NEAR OPTIMAL SHARED-TREE BASED MULTICASTING IN MESH NETWORKS

Talin Moradian

A Thesis
in
The Department
of
Computer Science and Software Engineering

Presented in Partial Fulfillment of the Requirements
For the Degree of Master of Computer Science
Concordia University
Montreal, Quebec, Canada
July 2008

Your file Votre référence
ISBN: 978-0-494-45494-7
Our file Notre référence
ISBN: 978-0-494-45494-7

Abstract

NEAR OPTIMAL SHARED-TREE BASED MULTICASTING IN MESH NETWORKS

Talin Moradian

Multicasting is an efficient information dissemination for a subscribed user group on networks. In this thesis the problem of multicasting in mesh-connected networks is studied. Having a group of processors distributed on mesh, the goal is to present a routing strategy such that every member of the group can effectively transmit data to the other members. The conventional strategies in solving the problem are source-based and shared-tree approaches, each having drawbacks in efficiency or scalability for large networks. To compromise between these two, we use the multi-shared trees approach. We apply a core-selection algorithm based on taxicab geometry to create a small number of distribution trees that are almost optimum with respect to multicast time and traffic.

Acknowledgments

I would like to express my deep and sincere gratitude to my supervisor, Dr. Hovhannes Harutyunyan. His understanding, encouragement and personal guidance has been instrumental in the writing of this thesis.

I warmly thank Shahin Kamali, for his valuable advice and friendly help.

I owe my loving thanks to my fiance Hrair Labajian who helped and encouraged me during my studies.

I owe everything to my very caring parents. I want to thank my lovely mom, my dad and my brother Sarmen who helped and supported me all my life. Without their endless love and support I wouldn't have been here.

Contents

List of Figures **vii**

List of Tables **x**

1 Introduction **1**

2 Literature Review **6**

2.1 Preliminary 6

2.1.1 Mesh Networks 6

2.1.2 Group Multicast 9

2.1.3 Multicast Evaluation Criteria 13

2.2 Review of Previous Studies 14

3 Core Selection Algorithm for Multi-Shared Trees **20**

3.1 Taxicab Geometry 21

3.1.1 General Concepts of Taxicab Geometry 22

3.2 Finding the Optimum Centers of the Multicast Group 25

4 Creating multi-shared trees 36

4.1 The Number of Optimum Centers 37

4.2 Building Shared-Trees 39

4.2.1 Multicast Delay Performance 41

4.2.2 Traffic Performance 45

5 Scalability 47

6 Conclusion and Future Work 52

Appendices 54

A Examples of optimum centers for uniformly distributed multicast nodes 54

B Examples of optimum centers for normally distributed multicast nodes 57

Bibliography 60

List of Figures

1 Types of network topologies 7

2 Examples of mesh and torus networks 8

3 Example of unicast-based multicast 11

4 Example of tree-based multicast approach 12

5 Calculation of optimum multicast time. It will take u, 5 time units to disseminate a message to all the children nodes.) 15

6 Two shortest paths between points A and B in taxicab geometry. The length of paths is 18. 21

7 P, in which $D_{tx}(P, A) + D_{tx}(P, B) + D_{tx}(P, C)$ is as small as possible, should be found. 23

8 Circle in Euclidean and taxicab geometries 24

9 Triangle non-equality in taxicab metric. 25

10 Perpendicular bisectors in taxicab geometry. 26

11 P_α and P_β will be the ultimate nodes for the nodes which are on the diagonal of their perpendicular bisector. 27

12 When C does not lie on the diagonal of the bisector of the furthest pair, it can not be an optimum center. 28

13 When C lies on the non diagonal part of the bisector of the furthest pair, it can not be an optimum center. 29

14 Applying point P_δ on the perpendicular bisector of P_α, P_β (the furthest pair). The potential optimum area is highlighted. 30

15 Finding optimum centers for four multicast nodes. 32

16 Algorithm 1 output: The bold area is the optimum centers area. 33

17 P_δ which is an ultimate node of P_α is an ultimate node of A as well. . . 34

18 If A has two ultimate nodes (P_α, P_β), it can not be a corner of the bounding box 35

19 Number of closest optimum centers for senders located in different regions. 38

20 Example of a shared-tree. 40

21 Example of having two shared-trees for two senders P_θ and P_φ. The tree for P_φ is the tree with the solid line and the tree for P_θ is dotted. 41

22 An example in which the shared-tree is not a shortest path tree for the sender P_θ. A shortest path between P_θ and P_δ is dashed. 42

23 Example of having one furthest node for the sender P_φ for calculating the multicast time. 43

24 Example of having two furthest nodes to the sender P_φ for calculating multicast time. 44

25 O and P_φ have the same ultimate node (P_α). 45

26 Example of having four furthest nodes. In this case the multicast time will be at most $D_{ult}(P_\varphi) + 2$ for a sender P_φ and $D_{ult}(O) + 3$ if O is the sender. 46

27 When there are two pairs of furthest nodes, the diagonals of their perpendicular bisectors will intersect at one point. 48

28 A large number of nodes distributed uniformly (20000 nodes). 50

29 A large number of nodes distributed normally (20000 nodes). 51

30 An example of 300 nodes distributed uniformly. The number of optimum centers in this case is 9. 55

31 An example of 2000 nodes distributed uniformly. The number of optimum centers in this case is 9. 56

32 An example of 300 nodes distributed normally. The number of optimum centers in this case is 167. 58

33 An example of 1000 nodes distributed normally. The number of optimum centers in this case is 137. 59

List of Tables

1 The number of optimum centers when the nodes are distributed uniformly or normally. 49

Chapter 1

Introduction

With the rapid expansion of Internet applications, networks have become more important especially in communication, media, e-commerce and business. Nodes that communicate to each other through direct or indirect paths (connected through a sequence of other nodes) will constitute a network. There are numerous advances in networking technology and among them *multicast* has continued to be an important research topic which is appealing both in industry and academy not only because of its involvement in numerous applications, but also of its challenging complexity [26].

Messages in a network can be passed in different ways. They can be passed from one to one (unicast), one to all (broadcast) or one to many (multicast). In fact, unicast and broadcast are specific cases of multicast. Multicast is a key component for the design of group communication applications which require efficient data delivery to multiple destinations. In other words, multicast is a collective communication in which the same message is delivered from a source to an arbitrary number of destinations [20].

Group multicast [17] refers to a kind of multicast in which every member of the group may transmit data to the whole group. The group is called *multicast group* and its members are called *multicast nodes*. Many multicast applications are group multicast in nature: video-conferencing, synchronization of distributed database systems, distributed simulations, etc. [9].

While in multicast a source node needs to send a message to k ($k \geq 1$) destinations, it is important to determine which path(s) should be used to deliver a message from the source to all its destinations. To implement an efficient multicast routing, *Efficiency* and *Scalability* issues should be tackled. Efficiency concerns a reasonable multicast delay or time to disseminate a message from one member of the multicast group to the other members. Scalability means the capability of the multicast routing to be scaled to a large network. Although scalability is desirable for multicasting in a large network, it should not decrease the efficiency [16].

Another parameter in evaluating the performance of multicast communications is *traffic* [20]. Traffic is the number of communication links used to deliver the message from the source to all destinations. Efficient multicast routing algorithms which minimize time and traffic, will be especially practical and useful in the field of parallel and distributed computing.

Mesh-connected network is a popular network topology which is being used extensively, for example in constructing multicomputers. A multicomputer consists of thousands of processors, which are connected to each other. In addition, two dimensional (2-D) mesh is a planar graph and suitable for VLSI (Very Large Scale Integration) [8] implementation. Multicomputers or Massively parallel computers (MPCs) with 2-D mesh topologies are used in information dissemination-based applications such as in large-scale scientific computations, large databases, and media on-demand servers. The reason is performance, reliability and availability of these mesh networks.

Multicast can be implemented through multiple unicasts. A more efficient approach is tree-based multicast. In this thesis, the focus is on a tree based multicast approach for group multicasting in 2-D mesh networks.

Multicast applications can be classified as either one-to-many or many-to-many services. The one-to-many refers to single-sender applications such as streaming media from a well-known server. For multi-sender applications such as video conferencing and multi-party network games, the many-to-many model has been used where all members are potential data sources. Group multicast is a type of many-to-many model which is concerned in this research. For many-to-many model the following trees conventionally have been used: i.) *single shared-trees*; and ii.) *source-based trees* [38]. A source-based tree protocol builds separate trees for each pair of source and group, that is, each source has its own tree that reaches the active group members, such as DVMRP (Distance Vector Multicast Routing Protocol) [40]. On the other hand, shared based protocols such as PIM-SM (Protocol Independent Multicast-Sparse Mode) [7] and CBT (Core Based Trees) [1] build distributed trees having a central point (or core) to whom all receivers are attached to [21].

To integrate both advantages of source-based and shared-tree routing and overcome their shortcomings, the *multiple shared-tree* (MST) approach is used [16]. By using multiple shared trees the fault tolerance and performance of multicasting will improve [45]. MST uses more than one multicast tree to serve multiple sources and destinations. Minimizing the number of multiple shared multicast trees in arbitrary graphs is an NP-complete problem [35].

In this thesis, the goal is to have efficient group multicasting in terms of efficiency and scalability. Having m nodes in the multicast group, $1 \leq k \leq m$ trees are created such that each source employs one of these trees as its multicast tree to disseminate a message. Among all the multi-shared trees, a sender tends to choose the one that prescribes the minimum multicast time. The goal is to create trees to minimize the multicast transmission delay, however other multicast performance metrics such as traffic should be considered as well. In this way when the data is transmitted by a member, one tree will be chosen and the data will be disseminated through specific tree.

Based on the geometric arrangement of the multicast group members on the mesh, the positions of a set of cores as *optimum centers* to be the roots of the multi-shared trees are calculated. The result will be a small number of distribution trees with almost optimum multicast time and acceptable traffic performance. In other words, the main goal in this dissertation is to propose an algorithm that can find the cores from m nodes in the multicast group. These cores will have the smallest maximum distance to all other nodes in the multicast group. The trees generated by the algorithm have almost optimum multicast delay and can scale very well in a large network. It will be shown that in a mesh network with many nodes distributed either uniformly or normally, the number of optimum cores will tend to one, therefore the number of multicast trees built on these cores tends to one.

It can be concluded that the main goal in this thesis is to have a novel and efficient multicast routing algorithm on mesh networks, using optimal multi-shared tree routing. The reason why multi-shared trees are created is that they enhance multicast efficiency, scalability and reliability compared to source-based and shared-tree schemes [16]. Since core selection is important in creating multi-shared trees, an algorithm which will give optimum center(s) or cores is proposed. Later the multi-shared tree(s) will be built on these center(s) to minimize the multicast transmission delay.

The remaining parts of this dissertation are organized as follows: Chapter 2 gives a literature review of the important network concepts and also the previous work conducted on the subject of this thesis. Chapter 3 is the theoretical work for selecting optimum multicast centers and its related proofs. In chapter 4, it is discussed how to build multiple shared trees. Chapter 5 concerns the scalability of the multicast algorithm proposed in this research in a large mesh network. The final chapter is a conclusion and contains possible directions for future research.

Chapter 2

Literature Review

In this chapter, we review basic concepts and previous results on multicasting in arbitrary networks as well as mesh connected graphs.

2.1 Preliminary

This section is devoted to the study of primary concepts related to multicasting. In Section 2.1.1, Mesh networks are introduced as a suitable topology for interconnection networks. Basic concepts related to group multicasting is reviewed in Section 2.1.2. In section 2.1.3, it is presented how to evaluate a multicast scheme.

2.1.1 Mesh Networks

A network is a collection of nodes, some of which are connected by links. What determines network topology is the configuration of connections between nodes.

There are many topologies that interconnect the nodes in a network. Examples of these topologies are Ring, Star, Tree, Bus, Mesh and Torus [24]. In Figure 1, Ring, Star, Tree and Bus are illustrated; while in Figure 2 Mesh and Torus topologies are shown.

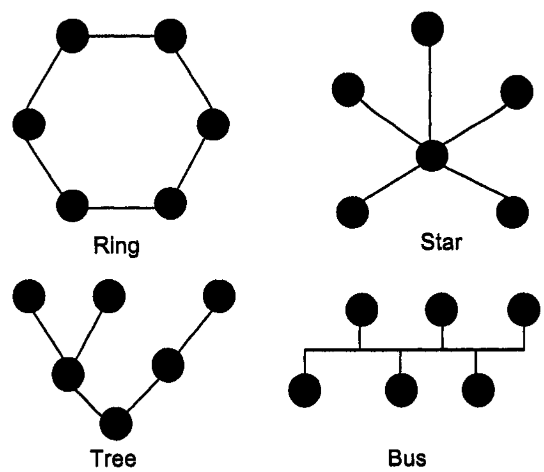
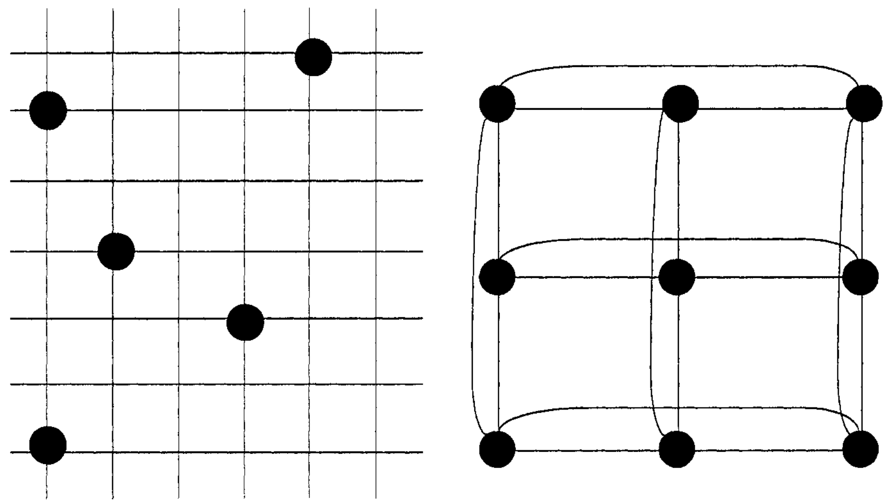


Figure 1: Types of network topologies

Between these types of networks, mesh networks will have a superior advantage since they are more reliable and fault tolerant, meaning that the network will not fail because of the failure of one node or several links. Another good characteristic of a mesh network is that it scales well. Because of the above properties the mesh network is a popular architecture for many applications. Here we present formal definitions for m-dimensional and 2-dimensional mesh networks.

Definition 1. *An m-dimensional mesh network has $k_0 \times k_1 \times \ldots \times k_{m-2} \times k_{m-1}$ nodes, k_i nodes along each dimension i, where k_i denotes the number of nodes in the i-th dimension. Each node in the mesh is identified by an m-coordinate vector $(x_0, x_1, \ldots, x_{m-2}, x_{m-1})$, where $0 \leq x_i \leq k_i - 1$. Two nodes $(x_0, x_1, \ldots, x_{m-2}, x_{m-1})$ and $(y_0, y_1, \ldots, y_{m-2}, y_{m-1})$*



(a) Mesh (b) Torus

Figure 2: Examples of mesh and torus networks

are connected if and only if there exists an i *such that* $x_i = (y_i \pm 1)$, *and* $x_j = y_j$ *for all* $j \neq i$. *Thus the number of neighbors of a node ranges from* m *to* $2m$, *depending on its location in the mesh [27].*

Definition 2. *A 2-dimensional mesh network has* $k_0 \times k_1$ *nodes. Each node* v *has 2 coordinates,* (x, y) *where* $0 \leq x \leq k_0 - 1$ *and* $0 \leq y \leq k_1 - 1$. *If* $(x_0 = x_1)$ *and* $(y_0 = y_1 \pm 1)$ *or* $(y_0 = y_1)$ *and* $(x_0 = x_1 \pm 1)$, *then two nodes* $v_0 = (x_0, y_0)$ *and* $v_1 = (x_1, y_1)$ *are neighbors [14]. In a 2-D mesh, a node can have 2, 3 or 4 neighbors.*

As mentioned before, a mesh network is easy to design and can be implemented with a lower cost and complexity because of its geometrical regularity. In addition to this, the distance and bandwidth of each link is almost the same, therefore the same weight can be assigned to each link to simplify the calculation and achieve a good scalability.

2.1.2 Group Multicast

Multicast is the problem of disseminating a piece of information, owned by a certain node called the source node, to all destination nodes. These destination nodes are the nodes in the multicast group. Multicasting is performed by placing a series of calls along the communication lines of the network. At any time, the informed nodes contribute to the information dissemination process by informing one of their uninformed neighbors. In this thesis, we always use classical multicast model which is the following:

1. In each call, only one informed node and one of its uninformed neighbors are involved.
2. Each call needs one unit of time.
3. A node can participate in one call per unit of time.
4. In one unit of time, many calls in parallel can be performed.

A *multicast scheme* of an originator or source u is a set of calls that completes the multicasting in the network, originated at vertex u. An optimal multicast scheme informs all the destination nodes in the least amount of time.

Definition 3. *Suppose there is a graph called $G(V, E)$, where each node in V corresponds to a node and each edge in E corresponds to a communication link. Also* *consider a set* $K = \{u_0, u_1, u_2, \cdots, u_k\}$ *such that in a communication process, any member* u_i *of* K *may need to multicast data to the other members* ($K - \{u_i\}$)*. The set* K *is a subset of* $V(G)$ *which is called multicast group.*

Unicast-based multicast and *Tree-based* multicast are two important approaches to implement group multicast.

Unicast-based Multicast

One of the ways to implement unicast-based multicast is to send a separate copy of the message from the source node to every destination node. This method is called *separate addressing* [32]. This is a simple but expensive solution because it is done sequentially and it does not use any of the previous links.

In Figure 3, it is shown how a unicast-based multicast works. Let v be a source and $\{A, B, C, D, E\}$ be the multicast group. In this case, v will send a message to A and after that to B, C, D and E one after another. As it can be seen, this approach is expensive both in multicast time and traffic.

Tree-Based Multicast

In *Tree-based multicast* approach, the source node sends the message to a selected set of its neighboring nodes. It will be disseminated from node to node in a certain order. This will continue until all destinations receive the message. All nodes and links which are involved in the dissemination, will occur only once at a certain time and a tree will be formed rooted at the source node. In this way, a multicast scheme can be prescribed as a tree spanning the multicast group members.

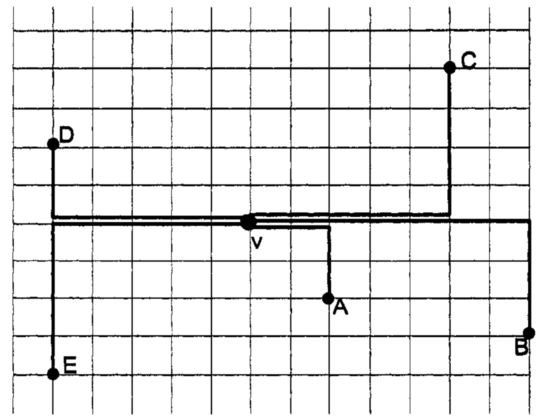


Figure 3: Example of unicast-based multicast

The advantage of tree-based multicast approach comparing to unicast-based multicast is its high efficiency on both time and traffic. This means that the destination nodes will share paths in order to never use the same link twice, which will reduce the amount of total traffic. Also high parallelism in message dissemination will reduce the communication latency significantly. It should be noted that a multicast scheme can be shown by a tree.

Figure 4 illustrates tree-based multicast approach. v is the source node and a tree is built on v. Since most of the paths are shared between v and the destination nodes, multicast latency will be smaller.

Multicast Tree (MT) Problem [20] is an important problem in network communication which asks for a tree involving the optimal scheme. In any multicast tree, as a solution for the multicast tree problem, the distance between two nodes is a major factor that will impact the communication time. In addition to multicast time, there may be

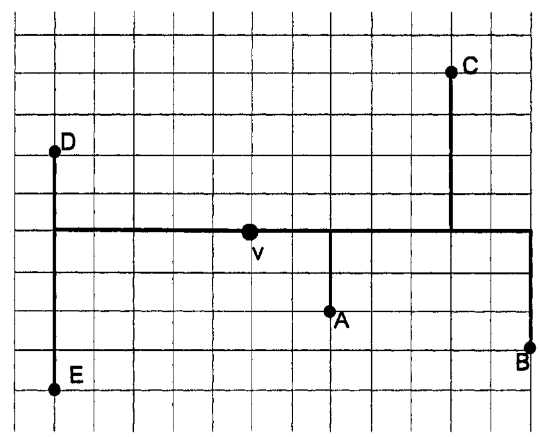


Figure 4: Example of tree-based multicast approach

other parameters that affect the performance of a multicast scheme. Therefore it will be reasonable to minimize time first, and then pay attention to other factors such as traffic.

As mentioned in Chapter 1, multicast trees can be divided into two groups: source-based trees and shared-trees.

Source-based tree routing has traditionally been used on mesh networks. The typical source-tree routing algorithm applies the shortest path tree (SPT) algorithm and therefore separates multicast trees need to be computed, one for each sender. One limiting factor in this case is the processing cost which is used to compute the shortest path tree for each active source. Another problem is that each node has to keep the information pair (source, group).

A shared-tree contains a central point. In this case a tree will be shared among all senders of the group in order to send their messages through it. Therefore for all the sources there will be one single tree which connects them to their destinations. This will improve the scalability of shared-trees [29, 28, 40]. However some problems may arise with the routing efficiency. Since the sender and receiver may not be connected through the shortest path, the end-to-end delay could be higher than the source tree routing counterpart. Another drawback of using a shared-tree in a network is the congestion: if every sender uses the same shared-tree, certain links of the shared-tree will get congested [16].

Because of the drawbacks of source-based trees and shared-trees, multiple shared-trees have been used recently in tree-based multicasting. These trees use more than one multicast tree for the purpose of serving multiple sources and destinations. In multiple shared-trees minimizing the number of shared-trees is an important problem which has been proven to be NP-complete [35].

2.1.3 Multicast Evaluation Criteria

As mentioned earlier, efficiency and scalability are considered to be the main parameters for evaluating multicast communication. A small multicast delay in disseminating a message results to a more efficient multicast. In this section the evaluation of multicast time is described.

To achieve a tree with small multicast delay, the time needed to deliver a message from a source to all destinations in the tree should be calculated. In other words, the maximum time it takes for the last destination to receive the message should be considered.

It should be noted that given a source node in a tree, automatically there would be a direction from the source to the leaves for passing information. So the source can be considered as the root of the tree and the neighbors via outcomming edges can be defined as the children of a node.

Given a tree $T(V, E)$ and a source node $u \in V$, the optimum multicast time $m(u, T)$ can be calculated with a bottom-up approach using the following formulas:

1. If u is an external node which means it does not have any children, $m(u, T) = 0$

2. $m(u, T) = max(i + m(v_i, T_{v_i}))$ in which v_i denotes the ordered set of the children of u and T_{v_i} is the subtree of T rooted at v_i such that subtrees T_{v_i} are labeled in a way that $m(v_i, Tv_i) \geq m(v_{i+1}, T_{v_{i+1}})$.

Figure 5, illustrates how the message transmition delay or latency is calculated.

2.2 Review of Previous Studies

In this section previous works which have been done in multicasting in arbitrary networks are described and numbers of approaches for multicasting in mesh networks are reviewed. After which the core selection methods studied before are being described and finally multi-shared tree related works are summarized.

Most of the research that has been done in multicasting has been to design a multicast algorithm with respect to multicast traffic and delay. For minimizing traffic, the

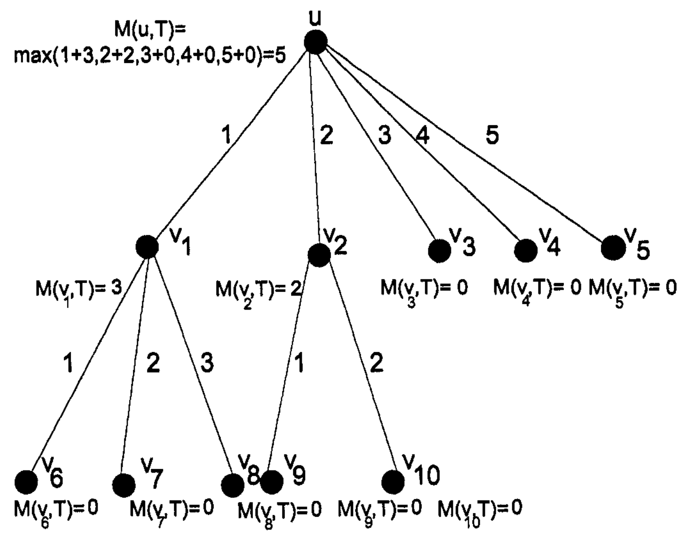


Figure 5: Calculation of optimum multicast time. It will take u, 5 time units to disseminate a message to all the children nodes.)

well known Steiner tree problem is introduced. Previously, many multicast routing algorithms have been proposed for mesh ([32], [25], [19], [34], [42], [31], [46]), but most of them are unicast-based multicast that proved to be inefficient on multicast delay or on multicast traffic [41].

In [22] two tree-based multicast algorithms for mesh and torus networks is proposed: the VH algorithm with time complexity of O(KD), and the DIST algorithm with time complexity of O(KDN), where K is the number of destination nodes, D is the maximum distance to a destination and N is the total number of nodes in the network. Simulations showed that the DIST algorithm generates less traffic compared to the VH algorithm, but at the price of a much higher multicast delay and computation complexity. The VH algorithm however, boasts a very low computation complexity and guarantees that every destination node can receive the message from the source through a shortest path in optimal multicast time (delay), but simulations indicate that it creates a very large amount of congestion which in practice will increase the chances of blockage or even deadlock. Later the MIN algorithm [11] proposed, which is intended to reduce multicast traffic. Simulations showed that although it has a high computation complexity of $O(KDN)$ in mesh, MIN algorithm produces the least traffic among all the algorithms without significantly degrading its performance on multicast time.

Each of the algorithms introduced above will require some kind of improvement on one aspect or another. Algorithms that could achieve an optimal result on an aspect (either multicast time or multicast traffic) without significantly damaging its performance on another and with a relatively low computation complexity are very much needed. Finding a solution that is optimal on all aspects is NP-hard and a solution that tries to find an optimal result on traffic in mesh is NP-complete (Steiner tree problem) [12].

In [12], they use the pro-time strategy and develop two tree-based multicast routing algorithms (DIAG and DDS) that obtain near optimal multicast time and reduce the multicast traffic and computation complexity as much as possible. It should be noted that all of the previous works are based on heuristics and are not optimal algorithms.

As mentioned earlier, the heart of all multicast routing protocols are single source, shortest-path trees and shared, core-based trees. The tradeoffs of shared-trees versus shortest path trees is studied in [43] for the first time. The results of the study in [43] and their simulation show that a Steiner approximation in shared-trees can have lower traffic than shortest-path trees, though the delay will be higher.

In addition to this, research has shown that selecting a satisfied core can improve shared-tree multicast performance. The choice of the core will influence the shape of the multicast routing tree and the performance of the routing schemes will be affected [16]. What happens in the core selection process is that a local router will be chosen as the core and a shortest path tree will be built from this core to all the destinations. Senders will send the data to the core and in turn it will send it to the receivers. One drawback of this approach is that there will be congestion around the cores, therefore studies show that it will be better to have multiple cores [32, 5, 15, 10, 30]. Regarding multiple cores, OCBT (Ordered Core Based Tree Protocol) [37] uses more than one core for fault tolerance. In this case if there is a core failure, it will not affect the tree. OCBT also improves scalability by allowing flexible placement of the cores that are the points of connection to a multicast tree [37]. The problem in OCBT is that each of the cores cooperate to form a single tree, therefore it does not behave differently than a single-core considering delay.

The core in general graphs has to be selected based on certain heuristics. Various heuristics for the core selection were investigated in [39, 2, 23, 14, 1]. Regarding single-core shared-trees, core placement has been more often studied by researchers [21, 3, 36]. In other studies, an initial core is selected, after which it is evaluated periodically to know whether or not a better choice exists. Also investigation is done for core migration between a configured set of candidate cores [6, 45].

The core selection heuristics can be summarized as follows [33]:

- Random Router (not necessarily member of the group)
- Random Member
- Topological Center of the entire network
- Topological Center of the multicast group (this center is not necessarily a member of the group.)
- Topological center of the multicast group (this center is necessarily a member of the group.)
- Random tree node (the nodes belonging to the current multicast tree.)
- Tree Center (the nodes belonging to the center of the current multicast tree.)

To benefit from the advantages of both single source trees and shared-trees, multiple shared trees have been studied. In [16] they constructed multiple shared trees by configuring one primary shared-tree, and based on that tree they made a decision for the specification and configuration of secondary trees. In that research, they also have a core selection. For the primary core selection, the random generation approach has been used. This approach has been investigated in [2]. In [45] they build multicast trees with multiple, simultaneously active, independent cores. Each of these multicast trees has a core which will be the tree center and there are no coordinations or dependencies between cores. Their results show that the average delay can be reduced by using multiple cores. In addition to these, some works are done for multiple shared trees in [38, 45].

As mentioned earlier minimizing the number of multiple-shared multicast trees is proven to be NP-complete [35]. In [35], the NP-completeness of the problem is proved and some heuristic algorithms are proposed. In [26] they propose three other algorithms for minimizing the number of multiple shared trees with much shorter run-time than the proposed algorithm in [35]. Their simulation results show that one of the proposed algorithms in [26] always has a solution as good as the algorithm in [35], while the run time is only one-tenth of the algorithm in [35].

It should be noted that all of the above mentioned research is based on heuristics. What will differentiate this research from them is that here there is a theoretical guarantee for the performance of the method proposed in this thesis.

First, an algorithm for finding the optimum cores in a mesh network is proposed and the related proofs are given. In the second step, the multi-shared trees are built on these cores. These trees that are more efficient and scalable than a single shared-tree are used for multicasting the messages.

In the multicasting algorithm proposed in this thesis, when a large multicast group is distributed uniformly or normally in a mesh layout, the number of optimum centers will tend to one. In this case, one shared-tree will be built and the message delay will almost be optimum.

Chapter 3

Core Selection Algorithm for Multi-Shared Trees

In this chapter a linear algorithm for selecting a set of *optimum centers* associated with a multicast group is presented. The results of this chapter will be used in Chapter 4, where it is shown how each sender can use one of the distribution trees built on these centers to disseminate messages in almost optimum time and an acceptable traffic performance.

The concepts of this chapter are mainly geometrical. The result is that on a mesh graph the distance of two nodes can be stated geometrically. This leads us to a specific type of non-Euclidean geometry called *taxicab geometry*.

3.1 Taxicab Geometry

Taxicab geometry is a form of geometry in which the usual metric of Euclidean geometry is replaced by a new metric. In this metric the distance between two points is the sum of the differences of their coordinates. In this way, the taxicab distance of two points A and B with coordinates (x_1, y_1) and (x_2, y_2) is $|x_1 - x_2| + |y_1 - y_2|$. This distance, denoted by $D_{tx}(A, B)$, is also known as *rectilinear distance* [13]. It is easy to verify that there are $\binom{\Delta y + \Delta x}{\Delta x}$ different shortest paths between any two nodes in mesh in which Δx and Δy stand for $|x_1 - x_2|$ and $|y_1 - y_2|$ respectively. Figure 6 shows two of these paths from A to B.

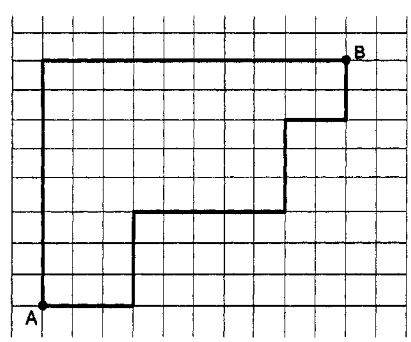


Figure 6: Two shortest paths between points A and B in taxicab geometry. The length of paths is 18.

Compared to Euclidean geometry, taxicab geometry is a much simpler mathematical model, especially for solving problems in urban geometry. To see the difference between the Euclidean geometry and the taxicab geometry, consider the following example:

Problem 1. *Given three points A, B and C, a point P is needed such that* $D_E(P, A) +$

$D_E(P, B) + D_E(P, C)$ *is as small as possible.*

The same problem can be defined in taxicab geometry:

Problem 2. *Given three points* A, B *and* C*, we need a point* P *such that* $D_{tx}(P, A) + D_{tx}(P, B) + D_{tx}(P, C)$ *is as small as possible.*

While it is not trivial to overcome Problem 1, there is a simple solution for Problem 2: Consider a horizontal line l_1 drawn through the point having the median y-coordinate (point C in Figure 7) and a vertical line l_2 through the point with median x-coordinate (point A in Figure 7). Considering an arbitrary point Q on l_1, and walking one unit up to reach Q', the distance to A decreases one unit; while the distance to B and C will increase by one unit each. In this case, the sum of the distances from A, B and C will increase:

$$D_{tx}(Q', A) + D_{tx}(Q', B) + D_{tx}(Q', C) > D_{tx}(Q, A) + D_{tx}(Q, B) + D_{tx}(Q, C).$$

Similarly, beginning anywhere on l_2 and moving horizontally, the sum of the distances from A, B and C will increase, which means any horizontal move off l_2 will increase the sum of the distances from A, B and C. Therefore $D_{tx}(P, A) + D_{tx}(P, B) + D_{tx}(P, C)$ is minimized in the intersection point of l_1 and l_2[18].

3.1.1 General Concepts of Taxicab Geometry

The distance between two nodes in a mesh graph is equivalent to their taxicab distance. This enables us to consider any vertex of a mesh graph as a *point* in the plane. Here, we

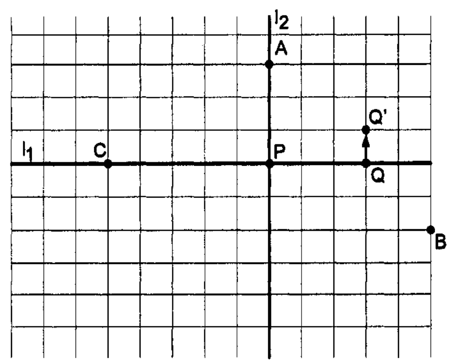


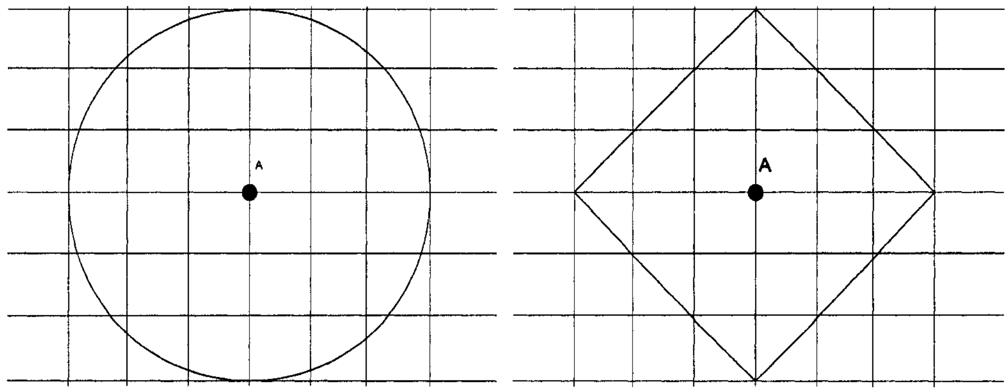
Figure 7: P, in which $D_{tx}(P, A)+D_{tx}(P, B)+D_{tx}(P, C)$ is as small as possible, should be found.

denote multicast nodes with P_α, P_β, etc. while the rest of the nodes within the mesh are denoted by A, B, C, etc.

Since in taxicab geometry the distance is determined by a different metric, shapes of objects change as well. Here circles in taxicab geometry will be introduced. It should be noted that a circle is defined as the set of points having the same distance from a specific point called *center*. It can be easily verified that in taxicab geometry, circles are squares with sides oriented at a 45 degree angle to the coordinate axes. In Figure 8(a), a circle in Euclidean geometry is depicted; while Figure 8(b) shows a circle in taxicab geometry.

Another concept in Euclidean geometry that can be stated in taxicab geometry is triangle non-equality:

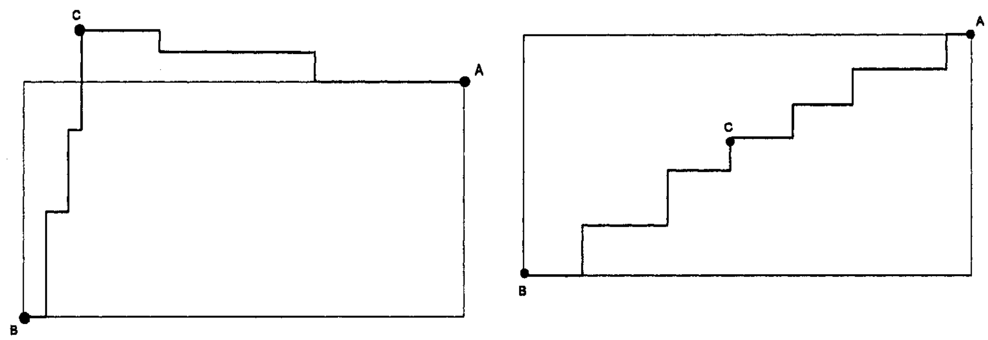
Theorem 1. *For arbitrary points A, B and C, we have* $D_{tx}(A, B) \leq D_{tx}(A, C) + D_{tx}(C, B)$. *The equality occurs if and only if C lies in the rectangle having A and B as its corners [Figure 9].*



(a) Circle in Euclidean geometry (b) Circle in taxicab geometry

Figure 8: Circle in Euclidean and taxicab geometries

Next, *perpendicular bisectors* in taxicab geometry will be introduced. It should be noted that the perpendicular bisector of two points A and B is the set of points with the same distance from them. It is easy to verify that the perpendicular bisector of any two points is a non-straight line consisting of two horizontal or vertical rays, connected by a slant line segment called the *diagonal* of the perpendicular bisector. Figure 10 depicts perpendicular bisectors of two points A and B, when they take different relative positions. It should be noted that the diagonal of the bisector always connects two parallel sides of the rectangle having A and B as two opposite corners. Also, the non-slant parts of the bisector will either be horizontal or vertical depending on the values of Δx and Δy.



(a) $D_{tx}(A, B) < D_{tx}(A, C) + D_{tx}(C, B)$ (b) $D_{tx}(A, B) = D_{tx}(A, C) + D_{tx}(C, B)$

Figure 9: Triangle non-equality in taxicab metric.

3.2 Finding the Optimum Centers of the Multicast Group

In this section, an algorithm for finding optimum centers for a set of multicast nodes distributed in a mesh layout will be presented.

Optimum centers are defined as the set of points (nodes of the mesh) with the smallest maximum distance to the members of the multicast group. The goal is to introduce a linear algorithm for finding the set of optimum centers. To achieve this, some notations and auxiliary results are needed :

Definition 4. *For an arbitrary point A on the mesh, the ultimate node(s) of A are multicast nodes with maximum distance to A. We define any ultimate node of A with $P_{ult}(A)$. Also the ultimate distance of A, denoted by $D_{ult}(A)$ is the distance of A to its ultimate node(s). An optimum center would be a point on the mesh with the smallest ultimate distance.*

Lemma 1. *Let P_α, P_β be two members of the multicast group with maximum distance.*

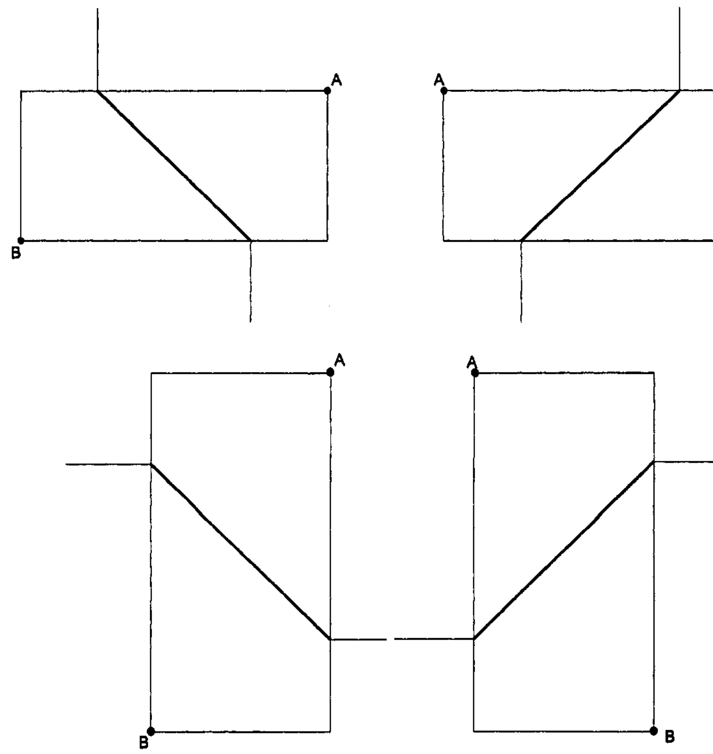


Figure 10: Perpendicular bisectors in taxicab geometry.

Let X be an arbitrary point on the diagonal of the perpendicular bisector of P_α, P_β. Then P_α and P_β are ultimate nodes for X.

Proof. As shown in Figure 11, let another member of the group, for example P_λ be the ultimate node for X, so $D_{tx}(X, P_\beta) < D_{tx}(X, P_\lambda)$. Since X is inside the bounding rectangle of P_α, P_β, using Theorem 1, we have: $D_{tx}(P_\alpha, P_\beta) = D_{tx}(P_\alpha, X) + D_{tx}(X, P_\beta)$. Consequently it can be concluded that $D_{tx}(P_\alpha, P_\beta) < D_{tx}(P_\alpha, X) + D_{tx}(X, P_\lambda)$. Without loss of generality, it can be assumed that P_β is closer to P_λ than P_α; therefore X is also a point on the bounding rectangle of P_α, P_λ implying $D_{tx}(P_\alpha, X) + D_{tx}(X, P_\lambda) =$

$D_{tx}(P_\alpha, P_\lambda)$. The conclusion is that $D_{tx}(P_\beta, P_\alpha) < D_{tx}(P_\alpha, P_\lambda)$ which is a contradiction, since P_α, P_β are considered to be two multicast nodes with maximum distance.

□

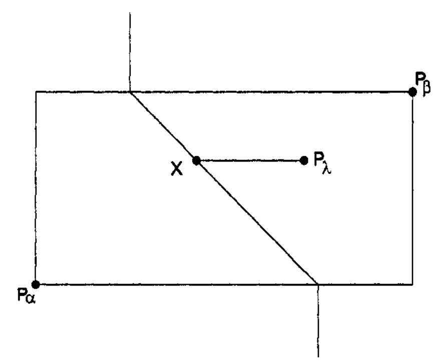


Figure 11: P_α and P_β will be the ultimate nodes for the nodes which are on the diagonal of their perpendicular bisector.

Theorem 2. *Having P_α and P_β as two multicast nodes with the maximum distance, any optimum center of the multicast group lies on the diagonal of the perpendicular bisector of P_α and P_β.*

Proof. As shown in Figure 12, let C be a point which is not on the diagonal of the perpendicular bisector of P_α, P_β. It can be shown that C can not be an optimum center. Without loss of generality, assume that P_β is closer to C than P_α. Also let X be the closest point to C on the diagonal of the perpendicular bisector of P_α, P_β. Employing Theorem 1, we have $D_{tx}(C, P_\alpha) = D_{tx}(C, X) + D_{tx}(X, P_\alpha) > D_{tx}(X, P_\alpha)$. Also, by the definition of ultimate node $D_{ult}(C) \geq D_{tx}(C, P_\alpha)$. Consequently $D_{ult}(C) > D_{tx}(X, P_\alpha)$. By Lemma 1, P_α is an ultimate terminal for X which means $D_{tx}(X, P_\alpha) =$

$D_{ult}(X)$. So we can conclude that $D_{ult}(C) > D_{ult}(X)$ which implies that C can not be an optimum center. □

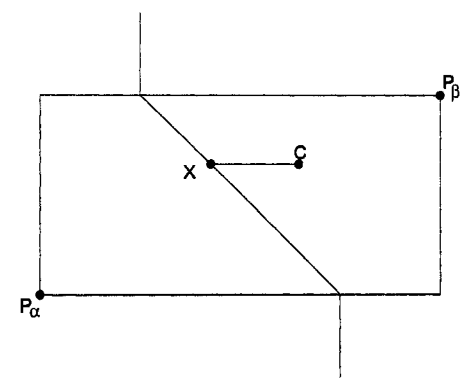


Figure 12: When C does not lie on the diagonal of the bisector of the furthest pair, it can not be an optimum center.

It should be noted that when C is on the perpendicular bisector of P_α, P_β, but not on its diagonal, it can not be the optimum center. As shown in Figure 13, suppose X is the closest point to C that lies on the diagonal of the perpendicular bisector. It is obvious that $D_{tx}(C, P_\beta) = D_{tx}(C, X) + D_{tx}(X, P_\beta)$. So $D_{tx}(C, P_\beta) > D_{tx}(X, P_\beta)$. As mentioned before, the furthest multicast node to C is $P_{ult}(C)$ which means $D_{tx}(C, P_{ult}(C) \geq D_{tx}(C, P_\beta)$. As a result $D_{tx}(C, P_{ult}(C) > D_{tx}(X, P_\beta)$. Since X lies on the diagonal of the perpendicular bisector of P_α and P_β, by Lemma 1, P_β is an ultimate node for X. The conclusion is $D_{tx}(C, P_{ult}(C) > D_{tx}(X, P_{ult}(X)$; therefore C can not be an optimum center.

Corollary 1. *If there is more than a pair of multicast nodes with maximum distance such that the diagonals of their perpendicular bisectors intersect at one point, then there is a*

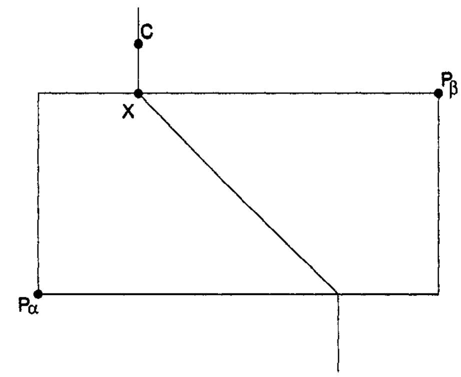


Figure 13: When C lies on the non diagonal part of the bisector of the furthest pair, it can not be an optimum center.

unique multicast center.

Theorem 2 states that it is necessary for optimum centers to be on the diagonal of the perpendicular bisector of the furthest multicast nodes. More constraints are applied in the following theorem:

Theorem 3. *Let $P\alpha$, P_β be two multicast nodes with maximum distance and let P_δ be another multicast node whose perpendicular bisectors with P_α, P_β intersect the diagonal of the perpendicular bisector of P_α, P_β in point N, cutting it to two line segments. To be an optimum center, the point needs to lie on the line segment which is closer to P_δ.*

Proof. As illustrated in Figure 14, the bisectors of (P_δ, P_α), (P_δ, P_β) and (P_α, P_β) all intersect at the same point N. The portion of the bisector of P_α, P_β which is further from P_β can not be a part of the optimum area because for any point N' in that part $D_{tx}(N', P_\delta) = D_{tx}(N', N) + D_{tx}(N, P_\delta) > D_{tx}(N, P_\delta) = D_{ult}(N) \rightarrow D_{tx}(N', P_\delta) >$

$D_{ult}(N)$. Consequently N' can not be an optimum center. The portion of the perpendicular bisector which is closer to P_δ can be a part of the optimum area because for any $N"$ in that part $D_{tx}(N, P_\delta) = D_{tx}(N, N") + D_{tx}(N", P_\delta)$; therefore, $D_{tx}(N", P_\delta) < D_{tx}(N, P_\delta)$ which means applying the effect of P_δ on this area does not change the functionality of the points on this portion as optimum centers.

□

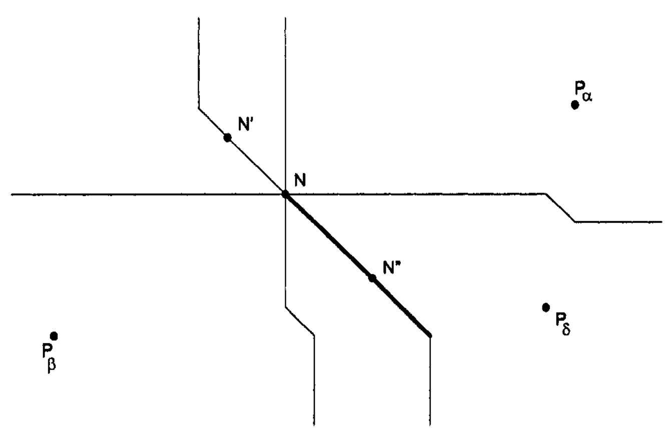


Figure 14: Applying point P_δ on the perpendicular bisector of P_α, P_β (the furthest pair). The potential optimum area is highlighted.

We use Theorems 2 and 3 to design an algorithm for finding the set of optimum centers. In this algorithm, we should find a pair (P_α, P_β) of multicast nodes with maximum distance and create the diagonal of the perpendicular bisector of P_α and P_β. Having m as the size of the multicast group, this diagonal must be cut $m - 2$ times with the perpendicular bisector of other multicast nodes with P_α or P_β. In this way, the diagonal of the original perpendicular bisector of P_α and P_β will become smaller and smaller and the final result will be the set of all multicast centers. It should be noted that if only two multicast nodes exist, the optimum centers will be all the points on the diagonal of their perpendicular bisector. The result of this discussion is illustrated in Algorithm 1.

Algorithm 1 Find_Opt_Center

Input: The set of multicast nodes $M = \{P_1, P_2, ..., P_m\}$ lying on the mesh network
Output: The set S of optimum centers for the input multicast group.

Find two multicast nodes P_α, P_β with maximum distance among all members of M
Initialize a line segment L with the diagonal of perpendicular bisector of P_α, P_β
for any vertex $P_\delta \neq P_\alpha, P_\beta$ **do**
  if the bisector of P_δ, P_α intersects L on a point N **then**
    Let A, B be two endpoints of L
    if $D_{tx}(P_\delta, A) < D_{tx}(P_\delta, B)$ **then**
      $L =$ the line segment (A, N)
    else
      $L =$ the line segment (N, B)
    end if
  end if
end for
$S =$ all the nodes lying on L
return S

We developed a software application which implements Algorithm 1 to find the optimum centers of input multicast groups. Figure 15 shows the output of this application when there are four nodes in the multicast group; while in Figure 16, the procedure of finding the final optimum centers is shown in four consecutive steps. It should be noted that P_α and P_β are the two furthest nodes.

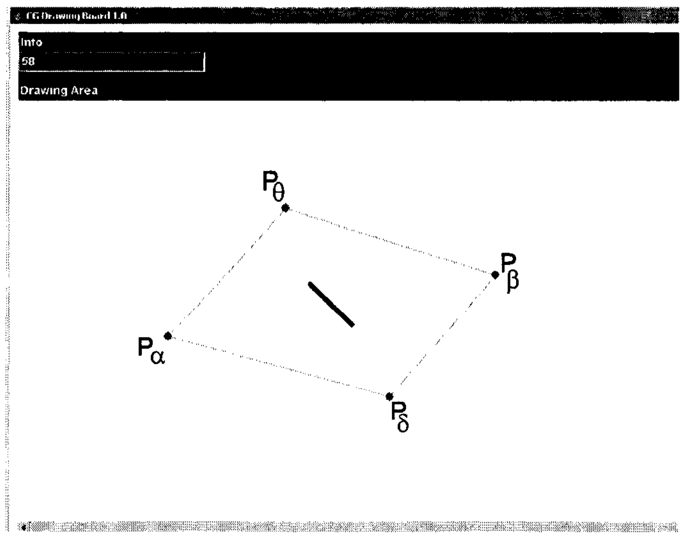
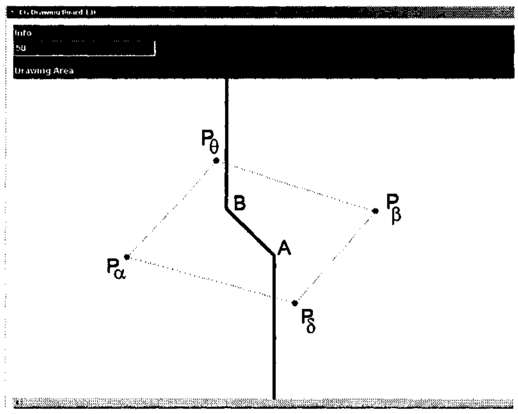


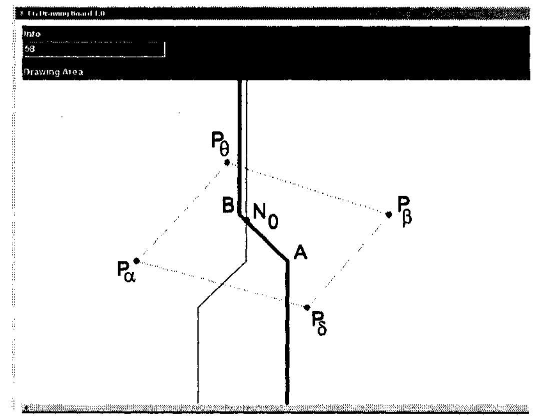
Figure 15: Finding optimum centers for four multicast nodes.

Figure 16(a) shows the very first step of the algorithm in which the optimum area is initiated with the diagonal of the perpendicular bisector of P_α and P_β. This optimum area is the area between A and B in this figure. In Figure 16(b), the perpendicular bisector of P_α and P_δ cuts the current optimum area in N_0. Since $D_{tx}(P_\delta, A) < D_{tx}(P_\delta, B)$, the line segment (A, N_0) will be set as the new optimum area. In Figure 16(c), the perpendicular bisector of P_α and P_θ cuts the optimum area in N_1; therefore the new optimum area will be between N_1 and N_0. Figure 16(d) shows the final result of the algorithm.

The rest of this chapter will prove that the algorithm *Find_Opt_Center* is a linear algorithm with respect to the size of the multicast set.

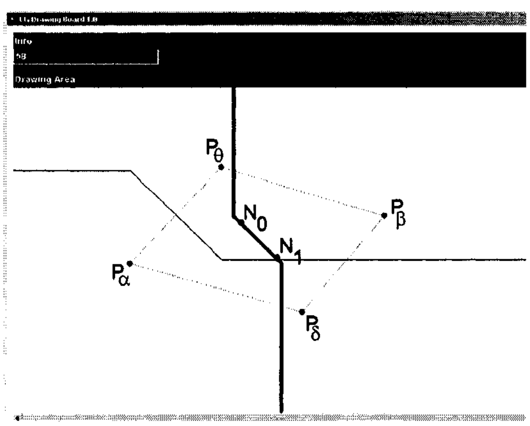
Lemma 2. *Let the rectangle $ABCD$ be a bounding box for the members of the multicast group. Then for any multicast node P_α we have $P_{ult}(P_\alpha) \subset P_{ult}(A) \cup P_{ult}(B) \cup P_{ult}(C) \cup P_{ult}(D)$.*

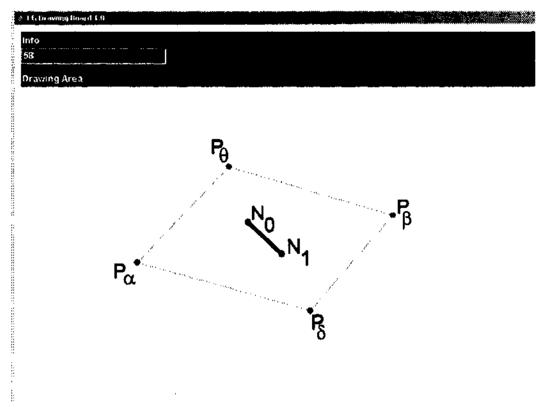




(a) First phase of Algorithm 1: Finding the perpendicular bisector of the two furthest nodes.

(b) Second phase of Algorithm 1: Applying the node P_δ on the result.





(c) Third phase of Algorithm 1: Applying the node P_θ on the result.

(d) The optimum center area for four multicast group members.

Figure 16: Algorithm 1 output: The bold area is the optimum centers area.

Proof. Considering Figure 17, let P_δ be an ultimate node of P_α. It is shown that P_δ is an ultimate node for either A, B, C or D. Drawing the orthogonal axioms on P_δ will divide the box into four segments. Consider that P_α is in segment 1 of Figure 17. Applying Theorem 1 we will have: $D_{tx}(A, P_\delta) = D_{tx}(A, P_\alpha) + D_{tx}(P_\alpha, P_\delta)$. Also P_δ is an ultimate terminal for P_α, which means $D_{tx}(A, P_\delta) \geq D_{tx}(A, P_\alpha) + D_{tx}(P_\alpha, P_\lambda)$ where P_λ is an ultimate node for A. So $D_{tx}(A, P_\delta) \geq D_{ult}(A)$ which means P_δ is an ultimate terminal for A. If P_α lies in other segments, the same result will be achieved just by replacing A with B, C and D. □

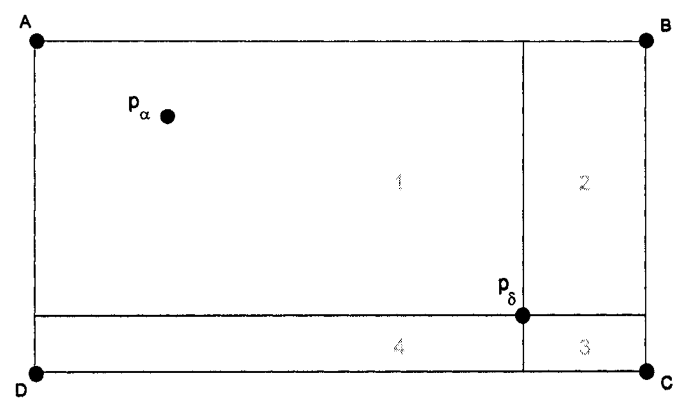


Figure 17: P_δ which is an ultimate node of P_α is an ultimate node of A as well.

Theorem 4. *It is possible to find two multicast nodes with maximum distance in $O(m)$ time where m is the size of the multicast group.*

Proof. It should be noted that the two multicast nodes with maximum distance are the ultimate nodes of each other. Also, having the rectangle $ABCD$ as a bounding box for the multicast members, applying Lemma 2 it is obvious that the two points with maximum distance lie in the set $S = P_{ult}(A) \cup P_{ult}(B) \cup P_{ult}(C) \cup P_{ult}(D)$. Therefore it is necessary to find the ultimate nodes of A, B, C and D, and among them the pair with maximum distance would be the two furthest multicast nodes. It should be mentioned that any of the points A, B, C and D has exactly one ultimate destination. Note that if a point A has two ultimate destinations (P_α, P_β), based on Lemma 1, A should lie on the diagonal of the perpendicular bisector of these two furthest nodes, contradicting it to be a corner of the bounding box.

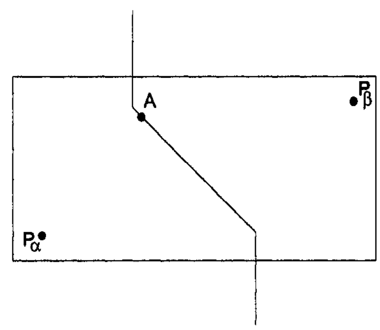


Figure 18: If A has two ultimate nodes (P_α, P_β), it can not be a corner of the bounding box

□

Theorem 5. *It takes linear time for the algorithm Find_Opt_Center to complete.*

Proof. Theorem 4 states that line 1 of *Find_Opt_Center* can be completed in linear time. The intersection of two lines in taxicab geometry can be found in constant time, therefore line 4 of Algorithm 1 takes constant time. Also the distance between two points in taxicab geometry can be calculated in constant time, in turn line 6 of the algorithm will also take constant time. As a result, each iteration of the 'For loop' takes constant time. It can be concluded that generally the algorithm *Find_Opt_Center* has $O(m)$ time complexity, where m is the size of the multicast group. □

Chapter 4

Creating multi-shared trees

Group multicasting is a problem in which there are many sources that send information and the goal is to minimize objectives such as delay and traffic. As mentioned in Chapter 2, the solution can be achieved through the use of multiple-shared multicast trees.

In this chapter, we use concepts from the previous chapter to create multi-shared trees that are *almost* optimum with respect to multicast time. The strategy is to create a set of shortest path trees having their roots on the optimum centers. Any sender employs the tree with the closest root to disseminate the message. It is shown that, although this tree is not a shortest path tree for the sender, it distributes the message in almost optimal time.

In Section 4.1, it is studied that a single sender can have several closest optimum centers while in Section 4.2, it is demonstrated how a shortest path tree rooted at any of these closest optimum centers can be used to disseminate messages in almost optimum time.

4.1 The Number of Optimum Centers

Depending on the position of the source node in the mesh layout, sometimes for one sender there may be more than one closest optimum centers or roots. In this situation, the sender, uses a tree rooted at an arbitrary closest optimum center (we usually consider that the sender employs the optimum center which is on the horizontal axis with the same y-coordinate as the sender but if we choose the optimal center which is on the vertical axis as the sender the result will be the same).

In Figure 19 the plane is divided into 9 regions. Region 0 contains the points lying on the rectangular area having the set of all optimum centers as its diagonal. This rectangle is called the *central rectangle*. As shown in the figure, if a sender S_0 lies inside the central rectangle, the optimum centers with the same x or y coordinates (O and O') will be the two closest optimum centers to S_0. It is easy to verify that any optimum center between O and O' (such as O'') has also the same distance to S_0. As a result, S_0 would have $k' \leq k$ optimum centers where k is the number of optimum centers; the equality occurs when a sender S_m lies on a vertex of the central rectangle.

It is also easy to verify that the number of optimum centers for the senders in regions 3 and 6 is equal to k. The reason for this is that it should be noted that for any sender S_1 in region 3 or 6, and also any arbitrary optimum center O_{opt}, using Theorem 1, we have $D_{tx}(S_1, O_{opt}) = D_{tx}(S_1, S_m) + D_{tx}(S_m, O_{opt})$ in which S_m is the closest vertex of the central rectangle to S_1. As mentioned before, S_m has exactly k closest optimum centers, which means $D_{tx}(S_m, O_{opt})$ is a fixed value for any O_{opt}. So $D_{tx}(S_1, O_{opt})$ is a fixed value as well (for an arbitrary optimum center O_{opt}); which means that all the optimum centers have the same distance to S_1.

Similarly, it can be shown that for any vertex S_2 in regions 1 and 8 of Figure 19, there is exactly one closest optimum center, which is the one lying on the closer vertex of the central rectangle.

Applying the same reasoning, in the other four regions (regions 2 , 4 , 5 , 7), any sender S_3 will have k' closest optimum centers ($1 \leq k' \leq k$).

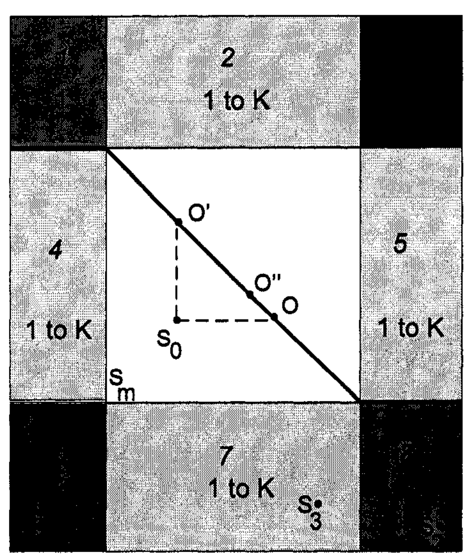


Figure 19: Number of closest optimum centers for senders located in different regions.

4.2 Building Shared-Trees

After choosing one of the closest optimum center(s), multi-shared trees will be constructed. For any optimum center, we need to create a separate shared-tree which is a shortest path tree rooted at that optimum center. Later we will show that there are no constraints for creating the trees as long as they are the shortest path trees for the optimum centers; therefore we can employ any method for creating shortest path trees that not only have almost optimum multicast time, but also have good traffic performance.

Here we present a very simple method for creating the shortest path trees as a base line for future approaches. The tree assigned to the optimum center O will be initiated with the links lying on a horizontal line segment passing O. Then, the links on the vertical segments covering destination nodes will be added in order to come up with a spanning tree. Figure 20 illustrates a multicast tree realized by the above mentioned approach. It is easy to verify that the tree, designed in this way is a shortest path tree for the optimum center O.

As mentioned before, any sender employs the tree rooted at the closest optimum center to disseminate the message.

Figure 21 shows the trees associated with two optimum centers O, O'. The multicast set in this example is $M = \{P_\alpha, P_\beta, P_\delta, P_\lambda, P_\theta, P_\varphi\}$; while P_α and P_β are the two furthest nodes. In this example, if P_φ needs to send a message, it will use the tree rooted at its closest optimum center (O) to multicast the message. The same way, P_θ employs the tree rooted at O'.

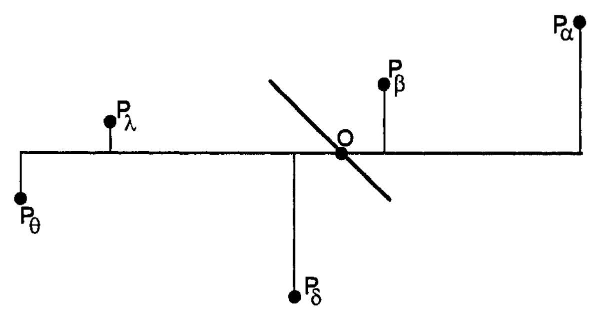


Figure 20: Example of a shared-tree.

It should be noted that using a shared tree is different from unicasting to the root and then multicasting to the destination nodes. For example in Figure 21, when P_θ multicasts the message, for a destination P_β, the message directly goes from P_θ to P_β (instead of being sent to O' first and then to P_β).

It should be reminded that the shared-tree employed by a sender as the multicast tree, may not be a shortest path tree for that sender. Figure 22 shows an example in which the shared-tree is not a shortest path tree for the sender P_θ. It is easy to see that the path from P_θ to P_δ on the shared-tree, is not the shortest path on the mesh. However we will show that the shared-trees created by this approach have almost optimum performance with regards to multicast delay.

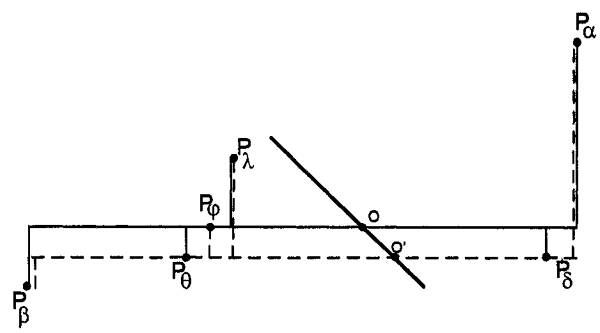


Figure 21: Example of having two shared-trees for two senders P_θ and P_φ. The tree for P_φ is the tree with the solid line and the tree for P_θ is dotted.

4.2.1 Multicast Delay Performance

This section will prove that the multi-shared tree approach presented has good performance with respect to multicast time. The following theorem provides an obvious lower bound for the multicast time of a sender:

Theorem 6. *Let $t_m(P_\varphi, M)$ denote the time needed to disseminate a message from a node P_φ to a multicast destination set M. Then $t_m(P_\varphi, M) \geq D_{ult}(P_\varphi)$ in which $D_{ult}(P_\varphi)$ is the maximum distance of P_φ to the nodes of M.*

A case in which the lower bound occurs is when the sender P_φ has exactly one ultimate node. Figure 23 shows that the minimum time taken for a message sent from P_φ to be received by all the destinations is equal to the distance of P_φ to P_α where P_α is the only furthest node to P_φ.

In the same way, it can be verified that if there are two furthest nodes with the

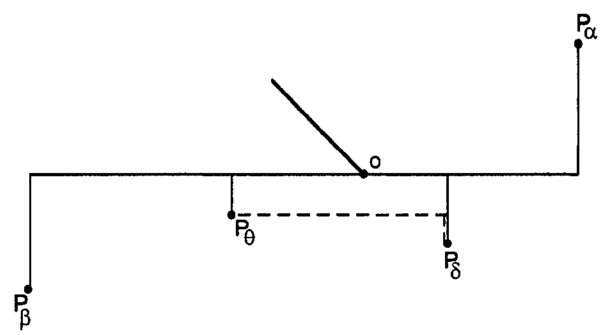


Figure 22: An example in which the shared-tree is not a shortest path tree for the sender P_θ. A shortest path between P_θ and P_δ is dashed.

same distance from the sender P_φ, it will take $D_{ult}(P_\varphi) + 1$ time for the message to be received by all of the multicast destination nodes. Figure 24 shows an example of this case, where P_α and P_β are the two furthest nodes to the node P_φ (P_φ lies on the perpendicular bisector of P_α and P_β). The distance from these two nodes to P_φ is 10; while the minimum time to multicast a message from P_φ is 11 (Figure 24).

Theorem 7. *Any multicast node P_φ and an arbitrary closest optimum center O to P_φ share an ultimate node.*

Proof. Based on Lemma 1, the furthest nodes for O are P_α and P_β (see Figure 25). Assume that P_φ is closer to P_β than P_α. Consider a point $P_\lambda \neq P_\alpha$ be the furthest multicast node for P_φ. By Theorem 1, $D_{tx}(P_\varphi, P_\lambda) \leq D_{tx}(P_\varphi, O) + D_{tx}(O, P_\lambda)$. Also $D_{tx}(O, P_\lambda) < D_{tx}(O, P_\alpha)$ because P_α is an ultimate node for O, therefore $D_{tx}(P_\varphi, O) + D_{tx}(O, P_\lambda) < D_{tx}(P_\varphi, O) + D_{tx}(O, P_\alpha)$. So $D_{tx}(P_\varphi, P_\lambda) < D_{tx}(P_\varphi, P_\alpha)$. Therefore

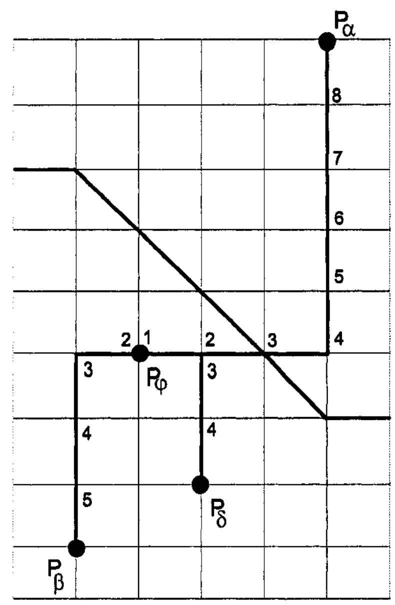


Figure 23: Example of having one furthest node for the sender P_φ for calculating the multicast time.

P_λ can not be the furthest point for O. □

Next it is shown that the multicast time in the multi-shared tree approach will be almost optimum:

Theorem 8. *If a multicast node P_φ employs the shortest path tree T rooted at the closest optimum center to distribute a message, the multicast time will be at most $D_{ult}(P_\varphi) + 3$.*

Proof. Let O be the closest optimum center to P_φ. Assume that P_φ sends the message to O and then O multicasts the message to the other multicast nodes. It takes $D_{tx}(P_\varphi, O)$ to inform O. Also the time that it takes for O to multicast the message through T is at most $D_{ult}(O) + 3$. The reason is that T is a shortest path tree for O and the degree

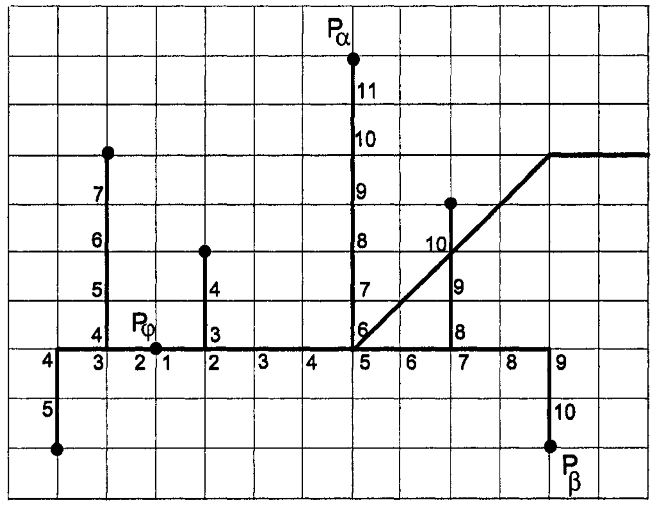


Figure 24: Example of having two furthest nodes to the sender P_φ for calculating multicast time.

of T is at most 4 (due to the mesh layout). Based on Theorem 7 we know that O and P_φ share the same ultimate node, therefore $D_{tx}(P_\varphi, O) + D_{ult}(O) = D_{ult}(P_\varphi)$. As a result, if P_φ disseminates the message through T, the multicast time will be at most $D_{ult}(P_\varphi) + 3$. □

Figure 26 shows an example in which the worst multicast time mentioned in Theorem 8 happens. It can be easily verified that this upper bound occurs when there exists two separate pairs (P_α, P_β) and (P_δ, P_θ) of destination nodes with maximum distance. As a result of Corollary 1, there would be a unique optimum center O for all the senders.

Consider a situation in which there is a multicast node P_O located on the unique optimum center O. When this node needs to multicast a message, for informing the 4 ultimate destinations, $D_{ult}(P_O) + 3$ rounds is needed; which is the worst case claimed in

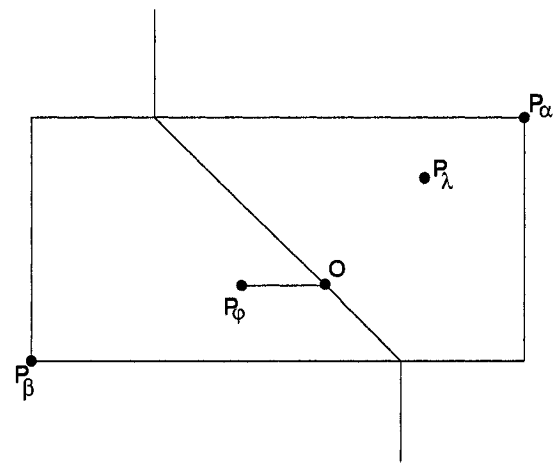


Figure 25: O and P_φ have the same ultimate node (P_α).

TheoremupperBond. It should be noted that if any other multicast node $P_\varphi \neq P_O$ needs to disseminate the message, the multicast time reduces to $d_{ult}(P_\varphi) + 2$ (since there are three destinations that get informed through P_O).

Theorems 6 and 8 state that if each sender uses the shared-tree with closest optimum center or root to distribute messages, the multicast time would be optimum with an additional constant value of 3. Using the multi-shared trees stated above, in its worst case, the multicasting would be performed in three time units more than the optimum time.

4.2.2 Traffic Performance

As mentioned before, any method for creating the set of shortest path trees can be applied to obtain multicast trees with almost optimal time. Therefore, we have the flexibility to create trees with good traffic performance. Creating shortest path trees with small

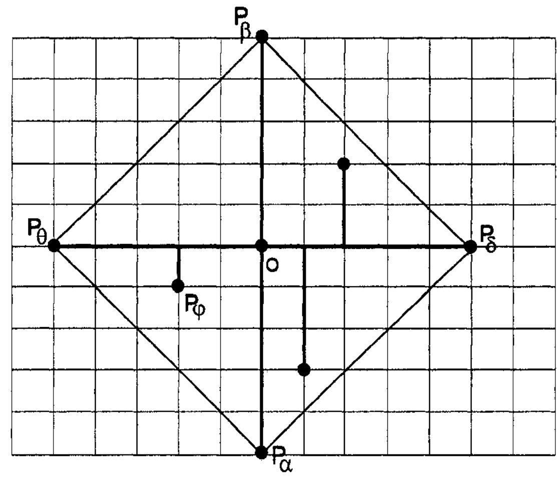


Figure 26: Example of having four furthest nodes. In this case the multicast time will be at most $D_{ult}(P_{\varphi}) + 2$ for a sender P_{φ} and $D_{ult}(O) + 3$ if O is the sender.

traffic has already been studied in [44] and [22]. The method called Diag presented in [12] looks to be more effective. However all of the methods studied in the past were designed for creating a single shared-tree and not multi-shared trees. A novel method for creating shortest path trees in multi-shared trees with smaller traffic is a subject for future work.

Chapter 5

Scalability

In the previous chapters, we presented how to find optimum centers for a multicast group and then build multiple shared trees rooted at these centers. In this chapter, we study the total number of needed shared-trees. Note that in the worst case, we need to create $min(m, k)$ multi-shared trees where m is the size of the multicast group and k is the number of optimum centers.

However, in most cases the number of trees is much smaller. It can be verified that when there is a large multicast group distributed in a mesh layout with a uniform or normal distribution (for large ks), the number of multicast centers(m) tends to 1. An informal explanation is that the convex hull of a large number of multicast nodes is expected to have a symmetric shape. It implies that there would be several pairs of multicast nodes with maximum distance. In this case, as mentioned in Corollary 1, there exists one optimum center and consequently one shared tree.

For example, when the number of nodes which are distributed uniformly increases, the convex hull of the multicast nodes will be a square. As shown in Figure 27, there will be two pairs of multicast nodes (P_α, P_β) and (P_δ, P_λ) with the furthest distance, and the diagonals of their perpendicular bisectors will intersect at one point, which is the center of the square.

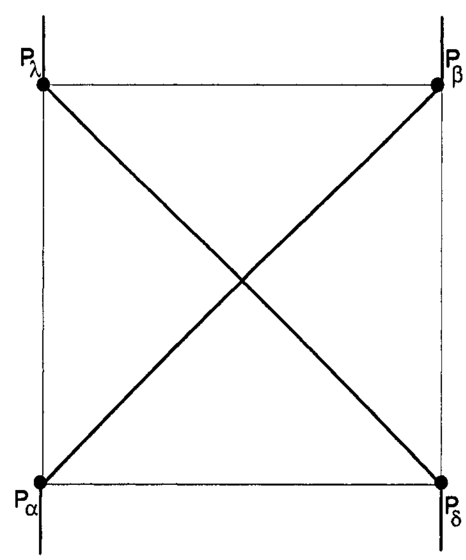


Figure 27: When there are two pairs of furthest nodes, the diagonals of their perpendicular bisectors will intersect at one point.

Using computer simulation, we studied how the number of optimum centers is affected by the size of the multicast group. In this application, a 500 * 500 mesh layout is considered on which the multicast nodes are distributed based on a random value with uniform or normal distribution for x and y coordinates.

Table 1 contains the results of the simulation. It should be mentioned that the presented numbers are the average result for 500 randomly generated multicast groups. The first column of this table shows the number of multicast nodes in multicast groups; while the second and third columns include the average number of optimum centers. As expected, the results of Table 1 confirms the idea that the number of optimum centers reduces for larger multicast groups.

Appendix A and also Appendix B contain snap shots of the simulation. Appendix A shows the number of optimum centers when nodes are distributed uniformly. Appendix B is the output of our simulation for nodes distributed normally.

Number of nodes	Number of optimum centers (uniform)	Number of optimum centers (normal)
100	41	56
150	32	56
200	29	48
300	24	46
400	20	45
500	19	41
750	15	40
1000	13	39
2000	10	36
5000	6	31
10000	4	30
20000	3	26
50000	1	12

Table 1: The number of optimum centers when the nodes are distributed uniformly or normally.

As mentioned earlier when the number of multicast nodes that are distributed uniformly increase, the convex hull for these nodes tends to be the boundary square of the mesh network. For example Figure 28 shows 20000 multicast nodes distributed uniformly in a mesh layout. In this figure it can be seen that the convex hull created for these nodes is almost a square.

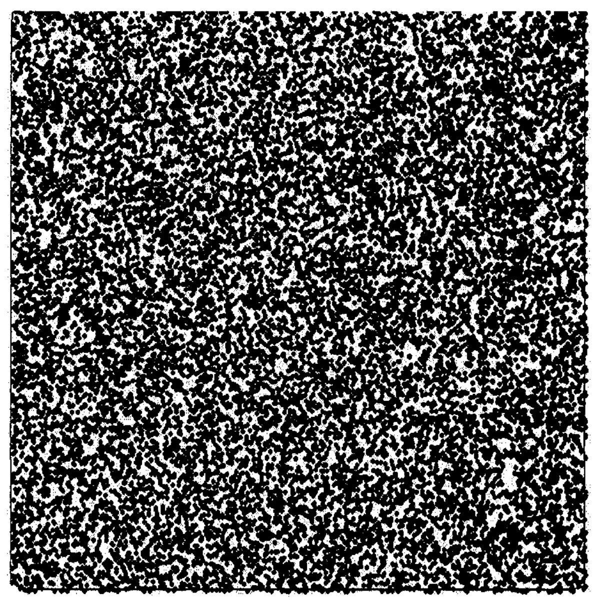


Figure 28: A large number of nodes distributed uniformly (20000 nodes).

Note that if the underlying mesh network be a $k \times k'$ mesh ($k \neq k'$), the convex hull turns out to be a rectangle. In this case again, there would be two pairs with maximum distance and just one optimum center would exist.

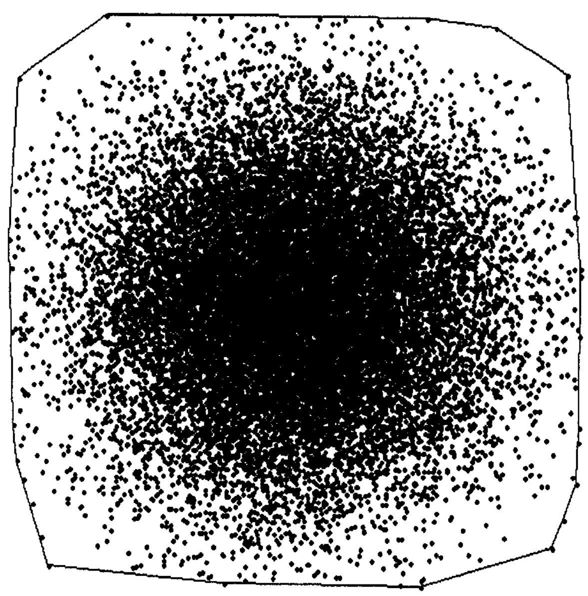


Figure 29: A large number of nodes distributed normally (20000 nodes).

Chapter 6

Conclusion and Future Work

In this thesis we studied the problem of group multicasting in mesh-connected networks. To compromise between multicast performance and scalability factors, we used the multi-shared tree approach. For building these trees we needed to find the location of their roots or centers. Based on taxicab geometry concepts, a linear core selection algorithm was presented. We found a pair of multicast nodes with maximum distance, after which we create the diagonal of perpendicular bisector of these two nodes. Depending on the size of multicast group (m), we cut this diagonal $m - 2$ times with the perpendicular bisector of the two nodes and other multicast nodes. After that, by using mathematical induction on the number of destination points, we showed that the result would be the set of all multicast centers. The result of this algorithm was employed to create a set of distribution trees that enables senders to disseminate messages in almost optimal time and effective traffic performance. We showed that a lower bound for multicast time of a sender v is $D_{ult}(v)$ and we also described that $D_{ult}(v)$ is the distance of v to its furthest multicast node. Also, we demonstrated that if we wanted to send a message from multicast node v to all the multicast group nodes, it employs the shortest path tree T rooted at the closest optimum center, after which the message will be sent to all the destinations. In this situation the multicast time would be at most $D_{ult}(v) + 3$ meaning that in worst case the multicasting would be performed in three time units more than the optimum time. While multicast efficiency is preserved, this approach has better scalability compared to conventional source-based approaches. Finally we had our simulation results for nodes which were distributed in a $(500 * 500)$ mesh layout either uniformly or normally. Our results showed that the number of optimum centers or cores gets smaller and tends to 1 when the number of multicast nodes, distributed either uniformly or normally, has been increased.

As mentioned in 4.2.2, existing methods for reducing traffic are designed for creating single shared trees, therefore a novel method for creating multicast trees with smaller traffic for multiple shared trees is a subject for future work.

In addition to this, in this thesis multi-shared trees for multicasting in mesh networks are used. Extending this approach to the similar network structures is a subject for future work. These structures include torus networks and hexagonal graphs [4] in which all the vertices have uniform constant degree.

Appendix A

Examples of optimum centers for uniformly distributed multicast nodes

In Figure 30, optimum centers are lied on the diagonal of the perpendicular bisector of the two furthest multicast nodes for a multicast group of 300 nodes distributed uniformly. Also Figure 31 shows the number of optimum centers for 2000 nodes.

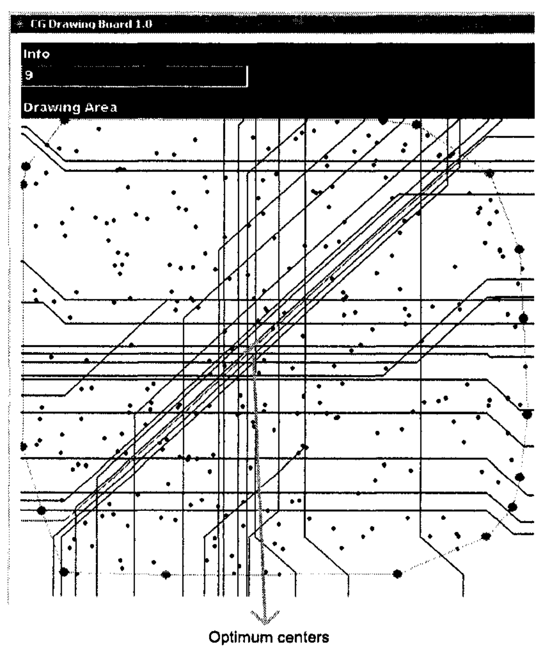


Figure 30: An example of 300 nodes distributed uniformly. The number of optimum centers in this case is 9.

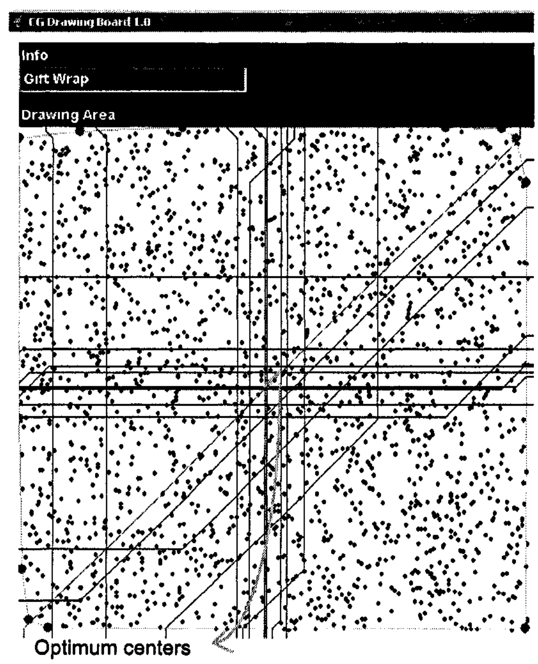


Figure 31: An example of 2000 nodes distributed uniformly. The number of optimum centers in this case is 9.

Appendix B

Examples of optimum centers for normally distributed multicast nodes

In Figure 32, optimum centers are lied on the diagonal of the perpendicular bisector of the two furthest multicast nodes for a multicast group of 300 nodes distributed normally. In addition to this, Figure 33 shows the number of optimum centers that is the output of our software application for 2000 nodes.

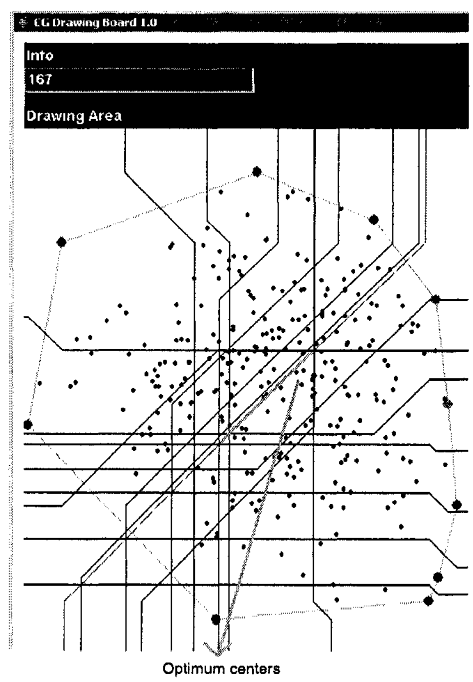


Figure 32: An example of 300 nodes distributed normally. The number of optimum centers in this case is 167.

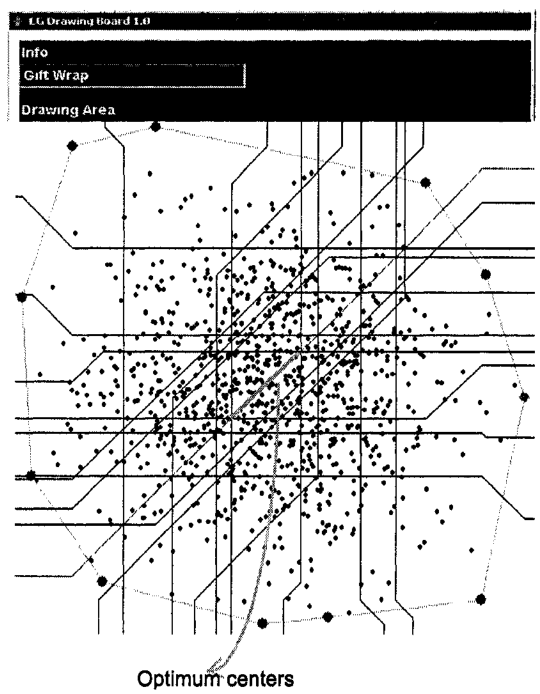


Figure 33: An example of 1000 nodes distributed normally. The number of optimum centers in this case is 137.

Bibliography

[1] T. Ballardie, P. Francis, and J. Crowcroft. Core based trees (CBT). *SIGCOMM Computer Communication Rev.*, 23(4):85–95.

[2] K. L. Calvert, E.W. Zegura, and M. J. Donahoo. Core selection methods for multicast routing. *ICCCN*, 00:638–642, 1995.

[3] C. Y. Chang, Y. S. Yen, C. W. Hsieh, and H. C. Chao. An efficient rendezvous point recovery mechanism in multicasting networks. In *Proceedings of international conference on Wireless communications and mobile computing*, pages 523–528, 2007.

[4] M. S. Chen, K. G. Shin, and D. D. Kandlur. Addressing, routing, and broadcasting in hexagonal mesh multiprocessors. *IEEE Transactions on Computers*, 39(1):10–18, 1990.

[5] F. Torre de Mer and D. Mlynek. Applying clustering algorithms as core selection methods for multiple core trees. In *Proceedings of International Conference on Communications*, Paris, France, 2004.

[6] M. J. Donahoo and E. W. Zegura. Core migration for dynamic multicast routing. Technical Report GIT-CC-95-28, CiteSeer [http://cs1.ist.psu.edu/cgi-bin/oai.cgi], United States, 1997.

[7] D. Estrin, D. Farinacci, A. Helmy, D. Thaler, S. Deering, M. Handlev, V. Jacobson, C. Liu. P. Sharma., and L. Wei. Protocol Independent Multicast-Sparse Mode (PIM-SM): protocol specification. RFC 2362, 1998.

[8] E. D. Fabricius. *Introduction to VLSI design.* McGraw-Hill, New York, NY, USA, 1990.

[9] A. Fei, Z. Duan, and M. Gerla. Constructing shared-tree for group multicast with QoS constraints. *Global Telecommunications Conference*, 4:2389–2394, 2001.

[10] F. Font and D. Mlynek. Choosing the set of Rendezvous points in shared trees minimizing traffic concentration. *IEEE International Conference on Communications*, 3:1526–1530, 2003.

[11] H. A. Harutyunyan and X. Liu. New multicast algorithms in mesh-connected networks. *International Symposium on Performance Evaluation of Computer and Telecommunication Systems*, pages 284–291, 2003.

[12] H. A. Harutyunyan and S. Wang. Efficient multicast algorithms for mesh-connected multicomputers. In *Proceedings of the conference on Information Visualization*, pages 504–510, London, UK, 2006.

[13] J. M. Ho, M. T. Ko, T. H. Ma, and T. Y. Sung. Algorithms for rectilinear optimal multicast tree problem. In *Proceedings of the Third International Symposium on Algorithms and Computation*, pages 106–115, London, UK, 1992.

[14] W. Jia, L. Cheng, and G. Xu. Efficient multicast routing algorithms on mesh networks. *Fifth International Conference on Algorithms and Architectures for Parallel Processing*, pages 110–117, 2002.

[15] W. Jia, W. Tu, and J. Wu. Distributed hierarchical multicast tree algorithms for application layer mesh networks. *IEICE - Trans. Inf. Syst.*, E89-D(2):654–662, 2006.

[16] W. Jia, W. Tu, W. Zhao, and G. Xu. Multi-shared-trees based multicast routing control protocol using anycast selection. *The International Journal of Parallel, Emergent and Distributed Systems* , 4(1):69–84, 2005.

[17] X. Jia and L. Wang. A group multicast routing algorithm by using multiple minimum steiner trees. *Computer Communications*, 20(9):750–758, 1997.

[18] E. F. Krause. *Taxicab geometry: an adventure in non-euclidean geometry*. W. H. Freeman & Co., New York, NY, USA, 1987.

[19] X. Lin, P. K. McKinley, and L. M. Ni. Deadlock-free multicast wormhole routing in 2-D mesh multicomputers. *IEEE Transactions on Parallel and Distributed Systems*, 5(8):793–804, 1994.

[20] X. Lin and L. M. Ni. Multicast communication in multicomputer networks. *IEEE Transactions on Parallel and Distributed Systems*, 04(10):1105–1117, 1993.

[21] Y. D. Lin, N. B. Hsu, and C. J. Pan. Extension of RP relocation to PIM-SM multicast routing. *IEEE International Conference on Communications*, 1:234–238 vol.1, 2001.

[22] X. Liu. Multicasting algorithms for mesh and torus networks, Concordia University Library, Canada. 2003.

[23] J. F. Macq, L. A. Wolsey, and B. Macq. A rendezvous point selection algorithm for videoconferencing applications. *IEEE International Conference on Multimedia and Expo*, 1:689–692 vol.1, 2002.

[24] A. J. Martin. The torus: An exercise in constructing a processing surface. In *2nd Caltech Conf. VLSI*, pages 527–537, 1981.

[25] P.K. McKinley, H. Xu, A.H. Esfahanian, and L.M. Ni. Unicast-based multicast communication in wormhole-routed networks. *IEEE Transactions on Parallel and Distributed Systems*, 05(12):1252–1265, 1994.

[26] M. Moh and B. Nguyen. Minimizing multiple-shared multicast trees for QoS-guaranteed multicast routing. *Eight International Conference on Computer Communications and Networks*, pages 132–138, 1999.

[27] P. Mohapatra. Wormhole routing techniques for directly connected multicomputer systems. *ACM Comput. Surv.*, 30(3):374–410, 1998.

[28] J. Moy. Multicast extensions to ospf.

[29] J. Moy. Ospf version 2.

[30] R. Mukherjee and J. W. Atwood. Rendezvous point relocation in protocol independent multicast-sparse mode. *Telecommunication Systems*, 24(2-4):207–220, 2003.

[31] N. Nupairoj, L.l M. Ni, J.-Y. L. Park, and H.A. Choi. Architecture-dependent tuning of the parameterized communication model for optimal multicasting. *International Parallel Processing Symposium*, 00:578, 1997.

[32] J. Y. Lee Park, H. A. Choi, N. Nupairoj, and L. M. Ni. Construction of optimal multicast trees based on the parameterized communication model. In *International Conference on Parallel Processing, Vol. 1*, pages 180–187, 1996.

[33] P. Paul and S. Raghavan. Survey of multicast routing algorithms and protocols. In *Proceedings of the Fifteenth International Conference on Computer Communication*, pages 902–926, Washington, DC, USA, 2002.

[34] D. F. Robinson, P. K. McKinley, and B. H.C. Cheng. Optimal multicast communication in wormhole-routed torus networks. *IEEE Transactions on Parallel and Distributed Systems*, 06(10):1029–1042, 1995.

[35] H. F. Salama. *Multicast routing for real-time communication of high-speed networks*. PhD thesis, North Carolina State University, 1996.

[36] H. F. Salama, D.S.Reeves, and Y.Viniotis. Shared multicast trees and the center selection problem: a survey, 1996.

[37] C. Shields and J. J. Garcia-Luna-Aceves. The ordered core based tree protocol. *INFOCOM*, 00:884, 1997.

[38] S. W. Tan, G. Waters, and J. Crawford. A multiple shared trees approach for application layer multicasting. *IEEE International Conference on Communications*, 3:1456–1460, 2004.

[39] D. Thaler and C. V. Ravishankar. Distributed center-location algorithms. *IEEE Journal on Selected Areas in Communications*, 15(3):291–303, 1997.

[40] D. Waitzman, C. Partridge, and S. E. Deering. Distance vector multicast routing protocol. RFC 1075, 1988.

[41] S. Wang. Efficient multicast algorithms for mesh-connected multicomputers. Master's thesis, Montreal, QC, Canada, 2005.

[42] S. Y. Wang, Y. C. Tseng, and C. W. Ho. Efficient multicast in wormhole-routed 2D mesh/torus multicomputers: A network-partitioning approach. In *FMPSC: Frontiers of Massively Parallel Scientific Computation*. National Aeronautics and Space Administration (NASA), IEEE Computer Society Press, 1996.

[43] L. Wei and D. Estrin. The trade-offs of multicast trees and algorithms. In *Proceedings of ICCCN*, 1994.

[44] J. S. Yang and C. T. King. Efficient tree-based multicast in wormhole-routed 2D meshes. pages 494–500, Washington, DC, USA, 1997.

[45] D. Zappala, A. Fabbri, and V. Lo. An evaluation of shared multicast trees with multiple active cores. In *Proceedings of the First International Conference on Networking-Part 1*, pages 620–629, London, UK, 2001.

[46] L. Ziqiang and J. Duato. Adaptive unicast and multicast in 3D mesh networks. *Proceedings of the Twenty-Seventh Hawaii International Conference on System Sciences, Architecture*, 1:173–182, 1994.