Examen de Introducción a las Redes de Computadoras y Comunicación de Datos (ref: sirc0612.doc) 26 de diciembre de 2006

***Atención**:*

- *La duración del examen de 3 horas.*
- *El examen debe realizarse sin material.*
- *Se debe responder al menos el equivalente a 15 puntos en las preguntas teóricas.*
- *El puntaje mínimo de aprobación es de 60 puntos.*
- *para todos los ejercicios, suponga 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).*

Preguntas Teóricas

Pregunta 1 (5 puntos)

Describa el modelo de capas OSI, y las capas que lo integran.

El modelo de referencia OSI (OSI - Open System Interconection presenteado por de ISO - International Standard Organization). Modelo utilizado por la mayoría los diseñadores.

Las hipótesis manejadas para llegar a 7 capas son las siguientes:

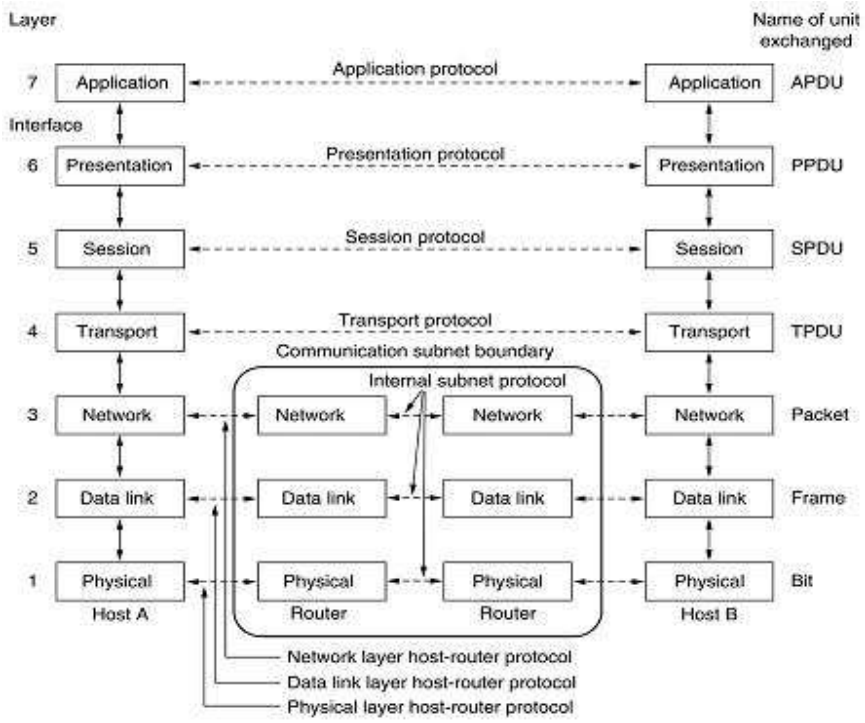
- Las capas deben delimitarse buscando minimizar el flujo de datos que se intercambia entre capas por la interfaz
- El número de capas debe ser suficientemente amplio para evitar que diferentes funciones no se requieran colocar en la misma capa.

El modelo de referencia OSI,

- **Capa Física**
  - Esta capa tiene la tarea de la transmisión de bits a través de canales de comunicación.
  - Esta capa tiene mucho que ver con las interfaces eléctricas y mecánicas para la transmisión de bits.
- **Capa Enlace de Datos**
  - La tarea fundamental es tomar un medio de transmisión, y transformarlo en una línea que parezca libre de errores.
  - En general esto involucra el generar frames de datos, y chequear la corrección de los frames llegados, así como pedir retransmisión de frames con errores o perdidos.
- **Capa de Red**
  - Se encarga de controlar el funcionamiento de la subred.
  - Su tarea fundamental es encaminar paquetes.
- **Capa de Transporte**
  - La función básica de la capa transporte es dar un flujo de datos sin errores ni fallas a la capa sesión.

- Se generará una conexión por donde se transferirá toda la información, garantizando la no existencia de:
    - perdida de información
    - duplicación de información.

- **Capa de Sesión**
    - Se busca que dos máquinas diferentes puedan establecer sesiones.
- **Capa de Presentación**
    - Busca la solución de problemas particulares, que hacen referencia a la semántica y sintaxis de lo enviado.
    - Por ejemplo la resolución de diferentes representaciones entre máquinas (pto flotante), corresponde a esta capa.
- **Capa de Aplicación.**
    - En esta capa se tratan aplicaciones que se ejecutan en la red.
    - Como ejemplo podemos tomar transferencia de archivo (ejemplo FTP), Correo Electrónico (SMTP), emulador de terminal TELNET



Pregunta 2 (10 puntos)

En capa 2 se introduce el concepto de “framing” o entramado. Explique por qué este proceso es necesario y describa:

a) una técnica para realizar “framing” en un canal orientado a bit
b) una técnica para realizar “framing” en un canal orientado a byte

Framing (Entramado):

- El framing le permite al DLC (Data Link Control) receptor, determinar cuando el frame comienza y termina.
- Adicionalmente busca la sincronización en caso de tiempos muertos de transmisión

Framing orientado a bits.

- En este caso se utilizará un flag para separar frames.
- El flag es un conjunto de bits conocidos, y para permitir la transparencia de la información se plantea el método de bit stuffing, que es la inserción de bits .

- Ejemplo:
  - Considero la flag 01111110, esto implica que la llegada de 5 1`s seguidos se debe insertar un 0

```
                0                                   0                0
0  1  1  1  1  1  1  1  0 1  0  0  0  1  1  1  1  1  1  1  1  1  1  0
```

Framing orientado a bytes.

- Bytes stuffing
  - Para implementar el modo transparente, se utiliza el siguiente procedimiento: se utiliza otro carácter de control DLE (Data Link Escape), de la siguiente forma.
  - El DLE, es insertado previamente a el carácter STX o ETX
  - ante la presencia en el frame de un DLE, el protocolo inserta uno más.
  - Garantiza transparencia de datos enviados, ya que si en los datos del frame envío DLE STX, se codificará como DLE DLE STX

Pregunta 3 (10 puntos)

Explique por qué puede ser necesaria la fragmentación de paquetes a nivel de capa 3. Explique cómo funciona la misma en IPv4 y mencione los problemas asociados a la misma. Dé un algoritmo para realizar el reensamblaje de un paquete IP a partir de sus fragmentos.

La fragmentación de paquetes supone la división de un paquete en varios paquetes más pequeños. La necesidad de fragmentación se basa en que todo medio de transmisión tiene un Maximum Transfer Unit (MTU) que puede transmitir. En caso que un paquete sea mayor que el MTU para poder ser transmitido debe fragmentarse.

Fragmentación en IPv4:

Cuando un dispositivo recibe un segmento cuyo tamaño es mayor que la MTU disponible entonces lo fragmenta en trozos que permitan su transmisión. En cada uno de los fragmentos realiza los siguientes cambios:

- El campo *total length* se ajusta al tamaño del segmento.
- La bandera MF *more fragments* se coloca en uno excepto para el último.
- Se setea el campo *fragment offset*, de acuerdo al offset del mismo en el segmento original.

Algoritmo de reensamblaje:

El algorítmico básico sería el siguiente, en la solución propuesta se supone que los fragmento llegaron en orden y el último llega con el bit MF en 0:

```
while (true) {
        recibo_paquete(p);
        if ( p.MF == 1 ){
                almaceno(p)
        }else{
                if  (p.fragment_offset <> 0 ){
                        almaceno(p);
                        p_tot = reensamblo(p);
                        if (p_tot <> NULL){
                                envo_paquete(p_tot);
                        }
                }
        }
}
```

Pregunta 4 (10 puntos)

Describa el proceso por el cual se establece una conexión TCP (three-way handshake), y cómo se cierra la misma. ¿Qué TPDU se intercambian en los mismos?.

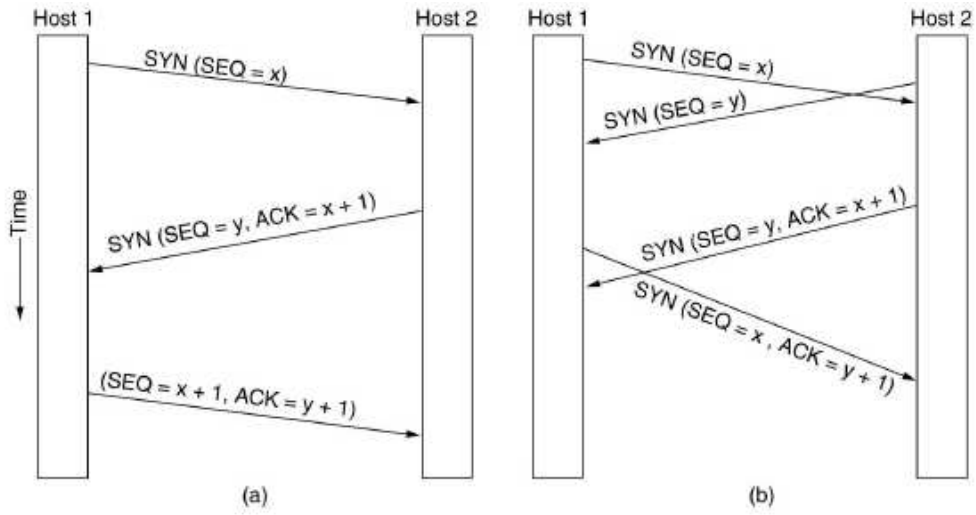
Establecimiento de la conexión en TCP:

El establecimiento se realiza con un three-way handshake.

Para iniciar una conexión se realiza un envío de un SYN, y de estar el puerto disponible para conexión, se envía un SYN ACK, enviando finalmente una confirmación ACK y de no estar se envía un RST.

También puede realizarse por intercambio de solicitudes de conexión (b), aunque no es común.

El número inicial de secuencia se acuerda en el momento de la conexión. No es 0.



Desconexión en TCP:
TCP supone la existencia de dos conexiones simplex que se desconectan independientemente.

El envío de un paquete con flag FIN, implica que yo no deseo enviar más datos al otro extremo.

Cuando se recibe la aceptación del FIN, entonces se da por cerrada la conexión en este sentido.

En general se necesitan 4 segmentos para cerrar una conexión, aunque pueden ser 3.

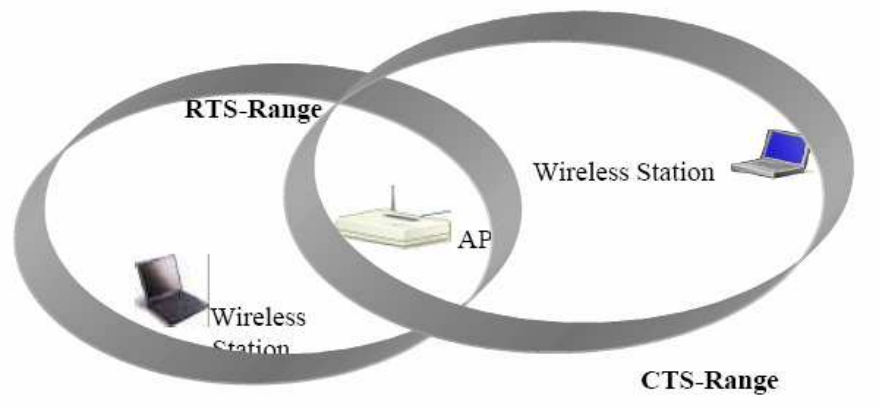
- FIN →
- ACK ←
- FIN →
- ACK ←

- FIN →
- FIN, ACK ←
- ACK →

Para evitar el problema de los dos ejércitos se colocan timers que indican que si ocurren 2 time_outs sin recibir el ACK del FIN se da por finalizada la conexión.

El del otro extremo notará que no hay nadie escuchando y cortará.

Pregunta 5 (5 puntos)
Describa el problema de “la estación oculta” que se presenta en las redes inalámbricas

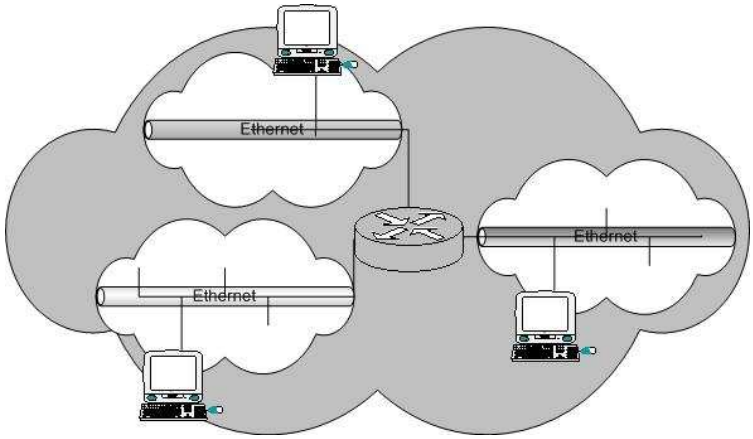


El problema de la estación oculta en las redes inalámbricas, refiere a dos estaciones las cuales el alcance de las mismas no les permite detectarse entre ellas, pero ambas tienen acceso al dispositivo Accesss Point (AP) que las intercontecta. Esto podría generar que mientras una se encuentra transmitiendo, la otra inicie una transmisión que interfiera con la primera.

Problemas Prácticos

Problema 1 (30 puntos)

Se cuenta con una red IP, la que se presenta segmentada en varios tramos ethernet los cuales son cada uno, una subred de la misma. La interconexión de las subredes mencionadas se realiza a través de un router, el cual tiene una interfaz en cada una de las subredes e interconecta la misma.

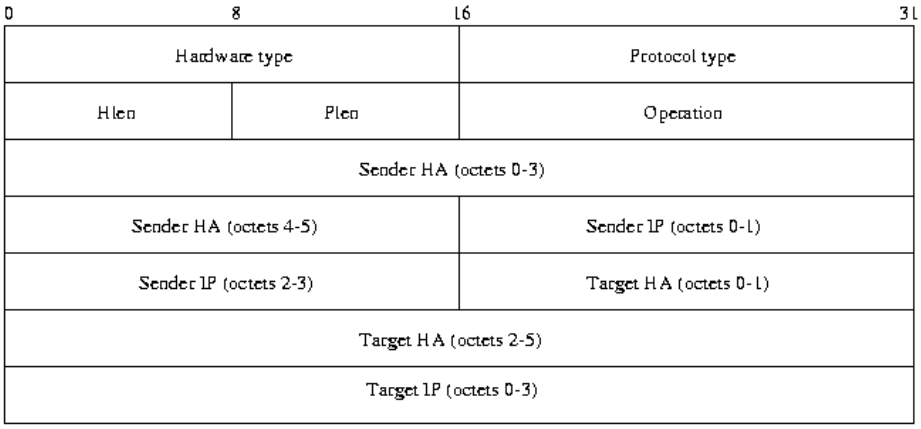


Los hosts que componen la red no conocen la existencia de las subredes y se encuentran configurados de tal manera que su conexión ethernet supone estar conectada a la red completa, independientemente de la subred a la que pertenezca.

Se cuenta con la siguiente información y procedimientos:

- El router dispone de una tabla con los componentes (interfaz, subred conectada)
- enviarARP(out interfaz: ifg, string: data)
- recibirARP(in interfaz:ifg, string: data)

El data pasado/recibido en las funciones especificadas anteriormente, tienen el formato de un mensaje ARP estándar presentado en el diagrama.



Se pide:
Especificar en un lenguaje de alto nivel, el demonio en el router que atenderá el protocolo Address Resolution Protocol (ARP), buscando que el mismo cuente con las siguientes características:

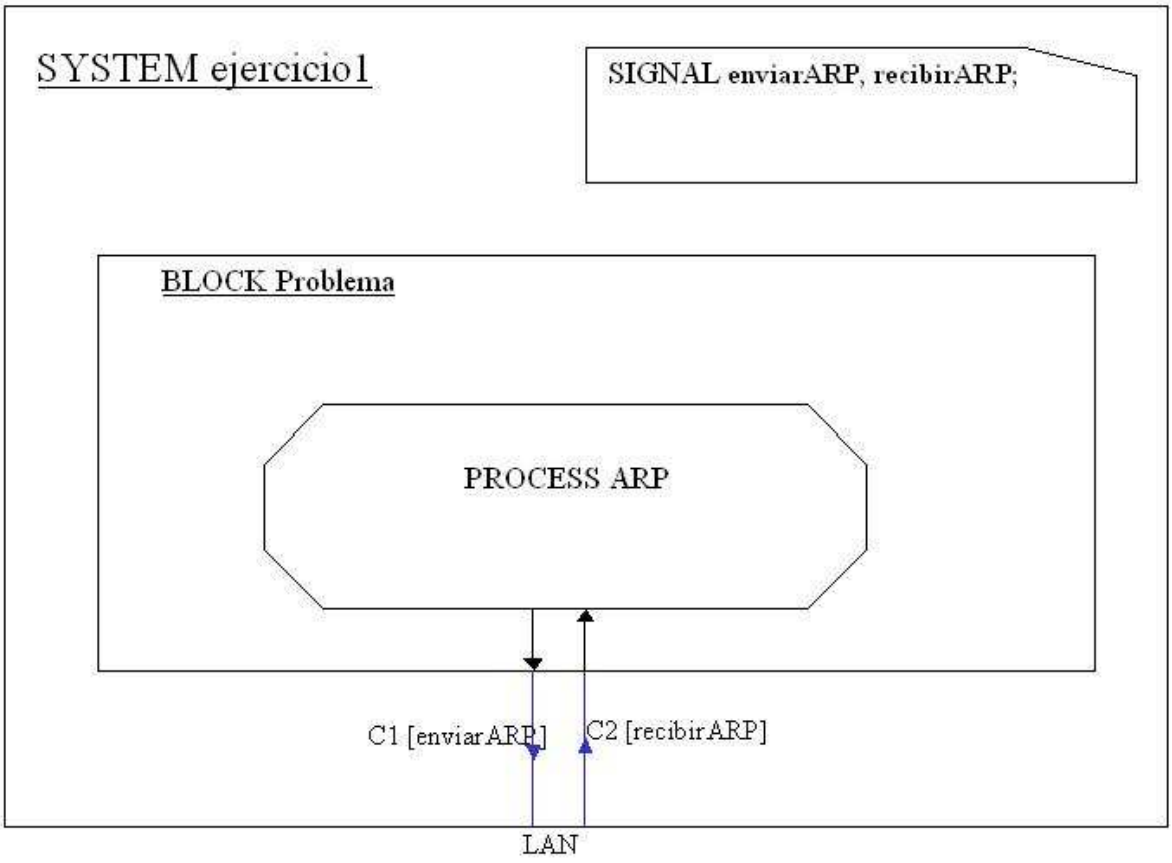
- Minimice los frames del protocolo ARP que transitan la red, en los segmentos de la red
- Funcione sin exigir reconfiguraciones de los equipos hosts.
- Ante una consulta ARP, el equipo reciba respuesta únicamente si existe un equipo con la IP buscada en la red.

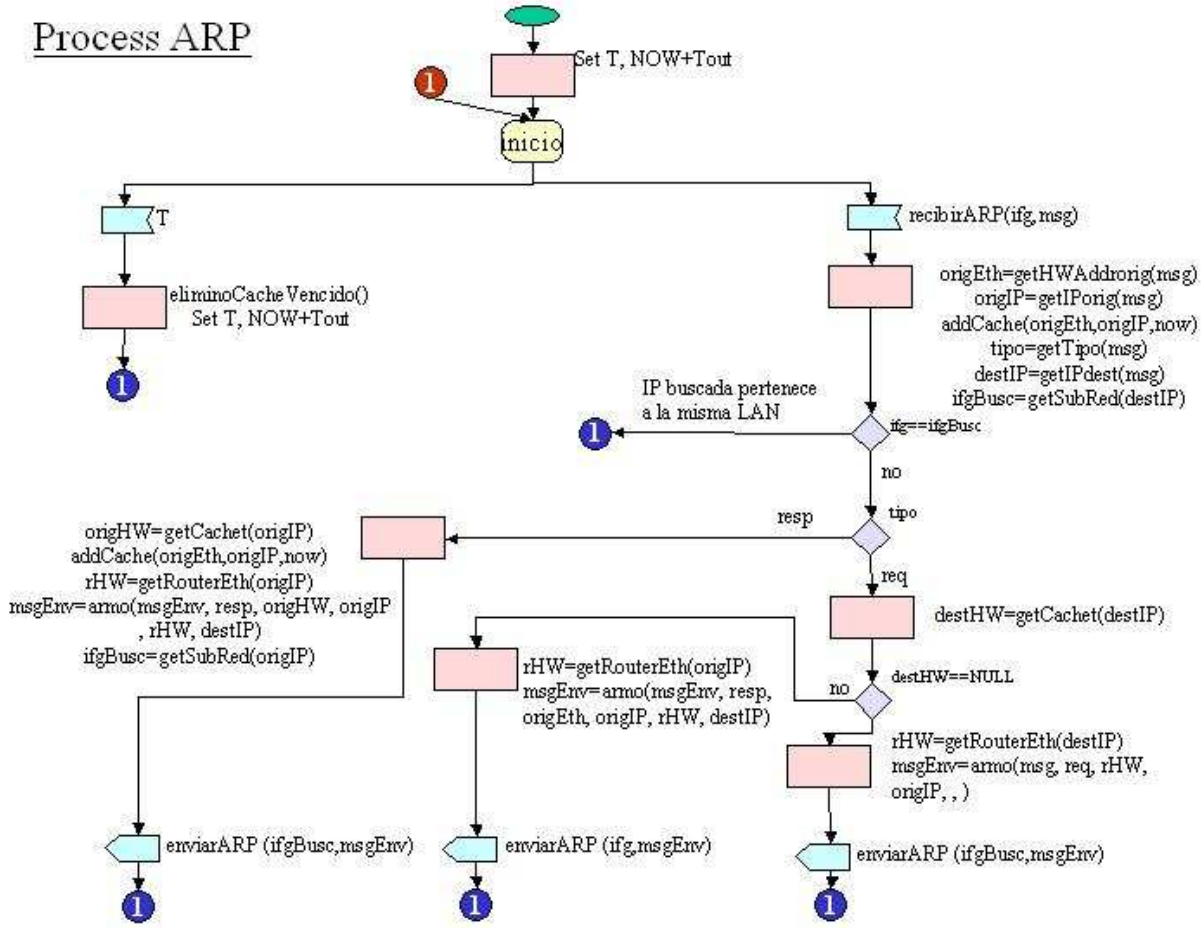
Solución:

Se debe implementar un procedimiento que corre en el router y funciona con la siguiente lógica:

- Dada una solicitud de dirección ethernet, para una IP que se encuentra en el mismo segmento, el router no contesta nada. De existir el equipo éste realizará la respuesta.
- Dada una solicitud de dirección ethernet, para una IP que se encuentra en otro segmento:
    - Verifico si tengo la información cacheada (para minimizar frames ARP), en caso de estar cacheada respondo indicando que la dirección ethernet del equipo es la correspondiente a la del router para la sub-red origen del equipo, en caso de no estar cacheada realizo la consulta en el segmento de red que corresponda, y al recibir la respuesta contesto modificando la dirección ethernet presentada por el equipo por la del router para la subred origen del equipo..
- Debo realizar para toda la información que pasa por el router un cache de la misma.

Resumiendo para las consultas siempre debe contestarse una dirección ethernet que se encuentre en el segmento, de lo contrario no existirá comunicación con el equipo.





Funciones utilizadas:

- getHWAddrorig(msg): EthAddress Devuelve la dirección ethernet SenderHA del mensaje ARP pasado por parámetro.
- getIPorig(msg): IPAddress Devuelve la dirección IP SenderIP del mensaje ARP pasado por parámetro.
- getTipo(msg):[req|resp] Devuelve el tipo de requerimiento solicitado en el mensaje ARP pasado por parámetro.
- getIPdest(msg): IPAddress Devuelve la dirección IP TargetIP del mensaje ARP pasado por parámetro.
- getIfgSubRed(IPAddress): interface Devuelve la interface del router correspondiente a la subred de la dirección IP pasada por parámetro.
- getCache(IPAddress): EthAddress Devuelve la dirección ethernet almacenada en el cache para la IP pasada por parámetro.
- getRouterEth(IPAddress): EthAddress Devuelve la dirección Ethernet del router correspondiente a la subred de la dirección IP pasada por parámetro.
- armo([req|resp], EthAddress, IPAddress, EthAddress, IPAddress): mensajeARP Devuelve un mensaje tipo ARP con los datos pasados por parámetro.
- addCache(EthAddress,IPAddress,timeStamp) Agrega al Cache los datos pasados por parámetro.
- eliminoCacheVencido () Elimina las instancias vencidas del cache.

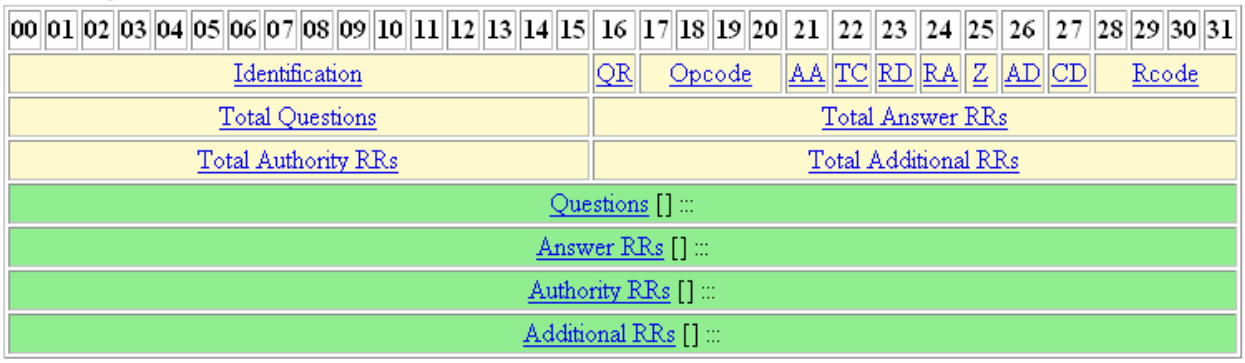
Problema 2 (30 puntos)

Un hacker desconforme con el programa argentino Intrusos esta intentando que todos los equipos de su trabajo no puedan acceder a la pagina del programa, www.intrusos.com.ar y sustituirla con su propia página modificada.

Suponga que el hacker puede ver todos los paquetes que circulan entre equipos de la red local, inclusive los que son solicitudes/respuestas del protocolo DNS (Domain Name System). La dirección IP del servidor DNS es conocida.

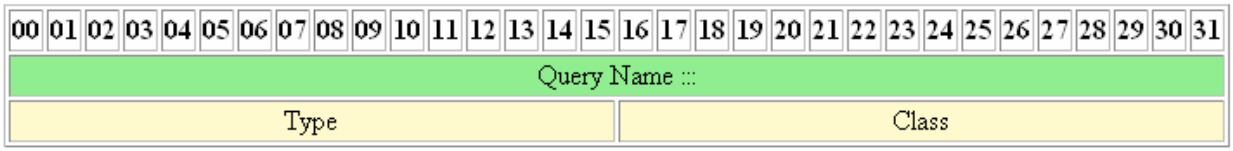
Los paquetes intercambiados por el protocolo DNS tienen el siguiente formato, y se incluye la información relevante del mismo para la resolución del problema.

Formato general:



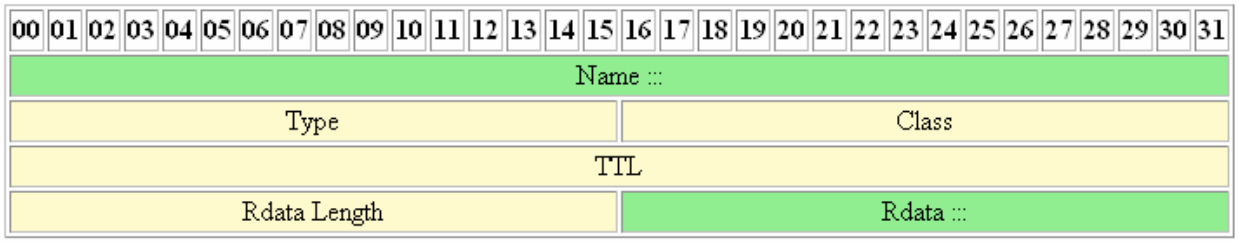
Identification → Identifica la solicitud en forma única.
QR → indica si corresponde a una pregunta o respuesta (Query/Response)
Rcode → Código de error de una solicitud procesada, 0 indica No error

Un paquete de consulta contiene un registro Question con la siguiente estructura



Query Name → contiene el dominio buscado (pude tener un largo variable)
Type → indica tipo de registro buscado, para el caso del ejercicio es A.

Un paquete de respuesta contiene un registro tipo Answer con la siguiente estructura:



Name → contiene el dominio respuesta (pude tener un largo variable)
Rdata Length → indica largo de la dirección respuesta (IP corresponde 4 bytes)
Rdata → contiene la dirección respuesta.

Se pide:

a) Describa e implemente un procedimiento que corre en la maquina del hacker conectada a la red que hace que todos los usuarios de la red accedan a su pagina, en lugar de la de Intrusos.

b) Proponga un método agregando elementos al protocolo DNS para que no se pueda realizar lo propuesto en a). Analice buscar la privacidad en la comunicación, e indique como se modifican cada una de las partes que intervienen (cliente/servidor)

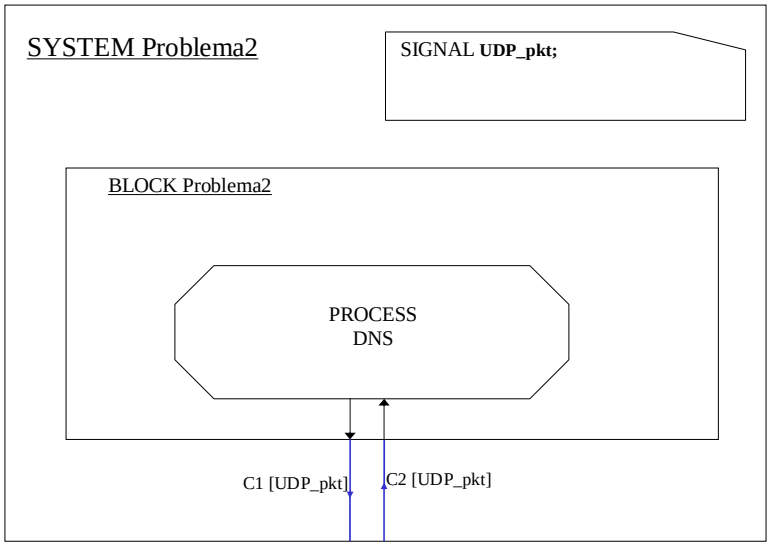
Nota:
Suponer que si un equipo recibe dos respuestas a una solicitud para resolución de un nombre de dominio, toma la primera e ignora la segunda.

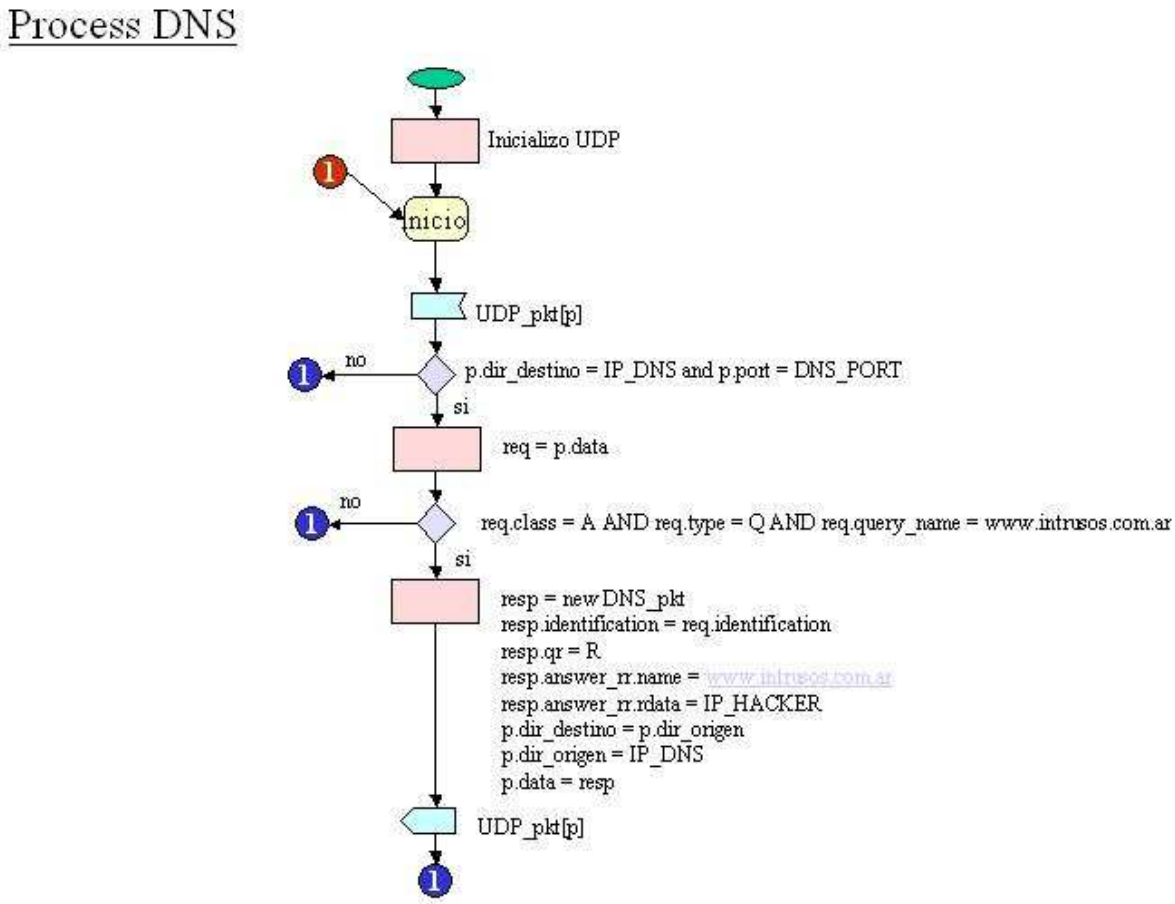
Solución:

a) Describa e implemente un procedimiento que corre en la maquina del hacker conectada a la red que hace que todos los usuarios de la red accedan a su pagina, en lugar de la de Intrusos.

El hacker ve los paquetes UDP de dns request de tipo Q, que en el campo query tengan la dirección de www.intrusos.com.ar, y guarda el número de identificación de dicho paquete. Inmediatamente devuelve la respuesta haciéndose pasar por el servidor DNS con dicho numero de identificación y el IP de su maquina que tiene una pagina falsa.
Se debe mandar los siguientes campos:
-IDENTIFICATION = ID (del paquete original)
-QR = R
-ANSWER_RR Name = www.intrusos.com.ar
-ANSWER_RR RData = IP_HACKER

Es de esperar que el paquete falso llegue antes que el verdadero al encontrarse la maquina del hacker en la misma red, y el host descartara el segundo paquete recibido.





b) Proponga un método agregando elementos al protocolo DNS para que no se pueda realizar lo propuesto en a). Analice buscar la privacidad en la comunicación, e indique como se modifican cada una de las partes que intervienen (cliente/servidor).

Una posibilidad es autenticar la respuesta del servidor de DNS. Esto se puede lograr por ejemplo con un esquema de clave pública.

El procedimiento podría ser que el servidor de DNS distribuye su clave pública, y al enviar una solicitud el host, el servidor de DNS “firma” el identificador de la solicitud con su clave privada, y no introduce en un nuevo campo destinado a la seguridad. El host al recibir la respuesta comprueba la firma del DNS con su clave pública previamente distribuida, aplicándolo al campo de seguridad y obteniendo el número el cual compara con el identificador original, si coinciden se puede asegurar que el DNS es el original.

Para lograr que la comunicación sea privada, se puede encriptar la información contenida en el campo de datos de las consultas y respuestas. Para esto se debe acordar una clave para poder encriptar los campos con un algoritmo de encriptación, por ejemplo 3DES.

La forma de acordar la clave privada es la siguiente: el host tiene la clave pública del servidor DNS por lo que genera una clave de 3DES la encripta con la clave pública del servidor DNS y la introduce en un campo especial de la consulta, a su vez encripta el campo de información con la clave de 3DES. El servidor de DNS al recibir el paquete desencripta la clave de 3DES con su clave privada, luego desencripta los datos de la consulta con la clave 3DES, resuelve la consulta y antes de enviar la respuesta, la encripta con la clave de 3DES.