Chapter 4 proposes performance measures for broadcast communication in such networks. The measures determine the overhead imposed on the communication subnet by a broadcast, and the delays in achieving broadcast. The various broadcast routing algorithms are quantitatively compared by determining lower bounds on their performance in regular graphs. The lower bounds determine the performance of these algorithms in "ideal" networks, against which network designers may measure their networks.

In Chapter 5 we discuss the structure of a distributed file system. We review current work in this effort, and propose a model to structure the catalog of the file system, and search algorithms for locating the files. We show how the performance of the system improves if there is an underlying broadcast communication capability. By using such a capability, it is not necessary to have multiple copies of the catalog, which must be kept consistent.

Chapter 6 wraps up this thesis by stating the results and conclusions of our research. Open problems stemming from our work are stated, since they provide important areas for future research. A set of Appendices enhance the description in various chapters. There is one Appendix per chapter - the first one provides a detailed table of contents to this thesis.

CHAPTER 2

DISTRIBUTED ALGORITHMS FOR CONSTRUCTING MINIMAL SPANNING TREES

2.1 Introduction

In this chapter, we present two distributed algorithms for constructing minimal spanning trees in computer communication networks. The algorithms can be executed concurrently and asynchronously by the different computers of a network. There is no one source of control. The algorithms are also suitable for constructing minimal spanning trees using a multiprocessor computer system. The first algorithm is static, in that it assumes certain initial conditions and constructs the minimal spanning tree. The second algorithm is adaptive and executes continuously. It dynamically converts an old (minimal) spanning tree into a new minimal spanning tree when edge costs change, nodes are removed from the network, or new nodes are added to the network. Some applications in computer communication networks that require constructing minimal spanning trees are now described.

The minimal spanning tree can be used for forwarding broadcast messages in distributed operating systems built on top of a packet switched network. If a minimal spanning tree was embedded on the existing subnet topology, then any node on this minimal spanning tree could initiate a broadcast, and the packets would be forwarded along this tree to all destinations. Each node knows which of the edges incident to it are branches of the minimal spanning tree. Hence, a "broadcast" packet arriving on one such branch would be delivered to all hosts connected to the node, and forwarded along the remaining branches. This technique assumes, of course, that the cost of communication along an edge of the network is same in both directions. This is not true, in general, for PSNs, since the traffic patterns determine the queueing delays in either direction of a link. It is, however, not a bad approximation. Figure 2.1 shows the communication subnet of a PSN with the embedded minimal spanning tree. If broadcast was initiated from a host connected to node 1, then a packet would be transmitted along each of the minimal spanning tree branches in the directions shown in the figure. Note that all the edges in the subnet do not have the same cost. We examine, in detail, the suitability of using the minimal spanning tree for broadcast routing in Chapters 3 and 4.

Broadcast routing is also used by the minimal spanning tree algorithms themselves. When the minimal spanning tree has been constructed, all nodes must be informed and the simplest way is to broadcast the message along the branches of the tree. In the construction process itself, a message may have to be sent from one node to another in the same subtree as the originator. We shall see that we can use the subtree for routing, by broadcasting the message rather than relying on another underlying routing mechanism.

Minimal spanning tree routing also appears to have application in the design of adaptive routing algorithms, since the branches of the minimal spanning tree could be used to broadcast delay estimates to all nodes, rather than using a hop by hop refinement technique

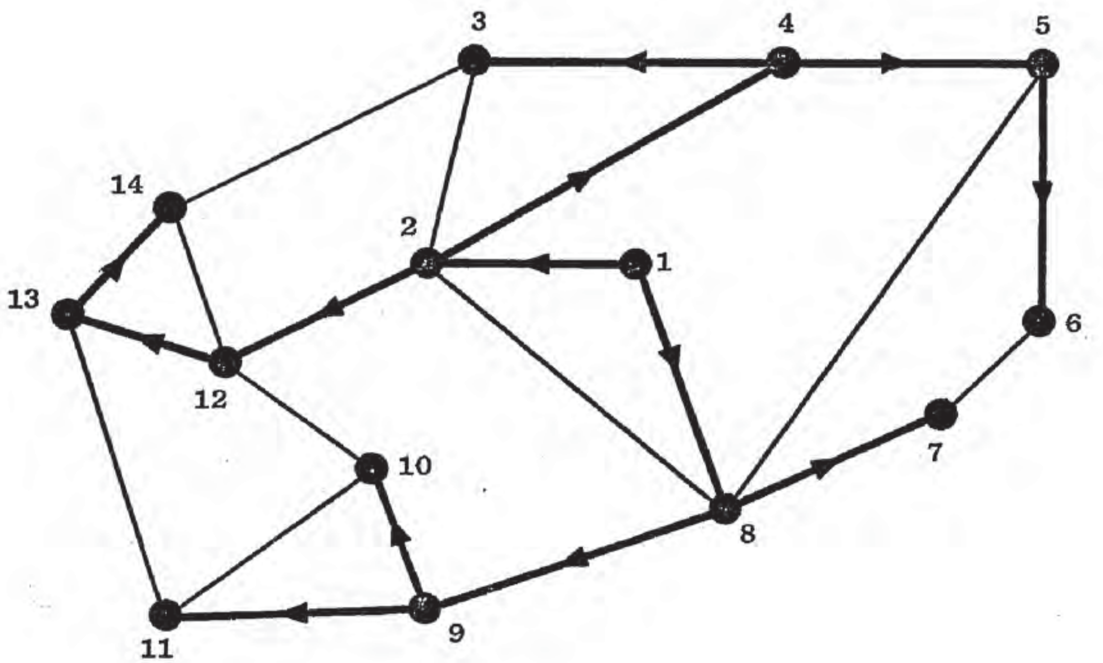

Figure 2.1. BROADCAST ALONG THE BRANCHES OF THE MST INITIATED FROM NODE 1.

The adaptive algorithm has special application in communication networks like the Packet Radio Network (PRNET) [Kahn75, Frank75]. The packet radio repeaters must configure themselves into a suitable topology when randomly placed in an operating environment, for example when dropped out of an airplane. Currently a tree-like topology is set up from a centralized point of control. If a minimal spanning tree is acceptable, then this algorithm could be used.

Section 2.2 discusses construction principles for minimal spanning trees, and section 2.3 describes an abstract parallel algorithm by which such trees can be constructed in a distributed environment. Sections 2.4 and 2.5 discuss the static algorithm, and 2.6 the adaptive algorithm. These algorithms are based on the abstract parallel algorithm.

DOCKET ALARM

Explore Litigation Insights

Docket Alarm provides insights to develop a more informed litigation strategy and the peace of mind of knowing you're on top of things.

Real-Time Litigation Alerts

Keep your litigation team up-to-date with **real-time alerts** and advanced team management tools built for the enterprise, all while greatly reducing PACER spend.

Our comprehensive service means we can handle Federal, State, and Administrative courts across the country.

Advanced Docket Research

With over 230 million records, Docket Alarm's cloud-native docket research platform finds what other services can't. Coverage includes Federal, State, plus PTAB, TTAB, ITC and NLRB decisions, all in one place.

Identify arguments that have been successful in the past with full text, pinpoint searching. Link to case law cited within any court document via Fastcase.

Analytics At Your Fingertips

Learn what happened the last time a particular judge, opposing counsel or company faced cases similar to yours.

Advanced out-of-the-box PTAB and TTAB analytics are always at your fingertips.

API

Docket Alarm offers a powerful API (application programming interface) to developers that want to integrate case filings into their apps.

LAW FIRMS

Build custom dashboards for your attorneys and clients with live data direct from the court.

Automate many repetitive legal tasks like conflict checks, document management, and marketing.

FINANCIAL INSTITUTIONS

Litigation and bankruptcy checks for companies and debtors.

E-DISCOVERY AND LEGAL VENDORS

Sync your system to PACER to automate legal marketing.