**Práctica 11**

**Capa de Enlace - Parte II**

**1. Si la PC A está en una red y se quiere comunicar con la PC B que está en otra red:**

[**https://www.youtube.com/watch?v=DoQcznhxbhw**](https://www.youtube.com/watch?v=DoQcznhxbhw) **→ Importante**

* **¿Como se da cuenta la PC A de esto?**

Para esto, se usa el proceso ANDing:

Los hosts y [routers](http://redesdecomputadores.umh.es/red/routers/default.html) utilizan el proceso de ANDing para determinar si un host de destino está en la misma red. El proceso ANDing se produce cada vez que un host quiere enviar un paquete a otro host en una red IP. Para conectar con un servidor, el host debe conocer la dirección IP del servidor o el nombre del host ) por ejemplo [www.cisco.com](http://www.cisco.com/) [www.umh.es](http://www.umh.es/) ). Si el host utiliza el nombre de host, el servidor de denominación de dominio (que utiliza el protocolo [DNS](http://redesdecomputadores.umh.es/aplicacion/dns.html)) lo convierte en una dirección IP.

**Primero, el host de origen compara (AND) su propia dirección IP con su propia máscara de subred. El resultado de ANDing es la identificación de la red en la que reside el host de origen. Entonces compara las direcciones IP de destino con su propia máscara de subred. El resultado del segundo ANDing será la red en la que está el host de destino. Si la dirección de la red de origen y la dirección de la red de destino son iguales, se pueden comunicar directamente. Si los resultados son distintos, están en diferentes redes o subredes, y necesitarán comunicarse a través de routers, o puede que no sean capaces de comunicarse.**

El proceso ANDing depende de la máscara de subred. La máscara de subred siempre son únicas. Una máscara de subred predeterminada para una red de Clase C es 255.255.255.0 o 11111111.11111111.11111111.00000000 . El host compara esta máscara con la dirección IP de origen bit a bit. El primer bit de la dirección IP se compara con el primer bit de la máscara de subred, el segundo con el segundo, y así sucesivamente. Si los dos bits son unos, el resultado de ANDing es un 1. Si los bits son un cero y un uno o ambos son cero, el resultado de ANDing es un 0. Básicamente, esto significa que una combinación de dos unos da como resultado un 1; cualquier otra cosa es un 0. El resultado del proceso ANDing es el número de red o subred en que está la dirección de origen o de destino.

* **Si la tabla ARP de la PC A esta vacía, ¿que dirección MAC necesita la PC A para poder comunicarse con la PC B?**

Va a necesitar la dirección MAC del default gateway. Una vez obtenida, va a poder iniciar una comunicación hacia otra red, usando el protocolo ARP.

La idea es que la PC A va a enviar una pregunta a la PC B, de la cual conoce su dirección IP. La PC B responderá con su dirección MAC a la PC A. Luego de esto, la PC A podrá comunicarse con la PC B.

* **En base a lo anterior, ¿que dirección IP destino tiene el requerimiento ARP? ¿Es la dirección IP del default gateway o es la dirección IP de la PC B? De ser necesario, ejecute de nuevo el experimento de ser necesario y complete los campos:**

**Trama Ethernet: ( mac origen:** MAC\_PC\_A **mac destino:** ff : ff : ff : ff : ff : ff → broadcast **)**

**Solicitud ARP: (mac origen:** MAC\_PC\_A **ip origen:** IP\_PC\_A **)**

**(mac destino:** 00:00:00:00:00:00 (desconocida) **ip destino:** IP\_ROUTER\_QUE\_COMUNICA\_CON\_B**)**

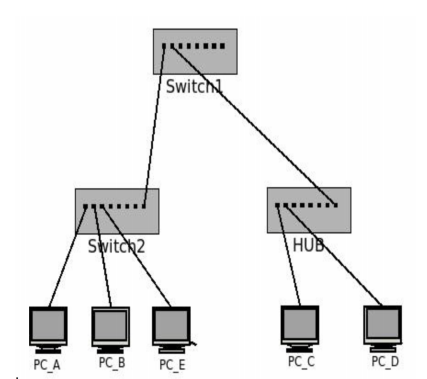
* **En base a lo anterior, indique la información de capa 2 y 3 del ICMP ECHO REQUEST que la PC A le envía a la PC B cuando ejecuta un ping, en el segmento de LAN de la PC B.**

La capa 2 es la capa de enlace. La capa 3 es la capa de red (se cuentan de abajo hacia arriba).

Trama Ethernet: (mac origen: MAC\_PC\_A mac destino: MAC\_GATEWAY )

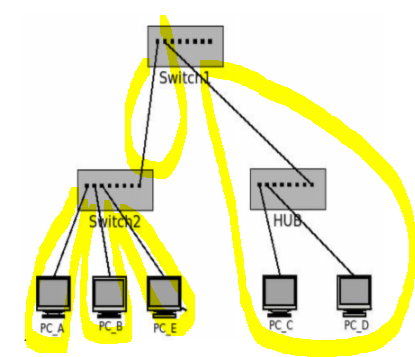
ICMP: (ip origen: IP\_PC\_A ip destino: IP\_PC\_B)

**2. En la siguiente topología de red indique:**

****

**a. ¿Cuántos dominios de colisión hay?**

Hay 5 dominios de colisión. Uno por cada boca del switch 1, y uno por cada boca del switch 2.



**b. ¿Cuántos dominios de broadcast hay?**

Solo uno. (no hay routers)

**c. Indique cómo se va llenando la tabla de asociaciones MAC ->PORT de los switches SW1 y SW2 durante el siguiente caso:**

[**https://capa8net.wordpress.com/2014/02/24/tabla-mac/**](https://capa8net.wordpress.com/2014/02/24/tabla-mac/)

**i. A envía una solicitud ARP consultando la MAC de C.**

Switch 1

|  |  |
| --- | --- |
| MAC | PUERTO |
| PC\_A\_MAC | 0 (CONEXIÓN CON PC\_A) |

Switch 2

|  |  |
| --- | --- |
| MAC | PUERTO |
| PC\_A\_MAC | 0 (CONEXIÓN CON PC\_A) |

**ii. C responde esta solicitud ARP.**

SW 1

|  |  |
| --- | --- |
| MAC | PUERTO |
| PC\_A\_MAC | 0 (CONEXIÓN CON PC\_A) |
| PC\_C\_MAC | 4 (CONEXIÓN CON PC\_C) |

SW 2

|  |  |
| --- | --- |
| MAC | PUERTO |
| PC\_A\_MAC | 0 (CONEXIÓN CON PC\_A) |
| PC\_C\_MAC | 4 (CONEXIÓN CON PC\_C) |

**iii. A envía una solicitud ARP consultando la MAC de B.**

SW 1

|  |  |
| --- | --- |
| MAC | PUERTO |
| PC\_A\_MAC | 0 (CONEXIÓN CON PC\_A) |
| PC\_C\_MAC | 4 (CONEXIÓN CON PC\_C) |

SW 2

|  |  |
| --- | --- |
| MAC | PUERTO |
| PC\_A\_MAC | 0 (CONEXIÓN CON PC\_A) |
| PC\_C\_MAC | 4 (CONEXIÓN CON PC\_C) |

**iv. B responde esta solicitud ARP.**

SW 1

|  |  |
| --- | --- |
| MAC | PUERTO |
| PC\_A\_MAC | 0 (CONEXIÓN CON PC\_A) |
| PC\_C\_MAC | 4 (CONEXIÓN CON PC\_C) |

SW 2

|  |  |
| --- | --- |
| MAC | PUERTO |
| PC\_A\_MAC | 0 (CONEXIÓN CON PC\_A) |
| PC\_C\_MAC | 4 (CONEXIÓN CON PC\_C) |
| PC\_B\_MAC | 2 (CONEXIÓN CON PC\_B) |

**d. Si la PC E y la PC D hubiesen estado realizando un tcpdump para escuchar todo lo que pasa por su interfaz de red, ¿cuáles de los requerimientos/respuestas anteriores hubiesen escuchado cada una?**

PC E:

* A envía una solicitud ARP consultando la MAC de C.
* A envía una solicitud ARP consultando la MAC de B.

**La razón es que las solicitudes ARP solicitando MAC son broadcast.**

PC D:

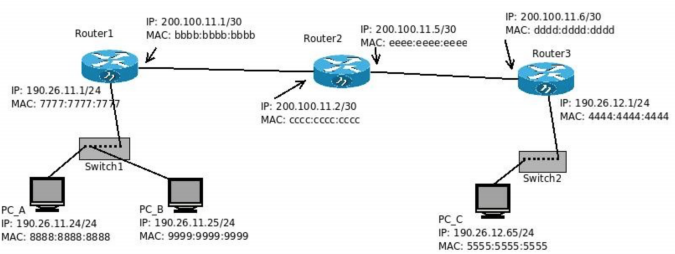
* A envía una solicitud ARP consultando la MAC de C.
* A envía una solicitud ARP consultando la MAC de B.

**La razón es que las solicitudes ARP solicitando MAC son broadcast.**

* C responde esta solicitud ARP.

**(Cuando C responde a A, deberia enterarse solo A, pero como C Y D estan ambos conectados a un HUB, este le manda la respuesta a todos los dispositivos conectados a el, el SWITCH 1, PC C Y PC D)**

**3. En la siguiente topología:**

****

**Suponiendo que todas las tablas ARP están vacías, tanto de PCs como de Routers. Si la PC\_A le hace un ping a la PC\_C, indique:**

* **¿En qué dominios de broadcast hay tráfico ARP?**

En todos.

* **¿En qué dominios de broadcast hay tráfico ICMP?**

En todos.

* **¿Cuál es la secuencia correcta en la que se suceden los anteriores?**

Primero los dispositivos tienen que aprender la MAC de los nodos a los que quieren enviar un paquete ICMP, así que una vez que identifican la dirección IP a la que tienen que enviar, comienzan el protocolo ARP para identificar la dirección MAC que se corresponde a la IP.

1. PC A, mediante su tabla de ruteo sabe que para enviar un datagrama ICMP a PC C primero debe dirigirlo a su default gateway 190.26.11.1/24 puesto que está en otra red. (router 1)

2. Como su tabla ARP está vacía, PC A averigua MAC de la interfaz 190.26.11.1/24 del router 1 por un ARP request.

3. El router 1 le contesta con un ARP reply (respuesta → 7777:7777:77:77).

4. PC A envía el paquete ICMP encapsulando el datagrama ICMP en una trama ethernet dirigida a 7777:7777:77:77, la dirección MAC recibida.

Este proceso se repite entre los nodos de las distintas redes hasta que llega a C y luego comienza el proceso otra vez hasta que le llegue la respuesta a PC A, posiblemente de manera más sencilla puesto que los dispositivos de las distintas LANs ya conocen las direcciones MAC de los otros dispositivos involucrados en la ruta. No vuelven a hacer ARP requests/replies.

DETALLE:

1- Como la PC A al aplicarle a la ip de la PC C su mascara de red se da cuenta que la PC C se encuentra en OTRA red, necesita comunicarse con la interfaz de salida del router al que esta conectado. Por eso, hace lo siguiente:

ARP (WHO-IS) PC\_A -> BROADCAST

Trama Ethernet: (mac origen: MAC\_PC\_A mac destino: BROADCAST )

Solicitud ARP: (mac origen: MAC\_PC\_A ip origen: IP\_PC\_A)

(mac destino: 00:00:00:00:00:00 ip destino: 190.26.11.1/24 → IP\_DEFAULTGATEWAY )

2- ARP (REPLY) ROUTER1 -> PC\_A

Trama Ethernet: (mac origen: MAC\_ROUTER1 mac destino: MAC\_PC\_A)

Solicitud ARP: (mac origen: MAC\_ROUTER1 ip origen: IP\_ROUTER1)

(mac destino: MAC\_PC\_A ip destino: IP\_PC\_A)

3- ICMP (echo request) PC\_A -> PC\_C (ROUTER1)

Trama Ethernet: (mac origen: MAC\_PC\_A mac destino: MAC\_ROUTER1)

◘ ICMP: (ip origen: IP\_PC\_A ip destino: IP\_PC\_C)

4- ARP (WHO-IS) ROUTER1 -> ROUTER2 (Default Gateway)

Trama Ethernet: (mac origen: MAC\_ROUTER1 mac destino: BROADCAST )

Solicitud ARP: (mac origen: MAC\_ROUTER1 ip origen: IP\_ROUTER1)

(mac destino: 00:00:00:00:00:00 ip destino: IP\_ROUTER2 )

5- ARP (REPLY) ROUTER2 -> ROUTER1

Trama Ethernet: (mac origen: MAC\_ROUTER2 mac destino: MAC\_ROUTER1)

Solicitud ARP: (mac origen: MAC\_ROUTER2 ip origen: IP\_ROUTER2)

(mac destino: MAC\_ROUTER1 ip destino: IP\_ROUTER1)

6- ICMP (echo request) PC\_A (ROUTER1) -> PC\_C (ROUTER2)

Trama Ethernet: (mac origen: MAC\_ROUTER1 mac destino: MAC\_ROUTER2)

◘ ICMP: (ip origen: IP\_PC\_A ip destino: IP\_PC\_C) → se mantiene

7- ARP (WHO-IS) ROUTER2 -> ROUTER3 (Default Gateway)

Trama Ethernet: (mac origen: MAC\_ROUTER2 mac destino: BROADCAST )

Solicitud ARP: (mac origen: MAC\_ROUTER2 ip origen: IP\_ROUTER2)

(mac destino: 00:00:00:00:00:00 ip destino: IP\_ROUTER3 )

8- ARP (REPLY) ROUTER3 -> ROUTER2

Trama Ethernet: (mac origen: MAC\_ROUTER3 mac destino: MAC\_ROUTER2)

Solicitud ARP: (mac origen: MAC\_ROUTER3 ip origen: IP\_ROUTER3)

(mac destino: MAC\_ROUTER2 ip destino: IP\_ROUTER2)

9- ICMP (echo request) PC\_A (ROUTER2) -> PC\_C (ROUTER3)

Trama Ethernet: (mac origen: MAC\_ROUTER2 mac destino: MAC\_ROUTER3)

◘ ICMP: (ip origen: IP\_PC\_A ip destino: IP\_PC\_C)

10- ARP (WHO-IS) ROUTER3 -> PC\_C

Trama Ethernet: (mac origen: MAC\_ROUTER3 mac destino: BROADCAST )

Solicitud ARP: (mac origen: MAC\_ROUTER3 ip origen: IP\_ROUTER3)

(mac destino: 00:00:00:00:00:00 ip destino: IP\_PC\_C )

11- ARP (REPLY) PC\_C -> ROUTER3

Trama Ethernet: (mac origen: MAC\_PC\_C mac destino: MAC\_ROUTER3)

Solicitud ARP: (mac origen: MAC\_PC\_C ip origen: IP\_PC\_C)

(mac destino: MAC\_ROUTER3 ip destino: IP\_ROUTER3)

12- ICMP (echo request) PC\_A (ROUTER3) -> PC\_C

Trama Ethernet: (mac origen: MAC\_ROUTER3 mac destino: MAC\_PC\_C)

◘ ICMP: (ip origen: IP\_PC\_A ip destino: IP\_PC\_C)

13- ICMP (echo reply) PC\_C -> PC\_A (ROUTER3)

Trama Ethernet: (mac origen: MAC\_PC\_C mac destino: MAC\_ROUTER3)

◘ ICMP: (ip origen: IP\_PC\_C ip destino: IP\_PC\_A) → comienza la respuesta

14- ICMP (echo reply) PC\_C (ROUTER3) -> PC\_A (ROUTER2)

Trama Ethernet: (mac origen: MAC\_ROUTER3 mac destino: MAC\_ROUTER2)

◘ ICMP: (ip origen: IP\_PC\_C ip destino: IP\_PC\_A)

15- ICMP (echo reply) PC\_C (ROUTER2) -> PC\_A (ROUTER1)

Trama Ethernet: (mac origen: MAC\_ROUTER2 mac destino: MAC\_ROUTER1)

◘ ICMP: (ip origen: IP\_PC\_C ip destino: IP\_PC\_A)

16- ICMP (echo reply) PC\_C (ROUTER1) -> PC\_A

Trama Ethernet: (mac origen: MAC\_ROUTER1 mac destino: MAC\_PC\_A)

◘ ICMP: (ip origen: IP\_PC\_C ip destino: IP\_PC\_A)

* **Para los paquetes ICMP que haya identificado:**
* **Especifique las direcciones (origen/destino) de capa 2 en los distintos dominios de broadcast.**

Se refiere a las direcciones MAC.

* **Especifique las direcciones (origen/destino) de capa 3 en los distintos dominios de broadcast.**

Se refiere a las direcciones IP.

Hecho arriba.

**4. Calcule los códigos de detección de error para las siguientes cadenas de bits utilizando paridad par y luego utilizando paridad impar:**

**a. 11010110101001111**

**b. 01011101011000010**

**c. 00100010001000111**

**Introducción:**

Definiciones:

Los **códigos de paridad** se usan en telecomunicaciones para detectar, y en algunos casos corregir, errores en la transmisión. Para ellos se añade en origen un bit extra llamado [*bit de paridad*](https://es.wikipedia.org/wiki/Bit_de_paridad) a los *n* [bits](https://es.wikipedia.org/wiki/Bit) que forman el carácter original.

Este valor del bit de paridad se determina de forma que el número total de bits *1* a transmitir sea par (código de paridad par) o impar (código de paridad impar).

Así, **para el código de paridad par, el número de unos contando el caracter original y el bit de paridad tiene que ser par.** Por lo tanto, el bit de paridad PAR será un 0 si el número total de unos a transmitir es par y un 1 para un número impar de unos.

Por el contrario, **para el código de paridad impar, el número de unos contando el caracter original y el bit de paridad ha de ser impar.** De esta forma, el bit de paridad IMPAR será un 0 si el número total de unos es impar y un 1 para un número par de unos.

Normalmente el bit de paridad se añade a la izquierda del caracter original.

**Ejemplos:**

Tenemos el caracter original **0111001**. Vemos que la trama a transmitir tiene un número par de unos (4 unos). Al añadir el bit de paridad obtendremos el siguiente caracter, que es el que se transmitirá a destino:

* Si usamos paridad **par**, ya hay un número par de unos, por tanto se añade un 0, y transmitiremos *0*0111001.
* Si usamos paridad **impar**, como hay un número par de unos, añadiremos otro 1 para conseguir un número impar, y transmitiremos *1*0111001.

Si se envía un dato y durante la transmisión se produce un único error, el destinatario puede detectarlo al comprobar la paridad en el destino. Usando los ejemplos anteriores, y alterando un solo bit de la trama transmitida, nos quedaría.

* Paridad par: se recibe 0011*0*001 en vez de 00111001. En la comprobación, al contar el número de unos salen 3 (impar), por lo que se ha producido un error.
* Paridad impar: se recibe 1011*0*001 en vez de 10111001. En la comprobación, al contar el número de unos salen 4 (par), por lo que se ha producido un error.

Siguiendo los ejemplos anteriores, y alterando dos bits en la transmisión, veremos como el método de detección de errores falla:

* Paridad par: se recibe 0011*01*01 en vez de 00111001. Al comprobar el número de unos salen 4 (par), y no se detectan los errores.
* Paridad impar: se recibe 1011*01*01 en vez de 10111001. Al comprobar el número de unos salen 5 (impar), y no se detectan los errores.

**Resolución del ejercicio:**

**a. 11010110101001111**

Paridad par:

Tiene once unos. Por ende, se debe añadir un 1 para lograr una cantidad de unos par.

**111010110101001111**

Paridad impar:

Tiene once unos. Por ende, se debe añadir un 0.

**011010110101001111**

**b. 01011101011000010**

Paridad par:

Tiene ocho unos. Se debe añadir un cero. **001011101011000010**

Paridad impar:

Tiene ocho unos. Se debe añadir un uno. **101011101011000010**

**c. 00100010001000111**

Paridad par:

Tiene seis unos. Se debe añadir un cero. **000100010001000111**

Paridad impar:

Tiene seis unos. Se debe añadir un uno. **100100010001000111**

**5. Se desea enviar la secuencia de bits 1100000111. Calcular la secuencia completa (datos + FCS) a transmitir considerando que el polinomio generador a utilizar es: G(x) = x^5 + x^4 + 1.**

Idea: enviar los datos originales más un FCS que me permita decir si los datos llegaron correctamente. Se supone que antes de la transferencia tanto el receptor como el emisor deben pactar cuál será el Generador, secuencia de bits que se usará para dividir el msj y comprobar.

Para enviar:

1. Se obtiene la longitud del FCS, será la longitud del Generador menos 1
2. Se le agregan tantos ceros como sea la longitud del FCS al final del msj a transmitir
3. Se divide esa nueva secuencia de bits con los ceros agregados POR el Generador, haciendo esta operación en módulo a 2, por lo que no interesan los carries al restar y se usa XOR en la resta (da 1 cuando los bits que se están operando son distintos, y 0 cuando son iguales)
4. Se obtiene el resto de esa división, el cual serà el FCS.
5. Se envían los datos originales reemplazando los ceros concatenados anteriormente por el FCS.

Para verificar la info recibida:

1. Como se que me llegó un msj mas un FCS, divido todo eso por el Generador y la división me debe dejar un resto = a 0. Si no es así, hubo error.

**NOTAS:**

* FCS → Frame Check Sequence, bits that you append to the original data (R, compuesto por r bits, siempre r = 1 bit menos que los del generador)
* BUENA EXPLICACION:[**https://www.youtube.com/watch?v=Fz6b1t8sr10**](https://www.youtube.com/watch?v=Fz6b1t8sr10)

**Resolución del ejercicio:**

G(x) = x^5 + x^4 + 1.

Mensaje = 1100000111

Parte 1:

Primero, convierto el polinomio a binario.

x^5 + x^4 + 1 → 1x^5 + 1x^4 + 0x^3 + 0x^2 + 0x^1 + 1 → 110001

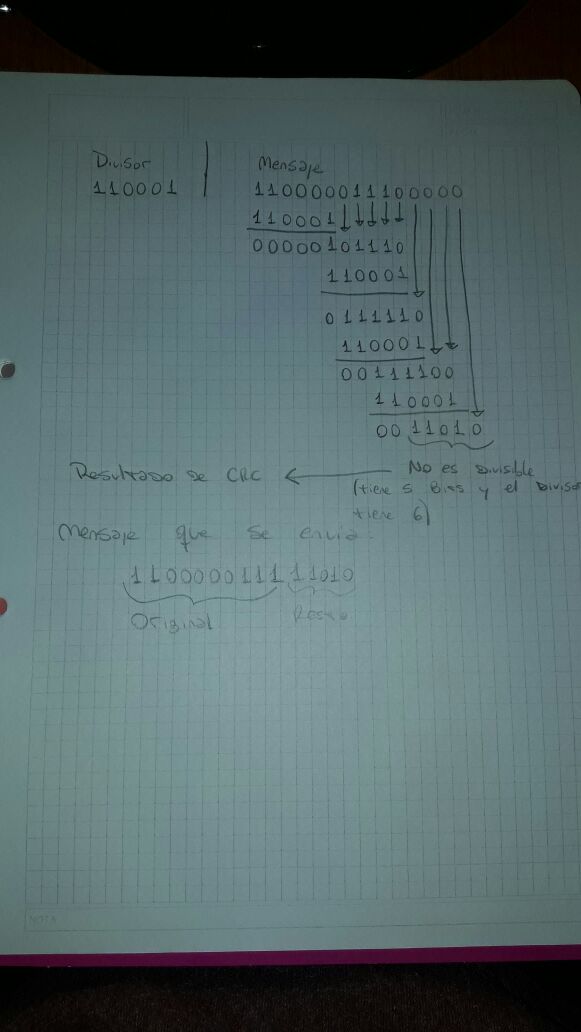
Ahora modifico el mensaje original:

La cantidad de bits a añadir al mensaje original debe ser un dígito menor al divisor.

Mensaje = M (cantidad de bits de mensaje) + 5 bits

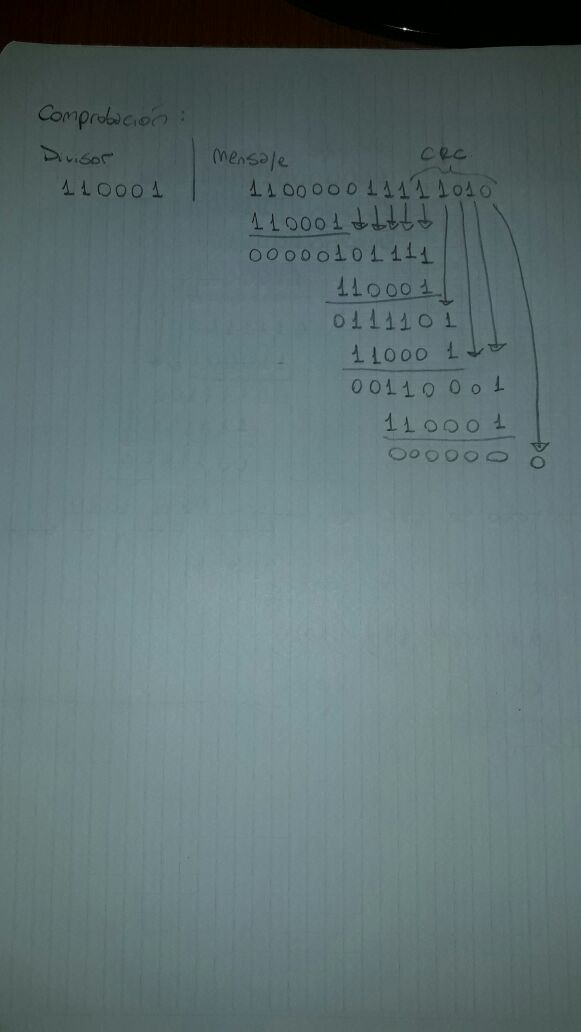
110000011100000

Parte 2: hacer división binaria entre el mensaje y el divisor con el propósito de enviar al receptor el mensaje (M) y R siendo R el resto de la división binaria.



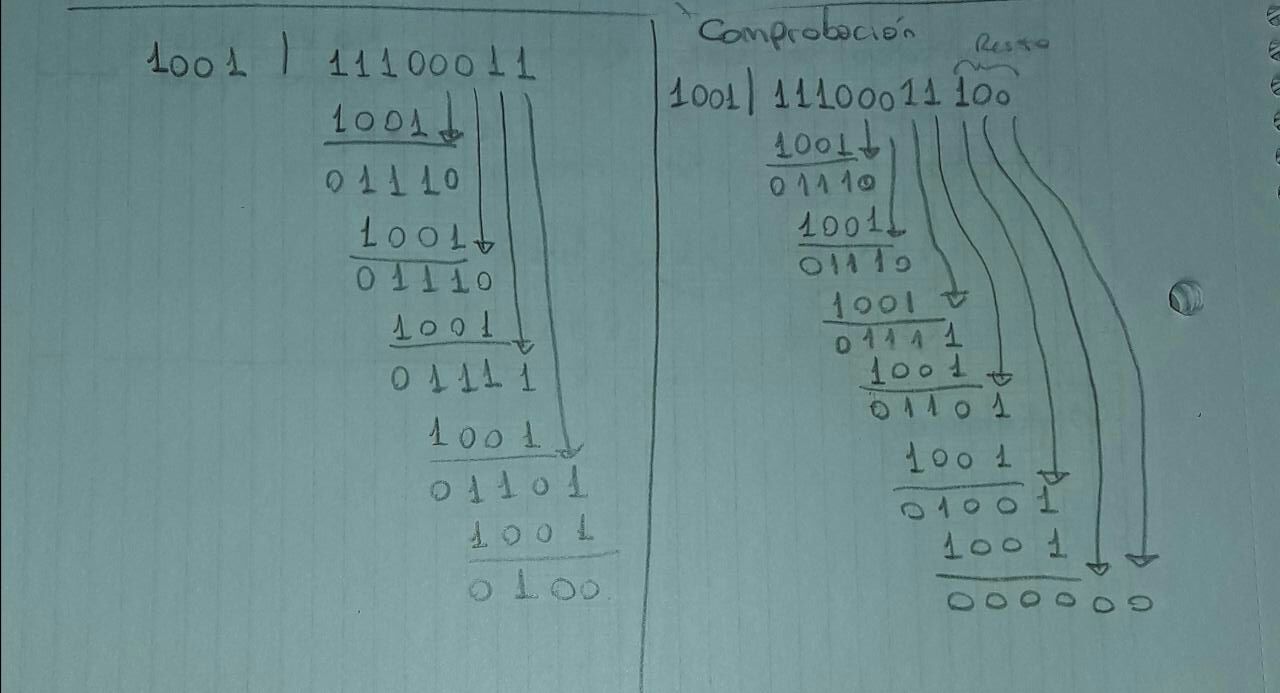
Comprobación:

En este punto es importante que el receptor conozca el divisor y el CRC.



Si el resultado es 0 significa que no hubo errores. Si es distinto de 0, significa que hubo error (se necesita retransmisión de datos).

**6. Encontrar el FCS si se utiliza la función generadora G=1001 y el mensaje M=11100011.**



**7. Indicar si es verdadero o falso. Justifique su respuesta.**

a. Si se utiliza paridad par y se invierte el valor de 2 bits a causa de errores en la transmisión, el receptor detectará el error.

b. 00101011 es un valor válido para ser usado como polinomio generador y el resto sería de 7 bits de longitud. **- El polinomio generador sería 101011 y el resto sería de 5 bits de longitud (1 menos que el polinomio)**

c. Los FCS calculados con el polinomio generador 11001 tendrán una longitud de 4 bits. **- Siempre tienen la cantidad de bits del polinomio generador - 1.**

**8. ¿Existe ARP en IPv6? ¿Por qué? ¿Quién cumple esa función?**

No.

La difusión del Protocolo de Resolución de Dirección (Address Resolution Protocol, ARP) de IPv4 afecta la eficiencia de la red. Esta situación no ha sido incluida en IPv6, y en su lugar se utiliza el Multicasting el cual funciona de la siguiente manera:

Se crea un grupo Multicast, formado por conjunto de interfaces de red.

Si se está interesado en que cierta computadora reciba los paquetes de difusión del grupo se agrega una interfaz de red, de esa forma se envía un paquete multicast al grupo X.

Ese paquete sólo llegará a aquellas computadoras que tengan su interfaz incluida en el grupo multicast X. Con ello se permite tener niveles de eficiencia de red superiores a los presentados en IPv4, lo cual se verá traducido en la disminución de los ciclos de procesamiento de CPU de las computadoras en la red local al no procesar paquetes de difusión que no van dirigidos a ellos y de la misma manera se estará eliminando el problema de las tormentas de paquetes de difusión de IPv4.

El funcionamiento del protocolo ND de IPv6 equivale a combinar los siguientes protocolos de IPv4: ARP (Address Resolution Protocol), ICMP (Internet Control Message Protocol, Router Discovery e ICMP Redirect. IPv4 carece de un protocolo general establecido y de un mecanismo para detectar la *‘alcanzabilidad’* de vecinos.

**Neighbor Discovery** (ND) es un [protocolo](https://es.wikipedia.org/wiki/Protocolo_de_red) de [IPv6](https://es.wikipedia.org/wiki/IPv6), y es equivalente al protocolo [Address Resolution Protocol (ARP)](https://es.wikipedia.org/wiki/Address_Resolution_Protocol) en [IPv4](https://es.wikipedia.org/wiki/IPv4), aunque se distingue porque también incorpora funcionalidades de [ICMP](https://es.wikipedia.org/wiki/Internet_Control_Message_Protocol).

Utiliza mensajes especiales de [ICMPv6](https://es.wikipedia.org/wiki/ICMPv6) construyendo así una manera simple para que los terminales aprendan las direcciones [IPv6](https://es.wikipedia.org/wiki/IPv6) de los vecinos de la capa de enlace. Consiste en un mecanismo con el cual un nodo que se acaba de conectar a la red, descubre la presencia de otros nodos en el mismo enlace, además de ver sus direcciones IP. Otra de las grandes funcionalidades de este protocolo es que se ocupa de mantener limpios los cachés dónde se almacena la información relativa al contexto de la red a la que está conectado un nodo. Así, cuando una ruta hacia cierto nodo falla, el router correspondiente buscará rutas alternativas. Basándose también en los mensajes ICMPv6 se permite un mecanismo de auto-configuración.

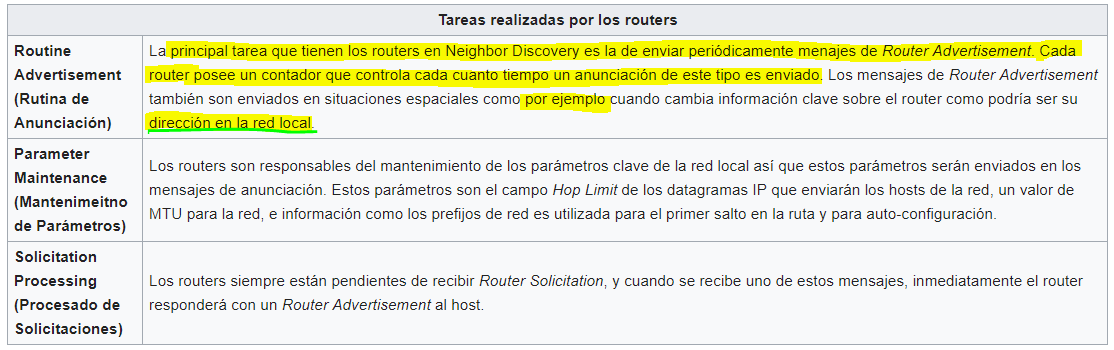
Mapea direcciones lógicas (IPv6) a direcciones de Hardware (MAC, EUI-48, EUI-64).

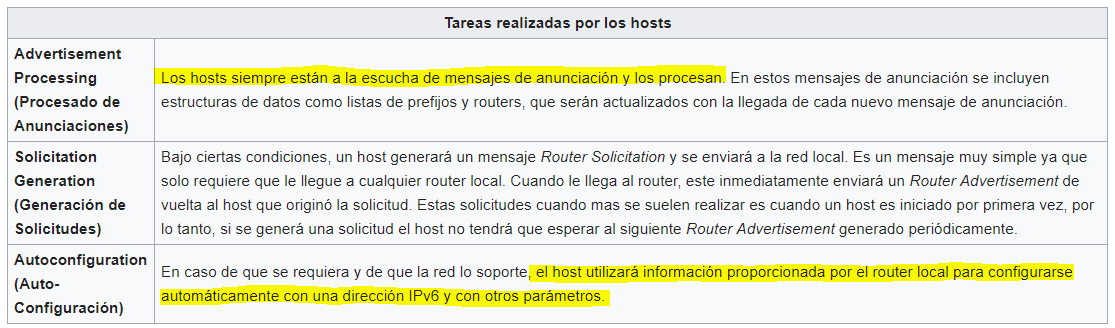
## Funcionalidades de descubrimiento Host-Router

Este es uno de los principales grupos de funcionalidades de Neighbor Discovery ya que facilitan el descubrimiento de routers locales e intercambian información entre ellos y los hosts. Estas funcionalidades se distinguen en cuatro específicas y están diseñadas para su uso en los mensajes *Router Solicitation* y *Router Advertisement*:

* **Router Discovery** (Descubrimiento de Router): es el método por el cual los hosts localizan routers en la red local.
* **Prefix Discovery** (Descubrimiento de Prefijo): los hosts utilizan esta función para determinar en qué red se encuentran. A su vez esta función indica como diferenciar entre destinos locales o distantes a la red local y si intentar la entrega directa o indirecta de datagramas.
* **Parameter Discovery** (Descubrimiento de Parámetros): es el método por el cual un host aprende parámetros importantes sobre la red local y/o routers, como por ejemplo la MTU del enlace local.
* **Address Autoconfiguration** (Auto-configuración de dirección): los hosts en IPv6 son capaces de configurarse automáticamente, pero esto requiere información que es proporcionada por un router.

Para proporcionar todas estas funcionalidades es necesario que los routers y los hosts realicen una serie de tareas:





## Funcionalidades de descubrimiento Host-Host

Este es otro de los principales grupos de funcionalidades de Neighbor Discovery está asociado con información determinante y la comunicación entre nodos (normalmente hosts). Algunas de las siguientes funciones también puede ser utilizada entre un router y un host, pero este grupo no fue específicamente diseñado para ello. Estas funcionalidades se distinguen en cuatro específicas:

* **Address Resolution** (Resolución de dirección): es el proceso por el cual un dispositivo obtiene la dirección de capa 2 (MAC) de otro dispositivo de la red local a partir de su dirección de capa 3 (IPv6) Para realizar este proceso de resolución el dispositivo que desea saber la MAC del destino debe enviar un mensaje Neighbor Solicitation con la dirección IP de destino. El destino responderá con un mensaje Neighbor Advertisement que contendrá su dirección MAC. Esta resolución de direcciones puede ser utilizada para hallar la dirección MAC de un host o de un router.
* **Next-Hop Determination** (Determinación del siguiente salto): es el método por el cual se consulta el destino del datagrama IP y así se determina dónde se debe enviar a continuación.
* **Neighbor Unreachability Detection** (Detección de vecino inalcanzable): es el proceso que determina si un vecino puede ser o no contactado.

Los mensajes Neighbor Solicitation y Neighbor Advertisement a menudo son asociados solamente con a la resolución de direcciones, pero tienen otros propósitos también. Uno de estos propósitos es la detección de vecinos inalcanzables. Cada dispositivo tiene información sobre sus vecinos y la actualiza dinámicamente según las condiciones de la red van cambiando. Esta información es guardada ya sea para un host o para un router de la red.

El hecho de saber cuando un dispositivo ya no es alcanzable es importante ya que un host deberá adaptar su comportamiento de acuerdo a esto. En caso de un host inalcanzable, el dispositivo debería esperar cierto periodo de tiempo antes de intentar enviar datagramas al host inalcanzable, evitando así inundar la red con repetidos intentos de envío. En cambio, el que un router sea inalcanzable indica que el dispositivo tiene que buscar un nuevo router para utilizar (si existe alguna alternativa).

Cada host tiene una caché de vecinos dónde guarda información sobre los dispositivos vecinos. Cada vez que un host recibe un datagrama de un vecino, sabrá que el vecino es alcanzable en ese momento, creándose así una entrada en la caché para indicarlo. Claro que cuanto más tiempo pase desde que se ha recibido un datagrama de un vecino, mayor es la posibilidad de que algo ocurrido para que el vecino ya no sea alcanzable. Por esta razón la información de caché debe ser considerada temporal. Cada vez que un vecino es anotado en la caché como alcanzable, se inicia un contador. Cuando el tiempo termina, la alcanzabilidad de cierto vecino es considerada antigua, y por lo tanto no se asume que el vecino siga estando disponible. Cuando se recibe de nuevo un datagrama del vecino cuya información ha caducado, el contador es reiniciado indicando así que el vecino es alcanzable de nuevo. El tiempo que un vecino debería ser considerado alcanzable es comunicado en los mensajes Router Advertisement.

Es posible que un host necesite saber si un vecino sigue estando alcanzable, para esto se mandará un Neighbor Solicitation esperando un Neighbor Advertisement, actualizándose la información en la caché.

* **Duplicate Address Detection** (Detección de direcciones duplicadas): determina si la dirección que un dispositivo desea utilizar ya existe en la red.

Cuando un host utiliza el proceso automático de configuración IPv6 uno de los pasos a realizar es asegurarse de que la dirección que se a a utilizar ya existe en la red. Esto se hace enviando un mensaje Neighbor Solicitation a la dirección que se pretende comprobar. SI se recibe un mensaje Neighbor Advertisement eso significa que la dirección ya se está usando.

**NOTA:**

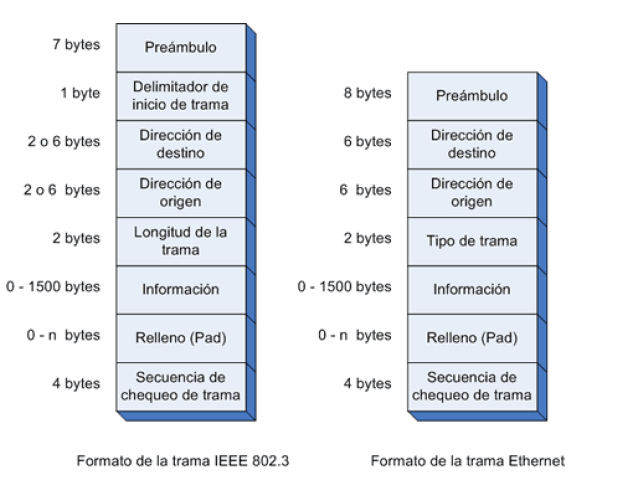
Updating Neighbors Using Neighbor Advertisement Messages

* Los mensajes de Neighbor Advertisement no son enviados periódicamente como los de Router Advertisement. Esto no es ninguna necesidad ya que los vecinos no cambian mucho con el paso del tiempo, y de esta manera las resoluciones ocurrirán de manera natural cada vez que los dispositivos se envíen datagramas entre sí. Además, sería poco eficiente para la red que se enviasen periódicamente anunciaciones por cada host.
* De cualquier manera, un host puede enviar un mensaje Neighbor Advertisement no solicitado cuando ve necesario actualizar la información de los vecinos en la red local. Por ejemplo cuando se reemplaza la tarjeta de red de un equipo (cambiará su dirección MAC), se podrá enviar un mensaje Neighbor Advertisement para que los vecinos actualicen la dirección MAC en su caché.

**9. ¿Qué es la IEEE 802.3? ¿Existen diferencias con Ethernet?**

Ethernet es una implementación que se convirtió en estándar de facto para LANs con acceso al medio tipo CSMA/CD, definiendo las características de cableado y señalización de nivel físico y los formatos de tramas de datos del nivel de enlace de datos del modelo OSI. Las primeras versiones surgieron en los '70. **Ethernet se tomó como base, en el '83, para la redacción del estándar formal IEEE 802.3, manejándose usualmente como sinónimos, aunque se diferencian en uno de los campos de la trama de datos.**

Las tecnologías más comunes de tipo Ethernet están agrupadas en el estándar 802.3 (10Base2, 10BaseT, 100BaseT, Gigabit Ethernet...). Se suele hacer referencia a Ethernet como una LAN 802.3.



**La diferencia más significativa entre la tecnología Ethernet original y el estándar IEEE 802.3 es la diferencia entre los formatos de sus tramas. Esta diferencia es lo suficientemente significativa como para hacer a las dos versiones incompatibles.**

Una de las diferencias entre el formato de las dos tramas está en el preámbulo. El propósito del preámbulo es anunciar la trama y permitir a todos los receptores en la red sincronizarse a sí mismos a la trama entrante. El preámbulo en Ethernet tiene una longitud de 8 bytes pero en IEEE 802.3 la longitud del mismo es de 7 bytes, en este último el octavo byte se convierte en el comienzo del delimitador de la trama.

La segunda diferencia entre el formato de las tramas es en el campo tipo de trama que se encuentra en la trama Ethernet. Un campo tipo es usado para especificar al protocolo que es transportado en la trama (IP, IPX...). Esto posibilita que muchos protocolos puedan ser transportados en la trama. El campo tipo fue reemplazado en el estándar IEEE 802.3 por un campo longitud de trama, el cual es utilizado para indicar el número de bytes que se encuentran en el campo da datos (puede diferenciarse una trama Ethernet de una 802.3 porque tipo toma valores superiores a 1500 y longitud valores menores o iguales).

La tercera diferencia entre los formatos de ambas tramas se encuentra en los campos de dirección, tanto de destino como de origen. Mientras que el formato de IEEE 802.3 permite el uso tanto de direcciones de 2 como de 6 bytes, el estándar Ethernet permite solo direcciones de 6 Bytes.

**10. Nombre cinco protocolos de capa de enlace. ¿Todos los protocolos en esta capa proveen los mismos servicios?**

- Ethernet

2,85 Mbit/s sobre cable coaxial en topología de bus.

- IEEE 802.3

10BASE5 10 Mbit/s sobre coaxial grueso (thicknet). Longitud máxima del segmento 500 metros - Igual que DIX salvo que el campo de Tipo se substituye por la longitud.

- Redes LAN Inalámbricas 802.11 (WiFi)

- Token ring

- PPP (Point-to-Point Protocol)

- Bluetooth en el 802.15 (IEEE 802.15).

Los servicios proporcionados por los diferentes protocolos de la capa de enlace a lo largo de la ruta terminal a terminal pueden ser distintos. Ej. Algunos protocolos proporcionan una entrega fiable mientras que otros no (los que son más propenso a errores, por la atenuación de sus señales o interferencias electromagnéticas, suelen tener este servicio).