Protocolo Multicast de Congestión Mínima

J. Ignacio Alvarez-Hamelin * y Luis Marrone.

Departamento de Electrónica, Facultad de Ingeniería de la U.B.A.,
Av. Paseo Colón 850, (1063) Ciudad de Buenos Aires, Argentina
e-mail: {ihameli,lmarro}@cactus.fi.uba.ar

Resumen Se propone un protocolo distribuído para comunicaciones del tipo *multicast*: MCT (*Minimal Congestion Tree*). Este protocolo genera un árbol que interconecta al emisor y a los nodos receptores de la comunicación *multicast*. El árbol obtenido es de congestión mínima, disminuyendo así la posibilidad de que se pierdan paquetes de información. La congestión es medida utilizando el modelo auto-similar del tráfico. Es este artículo se presentan los resultados de simulaciones que lo comparan con otros protocolos de *multicast*: QoSMIC, YAM y CBT.
Palabras clave: multicast, tráfico auto-similar, Internet II

1. Introducción

Las comunicaciones *multicast* tienen como objeto conectar a un grupo de usuarios (por ejemplo video en demanda, video conferencias, aulas virtuales, etc). En la mayoría de los protocolos existentes, el modelo de conexión de los usuarios se realiza mediante un árbol que interconecta a todos los miembros reenviando los mensajes a través de los enlaces correspondientes a dicho árbol [4,8].

Varios métodos fueron propuestos para la construcción de árboles de comunicaciones *multicast* ofreciendo una buena relación costo performance. Dentro de los protocolos distribuídos encontramos: los que utilizan la técnica del camino reverso más corto (*reverse shortest paths*): CBT [2] y PIM-SM [5], y los que utilizan la técnica glotona (*greedy*): YAM [3] y QoSMIC [6]. En la primera técnica se realiza el agregado de nodos mediante la búsqueda del camino más corto hacia un nodo central denominado "*core*", mientras que la segunda los nodos se agregan conectándose al punto más cercano del árbol. Estas dos aproximaciones están alejadas del caso óptimo, la primera porque no siempre la suma de caminos más cortos da un árbol mínimo, y la segunda porque la agregación y separación de miembros va dejando en el árbol enlaces que podrían ser eliminados.

En este trabajo se propone un protocolo denominado MCT (*Minimal Congestion Tree*), que por medio de activaciones periódicas construye un árbol de congestión mínima. La principal motivación de minimizar la congestión del árbol en redes tipo *best effort* reside en que los enlaces menos congestionados son los que menos paquetes descartan (por ejemplo ante una elevada congestión). En cambio, en redes basadas en calidad de servico (*QoS*), se podrá incrementar el número de conexiones totales utilizando el MCT.

* Participa en el Programa *αlpha Cordial*

Es importante y original destacar que la congestión se mide a través de la aplicación del modelo auto-similar del tráfico y de las colas generadas por éste.

El trabajo está organizado de la siguiente forma: una descripción sintética del MCT en la sección 2, luego se presenta el modelo de congestión utilizado en la sección 3, la sección 4 presenta las simulaciones, y finalmente las conclusiones en la sección 5.

2. Protocolo de Árbol de Congestión Mínima (MCT)

El protocolo MCT es distribuído y se ejecuta concurrentemente en cada nodo de la red. Está basado en el protocolo MPDR [1] (Multi-Path Distance Routing protocol), por lo tanto asume que todo nodo v de la red tiene asociada una tabla $\mathcal{D}_v$ tal que $\mathcal{D}_v[i, x]$ representa la longitud del camino más corto (en función del número de saltos) desde v a x, saliendo a través de la i-ésima interfase de v. El protocolo MCT consta de tres fases ejecutadas por cada nodo.

El algoritmo comienza en el nodo principal o en el nodo raíz s, que calcula la lista T, y envía los mensajes M_{expl} a los nodos vecinos. Esta primera fase explora para cada nodo x del grupo *multicast*, todos los caminos desde s a x de longitud a lo sumo $\varrho\, d(s, x) + r$ donde ϱ y r son constantes fijadas. Los vecinos hacen lo mismo hasta que se encuentre una lista T vacía. Luego comienza la fase de Selección, y el nodo s recibe toda la información y entonces calcula el árbol mínimo y comienza la fase de Construcción. Cuando finaliza, el algoritmo construye un árbol *multicast* de congestión mínima. El resultado de estas tres fases es un árbol $\mathcal{T}$ con raíz en el nodo principal s, y tal que, para cada nodo en el grupo, el camino desde s a x en $\mathcal{T}$ tiene una congestión mínima entre todos los caminos desde s a x de longitud a lo sumo $\varrho\, d(s, x) + r$. El protocolo en forma completa se indica en Algoritmo 1.

Los mensajes tienen diferentes parámetros. El mensaje M_{expl} de exploración tiene (T, L, m). T es una lista de elementos indexados por x (un nodo fuera del grupo *multicast*), como el árbol

$$T_{[x]} = \mathcal{D}_v[i, x] \leq \rho \min_{i=1,\ldots,\delta} \mathcal{D}_s[i, x] + r,$$

donde $\mathcal{D}_v$ es el camino de distancia más corta desde s a x pasando a través de la interfase i-ésima que tiene una longitud de a lo sumo ρ veces la distancia entre s y x más la constante r. L es la lista de los nodos visitados, y m es la identificación del grupo *multicast*. El M_{sel} (Fase de Selección) tiene (S, L, m). S es una lista de elementos $S_{[x]}$, tales que $S_{[x]}$ es la congestión mínima sobre todas las rutas posibles a través de los vecinos. En otras palabras,

$$S_{[x]} = \min_{i=1,\ldots,k} \max\{\lambda(v, u_i), S^i_{[x]}\},$$

donde $\lambda(v, u_i)$ es la congestión entre el nodo local v y el vecino u_ii, y $S^i_{[x]}$ es el elemento $Si_{[x]}$ del i-ésimo vecino.

Finalmente, la M_{cons} (Fase de Construcción) tiene (C, L, m). C es una lista de los nodos *multicast* a los que es necesario conectarse por esta rama del árbol *multicast*.

Algoritmo 1

Fase de exploración: $v \neq c$ recibe uno de éstos: $M_{expl} = (T, L, m)$ de w.
  Para todos los vecinos u_i de v, definir $T^i = [-1, \ldots, -1]$;
  Dado $L' = \{L, v\}$;
  Para cada vecino $u_i \notin L$ de v, hacer
    Para cada miembro x del grupo multicast, tal que $T[x] \neq -1$ hacer
      Si $|L| + \mathcal{D}_v[i, x] \leq \rho \cdot T[x] + r$, entonces $T^i[x] \leftarrow T[x]$;
    Si $T^i \neq [-1, \ldots, -1]$ entonces enviar (T^i, L', m) a u_i;
  Si $T^i = [-1, \ldots, -1]$ para todo i entonces
    Dado $S = [-1, \ldots, -1]$;
    Si v es un miembro del grupo, entonces $S[v] = 0$.
    Enviar $M_{sel} = (S, L, m)$ a w;
Fase de selección: Para todo i, v ha recibido un mensaje $M^i_{sel} = (S^i, L, m)$ a través de todas las interfases por las que había enviado un mensaje (T^i, L, m).
  Dado $S = [-1, \ldots, -1]$;
  Para cada miembro del grupo x hacer
    Si existe i tal que $S^i[x] \neq -1$ entonces
      $S[x] = \text{mín}_i \, \text{máx}\{\lambda(v, u_i), S^i[x]\}$;
      $I[x] = \text{argmin}_i \, \text{máx}\{\lambda(v, u_i), S^i[x]\}$;
  Si v es un miembro del grupo, entonces $S[v] = 0$.
  Si $v \neq s$ entonces
    $L = \{L', w\}$;
    Envíe $M_{sel} = (S, L', m)$ a w;
  En otro caso
    Para cada vecino u_i de s hacer
      $C := \emptyset$;
      Para cada miembro del grupo x hacer
        Si $I[x] = u_i$, entonces $C = (C, x)$;
      Si $C \neq \emptyset$ entonces enviar $M_{cons} = (C, \{s\}, m)$ a u_i;
Fase de construcción : $v \neq s$ receive un mensage $M_{cons} = (C, L, m)$.
  Asumiendo $L = \{L', w\}$;
  Para cada vecino $u_i \neq w$ de v hacer
    $C' := \emptyset$;
    Para cada nodo x en C hacer
      Si $I[x] = u_i$, entonces $C' = (C', x)$;
    Si $C' \neq \emptyset$ entonces enviar $M_{cons} = (C', \{L, v\}, m)$ a u_i;

3. Congestión de los Enlaces

La mayor parte de los ruteadores en la red tienen interfaces con varios enlaces. Cada uno de ellos puede representarse como una cola con una velocidad de servicio C_e, siendo C_e la capacidad del enlace e medida en bits por segundo.

Desde el punto de vista de los mensajes que se transmiten, la congestión del enlace puede medirse por el tiempo máximo que el mensaje tarda en atravesarlo: la suma del tiempo de tránsito y el tiempo de espera en la cola. La forma en que esa cola evolucione (y con ella la congestión en el enlace) dependerá del modelo de tráfico adoptado. Se propone el modelo auto-similar [7](*self-similar*) ya que refleja mejor la realidad. Observaciones exhaustivas de tráfico real [10,11] en redes de área local permitieron determinar que en una distribución de tráfico en el tiempo, si se varía la escala temporal el tráfico resulta *auto-similar* en un amplio rango de escalas. Esto se opone al modelo de Poisson en el que al tomar cada vez rangos de tiempo mayores el tráfico tiende a uniformizarse. En este modelo, la característica de ráfaga (*bursty*) del tráfico, no se pierde al variar la escala. A modo de ejemplo es importante recordar que los primeros conmutadores ATM que se utilizaron disponían de *buffers* cuya capacidad estaba determinada por el modelo de Poissson. Este modelo determina un tamaño menor que según el auto-similar. Se observó una mala performance en estos conmutadores producida por un descarte excesivo de celdas debido a la pronta saturación del *buffer*. Si bien, como indicamos anteriormente, los estudios iniciales se realizaron sobre tráfico *Ethernet*, se observa el mismo comportamiento en niveles superiores como de transporte (TCP) e inclusive a nivel de aplicación como ser tráfico de WEB, TELNET, FTP y video digital sobre ATM. En nuestro estudio utilizamos el modelo propuesto por Norros [9]. En conjunto, el tráfico queda entonces caracterizado por tres parámetros: μ, a y H donde a y H de alguna manera miden la *calidad* del tráfico y μ la *cantidad*. Norros [9] demostró que la longitud máxima Q_e de la cola de un enlace de capacidad C_e, que no es otra cosa que la cantidad máxima de paquetes en ella, está relacionada con el coeficiente ϵ de índice de paquetes perdidos en una cola (p. ej. 1 menos la probabilidad que la longitud de la cola exceda su tamaño; este parámetro debe fijarse) por la siguiente expresión: $Q_e = \left(\frac{H}{C_e-\mu}\right)^{\frac{H}{1-H}} (1-H)(-2a\mu \ln \epsilon)^{\frac{1}{2-2H}}$. La congestión $\lambda(e)$ se define como el tiempo total, tal que:

$$\lambda(e) = \frac{Q_e}{C_e} + \frac{1}{C_e} \simeq \frac{Q_e}{C_e}$$

4. Resultado de las simulaciones

Se han realizado simulaciones en dos redes diferentes. La primera es la red ARPANET (1995) que posee 47 nodos, en donde la congestión fue generada en forma aleatoria. La segunda simulación fue realizada en la red Abilene Figura 5, que pertenece al *backbone* de Internet II en los Estados Unidos, con 11 nodos y en la que se ha trabajado con mediciones reales de tráfico.

En las simulaciones se ha buscado comparar los distintos protocolos de *multicast* con el MCT. La técnica de camino más corto reverso (*reverse shortest path*), aparece en las figuras como `rsp`, como ejemplo de protocolo que utiliza esta técnica tenemos a CBT. La técnica glotona (*greedy*), `greedy` en las figuras, como representante tenemos al protocolo YAM. La técnica glotona con optimización de la congestión, `greedy-con` en las figuras; el representante es el protocolo QoSMIC.

Para el caso de la red ARPANET se generaron al azar 10 000 casos diferentes de congestión en la red, utilizando dos estadísticas distintas: la uniforme y la exponencial. Los valores de la congestión se fijaron arbitrariamente entre 0 y 100. Los gráficos representan el número de árboles con congestión menor o igual a c (el eje de las abscisas). En otras palabras, si realizamos la normalización obtendremos la función acumulativa de distribución de probabilidad. Para la red Abilene hemos utilizado el modelo definido en la sección 3 con $\epsilon = 0{,}01$. Se han calculado 389 situaciones diferentes de congestión en toda la red y se ha simulado los cuatro protocolos de *multicast*. En todos los casos las simulaciones realizadas con el MCT se utilizó $\rho = 2$ y $r = 1$. Los resultados de las simulaciones sobre la ARPANET se pueden ver en las figuras 1 y 2, que corresponden a la distribución uniforme y a la distribución exponencial respectivamente. Se puede observar que QoSMIC se desempeña mejor que CBT y YAM, pero MCT obtiene los árboles de menor congestión. Los resultados de la simulación sobre la red Abilene muestran que MCT obtiene árboles de menor congestión que los otros. Dado que la congestión se obtiene como la longitud máxima de la cola a la salida del ruteador en función de los parámetros del tráfico y de ϵ (ver sección 3), y que éste es un modelo teórico, los valores obtenidos de la congestión sólo deben verse como indicadores de lo que ocurre en la red. La figura 3 corresponde a un grupo *multicast* de 5 nodos, mientras que en la figura 4 el grupo *multicast* es de la totalidad de los nodos de la red.

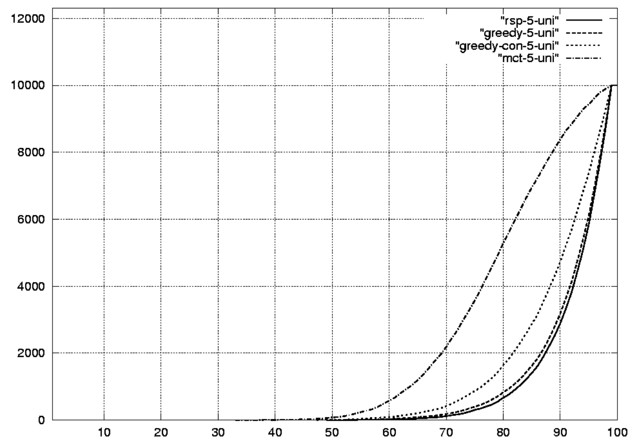

Figura 1. MCT comparado con QoSMIC, YAM y CBT: distribución uniforme. Número de árboles de congestión igual o menor a c (abscisas).

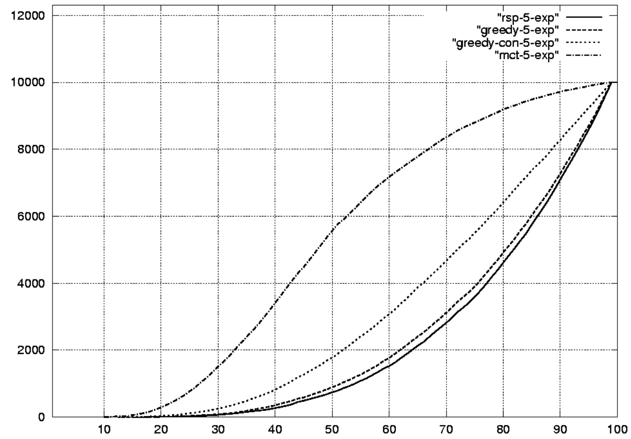

Figura 2. MCT comparado con QoSMIC, YAM y CBT: distribución exponencial. Número de árboles de congestión igual o menor a c (abscisas).

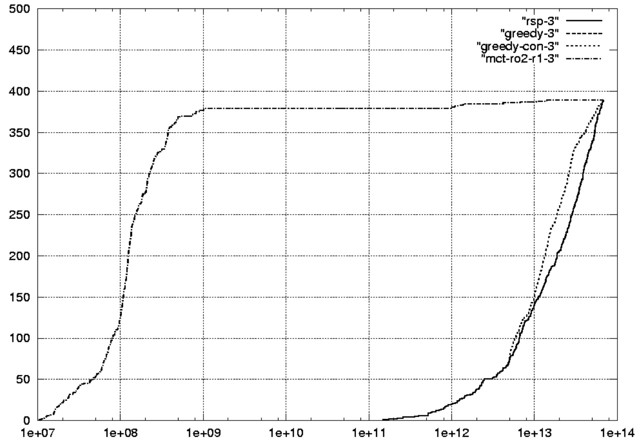

Figura 3. MCT comparado con QoSMIC, YAM y CBT: grupo de 5 nodos. Número de árboles de congestión igual o menor a c (abscisas).

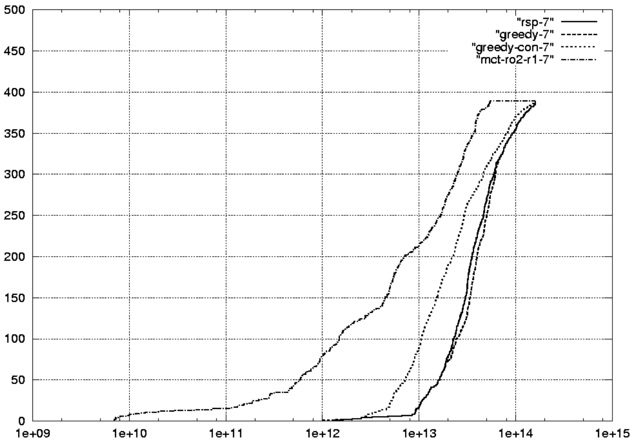

Figura 4. MCT comparado con QoSMIC, YAM y CBT: grupo de todos los nodos de la red: 11. Número de árboles de congestión igual o menor a c (abscisas).

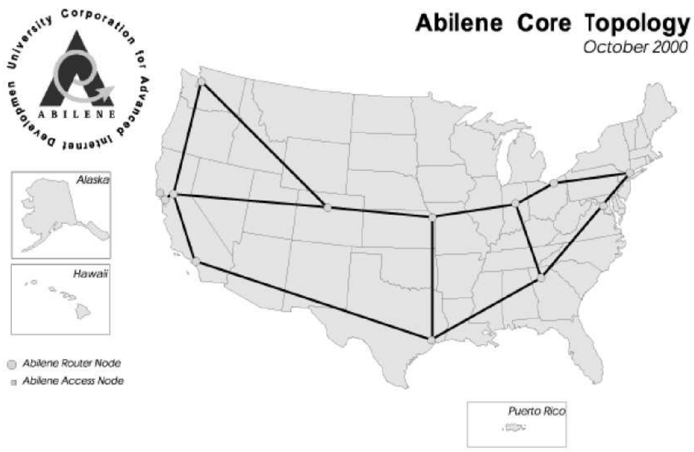

Figura 5. Red Abilene, *Core* de **Internet II**

5. Conclusión y trabajo futuro

Se ha presentado en forma sintética el protocolo MCT que construye árboles para las comunicaciones *multicast*. MCT utiliza el modelo de tráfico auto-similar para calcular la congestión de los enlaces, permitiendo la selección de los menos congestionados. Las simulaciones muestran que el protocolo MCT funciona adecuadamente y que se comporta mejor que los otros protocolos, como ser CBT y YAM, incluso que aquellos que poseen algún tipo de optimización, tal es el caso de QoSMIC

Se está trabajando para lograr una implementación del MCT utilizable en Internet. También se está comenzando a estudiar las implicancias de la utilización de tablas de ruteo reducidas.

Reconocimientos: Los autores desean agradecer especialmente a la Ing. Sara Tresens del Dep. de Electrónica de la Fac. de Ingeniería, por su invalorable ayuda en la estimación de los parámetros del modelo auto-similar. Tambien se desea agradecer a Brent Sweeny por facilitar las mediciones de tráfico sobre la red Abilene.

Referencias

1. J. I. Alvarez-Hamelin and P. Fraigniaud. Multicast Tree with Minimun Congestion . In *Conielecomp 2001*, pages 247–250. SAIEC, CENTIA, IEEE, Feb 2001.
2. T. Ballardie, P. Francis, and J. Crowcroft. Core based tree (CBT): an architecture for scalable inter-domain multicast routing. In *proceedings of SIGCOM*, pages 85–95. ACM press, 1993.
3. K. Carlberg and J. Crowcroft. Building Shared Trees using a one-to-many joining mechanism. In *ACM SIGCOMM Computer Communication Review*, pages 5–11, January 1997.
4. C. Diot, W. Dabbous, and J. Crowcorff. Multipoint Communication: A Survey of Protocols, Functions, and Mechanisms. *IEEE Journal on Seleted Areas in Communications*, 15(3):277–290, 1997.
5. D. Estrin, D. Farinacci, A. Helmy, D. Thaler, S. Deering, M. Handley, V. Jacobson, C. Liu, and L. Wei. RFC-2362: Protocol Independent Multicast-Sparse Mode (PIM-SM) . Technical report, IETF, 1998.
6. M. Faloutsos, A. Banerjea, and R. Pankaj. QoSMIC: Quality of Service sensitive Multicast Internet protoCol. In *SIGCOMM*, September 1998. Vancouver BC.
7. W. E. Leland, M. S. Taqqu, W Willinger, and D. V. Wilson. On the Self-Similar Nature of Ethernet Traffic. In *SIGComm '93*. ACM, 1993. San Francisco, USA.
8. W. Mostafa and M. Singhal. A taxonomy of multicast protocols for Internet applications. *Computer Communications*, 20:1448–11457, 1998.
9. I.Ñorros. On the Use of Fractional Brownian Motion in the Theory of Connectionless Networks. *IEEE Journal of Selected Areas in Communications*, 13(6):953–962, August 1995.
10. V. Paxon and S. Floyd. Wide-Area Traffic: The Failure of Poisson Modeling. *IEEE/ACM Transaction on Networking*, 3(3):226–244, June 1995.
11. P. A. Porras and A. Valdes. Live Traffic Analysis of TCP/IP Gateways. In *Networks and Distributed Systemd Security Symposium*. Internet Society, March 1998.