Directed Trees in Multicast Routing

Maria João Nicolau[1], António Costa[2], Alexandre Santos[2], and Vasco Freitas[2]

[1] Departamento de Sistemas de Informação,
Universidade do Minho, Campus de Azurém,
4800 Guimarães, Portugal
joao@uminho.pt
[2] Departamento de Informática,
Universidade do Minho, Campus de Gualtar,
4710 Braga, Portugal
{costa,alex,vf}@uminho.pt

Abstract. Traditional multicast routing protocols use RPF (Reverse Path Forwarding) concept to build multicast trees. This concept is based upon the idea that an actual delivery path to a node is the reverse of the path from this node to the source. This concept fits well in symmetric environments, but in a routing environment where Quality of Service is considered the guarantee that a symmetrical path will exist between two network addresses is broken. Available network resources impose specific Quality of Service asymmetries, therefore reverse path routing may not be used.
In this paper a new multicast routing strategy is proposed, enabling directed trees establishment, instead of reverse path ones. This new strategy, DTMP- Directed Trees Multicast Routing, is then implemented and simulated using Network Simulator. Simulation results, driven from several scenarios are presented, analyzed and compared with PIM-SM.

1 Introduction

Low communication costs, rapid deployment, and the ability to deal with almost every media type (namely audio, video and even video conferencing) make multicast applications very useful tools and so, large ISPs, are now supporting native multicast access inside their backbones.

Many applications in the Internet, such as video-conference, distance learning and other Computer Supported Cooperative Work (CSCW) applications require multicast support from the underlying network. These applications interconnect multiple users (several sources and receivers), exchanging large streams of data and thus an efficient use of network resources is needed. Multicast communication is the best way to send the same data simultaneously, and in an efficient way, to a non-empty set of receivers without incurring into network overloads. Hence, at each multicast-interested router, only a single copy (per group) of an incoming multicast packet is sent per active link, rather than sending multiple copies (per number of receivers accessed) via that link.

Routing multicast traffic requires building a distribution tree (or set of trees). Data packets are delivered using that tree, thus the major goal of the routing protocol is to build a tree with minimum cost (in what respects to some set of parameters). The problem of finding such a tree is NP-complete and is known as *Steiner Tree Problem*[1] and plenty of heuristics have been proposed to efficiently find multicast trees. The most commonly used heuristic consists of building a spanning tree by adding each participant at a time, by means of finding the shortest path from the new participant into the nearest node of the spanning tree. Such a tree is called *Reverse Path Tree*. This heuristic assumes that links connecting any two nodes are symmetric, in other words, assuming that link costs, in either direction, are equal.

However, when routing constraints are introduced there is no guarantee that this would be the case. Links may be asymmetric in terms of the quality of service they may offer, thus link costs are likely to be different in each direction. Therefore reverse path routing is not adequate to address Quality of Service Routing.

The Protocol Independent Multicast-Sparse Mode (PIM-SM)[2] is a widely deployed multicast routing protocol, designed for groups where members are sparsely distributed over the routing domain. It is based upon the concept of Rendez-Vous Points (RP), pre-defined points within the network known by all routers. A router with attached hosts interested in joining a multicast group will start a multicast tree by sending a join message on the shortest path to the RP. This join message is processed by all the routers in between the new receiver and the first in-tree node and a new branch for the new member is setup within the multicast tree.

PIM-SM has important advantages when compared to other multicast routing protocols: it does not depend on any particular unicast routing protocol and source rooted trees may be used, instead of the shared tree, if the data rate of a source exceeds a certain threshold. However, PIM-SM assumes symmetric routing paths as it uses reverse-path routing and thus it is not suited for use in conjunction with Quality of Service Routing.

In this paper, a new multicast routing protocol is proposed called DTMP (Directed Trees Multicast Protocol), inspired in PIM-SM that takes into account link asymmetry. Here, *directed-tree* based routing strategy as opposite to a *reverse-path-tree* based one is defined and tested.

2 Related Work

Most of previous works in this area assume that links connecting any two nodes are symmetric. The underlying networks are usually modeled by undirected graphs and the heuristics used address the Steiner Tree Problem in symmetric networks.

Finding a minimal multicast tree in asymmetric networks, called the *Direct Steiner Tree Problem*, is also NP-complete. There are some theoretical studies [3], [4] focusing on directed graphs, aiming to present approaches to this problem.

However, most of the deployed multicast routing protocols, like DVRMP[5], CBT[6] and PIM-SM are based upon reverse path routing. Only MOSFP[7] handles asymmetric networks topologies, since the topological database in MOSFP is stored as a directed graph. In PIM-SM, the packet deliver path is set-up as PIM-join messages propagates towards the RP or the source. Due to asymmetric links, the path taken by the join message may not be the shortest path that actual traffic toward the receiver should follow. Thus the resulting shared or source based trees may not be optimal. Tree construction for Core Based Trees (CBT) in asymmetric networks, also suffers from a similar problem.

In [8] a new directed tree construction mechanism is proposed based upon CBT. The join process is similar to the proposed CBT approach A new participant (sender or receiver) joins the group by propagating a join request to the core node. When the join request reaches the core node a join-ack is sent back to the participant along the shortest path from core node to the new participant (likely to be different from the path taken by the join request). As well as CBT, this approach may concentrate traffic in fewer links, thus increasing the network load, than protocols that use source-based tree schemes.

REUNITE[9] implements multicast distribution based on the unicast routing infrastructure. Although the focus of the REUNITE approach is to implement multicast distribution using recursive unicast trees, it potentially implements source shortest path trees. Besides the *Join Message*, REUNITE proposes the use of a *Tree message* that travels from source to destination nodes, thus installing forwarding state. Nevertheless REUNITE may fail to construct Shortest Path Trees in certain situations and may lead to unneeded packet duplications on certain links. In [10] some modifications to the REUNITE propose are presented in order to solve these problems, and a new protocol is proposed: the Hop-by-Hop Multicast Routing Protocol (HBHP). But only Source Based Trees are considered both in REUNITE and HBHP.

3 DTMP Overview

There are two basic approaches to implement multicast tree construction: the first one is to build a shared tree to be used by all participants, and the other is to construct multiple sources based trees, one for each sender. The shared tree is rooted at some pre-defined center and because it is shared by all senders, fewer resources are used. However for large groups it may concentrate too much traffic and certain links may become bottlenecks. With the source based trees approach, each sender builds a separate tree rooted at itself.

In PIM-SM the use of both, shared and source based trees, is proposed. It allows nodes to initially join a shared tree and then commute to source based trees if necessary. The same idea is used in the Directed Trees Multicast Protocol (DTMP), herein presented.

3.1 DTMP Tree Construction

First, a shared tree is proposed in order to give receivers the ability to joining the group without knowing where are the sources located. Explicit join requests must be sent by the receivers towards the Rendezvous Point (RP) router. When RP router receives a join request it must send back to the new receiver an acknowledgment packet. This acknowledgment packet is sent back to the receiver along the shortest path between RP router and the new receiver which may be different from the path followed by the join request.

Routers, along this path, receiving such an acknowledgment packet may then update their routing tables in order to build new multicast tree branches. Updating is done basically by registering with the multicast routing entry for that tree, the acknowledge packet's incoming and outgoing router interfaces.

The join to shared tree mechanism proposed by DTMP, is illustrated at Figure 1.

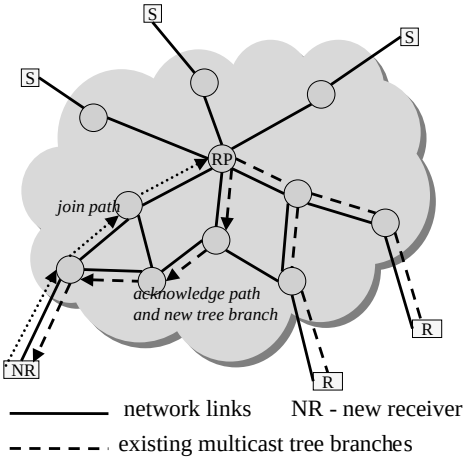
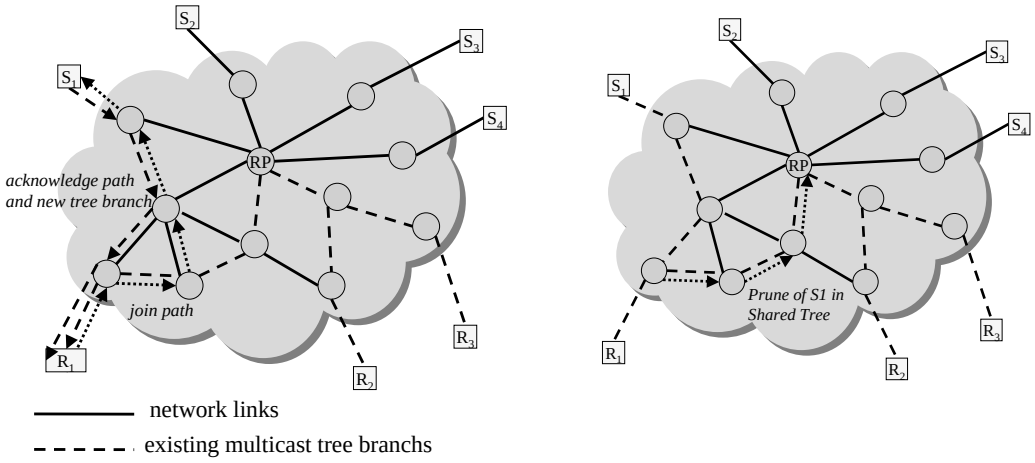


Fig. 1. *Building a DTMP Shared Tree*

After receiving a given set of data packets from a source, a receiver may decide to join a source-based tree. This procedure is similar to the one described above. An explicit join request must be sent from the receiver to the source. When accepting a join, the source must generate an acknowledgment packet, addressed to the corresponding receiver. This acknowledgment packet will signal the necessary routing table updates that will lead to the construction of a new source based tree branch. All the routers along this source tree branch should stop receiving data from that source through the shared tree in order to prevent duplicate of data packets. To accomplish this, a mechanism similar to "prune of source S in shared tree", proposed in PIM-SM specification, must be implemented, as described in section 4.

Figure 2 illustrates the mechanisms used to switch from a shared to a source base tree, as well as pruning that source from the shared tree.



(a) R1 forcing a DTMP Source Tree rooted at S1

(b) Prune of Source S1 in DTMP Shared Tree

Fig. 2. *Building a DTMP Source Tree*

3.2 De-construction of a tree branch

When participants leave a group additional mechanisms must be implemented to tear down state, and eventually cut out tree branches.

As the multicast trees are built from RP, or sources, toward the receivers, the PIM-SM leave tree mechanism should be modified. So, the leave group functionality is implemented by explicit *triggered prune messages* toward the shared tree and source based trees, as detailed in section 4.

Another possible approach would be to implement some kind of tree refresh mechanism instead of an explicit action to tear down state. In this case, when a receiver wants to leave the group it simply stops sending *periodic join request messages*. If tree routers do not receive *join ack messages* within a time-out period, the corresponding entry is deleted. This alternative mechanism is not yet included in DTMP implementation.

4 DTMP Implementation

Network Simulator (NS)[11] has been used to simulate the DTMP proposal and to analyze its characteristics and control overhead.

NS includes a multicast routing protocol able to construct shared trees and source trees with the same structure as the trees constructed by the PIM-SM protocol. However, as the NS's implementation is centralized, a distributed version of PIM-SM has been defined and implemented, in addition to DTMP implementation.

With PIM-SM, tree construction is based on explicit join requests issued by receivers. When a receiver wishes to join a multicast group it should send a *join-request* towards the RP of the respective group. Each upstream router creates or updates its multicast routing table when receiving a *join-request*. The interface where the join-request arrives is added to the list of outgoing interfaces of the corresponding entry. The routers in the shortest path between the new receiver and the RP only forward the *join-request* if they do no yet belong to the shared tree.

In DTMP this behavior has been modified. The *join-request* sent by the new receiver is just forwarded towards the RP by all the routers along the way. When the RP receives the *join request*, it sends back a *join-ack message* towards the new receiver. This *join-ack* will cause the construction of the new tree branch. All the routers in shortest path between the RP and the new receiver will process and forward the *join-ack*, updating their routing tables. The interface added in the corresponding outgoing interface list is the one that has been used to forward the *join-ack message* to the new receiver. Notice that no resource reservation is performed in any on-tree router, so there is no guarantees that there will be any dynamic route adaption besides the adaptation granted by underlying unicast routing protocols.

The process of joining the shared tree in DTMP is detailed in Figure 3, where variables and flags have the same meaning as defined in PIM-SM[2].

The routing table entries have the same fields as the PIM-SM ones, and an extra one: the upstream neighbor in the tree. This field has been introduced in order to be able to implement the prune mechanism.

The process of commuting to a source based tree is similar to the above described one. However, after the construction of the new branch, when a router between the source and the receiver starts to receive data from that source, it must issue a prune of that source on the shared tree. This prune indicates that packets from this source must not be forwarded down this branch of the shared tree, because they are being received through the source based tree. This mechanism is implemented by sending a special prune to the upstream neighbor in the shared tree. When a router at the shared tree receives this type of prunes, it creates a special type of entry (an (S,G)RPT-bit entry) exactly like a PIM-SM router. In DTMP the outgoing interface list of the new (S,G)RPT-bit entry is copied from the (*,G) entry and the interface deleted is the one being used to reach the node that had originated the prune, which may not be the arriving interface of the prune packet[1]. This is because in DTMP there are directed trees not reverse path ones.

[1] In PIM-SM the outgoing interface list of the new (S,G)RPT-bit entry is copied from the (*,G) entry and the arriving interface of the prune packet is deleted.

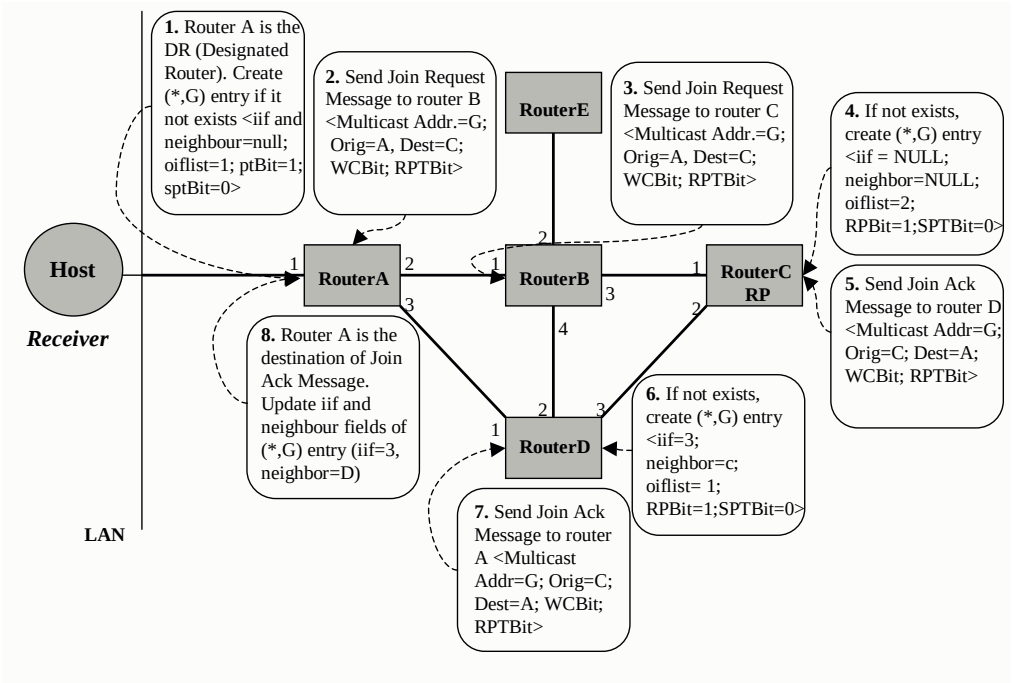


Fig. 3. *Set Up Shared Tree Implementation Actions are numbered in the order they occur*

These (S,G)RPT-bit entries must be updated too when a join-ack arrives in order to allow the join of a new receiver on a shared tree with source-specific prune state established.

The process of switching from the shared tree to a source based tree in DTMP is detailed in Figure 4.

As the actual NS implementation does not include the periodic Join/Prune process proposed in PIM-SM specification to capture member-ship changes, explicit prunes requests to tear down state when receivers wish to leave the group, had to be implemented. For that purpose the implementation of the DTMP uses the additional field that has been added to the routing table entries. As stated before, this field contains the identification of the upstream neighbor in the tree. So, the prunes must be sent toward that router. When the upstream neighbor receives this type of prunes (that are different from the "prune of a source on the shared tree") it must delete the interface used to reach the node that had originated the prune from the outgoing interface list of the corresponding (*,G) or (S,G) entry. Again, this interface may be not the arriving interface of the prune packet. If the outgoing interfaces list become empty the entry may be deleted and the prune should be forward to the upstream neighbor in the tree.

This way, although the construction process of the multicast tree was inverted, from RP or source toward the new receiver, the de-construction pro-

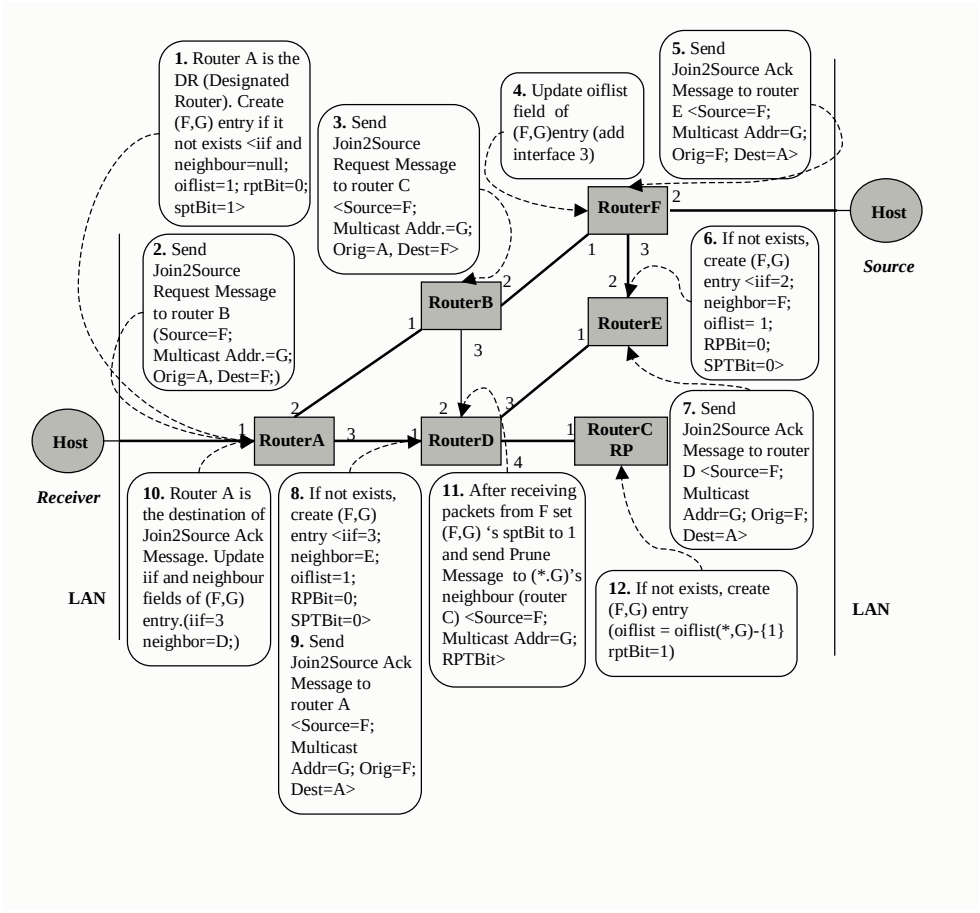


Fig. 4. *Switching from Shared Tree to Source Based Tree Implementation*

cess follows the same way that in PIM-SM: from the bottom to the top of the multicast tree. This fact saves a lot of unnecessary control messages in the deconstruction tree process. We believe that this field may be used to implement the periodic tree refresh, too.

5 Simulation Analysis

NS has been used in order to simulate DTMP and to compare DTMP with PIM-SM.

There are several different ways to measure the quality of multicast trees. One possible way is to count the number of links in the topology belonging to the tree; another way is to count the number of data packet replicas that are originated by nodes while forwarding those packets across the distribution

tree. Since each node should receive no packet duplicates from any source, the number of packet replicas it sends, is in fact the number of different outgoing interfaces which can be used to reach receivers. Therefore, both values are the same (except for transients) if a single tree is involved. Although, when DTMP or PIM-SM are used more than a single tree may be involved, because some receivers usually switch from shared to source based trees. In this later case, the metric accounting for the number of data replicas is the best one to use because it counts the resources effectively in use.

Another difference between these two metrics relates to the way values are to be computed. While the number of packet replicas can only be computed during data transfer, the number of links can be computed by the time when routing entries are created or deleted from multicast routing tables.

Note however that none of those two tree cost measures take the link characteristics into account, and they can't be used to realize how well the tree construction mechanism deals with link asymmetries. Therefore, instead of using only the number of data packet replicas, a metric combining this number with the cost associated to each link traversed by each packet replica is used. This has been the first metric taken into account. A second metric, just the total number of links[2] involved in all the trees has also been used.

5.1 Simulation Scenarios

Typical ISP network The first topology used in a simulation scenario is a typical large ISP network[12] as shown in Figure 5.

This topology includes 18 nodes and 30 links. Associated to each link there are two link utilization costs, one for each direction. Each cost is an integer randomly chosen from different intervals, as specified later. In this scenario simulations consider only one group with two fixed sources, at node 3 and node 9. It is assumed that a single receiver is connected to each node in the topology and that all nodes have one potential receiver attached. For each simulation run, the RP node is randomly chosen within the set of all the nodes. At the beginning there are no receivers joined to the group. After an initial period, receivers start to join the group building a shared tree rooted at RP.

After all the receivers have joined, one receiver, randomly chosen from all the receivers, issues a join to the source attached to the node 3 and later another receiver (randomly chosen as well) issues a join to the source attached to the node 9. This scenario (one shared tree and two source based tree) is then kept till the end of simulation. Before the simulation ends, all the receivers abandon the group.

Several experiments have been made with this topology. In the first one each link cost is an integer randomly chosen from the interval [1, 5], from the interval [1, 10] in the second one and finally the cost is randomly chosen from the interval [1, 20]. For each experiment a set of 100 independent simulations have been used and the results shown are the average from those 100 simulations.

[2] This value is not multiplied by the associated link cost because it is only used in order to assure that DTMP does not build larger trees than PIM-SM.

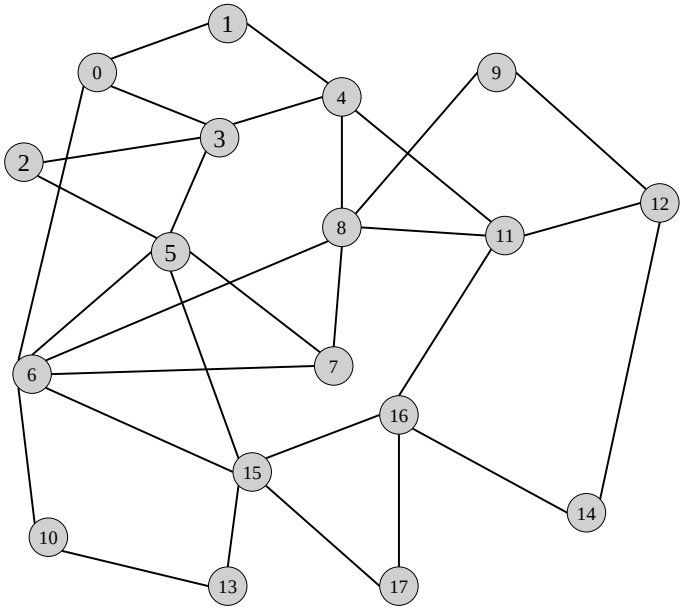


Fig. 5. *Network Topology: Typical large ISP*

Hundred Nodes Network Another experiment scenario, with a second topology randomly generated using GT-ITM[13][3], with 100 nodes and 354 links has been analyzed. With this topology a set of 10 independent simulations have been used. In each, the RP is randomly chosen within the set of all the 100 nodes, and the link costs are also randomly chosen from the interval [1, 10]. Like in the other experiments there is a potential receiver connected to each node (100 receivers) and two fixed sources in node 3 and node 9. All the receivers start by joining the shared tree and, some time after that, two receivers randomly chosen join source 3 and source 9 respectively. The results shown were taken from the average of those 10 simulations.

One final note about tree cost, just to to mention that it includes the aggregate cost of all the trees created during simulation: one shared tree rooted at RP (node 0) and two source trees rooted at nodes 3 and 9. Also note that since all receivers join and leave the group during simulation, there are always two different measures for each number of active receivers. For example, there are 0 receivers when starting and also 0 receivers when finishing. Presented values are averages of all observations, grouped by the number of active receivers.

[3] Using Pure Random edge generation method, 100 nodes, scale 100 and edge probability of 0.033

5.2 Simulation Results and Analysis

Simulations results are presented in Figures 6 and 7. Figures 6(a) to 6(f) show the average cost of the trees constructed by the two protocols (PIM-SM and DTMP) for the first topology. Tree cost reflects the quality of the constructed tree and thus provides a good way for comparison among different tree construction mechanisms, but as stated before there are several ways to measure those costs. The curves presented in Figure 6(a), 6(c) and 6(e) show results when the first metric, number of replicas `times` the link cost, is used. The curves presented in Figure 6(b), 6(d) and 6(f) show results using the second metric: the total number of links in the topology that are involved in the multicast trees.

These results demonstrate that DTMP constructs trees with costs smaller than those created by PIM-SM without enlarging the size of the trees. These results are more evident as link asymmetries became more significant. The average gain of DTMP over PIM-SM is 13,7% when the links costs are randomly chosen from the interval [1, 5], 20,8% when the links costs are randomly chosen from the interval [1, 10], and 27,5% when the links costs are randomly chosen from the interval [1, 20].

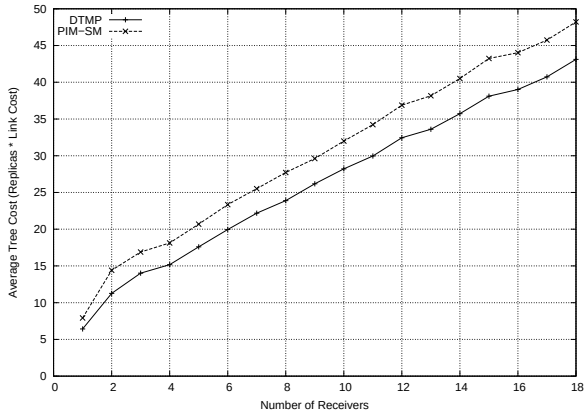
Figures 7(a) and 7(b) show the average cost of the trees constructed by the two protocols (PIM-SM and DTMP) for the second topology (the 100 nodes randomly generated topology).

With this second topology the advantage of DTMP over PIM-SM is also clear and the number of links shows also a similar value both in DTMP and in PIM-SM. This result fact indicates that DTMP builds up trees with similar number of links than those created by PIM-SM but with the advantage of being directed trees. In this experiment the gain of DTMP over PIM-SM is 21%, which led us to conclude that the advantage of DTMP does not depend on the topology, neither on the number of nodes or receivers involved. But of course it is straight related with the link asymmetries.

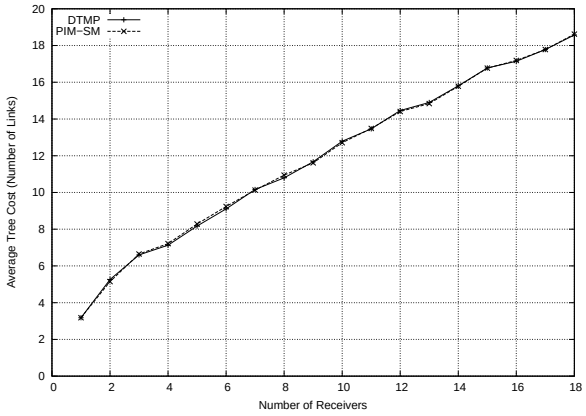
6 Conclusions and Future Work

A new proposal is presented in this paper, DTMP - a multicast routing protocol that implements directed trees construction in opposite of reverse path ones. The original idea is based on PIM-SM protocol, a widely deployed multicast routing protocol in the Internet. The PIM-SM, as the majority of multicast routing protocols, builds reverse path trees. This fact may lead to poor routes in the presence of asymmetric networks and problems may arise when trying to implement QoS Routing, as the links usually have different characteristics in each direction.

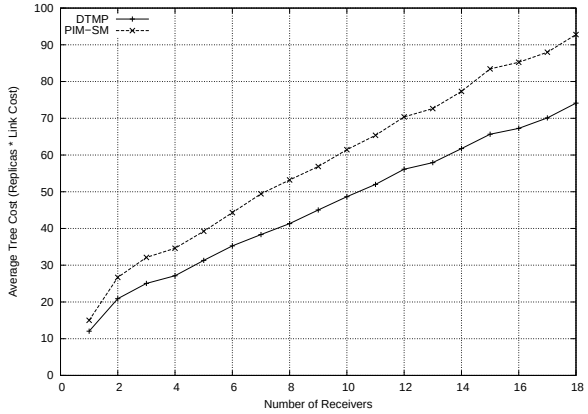
DTMP uses both shared trees and source based trees. Receivers begin joining a shared tree, rooted in a pre-defined point called Rendez-Vous Point. After having received a certain amount of data packets from a source, a receiver may switch to a source based tree. The protocol allows for an easy way of constructing a source based tree, pruning unnecessary tree branches within the shared tree.



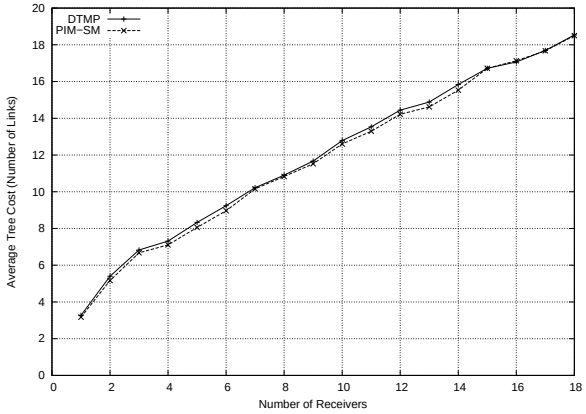
(a) Tree Cost - Link Cost from [1, 5]



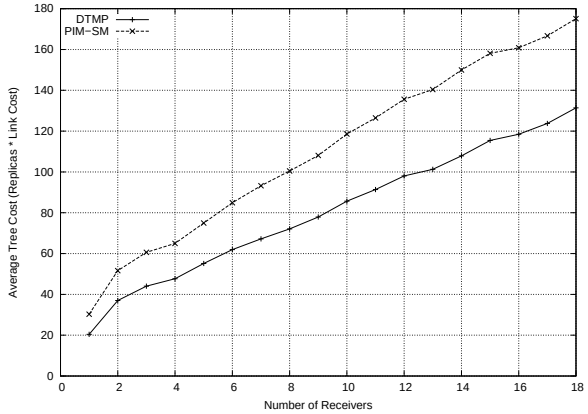
(b) Tree Cost - Link Cost from [1, 5]



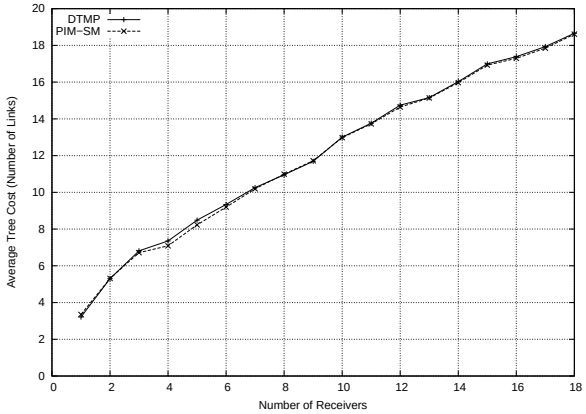
(c) Tree Cost - Link Cost from [1, 10]



(d) Tree Cost - Link Cost from [1, 10]

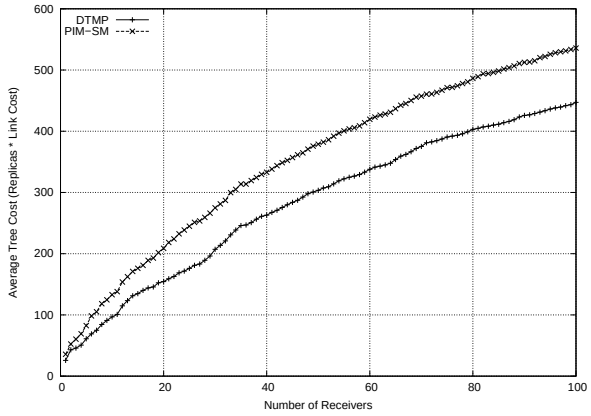


(e) Tree Cost - Link Cost from [1, 20]

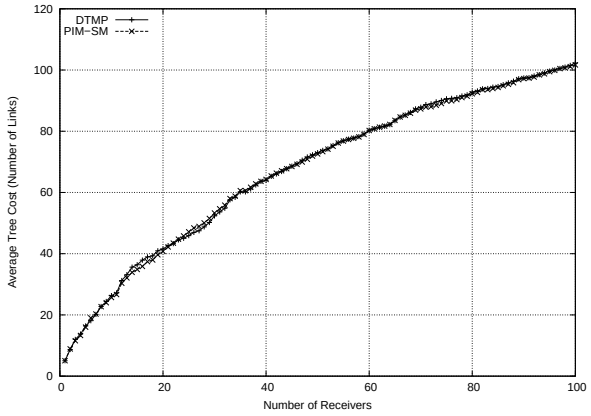


(f) Tree Cost - Link Cost from [1, 20]

Fig. 6. *Simulation Results for the 18 nodes ISP topology*



(a) Tree Cost



(b) Tree Cost

Fig. 7. *Simulation Results for the 100 nodes topology*

DTMP has been implemented and tested with Network Simulator. The simulations results show that in presence of asymmetries within the network the DTMP is a promising approach, enabling the establishment of directed multicast distribution trees that may account for asymmetric routing restrictions. DTMP has been tested both with link-state and distance-vector unicast routing protocols, with similar promising results.

Furthermore, the proposed protocol is also (unicast routing) protocol independent and easy to combine with the installed base of PIM, being able to extend PIM functionalities to directed tree establishment. Also, such as PIM, DTMP is really loop free as long as the underlying unicast routing protocol grants this basic routing characteristic. Again, if the underlying unicast routing protocol is able to provide QoS based routing information, DTMP will construct QoS-aware multicast distribution trees.

Nevertheless, further sets of tests are still needed to take a definitive conclusion about the effect of the control messages overhead introduced. As DTMP join messages must be forwarded up to the tree root and acknowledged back, it is clear that DTMP needs to exchange more control messages per tree change than PIM-SM. However, this situation is not too relevant, because this occurs just during the join group operations. Leave operations and all the other control messages, periodically exchanged, to keep multicast routing entries active, will not cause extra overhead. The refreshing strategy in DTMP is similar to the one used in PIM-SM: each node groups all control messages to each upstream neighbor into a single message. Therefore it is not expected to impact results in a way that may lead to a different conclusion.

7 Acknowledgments

This work has been partially funded by FCT under the Project QoS II, POSI EEI/10168/98.

References

1. P. Winter. Steiner problem in networks: A survey. *Networks*, 17:129–167, 1987.
2. D. Estrin, D. Farinacci, A. Helmy, D. Thaler, S. Deering, M. Handley, V. Jacobson, C. Liu, P. Sharma, and L. Wei. Protocol independent multicast-sparse mode (PIM-SM): protocol specification. Request for Comments 2362, Internet Engineering Task Force, June 1998.
3. Moses Charikar, Chandra Chekuri, To yat Cheung, Zuo Dai, Ashish Goel, Sudipto, and Ming Li. Approximation Algorithms for Directed Steiner Problems. *Journal of Algorithms*, 33(1):73–91, October 1999.
4. S.Ramanathan. Multicast Tree Generation in Networks with Asymmetric Links. *IEEE/ACM Transations on Networking*, 4(4):558–568, 1996.
5. D. Waitzman, C. Partridge, and S. E. Deering. Distance vector multicast routing protocol. Request for Comments 1075, Internet Engineering Task Force, November 1988.
6. A. Ballardie. Core based trees (CBT version 2) multicast routing. Request for Comments 2189, Internet Engineering Task Force, September 1997.
7. J. Moy. MOSPF: analysis and experience. Request for Comments 1585, Internet Engineering Task Force, March 1994.
8. J.Eric Klinker. Multicast Tree Construction in Direct Networks. In *IEEE MILCOM*, Whashington DC,USA, October 1996.
9. Ion Stoica, T. S. Eugene Ng, and Hui Zhang. REUNITE: A recursive unicast approach to multicast. In *INFOCOM (3)*, pages 1644–1653, 2000.
10. Luís Henrique M.K. Costa and Serge Fdida and Otto Carlos M.B. Duarte. Hop-by-hop multicast routing protocol. In *ACM SIGCOMM'2001*, pages 249–259, August 2001.
11. K. Fall and K. Varadhan. *The NS Manual*, Jan 2001.
URL=http://www.isi.edu/nsnam/ns/ns-documentation.html.
12. George Apostolopoulos, Roch Guerin, Sanjay Kamat, and Satish K. Tripathi. Quality of service based routing: A performance perspective. In *SIGCOMM*, pages 17–28, 1998.
13. K. Calvert and E.W. Zegura. *GT-ITM: Georgia Tech internetwork topology models* (software), 1996.
URL=http://www.cc.gatech.edu/fac/Ellen.Zegura/gt-itm/gt-itm.tar.gz.