Bridging

Learning bridging

Learning bridging es la acción de recibir un frame ethernet por un puerto y despacharlo por otro, de acuerdo al destino indicado en el frame, de forma transparente y sin recibir instrucciones o información adicional. Para poder hacer bridging, un dispositivo debe poder interpretar un frame y entender el contenido de al menos el origen y destino del frame.

Para poder hacer learning bridging, cada switch posee una tabla en donde se guardan las MAC address que se encuentran en cada puerto. Debido a que las estaciones pueden cambiar de puerto sin previo aviso, las entradas de esta tabla deben eliminarse periódicamente.

Cuando se envía un frame y este pasa a través de un switch, el switch puede determinar sin lugar a dudas que la MAC address de origen de ese paquete se encuentra en el puerto de donde vino el frame. Este es el único dato fehaciente que puede determinar.

El puerto de salida a usar para el frame es determinado por los contenidos de la tabla de MAC addresses. Si el destino se encuentra en la tabla, entonces el switch enviará el frame directamente por el puerto indicado en ella. Si el destino es desconocido (o es la dirección broadcast), el switch deberá inundar por todos los puertos excepto el de entrada, de forma similar a como lo hace un hub.

El tratamiento de un frame no cambia si hay involucrados varios switches. En la tabla figurarán varias MAC address distintas asociadas a un mismo puerto, que está conectado al otro switch. Este es un caso ya previsto que ocurre también cuando se conecta un hub a un switch.

El bridging es transparente. Si una estación es movida de un puerto a otro, el o los switches deberían adaptarse automáticamente. Si el switch está conectado directamente a la estación, la pérdida de señal de nivel 1 debería bastar para indicarle al switch que la estación no se encuentra más en el puerto indicado en la tabla, lo que haría que la entrada se borre inmediatamente y el switch pase a inundar los frames destinados a esa estación. Si la conexión no es directa, entonces no será posible detectar la desaparición de la estación y el switch seguirá enviando los frames por el puerto indicado en la tabla, hasta que esta entrada se venza o hasta que el host movido envíe un frame y el bridge aprenda la nueva localización del mismo. Esto puede causar pérdida de datos por varios segundos.

Es importante tener en cuenta que bridging no es un mecanismo de seguridad (aunque contribuye impidiendo que datos innecesarios sean enviados a puertos que no necesitan recibirlos). Existen varios ataques capaces de neutralizar el efecto de learning bridging de ocultar mensajes, incluyendo llenar la tabla de un switch con direcciones MAC falsas (haciendo que el switch deba inundar todos los mensajes) o enviar mensajes con origen falsificado, forzando al switch a mandar los datos al puerto equivocado (donde un intruso puede estar esperando para interceptarlos).

Spanning Tree

Puesto que una red ethernet no tiene forma de determinar el tiempo de vida de un frame, esta no debe tener ciclos. Si los tuviera, un frame broadcast que alcanzara el ciclo quedaría ciclando permanentemente y los switches enviarían constantemente copias de ese mensaje por todas las interfaces.

Para permitir tener enlaces redundantes al mismo tiempo que se evita este problema, el IEEE estandarizó la técnica de Spanning Tree (IEEE 802.1d). Usando spanning tree, los switches de una red pueden formar un árbol sobre un subconjunto de los enlaces que posee la red, manteniendo la opción de cambiar de enlaces si hubiera problemas con los enlaces elegidos. Para ello, los switches envían cada dos segundos unos paquetes que indican la topología de la red, llamados BPDU.

El algoritmo de spanning tree tiene varias fases: listening, learning, forwarding y blocking.

Cada switch tiene un valor llamado “bridge ID”, formado por un valor elegido por el administrador (bridge priority) y su propia dirección MAC. El valor del bridge priority tiene más influencia sobre el bridge ID, mientras que la dirección MAC sirve para desempatar en caso de haber dos bridge priority iguales.

Apenas arranca una red, todos sus switches entran en la fase de listening. En esta fase, los switches tratan de determinar cuál va a ser la raíz del árbol. Para ello, cada switch envía una propuesta indicando su bridge ID y un candidato a ser raíz (que comienza siendo el mismo switch). Un switch mantiene el valor de la propuesta de raíz hasta que recibe una propuesta con bridge ID más bajo. Cuando esto sucede, el switch cambia su propuesta por la que acaba de recibir, en cada paso anunciando un bridge ID más bajo que el anterior.

Este intercambio se repite siete veces, y al final de esta cantidad de iteraciones todos los switches de la red deberían estar anunciando como raíz al switch con bridge ID más bajo, que termina ganando el puesto de raíz.

Una vez terminada esta fase, los switches entran en la fase de learning. En esta fase, la raíz emite BPDUs indicando quién es la raíz y cuál es el costo hacia ella, si se usa como camino el enlace por el que están viniendo los frames (obviamente en la raíz este valor es cero). Cuando un switch recibe estos frames, los retransmite por todos sus puertos, excepto el que usa para llegar a la raíz (llamado root port), actualizando los costos para tener en cuenta los nuevos enlaces agregados. Si un switch recibiera BPDUs por dos puertos distintos, elegirá como root port al puerto con menor costo a la raíz. Estas acciones hacen que la red calcule de forma distribuída un árbol que minimice los costos de llegar a la raíz, de forma similar a como lo haría el algoritmo de dijkstra. Esta etapa se llama learning porque adicionalmente cada switch empieza a detectar las estaciones que se encuentran en cada puerto y a agregar sus MAC address a la tabla.

Una vez terminada esta etapa, cada puerto pasa a los estados de forwarding o blocking, dependiendo de si llevan a la raíz o no y cual fue el root port elegido. En todos los casos se envían y reciben BPDUs cada dos segundos, que son usados para determinar si hubieron cambios en la topología de la red.

Esta limitación de siete etapas define tanto el diámetro de la red como el tiempo máximo que tardará cada etapa. Puesto que a lo sumo se realizarán siete intercambios en cada etapa, el diámetro de la red (la distancia más corta entre los dos nodos más alejados) cuando se usa spanning tree no debe exceder siete saltos. Si esto ocurriera, distintas partes de la red podrían designar a distintos switches como raíz del árbol, cortando la red en dos. El IEEE, al elegir siete saltos como diámetro máximo, también impuso un límite de quince segudos para la etapa de listening y otros quince para la etapa de learning. Eso debería ser suficiente para realizar los siete intercambios, a un intercambio por segundo, con un segundo de margen para realizar el procesamiento de datos.

Una variante de spanning tree es Rapid STP. RSTP es una extensión de STP en donde los tiempos de reacción son más rápidos y se tienen puertos desigados como backup. El objetivo de RSTP es minimizar el tiempo en el cual la red no funciona debido a cambios de topología.

VLANs

Bajo ciertas circunstancias, es deseable dividir una LAN en varias partes aisladas una de otra. Esta división puede ser requerida por cuestiones de economía (por ejemplo si se tiene un único switch y no se desea o se puede comprar otro para la nueva LAN), por razones de seguridad (se desea aislar un sector de la red particularmente peligroso) o por razones de flexibilidad (se desea cambiar periodicamente a las estaciones de LAN, pero se quiere evitar cambiar constantemente el cableado físico). Para lograr esto, existen las llamadas LAN virtuales o VLANs.

Si la red está acotada a un solo switch, la división es fácil, ya que está limitada a una configuración

en el switch. Si el switch soporta VLANs, es posible indicarle que debe dividir la red asignando ciertos puertos a distintas LAN (algo común) o indicando que estaciones deben estar en que LAN, discriminando por dirección MAC (poco común hoy en día).

Cuando un switch es partido en varias VLAN, este las aísla, impidiendo que una estación en una VLAN se comunique con otra estación en una VLAN distinta. En teoría no debe permitir ningún tipo de tráfico entre ellas, ya que pertenecer a distintas VLAN debería ser equivalente a estar conectados a dispositivos diferentes, o sea a estar en distintas LAN. Si se deseara establecer algún tipo de comunicación entre VLAN, esta debería realizarse en otro nivel (siendo el nivel de red el más usado para esta operación).

Trunking

Si la división en VLAN debe abarcar a más de un switch, la situación cambia radicalmente. Normalmente tener múltiples VLAN abarcando varios switches implicaría cruzar un cable por cada VLAN entre cada par de switches que se desee conectar. Puesto que esto no es práctico, el IEEE estandarizó 802.1q. Este estandar define una forma de indicar la VLAN a la que pertenece un frame. Para usar este estandar, primero se deben configurar las VLAN en todos los switches, de forma tal que el mismo número de VLAN (que va entre 1 y 4095) coincida para cada una de ellas. Una vez hecho esto, se pueden tender los enlaces entre switches, indicando que los frames que viajen por ellos deben ir marcados usando el protocolo 802.1q. Estos enlaces son llamados “trunks”.

En un trunk, los paqutes deben llevar indicado a que VLAN pertenecen. Para ello, el frame es modificado de acuerdo con el estandar 802.1q. La modificación consiste en insertar cuatro bytes en el frame, inmediatamente antes del ethertype. Esta inserción consiste en un nuevo ethertype (0x0810), y dos bytes que llevan datos de priorización, flags y (lo más importante de todo) un número de VLAN de 12 bits. Cuando un frame de cierta VLAN debe ser enviado a otro switch pasando por un trunk, el switch debe alterar el frame para agregar estos valores. Una vez realizadas las alteraciones, el switch puede despacharlo al otro switch, que sabrá a que VLAN pertenece ese frame. Una vez que el frame es recibido por el switch y este deba pasarlo al destino, deberá deshacer los cambios realizados al entrar al trunk para luego enviarlo al destino final.

Es interesante notar que un frame en un trunk es compatible con un switch que no entienda de trunks 802.1q. Para ese tipo de switches, se trata de un frame ethernet normal, con un ethertype 0x0810. Obviamente este tipo de switches no podrá desmarcar los frames, por lo que no puede entregarlo a destino, pero perfectamente puede realizar el forwardeo entre dos switches que entiendan trunking.

Usar trunking tiene un impacto en la operación normal de un switch. En primer lugar, la posibilidad de hacer cut-through cuando se desea pasar de un enlace no marcado a un trunk se reduce drásticamente, ya que el switch debe alterar el frame. Por otra parte, se abren varias opciones en el uso de spanning tree.

Puesto que múltiples VLAN pueden tener distintas concentraciones de hosts en distintos switches, es necesario decidir una estrategia de spanning tree a usar. Una opción es usar un árbol para todas las VLAN (common spanning tree o CST), lo que es barato con respecto a los recursos usados (memoria, cpu y overhead del canal), pero que puede causar que la organización de algunas VLAN sea menos que óptima. Otra opción consiste en crear un árbol para cada VLAN (per VLAN spanning tree o PVST). Esta opción permite la mayor flexibilidad pero es extremadamente costosa, ya que las operaciones de spanning tree deben repetirse una vez por cada VLAN, lo que multiplica el uso de memoria, cpu y el uso del canal dedicado al envío de BPDU. Una tercera opción es crear varios árboles y asignar VLAN a cada árbol administrativamente (multiple spanning tree o MST). Aunque esto haga uso eficiente de los recursos, requiere hardware que comprenda MST, y la administración se complica debido a que hay que seleccionar manualmente las asignaciones.

Etherchannel

En algunos casos se puede querer habilitar múltiples enlaces entre dos switches para mejorar la performance entre ellos, por ejemplo cuando ese tramo es muy utilizado. Según las reglas impuestas por spanning tree, todos los enlaces menos uno deberían ser bloqueados, lo que no mejoraría la performance. Para poder aprovechar este tipo de conexiones, se pueden vincular estos enlaces usando IEEE 802.3ad (también conocido como Etherchannel).

Existen ciertas limitaciones a las que está sujeto un etherchannel. Todos sus enlaces deberían ser idénticos, y es posible que se forme un cuello de botella por falta de ancho de banda en el etherchannel o en las estaciones que ingresan datos a él. Aparte de estas restricciones obvias, también hay que tener en cuenta que el nivel de enlace no debería desordenar. Para garantizar el orden entre los frames de cada par de estaciones, la selección del cable a usar se basa en la dirección MAC de origen y de destino, en donde todos los frames con mismo origen y destino deberán pasar por el mismo cable del etherchannel. Esto agrega una limitación importante: es necesario tener una cantidad suficiente de estaciones de forma que el tráfico se envíe por distintos canales.