ELSEVIER

Computer Communications 21 (1998) 912–923

computer communications

Scalable and efficient broadcasting algorithms for very large internetworks

Samir Chatterjee[a,*], Mostafa A. Bassiouni[b]

[a]Computer Information Systems Department, Georgia State University, Atlanta, GA 30302, USA
[b]School of Computer Science, University of Central Florida, Orlando, FL 32816, USA

Received 13 February 1998; accepted 25 February 1998

Abstract

Broadcast is a special case of routing in which a packet is to be delivered to a set that includes all the network nodes. While dynamic and distributed broadcast techniques have been proposed and used in the Internet, unfortunately they suffer from scalability problems, i.e. they are not efficient with respect to the tremendous size of today's networks. Moreover, it has been observed that the cost of routing and the broadcast time are two conflicting performance measures as far as optimization is concerned, especially in large networks. Also, many of the current techniques are not robust enough and give low performance under events of link failure. First, we show that in order to achieve universal reachability, internets have naturally acquired a multi-level hierarchical structure. Second, utilizing this existing hierarchy, we propose scalable broadcasting protocols which achieve near-optimal cost and time measures. Because of the hierarchy, our proposed algorithm only maintains information of links connected to direct neighbors, thereby making it scalable to future growth in size of the network. We show that time-optimal broadcast in point-to-point networks can be achieved by formulating the problem as finding maximum marching in bipartite graphs. Several heuristics based on matchings are presented. Performance bounds are derived along with numerical and simulation results obtained that prove the validity and feasibility of the scheme. © 1998 Elsevier Science B.V.

Keywords: Broadcasting; Routing; Minimum spanning tree; Bipartite graphs; Maximum matching; Hierarchical networks; Distributed algorithms

1. Introduction

1.0.1. Motivation

The routing of packets to their intended destination forms a fundamental service provided by any packet switched network. Broadcast, the task of delivering a single message from a particular source node to all the other nodes in the network, is a special case of general routing. While the operation of unicast (one-to-one) or multicast (one-to-many) may arise quite often in a computer network, the process of broadcasting is usually reserved for special situations when an important message (like configuration, routing or setup control) is to be sent to all involved nodes. While broadcasting in a small conventional LAN environment is well studied, large internetworks possess special challenges for broadcasting.

The Internet uses broadcasting in a variety of different places. Fig. 1 shows a logical structure of the Internet, as it evolved naturally. At the backbone level, there exists a set of core routers[1] that maintain route information for all destination subnetworks in the Internet. In order to build valid routing tables, these core routers use either a distance-vector based or a link-state shortest path based algorithm. GGP protocol (gateway-to-gateway protocol) based on Bellman–Ford vector-distance algorithm was used to construct complete routing-tables which then each core gateway would share by broadcasting the route information to all other core gateways. GGP is no longer used today as it suffered from scalability (exchanged messages are of size proportional to the number of networks) and slow convergence problems. The basic idea behind the newer technique known as shortest-path-first (SPF) link-state is very simple:

1. Each core router is responsible for meeting its neighbors and learning their names.
2. Each router constructs a packet known as a 'link-state packet', or 'LSP', which contains a list of the names of and cost to reach each of its neighbors.
3. The LSP is somehow transmitted to all other routers (broadcasted), which then stores the most recently generated LSP from each other router.

* Corresponding author. Tel: 00-1-404-651-3886; fax: 00 1 404 651 3842; e-mail: schatter@gsu.edu

[1] In the Internet, only the gateways (or routers) are involved in routing and route propagation.

PII S0140-3664(98)00140-6

4. Finally each router armed with a complete map of the topology (the information in the LSPs yields knowledge of the graph), computes routes to each destination by applying the well-known Dijkstra shortest path algorithm to the resulting graph. Because routers perform route computation locally, it is guaranteed to converge and link-state algorithms are much more scalable since they exchange messages that contain only nearest neighbor information.

In a distributed database environment, maintaining consistency of replicated data requires broadcasting, may be not to all computers in the subnetwork, but to almost all which are sites to that database. Broadcasting is also used to determine location of a resource known only by name or property and is typically used by search engines. Recent advances in the Internet have produced increasing use of broadcasting to a large number of nodes. The world-wide-web servers and location servers for mobile IP (address migration over Internet) need to acquire and disseminate many routing related informations. Also, the domain name system (DNS) root servers keep updated domain-name to Internet address mappings inside large tables that we periodically shared amongst a set of core root name-servers. The point is, as the structure of the Internet changes (by addition and deletion of nodes or subnetworks), route servers learn new information which must be disseminated across all involved routers.

Generally, it is desirable to broadcast packets over the best possible path available. An important aspect of the design and implementation of routing procedures is the criteria used for path selection [24]. Each router must be configured with a cost for each link to which it is attached. When a router receives a data packet for forwarding, it must know what metric the user would like to employ for computing the route to the destination. Four metrics are typically used for the 'quality of service' option routing:

- Default: intended to correlate (inversely) with bandwidth.
- Delay: intended to correlate with the amount of delay on the link.
- Expense: intended to correlate with the amount of money it costs to use the link.
- Error: intended to correlate with the flakiness of the link.

Three major factors are driving the research in this area: scalability, efficiency and reliability of broadcasting in a large internetworking environment. Most existing techniques do not scale well to the current growth of the Internet. Efficiency deals with fast convergence and the ability to minimize both the broadcast delay and cost of routing which has been a rather difficult problem. Moreover, any new broadcast technique must be reliable, i.e. recover under events of link failure. Our objective in this paper is to show that for arbitrarily large internetworks, it is possible to achieve a scalable, reliable, new-optimal cost and time scheme using a hierarchical broadcasting approach [20].

1.0.2. Background and related work

Shortest-path based routing procedures can be classified into either static or dynamic and centralized versus distributed. If the cost used in the procedure is based on some real-time measurement of network conditions, the routing procedure is said to be dynamic. Otherwise it is assumed to be static. In a centralized routing procedure, a central site is in charge of computing the best possible paths in the network. In a distributed routing procedure, all network nodes are involved in shortest-path calculations. A node typically has direct knowledge of its local links only, a distributed procedure pools the information available at each node to collectively come up with a map of the best possible paths across the networks. Since our work is applicable to truly large networks, the procedures used here are dynamic and distributed in nature.

Broadcasting techniques in conventional small sized networks have been studied extensively by several researchers. McQuillan et al. [1] proposed the Arpanet:

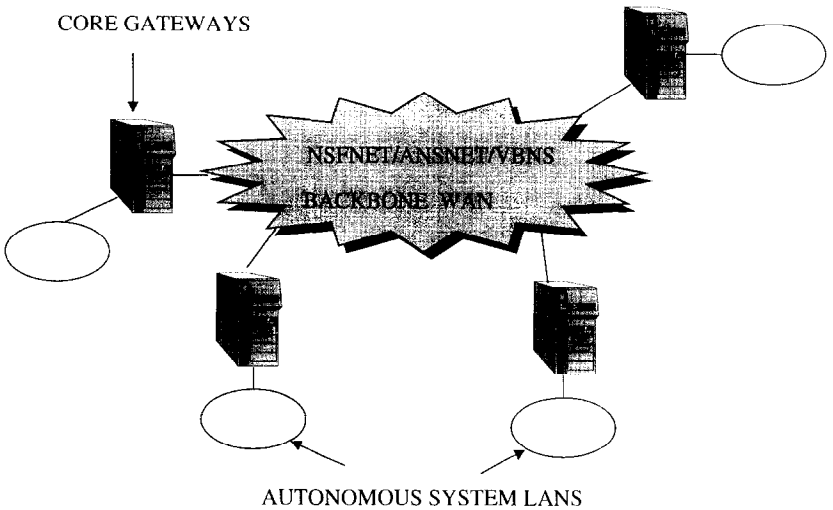


Fig. 1. The structure of Internet showing core gateways and autonomous systems.

routing technique based on flooding. Flooding can be wasteful of network bandwidth as the number of packets omitted far exceeds the minimum required. Farley [2] proposes several minimal broadcast networks. Kumar et al. [3] addresses the multidestination routing problem. In this approach, packet formats are modified to hold a set of destination addresses. A node delivers a copy of an incoming packet to its attached hosts and deletes their addresses from the list of destination addresses. In Ref. [4] an exact algorithm to achieve time optimal broadcasting in general point-to-point networks is presented.

Two broadcast techniques have become popular over the years: spanning tree forwarding and reverse path forwarding schemes [13,21]. In the first, a spanning tree of the network is defined and each node maintains information about which of its outgoing links belongs to the spanning tree. When a node receives a broadcast packet, it forwards a copy on each of its outgoing links that belong to the spanning tree except the incoming link. A drawback is that each switching node is required to maintain additional information which is required solely for broadcast routing. In reverse path forwarding, when a node receives a copy of a broadcast packet, it first determines, using its routing table, the outgoing link that it normally would use to reach the source of the broadcast packet. It then checks to see if the link over which the broadcast packet arrived is the same as the link just determined. If there is no match, the packet is simply discarded. If there is a match, a copy of the packet is sent on all of the node's outgoing links (including links to any attached hosts) except the incoming link. A self-stabilizing LSP distribution scheme has been in effect (sometimes called the new Arpanet LSP scheme) since the late 1980s on the Internet backbone. For each LSP in a router R's memory requires a $2 \times k$ set of flags associated with it, where k is the number of links connected to R [24].

Some of the earliest work on routing in distributed networks can be found in Ref. [5]. A good survey of routing schemes is presented by Schwartz [6]. A description of the use of variations of the distributed Arpanet routing algorithm is found in Refs. [7,8]. The revision of the delay metric for the new Arpanet algorithm is found in Ref. [9]. Scalability issues have been addressed in Refs. [10–12]. Broadcasting and routing is a subject of further research, particularly in the context of high-speed networks. Gopal et al. [14] proposes hardware flooding and Maxemchuk [15] includes good discussion on routing in ATM networks.

The rest of the paper is organized as follows: in Section 2, we state the research problem that is addressed in this paper. Section 3 provides a system model along with the concepts and definitions that are used throughout the paper. It also discusses relevant issues in hierarchical routing: Section 4 presents the algorithms. Analysis results and performance bounds are also derived here. Some numerical results are presented in Section 5 and are also compared with existing algorithms that are used in large networks. Finally, Section 6 concludes with a summary of the contributions of this research and points out some directions of future work.

2. Problem statement

The growth of the Internet and other similar public data networks has been unprecedented. To achieve universal reachability and to provide many, specialized services, the process of broadcasting is being repeatedly used amongst a large set of network nodes. One of the vital issues facing the current Internet and other large networks is scalability, i.e. any proposed algorithm must scale well with the tremendous size of the network. Moreover, as pointed out earlier, any broadcasting process can be measured in terms of the cost and time that it takes for all involved nodes to receive a broadcast message. Further, the proposed scheme should

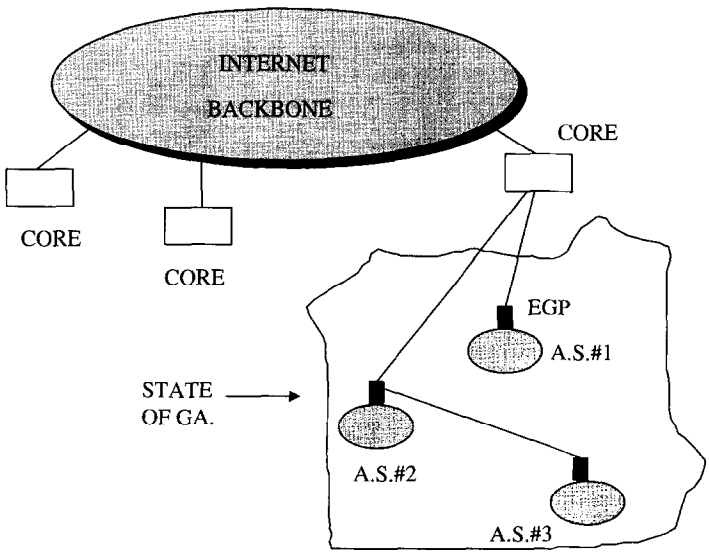


Fig. 2. Actual Internet architecture showing core gateways, autonomous systems and exterior gates.

have fault tolerance embedded inside so that under events of link failure, the large number of nodes are still able to broadcast their critical information and keep the internet operational. Our goal in this work is to compute a broadcast algorithm that is efficient (in terms of minimizing cost and time), scalable (in terms of size) and reliable.

3. The network model

The broadcast process does not take place between every host node on the Internet (currently greater than 10^8 hosts), but between a very large set of core and non-core routers. In order to understand the network model, it is important to track the actual development of the Internet. The initial architecture of the present day Internet consisted of a core backbone system of routers (Arpanet which later became NSFnet and now vBNS) to which attached single LANs. It has since then evolved to a more sophisticated and complex structure where most sites have multiple LANs and multiple routers interconnecting them. A large TCP/IP internet must have additional structure to accommodate administrative boundaries: each collection of networks and gateways managed by one administrative authority is considered to be a single autonomous system.

An autonomous system is free to choose an internal architecture and routing structure, but must collect information about all its networks and designate one or more routers that will pass the reachability information to other autonomous systems. Because the connected Internet uses a core architecture, every autonomous system must pass reachability information to Internet core routers.

Shown in Fig. 2 is the actual hierarchical structure that the Internet has acquired over the past years. Since any individual site can have an arbitrarily complex structure, a core system cannot attach directly to all networks. Hence, they connect to a group of networks and routers managed and controlled by a single administrative authority. The basic idea behind autonomous systems is to provide an additional way to hierarchically aggregate routing information in a large internet, thus improving scalability. Every autonomous system must declare at least one gateway which will broadcast reachability information to other autonomous systems and also to a core router at the backbone. We call such a machine an exterior gateway which usually advertises its network reachability using an exterior gateway protocol EGP [23]. It is important to understand that EGP restricts a router (noncore) to advertise only those networks reachable entirely from within its autonomous system. This third party rule is intended to control the propagation of information and allow each autonomous system to choose exactly how it advertises reachability. This restriction imposes a fundamental limitation to the architecture of the Internet; it restricts the topology of any internet using EGP to a tree structure[2] in which a core system forms the root; there are no loops among other autonomous systems connected to it.

Every state in the US is likely to have a number of autonomous systems whose exterior gateways running EGP connect to each other in an arbitrary manner using leased data circuits (see Fig. 2) operating at fractional T-1 rates. Two exterior gateways may also be connected to each other over a subnetwork like an NSF funded regional WAN.

Thus, large internets have naturally acquired a hierarchical architecture. For scalability purposes, this hierarchy can be exploited by organizing the nodes into clusters (groups of nodes) [16]. Our network model will consist of core router nodes and exterior gateway nodes. Each EGP gateway is a cluster 0 node. Since every state in the US will have many autonomous systems and, hence, many exterior gateways, we group such EGP nodes into a higher cluster called level 1. That means, a level 1 cluster contains a collection of many exterior gateways or cluster 0 nodes. Further, because of the tree restriction, at least one of the nodes within cluster 1 will have a leased connection to a backbone core router. All core routers at the backbone are cluster level 2 and, hence, the NSFNET/ANSNET backbone routers belong to cluster 2. All cluster 2 nodes (server gateways) are interconnected to each other via leased T-1 data circuits. Inductively, we can define third-level, fourth-level and so on, based on regions and other properties. Clustering helps to reduce the amount of information that is needed to be stored to accomplish routing as was demonstrated by Kamoun et al. [16]. The graph model that we consider for the rest of the paper is shown in Fig. 3.

3.1. System model: some definitions

We model large internetworks as an undirected graph $G = [V, L]$ in which $V = \{v_1, v_2, \ldots, v_n | n \text{very large}\}$ is a set of nodes containing gateways (both core and non-core). These gateways act as servers for routing. L is a set of links, having at least $n - 1$ links or edges. We assume links to be of two types: either point-to-point link (direct leased connections) or a subnetwork over which two gateways are connected. In order to evaluate the quality of routes between the network nodes, a positive cost $c(l)$ is associated[3] with every link $l \varepsilon L$.

Typically, WANs and regional-level MANs do not support broadcast, and hence delivery is carried out by forwarding individual copies of the message point-to-point. We assume that a node communicates with another by transmitting a message, or making a call. Each call involves two nodes and takes one unit of time[4] to complete. In other words, the unit to measure time is equal to the length of the interval needed to complete a message call. The notation

[2] In October of 1995, the global Internet changed to include multiple core backbones and now allows cycles within autonomous systems that are running BGP version 4 protocol. But there exists many portions of the global Internet and many corporate intranets which still obey the tree-like structure which have hanging autonomous systems from a single backbone.

[3] Cost here reflects the economics of leasing lines for various commercially available data rates.

[4] Time here is unrelated to data rates of leased circuits which determine the actual delay (transmission plus propagation).

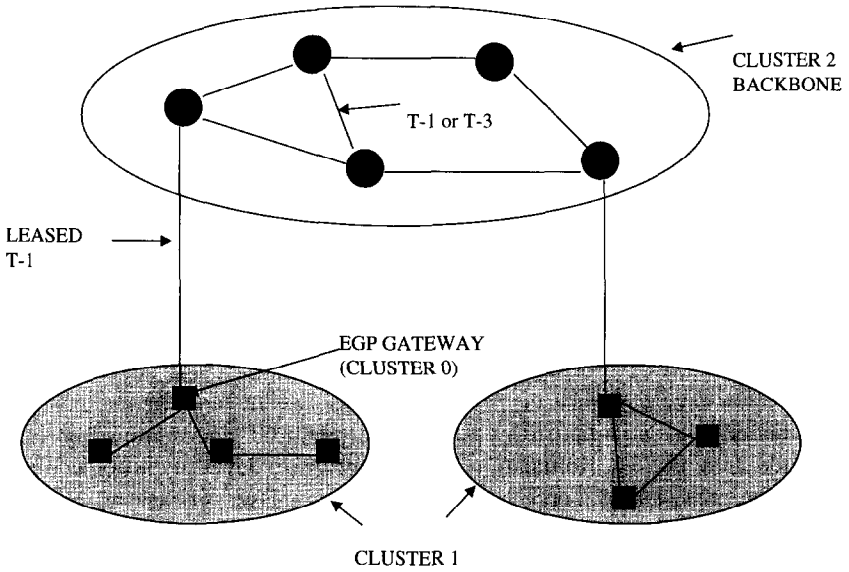


Fig. 3. A graph model showing hierarchical cluster of gateways and core nodes.

(v_i, v_j, t) is used to represent a call from sending node v_i to receiving node v_j during time unit t. Further, it is assumed that the two participating nodes in a call are adjacent and that a node may participate in at most one of a act of concurrent calls (i.e. calls sharing the same unit time interval)[5]. Thus, a broadcast $B(G, u)$ in a network G originating at node u is a set of $(n-1)$ calls (a tree spanning all the nodes in the network). Notice that our model is restricted to that of typical point-to-point networks in contrast to LANs which are equipped with broadcast facilities.

The span of a broadcast is the smallest interval (t_0, t_1) such that every call (v_i, v_j, t) in the broadcast satisfy, $t_0 \leq t \leq t_1$. The length of a broadcast with span (t_0, t_1) is $t_1 - t_0 + 1$. An optimal broadcast is a broadcast of minimum length. The broadcast time $b(G, u)$ of a node u in the network G is the length of an optimal broadcast in G, originating at node u. The broadcast time $b(G)$ of a graph is the maximum broadcast time among all nodes in G.

It is easy to note that if G is a complete graph on n nodes (K_n), the broadcast time is the best possible, i.e. $b(K_n) = [\log_2 n]$. The worst time occurs in the case of a path of n nodes and the originating node is either one of the extremes. Then $b(P_n) = n - 1$. For any other arbitrary connected graph, $b(G) \geq r[\log_2 n]$.

Each call (v_i, v_j, t) has a cost associated with the link (v_i, v_j). If B is a broadcast consisting of $(n-1)$ calls, then we define broadcast cost $C(B, u)$ as

$$C(B, u) = \Sigma_{1 \epsilon B(G,u)} c(1) \quad (1)$$

Ideally, the optimal broadcasting algorithm will be one which minimizes both the time $b(G, u)$ as well as broadcast cost $C(B, u)$ of a broadcast B. As we develop the algorithms in later sections, we shall show that these two performance measures are conflicting with each other and it is therefore hard to obtain a single algorithm which is optimal in this sense. We now discuss the details of our proposed broadcast scheme which obtain near-optimal results.

4. Overview of our broadcasting approach

The graph shown in Fig. 3 includes all the nodes that are involved in the broadcast process. Owing to the acquired hierarchy, there are some distinctions between the nodes in the graph that need to be pointed out. First there is only one cluster at level 2, as only a single backbone of core gateways is assumed. But there are many clusters at level 1, as each state may have at least one connecting to the backbone via some exterior gateway. Note that some areas (like the bay area) will likely have several clusters at level 1 owing to the high density of computers in the region.

The backbone cluster is a collection of small number of core gateways which store all information needed to reach all possible routes. This is typically a wide area network with any two nodes separated by a large distance. The cost of these wide area links (usually leased data circuits) is significant here and they are likely to exhibit a high degree of variability[6]. If we assume that costs are interpreted as proportional to data rates used and the distance of separation, then the backbone links (in NSFNET) will have varying speeds from T-1 to T-3 or higher depending on traffic patterns and loads. In order to achieve cost efficiency within the core backbone, the calls in the broadcast must be along the edges of a minimum cost spanning tree (MST). We propose the use of a fully distributed minimum cost

[5] Other assumptions have been defined elsewhere in which a node may participate in a number of concurrent calls or may make a call to a node that is not adjacent. Our restricted point-to-point model is reflective of actual Internet router connections across the regions.

[6] Some of these links are currently being upgraded to gigabit speeds using SONET/ATM technologies.

spanning tree algorithm at the backbone cluster of core gateways. Since the core gateways are separated by large distances and they are interconnected by links having a high degree of variability in cost, minimizing the broadcast cost here is justifiable. It is important that each core gateway routes broadcast message along the best possible path, which is measured here in terms of cost.

Moreover, in order to achieve scalability, any algorithm used at this cluster must be distributed with each node maintaining information about its directly connected neighbor. When the size of messages that are exchanged is of the order of the number of networks, then bandwidth consumed and the storage requirements become too high, thus not scaling well. In our hierarchical internet model, the core nodes in the backbone only maintains direct neighbor information for its connections to the other core gateways within the level 2 cluster. This makes our approach highly scalable to future growth. As far as the backbone is concerned, the nearest neighbor topology is least likely to change with time unless a new core gateway is added somewhere to the NSFNET. What changes in a dynamic internet are the nodes, autonomous systems and clusters, which get added or deleted from time to time. In order to broadcast to every cluster 1 node, the core gateway attached to the level 1 cluster maintains an updated topology database showing all the relevant routers within the level 1 cluster. Note that while the storage requirements for such a topology map may be moderately high, it does not consume any additional bandwidth nor suffers from scalability problems as such maps are not exchanged but each core stores a centralized database to execute the local cluster algorithms. Hence, any changes that take place within level 1 clusters are automatically reflected in the topology map maintained inside the core gateway. At the backbone level, we utilize the direct neighbor links information in a distributed algorithm which computes a minimum-cost spanning tree out of a graph composed with the link status information obtained by each neighbor using an adaptive message passing mechanism.

Once a message is received by all core gateways, it needs to be disseminated to all the attached clusters. Clusters at level 1 are expected to be in close physical proximity (compared with backbone nodes) and, hence, it is justifiable to broadcast the message to every node in the level 1 clusters in the minimum time. We propose to use time optimal algorithms to broadcast message to every node in all level 1 clusters. This again might deviate from cost optimal values. But we argue that the cost variation of links within clusters at level 1 is much less as they are all standard leased data circuits of common 56 kbps rates. If all links use the same rate, then a time-optimal tree is also a minimum cost tree. However, we show that in clusters which exhibit a high degree of cost variation, it is possible to modify the cluster algorithms slightly to achieve better cost values along with time optimal results. Hence, we propose using a distributed minimum-cost broadcast algorithm at the backbone core cluster and time-optimal broadcast within each level 1 cluster in the hierarchy.

Finally, an important point remains to be mentioned. The reason a broadcast process is initiated is because some critical information is generated from within one of the several autonomous systems. The message originates from a host node and arrives at an exterior gateway in that autonomous system using an internal message exchange protocol. The popular routed program implementing routing information protocol (RIP) is used by machines to provide consistent routing and reachability information within the autonomous system boundaries. We will not consider the couple of messages that RIP uses to reach the exterior gateway as part of our broadcast process. It is only after an EGP gateway[7] receives a broadcast message that it initiates our hierarchical and distributed broadcasting algorithms.

4.1. A distributed MST algorithm (DMST)

In a network using distributed procedures, network nodes cooperate in determining the shortest paths in the network. To achieve broadcasting within the backbone, three distinct phases are involved. The first phase is to gather link status information by continuous exchange of messages among network nodes. This process is already part of the existing link-state routing technique currently in use on the Internet. Generally, the information exchanged is in the form of internodal distances, with each node estimating the total distance (i.e cost) to other nodes based on its knowledge of local link conditions of its direct neighbors. The update interval of gathering this link state information may range from a few seconds to a few hours[8].

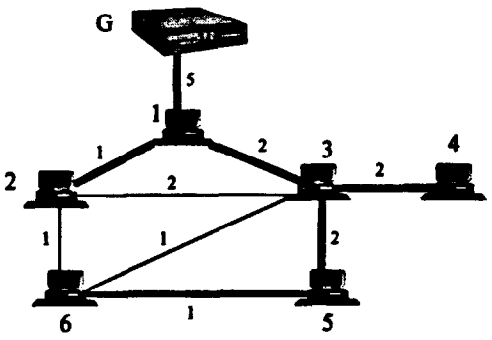
The second phase is the actual construction of a minimum-cost spanning tree (MST). We implemented a fully distributed MST algorithm as outlined in the work by Gallager et al. [17]. For a network of N nodes and E edges, the algorithm requires $5N\log_2 N + 2E$ messages to determine the MST. The algorithm uses the notion of fragments which is defined as a subtree of the MST, i.e. a connected set of nodes and edges of the MST. The algorithm starts with each individual node as a fragment and ends with the MST as a fragment. The algorithm actually specifies the edges that belong to the computed minimum-cost MST. A detailed discussion of the procedure is beyond the scope of this paper, but the reader is referred to Ref. [17]. From now on, we refer to this algorithm as DMST[9].

The third and last phase is the actual broadcast that utilizes the MST links in order to route through minimum cost

[7] Since the EGP may not attach directly to the core, the cost and time values we measure are average values over all possible sources.

[8] A 10 s period has been used as a basis of link performance in the revised ARPANET procedure.

[9] A new computation of DMST is initiated only if the cost values have changed significantly or new core gateways attaches to the backbone cluster.



TIME STEP	CALL	COST
1	(G,1)	5
2	(1,3)	2
3	(1,2), (3,5)	1+2
4	(3,4), (5,6)	2+1
TOTAL: 4		13

Fig. 4. An example cluster network and its time-optimal broadcast.

edges. This can be accomplished by forwarding broadcast packet along each of the outgoing links that belong to the MST except the link through which it receives a copy. Since each message has a timestamp and an age field, the controlled forwarding scheme is quite efficient.

A broadcast packet has a special destination address of all 1's. When a core node (belong to cluster 2) receives such a broadcast packet, it checks to see if a current MST exists. The core node will then send the packet on the outgoing links which belong to the MST. If an MST does not exist, it will initiate phase 1 to build a spanning tree. Note that the link-status update interval should detect any possible link failures and if the status changes from the last computed MST, a new computation would begin. The core node does not need any extra, storage for the construction of the DMST or for broadcasting within the backbone core.

4.2. Local cluster algorithms

The objective of cluster algorithm is to find broadcast paths such that broadcasting can be achieved in minimum time. At the successful completion of the DMST algorithm mentioned above, every core gatenode has received a copy of the message. It must now disseminate that message to all the routers within the local cluster.

4.2.1. An example

Shown in Fig. 4 is an arbitrary cluster connected to a core node G. The figure shows an optimal broadcast whose links are shown by thicker dark lines. The broadcast time equals 4 while the associated cost is 13. Note that while time is optimal in this case, it does not yield a minimum cost for the cluster. The minimum cost would be along the links that belong to the minimum cost spanning tree (MCST) which any generic cut-set type algorithm would yield the set $\{(G, 1), (1, 2), (2, 6), (6, 3), (6, 5), (3, 4)\}$ for a cost of 11. But the MCST has a broadcast time of 6. This example shows that a time-optimal broadcast may not be cost optimal (only if all cluster links have same[10] cost) and a MCST may not yield time optimal results. This shows that optimizing both the time and cost within any arbitrary cluster may be conflicting to each other.

[10] In that case the coefficient of variation $C.V$ for the link costs is zero.

4.2.2. The key concept

How do we achieve the minimum broadcast time in the above example? Note that we can achieve time optimality if during each time interval, we can broadcast to as many nodes as possible. Since a node can participate in only one call at any time interval, we must broadcast along edges that are node independent. Hence, during each interval we must maximize the number of such node-independent edges. The key notion can be formulated in terms of finding maximum matchings in bipartite graphs.

4.2.3. An optimal maximum-matching algorithm

Some brief definitions are required before we state our algorithm in terms of maximum matching. A graph $G[V = (X, Y), E]$ is called bipartite if $V = X \cup Y, X \cap Y = 0$ and each edge has one end vertex in X and one in Y. Given an arbitrary graph $G = [V, E]$ and a subset $S \subset V$, then a bipartite graph can be induced as follows. Let $R \subseteq V - S$ be the set of reachable nodes from S, i.e. $R = \{v | (u, v) \in E$ and $u \in S\}$ and E_s be $\{(u, v) | u \in S, v \in R\}$. Then $G_s[(S, R), E_s]$ is the bipartite graph induced by S. A matching in a bipartite graph is a set of edges such that no two edges in the set have a node in common. A maximum matching in a bipartite graph is such a set with maximum cardinality. If M is a matching between X and Y, then the image set of X under M will also be denoted as $M(X)$.

For any network G and an initial subset S, one can construct a sequence

$$V_0, M_1, V_1, M_2, \ldots, M_t, V_t$$

where $V_0 = S$, $V_t = V$ and t denotes the broadcast time. It can be easily shown that an optimal broadcast is one in which the sequence of matchings are maximum matchings.

Lemma 1. *If $V_1 \subseteq V_2$, then $b(G, V_2) \subseteq b(G, V_1)$.*

Proof. If $|V_1| \neq 1$, then an optimal broadcast originating at nodes in V_1 corresponds to a spanning forest F of G whose set of roots equals V_1. Thus $F = (T_{v1}, T_{v2}, \ldots, T_{vk})$where T_{vi} is a tree of the forest rooted at node v_i such that $\{v_1, v_2, \ldots, v_k\} = V_1$. Choose any node n such that $n \varepsilon V_2 - V_1$. The n necessarily belongs to a tree T_{vi} of F. Then we can

decompose T_{vi} into a pair $(T_{vi}^{\alpha}, T_{vi}^{\beta})$, where T_{vi}^{β} is the subtree rooted at n and $T_{vi}^{\alpha} = T_{vi} - T_{vi}^{\beta}$. One can easily observe that $b(T_{vi}^{\alpha}, v_i) < b(T_{vi}, v_i)$ and $b(T_{vi}^{\beta}, v) < b(T_{vi}, v_i)$. Hence, we obtain that $b(G, V_1) \geq b(G, V_1 \cup \{n\})$. By induction on the number of vertices in $V_2 - V_1$, it follows that $b(G, V_2) \subseteq b(G, V_1)$. □

4.2.4. Heuristic near-optimal scheme

It is not difficult to observe that in the sequence of matchings, there may be more than one maximum matching set in each stage of the induced bipartite graphs. To search for all possible maximum matchings requires a dynamic programming algorithm which is undesirable in networks of large sizes. Owing to this reason, it has been shown that the problem of finding an optimal broadcast time $b(G, u)$ for an arbitrary graph G and an arbitrary node u is NP-complete [22]. In order to overcome this, we present here a heuristic which leads to near-optimal time results. We consider constrained bipartite graphs, in which the nodes in the reachable set R we arranged according to their degrees. The process of matching nodes in S and R then considers those edges which have vertex in R with higher degrees[11]. This is a greedy approach that enables us to maximize the set of reachable nodes in each iteration of the algorithm. We will refer to this algorithm as largest degree maximum matching (LDMM).

4.3. Algorithm LDMM (largest degree maximum matching)

Step 1. $V_0 = S = \{u | u$ is a core gateway node$\}$;

$i = 0$;

Step 2. if $(V_i = V)$ then Halt.

Else form bipartite graph $G_s[(S, R), E_s]$; Order the nodes in R according to their degrees;

Step 3. $i = i + 1$;
Find a maximum matching M_i on G_s; $V_i = V_{i-1} \cup M_i$ (V_{i-1}); Go To Step 2.

In Step 3 of the above algorithm, we need an efficient method to find maximum matching in bipartite graphs. We implemented the simple labelling technique given in Ref. [18], modifying it to work on degree constrained bipartite graphs. For a good discussion on matchings in bipartite graphs and general graphs, the reader is referred to Lawler [18] and Syslo [19].

4.3.1. Modifications under high C.V of cluster link costs

The cost of the LDMM algorithm may be higher than the minimum given by a MCST algorithm. But in situations where all cluster links have the same cost, then cost(LDMM) = cost(MCST) and, hence, a time optimal algorithm is also cost optimal. Under situations where link costs differ significantly, i.e. $C.V$ is higher (connections may be FDDI or SMDS subnetworks rather than fractional T-1), then LDMM may deviate much from MCST values. Under such situations, we further modify our LDMM algorithm slightly such that during each step of matching, an edge is selected which maximizes the degree to cost ratio. We refer to this as the degree to cost maximum matching (DCMM) algorithm. Thus, DCMM not only tries to reach nodes with higher degrees, but gives priority to those links that reach them via minimum cost paths. Performance of both LDMM and DCMM under varying conditions are investigated using simulation in the next section.

4.3.2. Dealing with NP-completeness

Finding an optimal broadcast is an NP-complete problem. Our heuristics LDMM and DCMM achieve near-optimal solutions and compute broadcasts in polynomial time. All feasible broadcasts $B(G, V_0)$ can be represented by a state space tree in which the root of the tree stands for the message originator V_0, while the nodes at level i stand for subsets of V to which broadcast can be completed in i time units. An optimal broadcast is searching for the minimum length path in the state space tree. The state space tree has at the leaf-level the entire node set V of the cluster. Since there can be more than one maximum matching at each step of the algorithm (i.e. at each level of the tree), any path from the root to the leaf is a feasible solution. We have several paths from the root to the leaves and the path of minimum length is the optimal solution. By considering constrained bipartite graphs (listing nodes according to their degrees), both LDMM and DCMM are polynomial time algorithms since they eliminate the searching of several paths in the state space tree and are forced to take one path, that of largest degree nodes first (if two nodes have the same degree, then their eccentricity is compared and one which has larger value is chosen). Since broadcast is done along node-disjoint edges, we can actually maximize the number of nodes that are reached and hence obtain optimal or near-optimal broadcasting time.

4.4. Theoretical analysis: performance bounds

Lemma 2. *Let $\varphi(n)$ denote the worst-case computational complexity of the LDMM heuristic on any graph of n nodes. Then $\varphi(n)$ is bounded by $O[n^{5/2}]*log_2 n)$.*

Proof. To compute the degree of the vertices in the set R (step 2) and rank them in an order requires at most $O(|R|.|V|)$. The worst case for $|R|$ is $|R| = |V| = n$. The computation of the matching algorithm (using augmenting path labelling technique) inside each bipartitite graph of m nodes is bounded by $O(m^{5/2})$, i.e. bounded by $O(\{|S| + |R|\}^{5/2})$. So the complexity of step 2 and 3 is $O(|V|^{5/2})$.

[11] In case two nodes have the same degree, the node with higher reachability (eccentricity) is ranked higher. If reachability is also identical, the lower link cost neighbor is ranked higher, else a random selection is made.

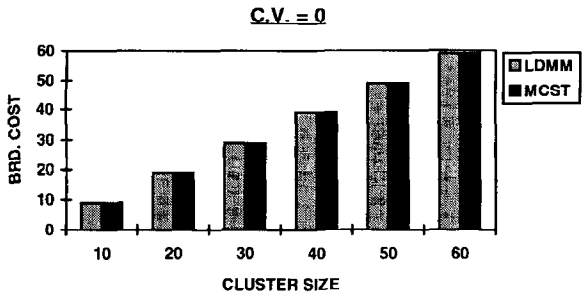


Fig. 5. Cost peformance of LDMM versus MCST.

Completely connected graphs K_n represent a case with maximum number of edges and a heavy computational demand to form the bipartite subgraph and perform maximum matching. For K_n step 3 has a complexity of $O(|n|^{5/2})$ and the maximum number of iterations is bounded by $O(\log_2 n)$. For completely connected graphs, the matching $M_i(V_{i-1})$ will have the same cardinality as that of V_{i-1}. This means that $|V_i|$ is equal to $2*|V_{i-1}|$ with possible exception only in the last iteration (if $n + 1$ is not a power of 2). Thus, the overall complexity of LDMM for K_n graphs is $O([n^{5/2}]*\log_2 n)$. □

Lemma 3. *Let G be a graph on n nodes and e edges with* $|e| \geq n - 1$. *Using the LDMM algorithm and with the ranking based on higher degrees (and higher eccentricity when degrees are equal), let b(G, u) be the broadcast time in a point-to-point network. Then*

1. *if G is* K_n, *u is any broadcast source, b(G, u) is optimal.*
2. *if G is a ring* R_n, *u is any broadcast source, b(G, u) is optimal.*
3. *if G is a star* S_n, *and u is either an edge node or the central node, then b(G, u) is optimal.*
4. *if G is a path* P_n, *u is any broadcast source, then b(G, u) is optimal.*

Proof.

1. *For* K_n, *using the same argument above (in Lemma 2),* $|V_i|$ *is equal to* $2 * |V_{i-1}|$ *which yields that the number of iterations and, hence, the broadcast time is* $O(\log_2 n)$, *which is optimal.*
2. *For a ring,* $b(G, u) = \lceil n/2 \rceil$ *assuming u is an endpoint. The algorithm accomplishes this in optimal time.*
3. *For a star, b(G, u) is n − 1 regardless whether u is the center or not. The algorithm gives this optimal time.*

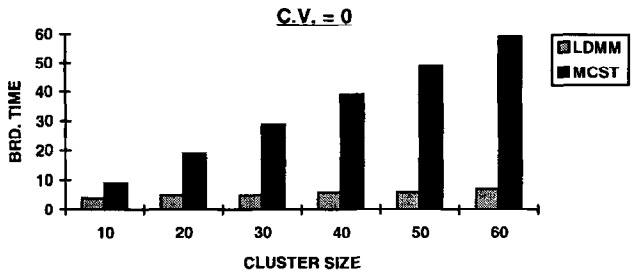


Fig. 6. Time performance of LDMM versus MCST.

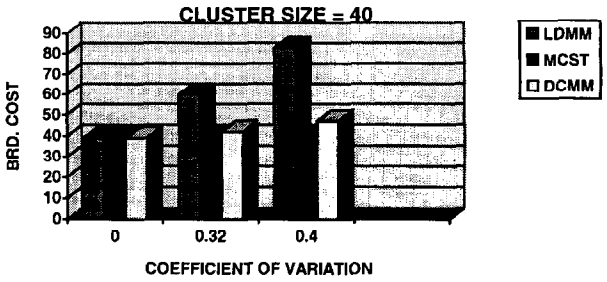


Fig. 7. Cost performance of the three schemes for various *C.V.*

4. *For a path,* $b(G, u) = n - 1$, *assuming u is an endpoint. This is optimal.*

5. Simulation results

In this section, we present a simulation study of the performance of our proposed schemes compared with the conventional policies based on minimum cost procedures. The model assumes 10 core gateways each connected to clusters of varying sizes. Each node within a cluster is an EGP router which is responsible for its autonomous system. An autonomous system may have on an average 20 networks under its administrative authority with each network having at least 10 nodes. Then an internet with average cluster size of 60 EGP nodes and 10 cores has effectively about 120 000 nodes. Our definition of large WANs is any network with more than 10 000 nodes.

The topology of each cluster is created randomly. The links we assumed to have randomly generated variable cost. If we denote the mean cost of a link by c, the second moment by $E[c^2]$, the variance by $var = E[c^2] - (c)^2$ and the standard deviation by $\sigma = \sqrt{var}$, then the coefficient of variation is given by $C.V = \sigma/c$.

The cost of the entire network using our hierarchical scheme is interpreted as follows:

$$C(G) = C(\text{DMST}) + \Sigma_{i=1}^{n} \{C(\text{LDMM}_i) \text{ or } C(\text{DCMM}_i)\} \tag{2}$$

where n is the number of clusters. The corresponding values for span time are derived as follows:

$$B(G) = b(\text{DMST}) + \max_{1 \leq i \leq n} \{b(\text{LDMM}_i) \text{ or } b(\text{DCMM}_i)\} \tag{3}$$

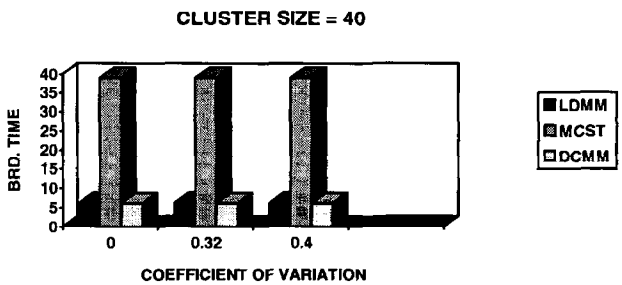


Fig. 8. Time performance of the three schemes for various *C.V.*

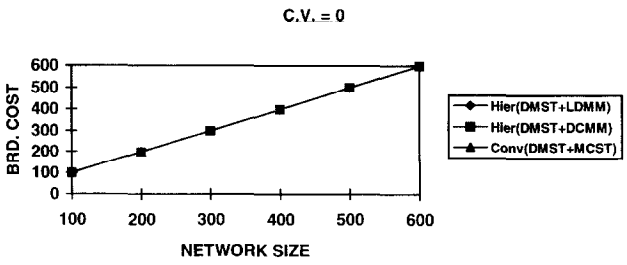


Fig. 9. Cost performance of hierarchical versus conventional scheme with $C.V = 0$.

Figs. 5 and 6 show the cost and time performance of the LDMM scheme compared with the conventional minimum cost spanning tree broadcast scheme. For clusters with same link costs (i.e. $C.V = 0$), the LDMM scheme yield optimal cost values (Fig. 5). But note that Fig. 6 demonstrates extreme improvement in broadcast time (optimal time values) performed by LDMM compared with MCST. Thus, in situations where cluster links are all operating at standard leased speeds of 56 kbps or T-1 rates, our LDMM scheme yield both optimal cost and time values.

Realistically, the link cost inside clusters are likely to vary since these are leased lines chosen by independent organizations trying to connect to the Internet. Thus, it is interesting to investigate the performance of LDMM in situations when $C.V$ is not zero. The cost and time performance for varying $C.V$ are shown in Figs. 7 and 8. Note that LDMM costs begin to deviate from the optimal (given by MCST) as the $C.V$ becomes higher. The deviation is greater for high $C.V$. But under such situations, the DCMM algorithm performs much better yielding near-optimal cost values. Note that the DCMM scheme has been designed to reach more node independent edges via least cost paths. It is likely that the $C.V$ for a cluster will not be very high (less than 0.4) and under such situations, the DCMM performs quite satisfactorily. Note from Fig. 8, that both LDMM and DCMM give best time values and show drastic improvement over MCST broadcast times.

In Figs. 9 and 10, we compare the performance of our hierarchical scheme (using DMST at the backbone and either LDMM or DCMM within the cluster) with those of existing spanning tree (forward or reverse) based schemes. For clusters with $C.V = 0$, both hierarchical schemes yield optimal cost values. The hierarchical scheme definitely yields much time improvement (Fig. 10) over the conventional schemes.

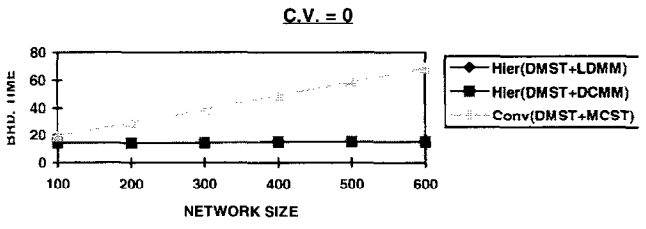


Fig. 10. Time performance of hierarchical versus conventional scheme with $C.V = 0$.

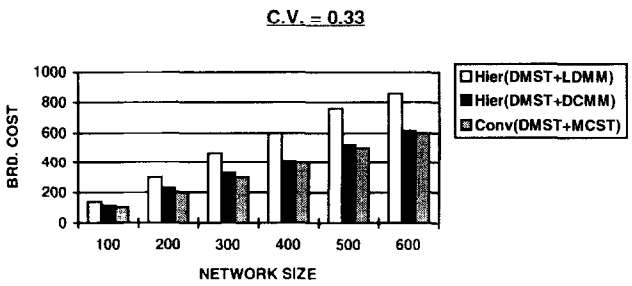


Fig. 11. Cost performance of hierarchical versus conventional scheme with $C.V = 0.33$.

For networks with $C.V$ not equal to zero, the performance of the hierarchical scheme versus the conventional is shown in Figs. 11 and 12. For an average $C.V = 0.33$, the hierarchical scheme using DCMM yield near-optimal cost values. Both schemes using LDMM or DCMM always yield much better span time values than those of conventional schemes.

Based on the results shown above, we conclude that LDMM can be effectively used for obtaining near optimal cost and time broadcast when $C.V = 0$ in the clusters. But under higher $C.V$, the DCMM scheme performs much better. Fig. 13 shows a three-dimensional plot of the behavior of DCMM with varying cluster size and $C.V$.

6. Conclusions

Broadcasting is a critical process that is used in a variety of situations within the Internet. This research work was motivated by finding a scheme that can scale well with the growth of the current cyberspace, can lead to efficient cost and time measures, and can withstand reliability problems of link failure. Having discussed the evolving architecture of the Internet, we proposed to use a hierarchical broadcasting technique. Two levels of hierarchy were pointed out: a backbone cluster where the DMST algorithm can be used and the local clusters (containing EGP routers of autonomous systems) where the largest degree maximum matching (LDMM) and the degree cost maximum matching (DCMM) have been proposed to yield optimal or near-optimal broadcast time and cost measures.

The hierarchical scheme proposed here is split

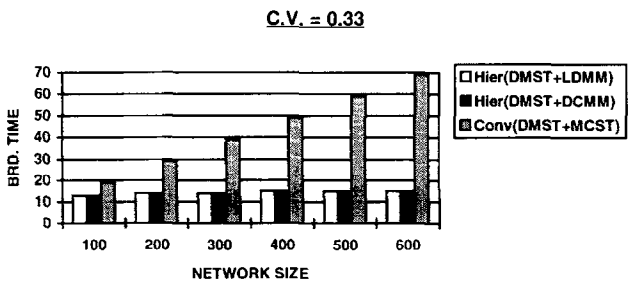


Fig. 12. Time performance of hierarchical versus conventional scheme with $C.V = 0.33$.

Table 1
Comparison of features of various broadcast techniques

Heuristic	Storage requirements	Distributed algorithm	Cost optimal	Time optimal	Static	Dynamic	Scalable	Robustness
New Arpanet LSP distribution	High	Yes	No	No		LSP gathering is dynamic	Yes	Yes
Pure flooding	Low		No	No	Yes		No	No
Spanning tree forwarding	High		Yes	No	Yes		No	No
Reverse path forwarding	Low		Yes	No	Yes		No	No
Hier (DMST + LDMM)	Medium	Partial	No	Yes		Yes	Yes	Yes
Hier (DMST + DCMM)	Medium	Partial	Yes	Yes		Yes	Yes	Yes

distributed. By that we mean the DMST algorithm within the backbone is entirely distributed in nature thereby improving scalability to a great extent. But the local schemes (both LDMM or DCMM) is essentially a centralized algorithm where the entire topology of clusters must be maintained inside a core gateway. While a distributed technique to find maximum matchings in bipartite graphs is non-trivial, it might be a worthwhile in the future to make the entire hierarchical scheme truly distributed.

The significant features and advantages that our scheme have over other proposed schemes for broadcasting in very large internetworks is listed in Table 1. Hierarchal modeling leads to significant memory savings, as a message need not be sent to every router, but only to the nearest higher level router. The bandwidth consumed in our scheme within each cluster hierarchy is reasonable and comparable to other schemes in use. Note that a link status change in the level 1 cluster generates a single message to the core to update the topology map database. If a link changes within the backbone, it causes a single LSP to be transmitted throughout the network which then initiates a new DMST computation. Between the time a topology change occurs and the time all the routers have reconstructed the new database, broadcasting and hence routing in the internet is disrupted, from minimally to severely, depending on the topological change. It is, thus, vital to keep the network in the 'converged' state as high a percentage of the time as possible. It has been proven that link state-based schemes (and especially hierarchical schemes) will converge more quickly than any others.

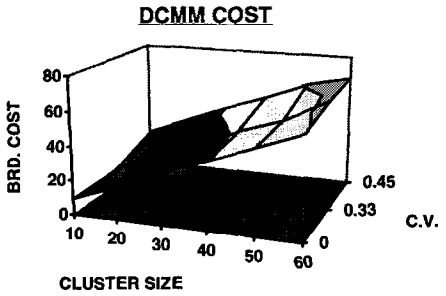


Fig. 13. Cost performance of DCMM for various *C.V* and cluster sizes.

With the tremendous growth and commercial success of the Internet in recent years, we are likely to see the increase in new services and applications. Broadcasting remains a fundamental part of the Internet routing and efficient and scalable techniques are thus desirable. As future work, we hope to look at DCMM more closely and possibly give weight to the degree and cost ratios. Extending beyond two levels of hierarchy to get even better performance might be interesting to investigate.

Acknowledgements

This work has been partially supported by a research grant from the College of Business Administration at Georgia State University. It has received additional funding from the Army Research Office under Grant No. DAAH04-95-1-0250 and was supported by an earlier grant from the Army's Simulation, Training and Instrumentation Command (STRICOM). The views and conclusions herein are those of the authors and do not represent the official policies of the funding agencies, Georgia State University or the University of Central Florida.

References

[1] J.M. McQuillan, I. Richer, E.C. Rosen, The new routing algorithm for the Arpanet, IEEE Transactions on Communications, COM28, May 1980.

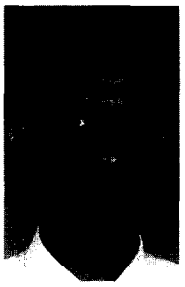
[2] A.M. Farley, Minimal broadcast networks, Networks 9 (1979) 313–333.

[3] K.B. Kumar, J.M. Jaffe, Routing to multiple destinations in computer networks, IEEE Transactions on Communications, COM31, March 1983.

[4] P. Scheuermann, G. Wu, Heuristic algorithms for broadcasting in point-to-point computer networks, IEEE Transactions on Computers, C-33(9), Sept. 1984.

[5] P. Baran, On distributed communication networks, IEEE Transactions on Communication Systems 12(3) (1964).
[6] M. Schwartz, T.E. Stern, Routing techniques used in communication networks, IEEE Transactions on Communications 28 (4) (1980) 265–279.
[7] D.E. Sproule, F. Mellor, Routing, flow and congestion control in the Datapac network, IEEE Transactions on Communications 28 (4) (1981) 386–391.
[8] W.T. Tsai, C.V. Ramamoorthy, W.K. Tsai, O. Nishiguchi, An adaptive hierarchical routing protocol, IEEE Transactions on Computers, 38(8), Aug. 1989.
[9] A. Khanna, J. Zinky, The revised Arpanet routing metric, in: Proceedings of SIGCOMM'89, ACM, 1989, pp. 45–56.
[10] P.F. Tsuchiya, Internet routing over large public data networks using shortcuts, in: Proceedings of SIGCOMM'92, 1992, p. 65.
[11] D. Estrin, Y. Rekhter, S. Hotz, Scalable inter-domain routing architecture, in: Proceedings of SIGCOMM'92, 1992, p. 40.
[12] S. Deering, Multicast routing in Internetworks and extended LANs, in: Proceedings of SIGCOMM'88, Stanford, CA, Aug. 1988.
[13] Y. Dalal, R. Metcalfe, Reverse path forwarding of broadcast packets, Communications of the ACM 21 (1978) 1040–1048.
[14] A. Gopal, I. Gopal, S. Kutten, Hardware flooding, in: Proceedings of SIGCOMM'91, Zurich, Switzerland, Sept. 1991.
[15] N.F. Maxemchuk, Dispersity routing on ATM networks, in: Proceedings of IEEE INFOCOM, 1993, pp. 347–357.
[16] F. Kamoun, L. Klienrock, Stochastic performance evaluation of hierarchical routing for large networks, Computer Networks 3 (1979) 337–353.
[17] R.G. Gallager, P.A. Humblet, P.M. Spira, A distributed algorithm for minimum weight spanning, ACM Transactions on Programming Languages and Systems 5 (1) (1983) 66–77.
[18] E.L. Lawler, Combinatorial Optimizations: Networks and Matroids, Holt, Rienhart and Winston, New York, 1976.
[19] M. Syslo, N. Deo, J. Kowalik, Discrete Optimization Algorithms with Pascal Programs, Prentice Hall, Englewood Cliffs, NJ, 1983.
[20] S. Chatterjee, M.A. Bassiouni, Hierarchical message dissemination in very large WANs, in: Proceedings of 17th IEEE Local Computer Network Conference, Minneapolis, MN, Sept. 1992.
[21] T.N. Saadawi, M.H. Ammar, A. El Hakeeem, Fundamentals of Telecommunication Networks, Wiley, New York, 1994.
[22] P.J. Slater, E. Cockayne, S.T. Heditniemi, Information dissemination in trees, SIAM Journal of Computing 10 (4) (1981) 692–701.
[23] D.E. Comer, Internetworking with TCP/IP: Principles, Protocols and Architecture, 3rd ed., vol. 1, Prentice Hall, Englewood Cliffs, NJ, 1995.
[24] R. Perlnan, Interconnections: Bridges and Routers, Addison-Wesley, Reading, MA, 1992.

Samir Chatterjee received the B.E. degree (Hons.) in Electronics and Telecommunications Engineering from Jadavpur University, India in 1988 and MS and Ph.D. in computer science from the University of Central Florida in 1991 and 1994. Since then, he has been an Assistant Professor with the Computer Information Systems Department at Georgia State University in Atlanta. He has published numerous technical papers in the areas of computer network architecture and protocols, residential broadband services, ATM networks, performance modeling of CATV systems and gigabit networking. He is affiliated to the broadband Telecommunications Center where he is looking into economic issues of offering broadband services to the home. His research has been funded by Georgia Center for advanced Telecommunications Technology and BellSouth Corporation.



Mostafa A. Bassiouni received his bachelor and master degree in computer science from Alexandria University in 1974 and 1977, respectively and the Ph.D. degree in computer science from Pennsylvania State University in 1982. Since then, he has been with the University of Central Florida where he is currently a Professor of Computer Science and Co-Director of the Distributed Computing and Networking Laboratory. He has published some 110 papers in various computer science journals, book chapters, and refereed conference proceedings. His research areas of interest include computer networks, scalable distributed protocols, distributed interactive simulation, data compression, temporal databases, concurrency control algorithms, and performance evaluation of computer systems. His work has been supported by several research grants/awards from industry and federal agencies. Funding agencies include: DoD/ARO, Army's PMTRADE, FHTIC, DoD/ARPA, NSF, Army's STRICOM, and CBIS.