Re-routing Multicast Connections: A Distributed Approach

Roman Novak, Jože Rugelj and Gorazd Kandus
Department of Digital Communications and Networks, Jozef Stefan Institute, Ljubljana, Slovenia

In virtual-circuit environments that provide a multicast capability, fixed routing algorithms provide solutions that are only efficient initially and for a short time due to dynamic nature of network environments. We propose a re-routing algorithm that continuously rearranges the topology of the initial connection, in response to changes in the destination set and network status. The re-routing algorithm is cost effective and minimally disruptive to the multicast session. It is based on a modified Steiner tree improvement technique, and it has been designed purposely to meet the requirements for distributed implementation in communications networks where only partial information is available to routing nodes.

Keywords: multicast communication, re-routing, distributed algorithm

1 Introduction

Emerging applications of computer networks, such as teleconferencing, remote collaboration and distance learning, make use of multicast communication. New applications involve more than two users. Multicasting is a mechanism that allows an efficient distribution of information among group of collaborating nodes. It is likely to be an essential part of future networks.

Techniques for transmitting messages to multiple users in a multicast communication session have evolved rapidly over the past few years. Broadcasting is a technique for transmitting a packet from a source node to every other node in a network. It makes sure that every node in a network receives one and only one copy of a packet. Multicasting, i.e. selective broadcasting, is the transmission of a packet from a source node to a limited set of destination nodes, the multicast group. It provides additional bandwidth savings over broadcasting, for applications where the multicast group consists of only a subset of the nodes in the network.

Existing multicasting mechanisms make use of tree-shaped topologies of connections because of the efficient use of network resources involved in a multicast session [Dio97]. A minimum number of data packets can be transmitted in parallel to the various destinations along the branches of the tree, with duplication carried out only where the tree branches.

When the utilisation of network resources is the main consideration, the problem of finding the topology of connections reduces to the well-known Steiner tree problem in graphs. The optimal solution of the minimum Steiner tree problem is NP-complete [Kar72] and thus not suitable for real-time applications. Therefore, multicast routing algorithms are based on heuristic algorithms [Dio97, Gel97, Sal97, Dee96].

1.1 Multicast re-routing

The design of the re-routing algorithm for connection-oriented multicast services is based on several considerations. Because multicast connection should be established as quickly as possible, rather inefficient Steiner tree heuristics are used. The cost of allocated network resources may not be even close to minimum.

Multimedia flows are relatively long-lived and routing decisions are made only in the connection set-up phase. Therefore, every routing decision made in the initial phase may influence

---

Work supported by Ministry of Science and Technology, Republic of Slovenia.

the state of the network for a long time [PZ96]. As the state of the network changes over time, the inefficiency of the initial tree tends to grow.

The further reason that speaks for re-routing is the dynamic nature of multicast communication. Unicast communication exists until one of the two communicating partners leaves the communication. That is not the case with the multicast communication. Each of the communicating partners is free to join and leave while others proceed with communication. The problem of updating the multicast tree after each addition and deletion is known as the on-line multicast problem in networks [BV97]. Some authors propose occasional tree re-computations to maintain optimal topology. This approach involves all the group nodes as well as other network nodes. The problems with smooth transition from the original path to the alternative path can be quite difficult, because every change of communication path affects ongoing traffic.

The need for re-routing a static connection topology has already been considered over the past. Much work has been done on re-routing unicast connections [PZ96] while re-routing multicast connections is still a relatively new subject [BV97].

1.2 Distributed versus centralised re-routing

The success of a routing algorithm is predicated on having an accurate view of the state of the network, including its topology and the availability of resources at every node. With centralised routing, the stability and granularity of this view become important issues.

When a network is small, it is possible to maintain a consistent view of the network status by broadcasting the information needed. As the network gets larger, broadcasting becomes less reliable and more expensive. When the nodes do not have complete knowledge of the topology and state of the network, distributed routing and re-routing algorithms are needed [Zha98, RK97, BV96]. Re-computation of the route would presumably take place on those nodes having the resources, and the resulting decisions would become effective immediately at those nodes. The distributed algorithm collects the information needed and re-routes the multicast connection in a decentralised manner, with each node carrying out a specified re-routing algorithm.

Attention has been given to the distribution of multicast routing algorithms, while a distributed re-routing algorithm has not yet, to our knowledge, been considered.

1.3 Organisation of the paper

In this paper we present a distributed algorithm for the dynamic re-routing a multicast delivery tree, which can further improve cost effectiveness of the initial multicast connection and can dynamically adapt the connection topology to changes in the network. Section 2 describes the relation between multicast connection topology and the well-known combinatorial optimisation problem. Section 3 introduces the Steiner tree improvement technique that forms the basis of the re-routing algorithm. Two modifications, given in Section 4, are essential for successful implementation of the technique in the distributed environment. A distributed algorithm that implements the dynamic re-routing is introduced in Section 5, while Section 6 presents the results based on multiple simulation runs. We conclude the paper by summarising our conclusions in Section 7.

2 Multicast connection topology

Problems arising in the domain of routing in communication networks may often be decomposed into a number of well-known combinatorial optimisation problems. When the aim is the overall minimisation of network resources used, routing multicast connection can be viewed as solving a minimum Steiner tree problem in graphs.

Given a weighted graph, the minimum Steiner tree problem is to find a minimum cost subgraph spanning a set of specified nodes. Formal definition of the minimum Steiner tree problem can be written as follows:

Definition [minimum Steiner tree problem]: Let $G = (V, E)$ be an non-directed connected graph of the communication network with the nodes set V, the connections set E, and non-negative weights associated with the connections. In this graph we have a set $Q \subseteq V$ of member nodes, called the multicast group. The minimum Steiner tree problem is to find a minimum cost subgraph of G, such that a path exists between every pair of member nodes.

The optimal solution of the minimum Steiner tree problem is NP-complete [Kar72] and thus not suitable for real-time applications. Due to its complexity, one is interested in developing efficient heuristic algorithms to find good approximate solutions. Extensive studies of Steiner tree problem heuristics have been made [HR92, Vos92]. Only a few of them are suitable for application to the multicast routing problem when distributed implementation is required [RK97, BV96, Kom96].

The shortcomings of the static multicast routing algorithms can be alleviated by re-routing multicast connections. The foundation of the proposed re-routing algorithm is the Steiner tree improvement technique.

3 Steiner tree improvement technique

Once we have a tree covering a set of member nodes, we can try to further reduce its cost. Improvement technique is a procedure applied to the heuristic solution of the minimum Steiner tree problem in order to improve the solution.

Only a very small number of improvement techniques have been developed within the framework of the minimum Steiner tree problem [Vos92]. We made some modifications to a simple technique in which the tree is viewed as a set of simple paths. The end nodes of the simple path are group members or nodes with degree greater than 2 with respect to the tree. The intermediate nodes of the simple path are not group members and are of degree 2 with respect to the tree.

Definition [simple path]: Let T denote the solution tree to the given problem generated by some heuristic and $P_T(i,j)$ a path between nodes i and j using tree edges, $P_T(i,j) \subseteq T$. A path $P_T(i,j)$ is said to be simple if:

i. all nodes of $P_T(i,j)$ except i and j are of degree 2 with respect to T and are not in Q,

ii. nodes i and j are in Q or they are of degree greater than 2 with respect to T.

Nodes i and j are called end nodes of the simple path $P_T(i,j)$.

Each step of the improvement technique reduces the cost of the tree by replacing a simple path with cheaper alternative path.

Iterated steps of the improvement technique

1. Select at random a simple path $P_T(i,j) \subseteq T$. Remove $P_T(i,j)$ from T with i and j remaining in T. Let T_i and T_j be the two disconnected components of T.
2. Find the least cost path $P(i^*,j^*)$ between the components T_i and T_j. Denote the end nodes of the least cost path i^* and j^*, such that $i^* \in T_i$, $j^* \in T_j$.
3. Add the nodes and edges of the least cost path between i^* and j^* to T, $T = T_i \cup P(i^*,j^*) \cup T_j$. Hereafter, we call the resulting path an alternative path to a given simple path.

4 Re-routing algorithm

The original improvement technique is very general, because it assumes no restriction on the location of the alternative path. Switching between the original simple path and the alternative path means a disruption of the multicast service offered to the clients. Some data packets in transmission may be lost; others may be duplicated. There is also a possibility that packets may arrive at the destination out of order. Problems that are even more serious emerge if we consider quality-of-service guarantees given to clients. We would like to dynamically re-route the multicast delivery tree without disrupting the service to clients; therefore, we impose some restrictions on the improvement technique.

In order to cope with the above mentioned problems efficiently and to simplify the distribution of the re-routing algorithm to network nodes, we allow only those alternative paths that have an end node in common with the simple path. In the above step, the requirement can be written as an equality $j = j^*$.

Serious problems emerge if we try to establish several alternative paths in parallel. Without proper synchronisation, parallel execution of the improvement steps may result in improper topologies. We significantly reduce the problem and enlarge the level of parallelism by restricting the search range for the node i^*. Therefore, we define the set of alternative nodes $A_{i,d}$:

Definition [set of alternative nodes]: Let $P_T(i, j)$ be a simple path, T_i one of the components after removing the path $P_T(i, j)$ from the tree T and d a search diameter. For each node $k \in T_i$ there exists a sequence of simple paths $P_T(i, 1)P_T(1, 2)P_T(2, 3)...P_T(n-1, n) \subseteq T_i$ such that the node k is a member of the last simple path only. The length of the sequence for the node k is $L(k) = n$. A set of alternative nodes $A_{i,d}$ is a union of nodes from component T_i such that their sequences of simple paths have lengths less than or equal to the search diameter, $L(k) \leq d$.

It is possible that the alternative path that satisfies the conditions $j = j^*$ and $i^* \in A_{i,d}$ intersects the remaining two components in additional nodes, i.e. nodes other than i^* and j^*. Because re-routing should preserve the tree topology of the re-routed connection, such alternative paths are forbidden. The re-routing algorithm can be written as follows.

Iterated steps of the re-routing algorithm

1. Select at random a simple path $P_T(i, j) \subseteq T$. Let T_i and T_j be the two components such that $T_i \cup P_T(i, j) \cup T_j = T$, $T_i \cap P_T(i, j) = \{i\}$ and $P_T(i, j) \cap T_j = \{j\}$.
2. Find the least cost path $P(i^*, j)$ between the set of nodes $A_{i,d}$ and the node j. The path $P(i^*, j)$ is an alternative path to the path $P_T(i, j)$ only if the conditions $T_i \cap P(i^*, j) = \{i^*\}$ and $P(i^*, j) \cap T_j = \{j\}$ are satisfied.
3. If the path $P(i^*, j)$ is the alternative path then remove the path $P_T(i, j)$ from the tree T and add the alternative path $P(i^*, j)$ to the tree T, $T = T_i \cup P(i^*, j) \cup T_j$.

5 Distribution of the re-routing algorithm

The execution of the re-routing algorithm requires an accurate view of the network topology and the availability of resources at every node. When the nodes do not have complete knowledge of the network topology and state, the distributed re-routing algorithm is needed.

We assume a point-to-point network environment, where each node has only limited knowledge about the network. The knowledge in the node consists only of information about the least cost paths to all other nodes and the next nodes on these paths. A multicast delivery tree is initially established by one of the multicast routing algorithms and is later optimised by means of re-routing. The tree nodes only need to participate in the execution of the algorithm.

The distributed re-routing algorithm consists of three steps. The three steps are advertising the end node of the simple path, interrogating the set of candidate nodes, and transition to the alternative path.

5.1 Advertising the end node of the simple path

During the first step each tree node becomes aware of the first end node that is reached using a particular outgoing connection. The outgoing connection is the network connection that is part of the multicast delivery tree and connects the node with one of its immediate neighbours. A tree node also memorises the cost of the tree path that leads to the particular end node.

This could be accomplished by requiring end nodes to send periodically the end node reports on each simple path to the other end node. Initially, the end node report contains the end node identification and the cost of the outgoing link. Intermediate nodes, i.e. non-member nodes of degree 2 with respect to the tree, update the end node report by summing the reported cost and the cost of the next outgoing link and then forward the report. On receiving the end node report, the other end node associates the reported end node and the reported cost with the outgoing connection from which the report has been received.

5.2 Interrogating the set of candidate nodes

Each end node gathers the information on possible alternative paths during the second step, i.e. during interrogation of the set of candidate nodes. The second step can be initiated in every end node in the tree. It is, as is the end node reporting, triggered periodically.

According to our notation from the previous section, the end node is the node i. During the second step its task is to determine node i^* for each neighbouring end node j. We assume that the end node has information about the least cost paths from itself to all other nodes. Obviously, this is not enough to make a decision on i^* for any neighbouring end node j. We introduce here a mechanism that makes this decision possible.

In our implementation, the end node i receives the requested information for each neighbouring end node j by distributing enquiry messages to the nodes from the set $A_{i,d}$. The distribution is similar to multicast data forwarding with the exception that border nodes of the set $A_{i,d}$ terminate forwarding. The enquiry message contains a proposition as to the least cost path to the node j. The proposition includes the cost of the potential path $P(i^*, j)$ and the identification of the node i^*. The border nodes of the set $A_{i,d}$ return enquiry messages back to the node i. After receiving all the enquiry messages, the end node i can select a least cost path $P(i^*, j)$.

Each enquiry message builds a trace of traversed end nodes. The trace is required by the third step of the re-routing algorithm.

5.3 Transition to the alternative path

The end node checks the existence of the alternative path before it triggers the next interrogation. It selects at random such a neighbouring end node j, for which it has information about an alternative path $P(i^*, j)$ that promises tree cost improvement.

The distributed algorithm allows many re-routing processes to proceed in parallel. However, co-ordination is needed while re-routing processes take place close to each other. The transition to the alternative path is based on a locking mechanism. The objective of locking is to prevent parallel transitions from corrupting the multicast tree topology and, at the same time, to verify consistency of the information about the alternative path. Loops and disconnected components might appear in the topology if locking is not implemented. The locked node should delay or refuse all requests that result in a tree topology change until it is unlocked.

We can divide the transition into several sequentially executed substeps. The first substep is locking the original path $P_T(i, j)$. The original path is locked by sending a message to the node j. The second substep represents locking the alternative path $P(i^*, j)$. Finally, in the third substep the algorithm locks the existing path $P_T(i^*, i)$. The message that performs the locking is passed to the node i using the trace that has been obtained during the interrogation step.

Locking can be cancelled at any time during forwarding of the lock message. This may happen for several reasons. The original path may be changed in the meantime. For instance, one of the intermediate nodes could become the end node because a new member node entered the multicast communication. The alternative path may intersect the tree in the node other than i^* and j; this case is forbidden by the re-routing algorithm. The trace of the path $P_T(i^*, i)$ may not be valid because the tree topology has been changed in the meantime. Finally, some node on the paths $P_T(i, j)$, $P(i^*, j)$ and $P_T(i^*, i)$ may already be locked by a parallel transition in progress. In the case of cancellation, the message is sent back along the locked path. At each node, the state before locking is restored.

If the locking succeeds, the node j sends the message with a success flag set back along the locked path. The message unlocks nodes on the path $P_T(i^*, i)$, activates nodes on the alternative path and removes the original path from the tree.

6 Simulation results

In order to evaluate the distributed re-routing algorithm presented in the last section, we have developed a tool for simulation-based distributed algorithm evaluation and carried out simulations on random graphs. In addition to performance evaluation our tool allows statistical verification of some correctness criteria. We formalise correctness criteria as claims about the behaviour of a distributed algorithm. Three fairly standard correctness criteria are the absence of deadlocks, livelocks, and improper terminations. Other assertions can be checked during simulation runs including system invariants, i.e. a more general case of boolean correctness criteria that, if true in the initial state, remain true in all attainable states. This section summarises the simulation results.

The re-routing algorithm assumes the existence of an initial solution derived by some heuristic algorithm. In our simulations, we combined the least cost path heuristic (LCP) with the re-routing algorithm. The initial tree, built by the least cost path heuristic, is the union of the least cost paths between the source node and each of the member nodes. Some of the Internet multicast routing protocols [Dee96, Bal93] are based on the least cost path heuristic.

6.1 Evaluation Environment

To ensure that simulations of the algorithm were independent of the idiosyncrasies of any specific network, we constructed random graphs. We used the network model introduced by Waxman [Wax88] and improved by Doar and Leslie [DL93].

Graphs were constructed by distributing n nodes across a Cartesian co-ordinate grid. The location of each node had integer co-ordinates and multiple nodes were not permitted to exist at any location. The connections were added to the graph by considering all possible pairs (u, v) of nodes and using the probability function $P(u, v) = k\overline{e} \exp^{-d(u,v)/(\alpha L)} / |V|$ where $\overline{e}$ is the mean degree of the node, $|V|$ is the cardinality of V in G, $d(u, v)$ is the Euclidean distance between the node locations, L is the maximum possible distance between two nodes, α is a parameter in the range $0 < \alpha \leq 1$, and k is a scale factor related to the mean distance between two random nodes. A large value of α increases the number of connections to nodes further away, while a large value of k increases the number of connections from each node. The cost of the connection was defined as $d(u, v)$. We tested each graph to ensure that only one connected component existed.

It is difficult to develop a reasonable model of a network state dynamic that describes the evolution of its topology and costs of the links. Due to that and because there is no other known algorithm to compare with, we chose to evaluate the cost improvements of the initial tree and the time needed in a static environment where the convergence of the topology actually takes place. For these reasons, the number of group member nodes, the topology of the network, and the cost function were fixed during each simulation run in order to achieve stable topology of the re-routed multicast connection.

6.2 Results

We compared the cost of the stable trees with the cost of trees obtained by the near-optimal centralised routing algorithm and with the cost of the initial trees. We selected the cheapest insertion algorithm CI [Vos92] for the near-optimal centralised routing algorithm.

The measure we used to evaluate the quality of the trees was the cost effectiveness Ω. The cost effectiveness is defined as the average of the expression $(T - T_O)/T_O$ where T is the cost of the evaluated tree and T_O is the cost of the near-optimal tree.

We were interested in several other aspects of the algorithm such as the impact of the search diameter and the network density on the algorithm's cost effectiveness, the average number of parallel transitions, and the total number of transitions. The last two measures relate to the convergence time of the algorithm.

First, we compared the cost of the tree obtained by the re-routing with the cost of the initial tree and with the cost of the near-optimal tree. In Figure 1 we show the cost effectiveness of the re-routing algorithm for three different values of the search diameter. We designated the re-routing algorithm as RR. The cost effectiveness of the algorithm that builds the initial tree is shown in the same Figure. On average, the re-routing algorithm with a search diameter 3 and above at least halves the difference between the cost of the initial tree and the cost of the near-optimal tree. The tree cost is reduced significantly, especially when group members are distributed sparsely across a wide area. For group sizes up to 15 nodes and reasonable average degree of the network nodes, $\overline{e} = 5$, the cost of the stable tree obtained by the re-routing algorithm comes within 10 percent of the optimal solution. The cost effectiveness of the initial tree and the cost effectiveness of the stable tree do not depend on the size of the network.

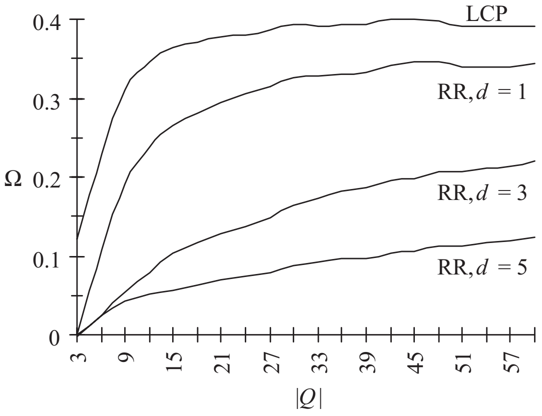

Figure 1: Cost effectiveness of the re-routing ($|V| = 60$, $\overline{e} = 5$, $\alpha = 0.25$, $k = 3.5$).

In Figure 2 can be seen the effect of the limited horizon of the algorithm with lower search diameter. It is interesting that the improvement grows rapidly for small values of the search diameter. The fact that the re-routing algorithm has a limited horizon when it chooses an alternative path has an impact on the probability of entering local minima.

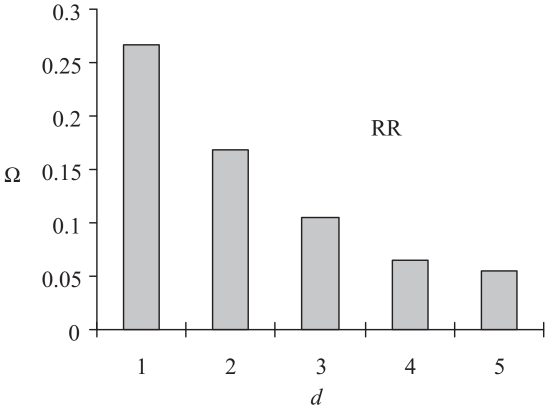

Figure 2: Impact of the search diameter on cost effectiveness ($|V| = 60$, $|Q| = 15$, $\overline{e} = 5$, $\alpha = 0.25$, $k = 3.5$).

The network density has an influence on the effectiveness of the re-routing algorithm. Figure 3 shows what improvement can be expected when we use the re-routing algorithm in networks with different density. Along the x-axis we have the parameter α, along the y-axis the parameter k and along the z-axis the cost effectiveness. With increasing network density, the cost of the stable tree converges to the cost of the near-optimal tree. The cost of the initial tree with increasing network density grows and is very soon above 100 percent of the near-optimal cost. The relation between the network density and the cost of the initial tree is shown in Figure 4.

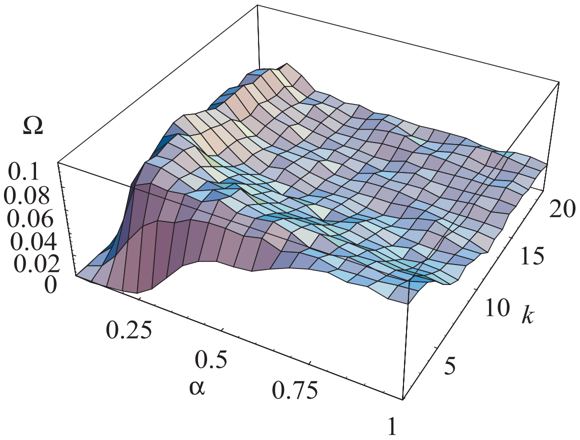

Figure 3: Influence of the network density on the cost effectiveness of the stable tree ($|V| = 60$, $|Q| = 15$, $d = 3$, $\overline{e} = 5$).

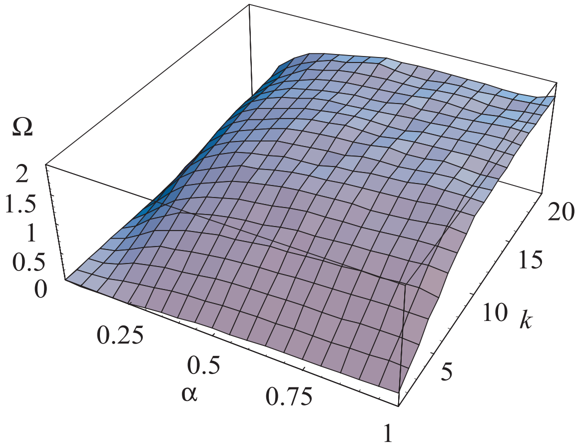

Figure 4: Influence of the network density on the cost effectiveness of the initial tree ($|V| = 60$, $|Q| = 15$, $d = 3$, $\overline{e} = 5$).

As expected, parallel transitions are possible (Figure 5). The average number of transitions running in parallel, ω, increases while the search diameter is relatively low.

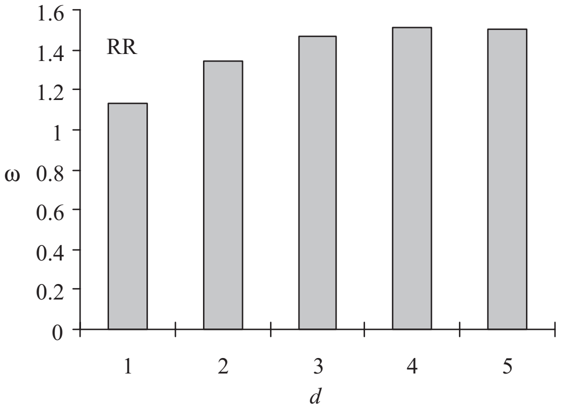

Figure 5: Average number of parallel transitions ($|V| = 60$, $|Q| = 15$, $\overline{e} = 5$, $\alpha = 0.25$, $k = 3.5$).

The total number of transitions from the original simple path to the alternative path grows with increasing size of the multicast group and with increasing search diameter. Approximately 10 transitions are required for larger groups and search diameters in order to obtain stable trees.

7 Conclusion

Static routing mechanisms in communication networks providing multicast capability are of limited efficiency as their solutions are optimal only for a limited period. Multicast connections are long-lived and routing decisions are made only at connection set-up. We have presented a cost effective distributed algorithm for dynamic re-routing of a multicast tree. The conclusion from this is that future multicast routing algorithms will have to have the ability to improve the originally chosen routes, rather than constructing monolithic solutions to a problem specification that may only be optimal for a short time.

The distributed re-routing algorithm is based on the Steiner tree improvement technique. The modifications made to the technique were necessary for the effective distribution of operation of the technique in a network environment. The iterated step of the re-routing algorithm reduces the cost of the tree by replacing a simple path with cheaper alternative path. The results of the least cost path routing algorithm were used as the point of departure for our re-routing algorithm. The reduction of the multicast tree cost depends on the size of the group, on the search diameter, and on the density of the network. The distributed algorithm can scale to large sized graphs and still provide low cost trees. We have shown that it is possible to dynamically reorganise the initial tree in a way that is cost effective and minimally disruptive to the multicast session.

Areas for future research include provisions for robustness of the algorithm in an environment where node and link failures occur. Challenging tasks are the adaptation of the algorithm to multicast connections with guaranteed quality-of-service and investigation of mechanisms for smooth transition to the alternative paths.

References

[Bal93] T. BALLARDIE ET AL. *Core-Based Trees (CBT): An Architecture for Scalable Inter-Domain Multicast Routing.* Computer Communications Review, 23-4 (1993), 85–95.

[BV97] F. BAUER, A. VARMA *ARIES: A Rearrangeable Inexpensive Edge-Based On-Line Steiner Algorithm.* IEEE Journal on Selected Areas in Communications, 15-3 (1997), 382–397.

[BV96] F. BAUER, A. VARMA *Distributed Algorithms for Multicast Path Setup in Data Networks.* IEEE/ACM Transactions on Networking, 4-2 (1996), 181–191.

[Dee96] S. DEERING ET AL. *The PIM Architecture for Wide-Area Multicast Routing.* IEEE/ACM Transactions on Networking, 4-2 (1996), 153–162.

[Dio97] C. DIOT ET AL. *Multipoint Communication: A Survey of Protocols, Functions, and Mechanisms.* IEEE Journal on Selected Areas in Communications, 15-3 (1997), 277–290.

[DL93] M. DOAR, I. LESLIE *How Bad is Naive Multicast Routing.* Presented at the Proceedings of IEEE INFOCOM'93, (1993), 82–89.

[Gel97] E. GELENBE ET AL. *Improved Neural Heuristics for Multicast Routing.* IEEE Journal on Selected Areas in Communications, 15-2 (1997), 147–150.

[HR92] F. K. HWANG, D. S. RICHARDS *Steiner Tree Problems.* Networks, 22 (1992), 55–89.

[Kar72] R. M. KARP *Reducibility among Combinatorial Problems.* In Complexity of Computer Computations (R.E. Miller and J.W. Thatcher eds.), (1972) pp. 85–104. Plenum Press, New York.

[Kom96] V. P. KOMPELLA ET AL. *Optimal Multicast Routing with Quality of Service Constraints.* Journal of Network and Systems Management, 4-2 (1996), 107–131.

[PZ96] C. PARRIS, H. ZHANG *Dynamic Rerouting of Guaranteed Quality-of-Service Connections.* Journal of Network and Systems Management, 4-2 (1996), 181–220.

[RK97] J. RUGELJ, S. KLAVŽAR *Distributed Multicast Routing in Point-to-point Networks.* Computers and Operations Research, 24-6 (1997), 521–527.

[Sal97] H. F. SALAMA ET AL. *Evaluation of Multicast Routing Algorithms for Real-Time Communication on High-Speed Networks.* IEEE Journal on Selected Areas in Communications, 15-3 (1997), 332–345.

[Vos92] S. VOSS *Steiner's Problem in Graphs: Heuristic Methods.* Discrete Applied Mathematics, 40 (1992), 45–72.

[Wax88] B. M. WAXMAN *Routing of Multipoint Connections.* IEEE Journal on Selected Areas in Communications, 6-9 (1988), 1617–1622.

[Zha98] Y. ZHANG ET AL. *An Approach for Cluster-Based Multicast Routing in Large-Scale Networks.* IEICE Transactions on Communications, E81-B-5 (1998), 1029–1040.