Acceso Compartido y Lan Switching

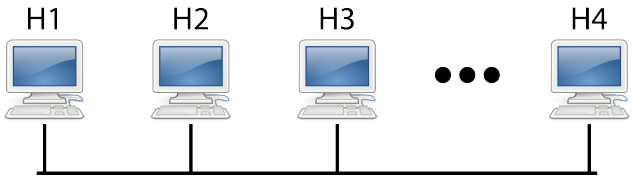
Resolución de ejercicios Práctica 3

05.04.2017

1. Primer ejercicio

1.1. Enunciado

En la siguiente LAN IEEE 802.3, los hosts H2 y H3 comparten un mismo segmento de 500 metros de cable, el host H4 está a 2500 metros de H1, pasando por 4 Hubs, y el Delay máximo es de 25.6µs.



a. ¿Cuál es el período de tiempo mínimo que deberá transcurrir para que las estaciones que enviaron un paquete se aseguren de que no ocurrió una colisión?

b. Calcule el tamaño mínimo del frame.

c. ¿Qué pasa si un emisor desea transmitir una cantidad de datos menor al mínimo especificado por la norma?

   En el momento t_0, H1 recibe en su buffer un dato para ser enviado por el enlace. Luego de sensar el medio, lo encuentra vacío y envía un paquete, ocupándolo por 10 ms.

d. Indique qué sucedería si en los momentos t_0+5ms y t_0+7ms los hosts H2 y H3 reciben en sus respectivos buffers, proveniente de la capa superior, datos para ser enviados por el enlace.

e. Indique qué sucedería si en el momento t_0+2µs el host H4 recibe en su buffer datos para ser enviados por el enlace.

1.2. Resolución

a. Recordemos que la administración de las colisiones en 802.3 es responsabilidad de cada host. A groso modo, la forma concreta de identificar que una trama colisionó es sensando el medio e identificando valores de la señal distintos a los inyectados. Como la consigna nos pide buscar el tiempo mínimo para el caso general dentro de cualquier segmento 802.3, lo que necesitamos es identificar el peor caso de temporal colsión, o sea, cuando dos hosts están lo más alejados posible.

   O sea que buscamos el caso donde haya colisión entre h_1 y h_2 siendo $distancia(h_1, h_2) = 2500m$.

Empezamos desarrollando el caso general y después lo instanciamos en el peor caso. Sabemos por definición del protocolo que:

i. Si un host sensa el medio y este está *ocupado* no transmite.

ii. Para identificar una colisión el host debe estar transmitiendo.

$\Rightarrow$

Sean $Prop_{h_1,h_2}$ el t_{prop} entre los hosts h_1 y h_2 y MAX el tiempo de propagación máximo del segmento

$\Rightarrow$

$\forall h_1, h_2 : hosts$,

Por i sabemos que si h_1 transmite una trama x en el tiempo t_0, a partir de $t_1 = t_0 + Prop_{h_1,h_2}$, h_2 no va va a transmitir hasta que termine de recibir x.

$\Rightarrow$

En $t_0 + Prop_{h_1,h_2}$ no hay colisión, pero en cualquier instante anterior h_2 podría comenzar a transmitir porque sensa el canal y lo encuentra disponible.

Como la propiedad debe valer para todo h1 y h2, buscamos el peor caso temporal $\Rightarrow$ desarrollamos para el caso en el cual h1 y h2 están a distancia máxima $\Rightarrow$ $Prop_{h_1,h_2} = MAX$

$\Rightarrow$

Peor caso: $t_{colision} = t_0 + MAX - t_j$ siendo t_j tan chico como sea posible.

$t_{colision}$ determina el tiempo máximo en que la colisión puede comenzar. Sin embargo la colisión se genera "muy cerca" de h_2. Recordemos que la consigna nos pide el tiempo mínimo hasta que el que envío (h_1) percibe la colisión $\Rightarrow$ resta asegurarnos que la colisión se propage hasta h_1.

Por ii sabemos que para detectar la colisión h_1 debe estar transmitiendo todavía su trama. Como la colisión es con h_2 y h_2 está a distancia máxima: en $t_{colision} + MAX$ estamos seguros que h_1 se enterá que colisionó y además sabemos que es el peor caso.

$\Rightarrow t_{colision} + MAX = t_0 + MAX - t_j + MAX = t_0 + 2MAX - t_j$

$\Rightarrow$ Dado que t_j lo definimos como un instante de tiempo que tiende a cero, el tiempo mínimo que debe transcurrir para asegurarnos que cualquier host detecta cualquier posible colisión en las condiciones de una ethernet 802.3 es + $t_{min} = 2 * MAX = 2 * 25{,}6\mu s = 51{,}2\mu s$

$\Rightarrow$

Debo transmitir durante 51,2µs para asegurarme que puedo detectar cualquier colisión en 802.3.

b. Como tengo $V_{tx} = 10mbps$, por regla de tres simple sabemos que en 51,2µs transmitimos $512b = 64B$.

c. Se rellena con padding. Hay dos opciones para luego descartar el padding:

- En el header: en lugar de usar el campo type para multiplexar se usa como length tamaño y se usa LLC como multiplexador para la capa de red.
- Se encarga la capa superior (por ejemplo IP *length*).

d. Ambos sensan el medio en los respectivos momentos temporales e identifican que el medio está siendo utilizado. Ergo, esperan (1-persistente) hasta que el medio esté libre y luego transmiten. Ambos encontrarán el medio libre en momentos muy cercanos (la diferencia está dada por la distancia entre H2 y H3 con H1) y sus tramas colisionarán.

e. H4 sensa el medio. Probablemente lo encuentre libre (por ejemplo si todavía no llega a sensar la información de la trama de H1 por estar a una distancia considerable), transmita y su trama colisione con la de H1.

2. Segundo ejercicio

2.1. Enunciado

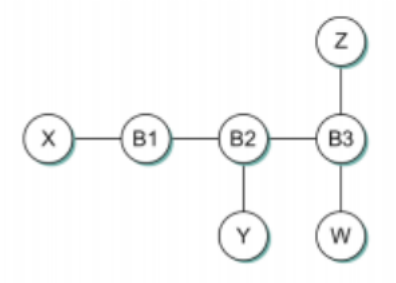
Dada la siguiente LAN se pide:

a. Si X transmite una trama con destino W. Qué bridges aprenden dónde está X? La interfaz de Y ve la trama?

b. Si luego Z transmite una trama con destino X. Qué bridges aprenden dónde está Z? La interfaz de Y ve la trama?

c. Si luego Y transmite una trama con destino X. Qué bridges aprenden dónde está Y? La interfaz de Z ve la trama?

d. Si finalmente W transmite una trama con destino Y. Qué bridges aprenden dónde está W? La interfaz de Z ve la trama?



2.2. Resolución

	X	W	Y	Z
B1				
B2				
B3				

a. Como todas las tablas inicialmente están vacias cada bridge que recibe la trama debe *floodearla*.

⇒

B1 aprende que X está en su puerto izquierdo.

B2 aprende que X está en su puerto izquierdo.

B3 aprende que X está en su puerto izquierdo.

Además como la trama llega a B2 y B2 debe *floodearla* la interfaz de Y ve la trama.

	X	W	Y	Z
B1	izq			
B2	izq			
B3	izq			

b.

La trama llega a B3 que sabe dónde está X. B3 aprende dónde está Z.

B3 lo manda a B2 solamente que sabe dónde está X. B2 aprende dónde está Z.

B2 lo manda a B1 solamente que sabe dónde está X. B1 aprende dónde está Z.

La interfaz de Y no ve la trama porque B2 ya aprendió que X está por el puerto que va hacia B1.

	X	W	Y	Z
B1	izq			der
B2	izq			der
B3	izq			arr

c.

La trama llega a B2 que sabe dónde está X. B2 aprende dónde está Y.

B2 lo manda a B1 solamente que sabe dónde está X. B1 aprende dónde está Y.

La interfaz de Z no ve la trama porque B2 ya aprendió que X está por el puerto que va hacia B1.

	X	W	Y	Z
B1	izq		der	der
B2	izq		ab	der
B3	izq			arr

d.

La trama llega a B3 que no sabe dónde está Y. B3 aprende dónde está W y *floodea*.

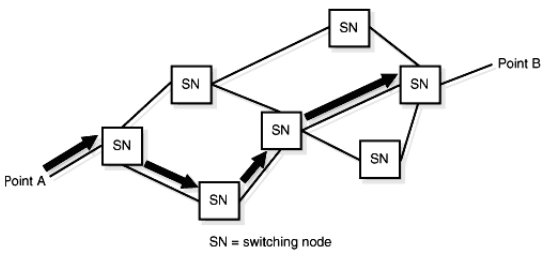
B2 lo manda solo a Y. Además aprende dónde está W.

La interfaz de Z ve la trama porque B3 no sabe dónde está Y.

	X	W	Y	Z
B1	izq		der	der
B2	izq	der	ab	der
B3	izq	ab		arr

3. Tercer ejercicio

La siguiente figura representa una topología de red en la que los *switches* corren el protocolo STP:



Se pide:

a. Elija los IDs para los switchs de modo tal que una trama siga el camino marcado en la figura. Justifique.

b. Indique y justifique el estado final de cada puerto de cada switch.

c. Asumiendo que:

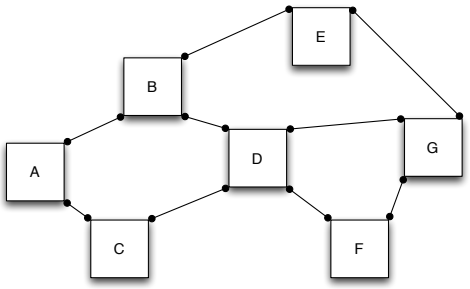
   1. Los switchs usan *Learning Bridge*.
   2. El protocolo STP ya convergió.
   3. Al momento de iniciar la transmisión la tabla de *forwarding* de cada switch está vacía.

   ¿Aprenden todos los switches la dirección de enlace de A la primera vez que se envía una trama desde A hacia B? Justifique.

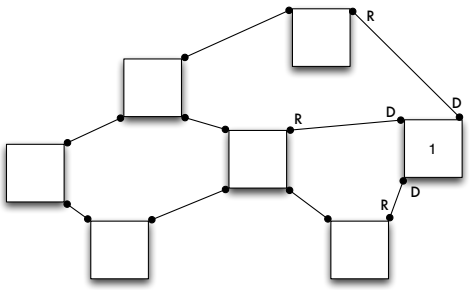
3.1. Resolución

a. Hay muchas maneras de resolverlo. Lo esencial es primero tener claro qué se busca. En este caso la única manera de que la trama siga el camino requerido es que los puertos correspondientes de cada uno de los bridges estén en estado **designado** o **root**. Si alguno de esos puertos luego de correr STP queda en estado **bloqueado** podemos estar seguros que la trama no recorrerá ese camino.

   Comenzamos nombrado los bridges para facilitar el seguimiento:

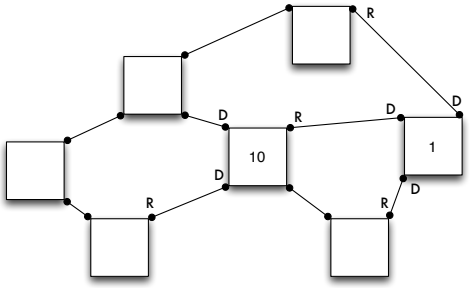


   Una posible forma de comenzar a resolver este ejercicio es asignandole un identificador al destino. Si le asignamos al bridge G el menor identificador, nos aseguramos que va a ser el root y que además todos los bridges directamente conectados van a tener su puerto con dirrección a G como root port. Por lo tanto, si la trama llega hasta D por el camino indicado, seguro seguirá camino a G por el enlace indicado también.

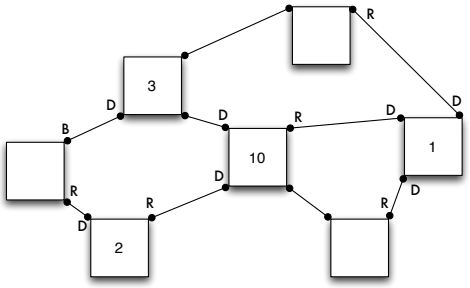


En un segundo paso podemos ver que por estar a distancia 1 del root, los dos puertos del bridge D que lo conectan tanto con B como con C estarán con estado designados. Esto se debe a que tanto B como C tienen una distancia al root = 2 mientras que D tiene distancia al root = 1. Por ende, pareciera que D puede tener cualquier identificador porque de todas maneras no va a incidir en los puertos que a nosotros nos interesan.

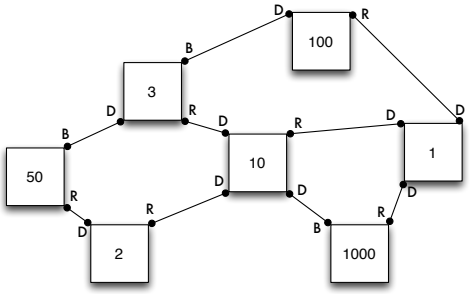
A su vez, esto implica directamente que C debe tener su puerto conectándolo a D como root port. Esto es porque el puerto de C que se conecta con D tiene menor distancia al root (2 saltos contra, al menos 4, para el otro puerto).



Lo único que resta es asegurarse que desde A, la trama se *fowardee* hacia C y no hacia B. Seguro que en el enlace entre A y C el puerto designado va a ser el de C porque está más cerca del root. Lo única posibilidad que resta entonces para seguir el camino indicado es que el root port de A sea el puerto que conecta con C. La competencia es solamente contra el otro puerto de A que conecta con B ya que sí o sí uno de los dos tiene que ser root port. Como la distancia por cualquiera de los dos puertos es 3 desempata el id del bridge. Por ende, para que la trama siga el camino requerido (por C) alcanza con que el id de C sea menor que el id de B.



b. Con el paso a) nos aseguramos el recorrido de la trama sin importar los datos de los demas puertos/bridges. Ahora completamos con ids a discreción y determinamos el estado de los puertos restantes.



c. Sí y hay que tener claro que esto es por definición. El protoclo STP nos asegura una topología lógica con forma de arbol donde **todos** los bridges están conectados. Si todavía no se conoce el destino (por ejemplo porque todas las tablas están vacías como en este caso) por definición la trama debe llegar a todos los bridges. Si no llegara no sería un arbol generador mínimo y la red podría perder conectividad. El caso de perdida de conectividad se ve claro cuando el nodo destino está directamente conectado a algún bridge al cual no le llegó la trama.