Mecanismos para la Transmisión Multicast Confiable: una conceptualización

Marta Barría
Departamento de Computación
Universidad de Valparaíso
mbarria@uv.cl

Reinaldo Vallejos
Departamento de Electrónica
U.T.F.S.M.
reinaldo@elo.utfsm.cl

Resumen

En este trabajo se entrega una conceptualización sobre el problema de transmisión multipunto confiable. En esta conceptualización se identifican las diferentes etapas que es necesario realizar para solucionar el problema de detectar y recuperar los paquetes perdidos. Además, se describen algunos de los protocolos más importantes publicados hasta la fecha.

Palabras Claves: Redes de Computadoras, Telemática, Internet.

1. Introducción

Actualmente una de las áreas de mayor interés en la evolución de las telecomunicaciones es el desarrollo de diversas aplicaciones multipunto, debido al gran impacto que éstas están teniendo en la sociedad actual. Ejemplos iniciales de estas aplicaciones lo constituyen las teleconferencias, pizarrones compartidos, distribución de películas, diarios interactivos, etc.
Desde el punto de vista social es indudable que las aplicaciones multipunto son extremadamente útiles, debido al hecho de que la relación entre las personas se lleva a cabo principalmente en forma grupal y no persona a persona. A pesar de esto, debido a limitaciones tecnológicas, inicialmente los sistemas de telecomunicaciones sólo permitían ofrecer servicios interactivos punto a punto, pero afortunadamente en este momento estamos en una situación en la cual la tecnología ha evolucionado permitiendo el desarrollo de aplicaciones multipunto, que, como ya se dijo, es lo más natural para el ser humano. Los principales desarrollos tecnológicos que permitieron ofrecer servicios multipunto son la implementación de las redes de alta velocidad, el aumento de la velocidad de procesamiento de los computadores actuales y el gran éxito que ha tenido Internet Multicasting usando Mbone [Eriksson94a].

Muchas de las aplicaciones multipunto necesitan que el servicio multicast sea confiable, lo que significa que todos los paquetes trasmitidos deben ser recibidos sin error por todos los miembros del grupo (receptores). Evidentemente que la probabilidad de que un paquete se pierda (debido a la ocurrencia de algún error en la red) se puede disminuir usando diversos mecanismos. Entre ellos podemos mencionar: 1) la mejora de la calidad de los elementos de la red, por ejemplo el uso de fibra óptica disminuye la probabilidad de error en los canales. 2) También se usan técnicas de control de congestión, ya que éstas disminuyen la probabilidad de pérdida en los buffers. 3) Además se pueden usar códigos Forward Error Correction (FEC) [Blahut83a,Rizzo97a], ya que ellos aumentan la confiabilidad debido a que algunos de los paquetes que se pierden pueden ser recuperados sin tener que ser retransmitidos, lo que aumenta considerablemente la razón (throughput) en la transmisión confiable a grupos grandes [Huitema96a, Nonnenmacher97a, Nonnenmacher97b]. Sin embargo, debido a que los

grupos de usuarios de una aplicación multipunto pueden estar constituidos quizás por millones de receptores, la probabilidad de que ocurra un error en algún lugar de la red no es despreciable.
Para recuperar los paquetes que se pierden en una comunicación multipunto confiable se usan mecanismos de recuperación de errores, los cuales están constituidos básicamente por una fase de detección y otra de recuperación.
Debido a la gran cantidad de investigación en relación con el tema de la transmisión multipunto confiable que se ha realizado en los últimos años, no es fácil tener una visión clara del problema y las alternativas de solución posibles. Por este motivo es útil generar una conceptualización que identifique los grados de libertad del problema de detección y recuperación de errores y las distintas alternativas de solución de cada componente.
La contribución de este trabajo consiste en proponer esta conceptualización, describir algunos de los protocolos más importantes publicados hasta la fecha y clasificarlos de acuerdo a la taxonomía presentada.Por restricciones de espacio, no es posible incluir la clasificación de los protocolos según la taxonomía propuesta. Una versión completa de este trabajo puede obtenerse en: http://www.elo.utfsm.cl/acad/rvcmain.htm

El resto de este artículo está estructurado de la siguiente manera: en la sección 2 se presenta una taxonomía útil para clasificar los mecanismos de detección y recuperación de errores. En la sección 3 se describen algunos protocolos de transmisión confiable y se los clasifica según la taxonomía propuesta. Por último, en la sección 4 se entrega un resumen del artículo.

2. Taxonomía

En esta sección presentamos una taxonomía sobre los protocolos para transmisión multipunto confiable. Esta taxonomía es realizada de acuerdo a diferentes ejes de clasificación, que corresponden a las etapas que es necesario llevar a cabo para solucionar el problema de la detección y recuperación de errores en una transmisión multipunto confiable. En las subsecciones siguientes se explican en detalle estos ejes de clasificación.

2.1 ¿Quién detecta la pérdida de un paquete?

En las comunicaciones punto a punto se pueden distinguir básicamente las siguientes formas de detectar y corregir la pérdida de un paquete transmitido:

1. El receptor envía a la fuente reconocimientos positivos o ACKS.
2. El receptor envía a la fuente reconocimientos negativos o NACKS.
3. El receptor envía a la fuente una combinación de ACKs y NACKs
4. El receptor o el transmisor hacen uso de un reloj con tiempo límite (*timeout*)

En el esquema basado en el uso de ACKS, el receptor envía a la fuente un mensaje de aceptación (o ACK) por cada paquete recibido correctamente. Por su parte la fuente lleva la contabilidad de todos los paquetes transmitidos que no se ha confirmado su correcta recepción mediante un ACK. Si después de un cierto intervalo de tiempo límite (llamado *timeout*) el transmisor no recibe el ACK del paquete transmitido supone que el paquete se perdió y procede a retransmitirlo. En esta forma de recuperación de errores es el transmisor quién tiene la responsabilidad de mantener la confiabilidad de la transmisión, por este motivo es del tipo "sender-initiated" [Pingali94]. TCP es un protocolo de confiabilidad unicast (tal vez el más popular) de este tipo.

La principal limitación de esta clase de protocolos es que la fuente o transmisor necesita llevar la contabilidad respecto a todos los paquetes transmitidos y procesar todos los ACKs que recibe. Por este motivo, no es eficiente extender el uso de ACKs para el caso de comunicaciones multipunto, ya que en un ambiente multicast puede existir un gran número de receptores (por ejemplo 10^6), lo que significa que la fuente debería recibir los ACKs enviados por todos los receptores respecto de cada paquete transmitido. Esto produciría una enorme implosión de ACKs en el transmisor. Además, el transmisor debería mantener el estado de avance (es decir, cual es el último paquete recibido correctamente) de cada uno de los receptores y procesar los ACKs que le llegan para determinar si todos los receptores recibieron correctamente un determinado paquete, lo que es muy ineficiente. Por este motivo, en el caso de una comunicación multipunto no es apropiado que la responsabilidad de la confiabilidad esté en el transmisor [Pingali94a,Towsley97a]. Un ejemplo de protocolo multicast confiable que usa ACKs es Reliable Multicast Transport protocol (RMTP) [Lin96a, Paul97a].

Por otro lado, los mecanismos que responsabilizan de la confiabilidad de la comunicación a los receptores, hacen que el transmisor retransmita un paquete solamente cuando recibe un NACK (respecto de ese paquete) de parte de al menos un receptor. Por su parte cada receptor es responsable por detectar la pérdida de información y pedir la retransmisión del paquete correspondiente. Las pérdidas generalmente son detectadas porque se producen saltos en la secuencia de la numeración de los paquetes recibidos. Para asegurar la retransmisión de un paquete, después de haber enviado el NACK, el receptor activa un reloj que es propio a esa solicitud de retransmisión. Si el reloj indica que se ha cumplido cierto tiempo límite sin recibir la retransmisión del paquete, el receptor envía nuevamente el NACK solicitando el paquete. El esquema recién descrito para lograr confiabilidad es del tipo "receiver-initiated" [Pingali94a, Towsley97a], ya que es el receptor quién inicia el ciclo de recuperación de la información perdida.

Las principales características de los protocolos "receiver-initiated" son:

a) La fuente no tiene que procesar un ACK por cada paquete y por cada receptor. Sólo procesa los NACKs que le son enviados cuando un paquete no llega a todos los receptores.
b) La fuente no necesita conocer el estado de los receptores. Esto implica que la implementación de esta técnica requiere menos memoria que la de "sender-initiated".
c) En comunicaciones multipunto es más eficiente que la técnica de "sender initiated", ya que la pérdida de paquetes es un fenómeno raro (respecto a la no-pérdida del mismo paquete). Por este motivo, la cantidad de NACKs que el transmisor debe procesar es mucho menor que los ACKs del caso "sender initiated". Esto implica que disminuye la implosión en el transmisor y que se produce una economía en el uso de los recursos de la red, tales como uso de procesadores, uso de los canales de comunicación y buffers.

Debido a las buenas características de los protocolos "receiver initiated", la mayoría de los protocolos multicast modernos han adoptado este esquema para la recuperación de paquetes perdidos [Pingali94a, Towsley97a, Yamamoto97a, Kasera98a]. Además, para lograr un buen desempeño respecto al *throughput*, estos protocolos usan el mecanismo de *selective repeat* para la retransmisión de los paquetes perdidos [Bertsekas87a].

Entre los protocolos de transporte multicast confiable que usan solamente NACKs para garantizar confiabilidad en la transmisión se pueden mencionar: Uniform Reliable Group Communication Protocol (URGC) [Aiello93a], Scalable Reliable Multicast (SRM) [Floyd95a, Floyd97a], Transport Protocol for Reliable Multicast (TRM) [Sabata96a], Multicast Transport Protocol (MTP) [Armstrong92a], Structured-Oriented Resilient Multicast (STORM) [Xu97a], Multicast Dissemination Protocol (MDP) [Macker96a] y Adaptive File Ditribution Potocol (AFDP) [Cooperstock96a].

Existen otros protocolos que logran confiabilidad al usar una combinación de ACKs y NACKs. Para esto, se usa una estructura jerárquica en la cual existen nodos intermediarios (o representantes). Un miembro que detecta la pérdida de un paquete envía el NACK correspondiente a su representante. Por su parte este representante fusiona todos los ACKs que recibe respecto de un paquete y envía un único ACK a la fuente. En el caso de detectar alguna falla el nodo intermediario envía un NACK a la fuente pidiendo la retransmisión del paquete perdido. Entre estos protocolos pueden mencionarse: Reliable Broadcast Protocol (RBP) [Chang84a], Reliable Multicast Protocol (RMP) [Whetten95a], Log-Based Receiver Multicast Protocol (LBRM) [Holbrook95a], Tree-Based Multicast Transport Protocol (TMTP) [Yavatkar95a], Xpress Transfer Protocol (XTP) [Strayer92a, Xtp], Multicast File Transfer Protocol (MFTP) [Miller97a], etc.

2.2 ¿Cómo se detecta la pérdida de un paquete?

En la mayoría de los casos la pérdida de paquetes se detecta porque se produce un salto en la secuencia de la numeración de los paquetes que llegan al receptor. Por otro lado, cuando la pérdida afecta a un paquete retransmitido o cuando no se retransmite el paquete, la detección es hecha usando en el mecanismo de timeout.

2.3 ¿Quiénes envían el NACK?

Dado que el mejor desempeño lo tienen los protocolos que sólo usan NACKs, en lo que resta de esta taxonomía nos referiremos sólo a este tipo de protocolos.

La pérdida de cada paquete naturalmente particiona la red en dos clases: los miembros que son afectados por la falla y los miembros que no fueron afectados por ella. Todos los miembros afectados por la pérdida son candidatos a enviar NACKs y todos los miembros no afectados por la pérdida son candidatos a retransmitir el paquete perdido (en los casos que mantengan copia de ella).

Para el caso del envío de NACKs el problema surge debido al hecho de que si todos los miembros afectados por la falla envían sus NACKs se producen implosiones de NACKs en los lugares donde transitan estos NACKs. La implosión es un fenómeno indeseable porque por un lado hace mal uso de los recursos de la red, tales como procesadores, canales de comunicación y buffers, y por otro lado genera congestión, lo que obviamente causa una degradación en las medidas de rendimiento globales de la red, como el retardo y el throughput [Pingali94a, Towsley97a]. Por estos motivos, es necesario disminuir la implosión. La primera técnica, que es compatible con la que se describe más adelante, consiste es disminuir el tamaño del grupo de los potenciales candidatos a enviar NACKs. Los métodos usados para generar grupos locales de potenciales transmisores de NACKs se explican en la sección 2.3.1 de este artículo.

Lógicamente que el problema de implosión es menos grave si el grupo que puede enviar el NACK es menor, sin embargo debido a que este grupo normalmente está compuesto por varios miembros, todavía persiste el hecho de que puede haber implosión de NACKs. Por este motivo además de la transmisión local de NACKs, se emplean otras formas de disminuir la implosión. En el caso de usar transmisión global de NACKs es imprescindible usa la forma de disminución de la implosión que se describe a continuación.

En términos generales, la otra técnica usada para disminuir la implosión, que consiste en que el (los) primer(os) miembro(s) que primero transmiten un NACK inhibe(n) al resto de los candidatos a enviarlo. Para lograr esto, cada uno de los miembros que son candidatos a enviar el NACK realizan un experimento aleatorio y, según el resultado de ese experimento determinan cuando (en el futuro) deben transmitir su NACK. Cuando el NACK es efectivamente transmitido, el envío se hace de forma que además de llegar a los miembros que tienen la información requerida, llegue también a los miembros que la perdieron. Con esto se logra que un miembro que recibe el NACK antes de transmitir el suyo, inhibe su propia transmisión, disminuyendo de esta forma la implosión.

Respecto del tipo de variable aleatoria que es más conveniente usar para determinar el instante en que se debería enviar un NACK, han aparecido en la literatura varias propuestas. Por ejemplo: SRM [Floyd97a] propone usar una distribución uniforme. Por su parte [Nonnenmacher98d] propone usar una variable aleatoria con distribución exponencial truncada. En [Grosslauser97a] se presenta un algoritmo denominado DTRM (*Deterministic Timeouts for Reliable Mullticast*) que usa tiempos límites deterministas para identificar al miembro que debe enviar el NACK.

2.3.1 ¿Cómo disminuir el tamaño del grupo de potenciales candidatos para enviar NACKs?

Para conseguir esto básicamente se distinguen tres técnicas. La primera consiste en que el primer nodo que detecta la falla inhibe a sus descendientes de enviar el NACK. Otra forma es estructurar los miembros del grupo en una jerarquía de múltiples niveles. Cada jerarquía tiene un encargado de procesar y fusionar los NACKS provenientes de los receptores de su nivel. De este modo, a través del encargado sólo se envía un NACK al nivel superior. Entre los protocolos que utilizan esta técnica podemos mencionar: uso de "representantes" [DeLucia96a], LBRM [Holbrook95a], TMTP [Yavatkar95a]. Finalmente existe la técnica que utiliza el paso de fichas (*tokens*). En este caso, los miembros del grupo se organizan según una estructura de *token ring* de modo que un grupo de receptores tiene un representante que procesa los reconocimientos provenientes de ellos. Entre estos representantes se pasa una ficha, de modo que solamente el centro que tiene la ficha envía el ACK si y solo si ha recibido todos los paquetes desde el último reconocimiento. En caso contrario envía un NACK pidiendo la retransmisión del paquete. Entre los protocolos que usan este tipo de técnica podemos mencionar RBP [Chang84a], RMP [Montgomery96a], MTP [Armstrong92a].

2.4. ¿Hacia dónde enviar el NACK?

Una vez que uno o varios de los miembros decide enviar el NACK, surge la cuestión de hacia adonde enviarlo. Respecto de esta decisión existen 3 posibilidades:

1. Enviarlo a todos los miembros del grupo, incluida la fuente (Broadcast dentro del grupo).
2. Enviarlo a un subconjunto de los miembros del grupo, en particular al subconjunto de los miembros que tiene la información requerida y que están más cerca del miembro que envía el NACK.

3. Enviarlo a la fuente.

2.5 ¿Quién retransmite el paquete perdido y hacia dónde?

En el caso en que el NACK sea enviado a todos los miembros del grupo, los miembros que poseen la información perdida son candidatos a retransmitir el paquete. Esto genera una situación análoga a la que causa la implosión de NACKs, pero en este caso se produciría una implosión del paquete retransmitido. Para evitar la implosión de retransmisiones se usa la misma técnica de reducción de implosión de NACKs, es decir, cada miembro que posee el paquete determina, a través de un experimento aleatorio, en que instante en el futuro lo transmitirá. En caso que el miembro llegue a este instante sin haber recibido previamente el paquete (proveniente de la retransmisión por otro miembro , envía dicho paquete en forma broadcast dentro del grupo. En el otro caso, es decir, cuando el miembro recibe el paquete antes de que se cumpla el plazo fijado para enviarlo, inhibe su transmisión, lo que disminuye la implosión de retransmisiones.

Cuando la retransmisión la puede efectuar solamente un subconjunto de miembros que reciben el NACK, la situación es similar, al caso anterior, sólo que menos dramática. Como en este caso también puede producirse implosión, esta se disminuye de la misma forma descrita en el párrafo anterior.

Cuando la fuente es la única encargada de efectuar la retransmisión, el NACK se transmite en forma unicast a la fuente. La primera vez que la fuente recibe un NACK solicitando el paquete perdido, lo retransmite en forma multicast a todos los miembros del grupo. Este método tiene la ventaja de que es simple y no tiene problemas de implosión de retransmisiones. Sin embargo, tiene la desventaja de que se puede producir una latencia muy grande si la fuente está lejos del lugar en el cual se produjo la falla.

3. Descripción de algunos algoritmos de transmisión multicast confiable.

Reliable Multicast Transport protocol (RMTP) [Lin96a,Paul97a]. Este protocolo fue diseñado para transmitir grupos de archivos desde un transmisor a múltiples receptores. La información es transmitida desde la fuente hacia todos los receptores, a través de un árbol de distribución. Usa una estructura jerárquica basada en árbol, donde la raíz es la fuente que transmite los archivos. En esta estructura los receptores están agrupados en regiones locales o dominios, de acuerdo a su proximidad a la fuente. Cada región cuenta con nodos especiales intermediarios llamados "Designated Receiver" (DR), que son los responsables por procesar los ACKs recibidos desde los receptores de su dominio, transmitir el ACK hacia sus DRs padre, y transmitir los paquetes perdidos a los receptores correspondientes. Esta estructura jerárquica multinivel es una forma de evitar la implosión de ACKS, debido a que los receptores envían periódicamente ACKs a su DR de nivel superior y éste lo transmite a su vez a su DR de nivel superior, hasta que el DR de nivel más alto envía el ACK al transmisor (fuente). Los paquetes perdidos son retransmitidos usando *selective repeat.* Además, los paquetes retransmitidos son enviados en forma unicast a los receptores si el número de ACKs que recibe el DR está debajo de un cierto umbral, y en forma multicast en caso contrario.

Uniform Reliable Group Communication Protocol (URGC) [Aiello93a]. Este protocolo usa control centralizado para recuperar los errores. El control rota entre los miembros del grupo. Los nodos mantienen un historial de mensajes enviados. Las retransmisiones se piden en forma unicast hacia el nodo central de turno.

Scalable Reliable Multicast (SRM) [Floyd95a, Floyd97a]. Este protocolo fue propuesto para trabajar sobre IP-Multicasting en aplicaciones de pizarra compartida (wb). El protocolo garantiza la correcta recepción de los datos enviados a un grupo multicast. Para esto hace uso del mecanismo "receiver-initiated". Además, adopta el esquema de hacer multicast tanto de los NACKs como de las retransmisiones, con el objeto de disminuir el tiempo de recuperación del error, ya que al hacer multicast del NACK los miembros más cercanos al nodo que pide retransmisión pueden enviarle el paquete solicitado. Los paquetes son numerados secuencialmente, y la falla se detecta al encontrar un salto en la secuencia de numeración de los paquetes. Cuando los receptores detectan alguna pérdida, solicitan la retransmisión de la información perdida mediante NACKs. Para evitar la implosión de NACKs cada receptor retarda el envío de su propio NACK un tiempo distribuido según una variable aleatoria uniforme en un intervalo $[AT, (A+B)T]$, donde T es el retardo de ida y vuelta del receptor a la fuente, y A y B son constantes. Si antes que expire el temporizador, el receptor recibe un NACK solicitando el mismo paquete, inhibe el envío de su propio NACK. Como todos los miembros del grupo no afectados por la falla poseen una copia correcta de la información recibida, cualquiera de ellos podría retransmitirla cuando recibe un NACK. Por lo tanto, para evitar enviar retransmisiones duplicadas, después de recibir un NACK cada nodo que tiene la información retarda su respuesta por un tiempo aleatorio según una distribución uniforme en el rango $[a\,t, (a+b)t]$, donde t es el tiempo de propagación entre el nodo que puede contestar y el receptor que solicita la información. Cuando el temporizador expira la información solicitada es enviada a todo el grupo en forma multicast.
Existen varios artículos que proponen mejoras a este protocolo[liu97a,Liu97b]. En [Liu97a] se exponen dos mecanismos de recuperación local para mejorar el rendimiento de SRM. Estos mecanismos están diseñados para evitar el fenómeno de implosión y restringir el ámbito de recuperación. El primero es un mecanismo denominado ``hop-scoped" que usa saltos para limitar la distancia que deben viajar los NACKs y los mensajes. Cuando un miembro recibe un NACK responde a la petición programando la retransmisión del paquete en un tiempo aleatorio mayor o igual al tiempo de propagación entre el receptor generador del NACK y el mismo. Cuando el temporizador expira el paquete es retransmitido en forma multicast a los miembros del grupo. El otro esquema denominado Group Scoped forma grupos de recuperación local. Por lo cual el NACK de un miembro lleva consigo la dirección del grupo a que pertenece. De esta manera la respuesta es enviada solo al grupo, lográndose así direccionalidad en la retransmisión, lo que implica ahorro de recursos.

Transport Protocol for Reliable Multicast (TRM) [Sabata96a]. Este esquema de comunicación multipunto confiable se utiliza en aplicaciones como pizarras compartida y aplicaciones multimediales. TRM supone que la fuente no conoce los integrantes del grupo y permite que los miembros se incorporen o abandonen el grupo multicast e cualquier momento. Cada receptor es responsable de detectar la información perdida o corrupta. Cada paquete de datos que es transmitido por la fuente, posee un número que lo identifica en forma única. Un receptor puede detectar una pérdida por encontrar un salto en la secuencia de numeración de los paquetes recibidos. Para detectar si existe pérdida del último paquete recibido, la fuente envía un mensaje de control indicando el número del último paquete transmitido. Si un receptor TRM detecta la pérdida de un paquete, envía en forma multicast un NACK al grupo. Cada receptor espera un tiempo aleatorio antes de enviar el NACK. Si recibe un NACK por el mismo paquete que él ha perdido, suprime su propio NACK. De esta manera se evita enviar NACKs redundantes al transmisor. Cualquier miembro del grupo que tenga el paquete solicitado puede retransmitirlo. Las retransmisiones repetidas pueden evitarse usando la misma técnica de retardo utilizada para evitar la implosión de NACKs.

Multicast Transport Protocol (MTP) [Armstrong92a,Borman94a]. Este protocolo ofrece orden total en la entrega de la información, para conseguir esto trabaja sobre la base de fichas. Existe un nodo maestro que garantiza la membresía en el grupo multicast y la ficha. Además controla el comportamiento del grupo multicast incluyendo su rendimiento. Los miembros del grupo multicast son clasificados en productores/consumidores o consumidores puros. Los productores/consumidores pueden enviar y recibir información, mientras que los consumidores puros solo pueden recibirla. Un productor/consumidor debe poseer un token para poder enviar información a los demás miembros del grupo. Esto garantiza que sólo un productor enviará información en cualquier instante. Los tokens son entregados por el maestro e incluyen el número de secuencia del mensaje. Cuando un consumidor detecta la pérdida de información envía, en forma unicast, un NACK al productor para solicitar la retransmisión de la información perdida, la cual es enviada en forma multicast a todo el grupo.

Structured-Oriented Resilient Multicast (STORM) [Xu97a]. Organiza los miembros del grupo multicast en forma de grafo (no necesariamente un árbol). Cada miembro del grupo mantiene una lista de sus nodos padres, y cada vez que detecta la pérdida de un paquete envía un NACK a cada padre por vez. Cuando el padre recibe un NACK retransmite el paquete solicitado en forma unicast al solicitante.

Multicast Dissemination Protocol (MDP) [Macker96a]: es un protocolo marco que se utiliza para implementar la transmisión confiable de grupos de archivos y otros tipos de objetos. MDP usa NACKs para avisar las pérdidas de un paquete y un mecanismo de supresión de NACKS redundantes para evitar implosión. Para las retransmisiones usa un esquema de selective repeat.

Adaptive File Ditribution Protocol (AFDP) [Cooperstock96a]. Este protocolo sirve para distribuir archivos grandes(p.ej: actualización de sistemas operativos o archivos MPEG) desde una fuente a varios destinos. ADFP esta basado en un modelo en el cual cualquier host que recibe datos es denominado "suscriptor" (*suscriber*) y cualquier host que envía datos es llamado "editor" (*publisher*). Entre los suscriptores del grupo se escoge un suscriptor especial, el que es llamado "secretario" (*secretary*). Este host es responsable por manejar la membresía en el grupo, es decir que host entran o salen del grupo, autorizar a los editores a transmitir, y determinar el mejor modo de transmitir la información a todos los suscriptores. Este modo puede ser unicast, broadcast o multicast, dependiendo de las características de los suscriptores. Los suscriptores usan NACKs para solicitar la retransmisión de los paquetes perdidos o corruptos. Los paquetes son retransmitidos al grupo entero, basándose en la suposición que si un host ha perdido un paquete, probablemente los otros también lo han hecho.

Reliable Broadcast Protocol (RBP) [Chang84a]. Entrega comunicación multipunto a usuarios conectados a una red de área local. Combina ACKs y NACKs para obtener confiabilidad. Este protocolo trabaja usando un esquema de "token site" para la comunicación entre transmisores y receptores. El "token site" envía en forma multicast un ACK al grupo por cada mensaje recibido. Cuando un ACK llega al transmisor le indica que el mensaje enviado por él fue recibido por el "token site". Por otro lado los receptores usan este ACK para detectar perdidas, ya que son numerados. Si el receptor detecta la pérdida de un paquete envía un NACK al "token site" quien reenvía el mensaje en forma multicast.
El "token site" es rotado entre todos los miembros de la red. Un miembro acepta ser el siguiente "token site" solo si ha recibido todos los mensajes anteriores al mensaje de cambio de site.

Reliable Multicast Protocol (RMP) [Whetten95a]. Este protocolo entrega los datos en forma ordenada. Es una derivación de RBP, pero difiere de él en el hecho que envía en forma multicast un ACK al grupo completo. Esto ordena los datos consistentemente en los nodos y es una forma de pasar la ficha a un nuevo nodo. También difiere de RBP en el sentido que mejora el througput al hacer que un ACK pueda reconocer un número arbitrario de paquetes. Para esto cada nodo que pertenece a un token ring, mantiene una estructura de datos denominada "Ordering Queue" en la cual los paquetes de datos y los reconocimientos son organizados sobre la base de "timestamps". Un paquete perdido aparece como un lugar vacío en la "Ordering Queue". Cuando un nodo se tranforma en el "token site" actual, la "Ordering Queue" debe estar completamente llena. Es decir, cuando todos los paquetes desde el último reconocimiento han sido recibidos por el token site actual, entonces el nodo envía en forma multicast su ACK y pasa el token al siguiente site. Cuando un receptor RMP detecta una pérdida envía un NACK para requerir retransmisión de la información perdida. Una versión más moderna de RBP [Montgomery96a] usa un esquema modificado de la política de pedido y recuperación de paquetes de SRM.

Log-Based Receiver Multicast Protocol (LBRM) [Holbrook95a]. Este protocolo ha sido diseñado para ofrecer confiabilidad a una aplicación de Simulación Distribuida Interactiva. Es un mecanismo "receiver initiated" en el cual los receptores tiene la responsabilidad por detectar la pérdida y pedir la retransmisión.
Para conseguir confiabilidad usa servidores especiales en los cuales se guarda la información que es transmitida desde la fuente. Cuando un receptor detecta una pérdida, pide la retransmisión directamente a este servidor. La fuente guarda los datos transmitidos hasta que el servidor le envia un ACK.
El protocolo no intenta evitar la implosión de NACKs, ya que suponen que cada servidor secundario atiende a un número razonablemente pequeño de receptores.

Tree-Based Multicast Transport Protocol (TMTP)[Yavatkar95a]. Usa una estructura de árbol en la cual los ACKs idénticos son fusionados en uno, y propagados árbol arriba hasta la raíz. También usa NACKs como una forma de suprimir mensajes de error.

Xpress Transfer Protocol (XTP) [Strayer92a,Xtp]. Este protocolo ha sido diseñado para sustentar una amplia gama de aplicaciones, entregando toda la funcionalidad de TCP, UDP y TP4. Posee un protocolo de administración de grupos explícito (MGM protocol) que sirve, por ejemplo, para negar o permitir el acceso de un usuario a un grupo multicast determinado. La comunicación confiable está basada en el uso de NACKs. Los NACKs son enviados en forma unicast hacia la fuente, mientras que las retransmisiones son hechas en forma multicast. El transmisor también puede iniciar un proceso de sincronización para determinar el estado de los receptores. En este caso los receptores usan una técnica de esperar un tiempo aleatorio antes de enviar el mensaje de control hacia la fuente, con el objetivo de reducir la implosión de mensajes de control.

Multicast File Transfer Protocol (MFTP) Miller97a]. Este protocolo opera sobre UDP. Sirve para entregar un medio confiable para la transferencia de archivos desde un transmisor a varios receptores simultáneamente. El transmisor MFTP tiene como funciones manejar el grupo multicast, iniciar la transferencia de los archivos y controlar la operación de transferencia.
El receptor tiene como funciones juntarse al grupo multicast, recibir la información enviada por el transmisor y entregar el estado de recepción cuando se lo solicita el transmisor, es decir un ACK es

enviado por el receptor sólo si el transmisor lo solicita. La detección de pérdidas es realizada por el receptor quién envía NACKs al transmisor para avisar de la pérdida de un paquete.

Deterministics Timeouts for Reliables Multicast (DTRM) [Grossglauser97a]. Este protocolo evita el problema de implosión de NACKs, suponiendo que la variación de retardo fin a fin es acotada y conocida por los nodos que son hojas del árbol de distribución. Además, determina en forma distribuida los retardos mínimos para emitir un NACK, para cada receptor de un árbol multicast, como una función de la topología del árbol y del retardo de ida y vuelta de cada receptor a la fuente.
Una ventaja de DTRM es que ni la fuente ni los nodos intermedios necesitan conocer el grupo multicast ni la topología del árbol, solo requieren tener información de la topología local. Además, cada receptor sólo requiere necesita conocer el retardo de ida y vuelta entre sí mismo y la fuente. La comunicación se hace sólo entre padres e hijos, por lo cual el número de mensajes permanece bajo aún para grupos multicast grandes. DTRM garantiza que solo un NACK es enviado por cada pérdida. Los timeouts calculados por DTRM son óptimos con respecto al máximo tiempo de respuesta experimentado por receptor y transmisor. Además es robusto frente a la pérdida de NACKs. Los timeouts y los tiempos de respuesta fueron estudiados en forma experimental usando para esto sesiones típicas de Mbone.

4. Conclusiones

En muchas aplicaciones multipunto es necesario que la comunicación sea confiable, en el sentido que todos los paquetes transmitidos deben ser recibidos correctamente por todos los receptores. La solución de este problema tiene un gran impacto sobre el retardo que experimentan los paquetes y el throughput de la red.
Los algoritmos publicados hasta la fecha utilizan variantes de unas pocas ideas básicas. Todavía es necesario encontrar métodos que además de asegurar la confiabilidad, produzcan simultáneamente una baja latencia y una baja implosión.

Agradecimientos

Este trabajo fue financiado en parte por el proyecto Fondecyt # 1980416 /1998 y por proyecto FONDEF D96/1064, Chile.
Parte de este trabajo fue realizado por M. Barría durante su estadía en el INRIA-Sophia Antipolis, Francia. Parte del trabajo de M. Barría estuvo financiado por una beca del ENST/ Francia.

Bibliografía

[Aiello93a] R. Aiello, E. Pagani and G.P. Rossi, "Design of a Reliable Multicast Protocol",Proc. INFOCOM'93,.pag. 75-81 Mar 1993.

[Armstrong92a], S. Amstrong, A. Freier and K. Marzullo,"Multicast Transport Protocol", Internet RFC 1301, Feb.1992.

[Bertsekas87a] D.Bertsekas and R. Gallager, "Data Networks", Prentice Hall Inc., 1987.

[Blahut83a],R. Blahut, 'Theory and Practice of Error Control Codes", Addison Wesley, 1983.

[Borman94a]C. Borman et al. "MTP-2: Towards Achieving the S.E.R.O. Properties for Multicast Transport", Proc. Int. Conf. Comp. Commun. And Networks'94, Sept.,1994

[Chang84a] J.M. Chang and N.F.Maxemchuk, "Reliable Broadcast Protocol", Journal ACM Transactions on Computer Systems, Vol. 2, Num. 3 Aug. 1984, pages 251-273.

[Cooperstock96a], J.R. Cooperstock and S. Kotsopulos, "Why Use a Fishing Line when you have net? An Adaptive multicast Data Distribution Protocol", Proc. USENIX'96, 1996.

[DeLucia96a] D. DeLucia and K. Obraczka, "Multicast Feedback Suppression Using Representatives", Tech. Report USC TR97-638, University of Southern California, June 1996.

[Eriksson94a], H. Eriksson," MBone. The Multicast Backbone ", Communications of the ACM ", Vol. 37, pag. 54--60, Aug.1994.

[Floyd95a], S. Floyd, V. Jacobson, C-G. Liu, S. McCanne and L. Zhang, "A Reliable Multicast Framework for Lightweight Sessions and Application- Level Framing, Extended Report ", LBNL, Sept 1995.

[Floyd97a], 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.

[Grossglauser97a], M. Grossglauser, "Optimal Deterministic Timeouts for Reliable Scalable Multicast ", IEEE Journal on Selected Areas in Communications, 15 (3), April 1997

[Holbrook95a], H.W. Holbrook, S.K.Singhal and D.R. Cheriton, Log-Based Receiver Reliable Multicast for Distributed Interactive Simulation ", Proc. ACM SIGCOMM'95, Oct. 1995, pag. 328-341

[Huitema96a], C.Huitema, " The case for packet level FEC ", Proc. of IFIP 5th. Int. Workshop on Protocols for High Speed Networks (PfHSN'96), INRIA. Sophia Antipolis, France, Oct. 1996, Chapman & Hall

[Kasera98a] S. Kasera, J. Kurose and D. Towsley, " A Comparison of Server-Based and Receiver-Based Local Recovery Approaches for Scalable Reliable Multicast", Proc.INFOCOM'98, April 1998, San Francisco, USA

[Lin96a] J.C. Lin and S. Paul, "RMTP: A Reliable Multicast Transport Protocol ", Proc. INFOCOM'96, Mar. 1996

[Liu97a] C-G. Liu, D. Estrin, S. Shenker and L. Zhang, "Local Error Recovery in SRM: Comparison of Two Approaches", Tech. Report USC TR97-648, University of Southern California, Feb 1997.

[Liu97b] C-G. Liu,, D. Estrin,, S. Shenker and L. Zhang, " Timer Adjustment in SRM" Tech. Report USC TR97-656, University of Southern California,, 1997

[Macker96a], J. Macker and W. Dang "The Multicast Disemination Protocol (MDP) Framework ", Internet draft, Nov. 1996,

[Miller97a] K. Miller et al.," Starbust Multicast File Transfer Protocol (MTFP) Specification ", Internet Draft, IETF, draft-miller-mtfp-spec-02.txt, Jan.1997

[Montgomery96a] T. Montgomery, J. Callahan and B. Whetten, "Specification and Design of a Fault Recovery Model for Reliable Multicast Protocol" Technical report NASA-IVV-96-009, NASA/West Virginia University Software IV& V Facility, April1996

[Nonnenmacher96a], J.Nonnenmacher and E.Biersack, "Reliable Multicast: Where to use FEC ", Proc.5th. Workshop om Protocols for High Speed Networks,editors W.Dabbous and C.Diot, pag. 134-148, Oct.,1996, Sophia-Antipolis,France

[Nonnenmacher97b] J. Nonnenmacher, E. Biersack and D. Towsley, " Parity-Based Loss Recovery for Reliable Multicast Transmissions ", IEEE/ACM Transactions on Networking, 1997; Also in Proc. ACM SIGCOMM'97, June 1997.

[Nonnenmacher98d] J. Nonnenmacher and E. Biersack, Scalable Feedback for Large Groups ", submitted to IEEE/ACM Transactions on Networking, 1998

[Paul97a] S. Paul, K.Sabnani, J. Lin and S. Bhattacharyya, "Reliable Multicast protocolo (RMTP), to appear in IEEE JSAC Special Issue on Network support for Multipoint Communications.

[Pingali94a] S. Pingali, D. Towsley and J. Kurose, " A comparaison of Sender-Initiated and Receiver-Initiated Reliable Multicast Protocols ", Proc. ACM Sigmetrics Conference, May., 1994

[Rizzo97a] L. Rizzo and L. Viciano, "A Reliable Multicast Data Distribution Protocol Based on Software FEC Techniques", Proc High perf. Communications Systems Conference, 1997.

[Sabata96a] B. Sabata, M.J. Brown and B.A. Denny, "Transport protocol for Reliable Multicast: TRM ", Proc. IASTED International Conference Networks, Jan. 1996, pag. 143-156

[Strayer92a] W.T. Strayer, B.J. Dempsey and A.C. Weaver, " XTP: The Xpress Transfer Protocol ", Addison-Wesley, 1992

[Towsley97a] 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 .

[Whang97a] Z. Whang et al., " Framework for Reliable Multicast Design Approach ", Proc. HIPPARCH'97, 1997.

[Whetten95a] B.Whetten, T. Montgonmery and S. Kaplan, " A High Performance Totally Ordered Multicast Protocol. Theory and Practice in Distributed Systems ", Springer Verlang, 1995.

[Xu97a] X.R.Xu et al.,"Resilient Multicast support for Continous-Media Applications", Proc. NOSSDAV'97, 1997

[Yamamoto97a] M. Yamamoto, J. Kurose, D. Towsley and H. Ikeda, A Delay Analisis of Sender-Initiated and Receiver-Initiated Reliable Multicast Protocols ", Proc. IEEE INFOCOM'97, April 1997, Kobe,Japan.

[Yavatkar95a] R. Yavatkar, J. Griffioen and M. Sudan, "A Reliable Dissemination Protocol for Interactive Collaborative Applications ", Proc. ACM Multimedia'95, 1995

[xtp] XTP Web, http://www.ca.sandia.gov/xtp/xtp.html