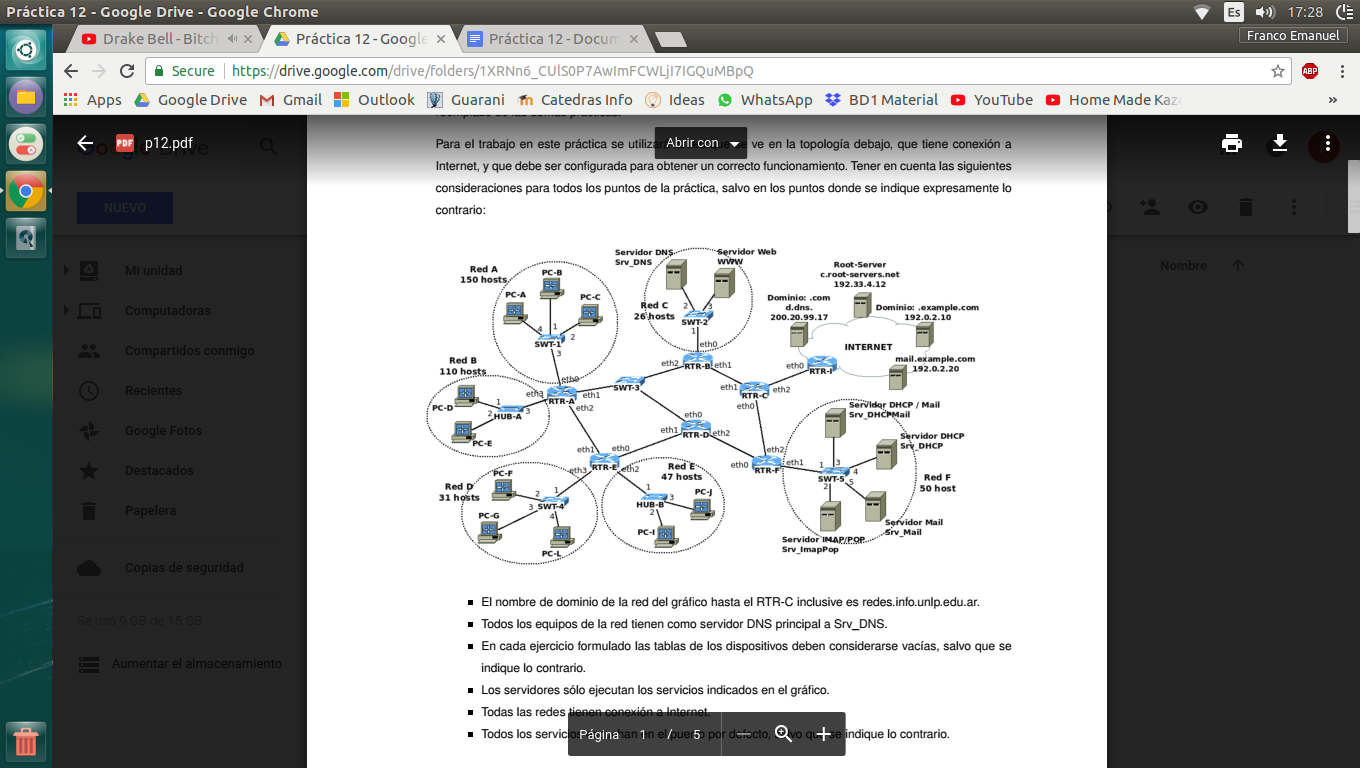
**Práctica 12**

**Práctica Integradora**

**Esta práctica tiene como objetivo integrar todos los conceptos aprendidos durante la cursada. Se recomienda su resolución para llegar con una mejor preparación a rendir los parciales pero no debe ser tomada como un reemplazo de las demás prácticas.**

**Para el trabajo en este práctica se utilizará la red que se ve en la topología debajo, que tiene conexión a Internet, y que debe ser configurada para obtener un correcto funcionamiento. Tener en cuenta las siguientes consideraciones para todos los puntos de la práctica, salvo en los puntos donde se indique expresamente lo contrario:**

****

* **El nombre de dominio de la red del gráfico hasta el RTR-C inclusive es redes.info.unlp.edu.ar.**
* **Todos los equipos de la red tienen como servidor DNS principal a Srv\_DNS.**
* **En cada ejercicio formulado las tablas de los dispositivos deben considerarse vacías, salvo que se indique lo contrario.**
* **Los servidores sólo ejecutan los servicios indicados en el gráfico.**
* **Todas las redes tienen conexión a Internet.**
* **Todos los servicios escuchan en el puerto por defecto, salvo que se indique lo contrario.**

**Capa de aplicación**

**1. Si PC-A desea acceder a www.redes.info.edu.ar, responder:**

**a. ¿Cuál es el primer paso que debe realizar PC-A? ¿A quién le hace la consulta?**

PC-A tiene un nombre de dominio ‘[www.redes.unlp.edu.ar](http://www.redes.unlp.edu.ar)’, que pudo haber sido solicitado desde un navegador (usando el protocolo HTTP, o HTTPS), por ejemplo (aplicación que ejecuta en el cliente). Sin embargo, al tener solo el nombre de dominio, PC-A no sabe a donde tiene que ir a buscar el recurso. Para esto, necesita ponerse en contacto con su servidor DNS local, en este caso Srv\_DNS.

Del lado de capa de aplicación, se usa el protocolo DNS. Las consultas DNS normalmente viajan sobre UDP (pueden hacerlo sobre TCP).

La función del protocolo DNS (protocolo de capa de aplicación) es obtener la IP correspondiente a un nombre de dominio dado (justo lo que necesitamos en este caso). No es la única función disponible de DNS; también podría hacer la operación inversa, o resolver nombres de dominio para servidores de mail.

Las consultas al servidor DNS local normalmente son recursivas, lo cual obliga a Srv\_DNS a responder al cliente con una respuesta satisfactoria o con un error. (Esto no es obligatorio, ya que desde el cliente se podría hacer una consulta iterativa).

Srv\_DNS podría tener la información guardada en caché, lo cual ahorraría tiempo de búsqueda, o podría no tenerla y tener que salir a la red a buscarla. En caso de salir a la red, el servidor DNS local debe conocer a un servidor DNS raíz, a quien debe preguntarle por ‘[www.redes.info.edu.ar](http://www.redes.info.edu.ar)’, y el servidor va a responderle como mejor sabe. El servidor DNS raíz también podría darse cuenta que la dirección no existe, e informar.

A priori, no es la única consulta que el servidor Srv\_DNS tiene que hacer. Lo mas normal es que el servidor haga consultas iterativas, consultando a los distintos servidores por el nombre de dominio todo entero (‘www.redes.info.edu.ar’), los cuales responden como mejor saben (con el resultado esperado, sin resultados útiles, o con una lista de resultados). En caso de recibir una lista de resultados, Srv\_DNS va a elegir una opción de dicha lista y seguir consultando.

Lo esperable es que en algún momento el servidor DNS dé con la información pedida, y responda al cliente la dirección IP del servidor de ‘[www.redes.info.edu.ar](http://www.redes.info.edu.ar)’. Una vez hecho esto, el cliente puede comunicarse con el servidor de ‘www.redes.info.edu.ar‘ y pedirle el recurso.

**b. Indique los mensajes que enviará y recibirá en la consulta anterior.**

PC-A va a hacer una consulta DNS (a Srv\_DNS) consultando el registro A para ‘[www.redes.info.edu.ar](http://www.redes.info.edu.ar)’.

Es esperable que se responda a la solicitud de PC-A con el proceso descrito anteriormente.

Después PC-A va a conectarse al servidor correspondiente a ‘[www.redes.info.edu.ar](http://www.redes.info.edu.ar)’, y pedirle los recursos que necesite.

**2. La respuesta del paso anterior, ¿es autoritativa? Justifique.**

Depende. En el enunciado no está especificado que el servidor web sea el autoritativo (autorizado a responder para ese nombre de dominio) para la zona, ni tampoco el servidor Srv\_DNS. Si alguno de éstos es el autoritativo, entonces la respuesta sí será autoritativa. Por el contrario, si no lo son, la respuesta puede ser autoritativa, siempre y cuando durante las consultas iterativas de Srv\_DNS no responda la IP correcta un servidor con la información guardada en su caché.

Los servidores no autoritativos pueden guardar información de consultas que pasan por él en su caché. Es una forma de ahorrar tráfico de red.

La única forma de que la respuesta sea autoritativa, sería hablar directamente con el servidor autoritativo.

**3. Si desde PC-G el usuario roger@redes.info.unlp.edu.ar le envía un mail al usuario** [**rafael@example.com**](mailto:rafael@example.com)**, contestar:**

[**http://wiki.inf.utfsm.cl/index.php?title=Electronic\_Mail:\_SMTP,\_POP3,\_IMAP**](http://wiki.inf.utfsm.cl/index.php?title=Electronic_Mail:_SMTP,_POP3,_IMAP) **→ explicación.**

**a. Detalle todo el proceso necesario para que el mail llegue a su servidor de correo. Por cada mensaje enviado indicar quién lo envía, quién lo recibe, protocolo utilizado, si es TCP o UDP y puertos origen y destino.**

Podemos asumir que en PC-G se está corriendo una aplicación de mail. Dicha aplicación asumimos que conoce a los servidores de mail para el dominio ‘redes.info.unlp.edu.ar’.

Por ende, PC-G va a tener que hacer una consulta DNS consultando el registro A del servidor de correo de mayor prioridad. Esta consulta va a tener como puerto origen uno mayor a 1023, y puerto destino 53 (puerto por defecto de DNS). DNS generalmente viaja sobre UDP.

La consulta es al servidor DNS local, quien puede responder por información que tenga guardada en su caché, o preguntar a distintos servidores de manera iterativa hasta dar con la IP del servidor de correo de ‘redes.info.unlp.edu.ar’.

En algún momento el servidor DNS local va a dar con la respuesta, y va a responderle al cliente. En este caso, el puerto origen será el 53, y el destino uno mayor a 1023 (el puerto origen en la solicitud del cliente). La respuesta viajará por UDP.

Una vez que PC-G tiene la IP de su servidor de correo, enviará el mail a dicho servidor usando SMTP (puerto origen > 1023, puerto destino 25). SMTP requiere una fase de autenticación, y viaja sobre TCP. Cada vez que se usa TCP hay que establecer la conexión, usando el saludo de tres vías (para esto se usan los flags SYN y ACK). Una vez terminado el flujo de datos útiles, hay que cerrar la conexión (participan los flags FIN y ACK).

**b. Detalle todo el proceso necesario para que el mail llegue al servidor de correo del dominio destino. Por cada mensaje enviado indicar quién lo envía, quién lo recibe, protocolo utilizado, si es TCP o UDP y puertos origen y destino.**

Asumiendo que el mail enviado por PC-G llegó correctamente a su servidor de mail, ahora su servidor de mail va a tener que hacer dos consultas DNS. La primera va a ser por los registros MX para ‘example.com’. Nuevamente la consulta DNS tendrá como puerto origen uno mayor a 1023, y puerto destino 53, y viajará sobre UDP.

La respuesta puede ser tanto un solo resultado, como una lista de resultados. El resultado va a tener asociado un valor, el cual es el valor de preferencia / prioridad (a menor número, mayor prioridad). El servidor ‘redes.info.unlp.edu.ar’ va a seleccionar el de mayor prioridad y va a hacer una nueva consulta DNS, esta vez por el registro A. La información de puertos y protocolos es la misma que en el paso anterior.

Finalmente, el servidor ‘redes.unlp.edu.ar’ se conectará usando el protocolo SMTP (puerto origen uno mayor a 1023, puerto destino 25) al servidor de mail de mayor prioridad de ‘example.com’, y le enviará el mail. Este mensaje viaja sobre TCP, por ende es necesario establecer la conexión y cerrarla posteriormente.

El servidor de correo ‘example.com’ procesará el mail, y lo pondrá en el buzón de correo de Rafael.

**4. El usuario rafael, ¿cómo recuperará sus mails desde PC-J? Indique los pasos necesarios para hacerlo. ¿Puede hacerlo desde otra PC? Justifique.**

El usuario Rafael seguramente va a interactuar directamente con una aplicación que a su vez va a actuar con el nivel de capa de aplicación de la pila de protocolos. Es posible que dicha aplicación usada por Rafael use el protocolo IMAP o POP3 para comunicarse con su servidor de correo y obtener sus correos.

Teniendo en cuenta que Rafael está usando un cliente de correo, es esperable (al igual que fué en el envío), que dicha aplicación conozca los servidores de correo para ‘example.com’. Por ende, PC-J va a tener que hacer una consulta DNS por el registro A del servidor de correo de mayor prioridad (puerto origen uno mayor a 1023, puerto destino 53; UDP como protocolo de capa de transporte). Una vez obtenida la IP de su servidor de correo, se tiene que conectar con POP3 o IMAP (recordar que ambos requieren autenticación) para leer sus correos. Los casos son bien diferentes:

* IMAP: El puerto origen sería uno mayor a 1023, y el puerto destino sería 143. Viaja sobre TCP (es necesario establecer la conexión y posteriormente cerrarla). PC-J se conecta al servidor de correos de ‘example.com’ y obtiene sus mails. IMAP es un protocolo que utiliza muchos recursos, y requiere siempre tener internet para poder leer los mails. Lo bueno de IMAP es que podemos conectarnos desde cualquier dispositivo y leer nuestros correos, ya que no los elimina del servidor a menos que el cliente lo pida explícitamente.
* POP3: normalmente trabaja con la configuración descarga-borrado, lo cual implica que nos conectemos al servidor (puerto origen mayor a 1023, puerto destino 110; viaja sobre TCP (necesario saludo de tres vías y cierre de conexión)), y nuestra PC-J va a descargar los mails localmente (luego de la fase de autenticación de POP3 (tiene 3 fases: autenticación, transacción y actualización)). Eso elimina los correos del servidor (fase de actualización), y por ende, no es posible acceder desde otro dispositivo y leerlo. Es un protocolo que consume menos recursos y además no es necesario tener siempre acceso a internet, solo la primera vez para descargar dicho correo localmente. Lo malo es que no puedo acceder desde otros dispositivos a leer los correos. Otra configuración, menos usada, es descarga-almacenamiento. En este caso los correos no son eliminados del servidor.

**Capa de transporte**

**5. Si desde PC-B se establece una sesión TCP al servidor www con los siguiente datos, ¿cómo sería el saludos de 3 vías?**

**PC-B: ISN: 150100 WWW: ISN: 204500**

El saludo de tres vías se hace en 3 pasos:

Paso 1

PC-B envía a WWW un mensaje sin carga útil, con el flag SYN (sincronizar números de secuencia) setteado, y número de secuencia 150100. PC-B queda en estado SYN-SENT, lo cual indica que solicitó iniciar la conexión.

Paso 2

WWW responde a PC-B, con el flag SYN setteado, y también el flag ACK. El número de secuencia será 204500, y el ACK será 150101. El estado de WWW será SYN-RCV.

Paso 3

PC-B envía a WWW un tercer mensaje, que puede o no tener carga útil. Tendrá setteado el flag ACK. El número de secuencia será 150101, y el ACK 204501. El estado de PC-B pasará a ser ESTABLISHED. Al llegar esta información al servidor, también pasará a estado ESTABLISHED.

**6. ¿Que responderá cada nodo ante cada una de los siguientes mensajes? Justifique, indicando flags, puertos, etc, siempre que sea necesario.**

* **PC-B envía un mensaje con el flag SYN seteado al servidor Srv\_DHCP al puerto 69.**

Recibirá RA (RST, ACK). El puerto 69 es el por defecto de TFTP (trabaja con UDP). El origen será 69 y el destino uno mayor a 1023.

* **PC-B envía un mensaje UDP al servidor Srv\_DHCP al puerto 69.**

UDP no envía nada en caso de éxito en la conexión.

* **PC-B envía un mensaje UDP al servidor WWW al puerto 80.**

El puerto 80 es normalmente usado por HTTP. Los mensajes HTTP normalmente viajan por TCP, por ende podemos asumir que dicho puerto no trabaja con UDP y va a responder un mensaje ICMP “puerto inalcanzable”. El puerto origen será el 80, y el puerto destino será mayor a 1023 (con el que haya establecido la conexión PC-B hacia el servidor)

* **PC-A envía un mensaje con el flag SYN seteado al servidor WWW al puerto 80.**

Recibirá SYN-ACK. HTTP trabaja normalmente con TCP, por ende estará escuchando en dicho puerto. El puerto origen será 80 y el destino será uno mayor a 1023.

* **PC-A envía un mensaje con el flag SYN seteado al servidor WWW al puerto 8080.**

Recibirá un mensaje RA. Los protocolos bien conocidos vistos en la materia usan puertos bien conocidos, menores a 1023. Por ende podemos asumir que no va a haber ningún protocolo conocido corriendo en dicho puerto. El origen será 8080 y el destino será uno mayor a 1023.

**7. Si entre PC-E y el Srv\_ImapPop se realiza el intercambio de datos que se muestra a continuación una vez finalizado el saludo de 3 vías, contestar:**

**a. Indicar cómo sería el saludo de 3 vías realizado para establecer esa sesión, suponiendo que el intercambio que se ve en el diagrama se realiza ni bien se terminó de establecer la misma.**

Paso 1

Tendría setteado el flag SYN. Número de secuencia = 998.

Paso 2

Tendría setteado el flag SYN ACK. Número de secuencia 1999, ACK 1999.

Paso 3

Tendría setteado el flag ACK. Número de secuencia 199, ACK 200.

Vale aclarar que el paso 3 podría en realidad ser el primero que se ve en el diagrama (el tercer paso del saludo de tres vías puede llevar carga útil). En este caso, los primeros dos pasos serian:

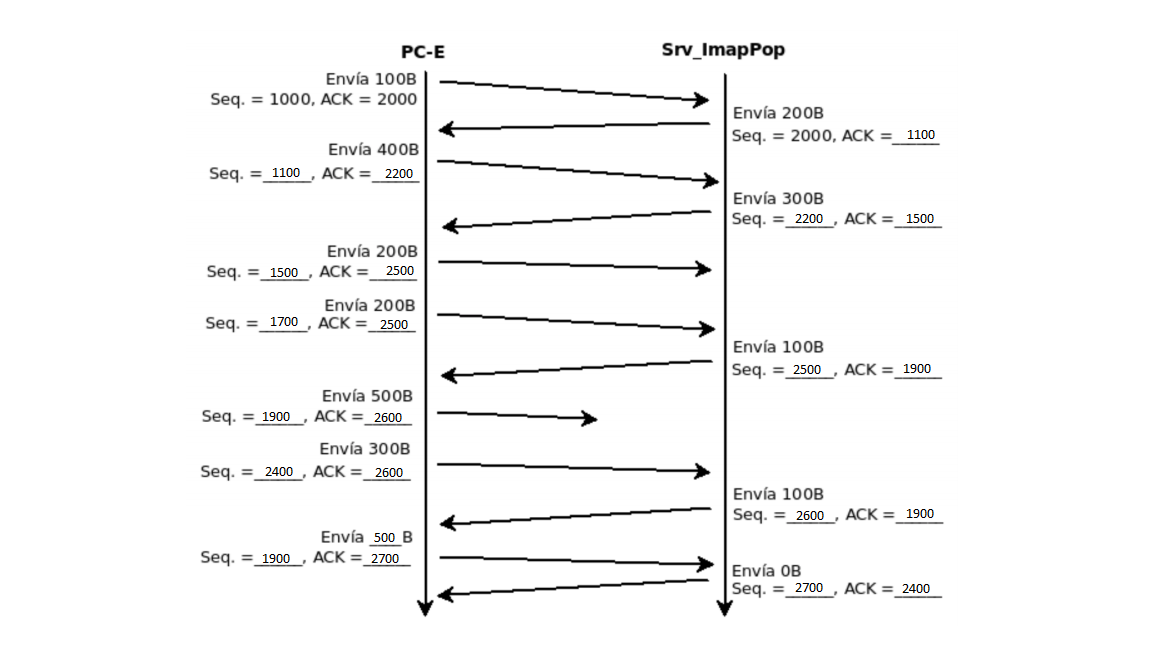
Paso 1

Tendría setteado el flag SYN. Número de secuencia = 999.

Paso 2

Tendría setteado el flag ACK. Número de secuencia = 1999, ACK = 1000.

**b. Completar los datos faltantes en el intercambio de datos.**

****

Nota: Hay una secuencia que se pierde (SEQ 1900, ACK 2600, LENGTH 200). Luego de enviar esa, PC-E envía otra secuencia. Cuando le llega al servidor, el servidor responde que todavía está esperando la secuencia que se perdió. Por eso en el último paso (del lado del cliente) debe reenviarla. En este reenvío, podría enviar tanto 500 bytes (los que se perdieron) como también 800. Si mandara 800, sería porque estamos asumiendo que todo paquete que sale del cliente, tiene que tener un correspondiente ACK (todo tiene que ser enviado en orden). Como está hecho acá, asumimos que Srv\_ImaPop trabaja con un buffer, y no es necesario que los mensajes lleguen en orden.

**c. Indicar cómo sería el cierre de la conexión si el cierre lo inicia el cliente y sólo consume el intercambio de 3 mensajes.**

Primero que nada, es bueno aclarar en qué casos el cierre de conexión es en 3 pasos, y en qué casos es en 4 pasos.

Cuando se hace en 3 pasos, es porque en el receptor de la solicitud de fin de conexión se procesó rápido el mensaje, se lo informó a capa de aplicación rápido, y se decidió también cerrar la conexión. Si esto es rápido, entonces en la respuesta al pedido de cierre, puede ir el flag FIN del otro lado también. Por el contrario, el flag FIN del otro lado (servidor) tiene que venir en un paquete aparte. Recordar que ambos lados tienen que cerrar la conexión.

Inicialmente ambos lados van a tener estado ESTABLISHED.

Paso 1

El cliente manda FIN, y queda en estado Fin-Wait-1.

Paso 2

El servidor manda FIN ACK, y queda en estado LAST-ACK. Notar que si los pasos fueran 4, este sería el paso 2, y el estado que tomaría el servidor sería Close-Wait. Luego, en el hipotético tercer paso, el servidor quedaría en estado LAST-ACK.

Paso 3

El cliente responde ACK, y queda en estado Time-Wait, esperando que los paquetes mueran para terminar la conexión (pasa a estado CLOSED). El servidor quedará en LISTEN. Notar que si hubiera sido en 4 pasos, el cliente hubiera quedado en FIN-WAIT-2 en caso de recibir solo un ACK que confirme el cierre de sesión por parte del servidor. Luego, el cliente hubiera quedado en Time-Wait al recibir el FIN del servidor.

**Capa de red**

**8. Asignar el direccionamiento IP a las redes del gráfico teniendo en cuenta lo siguiente:**

**a. Las redes que asigne deben ser válidas. Por ello, identifique primero las redes inválidas y explique por qué no pueden ser utilizadas.**

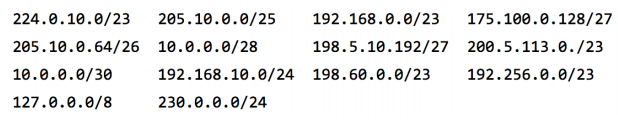
**b. Red C y F tiene direccionamiento público de Clase B y C respectivamente.**

**c. Redes A, B, D y E y las redes que interconectan las routers direccionamiento privado.**

**d. Red punto a punto contra Internet, red pública clase C.**

**e. Asignar las redes de manera que se desperdicie la menor cantidad de direcciones y que sea posible sumarizar las tablas de ruteo.**

**f. Todas las redes deben poder navegar por Internet menos Red D.**

****

Primero voy a identificar aquellas redes que no son válidas:

La red 230.0.0.0 es de clase D, está reservada para multicast. Lo mismo para la red 224.0.10.0/23.

La red 127.0.0.0/8 está reservada para localhost.

La red 200.5.113.0 es una dirección de host.

La red 192.256.0.0 no es válida. Los octetos oscilan entre 0 y 255.

La red 200.5.113.0./23 es inválida por tener un punto más.

Para las clases A, B y C, los rangos de redes privadas son:

Clase A: 10.0.0.0 - 10.255.255.255

Clase B: 172.16.0.0 - 172.31.255.255

Clase C: 192.168.0.0 - 192.168.255.255

Ahora si, empiezo el proceso de asignación de redes:

La red A es la que más host necesita (150). Tiene que tener direccionamiento privado.

Como necesito 150 host, necesito 8 bits. 2^8 = 256 - 2 = 254.

**192.168.10.0/24 → Asigno a red A.**

La siguiente red que mas host necesita es la B (110). Necesito 7 bits. 2^7 = 128 - 2 = 126. Tiene que ser privada.

Elijo 192.168.0.0/23. Me sobran 2 bits para subredes.

Voy a dividir sumando siempre un bit de máscara, para que a su vez me queden resumidas las libres.

Me quedaría:

* 192.168.0.0/24
* 192.168.1.0/24 → Libre → **Usada en un paso siguiente (enlaces entre routers).**

Ahora sumo un bit de máscara a 192.168.0.0/24. Me quedaria:

* **192.168.0.0/25 → Asigno a red B.**
* 192.168.0.128/25 → Libre → **Usada en siguiente paso.**

La siguiente red que más necesita es la E (47 host). Necesito 6 bits. 2^6 = 64 - 2 = 62. Tiene que ser privada.

Elijo la red que me quedó libre del paso anterior: 192.168.0.128/25. Me sobra un bit para subredes. Me quedaría:

* **192.168.0.128/26 → Asigno a red E.**
* **192.168.0.192/26 → Asigno a red D.**

La siguiente red que mas host necesita es la F (50 host). Necesito 6 bits. 2^6 = 64 - 2 = 62.

Tiene que ser pública de clase C.

* **205.10.0.64/26 → Asigno a red F.**

La siguiente red que mas host necesita es la C (26 host). Necesito 5 bits. 2^5 = 32 - 2 = 30.

Tiene que ser pública de clase B.

* **175.100.0.128/27 → Asigno a red C.**

Ahora tengo que asignar redes a las interfaces entre routers. Tienen que ser privadas. Uso una dirección que anteriormente me quedó libre: 192.168.1.0/24.

Tenemos una red entre RTR-A ETH1, RTR-F ETH0 y RTR-B ETH2. Necesita direccionar 3 host, por ende necesita 3 bits. 2^3 = 8 - 2 = 6. De la red elegida, me van a quedar 4 bits para subredes:

* 192.168.1.0/25
* 192.168.1.128/25 → Libre.

Divido denuevo 192.168.1.0/25

* 192.168.1.0/26
* 192.168.1.64/26 → Libre.

Divido denuevo: 192.168.1.0/26

* 192.168.1.0/27
* 192.168.1.32/27 → Libre.

Divido denuevo 192.168.1.0/27

* 192.168.1.0/28
* **192.168.1.16/28 → Libre → Usada en un siguiente paso.**

Divido denuevo 192.168.1.0/28

* **192.168.1.0/29 → Asigno a red entre entre RTR-A ETH1, RTR-F ETH0 y RTR-B ETH2**
* 192.168.1.8/29 → Libre → **Usada en siguiente paso.**

Ahora solamente me queda asignar 6 redes, las cuales todas tienen 2 host. Necesito 2 bits. 2^2 = 4 - 2 = 2.

Elijo la que me quedó libre del paso anterior: 192.168.1.8/29. La divido, me quedaria:

* **192.168.1.8/30 → Asigno a red entre RTR-A y RTR-E.**
* **192.168.1.12/30 → Asigno a red entre RTR-E y RTR-D.**

Para seguir, elijo una red que me quedó libre de un paso anterior: 192.168.1.16/28. La divido:

* 192.168.1.16/29
* 192.168.1.24/29 → Libre → **Usada en un siguiente paso.**

Vuelvo a dividir 192.168.1.16/29

* **192.168.1.16/30 → Asigno a red entre RTR-D y RTR-F.**
* **192.168.1.20/30 → Asigno a red entre RTR-B y RTR-C.**

Me falta asignar a una subred con las mismas condiciones que los anteriores enlaces. Para esto, vuelvo a dividir una libre de un paso anterior: 192.168.1.27/29. Me quedaría:

* **192.168.1.24/30 → Asigno a red entre RTR-F y RTR-C.**
* 192.168.1.28/30 → Libre

Solo me queda asignar una red, entre RTR-C y RTR-I. Tiene que ser pública, de clase C. Para esto, elijo 205.10.0.0/25. Me sobran 5 bits, divido:

* 205.10.0.0/26
* 205.10.0.64/26 → Libre.

Sigo dividiendo 205.10.0.0/26:

* 205.10.0.0/27
* 205.10.0.32/27 → Libre.

Sigo dividiendo 205.10.0.0/27:

* 205.10.0.0/28
* 205.10.0.16/28 → Libre.

Sigo dividiendo 205.10.0.0/28:

* 205.10.0.0/29
* 205.10.0.8/29 → Libre.

Sigo diviendo 205.10.0.0/29

* **205.10.0.0/30 → Asigno a red entre RTR-C y RTR-I**
* **205.10.0.4/30 → Libre.**

**9. Asignar IP a cada uno de los dispositivos.**

Red A: **192.168.10.0/24**

* PC-A: 192.168.10.1/24
* PC-B: 192.168.10.2/24
* PC-C: 192.168.10.3/24
* RTR-A ETH0: 192.168.10.4/24

Red B: **192.168.0.0/25**

* PC-D: 192.168.0.1/25
* PC-E: 192.168.0.2/25
* RTR-A ETH3: 192.168.0.3/25

Red C: **175.100.0.128/27**

* Srv\_DNS: 175.100.0.129/27
* WWW: 175.100.0.130/27
* RTR-B ETH0: 175.100.0.131

Red D: **192.168.0.192/26**

* PC-F: 192.168.0.193/26
* PC-G: 192.168.0.194/26
* PC-L: 192.168.0.195/26
* RTR-E ETH3: 192.168.0.196/26

Red E: **192.168.0.128/26**

* PC-I: 192.168.0.129/26
* PC-J: 192.168.0.130/26
* RTR-E ETH2: 192.168.0.131/26

Red F: **205.10.0.64/26**

* Srv\_DHCPMail: 205.10.0.65/26
* Srv\_DHCP: 205.10.0.66/26
* Srv\_Mail: 205.10.0.67/26
* Srv\_ImapPop: 205.10.0.68/26
* RTR-F ETH1: 205.10.0.69/26

Red entre RTR-A ETH1, RTR-F ETH0 y RTR-B ETH2: 192.168.1.0/29

* RTR-A ETH1: 192.168.1.1/29
* RTR-F ETH0: 192.168.1.2/29
* RTR-B ETH2: 192.168.1.3/29

Red entre RTR-A ETH2 y RTR-E ETH1: 192.168.1.8/30

* RTR-A ETH2: 192.168.1.9/30
* RTR-E ETH1: 192.168.1.10/30

Red entre RTR-E ETH0 y RTR-D ETH1: 192.168.1.12/30

* RTR-E ETH0: 192.168.1.13/30
* RTR-D ETH1: 192.168.1.14/30

Red entre RTR-D ETH2 y RTR-F ETH0: 192.168.1.16/30

* RTR-D ETH2: 192.168.1.17/30
* RTR-F ETH0: 192.168.1.18/30

Red entre RTR-B ETH1 y RTR-C ETH1: 192.168.1.20/30

* RTR-B ETH1: 192.168.1.21/30
* RTR-C ETH1: 192.168.1.22/30

Red entre RTR-F ETH2 y RTR-C ETH0: 192.168.1.24/30

* RTR-F ETH2: 192.168.1.25/30
* RTR-C ETH0: 192.168.1.26/30

Red entre RTR-C ETH2 y RTR-I ETH0: 205.10.0.0/30

* RTR-C ETH2: 205.10.0.1/30
* RTR-I ETH0: 205.10.0.2/30

**10. Determinar las tablas de ruteo de cada uno de los routers (excepto RTR-I) teniendo en cuenta que:**

**a. Red F llega a las redes A, B, D y E a través de router RTR-D y a la red C a través de RTR-C.**

**b. RTR-C llega a Red A, B y C a través del RTR-B. A las demás redes RTR-F.**

**c. RTR-D llega a Red A y B a través de red a través de RTR-A.**

**d. En todos los casos donde no esté especificado, el ruteo debe ser óptimo (entendiendo por óptimo que tenga la menor cantidad de saltos posibles).**

RTR-A

En este router se pueden sumarizar la red E y la red D. Son consecutivas y pares.

|  |  |  |  |
| --- | --- | --- | --- |
| **Direccion** | **Gateway** | **Máscara** | **Interfaz** |
| 192.168.10.0 (Red A) | 0.0.0.0 | /24 | ETH0 |
| 192.168.0.0 (Red B) | 0.0.0.0 | /25 | ETH3 |
| 175.100.0.128 (Red C) | 192.168.1.3 (RTR-B ETH2) | /27 | ETH1 |
| 192.168.0.128 (Red E y Red D) | 192.168.1.10 (RTR-E ETH1) | /25 | ETH2 |
| 205.10.0.64 (Red F) | 192.168.1.10 (RTR-E ETH1) | /26 | ETH2 |
| 192.168.1.8 | 0.0.0.0 | /30 | ETH2 |
| 192.168.1.0 | 0.0.0.0 | /29 | ETH1 |
| 0.0.0.0 | 192.168.1.3 (RTR-B ETH2) | /0 | ETH1 |

RTR-B

En este router se pueden sumarizar la red E y la red D. Son consecutivas y pares.

|  |  |  |  |
| --- | --- | --- | --- |
| **Direccion** | **Gateway** | **Máscara** | **Interfaz** |
| 192.168.10.0 (Red A) | 192.168.1.1 (RTR-A ETH1) | /24 | ETH2 |
| 192.168.0.0 (Red B) | 192.168.1.1 (RTR-A ETH1) | /25 | ETH2 |
| 175.100.0.128 (Red C) | 0.0.0.0 | /27 | ETH0 |
| 192.168.0.128 (Red E y D) | 192.168.1.1 (RTR-A ETH1) | /26 | ETH2 |
| 205.10.0.64 (Red F) | 192.168.1.22 (RTR-C ETH1) | /26 | ETH1 |
| 192.168.1.0 | 0.0.0.0 | /29 | ETH2 |
| 192.168.1.20 | 0.0.0.0 | /30 | ETH1 |
| 0.0.0.0 | 192.168.1.22 (RTR-C ETH1) | /0 | ETH1 |

RTR-C

En este router se pueden sumarizar la red E y la red D. Son consecutivas y pares.

|  |  |  |  |
| --- | --- | --- | --- |
| **Direccion** | **Gateway** | **Máscara** | **Interfaz** |
| 192.168.10.0 (Red A) | 192.168.1.21 (RTR-B ETH1) | /24 | ETH1 |
| 192.168.0.0 (Red B) | 192.168.1.21 (RTR-B ETH1) | /25 | ETH1 |
| 175.100.0.128 (Red C) | 192.168.1.21 (RTR-B ETH1) | /27 | ETH1 |
| 192.168.0.128 (Red D y E) | 192.168.1.25 (RTR-F ETH2) | /26 | ETH0 |
| 205.10.0.64 (Red F) | 192.168.1.25 (RTR-F ETH2) | /26 | ETH0 |
| 192.168.1.20 | 0.0.0.0 | /30 | ETH1 |
| 192.168.1.24 | 0.0.0.0 | /30 | ETH0 |
| 205.10.0.0 | 0.0.0.0 | /30 | ETH2 |
| 0.0.0.0 | 205.10.0.2 | /30 | ETH2 |

RTR-D

|  |  |  |  |
| --- | --- | --- | --- |
| **Direccion** | **Gateway** | **Máscara** | **Interfaz** |
| 192.168.10.0 (Red A) | 192.168.1.1 (RTR-A ETH1) | /24 | ETH0 |
| 192.168.0.0 (Red B) | 192.168.1.1 (RTR-A ETH1) | /25 | ETH0 |
| 175.100.0.128 (Red C) | 192.168.1.3 (RTR-B ETH2) | /27 | ETH0 |
| 192.168.0.128 (Red D y E) | 192.168.1.13 (RTR-E ETH0) | /26 | ETH1 |
| 205.10.0.64 (Red F) | 192.168.1.18 (RTR-F ETH0) | /26 | ETH2 |
| 192.168.1.12 | 0.0.0.0 | /30 | ETH1 |
| 192.168.1.0 | 0.0.0.0 | /29 | ETH0 |
| 192.168.1.16 | 0.0.0.0 | /30 | ETH2 |
| 0.0.0.0 | 192.168.1.18 (RTR-F ETH0) | /0 | ETH2 |

RTR-E

|  |  |  |  |
| --- | --- | --- | --- |
| **Direccion** | **Gateway** | **Máscara** | **Interfaz** |
| 192.168.10.0 (Red A) | 192.168.1.9 (RTR-A ETH2) | /24 | ETH1 |
| 192.168.0.0 (Red B) | 192.168.1.9 (RTR-A ETH2) | /25 | ETH1 |
| 175.100.0.128 (Red C) | 192.168.1.9 (RTR-A ETH2) | /27 | ETH1 |
| 192.168.0.192 (Red D) | 0.0.0.0 | /26 | ETH3 |
| 192.168.0.128 (Red E) | 0.0.0.0 | /26 | ETH2 |
| 205.10.0.64 (Red F) | 192.168.1.14 (RTR-D ETH1) | /26 | ETH0 |
| 192.168.1.8 | 0.0.0.0 | /30 | ETH1 |
| 192.168.1.12 | 0.0.0.0 | /30 | ETH0 |
| 0.0.0.0 | 192.168.1.14 (RTR-D ETH1) | /0 | ETH0 |

RTR-F

|  |  |  |  |
| --- | --- | --- | --- |
| **Direccion** | **Gateway** | **Máscara** | **Interfaz** |
| 192.168.10.0 (Red A) | 192.168.1.17 (RTR-D ETH2) | /24 | ETH0 |
| 192.168.0.0 (Red B) | 192.168.1.17 (RTR-D ETH2) | /25 | ETH0 |
| 175.100.0.128 (Red C) | 192.168.1.26 (RTR-C ETH0) | /27 | ETH2 |
| 192.168.0.128 (Red D y E) | 192.168.1.17 (RTR-D ETH2) | /26 | ETH0 |
| 205.10.0.64 (Red F) | 0.0.0.0 | /26 | ETH1 |
| 192.168.1.24 | 0.0.0.0 | /30 | ETH2 |
| 192.168.1.16 | 0.0.0.0 | /30 | ETH0 |
| 0.0.0.0 | 192.168.1.26 (RTR-C ETH0) | /0 | ETH2 |

**11. Si se desconecta SWT3, ¿cómo quedaría la tabla de ruteo de RTR-D para que la red completa siga funcionando según las condiciones establecidas?**

Tendría que tener dos entradas que reemplacen a la entrada actual que apunta a RTR-A y RTR-B. La red actual que forman los routers RTR-A, RTR-D y RTR-B tendría que ser en realidad tres redes, de dos host cada una (una entre RTR-A ETH1 y RTR-D ETH0, otra entre RTR-B ETH2 y RTR-A ETH1, y otra entre RTR-B ETH2 y RTR-D ETH3).

Para las condiciones establecidas, de esas nuevas 3 redes, solo serían necesarias una entre RTR-A ETH1 y RTR-D ETH0, y otra entre RTR-B ETH2 y RTR-A ETH1, ya que *“RTR-D llega a Red A y B a través de red a través de RTR-A.”*

**12. Siguiendo con el punto anterior, ¿es necesario modificar la tabla de ruteo de algún otro router? ¿Cuáles?**

Si, hay que modificar las tablas de RTR-A y RTR-B, ya que las redes que tienen directamente conectadas van a cambiar.

**13. Si se estuviese ejecutando un protocolo de ruteo dinámico, ¿cómo se solucionaría este problema?**

Si se usara un protocolo de ruteo dinamico, los distintos routers se encargaran (mediante algoritmos de ruteo dinamico) de llenar de forma automatica sus tablas de ruteo. Los routers intercambian información sobre las rutas o routers que conocen y con esa informacion calculan el siguiente salto correcto para llegar a una red destino. Esta tecnica es muy eficiente cuando se tiene redes de gran tamaño. Implica menos intervenciones por parte del ingeniero de red.

**Capa de enlace**

**14. Con respecto a la pregunta 1 de la Capa de Aplicación, responder:**

**a. Suponga que se encuentra analizando los paquetes que entran y salen de la placa de red de PC-A.**

**¿Qué información (headers) de capa 2, 3 y 4 son transportados en el primer paquete de la comunicación de cada protocolo necesario para realizar dicha consulta? Indique los paquetes que envía PC-A y las respuestas que obtiene. Considere que todas las tablas están vacías.**

Primero que nada, PC-A se tiene que comunicar con su servidor DNS, Srv\_DNS. Para esto, usa el proceso ANDing y se da cuenta que Srv\_DNS no está en la misma red que él. Será necesario que el paquete pase por al menos un router para poder llegar a destino.

Suponiendo que las tablas ARP están vacías, PC-A tiene que hacer una consulta ARP para averiguar la MAC de RTR-A ETH0 y poder comunicarse con él. En este caso participan capa 3 (capa de red) y capa 2 (capa de enlace):

* Capa 3, el protocolo es IP. La IP origen es la de PC-A, y la IP destino es la de RTR-A ETH0.
* Capa 2: Se usan tanto el protocolo ETHERNET como el protocolo ARP (ARP viaja dentro del protocolo ETHERNET).
  + En ARP, la MAC origen es la MAC de PC-A. La MAC destino es 00-00-00-00-00-00.
  + En Ethernet, la MAC origen es la MAC de PC-A. La MAC destino es FF-FF-FF-FF-FF-FF.

La solicitud ARP viaja por broadcast y por ende llega a todos los equipos de la red. Cada equipo va a tomar la trama, desempaquetarla, y ver si la IP destino se corresponde con su propia IP. Si se corresponde, entonces la procesa. Si no se corresponde, la descarta.

Cuando le llegue a RTR-A ETH0, va a darse cuenta que es para él. En este momento, genera una ARP REPLY, la cual no viaja por broadcast. La información, capa por capa es:

* Capa 3: protocolo IP. IP origen es la IP de RTR-A ETH0, la IP destino es la IP de PC-A.
* Capa 2: protocolo Ethernet y ARP (ARP dentro de Ethernet).
  + Tanto en Ethernet como en ARP, la MAC origen es la de RTR-A ETH0, y la MAC destino es la de PC-A.

Una vez que llega dicha respuesta a PC-A, dicha PC ya sabe la MAC del router que le permite salir de su red (recordar que necesita salir para hacer una consulta DNS). La consulta DNS involucra las siguientes capas:

* Capa 4: en la sección de datos lleva la consulta DNS de la capa de aplicación. En la cabecera, los puertos origen y destino. Puerto origen es uno mayor a 1023, puerto destino es 53 (puerto por defecto de DNS). La consulta DNS viaja normalmente sobre UDP.
* Capa 3: protocolo IP. IP origen es la IP de PC-A. IP destino es la IP de Srv\_DNS.
* Capa 2: protocolo Ethernet. MAC origen es la MAC de PC-A, la MAC destino es la de RTR-A ETH0.

Vale aclarar que en cada salto, los routers cambian la información de la capa 2 (setean nuevas MAC origen y destino), ya que la comunicación usando Ethernet solo puede ser entre dispositivos de una misma red.

Una vez que el servidor DNS responda, la información va a llegar en algún momento al router RTR-A. Desde este router hasta PC-A, la información va a ser:

* Capa 4: en la sección de datos, viene la información que se pidió en la consulta DNS (registro A). El puerto origen es 53 y el destino es uno mayor a 1023.
* Capa 3: protocolo IP. IP origen va a ser la IP de Srv\_DNS, IP destino va a ser la IP de PC-A.
* Capa 2: protocolo Ethernet. MAC origen va a ser la MAC de RTR-A ETH0, MAC destino va a ser la MAC de PC-A.

Luego de esto, PC-A ya puede conectarse al servidor para hacer su petición HTTP. La información será:

* Capa 4: en la sección de datos lleva la consulta HTTP. En la cabecera, los puertos origen y destino. Puerto origen es uno mayor a 1023, puerto destino es 80 (puerto por defecto de HTTP). La consulta HTTP viaja normalmente sobre TCP.
* Capa 3: protocolo IP. IP origen es la IP de PC-A. IP destino es la IP del servidor de ‘[www.redes.info.edu.ar](http://www.redes.info.edu.ar)’.
* Capa 2: protocolo Ethernet. MAC origen es la MAC de PC-A, la MAC destino es la de RTR-A ETH0.

En la respuesta, la información será:

Capa 4: en la sección de datos, viene la información que se pidió en la consulta HTTP. El puerto origen es 80 y el destino es mayor a 1023.

* Capa 3: protocolo IP. IP origen va a ser la IP del servidor de ‘[www.redes.info.edu.ar](http://www.redes.info.edu.ar)’, IP destino va a ser la IP de PC-A.
* Capa 2: protocolo Ethernet. MAC origen va a ser la MAC de RTR-A ETH0, MAC destino va a ser la MAC de PC-A.

**b. Indique cómo quedarían las tablas de los switches Swt-1 y Swt-3 después que finaliza todo el intercambio de mensajes.**

Swt-1:

|  |  |
| --- | --- |
| **Dirección de MAC** | **Puerto** |
| MAC PC-A | 4 |
| MAC RTR-A ETH0 | 3 |

Swt-3:

|  |  |
| --- | --- |
| **Dirección de MAC** | **Puerto** |
| RTR-A ETH1 | 1 |
| RTR-B ETH2 | 2 |

**15. Si a continuación del punto anterior, PC-A desea enviar un ping a:**

**a. PC-J. ¿Cuál sería el primer mensaje que enviaría?**

Para mandarle un ping, PC-J tiene que usar el protocolo ICMP (solicitud de eco). ICMP pertenece a la capa 3, trabaja justo encima de IP.

Sin embargo, PC-A usaria el proceso ANDing y se daría cuenta que PC-J no está en su misma red. Por ende, tiene que hacer una consulta ARP para averiguar la IP de la interfaz del router al que está conectado, y así poder salir de su red y comunicarse con otras computadoras (en este caso, PC-J).

Una vez obtenida la información, se podría comunicar con PC-J. Es esperable que PC-J responda con el mensaje ICMP eco reply.

**b. PC-C. ¿Cuál sería el primer mensaje que enviaría?**

Para comunicarse con PC-C, primero realiza el proceso ANDing y se da cuenta que PC-C está en la misma red. Sin embargo, puede no saber la MAC de PC-C:

* Si la sabe, se comunica directamente.
* Si no la sabe, realiza una consulta ARP para obtener su dirección de MAC, y luego sí comunicarse con PC-C.

**16. ¿Cómo quedaría la tabla de Swt-4 después que se produce el siguiente intercambio de mensajes (suponga que todas las tablas están vacías)? Indicar cómo se modifica la tabla del switch a medida que pasan los mensajes (considerar que cada punto involucra todo lo necesario para poder enviar únicamente el mensaje correspondiente).**

**a. PC-G envía mensaje DNS a Srv\_DNS.**

PC-G usando el proceso ANDing comprueba que Srv\_DNS no está en su red. Tiene que hacer una solicitud ARP para conocer la MAC de su router.

|  |  |
| --- | --- |
| **Dirección MAC** | **Puerto** |
| MAC PC-G | 3 |
| RTR-E ETH3 | 1 |

**b. PC-L envía un echo-request a PC-F.**

PC-L usando el proceso ANDing comprueba que PC-F está en su red. Sin embargo, no conoce su MAC. Tiene que hacer una solicitud ARP para averiguarla. PC-F responderá (ARP REPLY).

|  |  |
| --- | --- |
| **Dirección MAC** | **Puerto** |
| MAC PC-G | 3 |
| RTR-E ETH3 | 1 |
| MAC PC-L | 4 |
| MAC PC-F | 2 |

**c. PC-I envía un echo-request a PC-E.**

No se modifica.

|  |  |
| --- | --- |
| **Dirección MAC** | **Puerto** |
| MAC PC-G | 3 |
| RTR-E ETH3 | 1 |
| MAC PC-L | 4 |
| MAC PC-F | 2 |

**d. RTR-E envía la respuesta de DNS del punto a) a PC-G.**

No cambia. La MAC de RTR-E ya la había aprendido con la respuesta a la solicitud ARP del punto a).

|  |  |
| --- | --- |
| **Dirección MAC** | **Puerto** |
| MAC PC-G | 3 |
| RTR-E ETH3 | 1 |
| MAC PC-L | 4 |
| MAC PC-F | 2 |

**e. PC-G envía un mensaje HTTP un servidor en Internet.**

No se modifica.

|  |  |
| --- | --- |
| **Dirección MAC** | **Puerto** |
| MAC PC-G | 3 |
| RTR-E ETH3 | 1 |
| MAC PC-L | 4 |
| MAC PC-F | 2 |

**17. Dentro de la Red D, ¿qué nodos recibirán cada uno de los mensajes intercambiados en el punto 3?**

En el punto 3 tenemos dos pasos: primero Roger con su PC-G envía el mail a su servidor de correo de ‘redes.info.unlp.edu.ar’. Luego, el servidor de correo de ‘redes.info.unlp.edu.ar’ envía el correo al servidor de ‘example.com’

Para el primer paso:

* PC-G necesita conocer la MAC de RTR-E ETH3. Para eso hace una solicitud ARP por broadcast, y por ende, lo reciben todos los nodos.
* El router responde de manera unicast (lo recibe solo PC-G).
* Finalmente PC-G envía el correo su servidor de correo. Para esto, le envía la trama a su router.

Para el segundo paso: asumo que el servidor de correo de ‘redes.info.unlp.edu.ar’ es Srv\_mail (red F)

* Srv\_Mail hace una solicitud ARP para conocer la MAC de su router (le llega a todos los nodos de la red).
* El router responde (ARP reply) de manera unicast (lo recibe solo Srv\_Mail).
* Srv\_Mail hace una consulta por los registros MX para ‘example.com’. Lo recibe solo el router.
* En la respuesta, el router le responde solo a Srv\_Mail.
* Srv\_Mail hace una consulta por el registro A del servidor de mail mayor prioridad.
* En la respuesta, el router le responde solo a Srv\_Mail.
* Finalmente Srv\_Mail establece la conexión (con saludo de tres vías de TCP incluido) y envía la información. Srv\_Mail se comunica con su router.

**18. ¿Cómo quedaría la tabla ARP de RTR-E una vez finalizado todo el intercambio?**

|  |  |
| --- | --- |
| **Dirección MAC** | **Dirección IP** |
| MAC PC-G | IP PC-G |
| MAC RTR-A | IP RTR-A |