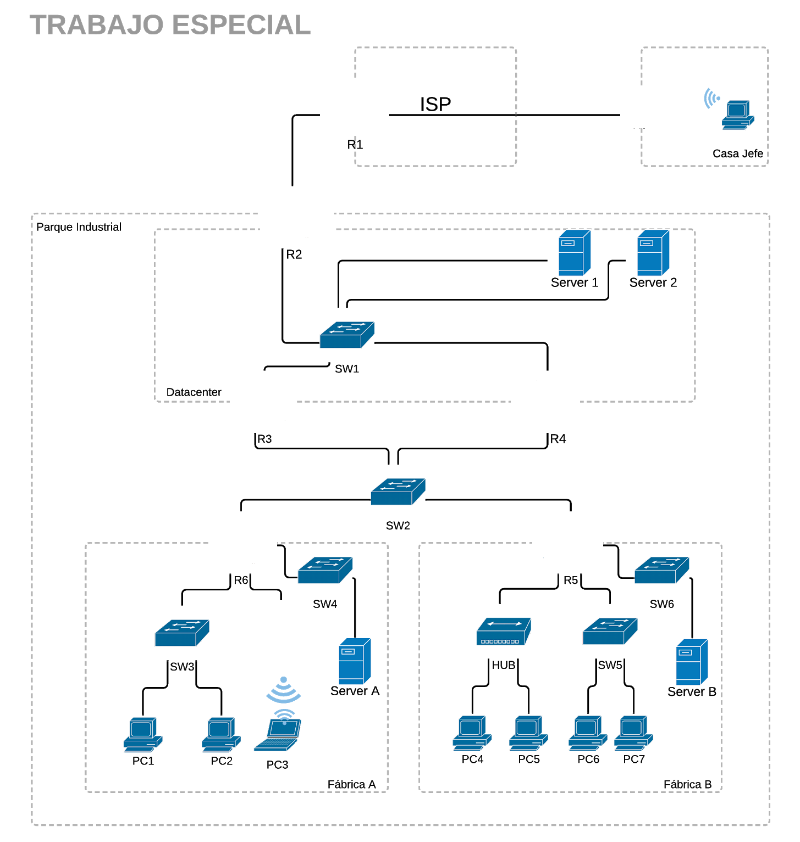
* Realizar asignaciones de direcciones IP, creando un VLSM para el parque industrial y uno particular para cada fábrica.
* Realizar una tabla que indique las subredes resultantes.
* Implementar la red en el emulador CORE.
* Configurar las interfases y rutas de los routers y hosts, minimizando la cantidad de entradas en las tablas de ruteo.

**PRIMERA PARTE**

Desarrollo

VLSM, Asignaciones de IP y Tablas de red

Para poder identificar claramente las redes y subredes usamos la imagen de la topología de la red y borramos los router, como muestra *Imagen 1,* así quedan claramente diferenciadas, las redes y subredes que componen el trabajo.



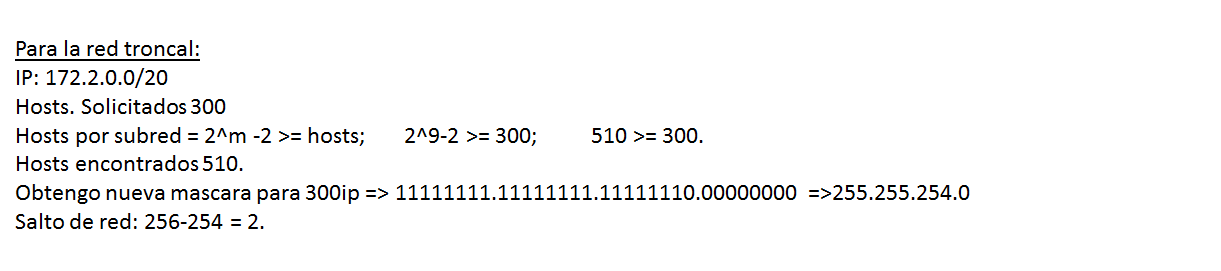
*Imagen 1: Topología de Red*

La red que más nos costó identificar fue la que comunica a R6 con R7.

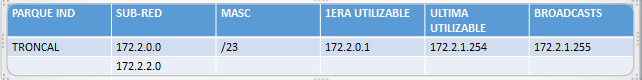
Tomando la IP: 172.2.0.0/20 ya dada, y según los datos requeridos se fueron obteniendo las direcciones bases para los lugares físicos. Primero se crea una tabla la cual representa a la red general.

Calculamos la cantidad de dispositivos o *hosts* necesarios, ordenando los mismos de mayor a menor según los requerimientos de dispositivos necesarios para cada sector y con el objetivo de optimizar el aprovechamiento de bloques.

Por ejemplo: Para la red troncal se solicitaban 300 *hosts*, para calcular las subredes utilizamos fórmula correspondiente 2m- 2 ≥ *hosts*, 9 bits libres son suficientes para cubrir los *hosts* solicitados, con lo cual cambia la máscara que pasa a ser /23. Haciendo la conversión a números decimales con la nueva máscara se calculó el salto de la red, obteniendo la nueva IP base para la fábrica A.



Esta nueva IP la utilizamos para determinar el *Broadcast* y la última utilizable.



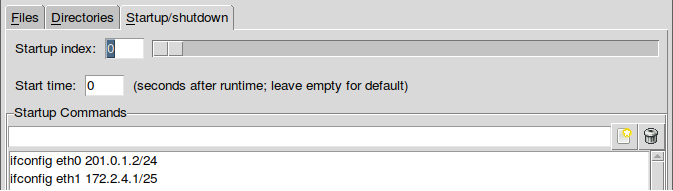
Para el resto de las redes se procedió de la misma manera.

Con los datos obtenidos del VLSM se armaron las tablas de redes y subredes.

Para calcular las direcciones IP de las subredes utilizamos las direcciones base calculadas anteriormente, que tiene que soportar los hosts requeridos por la red. Siguiendo el mismo procedimiento que el utilizado para el cálculo de las direcciones base se configuró una nueva tabla para cada subred.

Core

Para implementar las redes obtenidas en el simulador Core utilizamos las tablas obtenidas en los puntos anteriores y asignamos los IP’s a cada *router* y *host* utilizamos el comando **< ifconfig [interfaz] [dirección] >,** como se muestra en *Imagen 2*y los asignamos de manera correlativa empezando por el primer número asignable.



*Imagen 2: Carga de comandos*

Surgieron problemas con la conectividad de las redes *wi-fi* (ninguna PC se conectaba al *router*), lo solucionamos desactivando los SSH de las Pc y servidores.

El traceroute permite diagnosticar las redes para mostrar las posibles rutas o caminos de paquetes y medir la latencia de tránsito y los tiempos de ida y vuelta a través de redes de protocolos de internet.

El traceroute se utilizó para visualizar todos los routers por los que atraviesa cada paquete. El comando utilizado es **traceroute <IP>.** En la sección de resultados se muestran las imágenes de la ejecución de este comando.

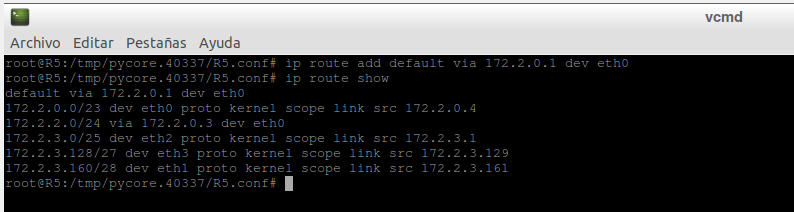
Ruteo

Para realizar las tablas de ruteo probamos en el Core por terminal, con el comando **<ip route add <Net/IP>/<Mask> vía <Gateway> dev <IntX>>**, como muestra la *imagen 3,* y luego lo agregamos en la tabla *Startup Commands* de *UserDefined*. En esta imagen podemos ver las rutas cargadas automáticamente como por ejemplo **172.2.0.0/23 dev eth0 proto kernel scope link src 172.2.0.4**, la ruta cargada por defecto **default via 172.2.0.1 dev eth0** que además se cargó manualmente.

Comenzamos con rutas puntuales y mucha cantidad de entradas en las tablas y las pudimos reducir cuando entendimos conceptos como *default* y subredes.

De los bloques contiguos que poseen la misma máscara, podemos fusionarlos; ampliando la cantidad de *hosts* y reduciendo direcciones para simplificar tablas de ruteo.

Analizando los *routers* 3 y 4, nos dimos cuenta de que con los dos llegábamos a los mismos lugares, con lo cual decidimos descartar el 4 y utilizar el camino de *router* 3, de esta manera también estamos generando una única vía de comunicación entre las redes.



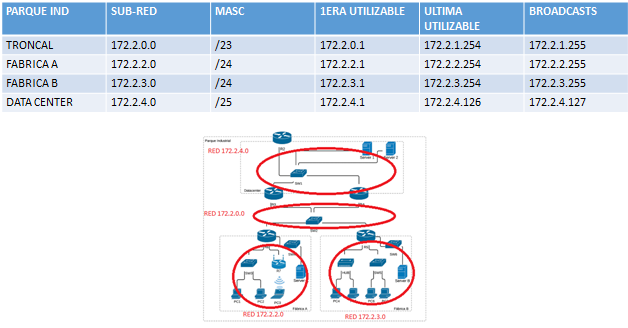
*Imagen 3: tablas de ruteo con CORE*

# Resultados

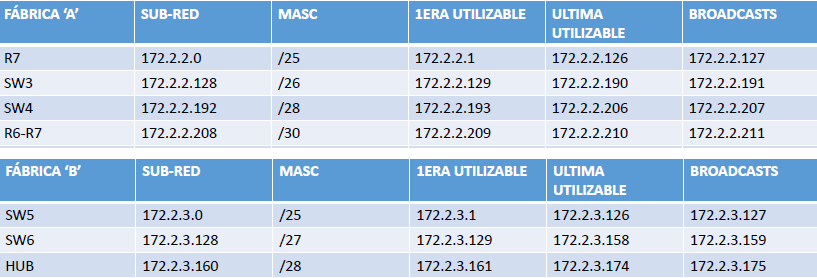
VLSM, Asignación IP

En la siguiente tabla se muestran las direcciones bases obtenidas para la red troncal del parque industrial, el *datacenter* y las dos fábricas analizadas, indicando las ISP, las máscaras y los *broadcasts*.

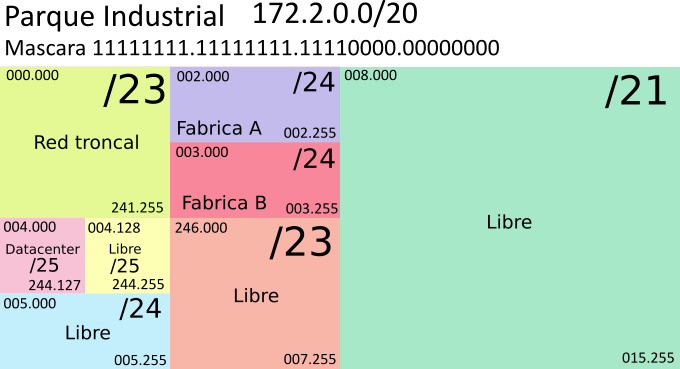
En la primera tabla vemos los resultados de las direcciones base de cada red. En el gráfico encontramos en el círculo superior el *datacenter*, el círculo del medio corresponde a la red troncal y los inferiores a las redes de las fábricas (a la izquierda la fábrica A y a la derecha la B).

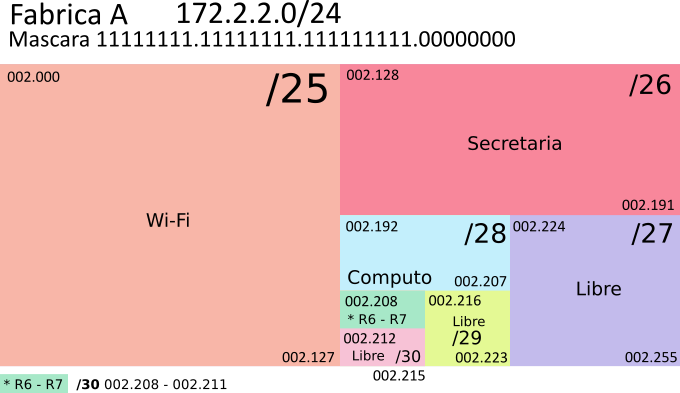
****

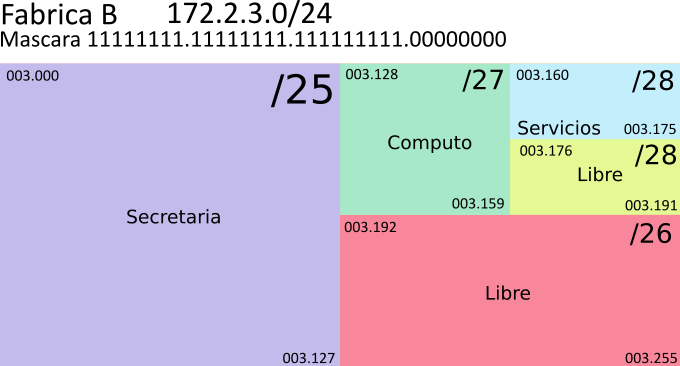
Una vez obtenidas las direcciones bases para cada red, se ordenan nuevamente los hosts de mayor a menor y se procede nuevamente a aplicar un VLSM para obtener las subredes. A continuación, a manera de ejemplo se muestran las tablas con las subredes de cada fábrica.



A continuación, se muestra la distribución gráfica, en la cual se puede visualizar lo que ocupa cada bloque.

****

****

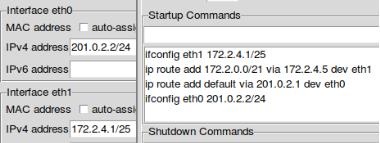
****

Core

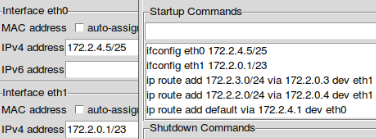
En las siguientes imágenes se muestra el resultado final en el Core, el cual puede verse en funcionamiento en el archivo \*.imn que adjuntamos con este informe.

Carga de asignación de ip y rutas

**R2**

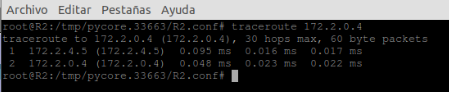
****

**R3**

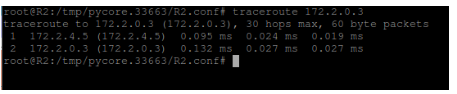
****

Traceroute

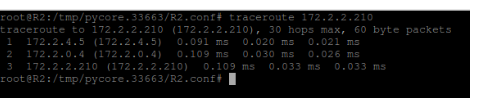
Comunicación de R2 a R6 y distintos caminos



R2 a R5



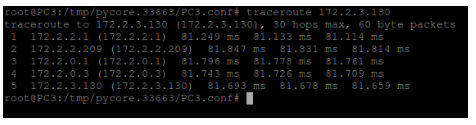
R2 a R7



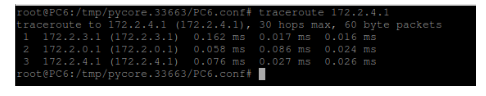
pc2 fabrica ‘A’ a server2 fabrica ‘B’



pc3 fabrica ‘A’ a serverB fabrica ‘B’



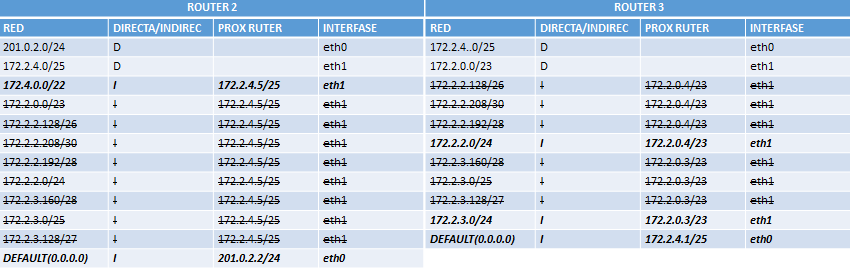
pc6 fabrica a R2



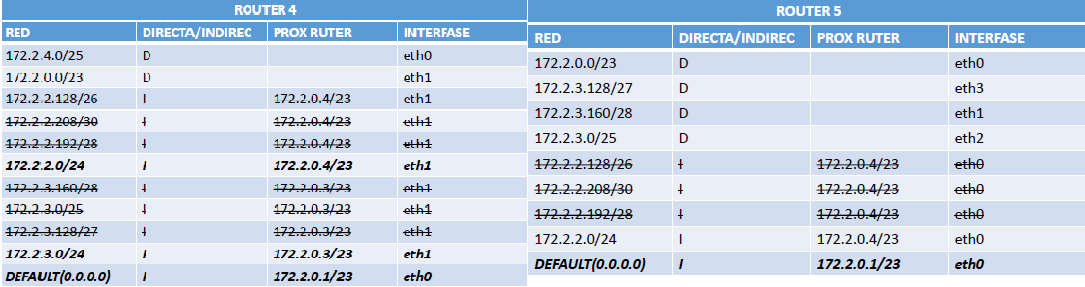
Ruteo

En las siguientes tablas puede verse el resultado final, minimizando entradas, de las tablas de ruteo para cada *router*.

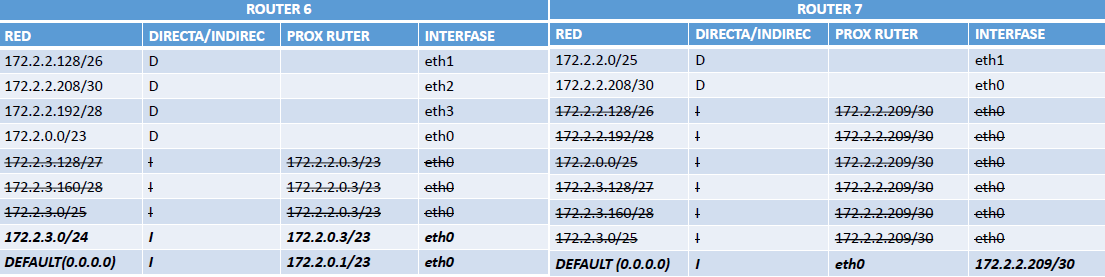
En R2 se simplifican 8 direcciones IP base, quedando solo una indirecta, dos directas y el *default*. En R3 se simplifican 6 direcciones IP base, quedando dos indirectas, dos directas y el *default*.



En R4 se simplifican 5 direcciones IP base, quedando tres indirectas, dos directas y el *default*. De todas maneras, este *router* no se utilizó. En R5 se simplifican 3 direcciones IP base, quedando solo una indirecta, cuatro directas y el *default*.



En R6 se simplifican 3 direcciones IP base, quedando solo una indirecta, cuatro directas y el *default.* En R7 se simplifican 6 direcciones IP base, quedando dos directas y el *default.*



# Conclusión

En el resultado obtenido se ve, como aplicando *Subnetting* (VLSM), se aprovechan mejor las direcciones IPs y como se simplifican las tablas de ruteos. Se simplificaron 33 direcciones IP con lo cual se logró una reducción en las tablas.

Se logro la comunicación de todo el parque industrial, su *datacenter* con la red troncal y las fábricas. La simulación en la máquina virtual CORE fue exitosa, cumpliendo los objetivos planteados.

**SEGUNDA PARTE**

* **Punto 5 - Configurar el R2 para que solo tengan acceso a internet los equipos conectados a la red *Datacenter*, Fábrica A y Fabrica B.**

En *services* del Router 2 (frontera), se agregan las siguientes reglas:

< **iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE** >

La regla se agrega a la tabla *NAT*, en la cadena *POSTROUTING*, ya que se aplica cuando ya ha pasado por el ruteo, en la interfaz de salida eth0 se aplica la acción enmascarar (el cambio de una IP privada por una pública) para permitir salida a internet.

Luego se agregan las siguientes reglas para dar acceso a internet a los equipos de la red.

Fábrica A, B y *datacenter* y rechazar todos los demás

< **iptables -t filter -A FORWARD -s 172.2.2.0/24 -j ACCEPT** >

< **iptables -t filter -A FORWARD -s 172.2.3.0/24 -j ACCEPT** >

< **iptables -t filter -A FORWARD -s 172.2.4.0/25 -j ACCEPT** >

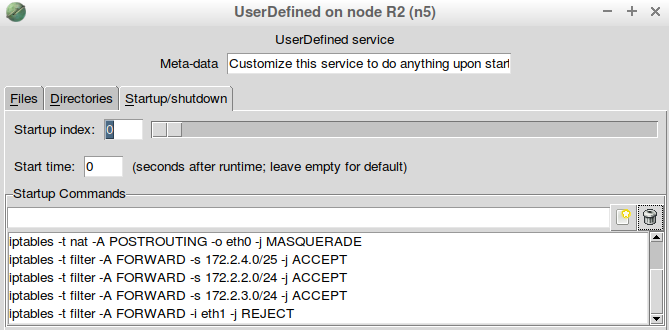
La regla se agrega a la tabla *filter*, en la cadena *FORWARD*, ya que se aplica cuando atraviesa el *router*, aceptando (correspondientes a fabrica A, B y *datacenter*) los paquetes provenientes del *source* establecido en el comando (-s IP).

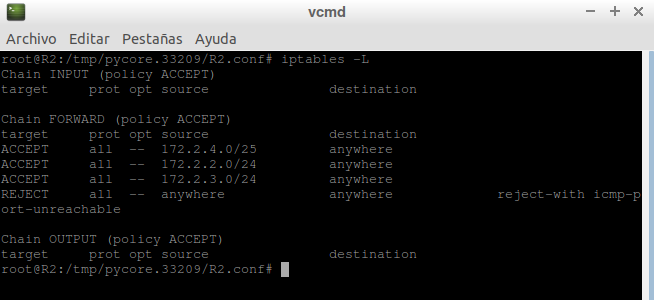
Con el comando < **iptables -t filter -A FORWARD -j REJECT** > rechazamos todos los paquetes de todos los destinos.

La regla se agrega a la tabla *filter*, en la cadena *FORWARD*, ya que se aplica cuando atraviesa el *router*.

Hay que tener en cuenta el orden de las reglas, debido que si se hubiese colocado el comando de rechazo al comienzo las reglas de aceptar no surgirían efecto.

Podemos ver las siguientes capturas de configuración de comandos y listado de reglas.

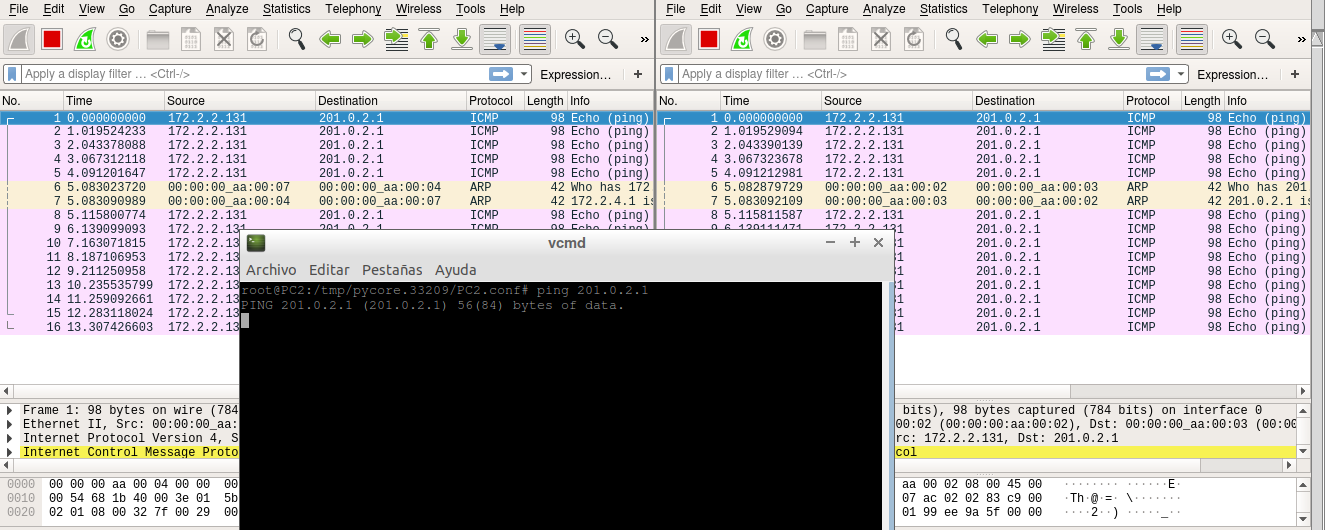




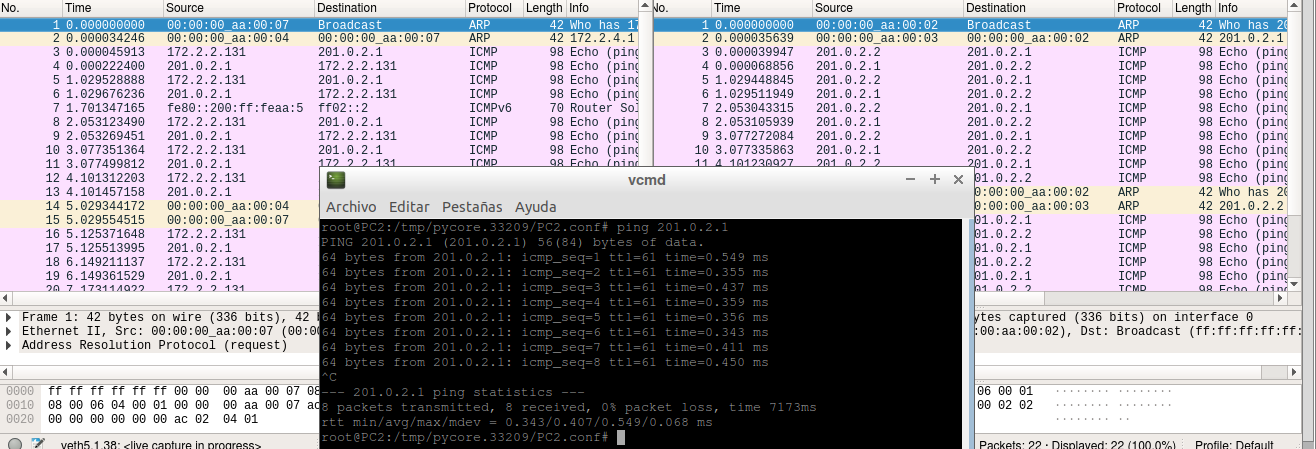
Comando iptables -L (listado y orden de reglas)

Las siguientes fotos se pueden apreciar como actúa el enmascaramiento;

en la primera foto sin el comando, el *ping* atraviesa R2 con la misma IP privada (de PC2) y al ser privada el ISP no sabe responderle debido que no puede *routear* IP's privadas.



En la siguiente imagen podemos apreciar como R2 convierte la IP privada en publica (201.0.2.2) y así lograr una respuesta de *ISP*.



*Wireshark* izquierdo R2 entrada por interfaz privada,

lado derecho salida por interfaz pública.

* **Configurar el R5 para que todos los dispositivos del área de servicio conectados a través del HUB solo se puedan conectar al resto de los equipos de la Fábrica B y también a los equipos del centro de cómputo conectados en el *Datacenter*.**

En Servicios del *Router* 5 (Fábrica B), se agregan las siguientes reglas:

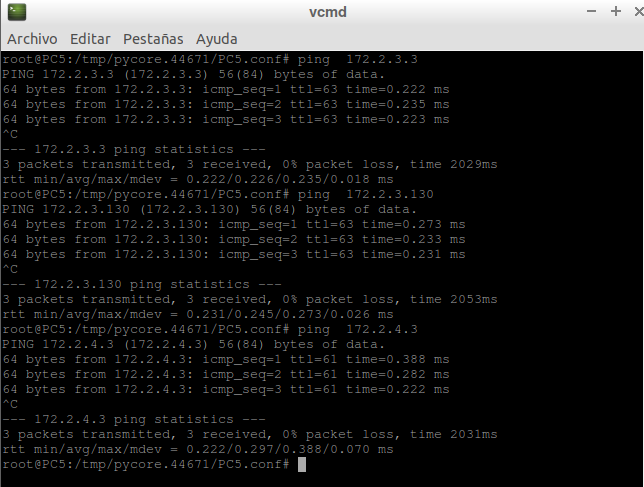
< **iptables -t filter -A FORWARD -s 172.2.3.160/28 -d 172.2.3.0/24 -j ACCEPT** >

< **iptables -t filter -A FORWARD -s 172.2.3.160/28 -d 172.2.4.0/25 -j ACCEPT** >

En este caso la regla se agrega a la tabla *filter*, en la cadena *FORWARD*,

ya que se aplica cuando atraviesa el *router*, se acepta todo lo que proviene del origen y tenga de destino la IP detallada (en los primeros dos comandos permitimos las subredes de la fábrica B y en el tercero las del *datacenter*).

En la siguiente foto se ve como un *host* conectado al *switch* alcanza a las subredes de la fábrica B y a un *host* del *datacenter*.

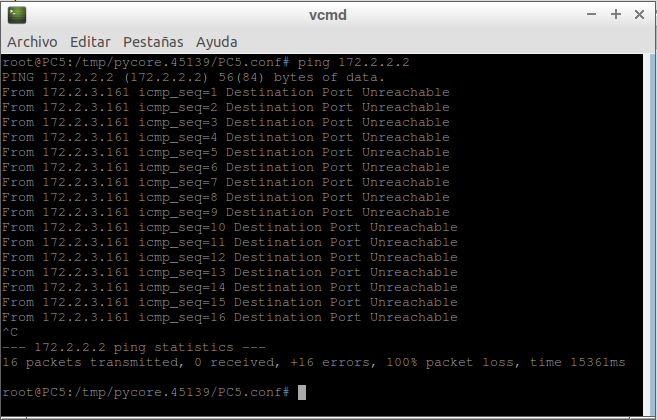


Luego procedemos con el siguiente comando:

< **iptables -t filter -A FORWARD –s 172.2.3.160/28 -j REJECT** >

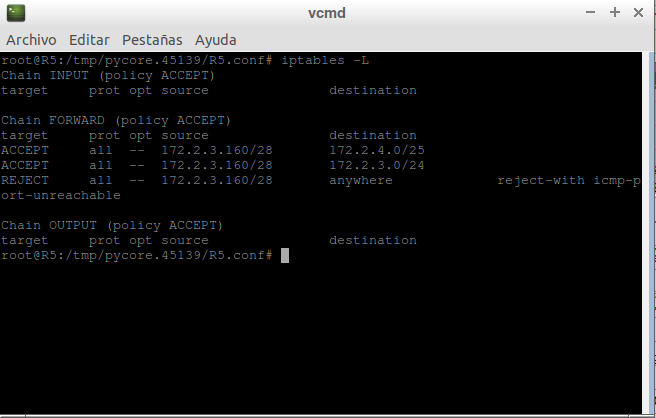
Esta regla (aplicada en R5) rechazamos todo lo proveniente de la sub red 172.2.3.160/28.

se puede ver en a continuación el *ping* de un *host* perteneciente a la subred que posee el *HUB* a un equipo de la fábrica A (PC2), el resultado es el rechazo.



El siguiente comando podemos apreciar el orden de las reglas, como lo mencionamos anteriormente el orden es fundamental.

Analogía: si se hubiese colocado *REJECT* primero, los demás comandos no surgirían efecto y rechazaría todo.



* **El ítem número 7 resulta similar al del 5, se pide dar acceso a internet a determinada subred, lo cual debemos generar una máscara para transformar las IP privadas que salgan de R8 (Red de la casa del jefe) en públicas.**

< **iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE** >

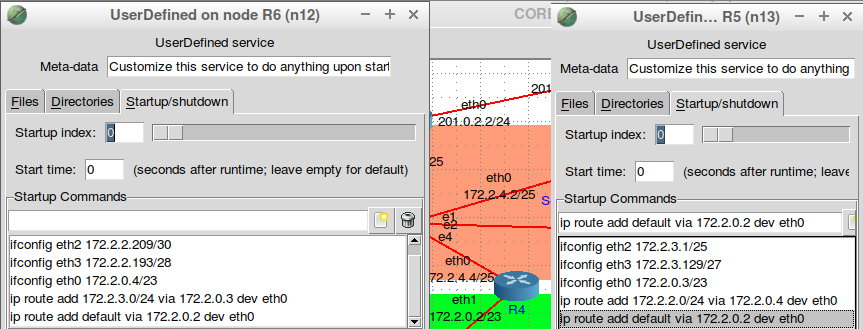
La regla se agrega a la tabla NAT, en la cadena *POSTROUTING*, ya que se aplica cuando ya ha pasado por el ruteo, para la condición de salida de la interfaz eth0, la acción enmascarar.

* **Por cuestiones de seguridad, todo el tráfico proveniente de las fábricas y dirigido al *Datacenter* debe pasar por el router R4, y el tráfico dirigido a Internet, debe pasar por el router R3.**

En el *Router* 2 (frontera), se configura la tabla de ruteo para que todo el tráfico proveniente de internet pase por el *Router* 3.

Se Configuran las tablas de ruteo del *Router* 5 y *Router* 6 para que, desde las Fábricas,

todo el tráfico dirigido al *datacenter*, pasen por el *Router* 4 y el tráfico de internet por el *Router* 3



NetCat

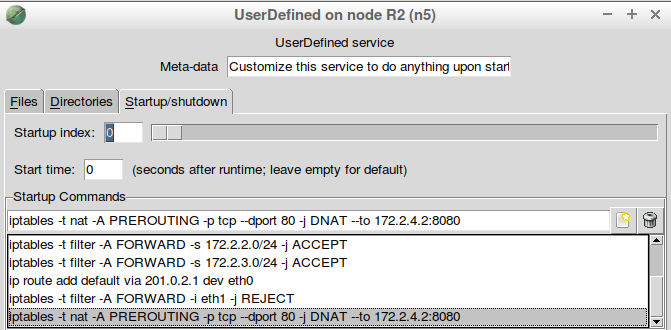
* **Configure la red de manera de poder enviar el mensaje “Hola Data Center” desde PC-Casa hasta Server 1, utilizando Netcat. Tenga en cuenta configurar el reenvío de paquetes en Router 2. Considere que el puerto que está abierto en el Router 2 es el 80, mientras que el servicio en el Server 1 está corriendo en el puerto 8080.   Luego, replique la conexión desde la PC1. Indique para ambos casos, ¿qué dirección IP y qué puertos se deben utilizar?**

En Router 2 se indica que las comunicaciones a su puerto 80, la redirija al puerto 8080 del Server1

< **iptables -t nat -A PREROUTING -p tcp --dport 80 -j DNAT --to 172.2.4.2:8080** >

La siguiente imagen detalla el comando utilizado.

En forma de entender el comando: todo lo proveniente al puerto 80 lo redirija a quien tenga el 8080, en este caso al Server 1.



La regla se agrega en la tabla *NAT*, ya que debemos traducir direcciones y puertos,

en la cadena *PREROUTING*, porque debe ejecutarse antes de pasar por el router.

Mediante *NetCat* se abre el servicio en el *Server 1* con el comando:

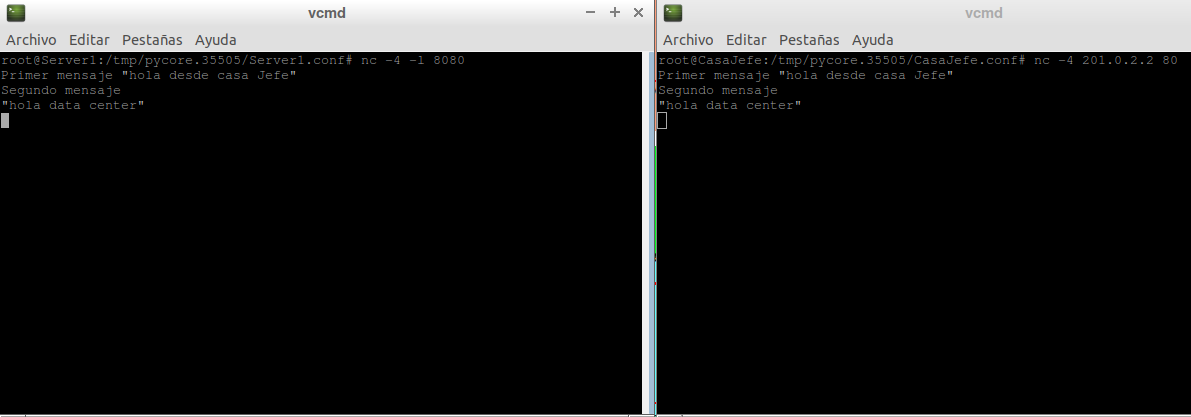
**< nc -4 -l 8080** > (con esta instrucción el servidor queda “escuchando” en el puerto 8080)

Una vez configurado el server1 se levantar el siguiente comando en la Casa Jefe:

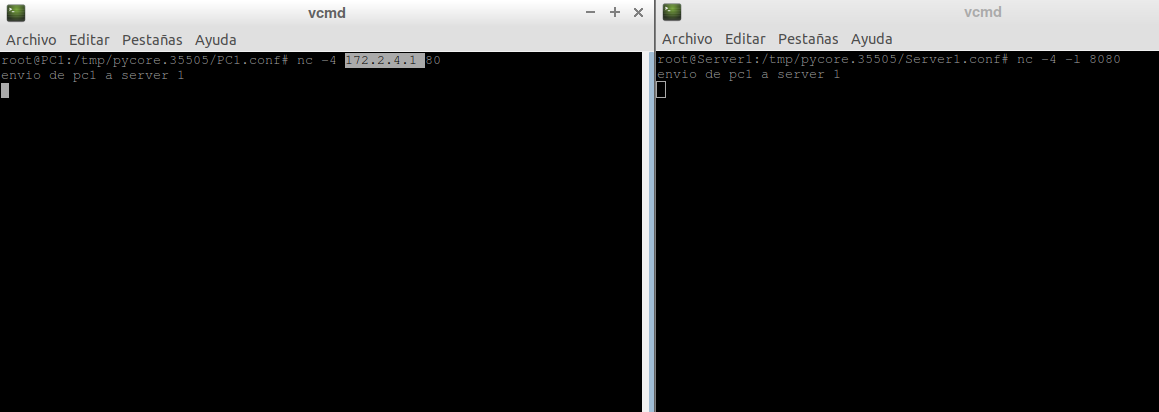
< **nc -4 201.0.2.2 80** >

Se indicar cual va a ser la IP de entrada en Router 2 precedido por el puerto al que se quiere enviar, el *router* estando configurado va a saber que lo que provenga con el puerto 80 lo va a tener que redirigir al quien tenga el 8080.

Ingresado ambos comandos ya los *hosts* se pueden comunicar por chat como se puede ver en la siguiente imagen, esto es posible debido a la configuración de la acción *DNAT*.



Para comunicar la PC1 con el Server 1 hay que replicar el primer comando estableciendo "una escucha" en Server1 y el segundo comando hay que modificar la IP correspondiente a la interfaz por la que se desea enviar el mensaje y de esta forma ya pueden establecer chat PC1 con Server 1.



* **En el ítem 10 se pueden observas las diferencias de funcionamiento en un *Hub* y un *Switch*. se solicita un *ping* de PC4 (conectada a un *Hub*) a PC6 (conectada a un *Switch*) pertenecientes a distintas subredes y a su vez comparte otros hosts.**

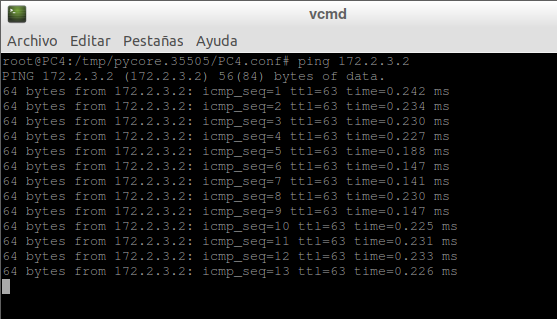
Para corroborar tal comportamiento, se debe abrir la herramienta Wireshark en las 4 PC’s (PC4, 5, 6 y 7).

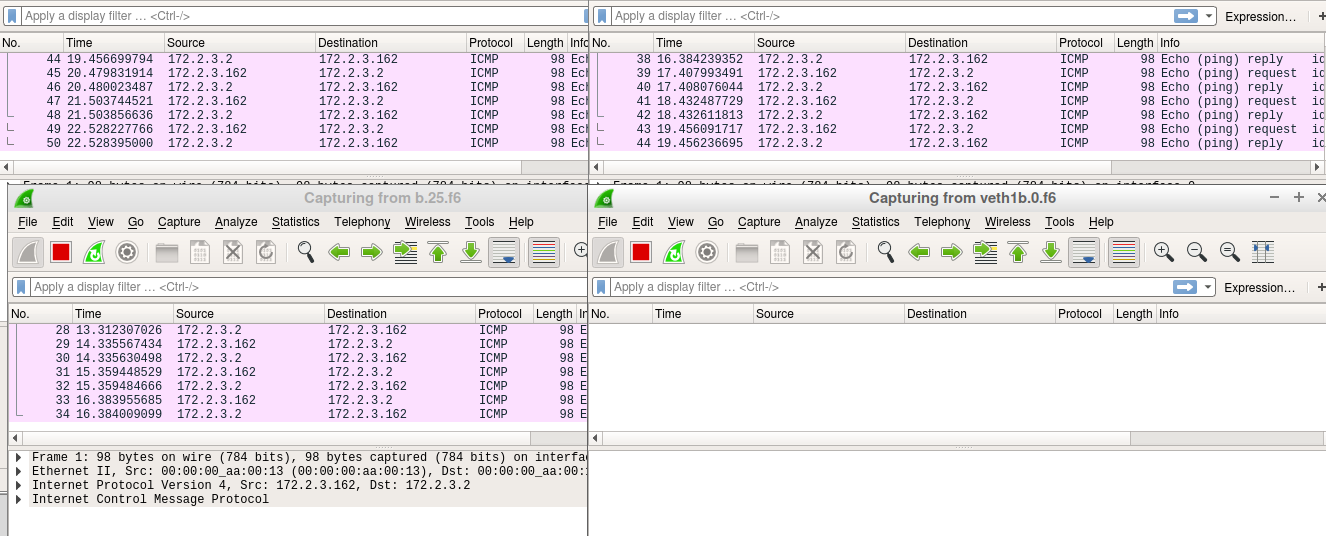
**Ejecutar un ping de PC4 a PC6**

< **ping -c 1 172.2.3.2** >

Como se comprueba en la siguiente captura el paquete no solo lo recibe la pc6, sino que también la PC5, pero con la salvedad que la PC7 no recibe ningún paquete.

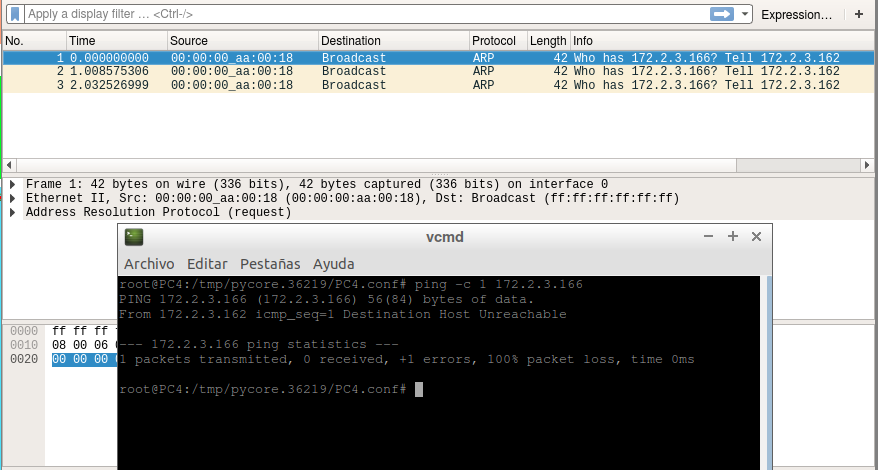
Aquí claramente demuestra que el *hub* reparte el paquete a toda la subred mientras que el *swith* filtra el paquete enviándolo solamente a *host* destino.





* **El ítem 11 solicita un < ping -c 1 [ IP] > (en la subred 172.2.3.160 perteneciente a la PC4) a un equipo que no se encuentre conectado a la misma y luego realizar otro *ping* a la red del *Datacenter* a otro *host* que no se encuentre conectado.**

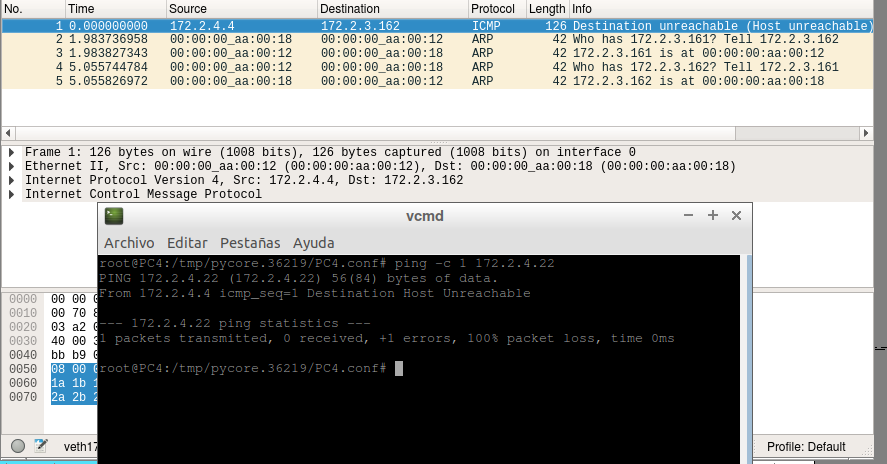
En el primer caso muestra que el destino es inalcanzable y un mensaje de error indicando que el paquete se perdió. analizando con *Wireshark* quien da la respuesta es Router 5 y también se observan tres mensajes *ARP* cada uno correspondiente a una interfaz del *Hub*, donde le consulta a broadcast quien posee la IP 172.23.166 pero el mismo no genera una respuesta debido a que no hay un *host* conectado que posea dicha IP.



En el segundo caso hacemos un *ping* -c 1 + IP a un host que no se encuentre conectado en el *Datacenter*, donde también se muestra un error de paquete perdido.

*Wireshark* de la PC4, muestra mensajes *ICMP*, el primero haciendo la solicitud a la IP

172.2.3.166, al no encontrar el *host*, el Router 4 devuelve el mensaje de *“Destination unreachable (Host unreachable)”*.



* **Realizar pruebas utilizando el comando ping entre los siguientes puntos y mediante la utilización de Wireshark, analice el camino seguido por los paquetes generados, adjuntando al informe las capturas de pantalla correspondientes:**

**a) Desde un equipo conectado a la Fábrica A hasta la dirección privada del R2**

**b) Desde un equipo conectado a la Fábrica A hasta la dirección pública del R2**

**c) Desde Pc4 a la interfaz pública del Router 8**

**Listado de pings**:

< -**c 1 172.2.4.1>** (de PC1 a IP privada Router 2)

**< -c 1 201.0.2.2**> (de PC1 a IP pública Router 2)

**< -c 1 201.0.1.2>** (de PC4 a IP pública Router 8)

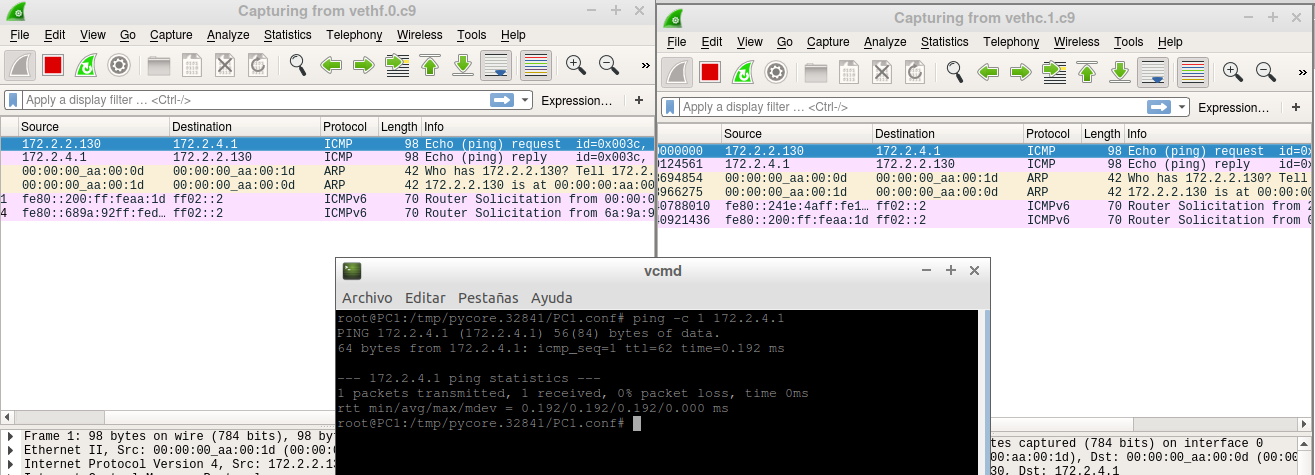
**ping -c 1 172.2.4.1 (de pc1 a IP privada R2)**

*Wireshark* de la PC1, muestra mensajes *ICMP*, el primero haciendo la solicitud a la IP

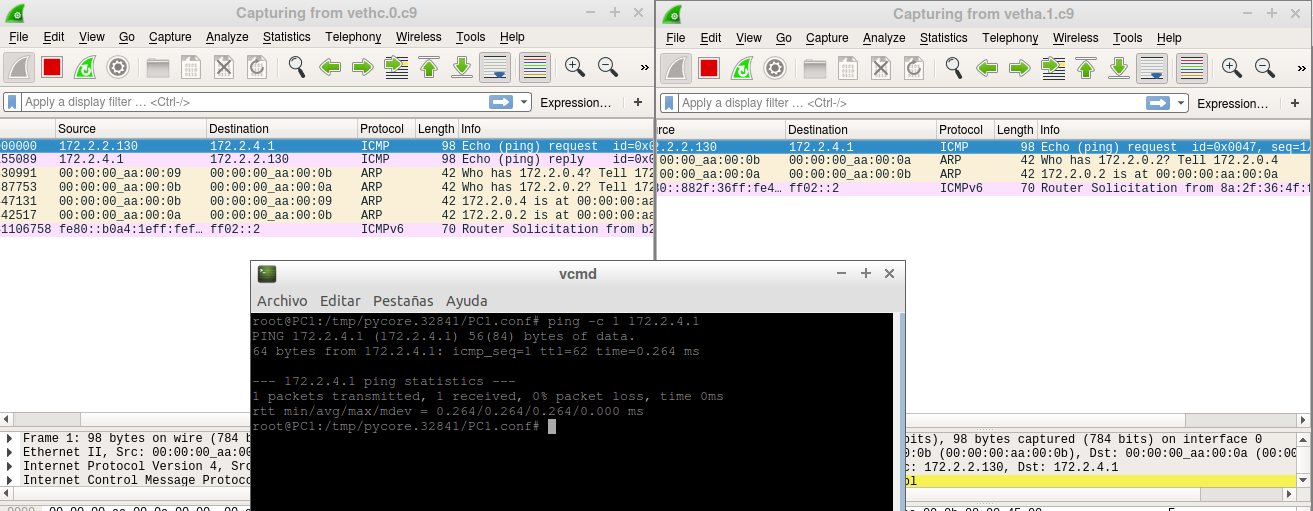
172.2.2.130, y el segundo respondiendo correctamente a dicha solicitud.

También se pueden observar los mensajes *ARP* solicitando las *MAC*, el Router 6 la de PC1 y viceversa.

*Wireshark* de Router 6, interfaz eth1, se observan mensajes similares al *Wireshark* de PC1



**Siguiendo el recorrido**



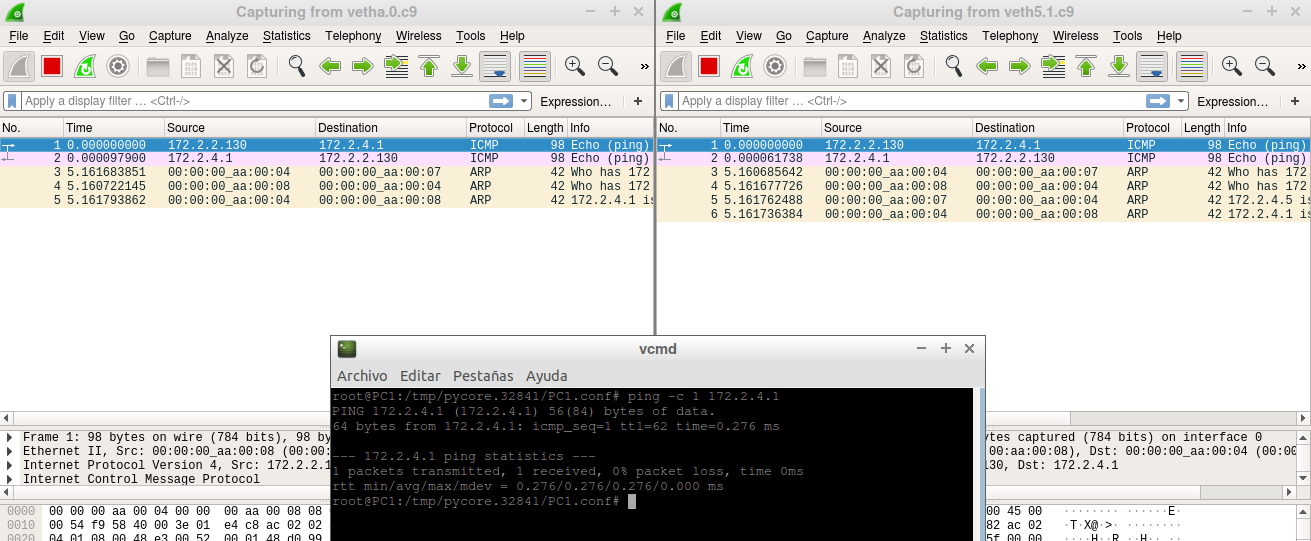
*Wireshark* de la Router 6, interfaz eth0, muestra mensajes *ICMP*, de solicitud y respuesta entre IP 172.2.2.130 y 172.2.4.1.

También se pueden observar los mensajes *ARP* solicitando las *MAC*, entre los *router*: del Router 6 al Router 4 (ida) y del Router 3 al Router 6 (vuelta).

*Wireshark* de la Router 4, interfaz eth1, muestra mensajes *ICMP*, de solicitud.

También se pueden observar los mensajes *ARP* solicitando las *MAC*, entre los *router*: Router 4 y Router 6.

**Viendo la interfaz de Router 4 eth0 y Router 2 eht1**



*Wireshark* de la Router 4, interfaz eth0, muestra mensajes *ICMP*, de solicitud y respuesta entre IP 172.2.2.130 y 172.2.4.1.

También se pueden observar los mensajes *ARP* solicitando las *MAC*, entre los *router*:

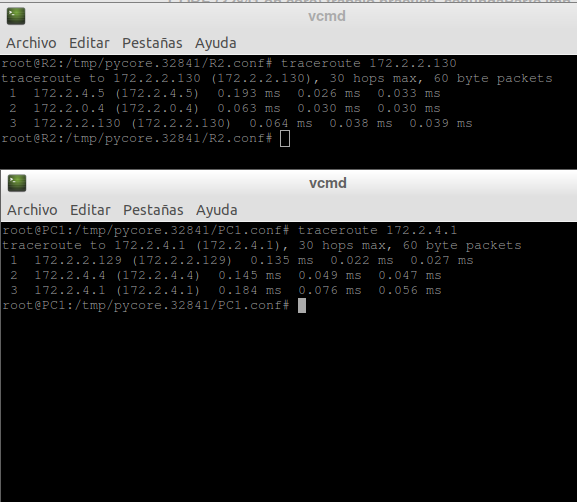
del Router 4 al Router 2 (ida) y del Router 2 al Router 3 (vuelta).

*Wireshark* de la Router 2, interfaz eth1, muestra mensajes *ICMP*, de solicitud y respuesta.

También se pueden observar los mensajes *ARP* solicitando las *MAC*, entre los *router*:

del Router 2 al Router 5 (vuelta) y Router 4 al Router 2 (ida).

También se puede visualizar el recorrido del paquete y ver los distintos routers que atraviesa.



En el caso el PC1: Router 6 - Router 4 - Router 2.

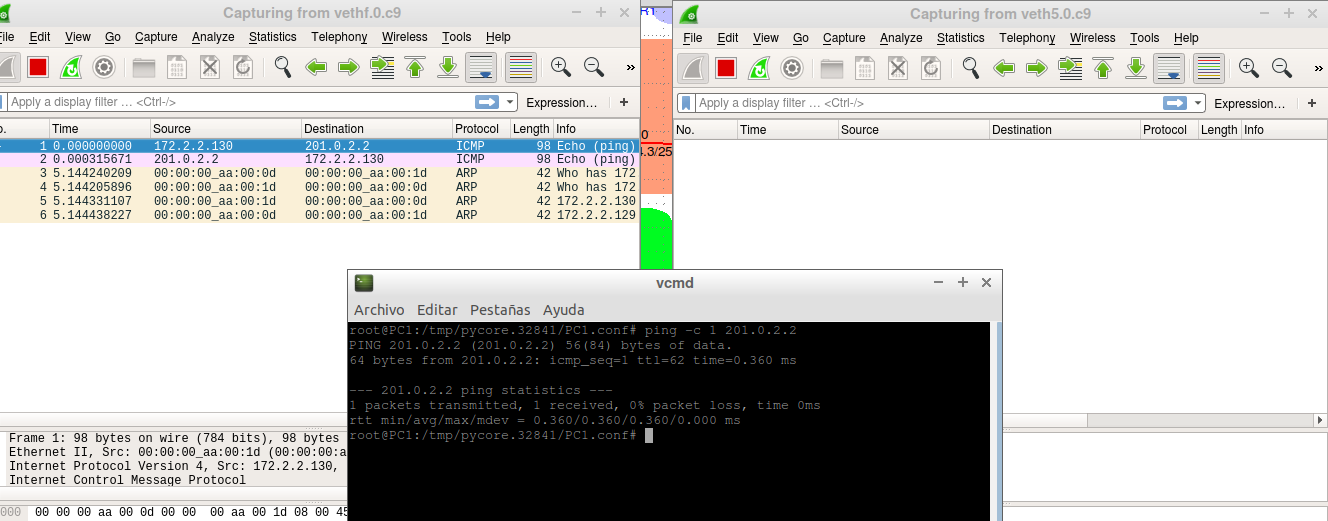
En el caso el Router 2: Router 3 - Router 6 - PC1.

**<ping -c 1 201.0.2.2> (de pc4 a IP publica R8)**

*Wireshark* de la PC1, muestra mensajes *ICMP*, de solicitud y respuesta proveniente del Router 2.

También se pueden observar los mensajes *ARP* solicitando las *MAC*, entre los *router*: del Router 2 al Router 3.

*Wireshark* del Router 2, donde al llegar el paquete a la IP privada, entiende que es el mismo router destino, por lo que no se observa actividad en la interfaz pública del Router 2.

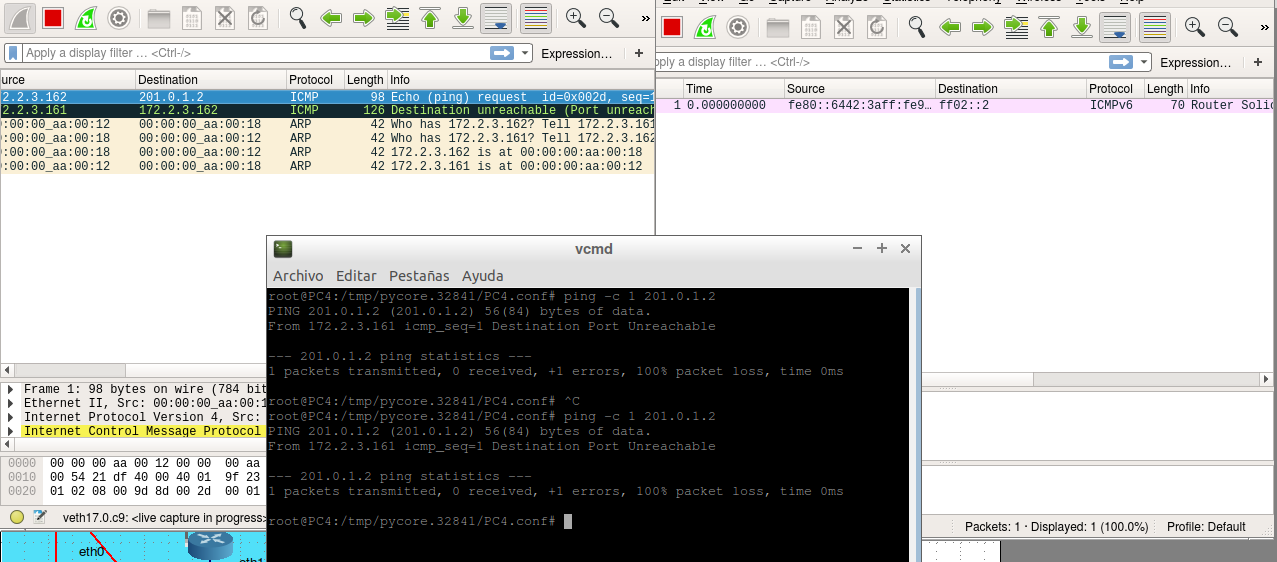


**<ping -c 1 201.0.1.2> (de pc4 a IP pública Router 8)**

*Ping* a la IP pública del Router 8, donde se observa en consola el mensaje paquete perdido, puerto de destino inalcanzable.

En *Wireshark* al enviar PC4 una solicitud a la interfaz pública del Router 8, se observa que no se encuentra respuesta, mensaje *ICMP*: Destino inalcanzable, esto se debe a que con anterioridad se configuró el Router 5, para que la subred del *Hub* (donde se encuentra PC4) tenga acceso solo a Fábrica B y *Datacenter*.

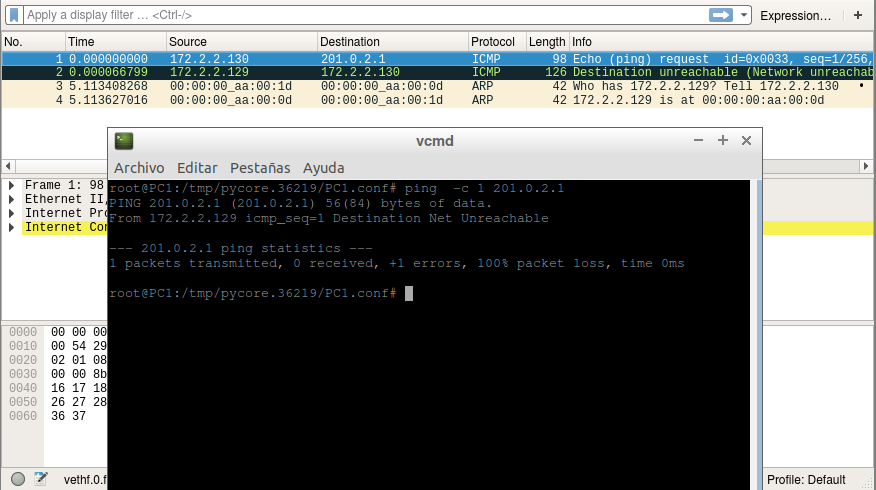
En *Wireshark* del Router 8, no se observa actividad alguna, ya que el paquete no atraviesa el Router 2 (frontera), debido a lo explicado en el punto anterior.



Izquierda representa pc4 y derecha pública de Router 8

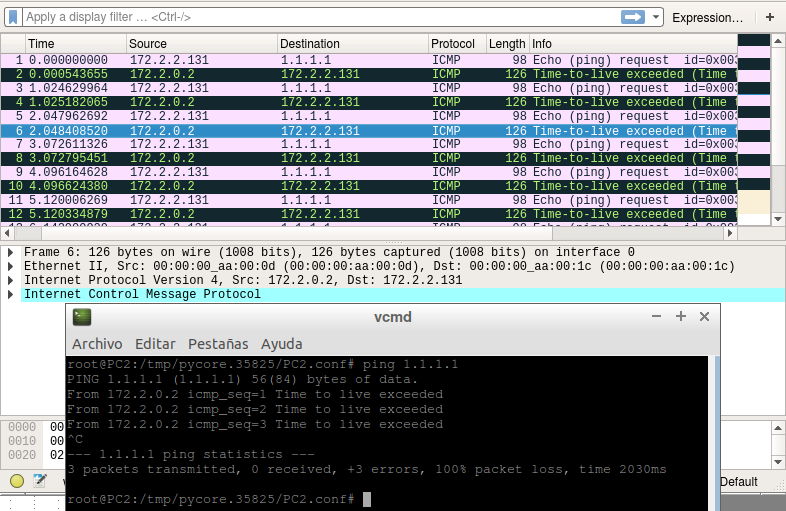
* **Llegando al final del trabajo practico especial se solicita obtener las siguientes respuestas:**

**a) *Destination Network Unreachable***: Para lograr dicho resultado se elimina el *default* de Router 6 y se realiza un *ping* de cualquiera de los *hosts* pertenecientes de la fábrica A hacia una IP pública.



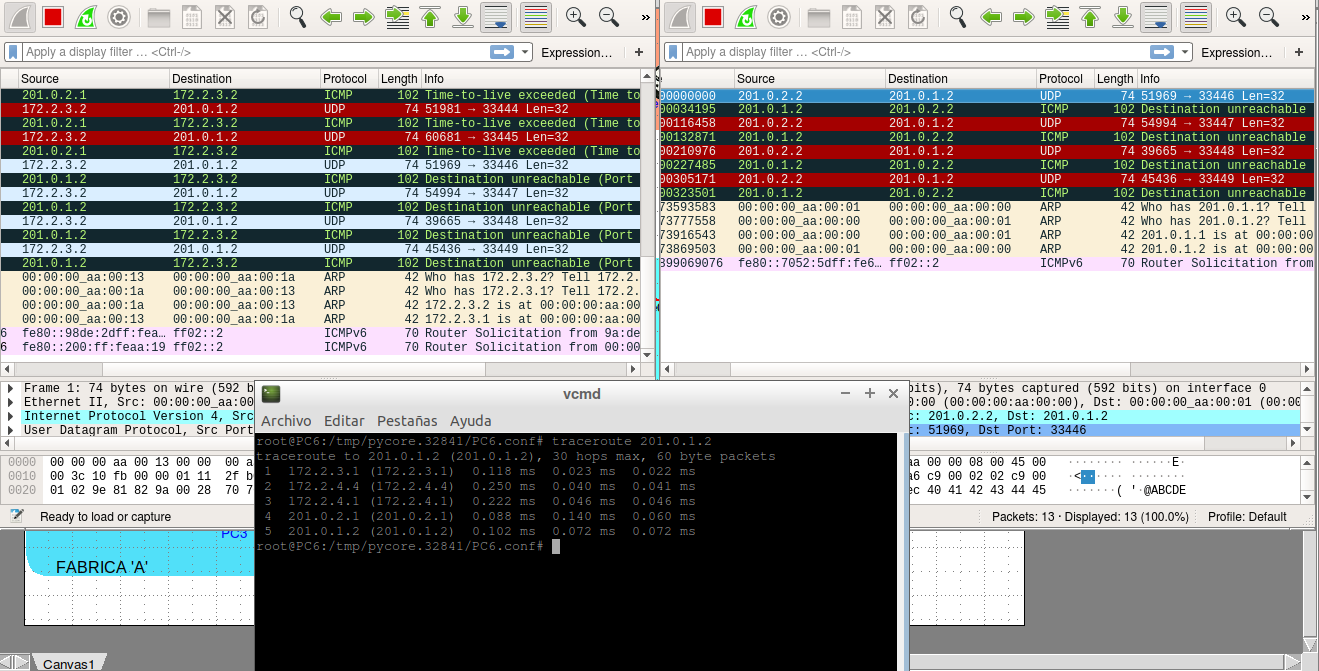
**b) *Time Exceeded***: Es logrado modificando el *default* (en este caso de Router 4 hacia Router 6 y viceversa). Luego de modificar el/os *defaults* se realiza un ping de algunos de los *hosts* pertenecientes a la fábrica A hacia internet (este caso la IP 1.1.1.1).

En este caso los *defaults* al estar apuntándose entre ellos se genera un bucle (Router 4 - Router 6) y se produce el error de “*Time-to-Live Exceeded*”.



* **El último ejercicio solicita usar la herramienta Tracerout y Wireshark para observar los resultados.**

Desde PC6 iniciamos, utilizando el comando *Traceroute* 201.0.1.2 (IP pública de Router 8).



PC6 izquierda - Router8 derecha.

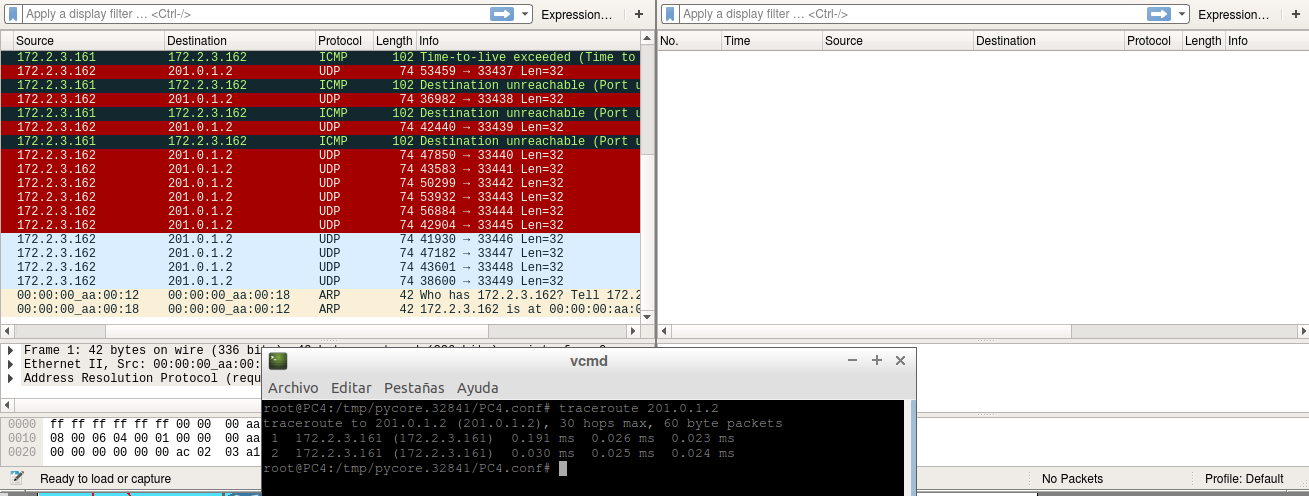
Al realizar un *Traceroute* desde PC6 hacia la IP pública del Router 8,

se observan los siguientes saltos de *router*: R5 -> R4 -> R2 -> R1 -> R8

Con *Wireshark* en PC6, se observa mensaje *ARP* desde PC6 solicitando quién tiene la *MAC* de IP 172.2.3.1 (Router 5) y su respectiva respuesta, a partir de allí se generan mensajes *UDP* por cada salto de *router*, el “*time-to-live- exceeded*" de cada paquete se va incrementando en uno, por cada *router* hasta llegar a destino, luego los mensajes *ARP* desde Router 5 solicitando la *MAC* de IP 172.2.3.2 (PC6), y su respectiva respuesta.

En *Wireshark* de Router 8, se observa mensaje *ARP* desde Router 1 solicitando quien tiene la *MAC* de IP 201.0.1.2 (R8) y su respectiva respuesta, a partir de allí se generan mensajes *UDP* entre el Router 1 y Router 8, luego los mensajes *ARP* desde Router 8 solicitando la *MAC* de IP 201.0.1.1 (Router 1), y su respectiva respuesta.

**Desde la PC4 a la IP pública de R8.**



PC4 lado izquierdo - Router 8 lado derecho.

Se observa mediante *Traceroute*, que el paquete no atraviesa el Router 5,

debido a la regla *iptables* en la tabla *filter* configurada con anterioridad.

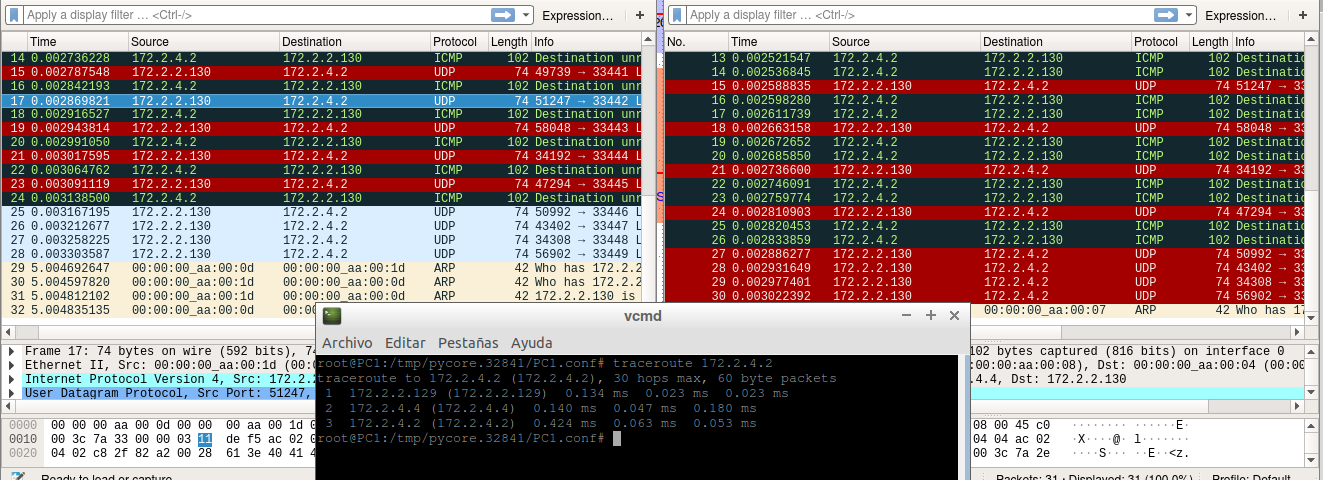
En *Wireshark* de PC4, se observa mensaje *ARP* donde PC4 solicita quién tiene IP 172.2.3.162 (en este caso Router 5) y su respuesta, luego varios mensajes *UDP* hacia Router 5, pero no puede atravesarlo (por la regla *iptables* antes dicha).

En *Wireshark* de Router 8, no se observa ninguna actividad, ya que el paquete nunca atraviesa el Router 5, por lo que nunca llega a su destino.

**Desde la PC1 a la IP del Server 1.**

En este caso el paquete viaja de forma satisfactoria y sin grandes modificaciones.

Viaja directo al *Datacenter* por Router 4 como se indicó en el ejercicio 8.



PC1 lado izquierdo - Server1 lado derecho.

Conclusión Segunda Parte

Se puede concluir en una devolución positiva respecto a los ítems solicitados por la catedra, logrando un resumen preciso de los detalles al momento de diagramar un red, de esta manera se logró un equilibrio entre las tareas y el aprendizaje logrado.