Multicast Tree with Minimum Congestion

J. Ignacio Alvarez-Hamelin[†], Pierre Fraigniaud[‡] and Alberto Dams[§]

L.R.I.,Bât 490 Université Paris Sud, 91405 Orsay Cedex France
Dep. Electrónica, Fac. de Ingeniería, U.B.A., Paseo Colón 850,(1316) Bs. As., Argentina

---

This paper describes a multicast protocol called MCT (Minimal Congestion Tree). Its periodic activation allows to construct a multicast tree spanning a group of users, such that the congestion along the route from the core to each member of the group is small. The congestion is measured by considering a self-similar statistic model of the Internet traffic. MCT can be used in arbitrary topologies. Simulations show that MCT performs better than QoSMIC and CBT as far as congestion in concerned.

Keywords: multicast, self-similar traffic, Internet

---

1 Introduction

As opposed to unicast protocols connecting pairs of users, multicast protocols support group communications (e.g., white board, video-on-demand, video-conferences, etc.) in which more than two users, possibly hundreds, interact. In most of the existing protocols, the structure of the multicast group is a tree connecting all the members, and messages exchanged among the group members are routed along the links of that tree [DDC97, MS98].

Several methods have been proposed for constructing multicast trees offering a good cost vs. performance ratio. A major difficulty comes from the dynamic of the group (members can join and leave at their convenience). On one hand, many algorithms have been derived to solve the *Steiner* tree problem. This problem is NP-complete but efficient $O(1)$-approximation algorithms have been derived. The main problem of this approach is that it is centralized and off-line. On the other hand, on-line decentralized protocols have been derived. These protocols can be roughly classified as *shortest paths* (e.g., CBT [BFC93], PIM-SM [EFH+98], and YAM [CC97]), or *greedy*, (e.g. QoSMIC [FBP98], and MIP [PGLA97]). In the former class, the tree is composed of the union of the (possibly reverse) shortest paths connecting some specific node (called "source" or "core") to each member of the group. In the latter class, a node joining a group is connected to the closest node in the current tree. The main problem of this second approach is that there are worse case join-leave sequences that make the resulting tree far from optimal.

In this paper, we propose a new protocol called MCT for *Minimal Congestion Tree*. Its periodic activation allows to construct a multicast tree that spans the group of users, and such that the congestion along the route from the core to each member is small. MCT is motivated by the fact that low congestion reduces packet loss in best-effort networks, or increases the number of accepted connections in QoS-based networks.

The congestion of the links resulting from MCT might not be *minimum* because there might exist long routes that are less charged. However, it is *minimal* in the sense that, if less charged routes exist, then they must be longer than a certain threshold. The length of the routes is controlled

---

[†]Dep. de Electrónica de la Facultad de Ingeniería, Univ. de Buenos Aires, and LRI, Univ. Paris-Sud. *ihameli@lri.fr*. Additional support from the *Alpha* program.

[‡]CNRS, LRI, Univ. Paris-Sud. *pierre@lri.fr*. Additional support from GdR A.R.P. of the CNRS, and from the RNRT project ROM.

[§]Departamento de Electrónica de la Facultad de Ingeniería, Univ. de Buenos Aires. *adams@cactus.fi.uba.ar*.

by two parameters: ϱ and r. MCT only considers routes from the core s to each member x whose length is at most $\varrho \cdot d(s,x) + r$, where $d(x,y)$ denotes the distance (i.e., number of hops) between two nodes x and y.

It is an original and important feature of the MCT protocol to take into account the congestion of the links. This congestion may be caused by dense traffic inside the network, or by the limited capacity of a link. In both cases, the link-congestion is estimated by using a self-similar model of the traffic. It results from that model that the queue-length is a good estimator of the congestion. MCT is therefore based on that property.

The paper is organized as follows. A very brief description of MCT is given in Section 2. Section 3 presents our network model for the estimation of the load of the links. Section 4 presents some comparisons between MCT on one hand, and QoSMIC and CBT on the other hand. Finally, Section 5 contains some concluding remarks.

2 Protocol MCT

MCT is a distributed protocol which runs concurrently at all the network nodes. MCT is built upon a protocol called MPDR for Multi-Path Distance Routing protocol. MPDR maintains a table $\mathcal{D}_v$ at each node v, such that $\mathcal{D}_v[i,x]$ is the length of the shortest path connecting node v to node x when leaving v through the i-th interface. MCT protocol consists of three phases executed by every node upon reception of a wake-up message sent by the core s to its neighbors.

The first phase is called *exploration phase*. It explores, for every node x of the group, all paths from s to x that are of length at most $\varrho \cdot d(s,x) + r$ where ϱ and r are constants fixed *a priori*, and $d(s,x)$ is the minimum distance in hops. The second phase is called *selection phase*. It selects, among all these paths, the path from s to x whose congestion is minimum. The third phase is called *construction phase*. It constructs the multicast tree according to the result of the selection phase. These three phases provide a tree $\mathcal{T}$ rooted at the core s, and such that, for every node x of the group, the path from s to x in $\mathcal{T}$ has a minimum load among all paths from s to x of length at most $\varrho\, d(s,x) + r$.

A formal description of the protocol MCT is given in Algorithm 1 at the end of the section. The current node is denoted by v and its neighbors are denoted by u_i. The congestion of link e is denoted by $\lambda(e)$. The congestion of a path $P = (e_1, e_2, \ldots, e_k)$ is defined as $\lambda(P) = \max_i \lambda(e_i)$. MCT selects, for each group member x, the path P from s to x of length at most $\varrho \cdot d(s,x) + r$, and of minimum congestion. (We use the maximum congestion over all links because this is the main cause of packets losses. However, other definitions of the congestion could be handled by the MCT protocol.)

The messages have different format according to the different phases:

Exploration messages: An exploration message M_{expl} is a triple (T, L, m). T is a distance table indexed by the nodes of the multicast group. L is a list of visited nodes. Finally, m is the identification of the multicast group.

Selection messages: A selection message M_{sel} is a triple (S, L, m). S is a congestion table indexed by the nodes of the multicast group.

Construction messages: The construction message M_{cons} is a triple (C, L, m). C is a list of group members.

The algorithm begins at the core s, in a diverge-cast fashion. Initially, s sends through its i-th interface the exploration message $(T^i, \{s\}, m)$ satisfying $T^i[x] = d(s,x) = \min_j \mathcal{D}_s[j,x]$ if $D_s[i,x] \leq \rho \cdot d(s,x) + r$, and $T^i[x] = -1$ otherwise. Upon reception of an exploration message, a node v looks for routes from s to every group member x, passing through v and of length at most $\rho \cdot d(s,x) + r$. If no such route exists, then the selection phase begins at v. This phase proceeds in a converge-cast fashion. Every intermediate node v receiving selection messages selects valid routes from s to every group member passing through v and of minimum load. When node s has completed its selection phase, the construction phase starts from s, again in a diverge-cast fashion aiming to inform nodes of the selected routes.

It is important to note that the algorithm has no loops. This is insured by the use of the list of visited nodes L included in all the messages of the MCT protocol.

Algorithm 1

Exploration phase: $v \neq c$ receives an exploration message $M_{expl} = (T, L, m)$ from w.
  For every v's neighbor u_i, define $T^i = [-1, \ldots, -1]$;
  Set $L' = \{L, v\}$;
  For every v's neighbor $u_i \notin L$ do
    For every group member x such that $T[x] \neq -1$ do
      If $|L| + \mathcal{D}_v[i, x] \leq \rho \cdot T[x] + r$, then $T^i[x] \leftarrow T[x]$;
    If $T^i \neq [-1, \ldots, -1]$ then send (T^i, L', m) to u_i;
  If $T^i = [-1, \ldots, -1]$ for all i's then
    Let $S = [-1, \ldots, -1]$;
    If v is a group member, then $S[v] = 0$.
    Send $M_{sel} = (S, L, m)$ to w;
End exploration phase.

Selection phase: for all i, v has received a message $M^i_{sel} = (S^i, L, m)$ through all interfaces through which it sent an exploration message (T^i, L, m).
  Let $S = [-1, \ldots, -1]$;
  For every group member x do
    If there exists i such that $S^i[x] \neq -1$ then
      $S[x] = \min_i \max\{\lambda(v, u_i), S^i[x]\}$;
      $I[x] = \text{argmin}_i \max\{\lambda(v, u_i), S^i[x]\}$;
  If v is a group member, then $S[v] = 0$.
  If $v \neq s$ then
    $L = \{L', w\}$;
    Send $M_{sel} = (S, L', m)$ to w;
  Otherwise
    For every s's neighbor u_i do
      $C := \emptyset$;
      For each group member x do
        If $I[x] = u_i$, then $C = (C, x)$;
      If $C \neq \emptyset$ then send $M_{cons} = (C, \{s\}, m)$ to u_i;
End selection phase.

Construction phase: $v \neq s$ receives a message $M_{cons} = (C, L, m)$.
  Assume $L = \{L', w\}$;
  For every v's neighbor $u_i \neq w$ do
    $C' := \emptyset$;
    For each node x in C do
      If $I[x] = u_i$, then $C' = (C', x)$;
    If $C' \neq \emptyset$ then send $M_{cons} = (C', \{L, v\}, m)$ to u_i;
End construction phase.

---

3 Estimating the Congestion of the Links

Each link of a router can be modeled as a queue with service rate equal to the capacity of the link measured in bits per second. From the point of view of a packet, the congestion of a link could be measured by the time it takes to traverse that link. We define the congestion of a link as the time spent in the queue plus the latency of the link. We assume the traffic to be self-similar [LTWW93]. This assumption is preferred to the Poisson model based on observations of real traffic [PF95, PV98]. Indeed, real traffic presents fractal behavior [LTWW93], that is, at every time scale (from a few milliseconds to hours), one can observe similar-looking traffic bursts. In other words, we have the same statistical characteristics at different time scales. This is typical of self-similar processes. More precisely, we model the traffic $A(t)$ by using a *fractional Brownian traffic* $Z(t)$ [Nor95]. $A(t)$ represents the total amount of bits offered to an input queue during the time-interval $[0,t)$ (with $A(0)=0$). It satisfies $A(t)=\mu t+\sqrt{a\mu}\,Z(t)$ where μ is the average of the traffic, and a is the variance of the traffic. $Z(t)$ is a stochastic process with $t\in]-\infty,+\infty[$ that is characterized by the so-called Hurst (or self-similarity) parameter H, $H\in[\frac{1}{2},1)$.

Therefore, we assume the traffic to be characterized by three parameters: μ, a, and H. In some way, a and H measure the *quality* of traffic, and μ measures the *quantity* of traffic. Norros [Nor95] has shown that the maximum length Q_e of the queue of a link e of capacity C_e, that is the maximum number of packets in the queue, is related to the coefficient ϵ of the packet loss ratio at a queue (i.e., 1 minus the probability that the length of the queue e exceeds its size). This relation can be expressed as

$$Q_e=\left(\frac{H}{C_e-\mu}\right)^{\frac{H}{1-H}}(1-H)(-2a\mu\ln\epsilon)^{\frac{1}{2-2H}}.$$

The congestion $\lambda(e)$ of link e is then estimated by:

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

4 Simulation results

We have performed simulations of two different networks. The first network is ARPANET (1995), including 47 nodes, in which the traffic was generated randomly. The second network is the core of Universities Interconnection Network of Argentine (2000), including 7 nodes connected as showed on Figure 1, and in which the traffic was obtained from real measurements.

Simulations were performed for different types of multicast trees. More specifically, we were interested in comparing our protocol MCT (`mct` in the figures) with

1. the reverse shortest path technique (`rsp` in the figures) used, e.g., in CBT
2. the greedy technique (`greedy` in the figures) used, e.g., in YAM and
3. the greedy technique with congestion-optimization (`greedy-con` in the figures) used, e.g., in QoSMIC.

For the ARPANET network we have generated 10,000 different configurations of congestion inside the network. For each configuration, the congestion of each link was chosen randomly between 0 and 100 according to two statistical laws: uniform and exponential. The congestion of a tree is measured as the maximum congestion of all the links of the tree. Simulations were performed with a multicast group of 5 nodes, including the root, that is about 10% of the total number of nodes in network. Each of the 10,000 configurations results in a tree of some congestion between 1 and 100. For every c, $1\le c\le 100$, we have counted the number n of trees of congestion less or equal to c. In other words, if we divide n by 10,000, we obtain a cumulative distribution of probability.

For the Universities Interconnection Network of Argentine we have estimated the congestion using the real traffic measurements and the self-similar traffic model as described at Section 3 with $\epsilon=0.01$. We considered 550 different configurations of congestion inside the network. Each of the 550 configurations results in a tree of some congestion normalized between 0 and 100. The figures represent the number n of trees with congestion less or equal to c, as a function of c. In other words, if we divide n by 550, we obtain a cumulative distribution of probability. We used groups of size 2, 3, and 4 (denoted `1`, `2`, and `3` in the figures). These groups represent 29%, 43% and 57% of the total nodes respectively.

In all cases, we set $\rho = 2$ and $r = 1$.

The results of our simulations of the ARPANET network are given in Figures 2 and 3, corresponding to the uniform distribution and the exponential distribution respectively. We observe that QoSMIC performs better than CBT and YAM as far as congestion is concerned, in particular for the uniform distribution (see Figure 2). The results for the Universities Interconnection Network of Argentine are given in Figures 4, 5 and 6, corresponding to the groups of 2, 3 and 4 nodes respectively. These figures show that MCT trees are of lower congestion in all cases. Actually, one can see on Figures 4 and 5 that CBT, YAM and QoSMIC offer the same behavior (the curves are superimposed). This is not true in Figure 6 which corresponds to a group of 4 nodes.

In all cases, for simulations based on real traffic as well as on random traffic, MCT performs better than known protocols, and even significantly improves the performances of QoSMIC.

5 Conclusion and Future works

We have briefly presented the protocol MCT aiming to construct multicast trees with minimum congestion. MCT uses a self-similar model for the traffic in order to estimate the congestion from the occupation of the queues. Simulations show that MCT performs significantly better than other standard protocols (e.g., YAM or CBT), even those including some optimization for the congestion (e.g., QoSMIC).

We are currently working on simulations using statistics obtained from real Internet traffics. We are also currently studying the feasibility of implementing MCT in Internet, i.e., we aim to provide the specifications of MCT for the Internet.

Acknowledgments The authors are thankful to Sara Tressens from Dep. de Electrónica, Facultad de Ingeniería, University of Buenos Aires, for her help on self-similar traffic.

References

[BFC93] 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.

[CC97] 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.

[DDC97] 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.

[EFH+98] 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.

[FBP98] M. Faloutsos, A. Banerjea, and R. Pankaj. QoSMIC: Quality of Service sensitive Multicast Internet protoCol. In *SIGCOMM*, September 1998. Vancouver BC.

[LTWW93] 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.

[MS98] W. Mostafa and M. Singhal. A taxonomy of multicast protocols for Internet applications. *Computer Communications*, 20:1448–11457, 1998.

[Nor95] I. Norros. 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.

[PF95] V. Paxon and S. Floyd. Wide-Area Traffic: The Failure of Poisson Modeling. *IEEE/ACM Transaction on Networking*, 3(3):226–244, June 1995.

[PGLA97] M. Parsa and J. J. Garcia-Luna-Aceves. A Protocol for Scalable Loop-Free Multicast Routing. *IEEE Journal on Selected Areas in Communications*, 15(3):316–331, April 1997.

[PV98] P. A. Porras and A. Valdes. Live Traffic Analysis of TCP/IP Gateways. In *Networks and Distributed Systemd Security Symposium*. Internet Society, March 1998.

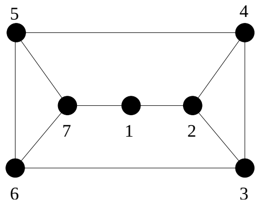

Fig. 1: Core of Universities Interconnection Network of Argentine (2000)

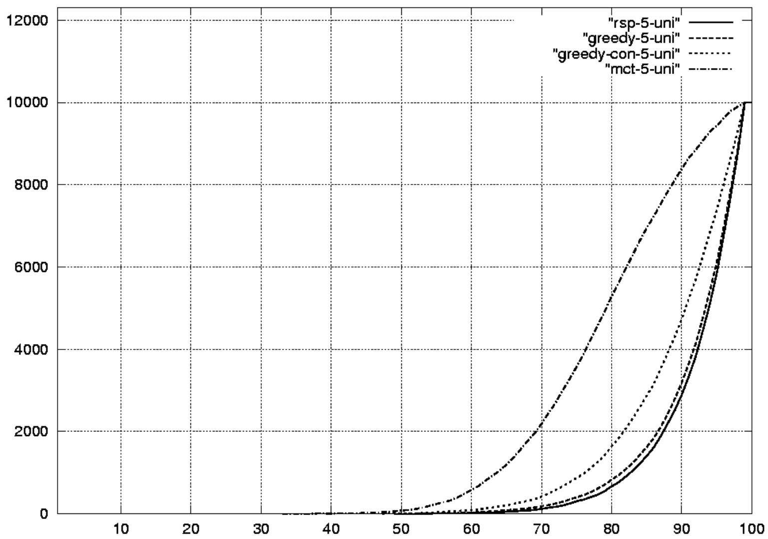

Fig. 2: MCT against QoSMIC, YAM and CBT: uniform distribution. For every c, number n of trees of congestion $\leq c$.

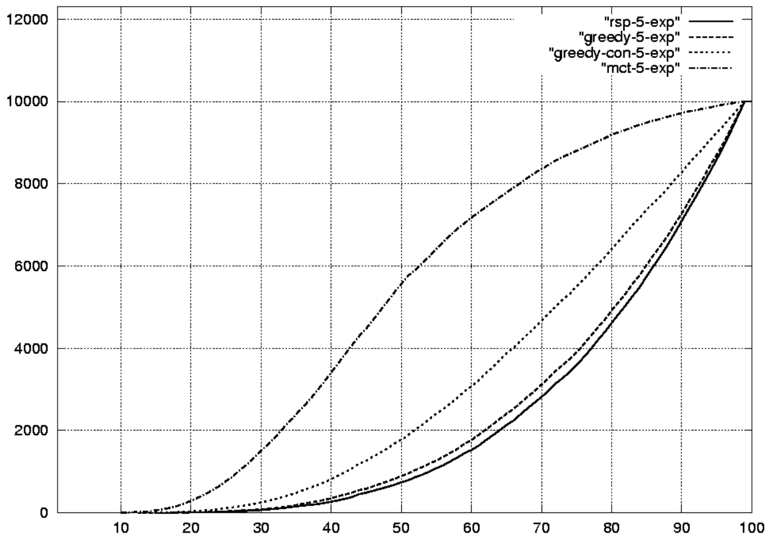

Fig. 3: MCT against QoSMIC, YAM and CBT: exponential distribution. For every c, number n of trees of congestion $\leq c$.

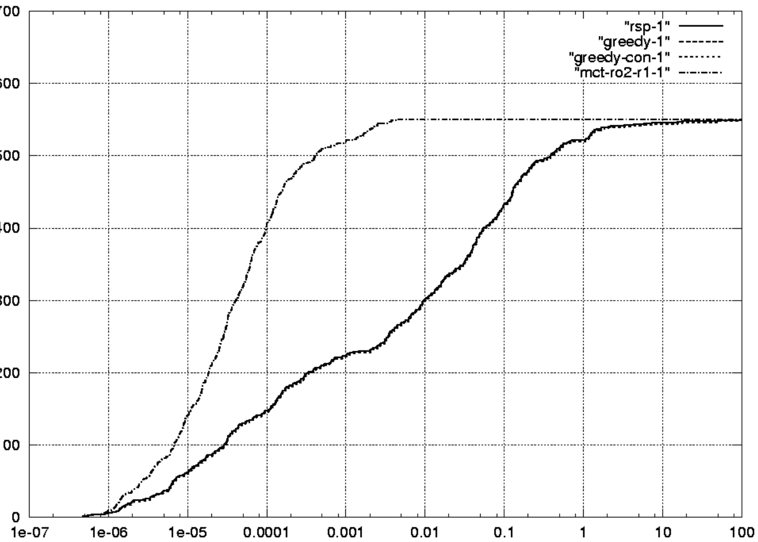

Fig. 4: MCT against QoSMIC, YAM and CBT: group of 2 nodes. For every c, number n of trees of congestion $\leq c$.

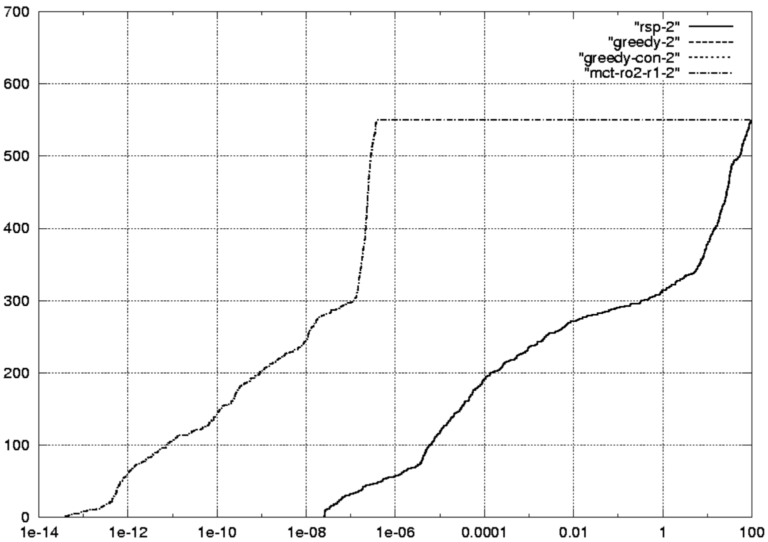

Fig. 5: MCT against QoSMIC, YAM and CBT: group of 3 nodes. For every c, number n of trees of congestion $\leq c$.

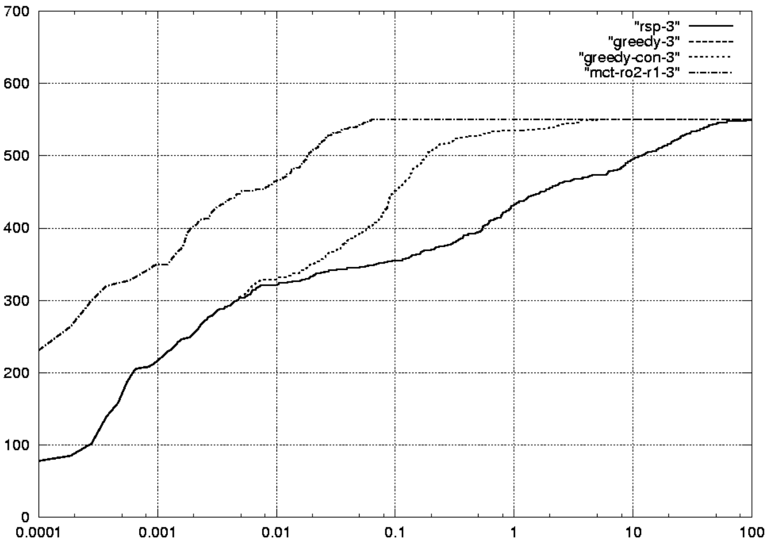

Fig. 6: MCT against QoSMIC, YAM and CBT: group of 4 nodes. For every c, number n of trees of congestion $\leq c$.