Instituto Politécnico Superior

Administración de Redes Locales

SUBCAPA MAC

Rodríguez Costello, Alejandro
Córdoba, Agustín Leonel

Índice

1. Asignación de canal

Los enlaces de red se pueden dividir en dos categorías: los que utilizan conexiones punto a punto y los que utilizan canales de difusión. En el apunte anterior, hemos tratado con los primeros, mientras que en este, se verán los segundos.

En cualquier red de difusión, el asunto clave es la manera de determinar quién puede utilizar el canal cuando tiene competencia por él. Los protocolos que se utilizan para determinar quién sigue en un canal (de multiacceso) pertenecen a una subcapa de la capa de enlace de datos llamada subcapa MAC (Control de Acceso al Medio). La subcapa MAC tiene especial importancia en las LAN, en especial las inalámbricas puesto que el canal inalámbrico es de difusión por naturaleza. En contraste, las WAN usan muchos enlaces punto a punto, excepto en las redes satelitales. El tema principal de este apunte será el control del canal.

La manera tradicional de asignar un solo canal, como una troncal telefónica, entre múltiples usuarios competidores es dividir su capacidad mediante el uso de esquemas de multiplexión tales como FDM (Multiplexación por División de Frecuencia): si hay N usuarios, el ancho de banda se divide en N partes de igual tamaño, y a cada usuario se le asigna una parte. Debido a que cada usuario tiene una banda de frecuencia privada, no existe interferencia entre ellos. Cuando sólo hay una pequeña cantidad fija y constante de usuarios, cada uno tiene un flujo estable. Sin embargo, cuando el número de emisores es grande y varía continuamente, el FDM presenta algunos problemas. Si el espectro se divide en N regiones y actualmente hay menos de N usuarios interesados en comunicarse, se desperdicia una buena parte del valioso espectro. Y si más de N usuarios quieren comunicarse, a algunos de ellos se les niega el permiso por falta de ancho de banda, aun cuando algunos de los usuarios que tengan asignada una banda de frecuencia apenas transmitan o reciban algo.

Aun suponiendo que el número de usuarios podría, de alguna manera, mantenerse constante en N, dividir el único canal disponible en varios subcanales estáticos es ineficiente por naturaleza. El problema básico es que, cuando algunos usuarios están inactivos, su ancho de banda simplemente se pierde. No lo están usando, y a nadie más se le permite usarlo. Una asignación estática es un mal arreglo para la mayoría de los sistemas de cómputos.

Precisamente los mismos argumentos que se aplican a la FDM se adaptan a otras formas de dividir estáticamente el canal. Si se usara la Multiplexión por División de Tiempo (TDM) y a cada usuario se le asignara cada N-ésima ranura de tiempo, en caso de que un usuario no utilizara la ranura asignada, simplemente se desperdicia.

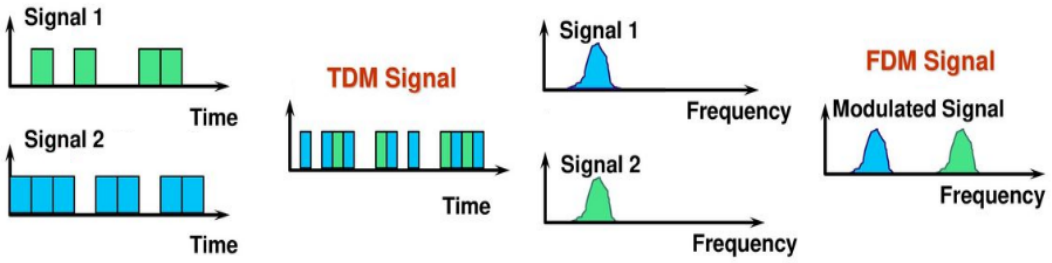

Figura 1: Asignación estática: TDM vs FDM

Para solucionar estos problemas, se desarrollaron los métodos de asignación dinámica de canal. Todos ellos comparten (en mayor o menor medida), los siguientes supuestos:

- Tráfico independiente. El modelo consiste en N hosts independientes, cada uno con un programa o usuario que genera tramas para transmisión. Una vez que se ha generado una trama, el host se bloquea y no hace nada sino hasta que la trama se haya transmitido con éxito.
- Canal único. Hay un solo canal disponible para todas las comunicaciones. Todas las estaciones pueden transmitir en él y pueden recibir de él. Se asume que las estaciones tienen una capacidad equivalente, aunque los protocolos pueden asignarles distintos roles (prioridades).
- Colisiones observables. Si dos tramas se transmiten en forma simultánea, se traslapan en el tiempo y la señal resultante se altera. Este evento se llama colisión. Todas las estaciones pueden detectar una colisión que haya ocurrido. Una trama en colisión se debe volver a transmitir después. No hay otros errores, excepto aquéllos generados por las colisiones.
- Tiempo continuo o ranurado. Se puede asumir que el tiempo es continuo, en cuyo caso la transmisión de una trama puede comenzar en cualquier momento. Por el contrario, el tiempo se puede ranurar o dividir en intervalos discretos (llamados ranuras). Una ranura puede contener 0, 1 o más tramas.
- Detección de portadora o sin detección de portadora. Con el supuesto de detección de portadora, las estaciones pueden saber si el canal está en uso antes de intentar usarlo. Si se detecta que el canal está ocupado, ninguna estación intentará utilizarlo. Si no hay detección de portadora, las estaciones no pueden detectar el canal antes de intentar usarlo. Simplemente transmiten. Sólo después pueden determinar si la transmisión tuvo éxito.

Para evitar cualquier malentendido, se debe tener en cuenta que ningún protocolo multiacceso garantiza una entrega confiable. Aun cuando no haya colisiones, el receptor puede haber copiado alguna trama en forma incorrecta por diversas razones. Otras partes de la capa de enlace o las capas superiores se encargan de proveer confiabilidad.

2. Protocolos de acceso múltiple

2.1. ALOHA

La historia deL primer MAC empieza en la pobre Hawai de principios de la década de 1970. En este caso, interpretaremos "pobre" como que "no tenía un sistema telefónico funcional". Esto no hizo la vida más placentera para el investigador Norman Abramson y sus colegas de la Universidad de Hawai, quienes trataban de conectar a los usuarios en islas remotas a la computadora principal en Honolulu. Tender sus propios cables bajo el Océano Pacífico no era una opción viable, por lo que buscaron una solución diferente. La que encontraron utilizaba radios de corto rango, en donde cada terminal de usuario compartía la misma frecuencia ascendente para enviar tramas a la computadora central. Incluía un método simple y elegante para resolver el problema de asignación de canal.

La idea básica de un sistema ALOHA es sencilla: permitir que los usuarios transmitan cuando tengan datos por enviar. Por supuesto, habrá colisiones y las tramas en colisión se dañarán. Los emisores necesitan alguna forma de saber si éste es el caso. En el sistema ALOHA, después de que cada estación envía su trama a la computadora central, ésta vuelve a difundir la trama a todas las estaciones. Así, una estación emisora puede escuchar la difusión de la estación terrena maestra (hub) para ver si pasó su trama o no. Si la trama fue destruida, el emisor simplemente espera un tiempo aleatorio y la envía de nuevo. El tiempo de espera debe ser aleatorio o las mismas tramas chocarán una y otra vez, en sincronía. Los sistemas en los cuales varios usuarios comparten un canal común de modo tal que puede dar pie a conflictos se conocen como sistemas de contención.

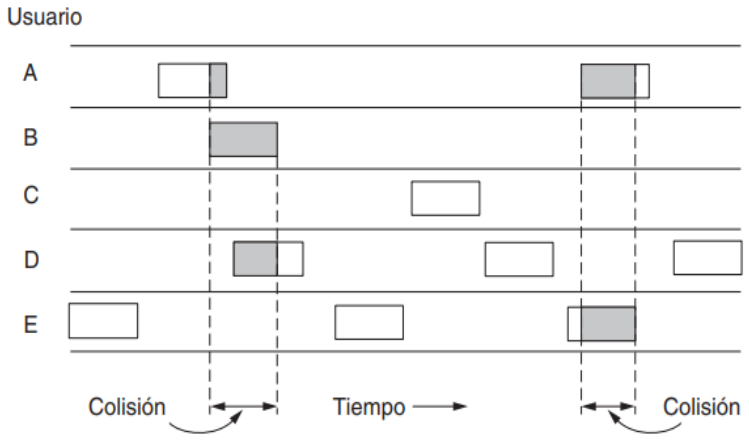

Figura 2: Colisiones en sistema ALOHA

Cada vez que dos tramas traten de ocupar el canal al mismo tiempo, habrá una colisión (como se muestra en la figura anterior) y ambas se dañarán. Si el primer bit

de una trama nueva se traslapa con el último bit de una trama casi terminada, ambas se destruirán por completo (es decir, tendrán sumas de verificación incorrectas) y ambas tendrán que volver a transmitirse más tarde. La suma de verificación no distingue (y no debe) entre una pérdida total y un error ligero. Lo malo es malo.

Poco después de que ALOHA apareció en escena, Roberts (1972) publicó un método para duplicar la capacidad de un sistema ALOHA. Su propuesta fue dividir el tiempo en intervalos discretos llamados ranuras, cada uno de los cuales correspondía a una trama. Este método requiere que los usuarios acuerden límites de ranura. Una manera de lograr la sincronización sería tener una estación especial que emitiera una señal al comienzo de cada intervalo, como un reloj. En el método de Roberts, que se conoce como ALOHA ranurado, en contraste con el ALOHA puro de Abramson, no se permite que una estación envíe cada vez que el usuario escribe una línea. En cambio, se le obliga a esperar el comienzo de la siguiente ranura. Por lo tanto, el ALOHA de tiempo continuo se convierte en uno de tiempo discreto.

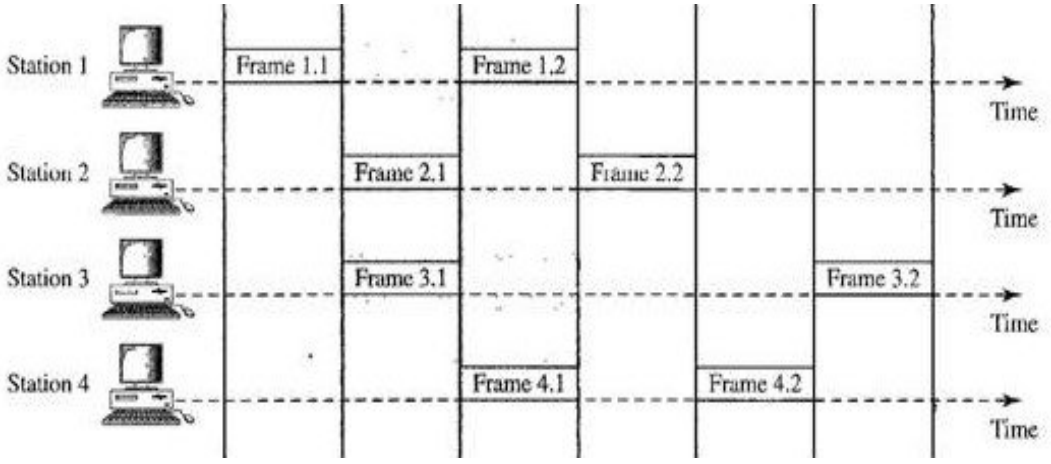

Figura 3: ALOHA ranurado

Aunque el ALOHA ranurado ofrece un poco más de orden al canal, aún pueden ocurrir colisiones, y esto es porque las estaciones transmiten a voluntad propia sin prestar atención a lo que están haciendo las demás estaciones. Sin embargo, en las redes LAN es posible que las estaciones detecten lo que están haciendo las demás estaciones y adapten su comportamiento con base en ello.

2.2. Protocolos de detección de portadora

Los protocolos en los que las estaciones escuchan una portadora (es decir, una transmisión) y actúan de manera acorde se llaman protocolos de detección de portadora. El primer protocolo de detección de portadora que estudiaremos se llama CSMA

(Acceso Múltiple con Detección de Portadora) persistente-1. Cuando una estación tiene datos por enviar, primero escucha el canal para saber si alguien más está transmitiendo en ese momento. Si el canal está inactivo, la estación envía sus datos. Por el contrario, si el canal está ocupado, la estación espera hasta que se desocupa. A continuación, la estación transmite una trama. Si ocurre una colisión, la estación espera una cantidad aleatoria de tiempo y comienza de nuevo. El protocolo se llama persistente-1 porque la estación transmite con una probabilidad de 1 (es decir, siempre) cuando encuentra que el canal está inactivo.

Podría esperarse que este esquema evite las colisiones, excepto en el extraño caso de los envíos simultáneos, pero de hecho no lo hace. Si dos estaciones están listas a la mitad de la transmisión de una tercera estación, ambas esperarán amablemente hasta que termine la transmisión y después ambas empezarán a transmitir exactamente al mismo tiempo, lo cual producirá una colisión. Si no fueran tan impacientes, habría menos colisiones. Aun así, este protocolo tiene un mejor desempeño que el ALOHA puro, ya que ambas estaciones tienen la decencia de dejar de interferir con la trama de la tercera estación.

Un segundo protocolo de detección de portadora es el CSMA no persistente. En este protocolo se hace un intento consciente por ser menos egoísta que en el previo. Como antes, una estación escucha el canal cuando desea enviar una trama y, si nadie más está transmitiendo, comienza a hacerlo. Pero si el canal ya está en uso, la estación no lo escuchará de manera continua con el fin de tomarlo de inmediato al detectar el final de la transmisión anterior, sino que esperará un periodo aleatorio y repetirá el algoritmo. En consecuencia, este algoritmo conduce a un mejor uso del canal pero produce mayores retardos que el CSMA persistente-1.

El último protocolo es el CSMA persistente-p, que se aplica a canales ranurados. Cuando una estación está lista para enviar, escucha el canal. Si se encuentra inactivo, la estación transmite con una probabilidad p. Con una probabilidad $q = 1 - p$, se posterga hasta la siguiente ranura. Si esa ranura también está inactiva, la estación transmite o posterga una vez más, con probabilidades p y q. Este proceso se repite hasta que se transmite la trama o hasta que otra estación comienza a transmitir.

En definitiva, los protocolos CSMA persistentes y no persistentes son una mejora respecto a ALOHA porque aseguran que ninguna estación empezará a transmitir mientras el canal esté ocupado. Pero si dos estaciones detectan que el canal está inactivo y empiezan a transmitir al mismo tiempo, sus señales de todas formas sufrirán una colisión. Otra mejora es que las estaciones detecten rápidamente la colisión y dejen de transmitir de inmediato (en vez de terminadas las transmisiones), ya que de todas formas se alterarán y no se podrán recuperar. Este protocolo, conocido como CSMA/CD (CSMA con Detección de Colisiones), es la base de la clásica LAN

Ethernet. Es importante tener en cuenta que la detección de colisiones es un proceso analógico. El hardware de la estación debe escuchar el canal mientras transmite. Si la señal que recibe es distinta de la señal que está enviando, sabe que está ocurriendo una colisión.

CSMA/CD utiliza el modelo conceptual de la figura a continuación. En el punto marcado como t_0, una estación ha terminado de transmitir su trama. Cualquier otra estación que tenga una trama por enviar puede intentar hacerlo ahora. Si dos o más estaciones deciden transmitir en forma simultánea, habrá una colisión. Si una estación detecta una colisión, aborta la transmisión, espera un tiempo aleatorio e intenta de nuevo (suponiendo que ninguna otra estación ha comenzado a transmitir durante ese lapso). Por lo tanto, el modelo de CSMA/CD consiste en periodos alternantes de contención y transmisión, con periodos de inactividad que ocurren cuando todas las estaciones estén en reposo.

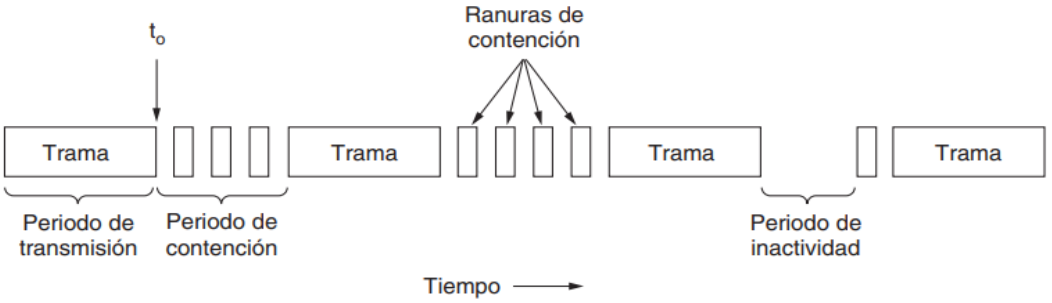

Figura 4: CSMA/CD puede estar en estado de contención, de transmisión o inactivo

Aunque las colisiones no ocurren en CSMA/CD una vez que una estación ha capturado el canal sin ambigüedades, aún pueden ocurrir durante el periodo de contención. Estas colisiones afectan en forma adversa el desempeño del sistema. Las colisiones no sólo reducen el ancho de banda, sino que también hacen variable el tiempo de envió de una trama, lo cual no es bueno para el tráfico en tiempo real. Para evitar las colisiones, se han desarrollado varios protocolos con estrategias para que, terminantemente, no ocurran. En la actualidad, la mayoría de estos protocolos no se utilizan en los sistemas grandes pero sirven de base para modelar futuros modelos en respuesta al constante cambio de la tecnología.

2.3. Protocolos libres de colisiones

El primer protocolo libre de colisiones es el método básico de mapa de bits. Cada periodo de contención consiste exactamente de N ranuras. Si la estación 0 tiene una trama por enviar, transmite un bit 1 durante la ranura 0. No está permitido a ninguna otra estación transmitir durante esta ranura. Sin importar lo que haga la estación 0, la estación 1 tiene la oportunidad de transmitir un bit 1 durante la

ranura 1, pero sólo si tiene una trama puesta en la cola. En general, la estación j puede anunciar que tiene una trama por enviar, para lo cual inserta un bit 1 en la ranura j. Una vez que han pasado las N ranuras, cada estación tiene un completo conocimiento acerca de cuáles son las estaciones que quieren transmitir. En ese punto, las estaciones empiezan a transmitir en orden numérico.

Como todos están de acuerdo en quién sigue a continuación, nunca habrá colisiones. Una vez que la última estación lista haya transmitido su trama, un evento que pueden detectar fácilmente todas las estaciones, comienza otro periodo de contención de N bits. Si una estación está lista justo después de que ha pasado su ranura de bit, ha tenido mala suerte y deberá permanecer inactiva hasta que cada una de las demás estaciones haya tenido su oportunidad y el mapa de bits haya comenzado de nuevo. Los protocolos como éste en los que el interes de transmitir se difunde antes de la transmisión se llaman protocolos de reservación, debido a que reservan la propiedad del canal por anticipado y evitan colisiones.

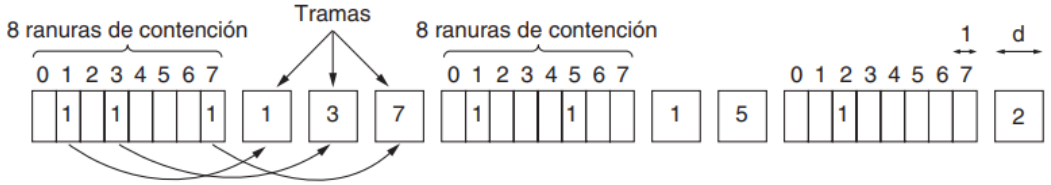

Figura 5: Método básico de mapa de bits

La esencia del protocolo de mapa de bits es que permite que cada estación transmita una trama por turno, en un orden predefinido. Otra forma de lograr lo mismo es pasar un pequeño mensaje conocido como *token* de una estación a otra, en el mismo orden predefinido. El token representa el permiso para enviar. Si una estación tiene una trama puesta en cola para transmitirla cuando recibe el token, puede enviar esa trama antes de pasar el token a la siguiente estación. Si no tiene una trama puesta en cola, simplemente pasa el token. En un protocolo token ring, la topología de la red se utiliza para definir el orden en el que las estaciones envían información. Las estaciones están conectadas una con otra en un solo anillo. Así, el proceso de pasar el token a la siguiente estación consiste en recibir el token proveniente de una dirección y transmitirlo hacia la otra dirección. Las tramas también se transmiten en la dirección del token. De esta forma, circularán alrededor del anillo y llegarán a la estación de destino.

El desempeño del protocolo de paso de token es similar al del protocolo de mapa de bits, aunque las ranuras de contención y las tramas de un ciclo están ahora entremezcladas. Después de enviar una trama, cada estación debe esperar a que las N estaciones (incluyéndose a sí misma) envíen el token a sus estaciones vecinas y que las otras N–1 estaciones envíen una trama, si es que la tienen.

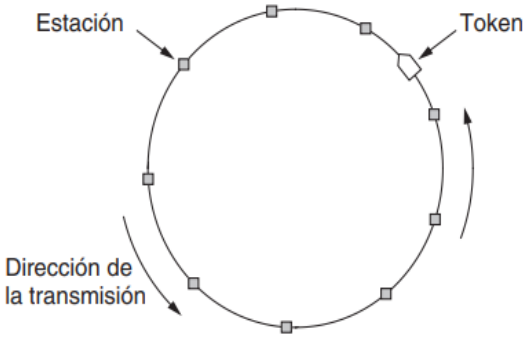

Figura 6: Protocolo token ring

Un problema con el protocolo básico de mapa de bits, y en consecuencia con el paso de token, es que la sobrecarga es de 1 bit por estación, por lo que no se escala bien en redes con miles de estaciones. Se puede tener mejores resultados si se usan direcciones de estación binarias con un canal que combine las transmisiones. Una estación que quiere utilizar el canal en un momento dado difunde su dirección como una cadena binaria de bits, comenzando por el bit de mayor orden. Se supone que todas las direcciones tienen la misma longitud. A todos los bits en cada posición de dirección de las diferentes estaciones se les aplica un OR BOOLEANO por el canal cuando se envían al mismo tiempo. A este protocolo se lo llama conteo descendente binario.

Para evitar conflictos, es necesario aplicar una regla de arbitraje: tan pronto como una estación ve que una posición de bit de orden alto, cuya dirección es 0, ha sido sobrescrita con un 1, se da por vencida. Por ejemplo, si las estaciones 0010, 0100, 1001 y 1010 están tratando de obtener el canal, en el primer tiempo de bit las estaciones transmiten 0, 0, 1 y 1, respectivamente. A éstos se les aplica el OR para formar un 1. Las estaciones 0010 y 0100 ven el 1 y saben que una estación de mayor numeración está compitiendo por el canal, por lo que se dan por vencidas durante esta ronda. Las estaciones 1001 y 1010 continúan. El siguiente bit es 0, y ambas estaciones continúan. El siguiente bit es 1, por lo que la estación 1001 se da por vencida. La ganadora es la estación 1010, debido a que tiene la dirección más alta. Después de ganar la contienda, ahora puede transmitir una trama, después de lo cual comienza otro ciclo de contienda.

3. Ethernet

Empezaremos nuestro estudio de los sistemas reales con Ethernet, que probablemente sea el tipo más usado de red de computadoras en el mundo. Existen dos tipos de Ethernet: Ethernet clásica, que resuelve el problema de acceso múltiple mediante el uso de las técnicas que se explicaron en las secciones anteriores; el segundo tipo es la Ethernet conmutada, en donde los dispositivos llamados switches se utilizan para conectar distintas computadoras. Es importante mencionar que, aunque se hace referencia a ambas como Ethernet, son muy diferentes. La Ethernet clásica es la forma original que operaba a tasas de transmisión de 3 a 10 Mbps. La Ethernet conmutada es en lo que se convirtió la Ethernet y opera a 100, 1000 y 10000 Mbps, en formas conocidas como Fast Ethernet, Gigabit Ethernet y 10 Gigabit Ethernet. Actualmente, en la práctica sólo se utiliza Ethernet conmutada.

Cada versión de Ethernet tiene una longitud de cable máxima por segmento (es decir, longitud sin amplificar) a través de la cual se propaga la señal. Para permitir redes más grandes, se pueden conectar varios cables mediante repetidores. Un repetidor es un dispositivo de capa física que recibe, amplifica (es decir, regenera) y retransmite las señales en ambas direcciones. En cuanto a lo que al software concierne, una serie de segmentos de cable conectados por repetidores no presenta ninguna diferencia en comparación con un solo cable (excepto por una pequeña cantidad de retardo que introducen los repetidores).

La Ethernet clásica utiliza el algoritmo CSMA/CD persistente-1. Este descriptor tan sólo significa que las estaciones detectan el medio cuando tienen una trama que desean enviar, y la envían tan pronto como el medio está inactivo. Monitorean el canal por si hay colisiones al momento en que envían. Si hay una colisión, abortan la transmisión con una señal de bloqueo corta y vuelven a transmitir después de un intervalo aleatorio. Si no hay colisión, el emisor supone que la trama probablemente se entregó con éxito. Es decir, ni CSMA/CD ni Ethernet proveen confirmaciones de recepción. Esta elección es apropiada para los canales de cable de cobre y de fibra óptica que tienen tasas de error bajas. Cualquier error que ocurra debe entonces detectarse mediante CRC y recuperarse en las capas superiores.

Pronto Ethernet empezó a evolucionar y a alejarse de la arquitectura de un solo cable extenso de la Ethernet clásica. Los problemas asociados con el hecho de encontrar interrupciones o conexiones flojas condujeron hacia un distinto tipo de patrón de cableado, en donde cada estación cuenta con un cable dedicado que llega a un hub (concentrador) central. Un hub simplemente conecta de manera eléctrica todos los cables que llegan a él, como si estuvieran soldados en conjunto.

En un principio, los cables eran pares trenzados de la compañía telefónica, ya que

la mayoría de los edificios de oficinas contaban con este tipo de cableado y por lo general había muchos de sobra. Esta reutilización fue una ventaja, pero a la vez se redujo la distancia máxima de cable del hub hasta 100 metros. Con las ventajas de usar el cableado existente y la facilidad de mantenimiento, los hubs de par trenzado se convirtieron rápidamente en la forma dominante de Ethernet. Sin embargo, los hubs no incrementan la capacidad debido a que son lógicamente equivalentes al cable extenso individual de la Ethernet clásica. A medida que se agregan más estaciones, cada estación recibe una parte cada vez menor de la capacidad fija. En un momento dado, la LAN se satura.

Por fortuna existe otra forma de tratar con el aumento de carga: una Ethernet conmutada. El corazón de este sistema es un conmutador (switch) que contiene un plano posterior de alta velocidad, el cual conecta a todos los puertos. . Desde el exterior, un switch se ve igual que un hub. Ambos son cajas que por lo general contienen de 4 a 96 puertos, cada uno con un conector estándar RJ-45 para un cable de par trenzado (UTP). Cada cable conecta al switch o hub con una sola computadora. Sin embargo, dentro del switch ocurre algo muy distinto. Los switches sólo envían tramas a los puertos para los cuales están destinadas. Cuando el puerto de un switch recibe una trama Ethernet de una estación, el switch verifica las direcciones de Ethernet para ver cuál es el puerto de destino de la trama. A continuación, el switch reenvía la trama a través de su plano posterior de alta velocidad hacia el puerto de destino. Después, el puerto de destino transmite la trama sobre el cable, de manera que pueda llegar a la estación de destino. Ninguno de los otros puertos sabe siquiera que existe la trama.

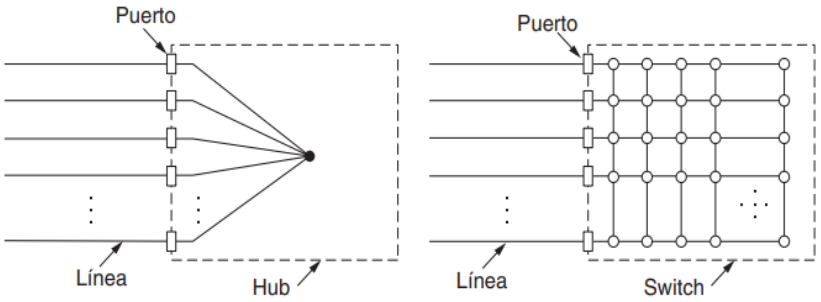

Figura 7: Hub vs Switch: estructura interna

¿Qué ocurre si más de una estación o puerto desea enviar una trama al mismo tiempo? De nuevo, los switches difieren de los hubs. En un hub, todas las estaciones están en el mismo dominio de colisión. Deben usar el algoritmo CSMA/CD para programar sus transmisiones. En un switch, cada puerto es su propio dominio de colisión independiente.

Un switch mejora el desempeño de la red en comparación con un hub de dos maneras. Primero, como no hay colisiones, la capacidad se utiliza con más eficiencia. Segundo y más importante, con un switch se pueden enviar varias tramas al mismo tiempo (por distintas estaciones). Estas tramas llegarán a los puertos del switch y viajarán hacia el plano posterior de éste para enviarlos por los puertos apropiados. No obstante, como se podrían enviar dos tramas al mismo puerto de salida y al mismo tiempo, el switch debe tener un búfer para que pueda poner temporalmente en cola una trama de entrada hasta que se pueda transmitir al puerto de salida. En general, estas mejoras producen una considerable ganancia en el desempeño que no es posible lograr con un hub. Con frecuencia, la velocidad real de transmisión total del sistema se puede incrementar en un orden de magnitud, dependiendo del número de puertos y patrones de tráfico.

Ethernet ha existido por décadas y no tiene competidores serios a la vista, por lo que es probable que exista por algunos años más. Pocas arquitecturas de CPU, sistemas operativos o lenguajes de pogramación han subsistido tanto sin mostrar debilidad. Quíza la razón principal de su longevidad es que Ethernet es simple y flexible. En la práctica, simple se traduce como confiable, económico y fácil de mantener. Una vez que se adoptó la arquitectura de hub y switch, casi no ocurrían fallas. Las personas dudaban en reemplazar algo que funciona bien todo el tiempo. Asimismo, el cableado de par trenzado tiene un costo relativamente bajo, al igual que los componentes de hardware. Pueden empezar con un alto costo cuando hay una transición; por ejemplo, nuevas NICs o switches de Gigabit Ethernet, pero son simples adiciones para una red bien establecida (no su reemplazo) y los precios bajan con rapidez a medida que el volumen de ventas aumenta.

Ethernet es fácil de mantener. No hay software que instalar (sólo los controladores) y tampoco hay tablas de configuración que manejar (con el riesgo de equivocarse). Además, agregar nuevos hosts es tan simple como conectarlos. Otro punto es que Ethernet interactúa fácilmente con TCP/IP, el cual se ha vuelto dominante, y que veremos en el transcurso del año.

4. Conmutación en la capa de enlace de datos

Muchas organizaciones tienen varias redes LAN y desean interconectarlas. ¿No sería conveniente si tan sólo se puediera unir las redes LAN para formar una LAN más grande? De hecho, este tipo de redes se puede conectar mediante dispositivos llamados puentes. Los puentes operan en la capa de enlace de datos, por lo que examinan las direcciones de la capa de enlace de datos para reenviar tramas. Como no tienen que examinar el campo de carga útil de las tramas que reenvían, pueden manejar paquetes IP al igual que otros tipos de paquetes. En contraste, los enrutadores (que veremos más adelante en el año) examinan las direcciones de los paquetes y realizan su trabajo de enrutamiento con base en ellas, por lo que sólo funcionan con los protocolos para los cuales se diseñaron. La topología de dos redes LAN conectadas por un puente se muestra en la figura a continuación.

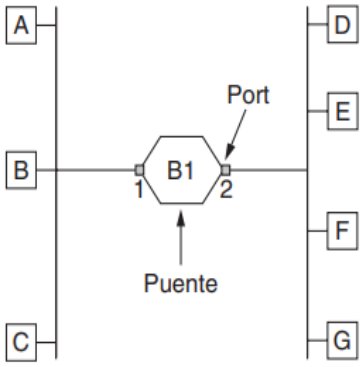

Figura 8: Topología de LANs con puente interconector

Ahora consideremos lo que ocurre dentro de los puentes. Cada puente opera en modo promiscuo; es decir, acepta cada una de las tramas que transmiten las estaciones conectadas a cada uno de sus puertos. El puente debe decidir si va a reenviar o desechar cada trama y, en caso de que sea la primera opción, también debe decidir por qué puerto enviar la trama. Esta decisión se basa en la dirección de destino. Una forma simple de implementar este esquema es mediante una gran tabla (hash) dentro del puente. La tabla puede listar cada posible destino y a qué puerto de salida pertenece. Cuando se conectan por primera vez los puentes, todas las tablas de hash están vacías. Ninguno de los puentes sabe dónde se encuentran los destinos, por lo que utilizan un algoritmo de inundación: todas las tramas que llegan con un destino desconocido se envían por todos los puertos a los que está conectado el puente, excepto por el que llegaron. Con el paso del tiempo, los puentes aprenden dónde están los destinos. Una vez conocido un destino, las tramas destinadas para él se colocan sólo en el puerto apropiado; no se inundan. El algoritmo que usan los puentes es el de aprendizaje hacia atrás. Cabe destacar que este mismo método es utilizado por los switchs.

La topología puede cambiar conforme las máquinas y los puentes se enciendan y apaguen, o cuando se trasladan de un sitio a otro. Para manejar topologías dinámicas, siempre que se realiza una entrada en una tabla de hash se registra en la entrada la hora de llegada de una trama. Cada vez que llega una trama cuyo origen ya está en la tabla, su entrada se actualiza con la hora actual. Así, la hora asociada a cada entrada indica la última vez que se registró una trama proveniente de esa máquina. Un proceso en el puente analiza de manera periódica la tabla de hash y purga todas las entradas que tengan más de algunos minutos de antigüedad. De esta manera, si una computadora se desconecta de su LAN, se traslada a otro lugar del edificio y se vuelve a conectar en algún otro lugar, en pocos minutos volverá a funcionar con normalidad, sin necesidad de intervención manual. Este algoritmo también significa que si una máquina está inactiva durante algunos minutos, el tráfico destinado a ella se inundará hasta que la máquina misma envíe una trama.

Cuando se necesita conectar dos puentes (o switchs), para incrementar la confiabilidad, se pueden usar enlaces redundantes como se muestra a continuación. Hay dos enlaces en paralelo entre un par de puentes. Este diseño asegura que si se corta un enlace, la red no se dividirá en dos conjuntos de computadoras que no se pueden comunicar entre sí.

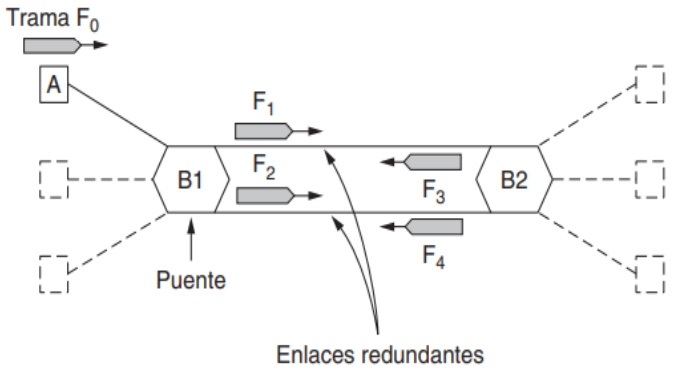

Figura 9: Puentes con enlaces redundantes

Sin embargo, esta redundancia introduce algunos problemas adicionales, porque crea ciclos en la topología. La solución a este problema es que los puentes se comuniquen entre sí y cubran la topología existente con un árbol de expansión que llegue a todos los puentes. En efecto, algunas conexiones potenciales entre los puentes se ignoran en el afán de construir una topología ficticia libre de ciclos, que sea un subconjunto de la topología actual.

Por ejemplo, en la figura que sigue se pueden observar cinco puentes interconectados y que también tienen estaciones conectadas. Cada estación se conecta sólo a un puente. Hay algunas conexiones redundantes entre los puentes, de modo que las

tramas se reenviarán en ciclos si se utilizan todos los enlaces. Podemos considerar esta topología como un grafo en el que los puentes son los nodos y los enlaces punto a punto son los bordes (estos términos y sus operatorias los aprenderan en la materia de 6to año "Teoría de Grafos"). El grafo se puede reducir a un árbol de expansión, el cual no tiene ciclos por definición, si se eliminan los enlaces que se muestran como líneas punteadas. Si usamos este árbol de expansión, hay exactamente una ruta de cada estación a cada una de las demás estaciones. Una vez que los puentes se hayan puesto de acuerdo en cuanto al árbol de expansión, todos los reenvíos entre las estaciones se hacen a través del árbol de expansión. Puesto que existe una única ruta de cada origen a cada destino, es imposible que se produzcan ciclos. Para construir el árbol de expansión, los puentes ejecutan un algoritmo distribuido. Cada puente difunde en forma periódica un mensaje de configuración a través de todos sus puertos hacia sus vecinos y procesa los mensajes que recibe de otros puentes. Estos mensajes no se reenvían, ya que su propósito es construir el árbol y usarlo para los reenvíos. A esta operatoria de la denomina Protocolo de Árbol de Expansión (STP).

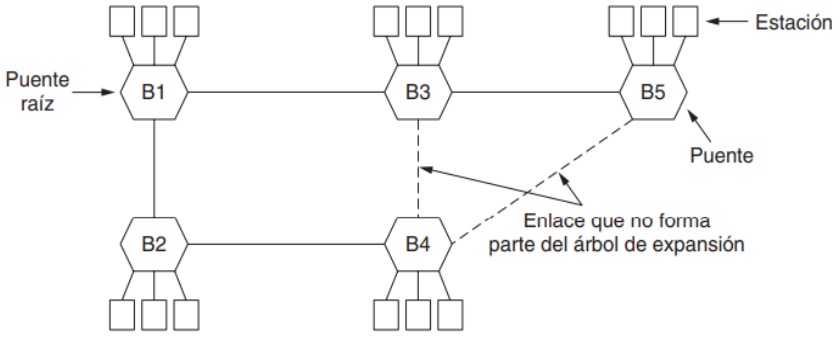

Figura 10: Puentes con enlaces redundantes

Para ver su proceso en detalle, la cátedra propone al alumno conectar cinco switchs de forma redundante en Packet Tracer, y observar como se distribuye el STP mientras cambian sus tablas hash. Notar en especial la denominación que se le asigna a cada puerto en cada uno de los switchs.

4.1. Redes LAN virtuales

En respuesta a la demanda de mayor flexibilidad por parte de los usuarios, los fabricantes de redes empezaron a trabajar en una forma de volver a cablear edificios completos mediante software. Esta situación puede ocurrir, por ejemplo, cuando se anexan nuevos puestos de trabajo en áreas completamente distintas a las que pertenecen pero aún así se desea conectarlos con los usuarios actuales. El concepto que surgió se denomina VLAN (LAN Virtual). Las redes VLAN se basan en switches especialmente diseñados para este propósito. Para configurar una red VLAN, el administrador de la red decide cuántas VLAN habrá, qué computadoras habrá en cuál VLAN y cómo se llamarán las VLAN. A menudo se les asignan nombres mediante

colores (de manera informal), ya que de esta manera es posible imprimir diagramas a color que muestren la disposición física de las hosts pertenecientes a cada una de ellas.

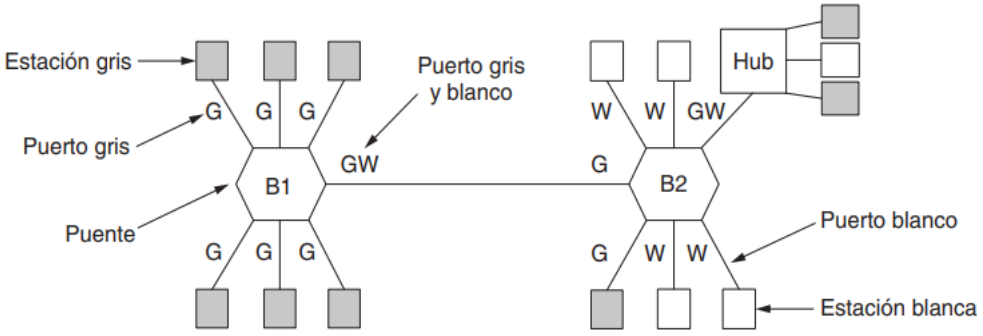

Figura 11: Distribución de VLANs gris y blanco en la LAN general

Para que las VLAN funcionen correctamente, es necesario establecer tablas de configuración en los puentes. Estas tablas indican cuáles VLAN se pueden acceder a través de qué puertos.

Como ejemplo, suponga que una de las estaciones grises conectadas al puente B1 en la figura, envía una trama a un destino que no se conoce de antemano. El puente B1 recibirá la trama y verá que proviene de una máquina en la VLAN gris, por lo que inundará esa trama en todos los puertos etiquetados como G (excepto el puerto entrante). La trama se enviará a las otras cinco estaciones grises conectadas a B1, así como a través del enlace de B1 al puente B2. En el puente B2, la trama se reenvía de manera similar a todos los puertos etiquetados como G. Esto envía la trama a una estación más y al hub (que transmitirá la trama a todas sus estaciones). El hub tiene ambas etiquetas debido a que se conecta a las máquinas de ambas redes VLAN. La trama no se envía en otros puertos que no tengan G en la etiqueta, puesto que el puente sabe que no hay máquinas en la VLAN gris a las que se pueda llegar por medio de estos puertos.

Para implementar este esquema, los puentes necesitan saber a qué VLAN pertenece una trama entrante. Sin esta información, por ejemplo, cuando el puente B2 recibe una trama del puente B1, no puede saber si reenviar la trama a la VLAN gris o blanca. Si se estuviera diseñando un nuevo tipo de LAN, sería muy fácil sólo agregar un campo VLAN en el encabezado de cada trama. El comité estandarizador se enfrentó a este problema en 1995. Después de muchas discusiones, hizo lo impensable y cambió el encabezado de Ethernet. El nuevo formato contiene una etiqueta VLAN.