Scalable and Efficient Broadcasting Algorithms for Very Large Internetworks*

Samir Chatterjee
Computer Information Systems Department
Georgia State University
Mostafa A. Bassiouni
Computer Science Department
University of Central Florida.

Abstract*

Most dynamic broadcasting techniques for large networks including the Internet suffer from *scalability* problems, i.e., they are not efficient with respect to the tremendous size of today's networks. We propose a distributed scalable broadcast algorithm for the Internet. Moreover our scheme attempts to optimize both *cost of routing* and *broadcast time* performance measures. Utilizing the hierarchical structure of the Internet, cost optimal protocols are proposed using a distributed *spanning tree* algorithm while near-optimal time broadcast can be achieved by formulating the problem as finding *maximum matching* in *bipartite graphs*. Simulation results obtained prove the validity and feasibility of the scheme.

1 Introduction

A. *Motivation:* The routing of packets to their intended destination forms a fundamental service provided by any packet switched network. Broadcast is the task of delivering a single message from a particular source node to all the other nodes in the network. While unicast (one-to-one) or multicast (one-to-many) communications may arise more often in a computer network, with the rapid explosion in the size of today's Internet and similar other proprietary corporate networks, broadcasting is increasingly becoming a tool to disseminate reachability information amongst a core set of computers.

The Internet uses broadcasting at its backbone, a set of core gateways that maintain route information for all destination subnetworks in the Internet. These core gateways use the GGP protocol (Gateway-to-Gateway Protocol) that propagates route information using Bellman-Ford vector-distance distributed algorithm [18]. Once a complete routing-table has been constructed, each core gateway share their information by *broadcasting* the route information to all other core gateways. In another class of algorithms, known as Shortest-Path-First (SPF) link-state, each core gateway tests the status of neighbor gateways and then periodically *broadcasts* a message that lists the status of each of its links. On receiving a broadcast link-status message, a gateway updates its map of the Internet and recomputes routes usually applying Dijkstra's shortest path algorithm to the resulting graph.

Broadcasting is used by search engines to determine the location of a resource and has been heavily used in recent applications like web-servers and distributed databases. MBONE servers and Domain Name System (DNS) servers require updated domain-name to IP address mappings stored inside large tables that are periodically shared using broadcasting. The dynamic changes to the Internet (addition and deletion of subnetworks) causes new information to be disseminated to all involved routers. Generally, it is desirable to broadcast packets over the best possible path available. While the simplest criteria chosen is to broadcast along paths that take the minimum amount of time to reach all destinations, minimizing broadcast delay is often quite difficult to achieve. Many algorithms use a link cost that is equal to the average delay experienced by packets traveling on the link (including queueing delays). It has been observed that minimizing both the broadcast delay and cost of routing is a difficult problem. We show that for arbitrarily large networks, it is possible to achieve near-optimal cost and time measures using a hierarchical broadcasting technique.

B. *Related Work*: Broadcasting techniques in conventional small sized networks have been studied extensively by several researchers. McQuillan et.al [1] , proposed the Arpanet routing technique based on *flooding*. Flooding can be wasteful of network bandwidth as the number of packets transmitted far exceeds the minimum required. Farley [2] proposes several *minimal broadcast* networks. Kumar et.al [3] addresses the *multidestination* routing problem. In [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]. 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

* This work has been partially supported by a research grant from the College of Business Administration at Georgia State.

0-7803-3250-4/96$5.00©1996 IEEE

copy of the packet is sent on all of the node's outgoing links (including links to any attached hosts) except the incoming link.

Some of the earliest work on routing in *distributed networks* can be found in Baran [5]. A good *survey* of routing schemes is presented in Schwartz [6]. A description of the use of variations of the distributed Arpanet routing algorithm is found in Sproule [7] and Tsai et.al [8]. The revision of the *delay metric* for the new Arpanet algorithm is found in Khanna [9]. Scalability issues have been addressed in [10, 11, 12].

2 System model

We model large internetworks as an undirected graph G = [V, L] in which set V contains "*n*" a large number of nodes (core and non-core gateways) while the set L is assumed to have at least *n-1* links or edges. The architecture of the present day Internet includes a core system of backbone gateways to which are attached many autonomous systems. Such systems accommodate administrative boundaries and includes a collection of networks and gateways managed by one administrative authority as shown in Fig. 1.

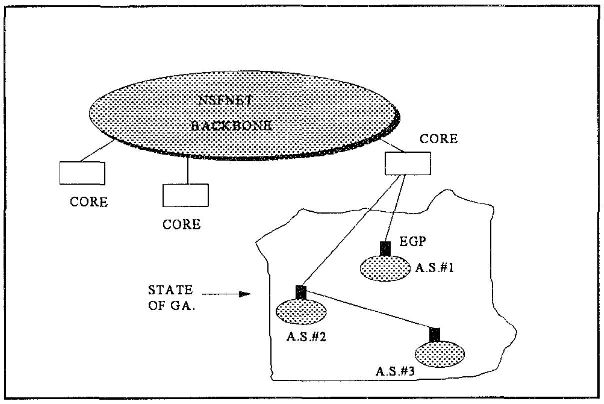


Figure 1: A hierarchical Internet architecture showing core gateways, autonomous systems and exterior gateways.

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 gateway (called an exterior gateway) to advertise its network reachability using a protocol such as EGP. The WAN that connects such exterior gateways do not support broadcasting and hence delivery is done 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 to complete. Since an individual site can have an arbitrarily complex structure, a core backbone system cannot attach directly to all networks. Hence exterior gateways running EGP connect to each other in an arbitrary manner using leased data circuits usually operating at fractional T-1 rates.

3 Overview of our broadcasting approach

We can now model the Internet as a hierarchical network organized into clusters that contain groups of nodes. Each EGP gateway is a cluster 0 node and many such nodes belonging to a state is grouped into cluster 1. At least one of the nodes within cluster 1 will have a leased connection to a backbone core gateway. All core gateways (those within NSFNET and Arpanet) belong to the higher level cluster 2.

The level 2 cluster is a collection of a 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[1]. If we assume that costs are interpreted as proportional to data rates used, then the backbone links (in NSFNET) will have varying speeds from T-1 to T-3 or higher depending on traffic patterns and loads. 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. In a dynamic changing internet, nodes, autonomous systems and clusters are likely to be added or deleted. While each core gateway (at cluster level 2) maintains a topology map of its connections to level 1 clusters, it only maintains neighbor information for its connections to the core gateways within cluster 2. This makes our approach highly scalable to future growth. So any addition/deletion of new clusters are only reflected in the topology. 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.

If one has to achieve a minimum cost broadcast within the core backbone, then 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 spanning tree algorithm at the backbone cluster of core gateways. We justify to optimize cost here (which may not lead to time optimal algorithm) as follows. Each core gateway is separated by a large distance. They are interconnected by links having a high degree of variability in cost. It is important that each core gateway route broadcast messages along the best possible path, which is measured here in terms of cost.

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 to 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

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

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 minimum-cost broadcast algorithm at the backbone core cluster and time-optimal routing within each level 1 cluster in the hierarchy.*

3.1 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. 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[2].

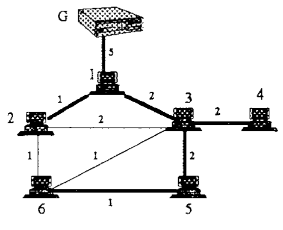
The *second phase* is the actual construction of a *minimum-cost spanning tree (MST).* We implemented a fully distributed MST algorithm as outlined in the work by Gallager et.al [15]. For a network of N nodes and E edges, the algorithm requires $5N\log_2 N + 2E$ messages to determine the MST. The algorithm 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 [15]. From now on, we refer to this algorithm as **DMST**[3].

The *third* and last phase is the *actual broadcast* that utilizes the MST links in order to route through minimum cost edges. This can be accomplished by using the classic *reverse path forwarding* technique proposed by Dalal [13].

3.2 Local cluster algorithms

The objective of cluster 1 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 gateway has received a copy of the message. It must now disseminate that message to all the EGP routers within the local cluster.

An Example: Shown in Figure 2 is an arbitrary cluster connected to a gateway 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 this 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[4] 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.*



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

Figure 2: An example cluster network and its time-optimal broadcast

The key concept: One 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. This key notion can be formulated in terms of finding *maximum matchings in bipartite graphs.*

The 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

[2] A 10-second period has been used as a basis of link performance in the revised ARPANET procedure

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

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

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 $\{V_0, M_1, V_1, M_2, \ldots\ldots, M_t, V_t\}$, 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. But in the above sequence, 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. Due 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 [17]. 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 are 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. This 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).

Algorithm LDMM(Largest Degree Maximum Matching)

```
Step 1. $V_0 = S = \{u | u \text{ is a core gateway node}\}$; $i = 0$;

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;   Go To Step 3.;

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.
```

Modifications under high Coefficient-of-Variation 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 (i.e., leased dedicated circuits of same speed 56 kbps), 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 subnetwork), 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 amd DCMM under varying conditions are investigated using simulation in the next section.

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. By formulating the problem in terms of maximum matchings in bipartite graphs, we end up in searching for the minimum length path in the state space tree. The state space tree has the source (core gateway) node as its root, and at the leaf level has 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[5], that of the largest degree node first. 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 Simulation results

In this section, we present a simulation study of the performance of our proposed schemes compared to 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 are 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:

$$C(G) = C(DMST) + \Sigma^{n}_{i=1} \{C(LDMM_i \text{ or } C(DCMM_i)\}$$

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

$$B(G) = b(DMST) + \max_{1 \le i \le n}\{b(LDMM_i \text{ or } DCMM_i)\}$$

[5] In case of two nodes with same degree, the lower link cost neighbor is selected. If the link costs are same, a random selection is made.

Figs. 3 and 4 show the cost and time performance of the LDMM scheme compared to the conventional cost-based broadcast scheme. For clusters with same link costs (i.e. C.V = 0), the LDMM scheme yield both optimal time and cost values. Fig. 4 demonstrates the large deviation in boadcast time by MCST compared to the optimal time values yielded by LDMM. 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. 5 and 6. 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. 6, that both LDMM and DCMM give best time values and show drastic improvement over MCST broadcast times.

In Figure 7 and 8, 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 schemes based only on minimum cost path broadcast. For clusters with C.V = 0, both hierarchical schemes yield optimal cost values. The hierarchical scheme definitely yields much time improvement (Fig. 8) over the conventional schemes.

For networks with C.V not equal to zero, the performance of the hierarchical scheme versus the conventional is shown in Figs. 9 and 10. 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.

Conclusions: This paper has proposed the use of a hierarchical broadcasting technique for large internetworks. 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.

References

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

[2] A. M. Farley, "Minimal broadcast networks", *Networks*, Vol. 9, pp. 313-333, 1979.

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

[4] P. Scheuermann, G. Wu, "Heuristic algorithms for broadcasting in point-to-point computer networks", *IEEE Trans. on Computers*, Vol. C-33, No. 9, Sept. 1984.

[5] P. Baran, "On distributed communication networks", *IEEE Trans. on Commu. Systems*, Vol. 12, No. 3, 1964.

[6] M. Schwartz, T. E. Stern, "Routing techniques used in communication networks", *IEEE Trans. on Communications*, Vol. 28, No. 4, pp. 265-278, 1980.

[7] D. E. Sproule, F. Mellor, "Routing, flow and congestion control in the Datapac network", *IEEE Trans. on Communications*", Vol. 28, No. 4, pp. 386-391, 1981.

[8] W. T. Tsai, C. V. Ramamoorthy, W. K. Tsai, O. Nishiguchi, "An adaptive hierarchical routing protocol", *IEEE Trans. on Computers*, Vol. 38, No. 8, Aug 1989.

[9] A. Khanna, J. Zinky, "The revised Arpanet routing metric", in *Proceedings of SIGCOMM'89*, ACM, pp. 45-56, 1989.

[10] P. F. Tsuchiya, "Internet routing over large public data networks using shortcuts", in *Proceedings of SIGCOMM'92*, pp. 65, 1992.

[11] D. Estrin, Y. Rekhter, S. Hotz, "Scalable inter-domain routing architecture", in *Proceedings of SIGCOMM'92*, pp. 40, 1992.

[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*, Vol. 21, pp. 1040-1048, 1978.

[14] F. Kamoun, L. Klienrock, "Stochastic performance evaluation of hierarchical routing for large networks", *Computer Networks*, Vol. 3, pp. 337-353, 1979.

[15] R. G. Gallager, P. A. Humblet, P. M. Spira, "A distributed algorithm for minimum weight spanning tree", *ACM Trans. on Programming Languages & Systems*, Vol. 5, No. 1, pp. 66-77, Jan 1983.

[16] T. N. Saadawi, M. H. Ammar, A. El Hakeem. Fundamentals of Telecommunication Networks. Wiley, 1994.

[17] P. J. Slater, E. Cockayne, S. T. Heditniemi, "Information dissemination in trees", SIAM Journal of Computing, Vol. 10, No. 4, pp. 692-701, Nov. 1981.

[18] D. E. Comer. Internetworking with TCP/IP: Principles, Protocols and Architecture. Volume 1, Third Edition, Prentice Hall, 1995.

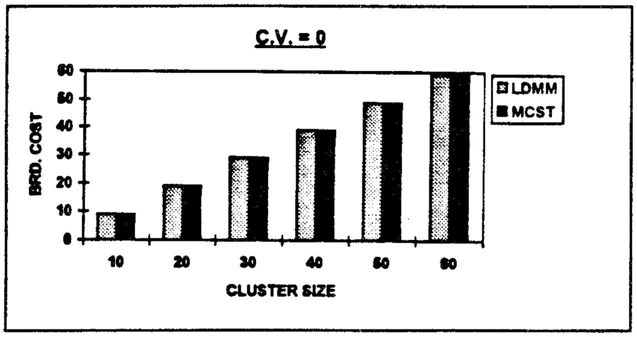


Figure 3: Cost performance of LDMM versus MCST

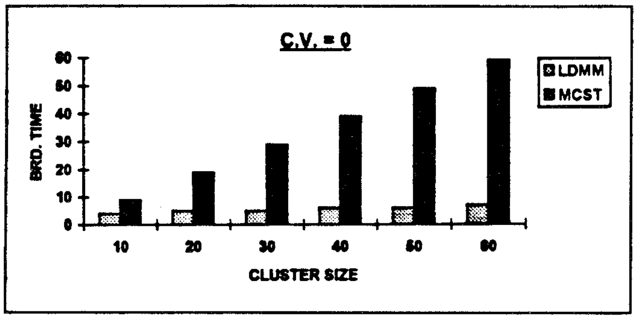


Figure 4: Time performance of LDMM versus MCST

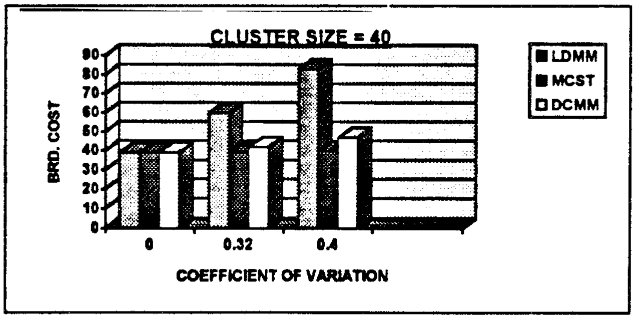


Figure 5: Cost performance of the three schemes for various C.V

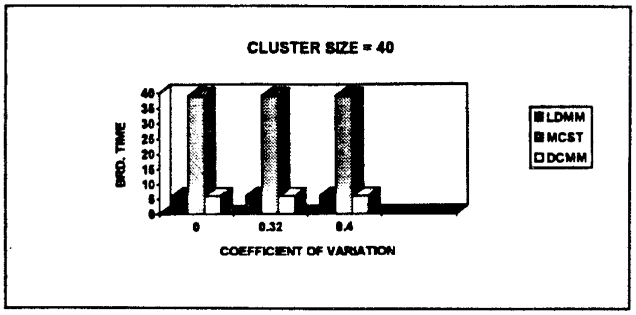


Figure 6:Time performance of the three schemes for various C.V.

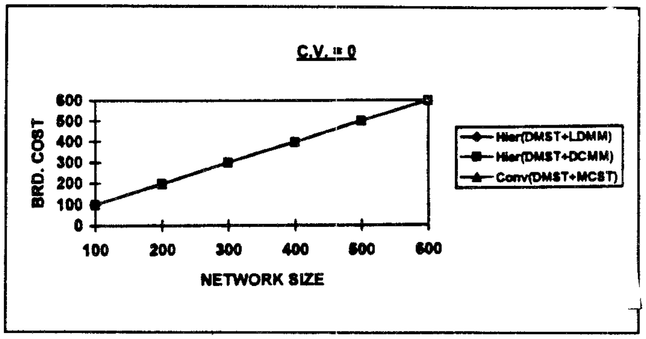


Figure 7: Cost performance of hierarchical versus conventional schemes with C.V. = 0

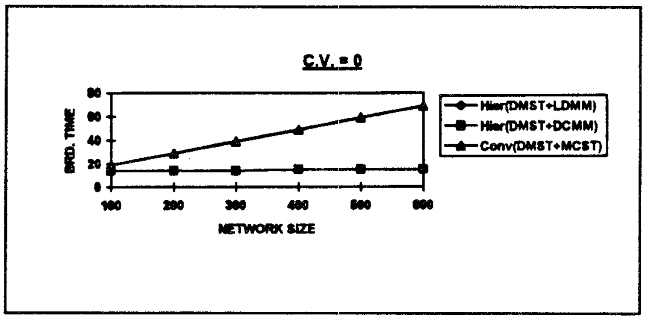


Figure 8: Time performance of hierarchical versus conventional scheme with C.V. = 0

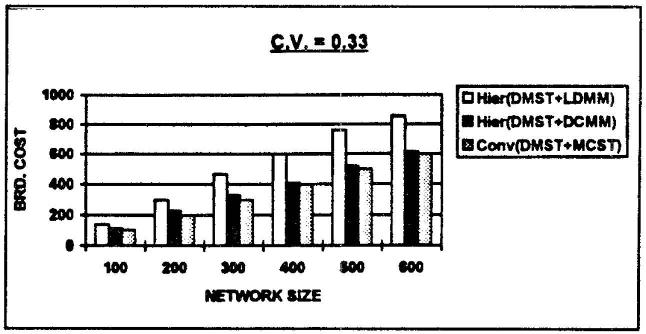


Figure 9: Cost performance of hierarchical versus conventional scheme with C.V. = 0.33

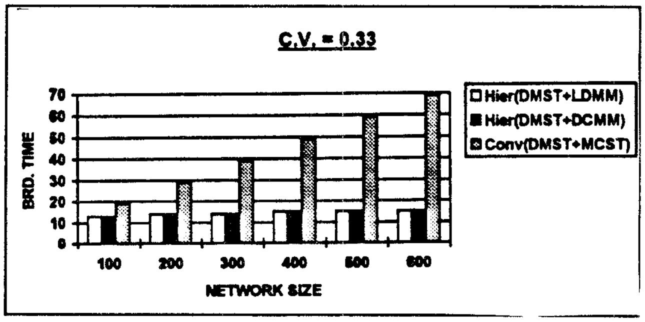


Figure 10: Time performance of hierarchical versus conventional scheme with C.V = 0.33.