Diseño e Implementación de un Protocolo de Transporte Multicast Fiable (PTMF)

M. Alejandro García,[1], Antonio Berrocal[1],
Ana Verónica Medina[2], Francisco Pérez[2]

[1] ENDITEL Endesa Ingeniería de Telecomunicaciones, Departamento de I+D, Edificio WTC, Isla de la Cartuja s/n, 41092, Sevilla, (ESPAÑA)
{magarcia,aberrocal}@enditel.es

[2] Departamento de Tecnología Electrónica, Escuela Técnica Superior de Ingeniería Informática, Avda Reina Mercedes s/n, 41012, Sevilla, (ESPAÑA)
{vmedina,francisco.perez}@us.es

Resumen Recientes investigaciones en el campo de transporte multicast fiable [1,2] han demostrado la eficiencia de los protocolos que utilizan como control de fiabilidad técnicas basadas en el receptor, y más aún, han demostrado el gran rendimiento que obtienen los protocolos con técnicas basadas en árbol o grupo local. En este artículo se describe el diseño e implementación de un protocolo en el nivel de transporte denominado PTMF que proporciona fiabilidad a las aplicaciones multicast. Puede ser utilizado tanto en entornos uno-a-muchos (1:N) como muchos-a-muchos (N:N). PTMF obtiene una gran escalabilidad organizando los miembros de la comunicación en grupos locales jerárquicos utilizando un algoritmo distribuido integrado en un pequeño subprotocolo interno denominado Control de Grupo Local. PTMF realiza un control de fiabilidad híbrido orientado al receptor y al emisor con un control de errores distribuido entre todos los miembros de cada grupo local. Permite operar en cuatro modos: Fiable, Fiable Retrasado (permite la incorporación de nuevos miembros a la comunicación en curso), No Fiable y No Fiable Ordenado. PTMF utiliza asentimientos y retransmisiones tanto multicast como unicast, incorpora seguridad de los TPDUs mediante los algoritmos RC2 y MD5, permite el control del ancho de banda consumido por cada emisor multicast y es parametrizable. Además de la especificación, diseño y funcionalidad de PTMF, en este artículo se muestran dos aplicaciones desarrolladas sobre él.

1 Introducción

Multicast es una técnica que permite que copias de un solo paquete (información que se quiere enviar desde un origen) se transfiera a un subconjunto seleccionado de posibles destinos (receptores de la información). El protocolo IP soporta Multicast mediante el uso del protocolo interno IGMP (Internet Group Management Protocol) [3] y reserva las direcciones de clase D para su uso. El IETF ha desarrollado varios protocolos de enrutamiento multicast como DVRMP [4],

MOSPF [5], PIM-SM [6], PIM-DM [7], CBT [8] y BGMP [9] que ya están implantados en muchos routers comerciales. La ventaja principal del uso de multicast es el ahorro de ancho de banda, al permitir transmitir información a múltiples receptores sin duplicar la información enviada a cada uno de ellos. La Internet Multicast Backbone (Mbone) [10] existe desde 1992 como una red virtual para la experimentación del uso de IP Multicast en Internet, recientemente esta red ha evolucionado y se ha convertido en Internet2 [11]. Desde el inicio de la MBone han ido surgiendo distintos grupos de investigación para estudiar y diseñar protocolos de transporte multicast fiables que permiten transmitir datos con fiabilidad a un conjunto grande de receptores. La tendencia general es desarrollar esquemas uno-a-muchos (1:N) en el que un emisor (1) envía datos a múltiples receptores (N) que no pueden convertirse a su vez en emisores. Recientes estudios [1,2] han demostrado la eficiencia de los protocolos que utilizan como control de la fiabilidad técnicas basadas en el receptor [12], y más aún, han demostrado el gran rendimiento que obtienen los protocolos con técnicas basadas en árbol [13,14,15,16] o grupo local [17], siendo estos los que obtienen un mejor rendimiento y una escalabilidad mayor. Otros protocolos no son tan genéricos aplicándose solo a un cierto tipo de servicios más específicos [18,19]. En este trabajo se presenta el diseño y la implementación de un protocolo de nivel de transporte multicast de uso genérico independiente de la aplicación, que puede ser utilizado tanto en entornos uno-a-muchos (1:N) como en muchos-a-muchos (N:N) consiguiendo escalabilidad mediante el uso de técnicas basadas en árbol y grupo local. PTMF (Protocolo de Transporte Multicast Fiable) recoge ideas importantes de otros protocolos precursores y aporta otras nuevas e innovadoras. En las siguientes secciones se muestran las características del protocolo, una implementación realizada del mismo y dos ejemplos de aplicaciones, transferencia de fichero multicast y chat multicast.

2 Características de PTMF

Entre las características más relevantes de PTMF se encuentran:

- Protocolo de la capa de transporte.
- Escalabilidad utilizando técnicas basadas en Grupos Locales jerárquicos.
- Control de Errores Distribuido.
- Control de Fiabilidad Híbrido (orientado al emisor y al receptor).
- Control de Flujo.
- Control de la Congestión.
- Asentimientos Multicast y Unicast.
- Retransmisiones Multicast y Unicast.
- Incorporación de nuevos miembros a la conexión en curso.
- Capacidad N:N (muchos a muchos).
- Seguridad mediante criptografía de los TPDUs (Transport Protocol Data Unit).
- Posibilidad de múltiples unidades de transferencia en una conexión.
- Parametrizable.

PTMF es un protocolo de transporte de nivel 4 de la pila OSI, se sitúa en el mismo nivel que TCP o UDP (ver figura 1) y utiliza IP con capacidad Multicast en la capa de red.

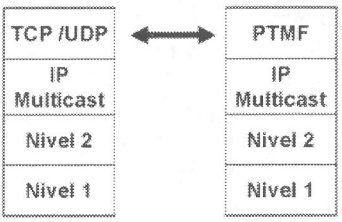

Figura 1. Situación del protocolo PTMF en la pila OSI.

Existen fundamentalmente dos tipos de estrategias para el control de la fiabilidad en los protocolos Multicast:
- Orientado al emisor (Sender-Initiated): en este tipo de protocolos la responsabilidad del control de la fiabilidad recae en el Emisor, éste suele utilizar técnicas basadas en el uso de asentimientos positivos (ACK). Esta técnica requiere que el emisor mantenga una lista por paquete de los receptores de los que ha recibido una confirmación positiva (ACK). Cada vez que el emisor envía un paquete inicia un timer, si expira el timer retransmite el paquete. Por otro lado cuando los receptores reciben un paquete correctamente envía un ACK. Los dos principales problemas que presenta son:

a. La reducción del Ancho de Banda por el exceso número de ACKs.
b. La sobrecarga del emisor por el procesamiento de estos ACKs.

Esta situación se conoce como Implosión de ACKs y se ilustra en la figura 2.
-Orientado al receptor (Receiver-Initiated): en este tipo de protocolos el control de la fiabilidad recae en el receptor, éste suele utilizar técnicas basadas en el uso de asentimientos negativos (NACK). En esta estrategia el emisor continúa transmitiendo paquetes hasta que el receptor le envía un asentimiento negativo (NACK) pidiendo una retransmisión. Cuando esto ocurre el emisor retransmite el paquete solicitado por el receptor. El receptor identifica los paquetes perdidos si recibe un paquete con número de secuencia mayor del esperado o después de un timeout si estaba esperando una retransmisión. El rol del receptor es verificar la recepción de los paquetes. Para evitar la pérdida de los NACK el receptor inicia un timer, si finaliza retransmite el NACK. En los momentos de inactividad el servidor debe de enviar paquetes de información periódicamente para indicar su estado de inactividad a todos los receptores. El problema que presenta esta aproximación es el siguiente:

c. El emisor debe mantener los TPDUs enviados en memoria para posibles retransmisiones durante un tiempo indeterminado, pues no hay mecanismos

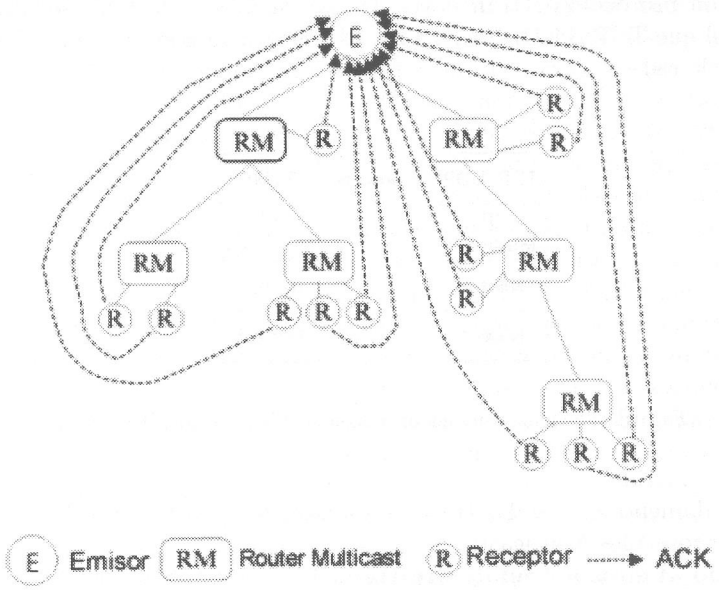

Figura 2. Implosión de ACKs.

que indiquen la correcta recepción de los datos por parte de los receptores y que permita liberar memoria al emisor.

Para resolver estos problemas PTMF utiliza:

1. Un **Protocolo de Control de Grupo Local (CGL)** que organiza a los participantes en la comunicación Multicast de forma que el proceso de asentimiento sea más eficaz.
2. Una **Estrategia de Control de Fiabilidad Híbrida** que combina la estrategia de control de fiabilidad Orientada al Emisor y al Receptor, aprovechando en cada caso lo mejor de ambas estrategias y optimizándolas para obtener un rendimiento mejor.

Con estas dos opciones se resuelven los tres problemas presentados anteriormente como veremos a continuación.

2.1 Protocolo de Control de Grupo Local (CGL)

Antes de proceder a la descripción del protocolo Control de Grupo Local necesitamos definir algunos conceptos: Definimos **Canal Multicast** como un identificador de un flujo de transmisión de información MULTICAST, se compone de una dirección IP Multicast más un Puerto Multicast o TSAP (Transport Service Access Point). Definimos un **Canal Unicast** como un identificador de flujo de transmisión bidireccional de información unicast, se compone de una dirección

Unicast más un Puerto Unicast. Definimos **SocketPTMF** como un punto final de comunicación Multicast entre dos procesos que están en la misma o distintas máquinas comunicándose mediante el protocolo PTMF, utiliza dos canales, un canal multicast y otro Unicast. A diferencia de otros protocolos multicast PTMF permite que existan varios procesos en el mismo host utilizando este protocolo. El Protocolo de Control de Grupo Local, al que denominaremos de aquí en adelante CGL, es un pequeño protocolo dentro de PTMF. CGL es a PTMF lo que ICMP o IGMP son dentro de IP. La función que desempeña CGL dentro de PTMF es: *Organizar los Sockets PTMF presentes en un determinado Canal Multicast y Ámbito (limitado por el campo TTL de los paquetes IP Multicast) en Grupos Locales establecidos de forma jerárquica para que el Control de la Fiabilidad se realice de forma eficiente.* Con ello se pretende conseguir:

1. Reservar lo máximo posible el *Ancho de Banda* de las líneas de comunicación.
2. Evitar la *saturación del Emisor* por procesamiento de la avalancha de ACKs generados para asentir un paquete, este último problema se conoce como cuello de botella en el lado del emisor por procesamiento de ACKs.

Todos los Sockets que pertenecen a un determinado Canal Multicast se organizan formando **Grupos Locales** (GL) en una estructura jerárquica basada en árbol. Se define un Grupo Local como un conjunto de sockets presentes en un determinado Canal Multicast y Ámbito que cooperan entre sí para *garantizar la fiabilidad del protocolo* PTMF y conseguir con su organización una *mayor eficacia* del proceso de asentimientos. El uso de una estructura jerárquica de Grupos Locales permite la recuperación de errores y la retransmisión de los datos perdidos de una forma muy eficiente dentro del Grupo Local, si en el Grupo Local no se puede realizar la recuperación del error se recurrirá al Grupo o Grupos Locales superiores en la jerarquía arborescente (ver figura 3). Dentro del grupo local existe un socket que actúa como **Controlador de Grupo** (CG), su misión es actuar como si fuese el emisor informando a sus CG superiores en el árbol o al emisor en última instancia, de la correcta recepción de los datos dentro del grupo local (retransmisiones, control de los receptores, ...). El Controlador de Grupo manda asentimientos positivos jerárquicos (HACK), este tipo de asentimiento sólo se permite que lo realicen los CG de cada grupo local, de esta forma se evita que se produzca una implosión de ACKs desde todos los receptores hacia el emisor, provocando un cuello de botella en el lado del emisor pero sobre todo una cantidad enorme de procesamiento de todos los ACKs. El CG mandará el asentimiento directamente al emisor si este está en el nivel de la jerarquía inmediatamente superior, en caso contrario se lo mandará al CG del nivel superior en la jerarquía. El concepto de "Grupos Locales" establecidos de forma jerárquica se muestra en la figura 4. CGL permite utilizar diferentes métricas para configurar los sockets en GL. En nuestra implementación hemos utilizado como métrica el número de saltos (TTL), sin embargo el protocolo está abierto a utilizar cualquier otro tipo de métrica. Para minimizar la posibilidad de pérdida los mensajes CGL se envían con redundancia. Un grupo local se identifica por un identificador de Grupo Local (IDGL) que representa a un Grupo Local de

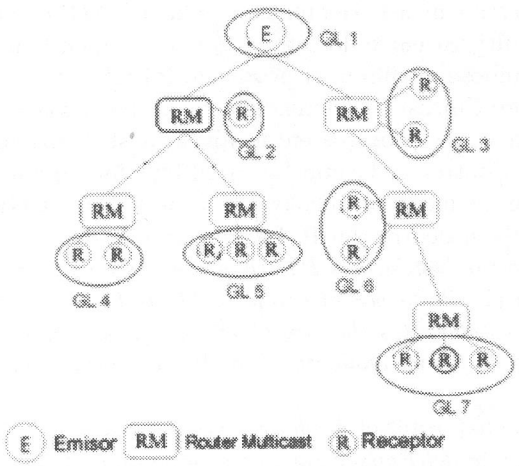

Figura 3. Grupos Locales Jerárquicos.

forma única en toda Internet. El IDGL está formado por el identificador del Canal Unicast del socketPTMF que crea el grupo local.

2.2 Control de Fiabilidad Híbrido

El control de la fiabilidad es realizado dentro de cada grupo local. Estos son tratados como una unidad por el resto de los Grupos Locales (ver figura 3). El control de la fiabilidad se distribuye entre todos los socketsPTMF que pertenecen al grupo multicast. Mediante asentimientos, que se van recogiendo a lo largo de la estructura jerárquica en la que se organiza el grupo multicast, los distintos emisores fuentes tienen constancia de que los datos que envían son recibidos por todos los miembros. Si se pierden datos la recuperación intenta hacerse en el ámbito local, antes de solicitarlos directamente a la fuente, con esto se consigue un Control de Error Distribuido.

Control de Error Distribuido. La información enviada por los emisores se organiza en **ráfagas**, cada una de las cuales está compuesta por un número consecutivo de TPDUs de datos. El número de TPDUs de datos por ráfaga depende del emisor fuente, todos ellos incluyen el número de ráfaga a la que pertenecen. Los miembros de los grupos locales negocian mediante contienda ser CG para cada ráfaga de cada emisor fuente. Una ráfaga delimita el número de TPDU de datos de un emisor fuente para los que permanece un mismo CG. Con esto se evita que haya múltiples negociaciones, y además, facilita reconocer quién es el CG de un TPDU observando únicamente el emisor fuente del que proviene y la ráfaga a la que pertenece. El emisor de los TPDUs **marca** con un **bit ack** aquellos TPDUs de los que quiere recibir asentimiento por parte de los

receptores, simplificando la información que hay que almacenar para asegurar la fiabilidad, así como los cálculos necesarios. Este método también es útil para controlar el número de asentimientos que circulan por la red. El CG asegura la fiabilidad de todos los TPDUs que pertenezcan a una de las ráfagas de la que es CG, actuando, de cara a los sockets vecinos y a los grupos locales hijos, como si fuese el emisor fuente de la misma. Para ello, mantiene una lista con los identificadores de los miembros del grupo local (sockets vecinos) y con los identificadores de los grupos locales hijos, **de los que espera asentimientos positivos** para los TPDUs de datos **con el bit ACK activado** que pertenezcan a una de las ráfagas de la que es CG.El CG es el encargado de retransmitir los datos, si los tiene, a los vecinos e hijos jerárquicos que no los hayan recibido. Si un socket recibe un TPDU de una ráfaga que no tiene un CG asignado se inicia un mecanismo para establecer el CG de la ráfaga, siguiendo un algoritmo de contienda con supresión, en éste el emisor fuente tiene prioridad para ser CG de sus propios paquetes. El algoritmo utiliza un TPDU de búsqueda de CG (MSCG - Multicast Search Controller Group) que es enviado por multicast con un ámbito local al grupo, realizándose supresión en el envío. Cada miembro del grupo local lanza un temporizador aleatorio y cuando finaliza envía el MSCG para hacerse con el control de la ráfaga, si aún no tiene CG asignado. Todos tienen que esperar el tiempo aleatorio, excepto que se trate del emisor original, en cuyo caso no esperará y mandará inmediatamente el MSCG. Es necesario que también envíe el MSCG para que exista la posibilidad de que, por algún problema, no pueda ser CG de sus propios TPDU y lo pueda ser otro.Cuando se recibe un MSCG de la red se anula el temporizador y se anota quien es el CG para enviarle los asentimientos.Por la posibilidad de parametrización que ofrece PTMF es posible configurar estáticamente un determinado socketPTMF en un host para que siempre sea CG, esto es muy útil para que los administradores de red administren de forma eficiente los recursos y puedan así establecer, por ejemplo, que ciertos hosts actúen como caché.

Esquema de asentimientos. PTMF utiliza un esquema ARQ de asentimientos. A través de ellos se asegura que todos los datos enviados al grupo multicast por los distintos emisores fuentes son recibidos por todos los miembros del grupo. Se utilizan **asentimientos positivos** para indicar los datos que se han recibido correctamente, y **negativos** para solicitar datos que no han sido recibidos o que se han recibido erróneamente. Todos los miembros del grupo multicast tienen que ir reconociendo la correcta recepción de los datos de forma independiente, con el objetivo de que los emisores fuentes puedan liberar memoria y seguir enviando nuevos datos. Todos los asentimientos positivos son acumulativos.
Los tipos de asentimientos utilizados en PTMF son:
- *Asentimiento positivo (ACK):* este tipo de asentimiento se envían para indicar que se han recibido todos los paquetes con número de secuencia igual o menor (acumulativos) al indicado. Los ACK se envían por **unicast** al controlador de grupo correspondiente, el cual espera recibir ACK de todos sus vecinos.
- *Semiasentimiento positivo jerárquico (HSACK):* se envía por **multicast** y re-

presenta a un Grupo Local . Es enviado a los padres jerárquicos por un CG cuando él u otro miembro del grupo local (vecino) ha recibido correctamente el TPDU. Este asentimiento no asegura que haya sido recibido por todos los miembros del grupo local, pero sí que al menos uno lo ha recibido. Es enviado para que los CG padres no retransmitan el TPDU cuando le venzan los temporizadores de espera, y no consideren que el grupo local se ha caído cuando venzan los temporizadores internos de retransmisión.

- *Asentimiento positivo jerárquico (HACK):* una vez que el controlador de fiabilidad para una ráfaga dada ha recibido ACK de todos los miembros de su grupo local, así como HACK de todos los grupos locales pertenecientes a un nivel inferior, envía un HACK por **multicast** a los grupos locales superiores de los que depende. Su funcionamiento es similar al de los asentimientos positivos (ACK), siendo también acumulativo, además, al enviarse por multicast, se realiza supresión en el envío. Los HACK fluyen en la jerarquía de grupos locales hasta llegar al emisor (Figura 4).

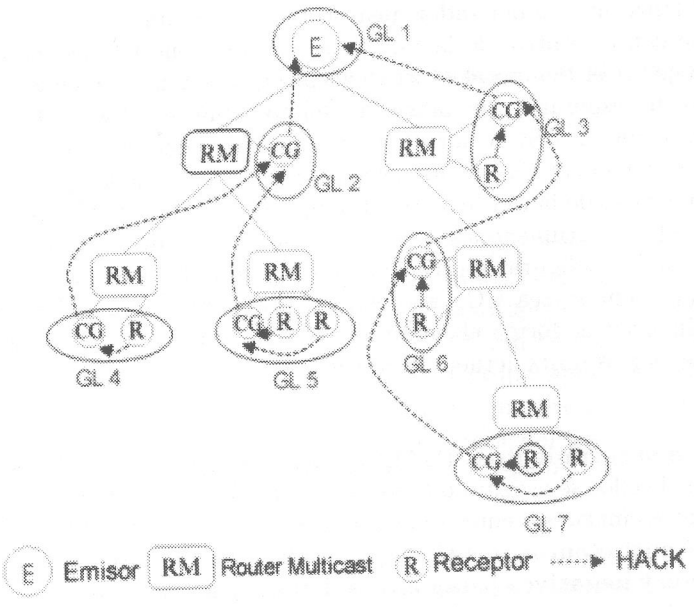

Figura 4. Asentimientos jerárquicos.

- *Asentimientos negativos (NACK):* se utilizan para pedir paquetes que no se han recibido. Dentro de un NACK se pueden solicitar uno o más paquetes de datos. Se envían por **multicast**, realizándose supresión en el envío y a diferencia de los asentimientos positivos estos son selectivos en lugar de acumulativos. Cuando un miembro del grupo multicast detecta la pérdida de un TPDU de datos, lanza un temporizador aleatorio transcurrido el cual envía un NACK por

multicast a los miembros del grupo local, solicitando los TPDU de datos que le faltan y que no hayan sido solicitados por otros miembros. El resto de los miembros del grupo al recibir este NACK cancelan sus temporizadores y suprimen el envío del suyo, con ello se evita la implosión de NACKs en el grupo. El CG contesta retransmitiendo por multicast el TPDU que se ha perdido. Cuando el grupo local está compuesto por un solo miembro el NACK es sustituido por un HNACK y se envía sin realizar ninguna espera.
- *Asentimiento Jerárquico Negativo (HNACK):* utilizados para solicitar paquetes al grupo local de nivel superior. Sólo se envían después de n intentos fallidos de recuperación local. Este paquete se envía por **multicast** a los grupos locales de nivel superior, y en última instancia, tras agotar NACK y HNACK multicast, se envía por **unicast** al emisor del paquete. En HNACK enviado por multicast se suprimen las solicitudes ya realizadas por otros.

PTMF retransmite los datos solicitados utilizando un **TPDU de Retransmisión** diferente al normal, en este TPDU PTMF especifica de quién no ha recibido aún asentimiento, de esta manera se mantiene informado a los receptores ante posibles pérdidas de asentimientos. Cada fuente de datos identifica el fin de un flujo de información marcando el último TPDU con un **bit fin de transmisión**, para que los receptores puedan liberar recursos (ventana de recepción, contadores, etc) asignados a esa fuente. Por la posibilidad de que no se reciba el paquete con la marca de fin, o de que la fuente deje el grupo sin anunciarlo, los receptores consideran que una fuente se ha caído si transcurre un tiempo sin recibir datos de la misma. En el caso de que la fuente no tenga datos que enviar, cada cierto tiempo envía un paquete IDLE (paquete sin datos) para mantener notificados a los receptores de su existencia. PTMF proporciona un **bit fin de conexión** que activa el emisor fuente en el último TPDU de datos que envía al grupo multicast para anunciar que cierra la conexión.

2.3 Modo Fiable y Fiable Retrasado

PTMF es capaz de operar en dos modos: Fiable o Fiable Retrasado. En el modo Fiable cualquier receptor que se incorpore al grupo multicast más tarde intenta recuperar todos los TPDUs enviados por el emisor desde el inicio de la conexión. Es muy probable que el emisor ya no los tenga en la ventana, con lo que el nuevo miembro no recibirá los TPDU solicitados y será dado por caído por todos los CG al no enviar asentimientos. Para facilitar la incorporación de nuevos miembros al grupo multicast se establece otro modo de trabajo del protocolo denominado **Fiable Retrasado**. El protocolo es el mismo en el modo Fiable y Fiable Retrasado, lo único que cambia es el mecanismo de incorporación de nuevos miembros. En el modo fiable retrasado, no se asegura que todos los miembros reciban los datos desde el primer TPDU de datos de cada emisor, pero sí a partir de uno dado, el primer TPDU de la siguiente ráfaga, es decir, PTMF permite la incorporación de nuevos miembros a un grupo local en cada nueva ráfaga enviada por el emisor.

2.4 Control de flujo y de congestión

PTMF utiliza un control de flujo y de congestión básicos basados en ventana de emisión y control del número de solicitud de asentimientos. PTMF no es actualmente tcp-friendly [20]. PTMF puede ser configurado para ocupar un ancho de banda determinado. Los elementos que intervienen son:
- Ventana de emisión: cada emisor indica en cada paquete el tamaño de su ventana de emisión, el cual puede variar dinámicamente.
-Marcación de paquetes por los que se espera asentimiento: limitan el número de asentimientos que se tienen que recibir. Como caso extremo si el emisor marca todos los paquetes y el tamaño de ventana es de 1 se puede llegar a una situación de parada y espera. Cuando se detecta la congestión el emisor reducirá la ventana de emisión a la mitad de su tamaño y aumentará el temporizador de retransmisión. En el instante que se comience a recibir HACK el emisor irá aumentando la ventana en un TPDU por cada asentimiento recibido en un esquema de arranque-lento similar al realizado por TCP. Igual hará con el tiempo de retransmisión hasta que alcance una situación estable. En el lado del receptor el control de la congestión se realiza evitando que los receptores inunden la red con asentimientos negativos, el temporizador de retransmisión asociado a cada asentimiento negativo se incrementará exponencialmente con cada retransmisión hasta un límite máximo establecido.

2.5 Seguridad

PTMF proporciona soporte para la codificación de los TPDUs utilizando el algoritmo RC2 y codificando en el modo ECB (Electronic Code Book). En el modo ECB cada codificación RC2 no depende de las demás, de esta forma los TPDUs recibidos se pueden descodificar sin ningún tipo de problema, aun cuando los TPDUs lleguen desordenados o se pierdan algunos. PTMF también proporciona autentificación del TPDU mediante el algoritmo de hashing MD5. Si el nivel de aplicación desea proporcionar mayor seguridad siempre puede codificar los datos de usuario antes de pasarlos al nivel de transporte. El intercambio de las claves de los algoritmos de codificación recae sobre otros protocolos externos a PTMF.

3 Implementación de PTMF

PTMF ha sido desarrollado sobre el lenguaje de programación JAVA. Todas las clases que implementan el protocolo PTMF están agrupadas en un paquete denominado ptmf que dispone de varias clases de interfaz para acceder a toda la funcionalidad del protocolo de forma transparente. Se ha utilizado una notación similar a la utilizada en Java para acceder a los servicios de TCP (Socket) y de UDP (DatagramSocket). Dependiendo del modo de trabajo se han creado dos interfaces:

Interfaz de acceso	Modos de fiabilidad
SocketPTMF	FIABLE y FIABLE RETRASADO
DatagramSocketPTMF	NO FIABLE y NO FIABLE ORDENADO

Tabla 1. Interfaces de acceso a PTMF

4 Aplicaciones sobre PTMF

Para probar la funcionalidad de PTMF se han desarrollado dos aplicaciones, una para la transferencia de ficheros y otra para chat multicast. La aplicación para la transferencia de ficheros, denominada FTPMulticast, implementa un protocolo de nivel de aplicación muy simple diseñado para transferir ficheros por multicast utilizando el protocolo PTMF. Esta aplicación sirve de ejemplo tanto de aplicación 1:N como N:N. La aplicación para el chat multicast, denominada ChatMulticast, implementa un protocolo simple de nivel de aplicación para establecer un chat multicast sobre PTMF y permite probar las características de retardo del protocolo. Esta es una aplicación típica N:N.

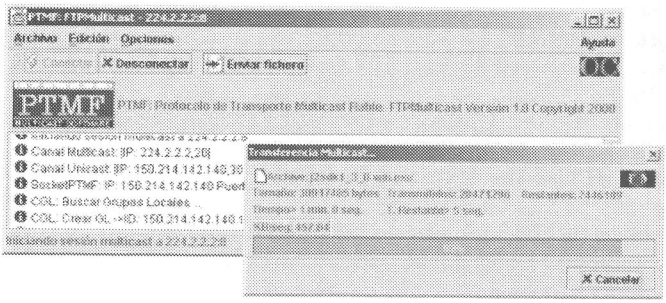

Figura 5. Aplicación FTPMulticast.

5 Conclusiones

PTMF es un protocolo de transporte multicast fiable diseñado para garantizar la fiabilidad tanto en entornos 1-N como N-N. Proporciona control de fiabilidad basado en grupos locales jerárquicos, con control de errores distribuido entre los miembros del grupo. PTMF utiliza asentimientos positivos y negativos, tanto multicast como unicast, realizando supresión en el envío en todos los asentimientos enviados por multicast, siendo acumulativos en el caso de los positivos y selectivos en el de los negativos. Además, permite la incorporación de miembros al grupo en cualquier instante, modo fiable retrasado. En lo que respecta al control de flujo y congestión, éste se basa en ventanas de emisión y recepción y en el control del número de asentimientos. En el caso de pérdida de paquetes

por los receptores, primeramente se intenta una recuperación local y en caso de fracaso se sube en la jerarquía arborescente en la búsqueda de los paquetes perdidos. La retransmisión de los paquetes se realiza por multicast o unicast. Por último, PTMF permite codificación de los TPDUs utilizando los algoritmos RC2 y MD5 y es totalmente parametrizable.

Referencias

1. S. Pingali, D. Towsley, J. Kurose. "A Comparison of Sender-Initiated and Receiver-Initiated Reliable Multicast Protocols", Proc. 1994 ACM SIGMETRICS Conf.
2. B. Levine and J.J. Garcia-Luna-Aceves, "A Comparasion of Known Classes of Reliable Multicast Protocols", Proc. Int'l Conf. Network Protocols, Oct. 1996.
3. W. Fenner. "Internet Group Management Protocol, Version 2". RFC 2236. Noviembre 1997.
4. Waitzman, D., Partridge, C. and S. Deering, "Distance Vector Multicast Routing Protocol", RFC 1075, Noviembre 1988.
5. Moy, J., "Multicast Extensions to OSPF", RFC 1584, Marzo 1994.
6. Estrin, D. et al. "Protocol Independent Multicast-Sparse Mode (PIM-SM): Protocol Specification", RFC 2362, Junio 1998.
7. Deering, S. et al. "Protocol Independent Multicast Version 2, Dense Mode Specification", Work in Progress.
8. Ballardie, A., "Core Based Trees (CBT version 2) Multicast Routing", RFC 2189, Septiembre 1997.
9. D. Thaler et al. "Border Gateway Multicast Protocol (BGMP)" Internet Draft, Work in Progress.
10. H. Eriksson, "MBone: the Multicast BackBone", Commun. ACM vol. 37, Agosto 1994, pp. 54 -60
11. Kevin C. Almeroth "The Evolution of Multicast: From the MBone to Interdomain Multicast to Internet2 Deployment", IEEE Network, vol. 14, n° 1, Enero 2000, pp. 10 -20
12. S. Floyd et al. "A Reliable Multicast Framework for Light-Weight Sessions and Application Level Framing", in Proc. ACM SIGCOMM'95, Boston, MA, pp. 342-56
13. John C. Lin, y Sanjoy Paul, "RMTP: A Reliable Multicast Transport Protocol". IEEE INFOCOM 1996, Mar. 1996, pp.1414-24
14. B. Whetten et al., "THE RMTP-II PROTOCOL", Internet draft, Apr. 1998, work in progress; http://www.talarian./rmtp-ii
15. Rajendra Yavatker, James Griffioen , y Madhu Sudan, "A Reliable Dissemination Protocol for Interactive Collaborative Applications", Proc. ACM Multimedia '95 Conf, Nov. 1995
16. M. Kadansky, D. Chiu, and J. Weslwy, "Tree-Based Reliable Multicast (TRAM)", Internet Draft, Nov. 1998, work in progress.
17. Markus Hofmann, "Enabling Group Communication in Global Networks". PhD Thesis. Junio 1997.
18. K. Miller et al. "MFTP: Multicast File Transfer Protocol". draft-miller-mftp-spec-02.txt, Enero 1997.
19. J. Gimmel, J. Gray, E. Schooler "Fcast Multicast File Distribution", IEEE Network, vol 14. n° 1. Enero 2000.
20. Jorg Widmer, Robert Denda, and Martin Mauve, "A survey on TCP-friendly congestion control", IEEE Network, vol. 15, pp. 28–37, Mayo-Junio 2001