Rapid Spanning Tree Protocol Shortest Path, extensión a RSTP

Erwin Oñate, *Universidad Técnica Federico Santa María*

***Abstract*— Spanning Tree Protocol (STP) es por excelencia el protocolo más usado para la eliminación lógica de líneas redundantes en las infraestructuras utilizadas en la red. A medida aumentan los sectores integrados a la red las topologías se han vuelto más complejas y ha sido fundamental la optimización de los algoritmos de forma tal que la experiencia de usuario sea placentera y confiable. De esta forma en el año 2004 STP fue remplazado por RSTP (Rapid Spanning Tree Protocol) el cual entregaba un tiempo de convergencia muy superior a su antecesor. Este año se presentó una propuesta de extensión a RSTP donde se dimensiona el nivel de inundación de la red cuando se debe reparar el árbol utilizado. Como extensión al paper de Bonada y Sala [1] se estudió la cantidad de saltos que se deben realizar en la comunicación de un nodo a otro.**

I. Introducción

A pesar de las nuevas tecnologías inalámbricas y su rápido crecimiento global, Ethernet sigue siendo la mejor opción para conformar la infraestructura de la red que utilizamos a diario, gracias a sus características como ser de bajo costo, entregar altas tasas de transferencia, baja complejidad y fácil mantenimiento. Con el tiempo la topología se volvió cada vez más compleja y comenzaron a surgir ciertos problemas. Los principales se muestran a continuación:

- **Tormentas de broadcast:** los broascast enviados una y otra vez permanecen circulando sin fin, ya que en Ethernet a diferencia de IP no existe un TTL.
- **Múltiples copias de una trama:** existe la posibilidad de que a un host le lleguen tramas repetidas.
- **Tabla MAC inconsistente:** una trama de una cierta MAC podría llegar desde distintas rutas en la red

Si bien, los bucles físicos frecuentemente son un beneficio para la red ya que permiten crear rutas distintas para alcanzar ciertos nodos, es necesario eliminarlos de forma lógica. Es por ello que nacieron protocolos a nivel de capa 2 para poder solucionar estos problemas, como es el caso de Spanning tree protocol que utiliza reglas de minimización de costos para poder generar los arboles que muestran las rutas lógicas con las que se regirán las comunicaciones. Un nodo comienza sin necesidad de configuración y envía todas las tramas de datos recibidos a todos los puertos excepto el entrante. De esta recepción el puente aprende el puerto que conduce a la dirección de origen y las tramas siguientes de dirigirán solo a este puerto. Todas las tramas desconocidas se siguen enviando a todos los puertos. De esta forma en Ethernet se deben cumplir 2 aspectos importantes. Uno, la operación de emisión del puente solo funciona en redes sin bucles. Y dos, la operación de aprendizaje se realiza a partir de las tramas entrantes suponiendo que el camino que viene de un nodo es el mismo que la ruta para alcanzar tal nodo, esto es, la ruta debe ser simétrica.

Estas características son naturales en STP, es más difícil de obtener en las extensiones de ruta más corta como se verá en este documento. Sin embargo lograr mantener esta propiedad permite mantener compatibilidad hacia atrás con equipos existentes.

Existe la probabilidad de que un nodo o una cierta conexión entre nodos se caiga, pero eso no significa que se ha perdido del todo la comunicación, lo único que se debe hacer es recalcular el árbol lógico del protocolo y así activar puertos que antes estuvieron inactivos. RSTP extiende a STP logrando tiempos de convergencia, mucho menores a su antecesor minimizando el número de estados en los que pueden estar los puertos de los nodos.

Hasta aquí, estos algoritmos intentan generar buenas rutas en la red de forma tal de eliminar las redundancias de la red, pero no aseguran que todos los nodos tendrán el camino más corto hacia todo el resto de los nodos. Por esta razón se propone un nuevo algoritmo RSTP, el que si asegura el camino más corto entre todos los nodos, usando N instancias de RSTP para N nodos en la red.

II. Spanning Tree Protocol

A. Terminología básica

Para comprender el funcionamiento del STP es necesario conocer alguna terminología indispensable asociada al mismo.

Bridge ID: es el identificador de cada bridge. Es el resultado de combinar la prioridad del bridge con su dirección MAC base.

Root bridge (puente raíz): es el punto focal de la red y el que se toma como referencia para las decisiones del STP. El RB será aquel switch que tenga el menor bridge ID.

BPDU (Bridge Protocol Data Unit): son pequeñas unidades de datos que transportan información de control del STP. Se las utiliza en primera instancia para escoger el RB y luego para detectar posibles fallos en la red.

Bridges no raíz: son todos los demás bridges de la topología. Participan en el intercambio de BPDUs y actualizan a su vez su base de datos del STP.

Costo de un puerto: se determina en base al ancho de banda del enlace y será el valor que se utilice para decidir el camino más corto al RB.

Costo del camino al RB: el costo de un camino al RB es la suma de los costos de cada enlace por el que pasa. El camino elegido por el STP al RB será aquel cuyo costo sea más bajo.

Puerto raíz (designado): es el puerto de cada bridge que se encuentra en el camino mínimo al RB. Sólo hay uno por bridge que siempre estará en estado de forwarding.

Puerto no designado: todo puerto en un bridge con mayor costo que el puerto designado. Será puesto en estado de bloqueo.

B. Estado de los puertos

Cada puerto que participa del STP puede estar en uno de cinco estados. Estos son:

Bloqueado (BLK): No reenvía tramas de datos, aunque sí recibe y envía BPDUs. Es el estado por defecto de los puertos cuando un switch se enciende y su función es la de prevenir ciclos.

Escuchando (LST): Recibe, analiza y envía BPDUs para asegurarse que no existen bucles.

Aprendiendo (LRN): al igual que el estado LST, recibe, analiza y envía BPDUs, aunque aquí también comienza a armar la tabla CAM. En este estado aún no se renvían tramas de datos.

Reenviando (FWD): Envía y recibe todas las tramas de datos. Los puertos designados al final del estado de LRN serán marcados como FWD.

Deshabilitado: Es un puerto deshabilitado administrativamente y que no participará en el STP. Para el STP un puerto en este estado es como si no existiera.

C. Operación STP

El protocolo de STP cumple con una serie de pasos antes de alcanzar el estado estable y comenzar a enviar tramas de datos. Los mismos son los que se listan a continuación.

1. Escoger el RB:
    a. Se elige el bridge con prioridad más baja.
    b. Si uno o más switches tienen la prioridad más baja se elige entre ellos el que posea la MAC base más baja.
2. Se eligen los puertos raíz: cada bridge encuentra el menor camino hasta el RB y, con él, su puerto designado.
3. Cada uno de los bridges escucha BPDUs en todos sus puertos y, si detecta algún bucle en un puerto, lo bloquea. De lo contrario lo pone en estado FWD. El criterio para decidir qué puerto bloquear en un switch es el siguiente:
    a. Si debe escogerse un puerto entre dos switches diferentes se elige para bloquear el de aquel switch con el mayor bridge ID.
    b. Si debe escogerse un puerto dentro del mismo switch entonces se escoge aquel que tenga el mayor costo. En caso de coincidir el costo, el puerto que se bloquea es aquel que tenga el identificador más alto.

III. RAPID SPANNING TREE PROTOCOL

Corresponde a una evolución de Spanning Tree Protocol, reemplazándolo en la edición 2004 de 802.1d. La idea central es reducir significativamente el tiempo de convergencia de la topología de la red cuando ocurre un cambio en la topología.
Estados de los puertos RSTP:

Aprendiendo: Escucha BPDUs y guarda información relevante.
Reenviando: Una vez ejecutado el algoritmo para evitar bucles, los puertos activos pasan a este estado.
Descartado: No recibe BPDUs por lo cual no se encuentra participando en la instancia activa de STP

El protocolo construye el árbol por el intercambio y la actualización de estado del nodo. La información de estado se almacena a nivel de puerto y a nivel del puente. El Estado del puerto describe la distancia para llegar a la raíz desde el puerto correspondiente y el estado puente describe la distancia para llegar a la raíz desde el puente a través del puerto con un recorrido más corto. Estos estados son almacenados en forma de Vectores prioritarios que incluyen: la identificación de la raíz (Root), el costo (Cost), la ID del puente que posee el vector (Bridge), y la ID del puerto al que pertenece el vector (Puerto). El vector completo se puede apreciar correctamente en la Tabla 1.

Tabla 1: Vector de Prioridad RSTP

Vector de prioridad en RSTP	
Root(R)	BridgeID de la raíz
Cost(C)	Costo a la raíz
PathArray(PA)	Arreglo de BridgeID desde la raíz en el Vector
Port(P)	PortID del puerto del Vector

IV. SIMETRÌA

En el ejemplo de la figura 1 (a), el tráfico generado en B0 es remitido a B1 utilizando el árbol con raíz B0 (T0). Del mismo modo en 1(b), el tráfico de B1 a B0 utiliza el árbol T1. Puesto que los requisitos de simetría se deben cumplir, la rama de T0 de B0 a B1 debe ser la misma que la rama en T1 de B1 a B0. En este ejemplo, los árboles T0 y T1 no son simétricos y el aprendizaje y funciones de envío no funcionaría correctamente.

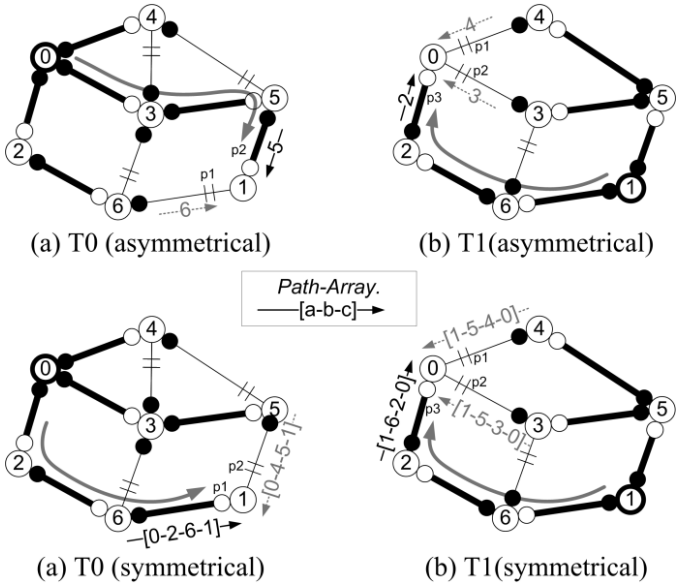

Figura 1: dos arboles asimétricos (a,b) convertidos en dos arboles simétricos.

El problema surge porque hay existen multiples caminos más cortos (recordar que se usa el mismo costo para cada una de las conexiones) entre el par de nodos (0-3-5-1, 0-2-6-1 y 0-4-5-1) y el algoritmo para construir los árboles decide por la primera trayectoria en T0 y para el segundo en T1. Una implementación común del algoritmo podría conducir a árboles no simétricos debido a que el camino más corto elegido es el siguiente salto inmediato que tiene el menor identificador. En el ejemplo de la figura 1(a), B1 selecciona B5 n lugar de B6 porque 5 <6. Del mismo modo en 1(b), B0 selecciona 2 porque 2<3<4. El camino entre B0 y B1 no es simétrico, puesto que la selección de la ruta depende de esta información local, lo que resulta en diferentes decisiones en diferentes puntos de la red.

Para solucionarlo se da el siguiente enfoque. La ruta de acceso de matriz de cada caminos más corto es primero ordenado de menor a mayor, y entonces los elementos se comparan uno por uno. Si el elemento es el mismo en ambos caminos, el siguiente elemento se compara. De lo contrario, el camino de arreglo con el menor elemento se considera mejor y por lo tanto seleccionado. En el ejemplo de la figura 1 (c), B1 decide entre el camino de matriz en p2, 0-3-5-1, y el camino de matriz en p1, 0-2-6-1. Una vez clasificados, los arreglos 0-1-3-5 y 0-1-2-6 convertido, respectivamente. Este último se considera mejor, ya que tiene 2<3 en la tercera posición. Del mismo modo en la figura 1 (d), B0 decide entre 1-5-4-0 en p1, p2 y 1-5-3-0 en 1-6-2-0 en p3. Una vez clasificados, los arreglos son 0-1-4-5, 0-1-3-5 y 0-1-2-6; B0 a continuación, se selecciona p3 porque 2<3<4. Dado que los dos trazados seleccionados son el mismo, pero en dirección opuesta, el camino desde B0 a B1 es el mismo.

V. RAPID SPANNING TREE PROTOCOL SHORTEST PATH

A. *Descripción del protocolo*

RSTP-SP se ejecuta una instancia diferente de protocolo de árbol único para cada nodo. Las instancias son completamente independientes y se ejecutan en diferentes niveles. El uso de instancias de árbol paralelas requiere que cada árbol gestione sus propios mensajes y, en consecuencia, cada nodo necesita almacenar la información separada por árbol (un puente almacena un vector puente por árbol y un vector de puerto por árbol y puerto).Cada evento que desencadena una operación de protocolo se aplica a una de las instancias de árbol y la operación ejecutada utiliza las variables que pertenecen a ese árbol en particular. El algoritmo utilizado se muestra a continuación:

Inicialización de puente b
-Convertirse en administrador de árbol b
-Enviar BPDU a todos designados

Recepción de BPDU en puerto p (para el árbol t)
-Comparar Vectores:
if (recibido es mejor o transmisor es el padre)
Almacenar vector recibido en el puerto p
Reconfigurar árbol.
else if (recibido es PEOR)
Enviar BPDU al puerto p
else if (son iguales)
no hacer nada

BPDU Periódico
-Enviar BPDU a los puertos designados

Detección de falla en puerto p (todos los árboles)
-Copia vector puente a vector puerto de p
-Reconfigurar árbol

B. *Detección de fallos*

En RSTP, el nodo con el menor BridgeID es elegido como la única raíz del árbol. Si este nodo falla, el protocolo se recupera eligiendo el segundo nodo con el BridgeID más bajo como nueva referencia. Sin embargo, en RSTP-SP cada nodo es la raíz de su propio árbol. Por tanto, cuando la raíz muere, su árbol debe quedar inactivo. Un nodo al detectar tal situación se anuncia mediante el envío de BPDU con un costo infinito. Estas BPDU se procesan normalmente por los nodos de recepción y se ven como BPDU comunes con un coste muy grande. Tenga en cuenta que la emisión de los mensajes con costo infinito es una solución eficaz para difundir la situación de no conectividad sin cambiar el proceso de transformación

BPDU y mantener la posibilidad de activar automáticamente el árbol cuando la raíz se vuelve otra vez.

C. *Consecuencias de fallo del nodo*

Una recuperación de un fallo de nodo en RSTP-SP no difiere de RSTP en términos de funcionamiento del protocolo y el comportamiento. Si un nodo no raíz falla, la recuperación es tan rápido como en el caso de fallo de enlace único. Por otro lado, si la raíz de un árbol falla, Se experimenta un comportamiento de cuenta hasta el infinito dentro de la instancia del árbol de la raíz que falló [2]. Esto significa que cualquier fallo en el nodo lleva a contar a infinito en uno de los árboles y una recuperación rápida en el resto.

Sin embargo, si no hay ninguna comunicación de datos, no hay urgencia para volver a configurar el árbol ya que las comunicaciones de este nodo no pueden establecerse hasta que se recupere. Sin embargo, la red está muy afectada por este comportamiento en términos de sobrecarga de mensajes. El conteo a infinito dentro del árbol muerto genera BPDU que bucle alrededor de las cuales reducen la potencia de procesamiento de los nodos y la capacidad disponible de los enlaces.

Para resolver este comportamiento se introduce en RSTP-SP un mecanismo de confirmación [3] con el fin de evitar por completo el comportamiento de la cuenta hasta el infinito. Esto se basa en la verificación de la disponibilidad raíz antes de enviar información falsa potencial que activa la cuenta hasta infinito. La implementación de este mecanismo ciertamente evita el conteo a infinito, pero retrasa la recuperación de los escenarios de fallo de enlace único. Por lo tanto, hay un equilibrio entre (1) utilizando el mecanismo de confirmación y retrasar la recuperación de fallo de enlace único y (2) aceptar el efecto de la cuenta hasta el infinito en el árbol muerto con ningún tráfico de datos está remitiendo. LA versión con la confirmación se refiere a RSTP-SP-Conf.

VI. EVALUACIÓN

Se implementó SPB, RSTP y SP-RSTP-SP-Conf en el simulador ns-3. El mecanismo de detección de fallos modelado es la detección inmediata falla física. Sólo los mensajes BPDU son simuladas y no el tráfico de usuario se modela a menos que se indique lo contrario. Tomamos como referencia para el procesamiento y la transmisión BPDU retrasar el estudio [4] que supone un retraso de 1.33msec por mensaje.

La evaluación del desempeño se centra en el (tiempo de convergencia (CT) y la sobrecarga de mensajes (MO). CT se define como el tiempo entre el fallo y el último nodo reconfigurar el árbol. También medimos la TC en retrasos hop como una unidad de tiempo normalizado. Esto es, un CT de 5 lúpulo significa que el protocolo tiene 5 veces el retardo de salto a converger. MO se refiere a la cantidad de mensajes que los nodos tienen que intercambiar con el fin de recuperar el árbol. El número de mensajes observados se utiliza para evaluar la sobrecarga en términos de (1) la capacidad del enlace utilizado y (2) requiere potencia de procesamiento de nodo.

A. *Tiempo de Convergencia*

Se utilizaron topologías tipo grilla de 4x4 y 8x8, y se evaluó el rendimiento en tres escenarios: inicio de la red, fallo de enlace en el centro de la red, y el nodo también en el centro. Para cada escenario se ejecuta 100 simulaciones con BridgeIDs diferentes. La Figura 3 muestra el promedio de CT, con intervalos de confianza del 95%, para cada protocolo y escenario. En la inicialización, todos los protocolos realizan lo mismo porque todos se basan en la inundación información. En el caso de un fallo de enlace central, RSTP-SP y SPB también proporcionan un rendimiento similar, porque los mensajes que comuniquen se propagan desde la ubicación de fallo al nodo más lejos, y por lo tanto depende del diámetro también. La pequeña diferencia se debe a SPB necesita para propagar la ruta completa, mientras RSTP-SP sólo necesita volver a configurar las ramas que están afectadas por el fallo. El CT más grande observado en RSTP-SP-conf es debido al retardo introducido por el mecanismo de confirmación.

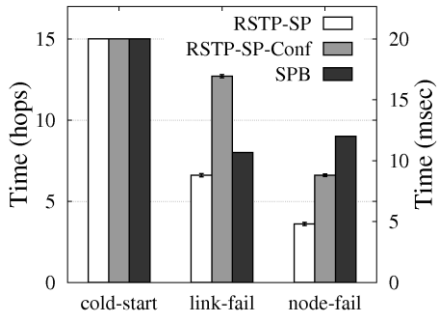

Figura 2: Tiempos de convergencia promedio (con intervalo de confiabilidad del 95%) para arranque, fallo con el nodo central y fallo en el link central

El último conjunto de columnas en la figura 2 muestra el CT observó después del fallo de un nodo central.
La Figura 3 muestra el promedio de CT con los percentiles 25% -75%. RSTP-SP y SPB funcionan de manera similar y RSTP-SP-Conf introduce el retardo de confirmación. También se realizaron experimentos con las topologías anteriores y variando el tamaño. Confirman que la CT en todos los protocolos y en todas las redes se relaciona con el diámetro de la red.

Particularmente, RSTP-SP-Conf toma más tiempo para proporcionar una conectividad total debido al retraso de confirmación. Por el contrario, SPB sufre un corte más alto debido a que la nueva información emitida durante la reconfiguración crea discordancias entre bases de datos de topología en diferentes nodos. La comunicación entre los nodos con diferentes bases de datos temporal es detenida a fin de evitar posibles bucles de reenvío.

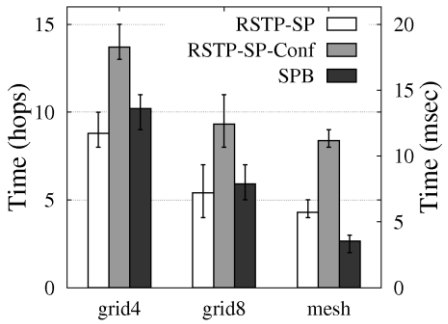

Figura 3: Tiempo de convergencia medio (en percentiles de 25%-75%) para fallo en distintas posiciones de la topología.

B. Gastos generales de mensaje

La Figura 4 muestra la sobrecarga de mensajes promedio por nodo medido en los distintos escenarios. Observe la escala logarítmica en el eje vertical. La sobrecarga en un arranque en frío-es similar en todos los protocolos porque todos se basan en la inundación de mensajes. Las diferencias aparecen cuando se comparan los protocolos en caso de fallas: SPB claramente supera RSTP-SP protocolo. La razón es que, por ejemplo, en las inundaciones de fallo de enlace SPB escenario sólo las actualizaciones de estado de enlace de los dos nodos de detección del fallo. En otras palabras, RSTP-SP reconfigura muchos árboles donde cada una BPDU problemas de actualización de los caminos. También tenga en cuenta que en los escenarios de caída de conexión del mecanismo de confirmación en RSTP-SP no representa una gran diferencia, ya que sólo introduce un retardo.

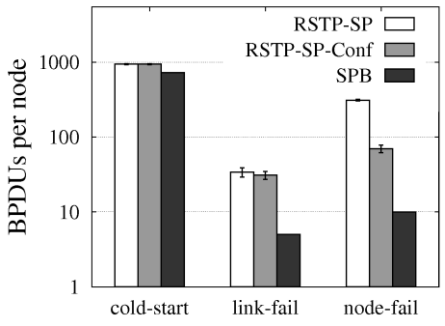

Figura 4: Sobrecarga de mensajes promedio para arranque, fallo en el nodo central y fallo en el link central.

Un comportamiento similar se observó en el caso de fallo de nodo. En este caso, la razón de la alta sobrecarga en RSTP-SP es el conteo a infinito que ocurre en el árbol de la raíz del error. Además, incluso si RSTP-SP-Conf permite eliminar este efecto, los mensajes transmitidos no se reducen al nivel de SPB debido a que estos se deben principalmente a las reconfiguraciones en los árboles donde un fallo no se produce.
Una comparación diferente es evaluar los protocolos en el estado de equilibrio una vez que el arranque en frío-se ha terminado y no hay fallos se han producido, como se aprecia en la figura 5. En esta situación, todos los protocolos de enviar mensajes de refresco para mantener la topología de la vida: los nodos SPB se aplican a nivel mundial y refrescante inundar todo su enlace-estado.

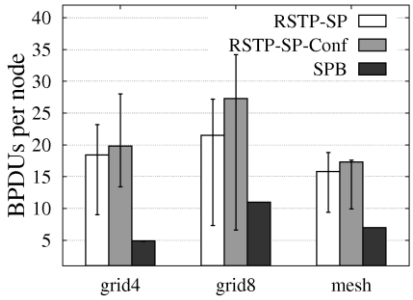

Figura 5: Sobrecarga de mensajes con fallo en distintas posiciones de la topología.

Una particularidad de los protocolos de estado de enlace es que cualquier fallo siempre conduce a una inundación de una nueva topología física, que da lugar a un nuevo cálculo de todos los árboles en todos los nodos. Por otro lado, un protocolo por vector de distancia sólo reconfigura los nodos afectados y árboles. Esto se puede observar en la figura 6 que muestra el porcentaje de nodos que reconfigurar una cierta cantidad de árboles afectados después de un fallo de enlace central. Por ejemplo, las cajas negras indican el porcentaje de nodos que se ven afectados en más de 75% de los árboles (por ejemplo, en una red de 4 nodos, la mitad de ellos se ven afectados en más de 3 árboles). Tenga en cuenta que cuando el tamaño de la red crece, el porcentaje de árboles afectados disminuye y la mayoría de los nodos sólo reconfigurar de menos de 25% de los árboles. Tenga en cuenta que SPB en esta trama siempre indicaría 100% de los árboles afectados en 100% de los nodos.

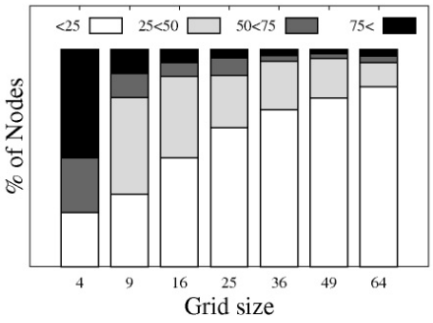

Figura 6: Porcentaje de nodos afectados por la recuperación del fallo de conexión.

VII. Extensión a las pruebas de Bonada y Sala

En el paper estudiado, la evaluación del algoritmo se basa en la comprobación de la inundación de la red usando dos criterios: tiempos de convergencia y sobrecarga de paquetes pero, ¿Qué sucede con los tiempos de transferencia de información entre nodos? La mejor forma de probarlo es la replicación del simulador que prepararon ellos, lo cual es bastante complejo y fuera de los plazos de tiempo que se poseían, por tanto, la mejor opción fue utilizar un simulado de RSTP provisto por Cisco y realizar pruebas estáticas,

simulando las el comportamiento de cada una de las instancias de árbol que deberían existir en RSTP-SP.

A. Metodología

El simulador de Cisco es de código abierto e implementado en C++, con librerías en C que pueden ser instaladas directamente sobre dispositivos de capa 2 de Cisco. Su funcionamiento es muy sencillo, basta con ejecutar un proceso manager para la implementación de conexiones y configuraciones generales y N instancias de un proceso bridge, para generar N nodos en la red. Se armaron configuraciones tipo grillas de 9 nodos (3x3) hasta 6x6 para la medición de saltos promedios para al comunicación de un nodo a otro y hasta 100 nodos para la medición de la máxima cantidad de saltos para el peor nodo de la red. La configuración de los se realizo en las condiciones más ideales posibles considerando el mismo costo para cada puerto y cada nodo de la red.

B. Resultados

Observando la figura 7 se puede apreciar que en RSTP el crecimiento de saltos promedios crece de forma lineal al igual que en RSTP-SP, pero este ultimo a una razón mucho menor que RSTP. Si pensamos que cada una de las conexiones permite la misma tasa de transferencia y que el retardo de procesamiento y tiempos de espera son iguales para cada nodo, implica que la recepción de paquetes desde que sale del nodo emisor hasta que llega al nodo receptor deberia ser proporcional a la cantidad de saltos que los paquetes realicen y en consecuencia usando RSTP llevaría menos tiempo que el actualmente usado RSTP.

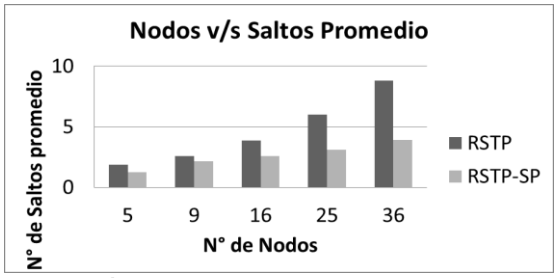

Figura 7: Número de saltos promedio para RSTP y RSTP-SP

La figura 8muestra la cantidad de saltos que debería realizar la peor comunicación que presenta la topología lógica de la red. En el caso de RSTP se produce entre las hojas más extremas en el arbol lógico. En el caso de RSTP-SP la comunicación que lleva más saltos corresponde a la comunicación realizada entre las esquinas de la grilla de la topología física que se encuentran opuestas (por ejemplo la esquina superior derecha con la esquina inferior izquierda). Como resultado note que nuevamente RSTP-SP va incrementando la cantidad de nodos a una razón más lenta que RSTP a medida que la cantidad de nodos implementados en la simulación de la red.

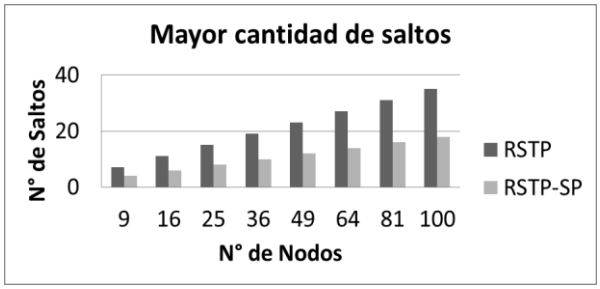

Figura 8: Número de saltos máximo para RSTP y RSTP-SP.

VIII. CONCLUSIÓN DE LA INVESTIGACIÓN

Basado en los resultados de la investigación de Bonada y Sala [1] y las nuevas pruebas realizadas para medir los saltos entre nodos, se puede ratificar que RSTP-SP es superior a su antecesor dada las condiciones necesarias, como sería tener una infraestructura robusta. De esta forma las perdidas de conexión serían mínimas en un cierto periodo de tiempo lo que asegura que las instancias de reconfiguración de los N arboles del algoritmo se producirían rara vez en una temporada. En consecuencia se evitaría la inundación de paquetes BPDU a la que se ve enfrentada la red y se ganaría mucho a nivel de carga sobre la red debido a la cantidad de paquetes en transferencia y las largas rutas que deben realizar para llegar a su destino.

Francamente este algoritmo sería de gran ayuda para la actualidad, dándole a Ethernet nuevo cimientos para que pueda seguir con tranquilidad su participación en la hoy fundamental internet.

References

[1] E.Bonada and D.Sala "RSTP-SP: Shortest Path Extension to RSTP".
[2] E.Bonada and D.Sala. "Characterizing the convergence time of RSTP"
[3] E.Bonada and D.Sala. "RSTP-Conf: efficiently avoiding count-to in finity in RSTP",".
[4] R.Pallos et al. "Performance of rapid spanning tree protocol in Access and metro networks".