Redes de Computadoras

Solución Examen – 13 de febrero de 2022

Instrucciones

- **Indique su nombre completo y número de cédula en cada hoja.**
- **Numere todas las hojas e indique la cantidad total de hojas que entrega en la primera.**
- **Escriba las hojas de un solo lado y utilice una caligrafía claramente legible.**
- **Comience cada pregunta teórica y cada ejercicio en una hoja nueva.**
- **Solo se responderán dudas de letra. No se responderán dudas de ningún tipo los últimos 30 minutos del examen.**
- **El examen es individual y sin material. Apague su teléfono celular mientras esté en el salón del examen.**
- **Es obligatorio responder correctamente al menos 15 puntos en las preguntas teóricas y 20 de los problemas prácticos. Los puntos ganados en el curso se suman a los puntos de teórico.**
- **El puntaje mínimo de aprobación es de 60 puntos.**
- **Para todos los ejercicios, si es necesario, puede suponer que dispone de los tipos de datos básicos (p.ej. lista, cola, archivo, string, etc.) y sus funciones asociadas (ej: tail(lista), crear(archivo), concatenar(string, string).**
- **Justifique todas sus respuestas.**
- **Duración: 3 horas. Culminadas las 3 horas el alumno no podrá modificar las hojas a entregar de ninguna forma.**

Preguntas Teóricas

Pregunta 1 (8 puntos)

a) Considere el protocolo HTTP en su versión 1.1.
   i. Muestre y explique el formato general de un mensaje de respuesta HTTP.
   ii. ¿Como hace el receptor del mensaje para diferenciar cada campo de la cabecera?
   iii. ¿Como hace el receptor del mensaje para diferenciar la cabecera del mensaje del cuerpo?
   iv. ¿Es necesario para el receptor conocer el final del mensaje? Justifique, y si la respuesta es afirmativa explique como lo hace.

b) Considere ahora el protocolo SMTP:
   i. Compare ambos protocolos en relación a su objetivo, las entidades que participan, quien inicia la conexión y el formato de datos en los mensajes.
   ii. ¿Es necesario para el receptor conocer el final del mensaje? Justifique y si la respuesta es afirmativa explique como lo hace.

Solución Pregunta 1

a)
i) El mensaje consta de una línea de estado, donde se indica si la respuesta es correcta o si hay algún error. Esta línea consta de tres campos: versión, código estado y frase que explica el código.
Luego sigue la cabecera, que consta de varias líneas. Cada línea corresponde a un campo que se identifica con el nombre del campo y le sigue el valor.
Por último, se tiene el cuerpo del mensaje, que es el que contiene el objeto solicitado.

ii) Cada línea de la cabecera representa un campo con el formato nombre<SP>valor, donde <SP> es el caracater de espacio. Cada línea se separa de la siguiente con los códigos <CR> (retorno de carro) y <LF> (salto de línea).

iii) Lo hace mediante una línea que solo incluye los códigos de control <CR> (retorno de carro) y <LF> (salto de línea).

iv) Dado que HTTP viaja sobre TCP y en TCP no existe el concepto de mensaje o datagrama que nos permita delimitar un mensaje, entonces es necesario para el receptor determinar en que parte del stream de datos finaliza un mensaje y comienza otro.
Para esto, en HTTP1.1 se incluye un campo de Content length, que indica el largo del cuerpo del mensaje. De esta forma, el receptor sabe cuantos datos leer del stream que corresponden al mensaje.

b)
i) Ambos protocolos tienen el mismo objetivo, transferir información (p.e. archivos) de un host a otro

host. En el caso de HTTP la transferencia es entre un host cliente y un host servidor mientras que en SMTP, además del intercambio cliente servidor (p.e. thunderbird) también ocurre entre dos servidores de correo (relay). Una diferencia importante es que en el caso de HTTP, el cliente que desea obtener un archivo del servidor, inicia la conexión y solicita el archivo. En cambio, en SMTP, es la entidad que tiene el archivo quien inicia la conexión y envía el archivo a la otra entidad. En cuanto al formato de los mensajes, SMTP requiere que cada mensaje, incluyendo el cuerpo, esté en formato ASCII de 7 bits mientas que HTTP no impone esta restricción.

ii) Si es necesario, por la misma razón dada para HTTP. Sin embargo, en este caso el final del mensaje se identifca con una línea con un caractar especial, el punto.

Pregunta 2 (6 puntos)
Considere el algoritmo Path MTU Discovery en IPv6.

a) Fundamente la necesidad del mismo.
b) Fundamente por qué no es un requerimiento en IPv4.

Solución Pregunta 2
a) Los routers no realizan fragmentación en IPv6, por lo que, si un PDU de IPv6 excede el MTU es descartado. Es importante para cualquier par de entidades que se comuniquen, asegurarse que sus mensajes lleguen a destino, y para ello, sus mensajes no deben exceder el mínimo MTU del camino por el que transitan.

b) La definición de IPv4 involucra que, cuando segmentos de determinado flujo arriban a un enlace con un MTU inferior, se realiza fragmentación y reensamblado en dicho enlace, sin que las entidades origen y destino de la comunicación eventualmente lleguen a enterarse. Los extremos de la comunicación pueden ignorar el mínimo MTU del camino, pues, los routers intermedios lo resuelven.

Pregunta 3 (8 puntos)

a) Explique el concepto de enrutamiento jerárquico, y señale dos razones por las cuales es necesario en Internet.
b) Mencione y explique brevemente el funcionamiento de los protocolos de enrutamiento que implementan este concepto a nivel inter-dominio e intra-dominio.

Solución Pregunta 3

a) Dada una red o conjunto de redes de gran escala como la Internet, el enrutamiento jerárquico consiste en dividir la o las redes en distintas zonas que implementan un enrutamiento detallado, interconectadas entre sí para lograr encaminar tráfico entre distintas zonas; cada host o router conoce en detalle su zona (vecindario), pero solo un resumen del resto de la/las redes. Este mecanismo es necesario en Internet por su **escala**, que hace inviable que todos los nodos de la red ejecuten un mismo protocolo de enrutamiento, y por la **autonomía administrativa** de distintas zonas de la red. EN el caso inter-dominio estas zonas se denominan **Sistemas Autónomos**, y en el caso intra-domnio, estas zonas se denominan **áreas** de enrutamiento.

b) A nivel inter-dominio, el protocolo utilizado en Internet es el Border Gateway Protocol (BGP), que implementa el concepto de Sistemas Autónomos, identificados por su Número de Sistema Autónomo (ASN). BGP es un protocolo de tipo “path-vector”, similar a los protocolos “distance-vector”, donde la métrica habitual es la longitud medida en cantidad de Ases entre un origen y un destino (prefijos de red); a esta métrica se le denomina “AS_PATH lenght”. BGP incorpora el concepto de “policy routing”, que implica que cada Sistema Autónomo puede modificar los atributos asociados a la alcanzabilidad de los prefijos de red, de forma de favorecer determinado objetivo administrativo.

A nivel intra-dominio, un protocolo que implementa el enrutamiento jerárquico es Open Shortest Path

First (OSPF). Un sistema autónomo que usa OSPF se puede configurar jerárquicamente en áreas. Cada área ejecuta su propio algoritmo de enrutamiento de estado de enlace OSPF, con cada enrutador en un área transmitiendo su estado de enlace a todos los demás enrutadores en esa área. Dentro de cada área, uno o más enrutadores de borde de área son responsables de enrutar paquetes fuera del área. El esquema jerárquico se completa con un área configurada para ser el "backbone", cuyo cometido es encaminar el tráfico entre las otras áreas en el AS. El backbone siempre contiene todos los enrutadores de borde de área en el AS y puede contener otros enrutadores también. El encaminamiento de un paquete entre áreas dentro del AS requiere que el paquete sea primero enrutado a un enrutador de borde de área (enrutamiento dentro del área), luego enrutado a través del backbone al enrutador de borde de área que está en el área de destino, y luego enrutado al destino final.

Pregunta 4 (10 puntos)

a) ¿Cómo percibe un emisor TCP que existe congestión en la ruta entre él mismo y el destino?
b) Mencione los tres principios del mecanismo de control de congestión de TCP.
c) Considere el algoritmo de control de congestión de TCP. En la fase de *Evitación de la Congestión*, si MSS es igual a 1.260 bytes y VentCongestion es igual a 12.600 bytes, ¿Cuántos segmentos se enviarán en 1 RTT? Justifique su respuesta.

Solución Pregunta 4

a) Un "suceso de pérdida" en un emisor TCP como el hecho de que se produzca un fin de temporización o se reciban tres paquetes ACK duplicados procedentes del receptor.

b)
- Un segmento perdido implica congestión y, por tanto, la velocidad del emisor TCP debe reducirse cuando se pierde un segmento.
- Un segmento que ha sido reconocido indica que la red está entregando los segmentos del emisor al receptor y, por tanto, la velocidad de transmisión del emisor puede incrementarse cuando llega un paquete ACK correspondiente a un segmento que todavía no había sido reconocido. La llegada de paquetes de reconocimiento se interpreta como una indicación implícita de que todo funciona bien. Luego se puede aumentar el tamaño de la ventana de congestión.
- Tanteo del ancho de banda. Puesto que los paquetes ACK indican que la ruta entre el origen y el destino está libre de congestión y la pérdida de paquetes indica que hay una ruta congestionada, la estrategia de TCP para ajustar su velocidad de transmisión consiste en incrementar su velocidad en respuesta a la llegada de paquetes ACK hasta que se produce una pérdida, momento en el que reduce la velocidad de transmisión. El emisor TCP incrementa entonces su velocidad de transmisión para tantear la velocidad a la que de nuevo aparece congestión, retrocede a partir de ese punto y comienza de nuevo a tantear para ver si ha variado la velocidad a la que comienza de nuevo a producirse congestión.

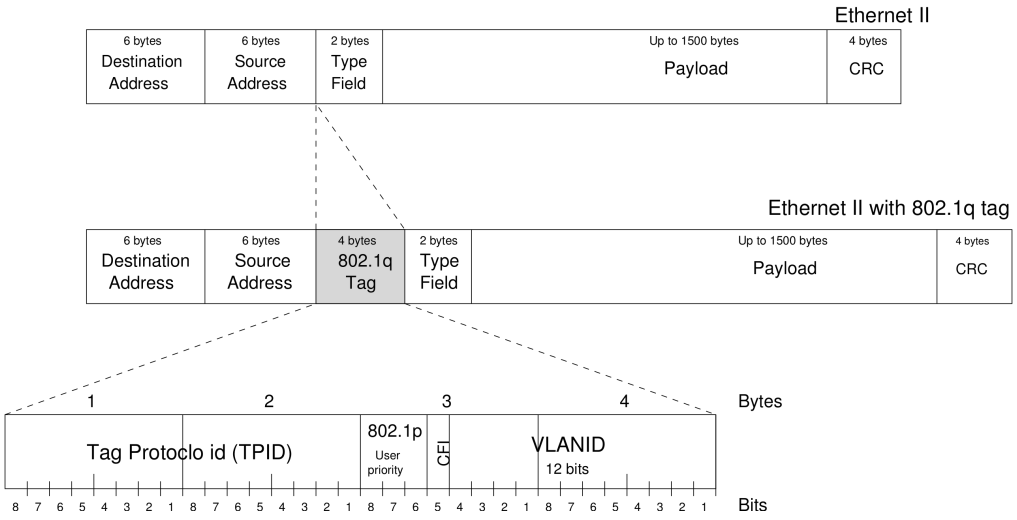
c) Si MSS es igual a 1.260 bytes y VentCongestion es igual a 12.600 bytes, entonces se enviarán 10 segmentos en un RTT.

Pregunta 5

Sea el protocolo 802.1Q, de etiquetado para VLANs.

a) Describa las modificaciones que introduce a nivel de los cabezales de capa 2.
b) Explique cómo se combina este tráfico en los *switches* con el tráfico *Ethernet* original que no utiliza ese etiquetado.

a) El protocolo 802.1Q involucra una redefinición del cabezal Ethernet, que introduce cuatro bytes adicionales luego de las direcciones MAC src y dst:



de los 32bits adicionales, se reservan 12 para identificar VLANs.

b) A nivel de los switches se distinguen puertos a los que se pueden conectar dispositivos que manipulan los cabezales extendidos y son capaces de intercambiar tráfico que incluye TAGS de los puertos en los que se conectan equipos que desconocen del uso de tags. Estos puertos, los que reciben tráfico sin tags se conocen como UNTAGGED PORTS y realizan la tarea de etiquetar tráfico al ingreso y quitar el tag al momento de forwardear el paquete hacia el host. Los puertos por los que se transmiten tramas extendidas se conocen tomo TRUNK PORTS. Cabe mencionar que la distinción es administrativa, y no hay diferencias físicas en los puertos.

Problema 1 (30 puntos)

Se desea implementar un sistema de entrega de números para atención de usuarios, con varios puestos de atención, un servidor y un conjunto de *displays* para informar el siguiente usuario (número) a ser atendido. Todos los equipos se encuentran conectados en la misma red local.

El proceso servidor se inicia y queda a la espera de que los puestos de atención soliciten atender usuarios.

Los puestos de atención, al iniciarse, se conectan por TCP al servidor (ya conocido), y van solicitando usuarios (números) para atender hasta que finaliza el turno.

Los *displays* reciben mensajes UDP con la información necesaria para desplegar los datos de asignación de números a los puestos de atención. El servidor no conoce los *displays* que existen en la red.

La comunicación entre los nodos del sistema utiliza los siguientes protocolos basados en texto:

- Comunicación TCP entre puestos de atención y el servidor:
  - **`ID_PUESTO <id_puesto>\n`**
    El puesto se anuncia con el servidor.
  - **`SOL_NUMERO\n`**
    El puesto solicita al servidor un número para ser atendido.
  - **`ASIG_NUMERO <numero>\n`**
    El servidor informa al puesto de atención el número de usuario a atender.
  - **`CIERRE_PUESTO\n`**
    El puesto informa que no atenderá más usuarios.

- Comunicación UDP entre el servidor y los *displays*:
  - **`SIG_NUMERO <id_puesto> <numero>\n`**
    El servidor informa a los *displays* una asignación de un número a un puesto.

Se pide:
Implemente en un lenguaje de alto nivel, usando la API de sockets del curso y los protocolos definidos anteriormente:

a) El procedimiento que ejecuta el *puesto de atención*. El identificador del puesto es una constante única dada *ID_PUESTO* presente en cada puesto. La IP y el puerto del servidor son conocidos por los puestos. Además, cuentan con una función bloqueante `bool atendiendo_numero(int numero)` que se invoca cuando se atiende un usuario y se desbloquea automáticamente cuando la atención del actual usuario termina. La función retorna `false` si finaliza el turno y `true` si debe solicitar otro usuario.

b) El procedimiento que ejecuta el servidor para la atención del servicio. Puede asumir que los números se asignan en orden secuencial, y que siempre hay usuarios esperando a ser atendidos. Además, puede asumir que el servidor conoce el puerto UDP donde reciben la información los displays.

Solución Problema 1

a) *El procedimiento que ejecuta el puesto de atención*. El identificador del puesto es *una constante única dada ID_PUESTO* presente en cada puesto. La IP y el puerto del servidor son conocidos por los puestos. Además, cuentan con una función bloqueante `bool atendiendo_numero(int numero)` que se invoca cuando se atiende un usuario y se desbloquea automáticamente cuando la atención del

actual usuario termina. La función retorna *false* si finaliza el turno y *true* si debe solicitar otro usuario.

```
function cliente_atencion()
  //creo socket para conectar atencion de puesto por TCP
  master = socket.tcp();
  client, err = master.connect(NODE_CENTRAL_IP, 5555)
  if err then return end
  //envio identificacion
  msg= 'ID_PUESTO $ID_PUESTO\n'
  repeat
    msg, err = client.send(msg)
  until (msg=='' or err=='closed')
  if err then return end
  repeat
    //armo mensaje de solicitud de proximo cliente
    msg = 'SOL_NUMERO\n'
    repeat
      msg, err = client.send(msg)
    until (msg=='' or err=='closed')
    if err then return end
    buff = ''

    repeat
      r, err = client.receive()
      if err then return end
      buff = buff + r
      //busco primer \n en datos recibidos
      pos = string.pos(buff, "\n")
    until (pos > 0)

    //parseo la linea
    dato = buff.split(' ')
    if (dato[0] == "ASIG_NUMERO" )
      seguir = atendiendo_numero(dato[2])
    end
  until (seguir == FALSE)

  msg= 'CIERRE_PUESTO\n'
  repeat
    msg, err = client.send(msg)
  until msg=='' or err=='closed'

  client.close()
end
```

b) *El procedimiento que ejecuta el servidor para la atención del servicio. Puede asumir que los números se asignan en orden secuencial, y que siempre hay usuarios esperando a ser atendidos. Además, puede asumir que el servidor conoce el puerto UDP donde reciben la información los displays.*

```
SIGUIENTE_NUMERO=1
NUMERO_LOCK = semaphore()
BROADCAST_IP="255.255.255.255"


//creo socket para envio de informacion por UDP
```

```
sktUdp = socket.udp()
// creo socket para atencion de puestos por TCP
master = socket.tcp()
master.bind(*, 5555)
server = master.listen()
while true do
  client, err = server.accept()
  new.thread(atiende_cliente, client)
end
server.close() // inalcanzable

function atiende_cliente (client)
  msg = ''
  buff = ''
  // espero identificacion del nodo
  repeat
    r, err = client.receive()
    if err=="closed" then return end
    buff = buff + r
    // busco primer \n en datos recibidos
    pos = string.pos(buff, "\n")
  until (pos > 0)

  // asumo que es ID_PUESTO
  dato = buff.split(' ')
  PUESTO=dato[2]

  while true do
    //espero solicitud de numero
    //espero linea protocolo del cliente
    repeat
      r, err = client.receive()
      if err=="closed" then return end
      buff = buff + r
      // busco primer \n en datos recibidos
      pos = string.pos(buff, "\n")
    until (pos > 0)

    // parseo la linea
    dato = buff.split(' ')

    if (dato[1] == “SOL_NUMERO”) then
        // contesto asignando numero
        NUMERO_LOCK.acquire()

        msg= ‘ASIG_NUMERO ‘+SIGUIENTE_NUMERO+’+’\n’
        repeat
          msg, err = client.send(msg)
        until (msg=='' or err=='closed')

        // envio mensaje UDP a toda la red informando asignacion
        msgUdp=’SIG_NUMERO ‘+PUESTO+’ ‘+SIGUIENTE_NUMERO+’\n’
        sktUdp.sendto(msgUdp, BROADCAST_IP, 6666)
        SIGUIENTE_NUMERO++

        NUMERO_LOCK.release()
    end
    if dato[1] == “CIERRE_CLIENTE” then
      client.close()
      return
```

```
    end
  end
end
```

Problema 2 (30 puntos)

Considere un segmento de red donde se encuentran conectados tres hosts y un router: HostA, HostB, HostC y R1. R1 contiene un servidor DNS y está configurado como gateway para los tres hosts. Antes de comenzar a responder lea completamente cada pregunta.

Se pide que todos los intercambios de paquetes y el posible orden en que ocurren estén justificados. En todos los casos que corresponda, deberá completar una tabla con el siguiente formato:

nodo	dst MAC	src MAC	dst IP	src IP	payload	justificación

Se pide:

a) Asuma que los cuatro nodos están interconectados mediante un switch y que HostA y HostB, debido a un error, poseen la misma dirección MAC (MAC_B). Inicialmente todas las tablas ARP están vacías. Si en HostC se ejecutan los siguientes comandos (en este orden):

```
ping -c1 IPhostB
ping -c1 IPhostA
```

Muestre todos los paquetes que pasan por la interfaz de red de HostC como consecuencia de ello.

b) Considere el mismo escenario de la parte a), pero reemplazando el switch por un hub. Asuma que todas las tablas ARP están vacías. Si se ejecutan los mismos comandos de la parte a), ¿cambia en algo su respuesta anterior? Jusitifique.

c) Considere el mismo escenario de la parte a) (nodos interconectados con un switch), pero donde se corrige el problema de las MACs, asignandole MAC_A al HostA. La tabla del switch y las tablas ARP se mantienen con la información aprendida luego del intercambio de paquetes de la parte a). Si en HostC se ejecuta el siguiente comando:

```
ping -c1 IPhostB
ping -c1 IPhostA
```

Muestre todos los paquetes que pasan por la interfaz de red de HostC como consecuencia de ello. Justifique detallando el funcionamiento del switch.

d) Considere el mismo escenario de la parte anterior y manteniendo las tablas ARP y del switch aprendidas. Si en HostC se ejecuta el siguiente comando:

```
ping -c1 www.google.com
```

Muestre todos los paquetes que pasan por la interfaz de red de HostC como consecuencia de ello.

Nota
Opción -c del comando ping: indica el número de solicitudes a enviar.

Solución Problema 2

a)

Id tra,a	Nodo	Dst MAC	Src MAC	Dst IP	Src IP	Payload	Justificación
1	C	FF:..:FF	MAC_C	N/A	N/A	MACo: MAC_C IPo:IPhostC MACd: 00:..:00 IPd: IPhostB	ARP Request para determinar la dirección MAC asociada a la dirección IPhostB
2	B	MAC_C	MAC_B	N/A	N/A	MACo: MAC_B IPo:IPhostB MACd: MAC_C IPd: IPhostC	ARP Reply que informa la dirección MAC asociada a la dirección IPhostB El switch identifica en este contexto, en qué interfaz está conectada un host con MAC_B
3	C	MAC_B	MAC_C	IPhostB	IPhostC	Mensaje ICMP Echo (Request) – tipo 8	Solicitud generada por el primer comando. Una sola debido a la opción “c1”
4	B	MAC_C	MAC_B	IPhostC	IPhostB	Mensaje ICMP Echo Reply – tipo 0	Respuesta generada en función de la solicitud recibida
5	C	FF:..:FF	MAC_C	N/A	N/A	MACo: MAC_C IPo:IPhostC MACd: 00:..:00 IPd: IPhostA	ARP Request para determinar la dirección MAC asociada a la dirección IPhostA
6	A	MAC_C	MAC_B	N/A	N/A	MACo: MAC_B IPo:IPhostA MACd: MAC_C IPd: IPhostC	ARP Reply que informa la dirección MAC asociada a la dirección IPhostA El switch identifica en este contexto, en qué interfaz está conectada un host con MAC_B. Interpreta que “la MAC_B ahora está en otra interfaz)
7	C	MAC_B	MAC_C	IPhostA	IPhostC	Mensaje ICMP Echo (Request) –	Solicitud generada por el primer comando. Una sola debido a la opción “c1”

						tipo 8	
8	A	MAC_C	MAC_B	IPhostC	IPhostA	Mensaje ICMP Echo Reply – tipo 0	Respuesta generada en función de la solicitud recibida

Nota
FF:..:FF es una notación abreviada de FF:FF:FF:FF:FF:FF
00:..:00 es una notación abreviada de 00:00:00:00:00:00:00

b) Debido a que un hub únicamente implementa interconexión eléctrica entre todos sus puertos (es un dispositivo de Capa Física), no interpreta nada de capas superiores (por ejemplo de direcciones MAC) y por lo tanto no asociará direcciones MAC a puertos.

No se consideran eventuales colisiones que se puedan presentar.

La secuencia de mensajes es igual a la mostrada en la parte a.

Cuando HostC envíe la trama 3, la misma será recibida y procesada tanto por HostA como por HostB, por tener ambos la misma dirección MAC y estar conectados mediante un hub. El HostB procesará el paquete en Capa de Red por tener dirección IP destino la suya, y actuará en consecuencia (en este caso armando y enviado el mensaje ICMP Echo Reply). El HostA, cuando deba procesar el paquete en Capa de Red identificará que no está destinado a él y procederá a descartarlo.
Cuando HostC envíe la trama 7, la misma será recibida y procesada tanto por HostA como por HostB, por tener ambos la misma dirección MAC y estar conectados mediante un hub. El HostA procesará el paquete en Capa de Red por tener dirección IP destino la suya, y actuará en consecuencia (en este caso armando y enviado el mensaje ICMP Echo Reply). El HostB, cuando deba procesar el paquete en Capa de Red identificará que no está destinado a él y procederá a descartarlo.
Por lo tanto, la secuencia es la misma que la mostrada en la parte a, pero es más ineficiente el uso de los recursos, pues la presencia del hub hace que algunos nodos procesen tramas con paquetes a ser descartados.

c)

Id trama	**Nodo**	**Dst MAC**	**Src MAC**	**Dst IP**	**Src IP**	**Payload**	**Justificación**
1	C	MAC_B	MAC_C	IPhostB	IPhostC	Mensaje ICMP Echo (Request) – tipo 8	Solicitud generada por el primer comando. Una sola debido a la opción “c1”. No se requiere consultar por la dirección MAC

							asociada a la IPhostB pues HostC ya tiene una entrada en su tabla ARP
El switch, a partir de la información de su tabla, envía el mensaje por la interfaz donde está conectado HostA. Por lo tanto, será recibido por éste y al procesarlo, identificará que no está dirigido a él, descartándolo y no generando ninguna respuesta.							
2	C	MAC_B	MAC_C	IPhostA	IPhostC	Mensaje ICMP Echo (Request) – tipo 8	Solicitud generada por el segundo comando. Una sola debido a la opción “c1”. Si bien HostA ya tiene su dirección MAC correcta (MAC_A) ello todavía no es conocido por el resto de los nodos de la red. Por lo tanto, el HostC enviará el mensaje ICMP a la dirección MAC_B, pues es la información que tiene vigente en su tabla ARP
El switch, a partir de la información de su tabla, envía el mensaje por la interfaz donde está conectado HostA. Por lo tanto, será recibido por éste y al procesarlo, identificará que no está dirigido a él, descartándolo y no generando ninguna respuesta.							

d)

Id trama	**Nodo**	**Dst MAC**	**Src MAC**	**Dst IP**	**Src IP**	**Payload**	**Justificación**
1	C	FF:..:FF	MAC_C	N/A	N/A	MACo: MAC_C IPo:IPhostC MACd: 00:..:00 IPd: IP_R1	ARP Request para determinar la dirección MAC asociada a la dirección IP_R1 El motivo de ello es resolver la IP asociada al nombre “www.google.com”
2	R1	MAC_C	MAC_R1	N/A	N/A	MACo: MAC_R1 IPo:IP_R1 MACd: MAC_C IPd: IPhostC	ARP Reply que informa la dirección MAC asociada a la dirección IP_R1
3	C	MAC_R1	MAC_C	IP_R1	IPhostC	DNS request tipo A Mensaje UDP PuertoOrigen: X (>1023)	Determinar la dirección IP asociada al nombre “www.google.com”

						PuertoDestino: 53	
4	R1	MAC_C	MAC_R1	IP_R1	IPhostC	DNS reply Mensaje UDP PuertoOrigen: 53 PuertoDestino: X (>1023)	Respuesta de DNS conteniendo, al menos, una dirección IP asociada al nombre “www.google.com” Por ejemplo: 142.251.133.4
5	C	MAC_R1	MAC_C	142.251.133.4 Se acepta una mención genérica como por ejemplo IP_Google	IPhostC	Mensaje ICMP Echo (Request) – tipo 8	Solicitud generada por el comando ping. Una sola solicitud debido a la opción “c1”
6	R1	MAC_C	MAC_R1	IPhostC	142.251.133.4 se acepta una mención genérica como por ejemplo IP_Google	Mensaje ICMP Echo Reply – tipo 0	Respuesta generada en función de la solicitud recibida. Es posible que según los hosts por donde pase el mensaje Echo Request, o por una política propia de Google: o no haya respuesta (“descarte silencioso”) o que se reciba algún mensaje ICMP Type 3 (“destination unreachable”) y con algún Code como por ejemplo “Protocol Unreachable” o “Communication with Destination Host is Administratively Prohibited”.