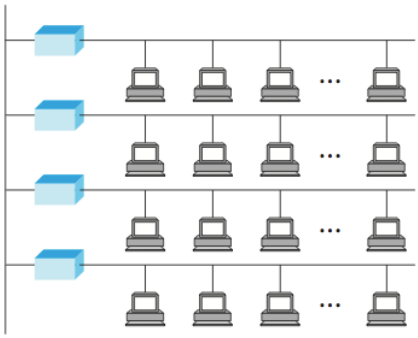
Unidad 5: Nivel de enlace (acceso compartido)

Ethernet (802.3)

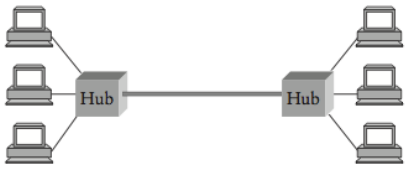
Se trata de una tecnología para crear redes de múltiple acceso sobre un canal compartido. Todos los nodos pueden sensar el estado del medio (*idle*/*busy*) y detectar colisones. El acceso compartido introduce dos nuevos desafíos: control de acceso y esquema de direccionamiento.

Configuración física

- **Las señales se distribuyen a lo largo de todo el cableado**
- Segmento de cable de hasta 500m
- Hosts conectados al segmento
- Posibilidad de juntar segmentos con repetidores:



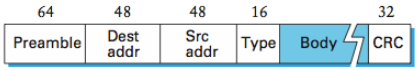
- Repetidores multidireccionales (hubs):



Protocolo de acceso

El protocolo de acceso suele implementarse en hardware en el adaptador de red.

Formato de frame:



- Preámbulo (sincronización)
- Destination address
- Source address
- Type (demultiplexación)
- Body, con longitud mínima (para poder detectar colisiones)
- CRC

Direccionamiento

- aka. MAC address
- Cada adaptador tiene una dirección única asignada en ROM
- Formato legible: seis números de 1 byte (ej: d9:b2:5a:8c:21:c7)
- Cada adaptador recibe todos los paquetes y acepta:
  - los dirigidos a ese adaptador
  - los dirigidos a la dirección de broadcast
  - los dirigidos a una dirección de multicast a la que está atento
  - todos, si está en modo promiscuo

Algoritmo de transmisión

- Si el canal está *idle* se envía (máximo 1500 bytes)
- Si no, espera y transmite inmediatamente cuando se libera
- Cuando dos emiten al mismo tiempo se produce una **colisión**
- Todos los adaptadores pueden detectar la colisión
- Mientras más lejos están los hosts, más se tarde en detectar la colisión
- El mínimo de longitud de frame está para garantizar que siempre se detecten (con longitud máxima del cableado acotada)
- Cuando un emisor detecta colisión, realiza **exponential backoff** con un componente random antes de reintentar
- Tras cierta cantidad de reintentos, el adaptador reporta error al host

Token Ring (802.5, FDDI)

- Otra forma de compartir el medio
- Topología de anillo
- Token (secuencia de bits específica) circulando todo el tiempo
- Cuando un host quiere enviar, espera a recibir el token.
- En vez de reenviarlo, envía los datos.
- Todos los hosts reenvían los datos alrededor del anillo
- El destinatario se guarda una copia
- Cuando el dato llega de vuelva al emisor, deja de retransmitir y pone en circulación el token nuevamente

Wireless (802.11)

Propiedades físicas

La transmisión puede realizarse a través de señales infrarojas o de radio. Las señales infrarojas ya no se utilizan por su poco alcance. En las de radio, la interferencia con otras señales es un problema. Para evitar limitarse a un rango de frecuencias limitado (y posiblemente saturado) se utilizan técnicas de **espectro disperso**.

Espectro disperso

Consiste en expandir el espectro de la señal en un ancho de banda mayor evitando concentrar la potencia sobre una única y estrecha banda de frecuencias. Hay tres técnicas principales:

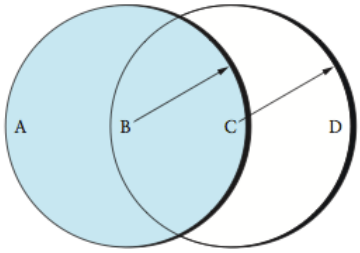
- Frecuency hopping: el emisor va cambiando la frecuencia y el receptor lo sigue, utilizando una secuencia pseudoaliatoria con semilla compartida
- Direct sequence: se agrega ruido pseudo-aleatorio a la señal para que ocupe un ancho de banda mayor
- Multiplexación por división de frecuencia ortogonal

Manejo de colisiones (CSMA-CA)

El método de Ethernet (*CSMA-CD*, *Carrier sense/multiple access with collission detection*) busca detectar las colisiones cuando se sensa el medio. Esto no se puede implementar para el medio inalámbrico:

- No todos los hosts están al alcance de otros, lo cuál dificulta el sensado del medio y la detección de colisiones
- Se necesitaría un medio de radio full duplex que sería muy costoso

Los problemas principales que pueden ocurrir son:



- **Hidden nodes:** dos hosts que no se ven (*A* y *C*) intentan enviar datos a uno que está al alcance de ambos (*B*). Al no verse entre sí, no pueden detectar la colisión.
- **Exposed node:** ocurre cuando un nodo interpreta que no está en condiciones de transmitir por detectar una transmisión que en realidad no lo perjudicará. Ejemplo: *B* transmite a *A*, y *C* quiere transmitir a *D* pero cree que no puede por la transmisión de *B*. En realidad, la transmisión de *B* sólo introducirá interferencia en el segmento entre *B* y *C*.

Lo importante para mediar acceso será detectar actividad **en las cercanías del receptor**, y no alrededor del transmisor. Notar que en el caso de Ethernet no hacía falta diferenciar estos casos.

El método **CSMA-CA**, **Carrier sense/multiple access with collision avoidance** soluciona estos problemas evitando la generación de colisiones (sabiendo que no se pueden detectar tan fácilmente como en *CSMA-CD*).

MACA (Multiple Access with Collision Avoidance)

La idea fundamental es que emisor y receptor intercambian frames de control que son vistos por todos los hosts en el alcance de cada uno y les permiten decidir cuándo se puede transmitir.

- Cuando se quiere transmitir, se envía un *RTS* (request to send) indicando el receptor y por cuánto tiempo se quiere disponer del canal.
- El receptor responde con un *CTS* (clear to send), que también contiene el campo *length*.
- Si un nodo ve el *CTS*, sabe que está cerca del receptor y se abstiene de transmitir por el tiempo que indique *length*.
- Si un nodo ve *RTS* pero no *CTS*, sabe que está lejos del receptor y entonces puede transmitir.
- Si dos nodos alejados envian *RTS* al mismo host, sólo uno recibirá el *CTS*. El otro esperará un tiempo random y reintentará más tarde.

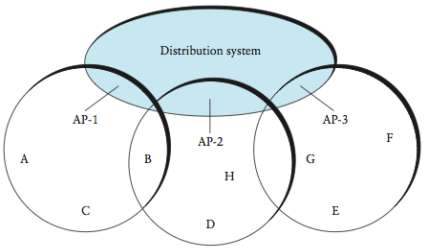
MACAW (MACA for Wireless LANs)

Agregados sobre **MACA**:

- ACK tras cada frame
- Intercambio de información sobre congestionamiento entre hosts
- Cambios en algoritmos de backoff en caso de no poder enviar

Distribución

El estándar define una estructura para las redes, de modo de permitir que nodos que no sean visibles entre sí se puedan comunicar. Además, ataca el problema de la movilidad de los nodos.



- Algunos nodos distinguidos serán **access points** y están conectados a la infraestructura cableada
- Cada access point hace de estación a los hosts que están dentro de su cobertura
- La distribución se realiza a nivel 2 de la arquitectura ISO (no requiere protocolos superiores)
- Cuando un hosts quiere transmitir a otro fuera de su alcance, envía los datos a su access point, que forwardea los mismos al acces point del destinatario a través del sistema de distribución

Para asociarse o cambiar de access point, un host envía un request al cuál responden todos los access points dentro del alcance. Luego el host elige el que prefiere y le confirma la asociación. En caso de que ya haya estado asociado a otro, el nuevo acess point lo da de baja del anterior a través del sistema de distribución.

Packet switching

Las redes punto a punto, ethernet o wireless tienen limitaciones en cuanto a cantidad de hosts y área de cobertura. Para lograr redes con mayor cobertura, se utilizan *switches* que permiten comunicarse a hosts que

no están directamente conectados. Cuando llega un paquete a un puerto de input, el trabajo del switch es decidir por qué puerto de output forwardearlo. Esto permite una nueva topología de estrella:

- Los switches se conectan entre sí y con hosts mediante enlaces punto a punto, permitiendo armar redes más grandes.
- Escalabilidad: con un bus compartido (Ethernet) la capacidad del enlace limita cuánto pueden transmitir los hosts. En esta nueva topología, cada host puede tener su propio enlace al switch y aprovecharlo al máximo.

A grandes rasgos, hay dos grandes tipos de estrategias de forwarding: las de datagramas y las orientadas a conexión.

Datagramas

- Cada paquete tiene la información suficiente para ser dirigido correctamente.
- El switch tiene una tabla que indica por qué puerto enviar los paquetes para cada dirección.
- Es *stateless*, en cualquier momento se puede enviar paquetes.
- Al momento de enviar un paquete, el host no tiene garantías de que la red lo puede manejar.
- Una falla en un enlace puede no traer grandes problemas si la red es capaz de encontrar una via alternativa.

Circuitos virtuales

Es un modelo distinto, donde primero se establece una conexión virtual y luego se transfieren los datos.

En la fase de establecimiento de conexión, los switches tienen estado: una entrada en una tabla de circuitos virtuales con:

- identificador del circuito interno del switch
- interfaz de entrada
- interfaz de salida
- identificador del circuito externo

Cuando entra un paquete por una interfaz con determinado identificador, se selecciona la interfaz de salida y se setea el identificador externo. Estas tablas de circuitos virtuales se pueden configurar estáticamente por el administrador (creando conexiones "permanentes"), pero también se puede utilizar *signalling* para que los hosts los creen y borren dinámicamente.

Una vez establecida la conexión, el origen sabe que el destino es alcanzable y está dispuesto a recibir datos. Además, se pueden reservar recursos para esta conexión (ejemplo: buffers en switches intermedios).

Signalling

- Para iniciar una conexión el host origien envía un mensaje de *setup*
- El mensaje recorre el camino desde el origen al destino (utilizando algoritmo de routing, ver más adelante).
- Cuando llega a destino, los switches desde el final envían a su predecesor el identificador interno que eligieron para esa conexión.
- Cada switch recibe este número de su sucesor y lo setea como el identificador externo.
- Llegado este punto, se tiene la conexión establecida y se pueden enviar datos.
- Cuando se quiere terminar, de envía un mensaje de *teardown*.

LAN Switching

- Llamamos **LAN switching** a la conmutación de paquetes entre LANs de acceso compartido (como Ethernets).
- A los switches en estos casos de los suele llamar LAN switch o **bridges**.
- Decimos que hay una *extended LAN* cuando tenemos una colección de LANs unidas por bridges.

Learning bridges

En el caso más simple, el switch puede forwardear los paquetes por todos los puertos de salida. Una forma de optimizar esto sería forwardear sólo al puerto que corresponda. Esto se puede hacer del siguiente modo:

- En principio se puede enviar todos los paquetes por todos los puertos de salida
- Se inspecciona la dirección de destino de cada paquete, de modo de conocer en qué puerto está el host origgen.
- Se asocia un timeout a cada entrada.
- No hace falta que la tabla esté completa para el funcionamiento de la red, es sólo una optimización.

Spanning tree algorithm

La topología de la LAN extendida puede tener ciclos, ya sea por error o porque se desea proveer caminos alternativos en caso de falla de bridges. Por esto, hace falta un mecanimos para evitar que los paquetes circulen constantemente en la presencia de ciclos.

Esto se logra implementando un algoritmo distribuído de *spanning tree*, que limita la topología a un árbol que cubra todos los puntos pero evite ciclos. Esto es, para cada bridge decide por qué puertos está dispuesto o no a retransmitir paquetes.

Descripción de alto nivel:

- Se asigna un id numérico a cada bridge (prioridad + MAC address)
- El bridge con menor id es elegido root, y siempre forwardea por todos los puertos
- Cada bridge computa el camino mínimo hacia el root, y marca el puerto que pertenece a este camino (*preferred port*).
- Por cada LAN, se elige entre los bridge el más cercano al root para que sea el encargado de forwardear paquetes a esa LAN (*designated bridge*). Se marca el puerto que conecta el bridge con esa LAN (*designated port*).
- Cada bridge desactiva todos sus puertos que no sean *preferred* o *designated*.

Este algoritmo se implementa con un protocolo de intercambio de mensajes de configuración. Los mensajes

contienen:

- Id del bridge
- Id del bridge que se cree que es root
- Hops hasta el supuesto root

Al inicio, todos los bridges se creen el root y envían un mensaje indicando eso. Cada bridge también recuerda cuál es el mejor mensaje que llegó. Un mensaje es mejor que otro si:

- identifica un root con menor id
- identifica un root con igual id pero menor distancia
- identifica un root con igual id a igual distancia, pero el id del emisor es menor

El sistema se estabiliza de la siguiente forma:

- Al momento que un bridge detecta que no es root, deja de enviar sus mensajes y sólo forwardea los que llegan (incrementando en uno la distancia).
- Al momento que un bridge detecta que no es el bridge designado para un puerto, deja de enviar mensajes a ese puerto. Esto ocurre cuando llega por ese puerto un mensaje de un bridge mejor ubicado o con menor id.
- Al final, sólo el root estará generando mensajes, y el resto de los bridges sólo forwardearán por los puertos en donde son bridge designado.

A su vez, el bridge continua enviando mensajes de configuración periódicamente. En caso de que falle un bridge, los que queden desconectados volverán a considerarse root, y el algoritmo vuelve a generar otro árbol. Este mecanismo soluciona fallas de bridges pero no provee routeo alternativo por congestión.