Multicast Tree with Minimum Congestion

J. Ignacio Alvarez-Hamelin* and Pierre Fraigniaud†

Departamento de Electrónica
Facultad de Ingeniería
Universidad de Buenos Aires
Paseo Colón 850
(1063) Buenos Aires ,Argentina
Tel : ++54 11 4343 0893, int 273
Fax: ++ 54 11 4342 6272

Laboratoire de Recherche
en Informatique–CNRS
Université Paris-Sud
Bâtiment 490
91405 Orsay, France
Ph : +33 (0)1.69.15.69.06
Fax: +33 (0)1.69.15.65.86

e-mail: {ihameli,pierre}@lri.fr

Abstract

We propose a multicast distributed protocol: MCT (Minimal Congestion Tree), whose periodic activation allows to construct multicast tree spanning the current group of users, such that the congestion along the route from the core to each group member is minimal. The congestion is measured considering the self-similar statistic model of the Internet traffic. This protocol can be used in arbitrary topologies. We present some simulations comparing the behavior of our protocol to QoSMIC or CBT with respect to congestion.

1 Introduction

As opposed to unicast protocols connecting pairs of users, multicast protocols support group communications, e.g., white board, video-on-demand, and video-conferences, in which more than two users, possibly hundreds of users, 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/performance ratio. The main difficulty comes from the dynamic of the group in the sense that members can join and leave at their convenience. On one side, many algorithms have been derived to solve the *Steiner* tree problem[1]. 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. One the other side, 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 PaGa-97. In the former class, the tree is composed of the union of the shortest paths between some node, called "source" or "core", and the members of the group. In the latter class, a node joining a group is connected to the closest node of the current tree. The main problem of this approach is that there are worse case join-leave sequences that make the resulting tree far from optimal [FV97].

In this paper, we propose a distributed protocol, called MCT for *Minimal Congestion Tree*, whose periodic activation allows to construct a multicast tree spanning the current group of users such that the congestion along the route from the core to each group member is minimal. The congestion is not minimum because there might exist longer routes that are less charged, but it is minimal in the sense that, if less charged routes exist, they must be much longer that the shortest path. The length of the routes is controlled by two parameters ϱ and r such that we do not consider routes from the core s to a member x which exceed $\varrho\ d(s,x) + r$ where $d(x,y)$ denotes the distance (in term of number of hops) between two nodes x and y. Minimizing the congestion reduces packet loss in best-effort networks, or increases the multicast connections accepted in QoS-based networks.

It is an original and important property of the MCT protocol to take into account not only the prop-

*Additional support from the *Alpha* program.

†Supported by the program A.R.P. of the CNRS, and by the French RNRT program "ROM".

[1] Given a edge-weighted graph $G = (V, E)$ and a set $D \subseteq V$, a Steiner tree is a tree spanning D and of minimum weight.

agation delay of the links or the inter-node distance, but rather the congestion of the links. This congestion may be caused by dense traffic inside the network, or by the limited capacity of a link. We have focused our attention on the congestion because packet loss has a dramatic effect on the performance of multicasting, and thus avoiding hot-spots significantly improves the quality of the multicast protocol. The congestion of the links is estimated based on a self-similar modeling of the traffic. It results from the model that the queue length is a good estimator of the congestion, and the MCT protocol is highly based on that property.

The paper is organized as follows: Section 2 introduces the MPDR protocol upon which is constructed the MCT protocol, introduced in Section 3. Section 4, introduces the traffic and network model used to estimate the load of the links. Section 5 discusses the efficiency of MCT compared to known multicast protocols. Finally, Section 6 contains some concluding remarks.

2 Multi-Path Distance Routing (MPDR) Protocol

The Multi-Path Distance Routing protocol maintains the routing information at each node of the network. Using the different interfaces of the router or node, the MPDR protocol considers multiple paths on each interface.

The MPDR protocol is implemented as a distributed algorithm. It generates a table of $\mathcal{D}_v[i,x]$ values at each node, which are the shortest distances in hops, from the local node v to a node x through the i-th interface of v.

The MPDR protocol runs at all nodes of the network. The MPDR protocol has two phases: the start phase, in which the local node wakes-up and retrieves the informations from its neighbors, and the second phase when it waits for modifications in the network and resend them to the others neighbors if they are relevant.

In the first phase the local node v starts sending notification messages to all the neighbors (i.e. all the nodes directly connected with v). The local node then waits for responses and builds the D table.

The second phase consists on waiting and processing update and link-up/down messages.

For the update message, node v makes the necessary changes at D table, looks for differences and resends them to the appropriate neighbor, as it does in the first phase.

For the link-down messages, node v deletes all the entries at D in relation with this interface, and informs all neighbors.

For the link-up messages, node v does the same as in the first phase, sends a notification to the new neighbor, waits for new routes through it, and rebuilds the D table, sending the differences to the other neighbors.

Algorithm 1

1. Send a notification to the neighbors through all the available links.
2. Wait the routing information from all neighbors, and build the D table.
3. Search at D the minimum distance for all the destinations and send them to all neighbors except to the neighbor which is the next hop in each of these routes.
4. Wait for the updates and link-up/down messages of neighbors.
5. **Updates:** put it into the D table, and proceed as 3 step, then return to step 4.
6. **Link down:** delete all the entries in D of this interface, and inform that these routes are not longer valid to the neighbors. Return to step 4.
7. **Link up:** send a notification to this new neighbor and proceed as 2 and 3 steps, then return to step 4.

3 Minimal Congestion Tree (MCT) Protocol

The MCT protocol is built upon the MPDR protocol, that is it assumes that every node v of the network has a table $\mathcal{D}_v$ such that $\mathcal{D}_v[i,x]$ returns the length of the shortest path (in term of number of hops) from v to x that is leaving v through the i-th interface of v. The MCT protocol consists of three phases executed by every node upon reception of a wake-up message broadcasted by the core s to its neighbors, and forwarded throughout the whole network. The first phase, called *exploration phase* explores, for every node x of the multicast group, all paths from s to x of length at most $\varrho\ d(s,x) + r$ where ϱ and r are fixed constants. The second phase, called *selection phase*, is selects, among all these paths, the path from s to x whose congestion is minimum. The last phase, called *construction phase* constructs the multicast tree according to this selection. The result of these three phases is a tree $\mathcal{T}$ rooted at the core s, and such that, for every node in 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 $\varrho\, d(s,x) + r$.

The protocol is showed at Algorithm 2. The local node is noted as v, the parent node is w, the neighbors are u_i.

Algorithm 2

1 **Wake-up phase:** v receives a $M_{wu} = (T, L, m)$ message.
2 Set for all i, $T^i = \emptyset$
3 Set $L' = \{L, v\}$
4 for each neighbor $u_i \notin L$ do
5 for each element of T do
6 if $\mathcal{D}_v[i,x] \leq \rho\, T_{[x]} + r$, then
7 Set $T^i_{[x]} = T_{[x]}$
8 for all u_i and $T^i_{[x]} \neq \emptyset$ do
9 Send to u_i the $M_{wu} = (T^i, L', m)$ message
10 if $T^i = \emptyset\ \forall i$ and if $v \in T$, then
11 Send to w the $M_{sp} = (S, L, m)$ message
12 **Selection phase:** v receives all the $M_{sp} = (S^i, L, m)$ messages, as responses of each M_{wu} sent message, by the i-th interface, and for all the neighbors.
13 $\forall\, T_{[x]}$, compute

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

14 Set $I_{[x]} = u_i$, where i is the node interface which minimize the later expression
15 Send to w $M_{sp} = (S, L, m)$ message
16 Remember (S, L, m) just up to the end of the construction phase
17 **Construction phase:** v receives a $M_{cp} = (C, L, m)$ message.
18 for each neighbor u_i do
19 for each node in C do
20 if $I_{[x]} = u_i$, then
21 $C' = \{C', x\}$
22 if $C' \neq \emptyset$ send to u_i the $M_{cp} = (C', L, m)$

The messages have different parameters. The M_{wu} (wake-up) have (T, L, m). T is a list of elements indexed by x (a node beyond the multicast group), such as the

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

where $\mathcal{D}_v$ is the shortest path distance from s to x passing through the i-th interface has a length at most ρ times the distance between s and x plus the constant r. L is the list of visited nodes, and m is the identification of the multicast group.

The M_{sp} (selection phase) have (S, L, m). S is a list of $S_{[x]}$ elements, such that $S_{[x]}$ is the minimum congestion over all possible routes thought the neighbors. In other words,

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

where $\mathcal{CO}(v,u_i)$ is the congestion between the local node v and the neighbor u_ii, and $S^i_{[x]}$ is the $Si_{[x]}$ element from the i-th neighbor.

Finally, the M_{cp} (construction phase) have (C, L, m). C is a list of the multicast nodes which are necessary to connect by this branch of multicast tree.

The algorithm begins at core o or root node s, which is charge of calculating the T list, and sends to the same node, as a ghost node, the first message: $M_{wu} = (T, L, m)$. Neighbors do the same until an empty T list is found. Then the selection phase begins, and node s receives all the information and resends the $M_{sp} = (S, L, m)$ message to w which is a ghost node. . It receives the messages and begins the construction phase. The algorithm running at the ghost node have little differences because it knows it is the first node. At the end, tha algorithm builds a multicast tree with minimum congestion.

4 Getting the Congestion of Links

A router in the network has several links. Each of one can be represented as a queue with μ service rate, where μ is the capacity of the link measured in bits per second. From the point of view of the packet, the congestion of a link could be measured by the time the packet takes to traverse it. Then, the congestion of the link is defined as the maximum time spent at the queue plus the bit traverse time through the link channel. We assume the network traffic to be statistically self-similar [LTWW93]. This assumption is preferred to the Poisson based on observations of real traffic [PF95, PV98]. Indeed, real traffic has a fractal behavior [LTWW93], and, 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, and 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, $H \in [\frac{1}{2}, 1)$.

Altogether, the traffic can be characterized by three parameters: μ, a, and H, where a and H somewhat 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 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 exceeds its size, this is a parameter which must be fixed) by:

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

The congestion is defined as the total time, then:

$$\mathcal{CO}_e = \frac{Q_e}{C_e} + \frac{1}{C_e} \simeq \frac{Q_e}{C_e} \qquad (1)$$

5 MCT vs. Existing Multicast Protocols

We worked with the ARPANET network at 1995. This network has 47 nodes and it is a good example of a real network. We generated 10000 different configurations of congestion inside the network. For each configuration the congestion of each link was chosen randomly according to a statistical law. We used two statistical laws: normal and exponential distributions. The simulation was done for different strategies for building the multicast tree. We were interested in comparing the reverse shortest path (named sal-rsp) as CBT, the greedy (named sal-greedy) as YAM, the greedy with congestion searching (named sal-greedy-con) as QoSMIC, and our protocol MCT (named sal-mct). The congestion of a tree is measured as the maximum congestion of all the links inside the tree. Simulations were performed with a multicast group of 5 nodes, including the root, that is the 5^{th} percent of the total nodes at network. We show the results of the two statistical laws: the normal distribution in Figure 1, the exponential distribution in Figure 2. The figures show the number of trees as function of congestion, that is the relative frequency of occurrence of a tree of x congestion into that network. In others words, if we check for instance the congestion 50, we find that there is nearly 220 trees of 10000 for MCT, which have this congestion. If we divide the y axe by 10000, we obtain the congestion probability function for that network and that group of users. As a result it is important to remark that MCT has a better behavior than other protocols since for nearly 80% of the cases a smaller congestion is achieved.

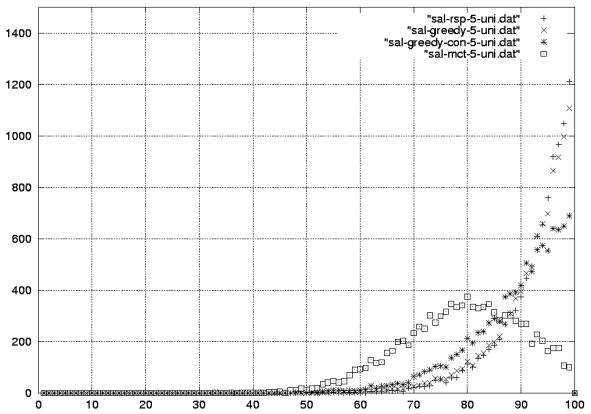

Figure 1: Normal Distribution

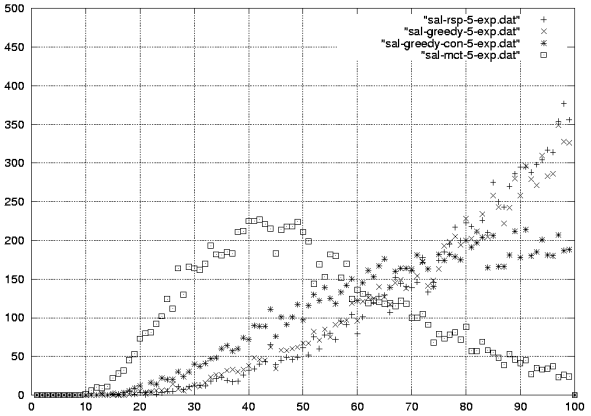

Figure 2: Exponential Distribution

6 Conclusion

We presented an MCT protocol that finds a multicast tree of minimum congestion. Congestion is computed according to a self-similar statistic traffic model. Simulations using the MCT protocol on real network traffic data (as ARPANET) show that it performs better as compared to other protocols. Future work will involve simulations using real parameterized traffic and a study on the feasibility of implementing the MCT protocol for Internet networks.

References

[BFC93] T. Ballardie, P. Francis, and J. Crowcroft, *Core based tree (CBT): an*

architecture for scalable inter-domain multicast routing, proceedings of SIGCOM, ACM press, 1993, pp. 85–95.

[CC97] K. Carlberg and J. Crowcroft, *Building Shared Trees using a one-to-many joining mechanism*, ACM SIGCOMM Computer Communication Review, January 1997, pp. 5–11.

[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** (1997), no. 3, 277–290.

[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)* , Tech. report, IETF, 1998.

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

[FV97] P. Fraigniaud and S. Vial, *Heuristic Algorithms for Personalized Communication Problems*, SIROCCO '97, 1997, pp. 240–252.

[LTWW93] W. E. Leland, M. S. Taqqu, W Willinger, and D. V. Wilson, *On the Self-Similar Nature of Ethernet Traffic*, SIGComm '93, ACM, 1993, San Francisco, USA.

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

[Nor95] I. Norros, *On the Use of Fractional Brownian Motion in the Theory of Connectionless Networks*, IEEE Journal of Selected Areas in Communications **13** (1995), no. 6, 953–962.

[PF95] V. Paxon and S. Floyd, *Wide-Area Traffic: The Failure of Poisson Modeling*, IEEE/ACM Transaction on Networking **3** (1995), no. 3, 226–244.

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