An Algorithm for Cost-Delay Balanced Trees and Its Application to Wide Area Network Multicast

Pavlin Ivanov Radoslavov[1*] and Takehiro Tokuda[2]

[1] Department of Computer Science
University of Southern California
Los Angeles, CA 90089-0781
Tel: +1-213-740-7287 Fax: +1-213-740-7108
Email: pavlin@catarina.usc.edu
[2] Department of Computer Science
Tokyo Institute of Technology
2-12-1 Ookayama, Meguro-ku, Tokyo 152, Japan
Tel: +81-3-5734-3213 Fax: +81-3-5734-2912
E-mail: tokuda@cs.titech.ac.jp

Abstract. We present a distributed algorithm for creation of a multicast communication tree based on data-flow discovery. Each receiver chooses its preferable Quality of Service (QoS), namely cheaper cost or shorter delay. For shorter delay QoS, the algorithm uses the shortest path from the receiver to the sender. If a receiver prefers cheaper cost QoS, then the algorithm tries to find an already existing data flow within the receiver's area.

Using a parameter we can control the size of the search area so that we can produce a cost/delay optimized communication tree. First we present the flat network version of the algorithm where we use local broadcast to find an already existing data flow. The simulation shows that we can save up to 30% of the tree cost compared to the Shortest Path Tree (SPT). Secondly we present the hierarchical version of our algorithm. We use a Multicast Registration Server tree to search for an existing data flow and to reduce the amount of broadcasted messages.

1 Introduction

The success of MBONE [3], the Multicast Backbone, has proved the usefulness of Wide Area Network (WAN) multicast as a method of communication among number of hosts distributed over the whole Internet. However, the used multicast routing protocol, named Distance Vector Multicast Routing Protocol (DVMRP) [6], does not scale up very well, because the information is broadcasted periodically over the whole network and its amount is proportional to the number of all senders. Currently there are approximately 3000 multicast routers participating in MBONE and the scalability is not yet a serious problem, but

* The first author is grateful to INES Corporation, Japan for supporting his work as a graduate student at Tokyo Institute of Technology.

in the future when all routers in the Internet have multicast functionalities, we must use more scalable methods.

We present a scalable distribution tree creation algorithm. First we describe the basic algorithm. Then we present two methods by applying the algorithm to flat networks and hierarchically organized networks. Finally we discuss the subject of making the algorithm compatible with already existing multicast architectures such as Protocol Independent Multicast (PIM) [2].

2 Cost and delay tradeoff

Cost and delay are two most important characteristics of network communications. In the case of unicast where we have a single source and a single destination, the cheapest solution in terms of used network links is usually the shortest delay solution. This solution is usually referred to as the shortest path between the source and the destination. Unfortunately, in general it is not true for multicast communications where we have a number of sources and destinations. For example, Fig. 1 shows two choices for the multicast distribution tree with one sender and two receivers. The first solution uses the shortest path from the source to each destination, while the second solution minimizes the amount of used network resources.

Some applications like bulk file transfer to many destinations or lecture-mode multimedia conference do not necessarily require shortest propagation delay. For such applications the network could use any cheaper solution. To do so, we need to create a minimum cost tree among all members.

Unfortunately, the problem of finding the minimum cost spanning tree for a subset of all nodes in the network (known as a Steiner tree) is an NP-complete problem [4]. Also, many approximation algorithms for construction of such a tree require knowledge of the whole network topology and this makes the Steiner tree approach inapplicable to large scale networks.

Our solution to the cost-delay problem requires each receiver to specify its QoS, preference of cost or delay. Then the network will choose the appropriate heuristic solution.

3 Previous approaches

3.1 Shortest Path Trees

The Shortest Path Tree (SPT) is the most commonly used single source-multiple receivers solution. The packets follow the shortest path from the source to each receiver and have the shortest propagation delay. With the help of the intermediate nodes (routers), the packets will travel only once over each used link. The method is simple and straightforward, but it does not bring cost optimization.

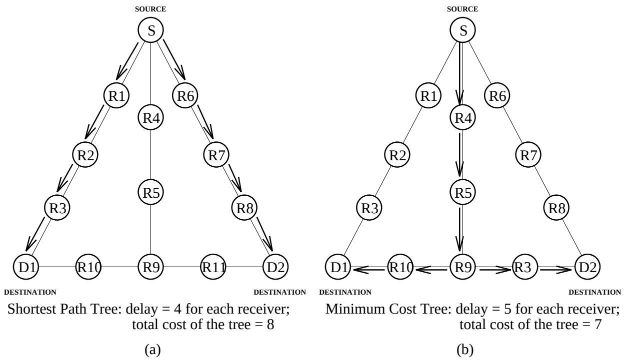


Fig. 1. Minimum delay and minimum cost distribution tree examples

3.2 Minimum Cost Trees

The SPT has a benefit for each destination because it offers the best possible delay. But it is not the best solution for the network itself. The Minimum Cost Tree (MCT) tries to minimize the amount of used network resources. If each link has some assigned cost, then the MCT solution will be the tree with the minimum sum of all used links cost.

Unfortunately, the MCT problem belongs to the NP-complete problems and its creation does not have an easy or scalable solution. This makes it unsuitable for multicast communication trees.

3.3 Core Based Trees

The Core Based Tree (CBT) [1] is neither the shortest delay nor the minimum cost solution. But it has some interesting advantages compared to the MCT and the SPT.

The CBT has one or more core routers as a backbone for a specific multicast group and uses receiver-oriented tree construction. Each receiver joins the core tree by simply sending a request to one of the core routers. The request is processed by all intermediate routers by adding the link with the incoming request to the distribution tree. There is only one distribution tree for the whole group, so the amount of information within each router is minimized. There is no need to know the network topology, the senders and other receivers addresses. What is necessary is to know the address of one of the core routers and to have a unicast routing protocol which will forward the requests of the receivers. Each sender just unicasts the data toward one of the core routers, which resends it to the other core routers and all receivers down the tree.

The CBT has following disadvantages.

- In general, the distribution tree is not the optimum one.
- All senders' traffic is concentrated on the same links.
- Currently, there is no simple algorithm for the core routers selection.

3.4 Protocol Independent Multicast

PIM is the first multicast architecture designed for WAN multicast. The used data distribution tree is either Center Based Tree, centered at the so called Rendezvous Points (RP), or (Reversed) Shortest Path Tree (RSPT). The first type of distribution tree has the advantage of reduced amount of state information in the routers, while the second one has the advantage of shortest possible propagation delay between a sender and a receiver. However, PIM does not try to reduce the cost of the used distribution tree.

4 Cost-delay balanced tree algorithm

Minimizing the cost of a tree connecting a subset of nodes in the graph is known to be an NP-complete problem. We propose a new heuristic algorithm for creation of the data distribution tree. Our distributed algorithm does not require knowledge of the network topology. The resulting tree can be optimized either in terms of cost or propagation delay. To achieve cost reduction, the algorithm tries to find a nearest existing multicast data flow within an area with predefined boundary. To optimize the propagation delay, we use the shortest path from the receiver toward the source. Initially, the algorithm uses local search for an existing multicast data flow within the receiver's area. If the search succeeds, the new branch of the tree will be constructed from the discovered node instead of the source. If the search fails, then the algorithm continues along the shortest path toward the source. By increasing the size of the searching area, the resulting tree may have smaller cost and the propagation delay may be longer.

The algorithm itself is not a resource reservation algorithm and the assumption is that each of the existing links in the network has sufficient bandwidth to be included in the distribution tree. However, during the search process it is possible to include an additional check for existence of sufficient network resources.

The algorithm is initiated by each receiver for particular source within a multicast group. We can represent the algorithm by the following pseudo code:

```
subscribe(source, mcast_group, local_search_method)
{
  do{
    do_local_search(source, mcast_group, local_search_method);
    if(data_flow_found)
      create_tree_branch_from_there;
    else{
      move_to_the_next_node_toward_the_source;
    }
  }while(data_flow_not_found)
}
```

Each receiver initiates the algorithm independently from the other receivers and there is no need for synchronization among any receivers.

The *local_search_method* is of main importance for the practical application of the algorithm. We present two versions of the algorithm which differ in the used *local_search_method*. The first version is simpler and can be used in flat, non-hierarchically organized networks of moderate size. The second version of the algorithm is more suitable for very large networks with sparse membership. But it requires the existence of hierarchically organized Multicast Registration Servers (MRSs).

5 Data-flow discovery based on restricted broadcast

The method is simple and easy to implement, because it uses only restricted broadcast for the local search, which is initiated by the intermediate nodes. We use Time To Live (TTL), representing a number of hops, to control the size of the search area. The TTL parameter is passed by the receiver in the initial subscription request toward the first router.

5.1 An example

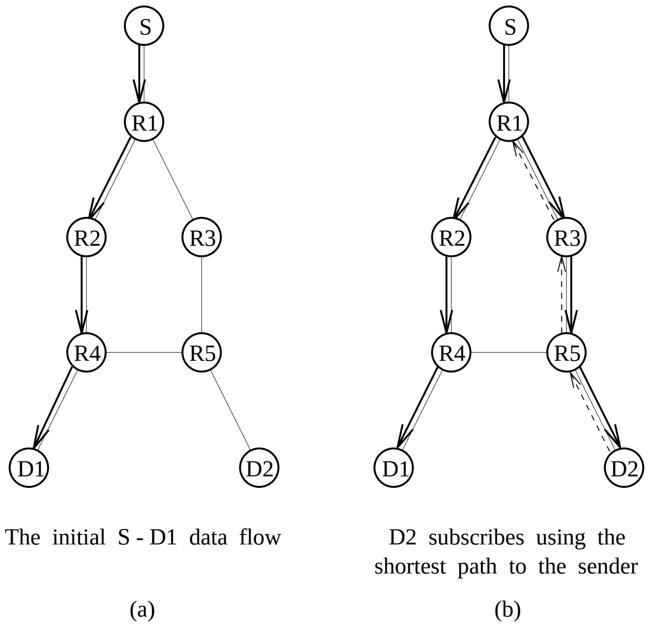
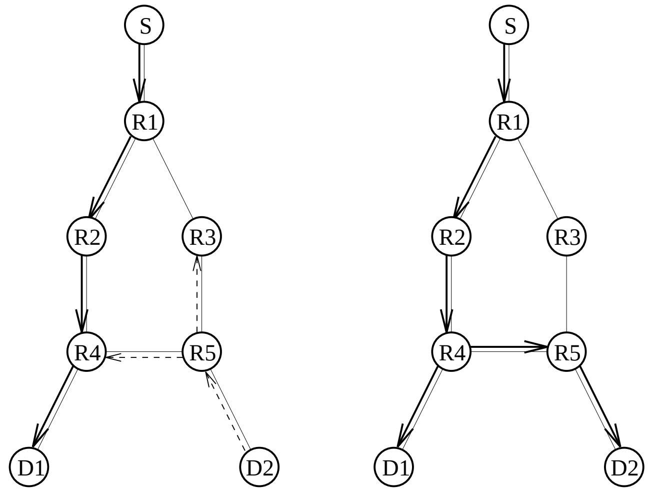


Fig. 2. An example of using shortest path tree for subscription

We illustrate our method by a simple example. In the network of Fig. 2(a) the first receiver (destination) D1 is already receiving the multicast data from the



R5 uses limited broadcast
to find an exisiting data flow

(a)

The new branch of the
tree (R4 - D2) is created

(b)

Fig. 3. An example of using limited broadcast to find an existing data flow

sender S. The established data path follows the shortest path between S and D1 (S - R1 - R2 - R4 - D1). If the new receiver D2 wants to subscribe using the shortest path toward the sender, its subscription request strictly follows the shortest path to S (D2 - R5 - R3 - R1 - S). When the request reaches a node which already belongs to an existing data flow from the same source to the same multicast group, this node (R1) starts forwarding the multicast data to the new receiver D2 and the new branch of the tree is created (Fig. 2b). If we assume that all links have *cost*=1 and *delay*=1, then the cost of the whole tree will be 7 and both D1 and D2 will be receiving the multicast data with delay 4.

If D2 prefers a cheaper solution, then the resulting tree will be different. In Fig. 3a the new receiver D2 sends the request for subscription to the next node toward the sender S (R5.) If R5 uses limited broadcast with distance of one hop, the broadcast reaches R4 and R3. Node R4 already belongs to the path of the searching data flow, so it returns a positive answer to R5. Node R5 collects all positive answers and chooses the closest node. After the selection is done, R5 sends a request toward the selected node R4 to create the new branch of the multicast distribution tree (Fig. 3b). The cost of the created tree is 6, which is less than the cost of the SPT of Fig. 2b (7). The propagation delay to the first receiver D1 is still 4, but for the new receiver D2 the delay will be longer (5) than the SPT solution (4).

5.2 TTL values

Different TTL values bring different cost or propagation delay. Bigger TTL improves the chances of discovering the data flow in some of the neighbors which otherwise do not belong to the traversed shortest path. The discovery of such a neighbor will help us to create a branch of the tree that is cheaper than the shortest path to the sender. Obviously, the propagation delay will be longer than the shortest path solution.

The TTL parameter specifies the number of hops a broadcast packet can propagate from the current node: When TTL=1, we check only the current node for existing data flows. When TTL=2, we broadcast using one hop distance. When TTL=3, we broadcast using two-hop distance. Obviously, TTL=1 creates the SPT. Although we did not evaluate source-centered trees where the source simply unicasts the data to each receiver, in this method we can use TTL=0 for such purpose. Initial subscription with TTL=0 indicates that we use a unicast request toward the sender and expect all data from the sender to be unicasted back.

5.3 Simulation results

To evaluate the effectiveness of our algorithm, we implemented a simulator to test a number of randomly generated networks. We used the random graph model described in [7].

For each network we created the MCT (we used the Minimum Spanning Tree approximation algorithm [5]) and the SPT. We used them as bases to evaluate the cost and the delay of our tree. For each network we used limited broadcast with different TTL. The other conditions of the simulations were as follows.

– One sender and one multicast group for each network.
– Network size from 60 to 1000 nodes.
– Number of receivers from one receiver to 30% of the network size (90% for networks with number of nodes smaller than 100).
– All receivers use the same TTL from 1 to 5.
– The number of simulations for each combination (network size, receivers, and TTL) is 500.

For each network we calculated the cost of the tree, the average and maximum propagation delay. For tree cost evaluation we used the MCT cost as a base (1.0). The SPT delay was the base for delay evaluation. We need to see how the TTL parameter will affect the cost and the delay of our tree. We found that the results of different network size were similar. We present the data for the smallest (60 nodes) and the largest (1000 nodes) networks.

The results are respectively shown in Fig. 4 (cost statistics for 60-node networks), Fig. 5 (average delay statistics for 60-node networks), Fig. 6 (maximum delay statistics for 60-node networks), Fig. 7 (cost statistics for 1000-node networks), Fig. 8 (average delay statistics for 1000-node networks) and Fig. 9 (maximum delay statistics for 1000-node networks).

The simulation results clearly show how the use of different TTL values can improve the cost and delay of the resulting tree. Increasing the TTL decreases the total cost of the resulting tree, but increases its average and maximum delay. However, only small values for TTL (in our experiments of up to 5) have relatively significant improvement of tree cost saving. Additional increase of TTL almost does not change the resulting tree cost. It is important to know that the use of large TTL values should be discouraged.

Our simulation shows that on the average we can save up to 30% of the tree cost compared to the RSPT and at the same time we are as low as 7% above the cost of the Minimum Cost (Steiner) Tree (MCT). It is worth noting that at the same time the propagation delay (average and maximum) can be 40% shorter compared to the MCT (see Figure 8 and Figure 9).

On the other side, decreasing the TTL value decreases the propagation delay, but increases the cost of the result tree. If TTL=1, the propagation delay can be as short as the RSPT delay, because the result tree is exactly RSPT. Any values for TTL between 1 and the threshold $TTLmax$ (in our simulations $TTLmax$=5) can be used as a finer granularity cost or delay optimization.

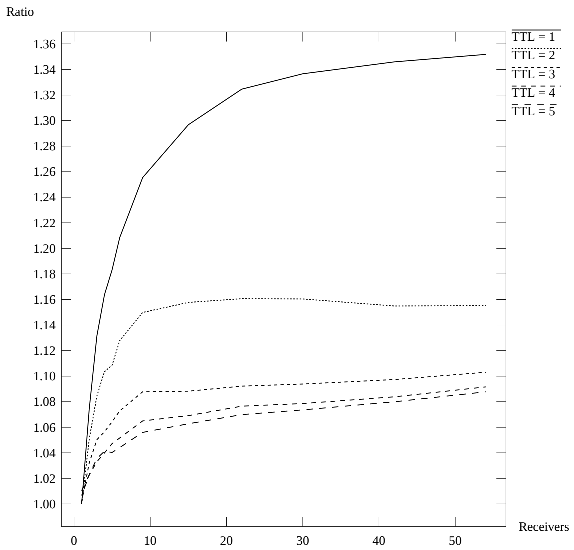


Fig. 4. Cost statistics for 60-node networks

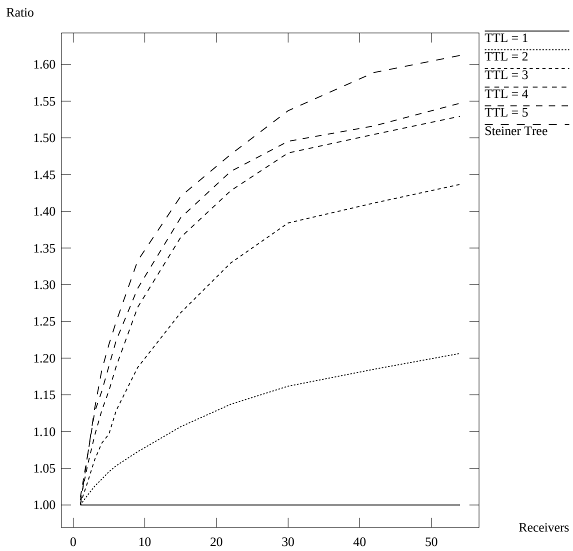


Fig. 5. Average delay statistics for 60-node networks

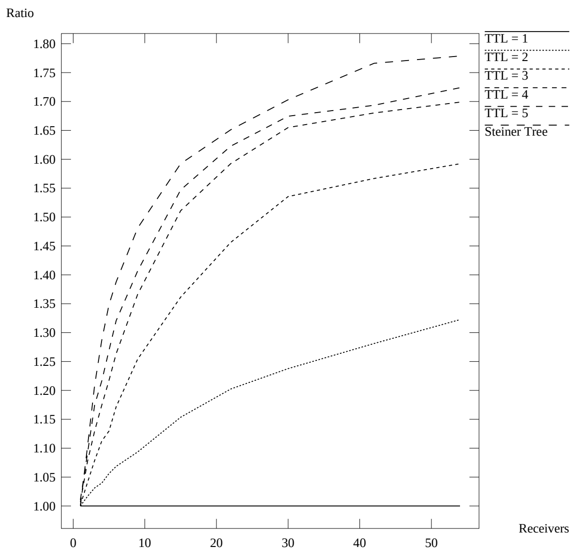


Fig. 6. Maximum delay statistics for 60-node networks

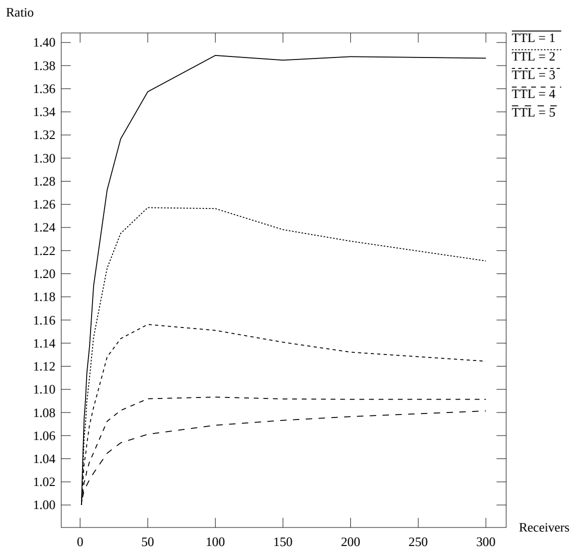


Fig. 7. Cost statistics for 1000-node networks

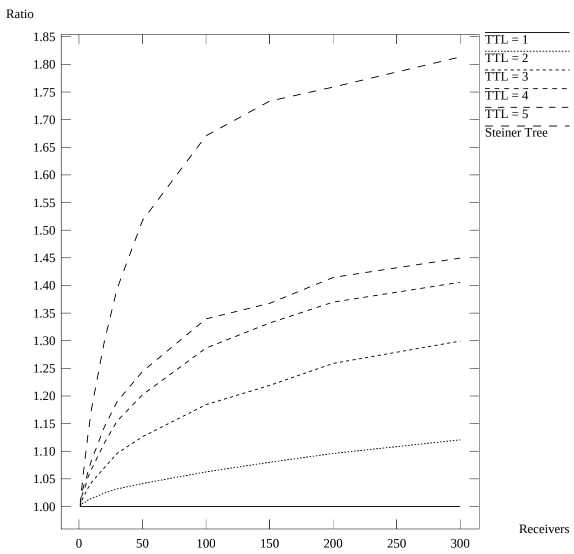


Fig. 8. Average delay statistics for 1000-node networks

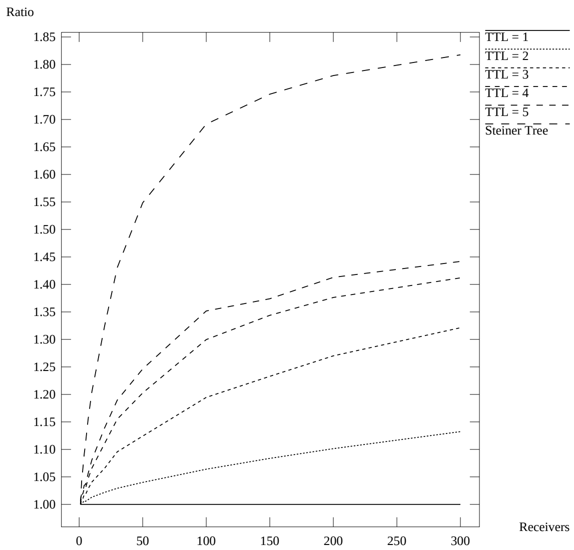


Fig. 9. Maximum delay statistics for 1000-node networks

6 A cost-delay balanced tree in a network with hierarchically organized servers

To reduce the amount of messages broadcasted by the intermediate nodes during the search process, we modify the algorithm to use a different search method. This version of the algorithm uses hierarchically organized Multicast Registration Servers (MRSs) for local search instead of restricted broadcast. Each MRS represents an area in the network. Higher level MRSs in the hierarchy are responsible of bigger areas. By using unicast requests to some MRSs instead of broadcast, we can significantly reduce the number of messages required to find an already existing data flow in the network. To do so, each MRS must know an existing multicast data flow within its area. The data flow registration is accomplished during the initial establishment of the data flow path. Fig. 10 shows a simple example of using MRS for data flow discovery within a single domain.

We use the hierarchy of MRSs to search for an existing multicast data flow in the network. This can be done by applying a bottom-up search to the MRS tree. Here we use a *vertical TTL* (vTTL) parameter, which is similar to the broadcast TTL parameter used in the flat network version. This parameter specifies the highest possible level of MRSs to ask about an existing multicast data flow. Larger vTTL values allow a query to reach MRSs of higher positions, which are responsible of bigger areas of the network. Below we will illustrate the process by an example.

In the initial network in Figure 11 we have a 2-level organized MRS hierarchy

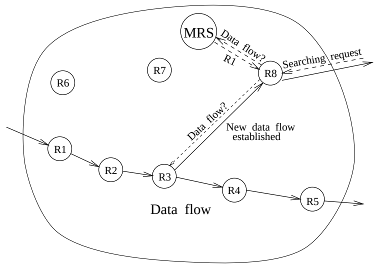


Fig. 10. MRS support of data flow discovery

and only the black servers have information about a particular multicast session. This information includes source addresses, but because of the restriction of space we will omit the description of its management. In the network shown in Figure 12 we have already established a dataflow from the sender S to the first receiver R1.

In Figure 13 the new receiver R2 does a local search within its area with vTTL=2. The search fails, but the top MRS returns to R2 the address of the sender S. Using this address, the search request is sent hop-by-hop toward the sender until it reaches the next domain. There, the first node N1 does a local search again (see Figure 14). The dataflow is discovered and the address of one of the nodes belonging to the dataflow (N2) is returned back. In Figure 15 the request is sent toward N2 instead the source S, but even before reaching N2 the dataflow is discovered at R1. Thus, the new branch of the tree is created (see Figure 16). Figure 17 shows the resulting tree if we had used the SPT. The advantage of our tree is that we avoid the usage of one of the links between two domains, while the propagation delay to the new receiver R2 is slightly longer in comparison to the shortest path solution.

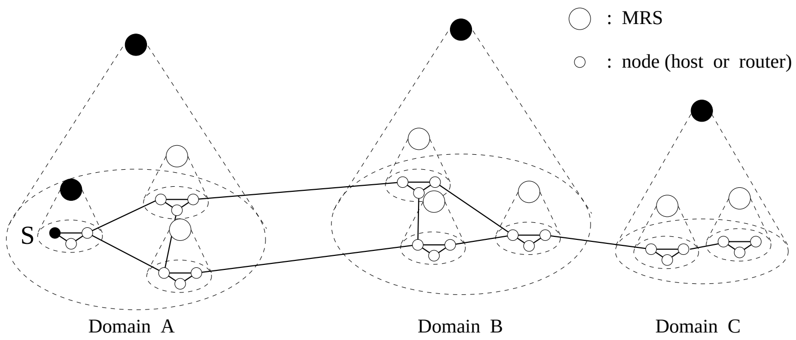


Fig. 11. The initial network. The black nodes only know about the multicast session

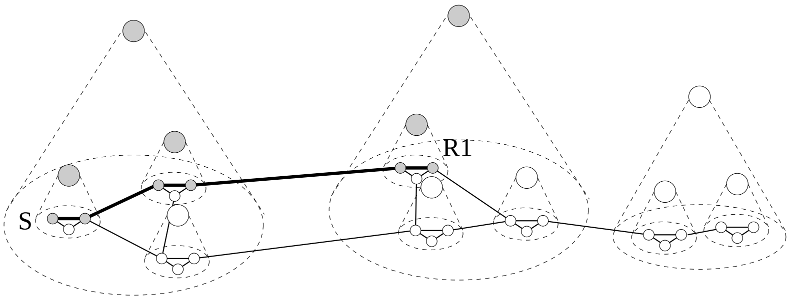


Fig. 12. The first dataflow. The grey nodes only know about the existing dataflow

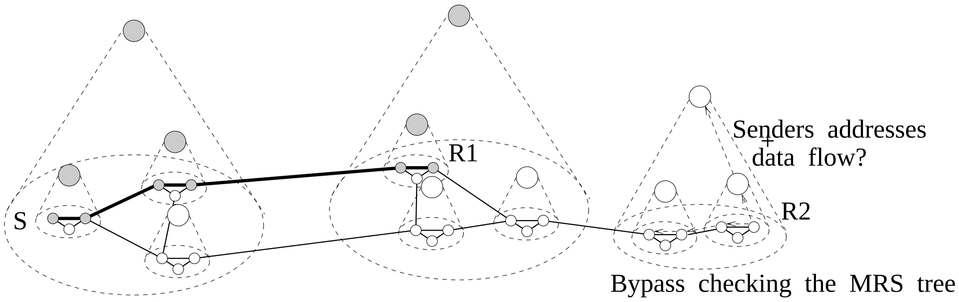


Fig. 13. The initial search by the new receiver R2

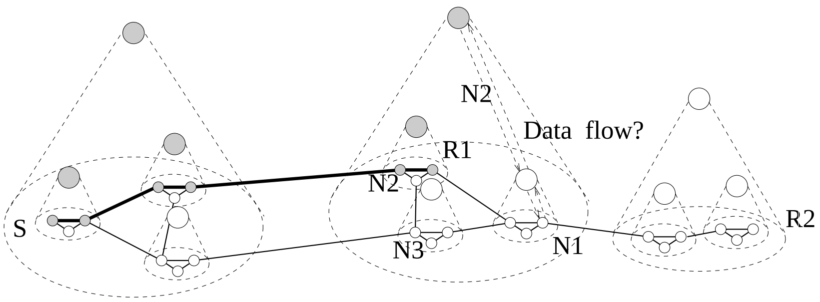


Fig. 14. Search initiated by the first node (N1) inside the new domain

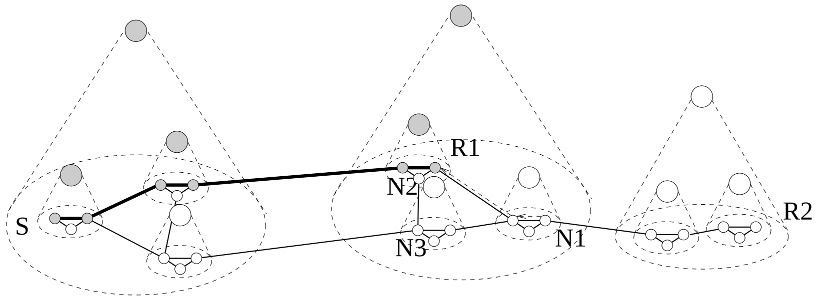


Fig. 15. Search request sent toward the discovered dataflow

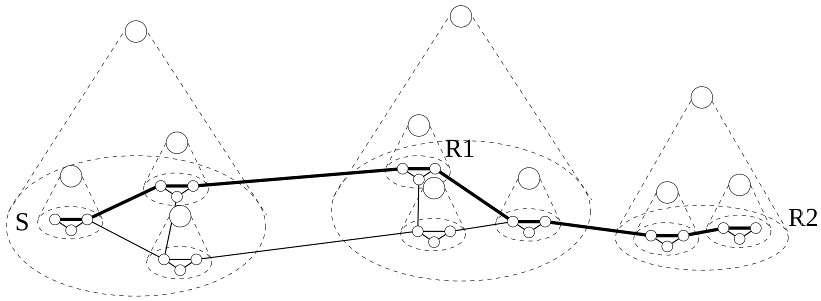


Fig. 16. The resulting multicast distribution tree

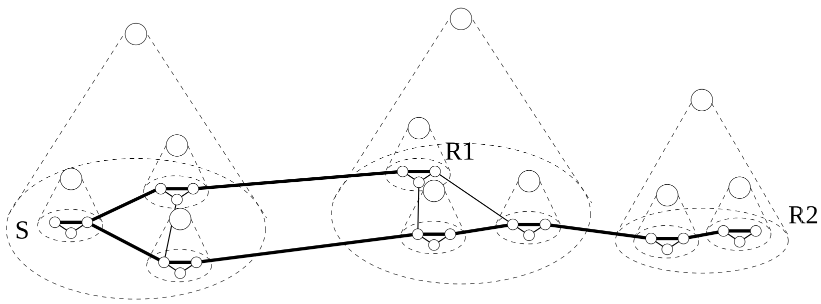


Fig. 17. The SPT tree solution

7 PIM compatibility

Although our WAN multicast method is designed as an independent multicast architecture, it can be modified as an extension to PIM. For example, when the last hop router chooses to switch from the RP centered tree to a source-specific tree, it can use our heuristic algorithm to subscribe to a specific sender's data distribution tree instead of using the RSPT. Currently, the RP hierarchical organization is not defined yet and our MRS tree can be a solution of extending the RPs functionalities.

8 Conclusions

We have presented a simple and scalable algorithm for cost-delay balanced tree construction. The algorithm does not require knowledge of the network topology. It is receiver initiated algorithm and the resulting tree is created on demand. We have also presented a modified version of the algorithm which uses hierarchically organized MRSs for data-flow discovery. Our current version of the MRS tree uses statically configured MRSs. A dynamic creation of the MRS tree will be explored our future work.

An extreme case of our tree construction is a sender centered tree (TTL=0), which has the advantage of no state information in the intermediate routers, but this is still an open issue.

References

1. Ballardie, T., Francis, P. and Crowcroft, J.: Core Based Trees, in *Proceedings of the ACM SIGCOMM*, San Francisco, 1993
2. Deering, S., Estrin, D., Farinacci, D., Jacobson, V., Liu, Ching-Gung, and Wei, L.: An Architecture for Wide-Area Multicast Routing, *Proceedings of SIGCOMM'94*, October 1994, pp 126–135.
3. Macedonia, M.R. and Brutzman, D.P.: MBone Provides Audio and Video Across the Internet, *IEEE Computer,* April 1994, pp. 30–36.
4. Miller, R.E. and Thacher, J.W., Reducibility among combinatorial problems, in *Complexity of Computer Communications*, R. M. Karp, Ed. New York: Plenum, 1972, pp. 85–103.
5. Takahashi, H. and Matsuyama, A., An approximate solution for the Steiner problem in graphs, *Mathematica Japonica,* Vol. 24, No. 6, 1980, pp. 573–577.
6. Waitzman, D., Partridge, C., and Deering, S., Distance Vector Multicast Routing Protocol, RFC 1075, November 1988
7. Waxman, B.M.: Routing of Multipoint Connections, *IEEE Journal on Selected Areas in Communications*, Vol.6, No.9, December 1988, pp. 1617–1622.