

Proyecto 2 Capa de Enlace

Para esta 2da etapa del proyecto de redes incorporaremos nuevos dispositivos y conceptos relacionados con la capa de enlace.

Elementos a implementar

1. **Cable Duplex.** A partir de ahora, los **cables** tienen la capacidad de transmitir información en dos sentidos. Los **hosts** pueden estar leyendo y transmitiendo a la misma vez. Piénsenlo de la siguiente forma: ahora el **cable** físico, tiene internamente varios hilos/pelos por los que viaja información, similar a los cables de corriente que tienen un vivo y un neutro. En este caso hay uno que transmite “hacia una dirección” y el otro que transmite “en la otra dirección”.
 - Puede seguir habiendo colisiones si varios **hosts** transmiten simultáneamente.
 - Se debe garantizar que si hay solamente 2 **hosts** transmitiendo en el mismo **cable/hub** que no haya colisiones porque cada uno de los **hosts** transmitirá lo suyo y leerá lo del otro. Pero en el momento que aparezca un 3ro transmitiendo deben empezar a ocurrir las colisiones.
2. **Dirección MAC.** Cada **computadora** a partir de ahora, tiene una dirección **MAC** única. Esta dirección **MAC** está compuesta por 2 bytes (16 bits)
3. **Trama (frame).** Cuando una **computadora** quiera transmitir información, la transmitirá en forma de una **trama** que tiene la siguiente estructura:
 - Primeros 16 bits: dirección **MAC** a la que está dirigida la **trama**. Si el valor de la dirección **MAC** tiene 1 en todos sus bits (valor entero = 65535) significa que está dirigida a todo el mundo (*broadcast*).
 - Próximos 16 bits: dirección **MAC** del **host** que transmite.
 - Próximos 8 bits: tamaño de los datos (en bytes). O sea si el byte contiene un 00001010 lo cual es un 10 en decimal, eso significa que a continuación vienen 10 bytes (80 bits) de información.
 - Próximos 8 bits: campo extra, que por el momento será 00000000.
 - A continuación vienen los datos (el tamaño especificado en el 3er campo).
4. **Switch (conmutador).** El **switch** es un dispositivo de conexión de redes, que tiene varios puertos (similar al **hub**) pero con una importante diferencia. El **switch** contiene una tabla de las direcciones **MAC** de los dispositivos conectados a él y utiliza esta información para solo transmitir por el **puerto** al que está conectado el **host** receptor. De forma que un **switch** de 4 puertos puede tener 4 **hosts** conectados, por ejemplo: **pc1** y **pc2** comunicandose entre ellos y **pc3** y **pc4** comunicandose entre ellos, sin que ocurran colisiones.

- Al iniciar la red el *switch* no tiene información sobre las direcciones *MAC* de los dispositivos conectados. Pero va “aprendiendo” en la medida que se van transmitiendo datos por la red.

Transmisión de información.

- A partir de ahora hablaremos de transmitir *bytes* enteros, pero la transmisión sigue ocurriendo un *bit* cada vez. **Estos bytes se representarán en formato hexadecimal.**
- Si hay varios dispositivos conectados al mismo *switch* y uno transmite. Solo debe recibir información aquel dispositivo al que esté dirigido el mensaje. A menos que esté dirigido a todo el mundo (cuando la dirección *MAC* de destino es 11111111111111 o FFFF).
- Puede darse el caso, que existan 3 *hosts* en la red y que 2 de ellos quieren transmitirle al 3ro simultáneamente. **Este es uno de los temas que ustedes tienen que aportar ideas o variantes de como enfrentar esa situación.**

Instrucciones

A continuación se muestran las posibles instrucciones adicionales que se incluyen en este 2do proyecto que puede tener el *script.txt*

- `<time> create switch <name> <cantidad de puertos>`
 - **ejemplo:** `0 create switch sw1 4`
- `<time> mac <host> <address>`
 - Este comando permite que asignarle una dirección MAC a una computadora.
 - La dirección *MAC* se especifica en hexadecimal
 - **ejemplo:** `0 mac pc_1 A4B5`
- `<time> send_frame <host> <mac destino> <data>`
 - con este comando se crea una *trama* y se transmite
 - la *trama* a enviar debe contener todos los campos especificados:
 - *MAC* destino
 - *MAC* origen
 - tamaño
 - campo extra
 - datos
 - la *MAC* de origen, el tamaño, etc. deben calcularse.
 - Una *trama* nunca podrá tener más de 255 bytes de datos.
 - Los datos se especifican en formato hexadecimal al igual que la *MAC*
 - **ejemplo:** `20 send_frame pc A5B4 A6F43400FFB34E...`

Salida

- Además de todo lo que se escribe actualmente (del proyecto anterior) se deben crear ahora unos ficheros nuevos por cada computadora de la red. Este fichero tendrá el nombre de la computadora seguido de *_data.txt*
- En este fichero se escribirán solamente los datos recibidos por esa computadora y quién se los envió. Aquí hay que tener en cuenta los datos que fueron enviados directamente a esa computadora, y también los datos que fueron enviados a todo el mundo.
- El formato de salida debe tener en cada línea (separadas por espacio)
 - tiempo final de recepción de los datos

- *MAC* que envió los datos
- datos
- **Ejemplo:** fichero *pc_data.txt*
 - 100 FFA4 A6F43400FFB34E...

Objetivo del proyecto

Como mencionamos anteriormente, la situación a resolver en este 2do proyecto es el caso en el que 2 o más computadoras decidan transmitir información simultáneamente hacia un mismo destino incluso con los *cables* duplex y los *switch*, hay desiciones que tomar. **Lo que se quiere es que ustedes piensen e implementen variantes de como resolver eso.**

Hasta el Momento

Elementos Físicos

- [1] *Cable*
- [1] *Computadora / Host*
- [1] *Concentrador / Hub*
- [2] *Cable Duplex*
- [2] *Conmutador / Switch*

Conceptos

- [2] Dirección *MAC*
- [2] *Trama / Frame*

Instrucciones

- [1] <time> **create host** <name>
- [1] <time> **create hub** <name> <cantidad-de-puertos>
- [1] <time> **connect** <port1> <port2>
- [1] <time> **disconnect** <port>
- [1] <time> **send** <host> <data>
- [2] <time> **create switch** <name> <cantida-de-puertos>
- [2] <time> **mac** <host> <address>
- [2] <time> **send_frame** <host> <mac-destino> <data>