Algoritmo para la Recuperación de Errores de una Transmisión Multipunto

Marta Barría
Departamento de Computación,
Universidad de Valparaíso
Avenida Gran Bretaña 1041, Valparaíso, Chile
Teléfono: +56-32-50-8307
Fax: +56-32-50-8301
y
Departamento de Informática PUC/ RJ , Brasil
e-mail: marta.barria@uv.cl
marta@inf.puc-rio.br

Reinaldo Vallejos
Departamento de Electrónica, UTFSM
Avenida España 1680, Valparaíso, Chile
Teléfono: +56-32-654-207
Fax: +56-32-797-469
e-mail: reinaldo@elo.utfsm.cl

L.F.G. Soares
Departamento de Informática PUC/ RJ , Brasil
Rua Marquês de São Vicente 225
22453-900 Rio de Janeiro, RJ
Teléfono: +55-21-529-9523
Fax: +55-21-511-5645
e-mail: lfgs@inf.puc-rio.br

Resumen

En este trabajo se presenta un nuevo algoritmo para la recuperación de errores (pérdida de paquetes) que ocurren en una transmisión multipunto. El algoritmo realiza la recuperación de errores en forma local, lo que significa que únicamente los nodos que se encuentran en la vecindad del error participan de la recuperación. Una consecuencia de la forma específica en que se realiza la recuperación de errores, es que el algoritmo propuesto presenta una latencia (tiempo necesario para recuperar un error) y una implosión (número de mensajes que recibe la fuente solicitando la retransmisión del paquete perdido) que compiten favorablemente con los algoritmos actuales. Esta afirmación se basa en la evaluación matemática del desempeño del algoritmo para una topología típica del árbol de distribución.

Palabras Claves: Comunicaciones Multipunto, Confiabilidad, Redes de Computadores

1. Introducción

Mbone [3] es la parte de la Internet pública que está conectada por IP Multipunto (multicast) y que permite enviar paquetes a los N (N>1) miembros receptores de un determinado grupo. La transmisión de información en Mbone es realizada en forma no confiable, es decir no se garantiza que los paquetes transmitidos lleguen sin error a sus destinos. Mbone está permitiendo el desarrollo de diversas aplicaciones multipunto [6] y por lo tanto está teniendo un gran impacto social. A grandes rasgos, las aplicaciones multipunto pueden ser divididas en aplicaciones que necesitan que la transmisión de la información sea confiable y aplicaciones que no necesitan de esta confiabilidad. Aplicaciones tales como videoconferencia, audio en Internet, eventos multimediales, kioskos electrónicos, etc. no requieren una transmisión confiable, ya que pueden aceptar un cierto nivel de pérdida de información sin que el usuario detecte un deterioro significativo en el servicio que recibe. Por otra parte, aplicaciones tales como distribución de software, pizarrones compartidos, juegos interactivos, transmisión de cuentas bancarias, replicación de bases de datos, etc., sí necesitan que la transmisión de la información sea confiable.

Para que las redes puedan ofrecer un servicio de comunicación multipunto confiable a las aplicaciones que lo requieren, es necesario desarrollar un protocolo que permita recuperar los paquetes que se pierden. Los mecanismos de recuperación de error están constituidos básicamente por una fase de detección y otra de corrección. La fase de detección normalmente se lleva a cabo en los receptores ya que resuelven esta tarea en forma más eficiente que cuando la ejecuta la fuente [7]. Para recuperar un paquete perdido los receptores envían al transmisor un paquete especial llamado NACK que le indica a la fuente que debe retransmitir el paquete perdido. Aunque el mecanismo de recuperación de errores recién descrito parece muy simple, en la práctica surgen algunas dificultades, ya que si cada uno de los receptores afectados por la pérdida de un paquete enviara un NACK a la fuente, en ella se produciría la llegada sincronizada de (posiblemente muchos) NACKs. Este fenómeno se conoce con el nombre de “implosión de NACKs”. La implosión de NACKs es indeseable porque produce congestión en la fuente y alrededor de ella, lo que obviamente causa degradación en las medidas de rendimiento globales de la red, tales como retardo y throughput [7]. El tamaño de una implosión es definido como el número de NACKs que recibe el transmisor (o fuente) debido a la pérdida de un paquete. Con el objeto de disminuir el tamaño de la implosión, el mecanismo diseñado para recuperarse del error necesita lograr que el sub-conjunto de receptores afectados por el error se pongan de acuerdo (de alguna manera) en quien (o quienes) enviará el NACK. Lamentablemente, lograr este acuerdo necesariamente consume tiempo, lo que a su vez aumenta la latencia que corresponde al tiempo total empleado en la recuperación del paquete perdido. En otras palabras, la disminución de la implosión tiende a causar un aumento en la latencia, lo que es indeseable.

El desempeño ideal que debería lograr un mecanismo de recuperación de paquetes perdidos, es que el tamaño de la implosión sea igual a 1 y la latencia sea mínima. Por este motivo, los mecanismos para la recuperación de errores que se han propuesto en la literatura buscan disminuir simultáneamente la implosión y la latencia, pues esto implica mejorar las medidas de desempeño globales de la red (retardo y throughput). Sin embargo, como estos objetivos son contradictorios entre sí, en la práctica los algoritmos propuestos intentan obtener un buen compromiso entre el tamaño de la implosión y la latencia.

En este trabajo se presenta un nuevo algoritmo de recuperación de errores que tiene algunos atributos que lo hacen competitivo con respecto a los algoritmos existentes. En particular, la recuperación de errores se hace en forma local, es decir, los únicos miembros del grupo que participan

de la recuperación del paquete perdido son los que se encuentran en la vecindad de donde se produce el error, consiguiendo un tamaño de implosión cercano al ideal y una baja latencia. Esta afirmación se basa en un análisis matemático del algoritmo propuesto, donde se evalúan la implosión, la latencia y la eficiencia del algoritmo (esta última medida será definida más adelante).

En lo restante, el artículo está organizado de la siguiente manera: en la sección 2 se define el modelo usado para representar el problema de confiabilidad en una comunicación multipunto; en la sección 3 se describe el algoritmo propuesto; la sección 4 contiene el análisis matemático del nuevo algoritmo usado en la evaluación de su rendimiento; finalmente, en la sección 5 se presenta las conclusiones del trabajo.

2. Modelo

La red es representada por un grafo $G=(V, E)$, donde $V=\{i, 1 \leq i \leq N\}$ corresponde al conjunto de nodos (o switches) de la red, N es el número total de nodos de la red y $E=\{e_{ij}, 1 \leq i,j \leq N; , i \neq j \}$ es el conjunto de arcos de estos nodos (los que representan los canales de comunicación de la red). Se supone un árbol de distribución [1, 2] de la información, el cual es generado por el protocolo de ruteamiento multipunto subyacente. La topología del árbol de distribución varía en el tiempo, ya que es posible que en cualquier instante nuevos miembros se integren al grupo o alguno de sus miembros lo abandone.

Los paquetes que se originan en una fuente son enviados en forma multipunto, a través del árbol de distribución, a todos sus descendientes. Cuando un paquete llega a un nodo del árbol, este nodo envía una copia del paquete a cada uno de sus hijos dentro del árbol, de esta manera los paquetes llegan a todos los miembros del grupo. Además, cada miembro del grupo mantiene una copia de la información que ha recibido correctamente, la que, en caso necesario, puede ser retransmitida a sus descendientes en el árbol multipunto. Los nodos que pertenecen al árbol de distribución, pero que no son miembros del grupo, sólo actúan como transmisores intermediarios de los paquetes que reciben.

Los receptores son los responsables por la confiabilidad de la comunicación. Esto significa que los receptores tienen la misión de detectar la pérdida de paquetes y solicitar la retransmisión de los paquetes perdidos (NACK). La pérdida de paquetes normalmente es detectada por los receptores al notar un salto en la secuencia de numeración de los paquetes que llegan a él desde una fuente determinada.

Para representar la pérdida de paquetes cada canal de la red tiene asociada una probabilidad de pérdida de paquetes de datos y una probabilidad de pérdida de NACKs, las que son diferentes debido fundamentalmente a que el tamaño de los paquetes de NACKs es menor que el de los paquetes de datos. Tanto las probabilidades de pérdida de paquetes de datos como de NACKs son homogéneas en el tiempo, lo que significa que dos paquetes del mismo tipo que son transmitidos por el mismo canal en tiempos diferentes, tienen la misma probabilidad de perderse debido a un error. Sin embargo las probabilidades de perdidas no son homogéneas en el espacio, ya que en general canales diferentes tienen asociadas probabilidades de error diferentes.

3. Algoritmo Propuesto

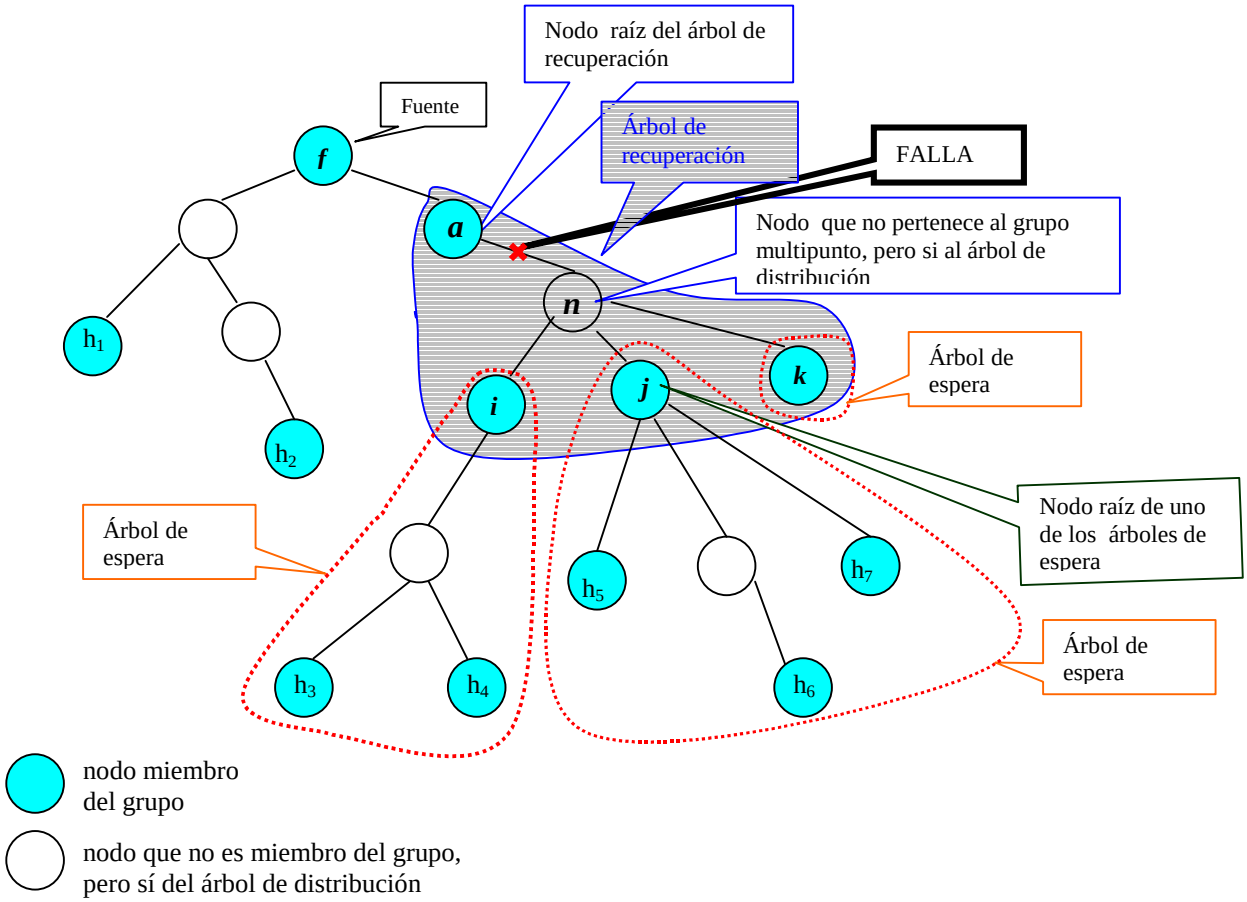

Figura 1: Árbol de distribución , árbol de recuperación y árboles de espera

A continuación se definen algunos conceptos que ayudarán a explicar la operación del algoritmo.

Se define como ***árbol de recuperación*** a cualquier sub-árbol del árbol de distribución donde solamente la raíz y las hojas son miembros del grupo. En caso que el árbol de recuperación posea nodos intermedios, estos no pueden ser miembros del grupo. De la definición, se deduce que un árbol de distribución puede tener muchos árboles de recuperación. Por ejemplo, el árbol de distribución de la Figura 1 posee 4 árboles de recuperación, que son: el árbol con raíz ***f*** y hojas h_1, h_2 y ***a***; el árbol con raíz ***a*** y hojas ***i, j*** y ***k***; el árbol con raíz ***i*** y hojas h_3 y h_4; y el árbol con raíz ***j*** y hojas h_5, h_6 y h_7. La definición de árbol de recuperación implica que un miembro del grupo que no es raíz ni hoja del árbol de distribución, es simultáneamente hoja de un determinado árbol de recuperación y raíz de otro árbol de recuperación. Por ejemplo, en la Figura 1 el nodo ***a*** es raíz del árbol de recuperación encerrado en línea continua y simultáneamente es hoja del árbol de recuperación cuya raíz es ***f.***

La filosofía general del algoritmo consiste en asociar la pérdida de un paquete al árbol de recuperación cuya raíz corresponde al último miembro del grupo no afectado por la pérdida, y luego responsabilizar de la recuperación del paquete perdido a los nodos de dicho árbol que fueron afectados por la pérdida. Por ejemplo, para el caso de la Figura 1, debido a que la falla ocurre en el enlace ***(a,n)***,

el árbol de recuperación asociado a esta falla es el que está encerrado en línea continua, y por lo tanto los únicos nodos que participan activamente en la recuperación del paquete perdido son los nodos ***a***, ***n***, ***i***, ***j*** y ***k***.

Cada hoja de un árbol de recuperación tiene asociado un (único) ***árbol de espera***. El árbol de espera asociado a un miembro del grupo (es decir, una hoja de un árbol de recuperación) es el sub-árbol del árbol de distribución cuya raíz es dicho miembro. El árbol de espera contiene además a todos los descendientes (en el árbol de distribución) del referido miembro. Por ejemplo, en la Figura 1 el árbol de espera asociado al miembro ***j*** es el árbol que está delimitado por una línea segmentada que contiene a ***j*** . De la definición anterior se desprende que cada árbol de recuperación tiene asociado un número de árboles de espera igual al número de sus hojas. Por ejemplo, para el árbol de distribución de la Figura 1, los 3 sub-árboles del árbol de distribución que están encerrados en línea segmentada, son los árboles de espera asociados al árbol de recuperación que está encerrado en línea continua.

La clave que permite al algoritmo obtener una alta eficiencia en la recuperación del paquete perdido, es que los nodos que no pertenecen al árbol de recuperación ni a algún árbol de espera no participan en la recuperación de la falla. Más aún, estos nodos en ningún momento se enteran de la ocurrencia de la falla. Además, los nodos que pertenecen a algún árbol de espera que no es afectado por la falla, tampoco participan en la recuperación del paquete perdido. Por ejemplo, si en la Figura 1 el error hubiese ocurrido entre los nodos ***n*** y ***j,*** los árboles de espera con raíces ***i*** y ***k*** no participarían de la recuperación del error. Los únicos nodos que participan en la recuperación del error son aquellos que pertenecen al árbol de recuperación y que son afectados por la falla. En particular, de los nodos de este árbol, el nodo raíz (nodo ***a*** en el ejemplo de la Figura 1) es el único miembro que contiene en memoria al paquete perdido, y en consecuencia este es el nodo encargado de retransmitir el paquete perdido. Respecto a las hojas del árbol de recuperación (nodos ***i, j***, ***k***, en la Figura 1), sólo aquéllas que se ven afectadas por la pérdida son las encargadas de solicitar la retransmisión del paquete perdido. En cuanto a los nodos que pertenecen a algún árbol de espera afectado por la pérdida y que no son hojas del árbol de recuperación, ellos participan *pasivamente* en la recuperación del error, de la siguiente manera: antes de que puedan detectar por sí mismos la pérdida del paquete, reciben un mensaje que les informa de la ocurrencia de la pérdida, a partir de ese momento estos nodos simplemente esperan que el paquete perdido les sea retransmitido.

Considerando las definiciones anteriores, a continuación se explica la operación del algoritmo de confiabilidad multipunto propuesto.

La operación normal de cualquier nodo del árbol de distribución consiste en transmitir en forma multipunto cada paquete que le llega de su padre (en el árbol de distribución) a todos sus descendientes. La primera vez que un nodo recibe un determinado paquete, además de transmitirlo a sus descendientes, guarda una copia de éste en su memoria. Esta copia sirve para retransmitir el paquete en caso que el miembro reciba una solicitud de retransmisión del paquete.

En el caso que un nodo miembro del grupo detecta la pérdida de un paquete, determina que debe actuar como ***hoja del árbol de recuperación*** de ese error y como ***raíz del árbol de espera*** que le corresponde. En consecuencia este nodo, que se denota ***nodo i***, realiza las siguientes actividades:

1. Transmite inmediatamente a todos sus nodos descendientes un mensaje multipunto indicándoles que ha detectado la pérdida del paquete. Este mensaje es tratado como un paquete normal por los descendientes de ***i,*** motivo por el cual se propaga a todos ellos. Al recibir el mensaje los miembros del grupo que son descendientes de *i* lo interpretan como que el nodo *i* se hará cargo de la

recuperación, y por lo tanto quedan a la espera de que les llegue el paquete perdido. En otras palabras, este es un ***mensaje de inhibición*** que informa de la detección de la pérdida y así impide que los miembros del grupo que lo reciben envíen su propio NACK.

Luego de transmitir el mensaje de inhibición, el nodo *i* continúa transmitiendo los demás paquetes que le llegan desde la fuente.

2. A continuación el nodo *i* decide si envía o no un NACK al nodo raíz del árbol de recuperación de la falla. Para determinar si envía o no el NACK, ***i*** efectúa un experimento aleatorio con distribución Bernoulli de parámetro p_1(i), que se denotará ***Be(p_1(i)).*** El parámetro $p_1(i)$ del experimento Bernoulli se escoge de la forma que se explicará más adelante. Si el resultado de este experimento es un éxito, entonces el nodo ***i*** envía el NACK hacia su padre en el árbol de distribución.

   La ejecución del experimento Bernoulli tiene el doble objetivo de disminuir la latencia y disminuir el tamaño de la implosión que se puede producir en el nodo raíz del árbol de recuperación. La latencia se disminuye porque el experimento Bernoulli se ejecuta en el mismo instante que se detecta la falla, y por lo tanto el resultado de este experimento se obtiene en forma inmediata. Por otra parte, la implosión se disminuye debido a que sólo algunas de las hojas del árbol de recuperación que efectúan el experimento Bernoulli obtienen un resultado que les hace transmitir el NACK. Más precisamente, el tamaño de la implosión es igual al número de hojas del árbol de recuperación afectadas por la falla que deciden enviar el NACK, que en la mayoría de los casos, es mucho menor que el número de miembros del grupo afectados por la falla. Según se verá más adelante, el algoritmo intenta lograr que el tamaño de la implosión sea aproximadamente 1, lo que se consigue escogiendo apropiadamente el parámetro $p_1(i)$ de la distribución Bernoulli.

3. Luego, independientemente de haber enviado o no el NACK, el nodo ***i*** inicia un "*timeout" (TO),* para esperar el paquete que falta. El *TO* debe ser igual o mayor al tiempo de propagación de ida y vuelta entre la hoja más alejada del árbol de recuperación y la raíz de este árbol, más el tiempo de retransmisión del paquete perdido.

   El *TO* es actualizado cada vez que una hoja del árbol de recuperación se integra o abandona el grupo multipunto. En estos casos, la hoja intercambia mensajes con la raíz del árbol de recuperación, lo que le permite a esta raíz determinar el valor necesario del *TO* e informarlo a las demás hojas del árbol de recuperación.

4. En caso que el *TO* expire sin que se haya recibido el paquete solicitado, el nodo ***i*** repite los puntos 2 y 3, pero esta vez el parámetro de la distribución Bernoulli que le sirve para decidir si envía o no un NACK tiene asociada una probabilidad $\boldsymbol{p_n(i)}$, donde ***n*** corresponde al número de veces que el nodo ***i*** ha ejecutado el paso 2 del algoritmo, respecto a la recuperación de un mismo error. En la sección siguiente se explica la forma en que se determina $p_n(i)$.

5. El nodo ***i*** repite el paso 4 hasta recibir el paquete perdido.

6. Cuando el nodo ***i*** recibe el paquete solicitado, lo retransmite en forma multipunto a todos sus descendientes. De esta forma los demás miembros del grupo que pertenecen al árbol de espera de ***i*** recuperan el paquete perdido sin haber participado activamente en su recuperación.

Cuando un nodo interno del árbol de recuperación recibe un NACK:

7. Retransmite el NACK hacia su nodo padre en el árbol de distribución.

El paso 7 del algoritmo asegura que los mensajes de NACK se propaguen hacia la raíz del árbol de recuperación.

Cada vez que un miembro del grupo, que con el objeto de simplificar la explicación denotaremos por ***a***, recibe un NACK respecto de un determinado paquete, realiza las siguientes acciones:

8. Elimina el mensaje NACK. Como resultado de esta operación, los nodos ancestros de ***a*** no se informan que ocurrió una falla, por lo cual ellos nunca intentan retransmitir el paquete perdido. Esto a su vez implica que no existe implosión de retransmisiones, ya que únicamente el nodo ***a*** es el encargado de retransmitir el paquete perdido.
   De los pasos 2, 7 y 8 de este algoritmo se desprende que la trayectoria de los mensajes de NACK va desde las hojas del árbol de recuperación afectadas por la falla (y que deciden enviar el NACK) hasta la raíz de este árbol.

9. Después de eliminar el mensaje de NACK, el nodo ***a*** retransmite el paquete solicitado en forma multipunto hacia todos sus descendientes. De esta forma todos los nodos que son raíz de algún árbol de espera reciben el paquete perdido.

Es interesante notar que la primera vez que un miembro ejecuta el paso 8 del algoritmo, detecta que se ha producido una falla en el ***árbol de recuperación*** del cual él es ***raíz*** y, en consecuencia es el encargado de retransmitir el paquete perdido (paso 9 del algoritmo). En las recepciones posteriores de NACKs respecto de un mismo paquete, el miembro simplemente ejecuta los pasos 8 y 9 del algoritmo.

Finalmente, cuando ***la raíz de un árbol de espera*** recibe la retransmisión del paquete perdido, procede de la siguiente forma:

10. Si fue afectado por la falla, entonces transmite el paquete perdido a todos sus descendientes. En caso contrario descarta el paquete que le llega.
    Nótese que, si la falla afectó al árbol de espera, entonces el paso 1 y 10 de este algoritmo permiten a todos los miembros del grupo pertenecientes al árbol de espera recuperar el paquete perdido.

3.1 Determinación de la probabilidad de enviar un NACK .

La filosofía del algoritmo consiste en hacer que la probabilidad con la cual cada nodo hoja del árbol de recuperación envía un NACK sea proporcional a la distancia entre este nodo y la raíz del árbol de recuperación. Esto implica que las hojas más alejadas de la raíz del árbol de recuperación tienen asociada una mayor probabilidad de enviar un NACK. El motivo de esta definición es que, cuando ocurre una falla, las hojas más alejadas del árbol de recuperación son las que tienen más probabilidad de ser afectadas por la falla. A continuación se describe la forma específica en que se determinan estas probabilidades.

Sea $p_1(i)$ la probabilidad de que la hoja i del árbol de recuperación envíe un NACK la primera vez que ejecuta el paso 2 del algoritmo respecto de una falla determinada. Sea α, con $\alpha>1$, un parámetro

que es usado para conseguir un equilibrio entre la latencia y la implosión. Entonces el valor de $p_1(i)$ está dado por:

$$p_1(i) = \frac{\alpha \quad f(c_i)}{f(A)} \tag{1}$$

Donde c_i es el camino que va desde la raíz del árbol de recuperación hasta una determinada hoja *i*; $f(c_i)$ corresponde a la probabilidad que un paquete enviado a través de c_i se pierda; *A* identifica al árbol de recuperación, y *f(A)* corresponde a la suma de la probabilidad de falla de todos los caminos que van desde la raíz de *A* a todas las hojas de *A*. Sea x_l la probabilidad de pérdida de paquete asociada al canal *l*, entonces usando el hecho de que los canales fallan en forma independiente, se tiene que $f(c_i)$ y $f(A)$ están dados por:

$$f(c_i) = \left(1 - \prod_{\forall l \in c_i} (1 - x_l)\right) \tag{2}$$

$$f(A) = \sum_{\forall\, hoja\ \ i \in A} f(c_i) \tag{3}$$

La atribución de probabilidades de (1) supone que el paquete perdido afecta a todas las hojas del árbol de recuperación (este es el caso ilustrado en el ejemplo de la Figura 1). Sin embargo, si el paquete no es recuperado la primera vez que se ejecutan el paso 2 del algoritmo, con alta probabilidad esto significa que la falla afectó solamente a un sub-conjunto propio de las hojas del árbol de recuperación (por ejemplo, en la Figura 1 la falla podría haber afectado solamente al árbol con raíz en ***i*** y no a los árboles con raíz en ***j*** o ***k)***. Por este motivo, y con el objeto de garantizar que en la segunda o posteriores ejecuciones del paso 2 del algoritmo (respecto de la misma falla), efectivamente al menos una hoja del árbol de recuperación envíe su NACK, es necesario aumentar la probabilidad de que las hojas de este árbol que fueron afectadas por la falla envíen un NACK. Una alternativa de lograr esto es aumentar exponencialmente la probabilidad de enviar su NACK. Por ejemplo si el árbol de distribución es r-ario, es razonable que la probabilidad usada en la n-ésima ejecución del paso 2 del algoritmo se aumente por un factor *r* con relación a la iteración previa, es decir:

$$p_n(i) = min\begin{cases} 1 \\ r \cdot p_{n-1}(i) \end{cases}, \qquad n > 1 \tag{4}$$

Donde *i* es una hoja del árbol de recuperación afectada por la falla.

4. Rendimiento del Algoritmo

Las dos medidas más usadas para caracterizar el comportamiento de los mecanismos de recuperación de errores son el tamaño de la implosión y la latencia. En consecuencia, a continuación se evalúa matemáticamente ambas medidas para el algoritmo propuesto.

Sea *I* la variable aleatoria que representa el tamaño de la implosión de NACKs y sea *E[I]* su valor medio. Además, se define *T* como la variable aleatoria que representa la latencia, y *E[T]* corresponde a su valor medio. Con el objeto de evaluar simultáneamente el desempeño del algoritmo con relación a la implosión y la latencia, en este artículo se propone una nueva métrica. Específicamente, definimos *E[P]* como el producto de los valores medios de la implosión y de la latencia, es decir:

$$E[P]=E[I]E[T] \tag{5}$$

Para explicar el sentido de *E[P]*, recordamos que la disminución de la implosión, normalmente causa un aumento en la latencia, y viceversa. Esto significa que, para evaluar el rendimiento de un algoritmo, no basta evaluar ambas medidas por separado. Por este motivo, *E[P]* corresponde a una combinación de *E[I]* y *E[T]*, de forma que *E[P]* aumenta cuando el aumento porcentual de *E[I]* es mayor que la disminución porcentual de *E[T]*, o viceversa. En otras palabras, si el deterioro de *E[I]* (*E[T]*) es mayor que la mejora de *E[T](E[I])*, entonces *E[P]* aumenta reflejando que globalmente el desempeño del algoritmo se deterioró.

Sin embargo, E[P] no es suficiente para cuantificar el rendimiento del algoritmo propiamente tal, ya que por ejemplo E[T] depende del valor de TO, lo que implica que la latencia (y por lo tanto E[P]) pueden variar significativamente al evaluar el rendimiento del mismo algoritmo en árboles de distribución diferentes.

Con el objeto de evaluar el rendimiento del algoritmo independientemente del árbol de distribución sobre el cual está operando, en primer lugar notamos que cuando ocurre una falla dentro de un determinado árbol de recuperación, un algoritmo ideal debería presentar una latencia igual al *timeout* (*TO*) de dicho árbol de recuperación y una implosión igual a 1. Esto implica que el valor ideal de *E[P] (E[*P_{ideal}*]*), debería ser igual al referido *TO*. Además, como *E[*P_{ideal}*]* constituye una referencia ideal para cualquier algoritmo, definimos la "eficiencia", ε, de un algoritmo" como:

$$\varepsilon = \frac{E\left[P_{ideal}\right]}{E\left[P\right]} = \frac{E\left[P_{ideal}\right]}{E\left[I\right]E\left[T\right]} \tag{6}$$

De la ecuación 6 se deduce que $0 \leq \varepsilon \leq 1$; además se puede afirmar que mientras mayor es la eficiencia de un algoritmo, más conveniente es éste.

El desempeño de un algoritmo de confiabilidad multipunto depende de muchos factores, entre los cuáles se encuentran la topología del árbol de distribución y cuales de los nodos de este árbol son miembros del grupo. Por este motivo, para evaluar el desempeño del algoritmo deberían analizarse diversas variantes en cuanto a la topología del árbol y la distribución de los miembros, dando énfasis a los casos más representativos de las situaciones que ocurren en la práctica. Se han realizado varios análisis, sin embargo, debido a la restricción de la extensión de artículo sólo se presenta el análisis para el árbol de distribución mostrado en la Figura 2. Se presenta esta topología debido a que es una topología típica de Mbone para redes regionales [4]. Además, el mismo caso fue analizado en [5] para evaluar el protocolo SRM, lo que permite comparar el desempeño del algoritmo propuesto en este trabajo con respecto de SRM, que es uno de los algoritmos de confiabilidad multipunto más conocidos.

Tal como se muestra en la Figura 2, el grupo multipunto está compuesto por R miembros, de los cuales R-1 son receptores y el otro es la fuente que transmite los paquetes. También existe un nodo intermedio, que no pertenece al grupo, el cual solo se encarga de transmitir la información a los miembros del grupo. Para simplificar el análisis se supone que la distancia que existe entre los nodos está normalizada y tiene valor 1. También se supone que no existe pérdida de NACKs ni de paquetes retransmitidos.

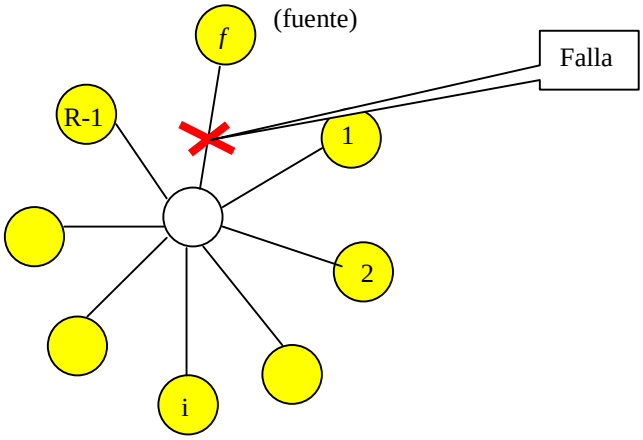

Figura 2: Topología estrella.

Para simplificar el análisis, y sin pérdida de generalidad, se supone que todos los canales de topología tienen la misma probabilidad x de que se pierda un paquete transmitido por ellos. Entonces la probabilidad $f(c_i)$ (ver ecuación 2), está dada por:

$$f(c_i) = x(2-x), \qquad 1 \le i \le R-1 \tag{7}$$

Luego, la probabilidad de pérdida del árbol de recuperación está dada por:

$$f(A) = \sum_{i=1}^{R-1} f(c_i) = (R-1)x(2-x) \tag{8}$$

Reemplazando (7) y (8) en (1), se tiene que la probabilidad de que una hoja envie un NACK en la primera iteración del algoritmo, está dada por:

$$p_1(i) = \frac{\alpha}{R-1}, \qquad 1 \le i \le R-1 \tag{9}$$

Debido a que $p_1(i)$ no depende de i, en las ecuaciones siguientes se omite el índice i que identifica una hoja específica, es decir: $p_1(i) = p_1$. Además, con el objeto de mantener simple el análisis, se supone que la probabilidad de que una hoja envíe un NACK en la n-ésima vez que ejecuta el paso 2 del algoritmo (respecto de un mismo paquete perdido) es igual a la probabilidad de la primera vez, o sea: $p_n = p_1 = p$.

A continuación se evalúa E[T]. Observe que E[T] es igual al número de intentos realizados antes de enviar el primer NACK multiplicado por el valor de TO del árbol de recuperación correspondiente. Como TO es conocido solo resta calcular el número de intentos realizados antes de enviar el primer NACK, el cual se evalúa a continuación. Usando el hecho de que en cada repetición del paso 2 del algoritmo (respecto del mismo error), una hoja del árbol de recuperación decide si envía o no un NACK en base a un experimento Bernoulli de parámetro p, se tiene que:

$$\Pr\{\textit{enviar NACK en la } n-\textit{ésima repetición del paso 2 del algoritmo}\} = (1-p)^{n-1}\, p$$

Además, como todas las hojas del árbol de recuperación actúan en forma independiente, la iteración en la cual se envía el primer NACK a la raíz del árbol de recuperación es una variable aleatoria (v.a.) con distribución $\min_{1 \le i \le R-1} \left(Ge(p(i))\right)$, lo que a su vez corresponde a una v.a. con

distribución $Ge\left(1-(1-p)^{R-1}\right)$. En consecuencia, *E[T]* corresponde al valor medio de la v.a. $Ge\left(1-(1-p)^{R-1}\right)$ multiplicado por TO, es decir:

$$E[T]=\frac{TO}{1-(1-p)^{R-1}} \tag{10}$$

Para evaluar el tamaño medio de la implosión, *E[I]*, se observa que ésta sólo ocurre cuando al menos uno de los receptores transmite el NACK. En este caso, considerando que los NACKs y los paquetes retransmitidos no se pierden, que el número de NACKs transmitidos tiene distribución binomial de parámetros *(R-1) y p*, y usando la definición del valor esperado condicional, se tiene que:

$$E[I \mid se\ transmite\ \ al\ menos\ \ 1\ NACK\ ]=\frac{\sum_{i=1}^{R-1} i\ \Pr[se\ transmiten\ \ i\ NACK\ ]}{1-\Pr[no\ transmitir\ \ \ NACK\ ]}=\frac{\alpha}{1-\left(1-\frac{\alpha}{R-1}\right)^{R-1}} \tag{11}$$

De (10) y (11) , se concluye que para este caso, la eficiencia está dada por:

$$\varepsilon\ =\ \frac{\left(1-\left(1-\frac{\alpha}{R\ -1}\right)^{(R-1)}\right)^{2}}{\alpha} \tag{12}$$

La Figura 3 muestra el valor óptimo de ε (que se obtiene para α=1.25) en función de R, tanto para el algoritmo propuesto como para el algoritmo SRM [5].

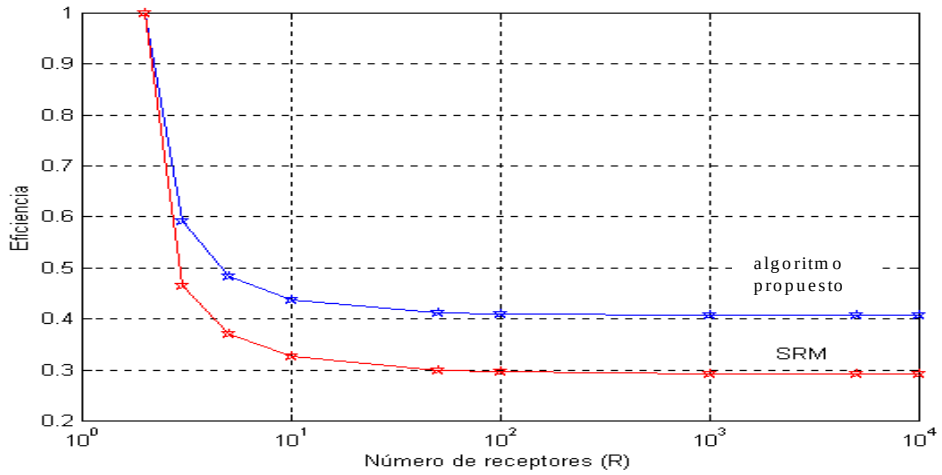

Figura 3: Eficiencia del algoritmo propuesto versus SRM.

En el gráfico de la Figura 3 se observa que, para cualquier número de receptores del grupo multipunto, la eficiencia del algoritmo propuesto supera claramente a la eficiencia de SRM. Es importante destacar que, a diferencia del algoritmo propuesto en este trabajo, en el algoritmo SRM todos los miembros participan de la recuperación del error. Sin embargo, en el ejemplo analizado

(mostrado en la figura 2), el árbol de distribución coincide con el árbol de recuperación lo que significa que todos los miembros del grupo participan de la recuperación, lo cual es el peor caso para el desempeño del algoritmo propuesto. A pesar de esto la eficiencia del algoritmo propuesto supera a la eficiencia de SRM. Lógicamente que en los casos en que el árbol de recuperación es mucho menor que el árbol de distribución, que corresponde a la generalidad de los casos, el algoritmo propuesto en este artículo mejora significativamente su desempeño respecto de un algoritmo con recuperación global.

5. Conclusiones

En este artículo se propuso un nuevo algoritmo para la recuperación de errores en una transmisión multipunto. En este algoritmo, la recuperación de errores es realizada solamente por los miembros del grupo vecinos a la falla. Estos nodos delimitan lo que hemos denominado árbol de recuperación.

A diferencia de los algoritmos existentes, el algoritmo propuesto le atribuye mayor probabilidad de solicitar la retransmisión de un paquete perdido a las hojas del árbol de recuperación más lejanas a la raíz de este árbol, pues son ellas quienes tienen mayor probabilidad de ser afectadas por la pérdida de un paquete.

El algoritmo propuesto consigue una baja latencia y una baja implosión en la recuperación de errores. Esto se debe a que la recuperación de errores es hecha en forma local y a la forma con que se determina la probabilidad de que a las hojas del árbol de recuperación soliciten la retransmisión del paquete perdido.

Hasta el momento, la implosión y la latencia han sido evaluadas en forma independiente. Sin embargo, es conocido que una disminución en la implosión normalmente causa un aumento en la latencia, y viceversa. Por este motivo, para evaluar el rendimiento del algoritmo respecto de ambas métricas simultáneamente, en el artículo se propuso una nueva medida de rendimiento, denominada eficiencia. La eficiencia es una medida relativa, por lo cual a diferencia de la latencia, la eficiencia es independiente del árbol en el cual se evalúa.

Agradecimientos

Este trabajo fue financiado por el proyecto Fondecyt Nº 1980416 / 1998. Los autores expresan su agradecimiento a José Manuel Martínez, Alejandra Zapata e Isabel Martin por la cuidadosa lectura de este artículo.

Bibliografía

[1] T. Ballardie, "*Core based tree (CBT) multicast: Architectural overview and specification*" Internet Draft RFC, July 1994.

[2] T. Bilhartz, J.B. Cain, D. Fieg and S.G. Batsell, "Performance and Resource Cost Comparisons for the CBT and PIM Multicast Routing Protocols*", IEEE JSAC*, 15(3), April 1997.

[3] H. Eriksson, "MBone. The Multicast Backbone", Communications of the ACM, vol. 37, Aug.1994,

[4] http://www.cs.columbia.edu/~hgs/internet /mbone-faq.html

[5] S. Floyd, V. Jacobson, C-G. Liu, S. McCanne and L. Zhang, "A Reliable Multicast Framework for Lightweight Sessions and Application- Level Framing", *IEEE/ACM Transactions on Networking*, 1997, An earlier version of this paper appeared in Proc. ACM SIGCOMM'95,pp 342-345, Oct. 1995.

[6] C. Kenneth Miller, "Multicast Networking and Applications", Addisson - Wesley, 1999.

[7] D. Towsley, J.Kurose and S. Pingali, " A Comparison of Sender-Initiated and Receiver-Initiated Reliable Multicast Protocols ", *IEEE Journal on Selected Areas in Communications*, April 1997.