

Parcial2

Federico Polidoro

Contents

1	Clase 9 - Capa de enlace de datos	1
2	Clase 10 - Capa de enlace de datos - 2	4
3	Clase 11 - Protocolos Acceso aleatorio	5
4	Clase 12 - Ethernet (creado por el sr macalfe)	7
5	Clase 13 - Dispositivos capa 2	7

1 Clase 9 - Capa de enlace de datos

Esta capa provee los servicios:

- Armar las tramas.
- Controlar Errores
- Controlar el flujo

1.1 Armar Tramas

Esta junta los bits enviados en la capa 1 y los agrupa en “tramas”.

1.2 Control Errores

En esta capa se detecta si hubieron errores de transmision con las tramas y si son correjibles.

1.3 Control Flujo

Es como un manejo de velocidad entre las maquinas, Esto para que en caso de que las maquinas a las que nos conectamos sean muy rapidas o hayan muchas nuestra maquina no se sature al intentar procesar una cantidad enorme de datos. El control de flujo pide que bajen un cambio.

1.4 Estructura

Los paquetes de capa 3 se encapsulan con un header y un trailer en la capa 2 (tramas = header + payload + trailer).

1.5 Camino real

Cada capa solo puede comunicarse con su equivalente en el receptor de la comunicacion. pero no quiere decir que el camino de la informacion sea desde capa 2 a capa 2 sino que se translada desendiendo de capa en capa hasta la fisica (1) donde la informacion pasa el emisor al receptor.

1.6 Entramado

- Cuenta de caracteres,

Se dividen las tramas por un numero fijo de caracteres, se indica a la capa 2 donde inicia y termina cada trama dedicando el primer caracter de la trama para indicar la longitud de la misma (contando ese char).

Tiene la desventaja de que si el caracter de longitud se corrompe se modifica el tamaño de la trama.

- Agregado Chars (char stuffing),

Se utilizan flags para definir Donde empieza una trama y donde termina. es como los wakie talkie donde se avisa que uno va a empezar a hablar y que va a dejar de hablar.

Un flag-char es 0b **01111110**.

Pero que pasa si en el payload de la trama hay una secuencia identica al **flag**, para eso se añaden caracteres de escape antes de los caracteres que se puedan confundir con un comando:

$A + \text{FLAG} + B \Rightarrow A + \text{ESC} + \text{FLAG} + B(1)$

$A + \text{ESC} + B \Rightarrow A + \text{ESC} + \text{ESC} + B(2)$

$A + \text{ESC} + \text{FLAG} + B \Rightarrow A + \text{ESC} + \text{ESC} + \text{ESC} + \text{FLAG} + B(3)$

1.7 Bit Stuffing

Se ingresan bits con valor 0 dentro la trama de forma en la cual no se pueda confundir con un caracter flag, por lo que, cada cinco '1' se pone un 0.

1.8 Deteccion de Errores

Se transmiten los datos seguido del resultado de pasar los datos por una funcion tipo hash. si el receptor comprueba los datos recibidos pasandolos por la misma funcion hash y el resultado que tiene es identico al hash recibido quiere decir que los datos fueron recibidos de forma correcta.

1.9 Control de paridad

Se obliga a que la cantidad de '1' transmitidos sea par. en caso de no cumplirse es que los datos estan corruptos. Aunque... pueden haber 2 fallos y la suma de '1's daria como si no lo hubiese. solo puede detectar cantidades impares de errores.

1.10 Hamming

Es un codigo que permite la correccion de error.

Consiste en añadir una redundancia a un mensaje, Permite corregir hasta un 1 bit, en caso de haber mas de uno no puede corregirlo pero si detectar que hay errores.

La cantidad de redundancia necesaria es descripta como:

$$2^r \geq m + r + 1$$

Aplicacion

Se elije como redundancia los bits potencia de 2 (1,2,4,8...). Para el mensaje 10110 m=5 con 4 bits de redundancia seria

*	*	1	*	0	1	1	*	0
1	2	3	4	5	6	7	8	9

ahora hay que completar los espacios en blanco (*) con bits que cumplan la paridad

0	1	1	0	0	1	1	0	0
1	2	3	4	5	6	7	8	9

1.11 CRC

Es un código que permite la detección de errores.

Verifica antes de transmitir una palabra que esta tenga resto 0, al dividirla por un polinomio generador $G(x)$.

Una desventaja es que todo lo que enviamos tiene que tener un resto cero y en caso de no ser así necesitamos transformarlo para que sea así.

- También se puede dar como polinomio.

2 Clase 10 - Capa de enlace de datos - 2

La trama dentro de la capa de enlace de datos tiene esta estructura (Syn + Addr + Ctrl + DATA + CRC).

2.1 ARQ

Automatic Repeat ReQuest

En caso de que la trama llegue correctamente se responde con un **ACK** y se pasa a la siguiente trama. En caso contrario se devuelve un **NACK** y la trama debe ser reenviada.

2.2 Parar y esperar

El emisor espera recibir un ACK/NACK para enviar la siguiente trama. Es muy ineficiente en tiempo.

2.3 Regresar a N

El transmisor no espera al recibir el reconocimiento antes de transmitir otra trama.

En caso de que no se reciba un ACK/NACK la trama se retransmite igualmente.

3 Clase 11 - Protocolos Acceso aleatorio

En los enlaces de difucion el canal de transmision es compartido entre varios nodos.

Cuando hay tramas que enviar, se transmite usando todo el ancho de banda del canal, no hay coordinación inter-nodo y si hay 2 o más nodos transmitiendo en paralelo se crea una colisión.

Los protocolos MAC de acceso aleatorio especifican:

- Como detectar colisiones.
- Como recuperarse de una colisión

Ejemplos de protocolos MAC:

- Aloha
- Aloha Ranurado
- CSMA, CSMA/CD, CSMA/CA.

3.1 Aloha

Es un protocolo desarrollado en Hawaii, circa 1970.

Con el objetivo de comunicar varias islas del archipielago con un nodo central, todos los nodos perifericos usan la misma frecuencia de transmision. por lo que los envios pueden colisionar si 2 nodos transmiten al mismo tiempo.

Pasos:

1. Transmitir Tramas
2. Esperar ACK - fin
3. else Esperar tiempo y volver al paso 1.

Colisiones

- Los nodos envian sus tramas sin tener en cuenta las transmisiones en curso.
- Una trama enviada en t_0 puede colisionar con otra trama enviada en t_0-1 o t_0+1 .
- La eficiencia del protocolo es $\approx 18\%$.

3.2 Aloha Ranurado

- Todas las tramas tienen el mismo tamaño.
- El tiempo se divide en intervalos de longitud fija.
- Las transmisiones solo pueden comenzar al principio de un intervalo.
- Eficiencia $\approx 37\%$.

3.3 CSMA

Este escucha a la señal portadora antes de transmitir:

- Canal libre: transmite la trama completa.
- Si detecta el canal ocupado: retrasa al transmision.

“Analogia humana a no interrumpir cuando alguien habla.”
~Nadie nunca

Aun asi pueden generarse repeticiones en redes muy grandes por culpa del retardo de propagación de la señal si dos nodos comienzan a transmitir casi simultaneamente.

3.4 CSMA/CD

Similar a CSMA pero compueba el canal mientras transmite -> detecta colision.

Usado en Ethernet.

- Las colisiones son detectadas rápidamente. Si se detecta una colisión se aborta la transmisión, reduciendo el mal uso del canal.

Detección de colisiones

Fácil en LAN cableadas: se mide la potencia de la señal, comparando señales transmitidas y recibidas.

Difícil en LAN inalámbricas: receptor apagado mientras se transmite.

4 Clase 12 - Ethernet (creado por el sr macalfe)

Es un protocolo de comunicación utilizado ampliamente para comunicar distintos dispositivos dentro de una sola red de área local, como por ejemplo una oficina, una casa o edificio.

IEEE define a ethernet como el protocolo 802.3

4.1 Historia

Entre 1973 y 1974, ethernet fue desarrollado en los Xerox labs, Palo Alto, CA.

4.2 Datos

- Los coaxiales BNC casi no se usan más.
- CAT5e el más común actualmente.
- CAT6 usado en redes fast ethernet.
- CAT8 el más rápido actualmente.
- Half duplex, solo envía o recibe. Full duplex, envía y recibe al mismo tiempo.
- Fibra óptica, puede ser de vidrio o plástico, bastante más rápido y con más alcance que un coaxial.

5 Clase 13 - Dispositivos capa 2

5.1 Switch

El switch solo te comunica con el dispositivo al que estás queriendo comunicarte distinguiendo por direcciones físicas. Se puede tener varias comunicaciones en paralelo.

Voip

Se puede dar energia a telefonos a travez de la ficha de telefono saliente de la ficha.

5.2 Hub

No distinguen entre direcciones fisicas y envian los paquetes que reciben a todos los dispositivos en la red. además los dispositivos que quieren enviar paquetes deben parar hasta que el hub no este enviando nada para poder comunicarse con él.