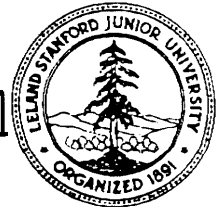


COMPUTER SYSTEMS LABORATORY

STANFORD ELECTRONICS LABORATORIES
DEPARTMENT OF ELECTRICAL ENGINEERING
STANFORD UNIVERSITY - STANFORD, CA 94305



CENTER-BASED BROADCASTING

David W. Wall
Susan S. Owicki

TECHNICAL REPORT NO. 189

June 1980

This research was supported by the Joint Services Electronics Project
under contract N-00014-75-C-0601



CENTER-BASED BROADCASTING

David W. Wall

Susan S. Owicki

TECHNICAL REPORT NO. 189

June 1980

COMPUTER SYSTEMS LABORATORY

Departments of Electrical Engineering and Computer Science

Stanford University

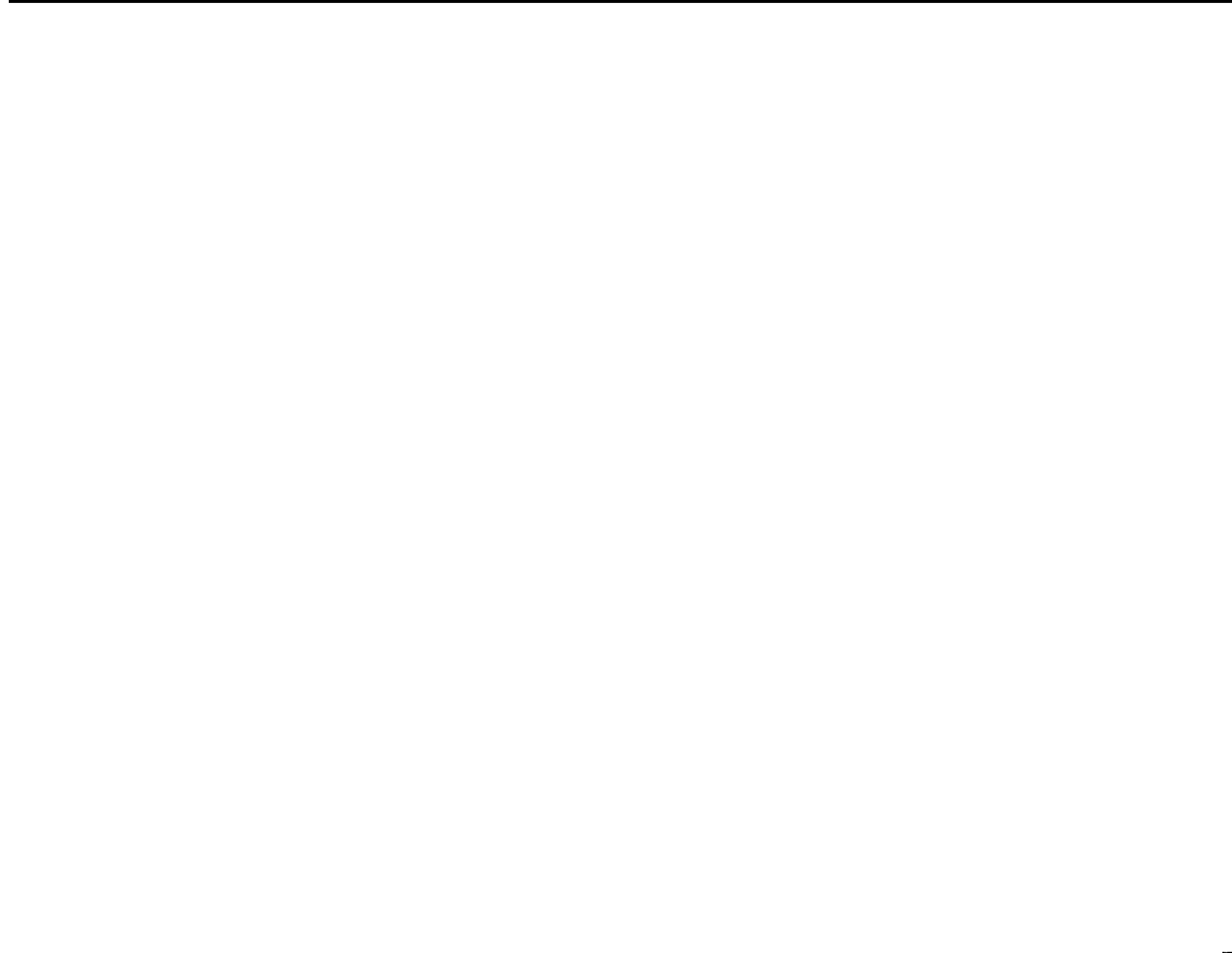
Stanford, California 94305

ABSTRACT

We consider the problem of routing broadcast **messages** in a loosely-coupled store-and-forward network like the ARPANET. Dalal [2] discussed a solution to this problem that minimizes the cost of a broadcast; in contrast, we are interested in performing broadcast with small delay. Existing algorithms can minimize the delay but seem unsuitable for use in a distributed environment because they involve a high degree of overhead in the form of redundant messages or data-structure space. We propose the schemes of center-based forwarding: the routing of all broadcasts via the shortest-path tree for some selected node called the center. These algorithms have **small delay and also are easy to implement in a distributed system.**

To evaluate center-based forwarding, we **define** four measures of the delay associated with a given broadcast mechanism, and then propose three ways of selecting a center node. For each of the three forms of center-based forwarding we compare the delay to the minimum delay for any broadcasting scheme and also to the minimum delay for any single tree. In most cases, a given measure of the delay on the centered tree is bounded by a small constant **factor** relative to either of these two minimum delays. When it is possible, we give a tight bound on the ratio between the center-based delay and the minimum delay; otherwise we demonstrate that no bound is possible. These results give corollary bounds on how bad the three centered trees can be with respect to each other; most of these bounds are immediately tight, and the rest are replaced by better bounds that are also shown to be, tight.

KEYWORDS: Distributed networks, Distributed **algorithms**, **Message routing**, **Broadcast**, Spanning trees, ARPANET



1. Introduction.

In a loosely-coupled store-and-forward network like the **ARPANET**, a **message** is routed from one node to another along some series of links starting at the source and ending at the destination. The problem of selecting the best route for a given message has been considered in detail, and a simple but effective mechanism is provided by the **ARPANET** [6,7]. Much of the recent work on the problems of effectively using such a network ([1,3,5,8]) has assumed the existence of an additional mechanism that provides message broadcast-sending an identical message to every node in the network. No explicit facility for message broadcast is provided by the **ARPANET**.

Two important criteria for evaluating a routing algorithm are **cost and delay**. Dalal [2] considers several methods of message broadcast but concentrates on a scheme for minimizing the cost. He models the network as a weighted graph in which the weight on an edge represents the cost to the network of sending a message over that link; then a broadcast can be done at minimum cost by routing it over the branches of a minimum spanning tree. Dalal describes a distributed algorithm by means of which a network can construct its own minimum spanning tree. The result is that each node records a "**Local Image**" that specifies which of the links incident to that node are branches of the tree. A node can then initiate a broadcast by sending one copy along each incident branch; a node forwards a broadcast by sending a copy along each incident branch except the one on which the broadcast arrived.

In contrast to Dalal, we are interested in mechanisms that provide broadcast with small delay. We define two measures for the delay associated with a given broadcast. The first is the maximum delay D , the time required for the broadcast to reach the most distant destination; the second is the average delay A , the expected time required to reach a destination. We want the algorithms not only to provide good values for these measures, but also to be suitable for use in a dynamic network on which the **delays** across a link may change. Such an environment requires that the routes used be repeatedly updated in order to reflect the changing delays, so we want a broadcast mechanism for which this does not require a lot of computation.

One approach to the problem of broadcasting with low delay is simply to send an individual message to each destination via the shortest path. This method of singly-addressed messages has the advantage of conceptual simplicity, especially because a mechanism for sending to a single destination may be assumed already to be present in a network. However, it has the disadvantage that it results in redundant transmissions: several copies of the same message may be sent over the same link. This problem can be removed by changing to *multiply-addressed* messages, in which each copy of the message contains a list of destinations; the extra information in the address header allows all the copies sent across a single

□ *Center-Based Broadcasting*

link to be coalesced into a single copy, but it involves the overhead of a complicated message header that must be processed for each and every broadcast.

Another approach is source-based *forwarding*. We can model the network as a weighted graph whose weights represent not costs but delays across the links. Then for any vertex r we can define a **shortest-path** tree T_r as a spanning tree such that the path via T_r from r to a vertex v is a shortest path from r to v in the network as a whole. We will call r the roof of its shortest-path tree T_r . In source-based forwarding, each broadcast is sent via a shortest-path tree for the source of that broadcast; as a result any broadcast will reach each destination in the smallest possible time, thus minimizing both D and A . Unfortunately, the shortest-path trees for two different vertices are unlikely to be the same. To give each vertex a Local Image of a shortest-path tree for each vertex in the network is a large investment of time and data structure space; moreover it is hard to change this structure in the face of changing network conditions. Nonetheless, source-based forwarding is worth considering because it provides an optimum algorithm with which other algorithms can be compared.

An alternative approach is to select a single tree with good delay properties and use it to forward a broadcast regardless of which vertex initiated the broadcast, just as the minimum spanning tree is used in Dalal's approach. This gives a simple, uniform mechanism that is easy to maintain in a distributed environment. Unlike source-based forwarding, it may not minimize the delay for all broadcasts; however, we can still choose a tree that gives a fairly small delay.

In this report we discuss three ways of selecting a single tree with good delay properties. Each of our approaches involves using the shortest-path tree for a vertex that is in some sense "in the center" of the network; we will call this class of approaches **center-based forwarding**. For each form of centered tree, we will show that the delay is tightly bounded by a small factor relative to the delay for the best possible single tree and relative to the minimum delay, which is that attained by source-based forwarding. We also show tight bounds on the delay of each form of centered tree compared to that of the other forms.

We begin by **defining** some notation and terminology in section 2. Section 3 describes the three types of centered trees and briefly discusses how each might be constructed. Section 4 establishes some essential lemmas, and section 5 gives some simple bounds on the delay for a given broadcast over a given type of centered tree. The main results deal with the delay measures that apply to broadcast techniques rather than simply to individual broadcasts. These results begin in section 6, where we give bounds on these measures applied to centered trees, relative to the best possible broadcast mechanism and relative to the best single spanning tree. In section 7 we give bounds for the three kinds of centered tree relative to each other) and in section 8 we give examples that demonstrate the tightness of some of the

bounds in the previous sections. Section 9 discusses in more detail the problems of constructing the centered trees in a distributed environment, and section 10 presents some conclusions we have drawn from this research.

2. Notation and Terminology.

Let $G = (V, E, t)$ be a connected weighted undirected graph consisting of a set V containing n vertices, a set $E \subseteq V \times V$ of edges, and a positive weight $t(v, w) = t(w, v)$ associated with each edge $\{v, w\}$. This weight function will represent the time associated with sending a message across a given link. Let $d(v, w) = d(w, v)$ be the delay between v and w in the graph; namely the sum of the times on the shortest path between v and w . Similarly, for any tree T in G , let $d(T, v, w) = d(T, w, v)$ be the delay between v and w via the tree T ; namely the sum of the times on the unique path between them in T .

Define the **diameter** of a tree to be the longest acyclic path in the tree, and denote it by $\text{diam}(T)$. Thus

$$\text{diam}(T) = \max_{\substack{v \in V \\ w \in V}} d(T, v, w).$$

Let T_s be a shortest-path tree for s . Note that on this shortest path tree, the delay from s is $d(T_s, s, v) = d(s, v)$; that is, the path from s to some other node via s 's tree is in fact a shortest path in the graph as a whole.

Given a source node s and a tree T we can **define** two measures of the delay associated with a broadcast from s via T . The first measure is the maximum delay

$$D(s, T) = \max_{v \in V} d(T, s, v)$$

which is the time **necessary** to get the broadcast to the most distant destination. The second measure is the average delay

$$A(s, T) = \frac{1}{n} \sum_{v \in V} d(T, s, v)$$

which is the expected time to get the broadcast to a destination.

The functions D and A are measures of the delay for a specific broadcast. To compare one tree to another tree or to source-based forwarding, we need measures that depend only on the structure of the broadcast mechanism, and not on the broadcast source. Four such general delay measures are simply the worst-case and **average-case** behaviors of D and A . Define the maximum D for a tree to be

$$\text{MaxD}(T) = \max_{s \in V} D(s, T)$$

□ *Center-Based Broadcasting*

and the maximum A to be

$$\mathbf{MaxA}(T) = \max_{s \in V} A(s, T).$$

Thus the \mathbf{MaxD} (or \mathbf{MaxA}) for a tree is the worst value of D (or A) over all possible broadcast sources. Similarly, define the average D and A to be

$$\mathbf{AveD}(T) = \frac{1}{n} \sum_{s \in V} D(s, T)$$

and

$$\mathbf{AveA}(T) = \frac{1}{n} \sum_{s \in V} A(s, T).$$

Since A and D are well-defined for any deterministic forwarding scheme, we can define these measures for source-based forwarding as **well**. We will define

$$\begin{aligned} \mathbf{MaxD}(\text{SBF}) &= \max_{s \in V} D(s, T_s), \\ \mathbf{MaxA}(\text{SBF}) &= \max_{s \in V} A(s, T_s), \\ \mathbf{AveD}(\text{SBF}) &= \frac{1}{n} \sum_{s \in V} D(s, T_s), \end{aligned}$$

and

$$\mathbf{AveA}(\text{SBF}) = \frac{1}{n} \sum_{s \in V} A(s, T_s).$$

Thus in source-based forwarding we are considering the maxima and averages over all possible broadcast sources, for a broadcast done via the source's own shortest-path tree.

3. Trees Considered.

We want our method of selecting a center to be one that works well in a distributed environment. This leads us to impose two locality *restrictions*. First, we want to be able to make the selection with relatively little interaction among the vertices. Second, each vertex should be able to do its share with relatively little information, all of it local: for instance, with only the information in the local routing tables for the underlying single-message mechanism. Ideally, each vertex will do a small amount of local computation that produces a numerical value for some criterion that tells how good a center that vertex would be; the vertices then pool these values by means of a simple protocol that compares these criterion values and decides which vertex would be best.

In contrast, constructing the tree that minimizes any of the four general delay measures in a distributed environment is more difficult. The problem of constructing the tree with minimum **AveA**, in fact, is NP-complete [4], as is the construction of the tree with minimum **MaxA** [9].

Two of the most obvious candidates for center are the vertex dc such that $D(dc, T_{dc})$ is smallest, and the vertex uc such that $A(uc, T_{uc})$ is smallest. The measures D and A are easy to compute locally using the underlying single-message tables, and later we will discuss a simple protocol to select the vertex with the smallest value and construct its shortest-path tree. We will call these nodes the **D-center** and the **A-center** respectively, and their shortest-path trees the **D-centered tree** and the **A-centered tree**.

The third candidate for center is the node **diamc** which has the shortest-path tree of least diameter. We will see later that the diameter of a vertex's shortest path tree need not pass through that vertex, and so the vertex might not know the diameter of its own shortest-path tree. As a result, it is not obvious that we can compute a criterion that selects such a **diamc** using only the underlying local tables. Nevertheless we will see that the proper center can be selected in a manner **that** is consistent with our locality requirements. We will call this center the **diameter-center** and its shortest-path tree the **diameter-centered tree**, denoted by T_{diamc} .

In the following sections we compare the delay associated with broadcasts on these three forms of centered tree with that of the optimal algorithm of source-based forwarding, and with that of any single tree. In addition we compare the delay for each form of centered tree with that of the other centered trees.

4. Useful Lemmas.

There are a number of general observations about trees that will be useful in deriving the bounds in this paper. First of all, we note that shortest paths satisfy the triangle inequality.

Triangle Inequality. For any vertices v , w , and x , the delays between them in the graph satisfy

$$d(v, x) \leq d(v, w) + d(w, x)$$

and the delays between them via any tree T satisfy

$$d(T, v, x) \leq d(T, v, w) + d(T, w, x).$$

Proof. The delay is defined to be the time of the shortest available path between the two points; in the case of a tree this is the only acyclic path. If the triangle inequality did not hold, the path from v to x via w would be shorter than the shortest path, which is impossible. ■

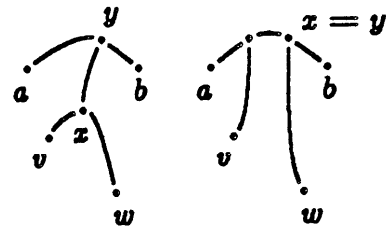
□ *Center-Based Broadcasting*

When we are considering the maximum delay D associated with a broadcast from a particular vertex it is useful if we can pin down some specific vertex that is farthest from that vertex. The following lemma gives us a better grip on that problem.

Diameter Lemma. Given a tree T , a diameter of T whose endpoints are a and b , and a vertex v , then either the vertex a or the vertex b is as far from v via T as any vertex in T .

Proof. Let w be a vertex maximally far from v via the tree T and consider the unique path in T between them. This path may or may not cross the diameter,

If the path does not intersect the diameter, then there is some single point on this path that is closest via T to the diameter; let x be this vertex, and let y be the vertex on the diameter to which it is closest. These two vertices are unique **because** we are dealing with a tree. If the path does cross the diameter, the two may share several edges before separating again, but once they separate they will not intersect again since T is a tree. In this case let x and y both refer to the last vertex on the path from v to w that is also a vertex on the diameter. Assign the names a and b to the endpoints of the diameter in such a manner that the path from v to y intersects the path from y to b only at the point y .



Path and Diameter

Since the path from a to b is a diameter, we know that b is as far from a as any other vertex, and in particular that $d(a, b) \geq d(a, w)$. Subtracting $d(a, y)$ from both, we see that

$$d(y, b) \geq d(y, w). \tag{1}$$

Similarly, since w is maximally far from v , it is at least as far as b is, so $d(v, w) \geq d(v, b)$ and hence subtracting $d(v, x)$ from each gives us

$$d(x, w) \geq d(x, b).$$

But we can add $d(x, y)$ to each side to get

$$d(y, w) \geq d(x, y) + d(x, b) = 2d(x, y) + d(y, b)$$

and hence

$$d(y, w) \geq d(y, b). \tag{2}$$

Relations (1) and (2) tell us that these two delays are in fact equal:

$$d(y, w) = d(y, b). \quad (3)$$

This also shows that the path and the diameter must **always** intersect, since (2) is an equality only if $d(x, y)$ is **zero**.

But now we simply work back to the vertex v . We note first that

$$\begin{aligned} d(x, b) &= d(x, y) + d(y, b) \\ &= d(x, y) + d(y, w) \\ &= 2d(x, y) + d(x, w) \end{aligned}$$

and so

$$d(x, b) \geq d(x, w).$$

Adding $d(v, x)$ to each gives us

$$d(v, b) \geq d(v, w), \quad (4)$$

but since w is maximally far from v this means that b is just as far, which completes the proof. ■

This result tells us that if we are considering the values of $D(s, T)$ for various vertices s on some spanning tree T , it suffices to select a diameter whose endpoints we will call a and b , and consider only $d(T, s, a)$ and $d(T, s, b)$, since one of these two will be the maximum.

The next lemma tells us that two vertices that are far apart cannot both have small average delays.

Sum of Averages Lemma. Given a graph G and two vertices v and w , the delay $d(v, w)$ between v and w is no bigger than the sum $A(v, T_v) + A(w, T_w)$ of the average delays for broadcasts done from v and w via their own shortest-path trees.

Proof. The triangle inequality tells us that for any vertex z ,

$$d(v, w) \leq d(v, z) + d(w, z).$$

If we average this over all z , we see that

$$d(v, w) \leq A(v, T_v) + A(w, T_w),$$

which is the desired result. ■

□ Center-Based Broadcasting

Corollary. Given a tree T and two vertices v and w , the delay $d(T, v, w)$ between the vertices via T is no bigger than the sum $A(v, T) + A(w, T)$ of the average delays for **broadcasts** done from v and w via T .

Proof. This is a special case of the Sum of **Averages Lemma**, in which the graph is simply a tree T and hence the **averages via** the shortest-path trees are exactly the averages via T itself. ■

The final lemma has a rather specific purpose, but may be of general interest nonetheless.

A-Plus-D Lemma. For any tree T , the measure $\text{AveD}(T)$ is no more than the sum $A(s, T) + D(s, T)$ of the **A** and **D** for a broadcast from any node s via T .

Proof. For any vertices v and w , the triangle inequality tells us that

$$d(T, v, w) \leq d(T, v, s) + d(T, s, w).$$

If we maximize this over w we see that

$$D(v, T) \leq d(T, v, s) + D(s, T)$$

and **then averaging** over v gives us

$$\text{AveD}(T) \leq A(s, T) + D(s, T)$$

which is what we claimed. ■

Informally, this lemma is true because at worst we could do any broadcast by sending it to s via T and letting s initiate the **broadcast**. In that case the delay would be the time it takes to get the message to s , which on the average is $A(s, T)$, plus the delay for a broadcast from s via T .

5. Delay for Single Broadcasts.

We will first consider some results about single broadcasts on centered trees, **If** a broadcast is being performed from a given source vertex s , we incur the minimum delay by routing the broadcast via the shortest-path tree T_s for a. **How bad can** the delay be via a centered tree compared to the minimum?

Since we have two delay measures **A** and **D**, and we have discussed three different kinds of centered tree, we can produce (at least) six different **answers** to **that** question. Here is one answer, dealing with the maximum delay **D** for a broadcast via the diameter-centered tree.

5. Delay for Single Broadcasts \square

Theorem. The maximum delay $D(s, T_{diamc})$ for a broadcast from s routed via the diameter-centered tree is at worst twice the minimum value $D(s, T_s)$; furthermore, there exist networks for which this bound is attained.

Proof. For any vertex v , the definition of a tree's diameter means that

$$d(T_{diamc}, s, v) \leq \text{diam}(T_{diamc}).$$

Hence

$$d(T_{diamc}, s, v) \leq \text{diam}(T_s)$$

since the diameter-centered tree has as small a diameter as any shortest-path tree. If a and b are endpoints of a diameter of T_s , this means that

$$\begin{aligned} d(T_{diamc}, s, v) &\leq d(T_s, a, b) \\ &\leq d(T_s, s, a) + d(T_s, s, b) \end{aligned}$$

by the triangle inequality. Each of these latter two delays is no more than $D(s, T_s)$, so we see that

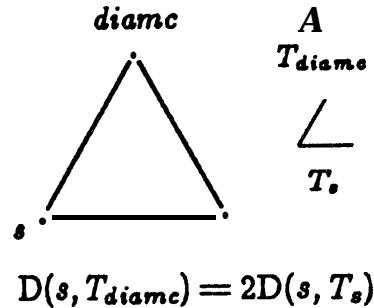
$$d(T_{diamc}, s, v) \leq 2D(s, T_s).$$

Maximizing this over v gives us

$$D(s, T_{diamc}) \leq 2D(s, T_s),$$

the desired bound.

This bound can be attained by any complete graph of three or more vertices, with edge weights that are all unity. Any shortest-path tree then gives a D of 2 for any vertex s other than the root, but a D of 1 is possible by using s 's own shortest path tree. Thus the bound we proved above is tight.



This completes the proof. ■

Similar results for the remaining combinations of the three kinds of tree and the two delay measures can be proven with equal ease. These bounds are summarized in Table 1, which for each form of centered tree gives the bound on the ratio of each single-broadcast delay measure compared to its minimum value for a broadcast from that source; this minimum is attained by routing the broadcast via the shortest-path

□ **Center-Based Broadcasting**

tree for the source. All the bounds in this table are tight, though some can only be attained by defining a class of graphs and considering the limit as the number of vertices increases. A bound of infinity indicates that the measure can be arbitrarily bad with respect to the minimum value; again, examples exist that demonstrate this fact. The last column of this table contains bounds that apply to any shortest-path tree; the fact that these bounds are somewhat worse than those for the specific trees shows that we do gain something by going to the effort of selecting a center.

$T =$	T_{dc}	T_{ac}	T_{diame}	any T_r
$D(s, T)$	2	3	2	3
$A(s, T)$	∞	3	∞	∞

Table 1. Single-broadcast delay measures. Bounds on the delay for certain shortest-path trees compared to the minimum delay.

Proofs of the validity and tightness of the remaining bounds in Table 1 can be found in Wall's dissertation [9].

6. Delay for Broadcast Mechanisms.

In a sense the situation is actually better than the previous results would lead us to believe: those results compare the delay for a given source node to the best delay for any broadcast from that source, and in general there is no single tree that can perform that well for every source. We will see in the results to come **that even** source-based forwarding cannot be that much better for every source.

We are concerned here with the four measures **MaxD**, **MaxA**, **AveD**, and **AveA**, and with how bad they can be for each type of centered tree when compared with the minimum technique of source-based forwarding and when compared with the best possible single tree. In addition we can prove certain bounds on these measures for any shortest-path tree. These results are **summarized** in Tables 2 and 3. Again, all these bounds are tight, including the cases where we have no **bound**; as before, however, some can only be attained in the limit as the number of vertices increases.

	T_{dc}	T_{ac}	T_{diame}	any T_r
MaxD	2	2	2	2
MaxA	3	3	3	3
AveD	2	3	2	3
AveA	∞	2	∞	∞

Table 2. General delay measures. Bounds on the delay of certain shortest-path trees compared to source-based forwarding.

	T_{dc}	T_{ac}	T_{diamc}	any T_r
MaxD	2	2	2	2
MaxA	3	3	3	3
AveD	2	$\frac{5}{3}$	2	3
AveA	∞	2	∞	∞

Table 3. General delay measures. Bounds on the delay of certain shortest-path trees compared to any single tree.

These bounds are better than Table 1 might have led us to expect. For instance, the value of **MaxA** for a centered tree is at worst three times the minimum value, even though the A for any single broadcast can be arbitrarily worse than the minimum for that source. This indicates that if there is a source for which the centered tree gives a large A, then there is also a source (not necessarily the same one) for which even source-based forwarding must give a large A.

In proving the validity and tightness of the bounds in Tables 2 and 3, the following observations are useful. With one exception, the bounds in Table 3 are immediate corollaries of the bounds in Table 2; that is, if a given measure for a given tree can be no worse than (say) three times its value for source-based forwarding, it can surely be no worse than three times its value for any single tree, since source based forwarding minimizes the D and A for each source independently and is therefore at least as good as any single tree. We will therefore prove the validity of the bounds in Table 2 first. On the other hand (with the same exception), an example that shows the tightness of a bound in Table 3 also demonstrates its tightness in Table 2. We will postpone a discussion of the examples that attain these bounds until section 8.

We begin with a result that gives us the entire first row of Table 2.

Theorem. The **MaxD**(T_r) for any shortest-path tree T , is at most twice that of **source-based** forwarding.

Proof. For any vertices v and w , the triangle inequality tells us that

$$d(T_r, v, w) \leq d(T_r, r, v) + d(T_r, r, w).$$

If we maximize this over w , we see that

$$D(v, T_r) \leq d(T_r, r, v) + D(r, T_r)$$

and maximizing over v gives us that

$$\text{MaxD}(T_r) \leq D(r, T_r) + D(r, T_r) = 2D(r, T_r).$$

□ Center-Based Broadcasting

Since $\text{MaxD}(\text{SBF})$ is the largest delay for a source via its own shortest-path tree, it follows that $D(\mathbf{r}, T) \leq \text{MaxD}(\text{SBF})$ and therefore that

$$\text{MaxD}(T_r) \leq 2\text{MaxD}(\text{SBF}),$$

which is the desired result. ■

A slightly more complicated proof gives us the second row of Table 3.

Theorem. The $\text{MaxA}(T_r)$ for any shortest-path tree T , is at most three times **that** of source-based forwarding.

Proof. For any vertices \mathbf{v} and w , the triangle inequality tells us that

$$d(T_r, \mathbf{v}, w) \leq d(T_r, \mathbf{r}, \mathbf{v}) + d(T_r, \mathbf{r}, w).$$

If we average over w this gives

$$A(\mathbf{v}, T_r) \leq d(T_r, \mathbf{r}, \mathbf{v}) + A(\mathbf{r}, T_r)$$

and if we then maximize over \mathbf{v} we see that

$$\text{MaxA}(T_r) \leq D(\mathbf{r}, T_r) + A(\mathbf{r}, T_r). \quad (1)$$

This says in effect that we can always do a broadcast over T , by sending it to \mathbf{r} and, letting \mathbf{r} initiate the broadcast, in which case the average would be the delay to get it to \mathbf{r} , which is at most $D(\mathbf{r}, T)$, plus the average for a broadcast from \mathbf{r} .

We will consider two cases. First, suppose that $A(\mathbf{r}, T) \geq \frac{1}{2}D(\mathbf{r}, T)$. Then relation (1) tells us that

$$\begin{aligned} \text{MaxA}(T_r) &\leq 3A(\mathbf{r}, T_r) \\ &\leq 3 \max_{\mathbf{v}} A(\mathbf{v}, T_r) \\ &= 3\text{MaxA}(\text{SBF}), \end{aligned} \quad (2)$$

which is the bound desired.

On the other hand, suppose that $A(\mathbf{r}, T) \leq \frac{1}{2}D(\mathbf{r}, T)$. Then

$$\frac{1}{2}D(\mathbf{r}, T_r) \leq D(\mathbf{r}, T_r) - A(\mathbf{r}, T).$$

If we let \mathbf{z} be some vertex farthest from \mathbf{r} , i.e. some vertex at a distance of $D(\mathbf{r}, T)$, then this is the same as

$$\frac{1}{2}D(\mathbf{r}, T_r) \leq d(\mathbf{z}, \mathbf{r}) - A(\mathbf{r}, T_r)$$

and hence

$$\frac{1}{2}D(r, T_r) \leq A(z, T_z)$$

by the Sum of **Averages** Lemma. But (1) tells us that

$$\text{MaxA}(T_r) \leq \frac{3}{2}D(r, T_r)$$

since $A(r, T_r)$ is less than half of $D(r, T_r)$. Combining these two relations we see that

$$\begin{aligned} \text{MaxA}(T_r) &\leq 3A(z, T_z) \\ &\leq 3\text{MaxA}(\text{SBF}) \end{aligned} \tag{3}$$

since as before $\text{MaxA}(\text{SBF})$ must be at least as big as any individual $A(v, T_v)$. Thus the bound holds regardless of the relative values of $A(r, T_r)$ and $D(r, T_r)$. ■

The third row of Table 2 is not uniform; the value of the bound depends on the tree to which it applies. Nonetheless there is a simple relationship between this row and the first row of Table 1, which is expressed in the following result.

Theorem. Given a tree T and a bound k such that $D(s, T) \leq k D(s, T_s)$ for every source vertex s , then the same bound k holds in $\text{AveD}(T) \leq k \text{AveD}(\text{SBF})$.

Proof. By assumption

$$D(s, T) \leq k D(s, T_s).$$

Averaging this relation over s gives

$$\text{AveD}(T) \leq k \text{AveD}(\text{SBF})$$

which is the desired bound. ■

Thus when Table 1 gives a bound on $D(s, T)$ that applies to **all** trees T in a given class, this theorem allows us to place the same bound on $\text{AveD}(T)$ for **all** trees T in that **class**.

The fourth row of Table 2 contains only one finite bound, given by the following result.

Theorem. The $\text{AveA}(T_{ac})$ for the A-centered tree is at most twice that of **source-based** forwarding.

Proof. The triangle inequality tells us that for any vertices v and w ,

$$d(T_{ac}, v, w) \leq d(T_{ac}, ac, v) + d(T_{ac}, ac, w).$$

□ **Center-Based Broadcasting**

Averaging this over w gives

$$A(v, T_{ac}) \leq d(T_{ac}, ac, v) + A(ac, T_{ac})$$

and then averaging over v gives

$$\text{Ave}A(T_{ac}) \leq A(ac, T_{ac}) + A(ac, T_{ac}) = 2A(ac, T_{ac}).$$

The definition of the A-center tells us that for any vertex r , $A(ac, T_{ac}) \leq A(r, T_r)$ and hence averaging over r , that $A(ac, T_{ac}) \leq \text{Ave}A(\text{SBF})$. Thus

$$\text{Ave}A(T_{ac}) \leq 2\text{Ave}A(\text{SBF}).$$

This completes the proof. ■

We have shown that **all** the bounds in Table 2 are valid. This means that they would also be valid bounds to use in the corresponding places in Table 3, since source-based forwarding is **at least** as good as **any** single tree. However, in one case we can improve the bound for Table 3, as seen in the following interesting theorem.

Theorem. The $\text{Ave}D(T_{ac})$ of the A-centered tree is at most $\frac{5}{3}$ times that of any single tree T .

Proof. Choose some diameter of T and let its endpoint vertices be a and b . Consider a point p at the midpoint of this diameter. There need not be a vertex here. Nonetheless, this midpoint lies on some edge and we could imagine adding **an** $(n + 1)^{\text{st}}$ vertex at that location; the delays both in the graph and in the tree T between this imaginary vertex and the other vertices **would** then be **well-defined**. Thus it makes sense to define the quantity

$$\alpha = \frac{1}{n} \sum_v d(T, p, v)$$

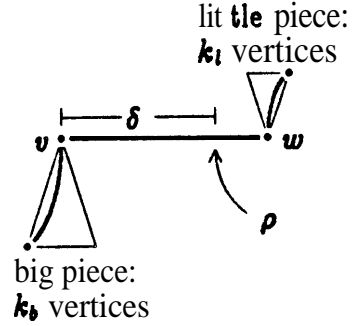
which is the **average** delay from p to the vertices via the tree T .

This quantity α looks rather like $A(v, T)$ for some vertex v ; in fact if there is a vertex at p then α is precisely the A for that vertex. So it comes as no surprise that α is at least as big as the minimum average over the vertices, namely $A(ac, T_{ac})$, which we show as follows.

If p coincides with a vertex v then the proof is trivial since α is exactly the **average** $A(v, T)$ for that vertex, which is **at least** as big as the **average** $A(v, T_v)$ for v over its own shortest-path tree, which in turn is at least as big as the minimum such value $A(ac, T_{ac})$.

6. Delay for Broadcast Mechanisms \square

Assume, therefore, that ρ lies on some edge rather than on a vertex. That edge divides the tree T into two pieces which we will call the big and the little pieces such that the big piece has at least as many vertices as does the little piece. Specifically, suppose that there are k_b vertices in the big piece and $k_l \leq k_b$ vertices in the little piece. Let $\{v, w\}$ be the edge on which ρ lies, with v in the big piece of the tree, and let δ be the delay from ρ to v ; that is the delay over the portion of ρ 's edge running between ρ and v . Then



What ρ 's Edge Does

$$\begin{aligned} \alpha &= \frac{1}{n} \sum_x d(T, \rho, x) \\ &= \frac{1}{n} ((k_b - k_l)\delta + \sum_x d(T, v, x)) \\ &= \frac{1}{n} (k_b - k_l)\delta + A(v, T) \\ &\geq A(v, T) \end{aligned}$$

since $k_b \geq k_l$. But this leads us back to our previous reasoning: $A(v, T) \geq A(v, T_v) \geq A(ac, T_{ac})$. Thus

$$\alpha \geq A(ac, T_{ac}). \tag{1}$$

Now the reason we chose to consider this midpoint in the first place is that we can use it to give an exact expression for the AveD of T . The Diameter Lemma tells us that for any vertex v , one or the other of the endpoints a and b of the diameter we chose is as far from v via T as it is possible to get. This means in particular that the delay $D(v, T)$ for a vertex v in one piece of the tree is exactly equal to the delay from v to the endpoint a or b in the other piece. As a result, the path associated with this delay must pass through ρ . In other words, for any v ,

$$D(v, T) = d(T, v, \rho) + \frac{1}{2} \text{diam}(T)$$

and if we average this over v we see that

$$\text{AveD}(T) = a + \frac{1}{2} \text{diam}(T). \tag{2}$$

We want to know how bad the AveD can be for an A-centered tree T_{ac} embedded in the same graph. To accomplish this, we will derive a pair of upper bounds for $\text{AveD}(T_{ac})$ in terms of a and $\text{diam}(T)$.

□ *Center-Based Broadcasting*

First of all, the A-Plus-D Lemma means that

$$\mathbf{AveD}(T_{ac}) \leq \mathbf{A}(ac, T_{ac}) + \mathbf{D}(ac, T_{ac})$$

and hence that

$$\mathbf{AveD}(T_{ac}) \leq a + \mathbf{D}(ac, T_{ac})$$

by (1) **above**. But $\mathbf{D}(ac, T_{ac}) \leq \mathbf{diam}(T)$, since T contains a path from ac to any other point, of length at most $\mathbf{diam}(T)$, and each path in UC's own shortest-path tree can be no longer (or it would not be a shortest-path tree). Thus the longest such path, whose length is $\mathbf{D}(ac, T_{ac})$, is also no longer than $\mathbf{diam}(T)$. Hence

$$\mathbf{AveD}(T_{ac}) \leq a + \mathbf{diam}(T). \quad (3)$$

We can get a different bound by going in a slightly different direction. The A-Plus-D Lemma again says that

$$\mathbf{AveD}(T_{ac}) \leq \mathbf{A}(ac, T_{ac}) + \mathbf{D}(ac, T_{ac}). \quad (4)$$

We know that the graph contains a path of length at most $\frac{1}{2}\mathbf{diam}(T)$ from ρ to any vertex. Thus there is a path of length at most $d(ac, \rho) + \frac{1}{2}\mathbf{diam}(T)$ from ac via ρ to **any** other vertex. The shortest paths from UC can be no longer than that, and so $\mathbf{D}(ac, T_{ac}) \leq d(ac, \rho) + \frac{1}{2}\mathbf{diam}(T)$. Thus (4) implies that

$$\mathbf{AveD}(T_{ac}) \leq \mathbf{A}(ac, T_{ac}) + d(ac, \rho) + \frac{1}{2}\mathbf{diam}(T).$$

The same reasoning that gave us the Sum of Averages Lemma tells us also that $d(ac, \rho) \leq \mathbf{A}(ac, T_{ac}) + a$, and so

$$\mathbf{AveD}(T_{ac}) \leq \mathbf{A}(ac, T_{ac}) + (\mathbf{A}(ac, T_{ac}) + a) + \frac{1}{2}\mathbf{diam}(T) \quad (5)$$

and since (1) tells us that $\mathbf{A}(ac, T_{ac}) \leq a$, this gives us

$$\mathbf{AveD}(T_{ac}) \leq 3a + \frac{1}{2}\mathbf{diam}(T). \quad (6)$$

Combining the two bounds in (3) and (6) with the formula for $\mathbf{AveD}(T)$ given in (2) gives us the two relations

$$\frac{\mathbf{AveD}(T_{ac})}{\mathbf{AveD}(T)} \leq \frac{a + \mathbf{diam}(T)}{a + \frac{1}{2}\mathbf{diam}(T)} \quad (7)$$

and

$$\frac{\text{AveD}(T_{ac})}{\text{AveD}(T)} \leq \frac{3a + \frac{1}{2}\text{diam}(T)}{a + \frac{1}{2}\text{diam}(T)} \tag{8}$$

For any value of $\text{diam}(T)$, the right-hand side of (7) is a decreasing function of a and the right-hand side of (8) is an increasing function of a .

The behavior of these functions leads us to consider two cases. First, suppose that $a \geq \frac{1}{4}\text{diam}(T)$. For a in that range, the decreasing function in (7) is maximized when $a = \frac{1}{4}\text{diam}(T)$. Thus

$$\frac{\text{AveD}(T_{ac})}{\text{AveD}(T)} < \frac{\frac{1}{4} + 1}{\frac{1}{4} + \frac{1}{2}} = \frac{5}{3}.$$

On the other hand, if a is in the range $0 \leq a \leq \frac{1}{4}\text{diam}(T)$, the increasing function on the righthand side of (8) is maximized when $a = \frac{1}{4}\text{diam}(T)$. Thus

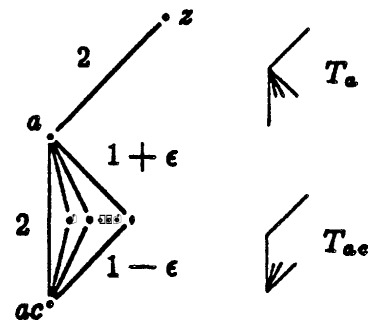
$$\frac{\text{AveD}(T_{ac})}{\text{AveD}(T)} < \frac{\frac{3}{4} + \frac{1}{2}}{\frac{1}{4} + \frac{1}{2}} = \frac{5}{3}.$$

Hence for a in either range we see that

$$\text{AveD}(T_{ac}) \leq \frac{5}{3}\text{AveD}(T),$$

which is the asserted bound. ■

As this is a rather surprising bound-it is, for instance, the only non-integral bound in this report-a bit of informal discussion may be in order. Perhaps we can gain some intuitive grasp of this bound by considering a case for which it is attained* In the example at right, we force UC to be the A-center by perturbing each unit edge slightly and having sufficiently many of the middle vertices. Thus on T_{ac} almost all of the vertices have a delay D of 5, for the path leading to UC, and then to a , and finally to z . If we had picked a as the center instead-in fact a is both the D- and the diameter-center-these vertices would have a delay of only 3; thus the ratio is $\frac{5}{3}$.



Attaining the $\frac{5}{3}$ Bound

How could we change this example to increase this ratio? We might shorten the edge to z , but this decreases the delay of 5 without decreasing the delay of 3,

□ *Center-Based Broadcasting*

so their ratio gets smaller. We might lengthen the edge to z , but this increases both delays by the same amount, which decreases their ratio. We might shorten the edge between a and uc , but this again decreases the 5 but not the 3. We cannot lengthen the edge between a and uc , since it would cease to be the shortest path between them and would not be used in any trees

We might move the middle nodes closer to either a or uc , but the Sum of Averages Lemma says that to do this without changing the distance between a and uc we must make the middle nodes farther from the other of the two. If we move them closer to uc , we decrease the 5 and increase the 3, which **again** reduces the ratio. Our last hope is to move the middle nodes closer to a , and doing this does in fact make the ratio larger-but it also changes the A-center from uc to a , so the whole example falls apart.

So we might claim in passing that this bound holds because of the Sum of **Averages Lemma**, but it must be admitted that the connection is subtle enough to be visible only on close examination.

7. Comparing the Kinds of Centered Trees.

It should be noted that although for a given measure one kind of centered tree may have a better bound than another with respect to either the minimum or the best **single** tree, it does not follow that the first kind of tree is always better than the second. Table 4 gives tight bounds on the ratio of each measure when comparing pairs of centered trees.

bound on: w.r.t.:	T_{dc}		T_{ac}		T_{diame}	
	T_{ac}	T_{diame}	T_{dc}	T_{diame}	T_{dc}	T_{ac}
MaxD	2	2	2	2	1'	1*
MaxA	3	3	3	3	2'	2''
AveD	2	2	$\frac{5}{3}$	$\frac{5}{3}$	2	2
AveA	00	∞	2	2	∞	∞

*better bound than implied by **Table 3**.

Table 4. General delay measures. Bounds comparing pairs of centered trees.

In almost all cases a bound in Table 4 is a simple corollary to the results in Table 3; that is, given a measure and two types of centered tree it is possible to find an example for which one type is the best possible tree for that measure and the other type is the worst. There are four exceptions. The first two are trivial-a moment's consideration will reveal that the measure **MaxD** of a tree is precisely the diameter of that tree. This leads immediately to the following result.

Theorem. The $\text{MaxD}(T_{\text{diamc}})$ of the diameter-centered tree is at least as small as that of any other shortest-path tree.

Proof. The diameter and the MaxD of a tree are synonymous. The **diameter-centered** tree, since it was selected explicitly to be the shortestpath tree with the smallest diameter, must therefore have a MaxD at least as small as any other shortest-path tree. \square

This accounts for the pair of ones in the first row of Table 4. The other two exceptions are more surprising. Not only is the MaxD of the diameter-centered tree better compared to other centered trees than Table 3 leads us to expect, but so is its MaxA . Table 3 implies a bound of three, but the tight bound is only two.

Theorem. The $\text{MaxA}(T_{\text{diamc}})$ of the diameter-centered tree is at worst twice that of any other shortest-path tree.

Proof. Given a tree T , the definition of D tells us that for any v and w ,

$$d(T, v, w) \leq D(v, T)$$

and averaging over w gives us

$$A(v, T) \leq D(v, T),$$

so we know that

$$\text{MaxA}(T) \leq \text{MaxD}(T)$$

by maximizing over v . On the other hand, if a and b are endpoints of a diameter of T then

$$\begin{aligned} \text{MaxD}(T) &= d(T, a, b) \\ &\leq A(a, T) + A(b, T) \end{aligned}$$

by the Sum of **Averages** Lemma. Each of these averages can be no more than MaxA , so

$$\text{MaxD}(T) \leq 2\text{MaxA}(T).$$

Thus we see that

$$\text{MaxA}(T) \leq \text{MaxD}(T) \leq 2\text{MaxA}(T). \tag{1}$$

So for any shortest-path tree T , we see that

$$\begin{aligned} \text{MaxA}(T_{\text{diamc}}) &\leq \text{MaxD}(T_{\text{diamc}}) \\ &\leq \text{MaxD}(T) \\ &\leq 2\text{MaxA}(T), \end{aligned}$$

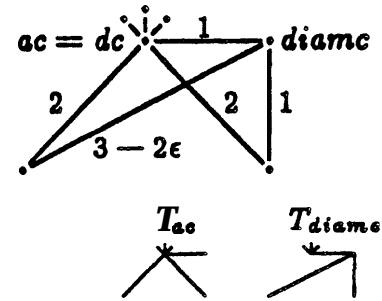
since the diameter-centered tree has the smallest diameter and hence the smallest MaxD of any shortest-path tree. \blacksquare

8. Demonstrating Tightness of the Bounds.

So far we have shown the tightness of only one of the bounds given in Tables 2, 3, and 4. A complete discussion of the examples that attain these bounds would be tedious, but we may profitably consider a few of **them**.

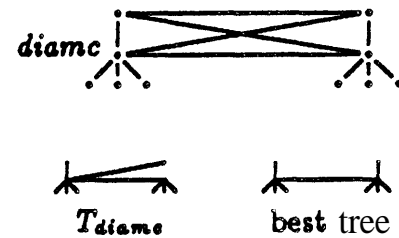
As an illustration, consider the bounds for $T_{diameter}$ in the third column of Tables 2 and 3 and in the last two columns of Table 4. Three examples suffice to demonstrate the tightness of all these bounds.

The first example deals with the measures **MaxD** and **MaxA** for $T_{diameter}$ compared to the other two forms of centered tree. In this example the **A**- and **D**-centers are identical. There are a large number of vertices adjacent only to this center, at a distance of ϵ . Thus we see that the diameter-centered tree has both a **MaxD** and a **MaxA** of $4 - \epsilon$, given enough c-edges incident to **ac**. The **A**- and **D**-centered tree, on the other hand, has a **MaxD** of 4 and a **MaxA** of essentially $2 + \epsilon$. We can make the number of nodes arbitrarily large and ϵ arbitrarily small, to bring the ratios arbitrarily close to the bounds of 1 and 2 given in Table 4.



**MaxD and MaxA
In Table 4**

The second example deals with **MaxD** and **MaxA** for $T_{diameter}$ when we compare them with those of *any* single tree. It consists of a rectangle whose long sides and diagonals all have weight 1, and whose short sides have weight ϵ . In addition there are a large number of c-edges incident to the two bottom corners. In this case either bottom corner may be the diameter-center; the diameter-centered tree contains two adjacent sides and a diagonal, giving a **MaxD** of $2 + \epsilon$ and a **MaxA** of $\frac{3}{2} + \epsilon$. The best tree, on the other hand, contains two short sides and one long one, giving a **MaxD** of $1 + 2\epsilon$ and a **MaxA** of $\frac{1}{2} + 2\epsilon$. Again, the ratios can be made arbitrarily close to the bounds of 2 and 3.



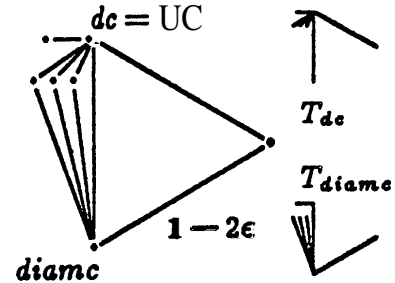
**MaxD and MaxA
In Table 3**

It should be noted in the previous example that the best tree is *not* a shortest-path tree. This is consistent with the proofs of the bounds on $\text{MaxD}(T_{diameter})$ and $\text{MaxA}(T_{diameter})$ in Table 4, which show that no shortest-path tree can be as much better than $T_{diameter}$ as Table 3 allows. To show that the bounds in Table 3 are tight, we have no choice but to use a tree that is not a shortest-path tree.

0. Distributed Construction of Centered Trees \square

It should be noted in addition that this example also demonstrates tightness of the bounds in the corresponding positions of Table 2. Since source-based forwarding is at least as good as any single tree, an example that attains the bound when T_{diamc} is compared to another tree must also attain that bound when T_{diamc} is compared to source-based forwarding.

The third and final example deals with the measures of **AveD** and **AveA**. This example is sufficient for all three tables, since we have the same pair of bounds whether we compare T_{diamc} to the other centered trees, to other trees in general, or to the optimal scheme of source-based forwarding. This example has three vertices linked in an almost-equilateral triangle together with a large number of vertices that can be reached from the diameter-center in 1 and from the D- and A-center (which are the same vertex) in ϵ . In addition there is a single c-edge incident to dc that cannot be reached directly from $diamc$.



AveD and AveA

The **AveD** and **AveA** of the diameter-centered tree are each essentially 2. The D- and A-centered tree has an **AveD** of $1 + \epsilon$ and an **AveA** of 2ϵ . Thus the ratio of the **AveD**'s is essentially 2, which is the bound given in the tables. The ratio of the **AveA**'s, on the other hand, is $1/\epsilon$, which can be made arbitrarily large; thus no bound is possible.

Thus we see that the bounds claimed for the diameter-centered tree are indeed tight. A more precise discussion of the preceding examples, together with examples that attain the rest of the bounds, may be found in Wall's thesis [9].

9. Distributed Construction of Centered Trees.

As we discussed previously, our scheme for constructing a centered tree has two phases. First, each vertex independently looks at the routing tables for the underlying single-destination mechanism and computes the criterion by which the center will be picked. Then a distributed protocol pools these results, selects the vertex that minimizes this criterion, and constructs the appropriate shortest-path tree. The same protocol suffices for each of the three forms of centered tree.

The underlying tables are assumed to have information that allows a message to be forwarded to the appropriate neighbor, but not global information about the structure of the network far away. Specifically, the tables at a given vertex v will include, for each other vertex w in the network:

- (a) the length of the shortestpath from v to w , and

□ *Center-Based Broadcasting*

(b) the identity of each neighbor of v that is the first vertex on a shortest path from v to w .

The latter consists simply of each neighbor z such that $d(v, w) = t(v, z) + d(z, w)$, where $t(v, z)$ is the time associated with edge $\{v, z\}$ and $d(z, w)$ is the delay on the shortest path from z to w .

Given this information, it is easy for a vertex v to calculate its own $D(v, T_v)$ or $A(v, T_v)$ —it simply takes the table of distances and finds its maximum or average value. However, computing $\text{diam}(T_v)$ from this information is not possible because the diameter need not pass through v , and the vertex v may not be able to **tell** whether it does or not. The diameter-centered tree may still be constructed, however, because a vertex **can** tell whether or not one of the incident branches of its shortest-path tree contains the midpoint of a diameter; and furthermore there is at least one vertex which has this property and whose tree is a minimum-diameter shortest-path tree. This is enough information to allow a vertex to tell whether it is a candidate or not; such vertices contribute their diameter to the pool, and others contribute **infinity**. Wall's dissertation [9] contains details of this construction.

We must also specify the protocol by means of which this pool can be used to select the center. A complete description of this protocol is beyond the scope of this paper but in brief the idea is this: if a vertex could somehow know that it is the **center**, it could send out a multiply-addressed message announcing the fact, A vertex that receives this message could note which edges it uses to forward it, and thereafter a broadcast message could be forwarded by forwarding it along those same edges; thus the shortest-path tree is constructed and future **broadcast** messages need not have the multiple-address header.

But there is no way for a vertex to know by magic that it is the center **We can** accomplish the same thing, however, by letting every vertex send out a **multiply-addressed** message containing its value of the criterion. When all these **messages** are finished, each vertex knows the criterion value for every other vertex, and can minimize this to discover which vertex is the center and hence which of the sets of forwarding edges it should use thereafter.

This is an unnecessary amount of work, however; in fact it is essentially the scheme we would use to set up the tables for source-based forwarding! **We can** reduce it considerably since many of the messages will be telling a **vertex about** a criterion value that is larger than another value the vertex has already **seen**. There is no need for the vertex to forward such a message, since the source of the message cannot possibly be the center. Moreover, there is no need to remember several sets of forwarding edges; it suffices to remember only those edges associated with the smallest criterion value seen so far. This results in a significant reduction of the number of messages sent and the amount of storage needed to set up the tree.

A protocol that can be used in an actual network environment must be rather

more complicated, since it must also **allow** for the fact that as edge weights change, the routing tables may become inconsistent from vertex to vertex; the path actually followed by a message may loop back on itself if a weight has changed but the fact has not yet propagated through the entire network. These difficulties **are discussed in** more detail in Wall's thesis [9].

10. Conclusions.

Source-based forwarding can be used to minimize the delay associated with broadcasting, but it incurs a large amount of overhead and is not easy to make responsive to changing network conditions. Broadcasting can be done without these drawbacks by routing over a single spanning tree, but this tree should be selected carefully if the delay is to be small. Constructing the spanning tree that minimizes one of the four delay measures we have discussed is in general hard to do, especially in a network environment in which the necessary information **may** be distributed **among the nodes**.

Center-based forwarding provides a broadcast mechanism that is **easy** to maintain in a distributed environment. Furthermore, by selecting the type of centered tree according to what measures of the delay concern us, we can provide a **delay** which is a small constant times the minimum delay. Thus some form of center-based forwarding seems an excellent choice for a broadcast mechanism.

The bounds shown in this paper give us some information that may help in selecting which center-based algorithm to use. The measures of **MaxD** and **MaxA** demonstrate an advantage to using the diameter-centered tree rather than either of the other two, in that T_{diamc} has a **MaxD** at least as good and a **MaxA at worst** twice that of T_{dc} or T_{ac} , while the latter two can have a **MaxA** three times as bad as that of the diameter-centered tree. However, no better bound can be shown comparing it to the best tree or to source-based forwarding. The measures **MaxD** and **MaxA** do not distinguish between the A- and D-centered trees.

If in selecting a center-based algorithm the overriding concern is to finish a given broadcast as soon as possible, so that the measure D is much more important **than** the measure A, using the D-centered or diameter-centered tree may be acceptable. However, the A-centered tree may still be better, since it gives a guaranteed bound of 2 for the worst- and average-case A, while the other two cannot give any bound. Moreover the A-centered tree gives as good a bound on the average-case D, though the other two give a better bound on the worst-case D. The final decision must be based on the question of which delay measures **are** more important to the applications using the broadcast mechanism.



References.

- 1] E. Ball, J. Feldman, J. Low, R. **Rashid**, and P. Rovner. RIG, Rochester's intelligent gateway: System overview. **TR5**, Computer Science Department, University of Rochester, April 1976.
- 2] **Yogen** Kantilal Dalal. Broadcast Protocols in Packet Switched Computer Networks. **PhD** thesis, Stanford University, April 1977. (Stanford Computer Systems Lab Technical Report 128.)
- 3] Jim Gray. Notes on data base operating systems. **IBM Research** Report RJ2188 (**30001**), San Jose, California, **1978**.
- 4] D. S. Johnson, **J. K.** Lenstra, and A. H. **G.** Rinnooy Kan. The complexity of the network design problem. *Networks*, Volume 8, Number 4, 1978, pages 279-285.
- 5] William A. Kornfeld. ETHER-a parallel problem solving system. Sixth International Joint Conference on Artificial Intelligence, **August 1979**, pages **490-492**.
- 6] John **M.** McQuillan. Adaptive Routing **Algorithms** for Distributed Networks. **PhD** thesis, Harvard University, May 1974 (BBN Report 2831).
- 7] John **M.** **McQuillan**, Ira Richer, and Eric C. Rosen. **An** overview of the new routing algorithm for the **ARPANET**. Sixth Data Communications Symposium, November 1979, pages **63-68**.
- 8] Reid Garfield Smith. A **Framework** for Problem Solving in a Distributed Environment. **PhD** thesis, Stanford University, December 1978. (Computer Science Department Technical Report **HPP-78-28**.)
- 9] David Wayne Wall. Mechanisms for Broadcast and Selective Broadcast. **PhD** Thesis, Stanford University, June 1980, (Stanford Computer Systems Lab Technical Report 190.)

