This research has been supported in part by European Commission
FP6 IYTE-Wireless Project (Contract No: 017442)

Implementation of a Cluster Based Routing Protocol for Mobile Networks

Geoffrey Marshall[1] and Kayhan Erciyes[2]

[1] California State University San Marcos,
Computer Science Dept., 333 S.Twin Oaks Valley Rd.,
San Marcos, CA 92096, U.S.A.
[2] Izmir Institute of Technology
Computer Eng. Dept., Urla, Izmir 35430, Turkey
{marsh021,kerciyes}@csusm.edu

Abstract. We show the implementation and the simulation results of a hierarchical, cluster based routing protocol for mobile ad hoc networks using *Parallel Virtual Machine* (PVM). The network represented by a graph is partitioned into clusters by a graph partitioning algorithm and the shortest routes are first calculated locally in each cluster in the first srep. The simplified network which consists only of the nodes that have connections to other clusters called the neighbor nodes is then formed and the shortest routes are calculated for this simple network as the second step. A complete route between the two nodes of different clusters is formed by the union of intra-cluster and inter-cluster routes. We show the implementation results using PVM where a workstation represents a cluster and each node is a PVM process. The results obtained support the theoretical considerations where the efficiency increases by the number of clusters in use . . .

1 Introduction

Mobile ad hoc networks do not have central administration or fixed infrastructure and consist of mobile wireless nodes that have temporary interconnections to communicate over packet radios. As the topology of a mobile network changes dynamically, routes are needed to be calculated much more frequently than the wired networks. Various methods such as distributed, adaptive and self-stabilizing algorithms are used to perform routing in mobile networks. In *Link reversal routing* algorithms, a node reverses its incident links when it loses routes to the destination. Performance analysis of link reversal algorithms are given in [1] and TORA [9] is an example system that uses link reversal routing. Routing in mobile networks can be performed by clustering, that is, partitioning of the network into smaller subnetworks to limit the amount of routing information stored at individual nodes. In [8], a mobile network is partitioned into clusters of a two level graph. In the *zone routing* proposed in [5] where a zone functions similar to a cluster, the requested routes are first searched within the local zone. For inter-zone routes, the search is carried by multicast messages to the boundary nodes within the zones. In *k-way clustering*, the mobile network is divided into non-overlapping clusters where two nodes of a cluster are at most k hops away from each other. A k-way clustering method is proposed in [3] where the spanning tree of the network is constructed in the first phase and this tree is partitioned into subtrees with bounded diameters in the second phase.

In this study, we evaluate the performance of a hierarchical, two-level dynamic routing protocol described in [4] using PVM. The protocol consists of three main phases of partitioning the mobile network graph into clusters, calculating local cluster routes and finally calculating the simplified network graph routes. The rest of the paper is organized as follows. The background is given in Section 2, the analysis is discussed in Section 3, the PVM test results are given in Section 4 and the conclusions are outlined in Section 5.

2 Background

2.1 Partitioning of the Mobile Network

Graph partitioning algorithms aim at providing subgraphs such that the number of vertices in each partition is averaged and the number of edges cut between the partitions is minimum with a total minimum cost. An arbitrary network can be constructed as an undirected connected graph $G = (V, E, w)$ where V is the set of routing nodes, E is the set of edges giving the cost of communication between the routing nodes and w: $E \rightarrow \Re$ is the set of weights associated with edges. *Multilevel partitioning* is performed by coarsening, partitioning and uncoarsening phases [6]. During the coarsening phase, a set of smaller graphs are obtained from the initial graph. In the maximal matching, vertices which are not neighbors are searched. In Heaviest edge matching (HEM), the vertices are visited in random order, but the collapsing is performed with the vertex that has the heaviest weight edge with the chosen vertex. In Random Matching (RM) however, vertices are visited in random order and an adjacent vertice is chosen in random. The coarsest graph can then be partitioned and further refinements can be achieved by suitable algorithms like Kernighen and Lin [7]. Finally, the partition of the coarsest graph is iteratively reformed back to the original graph.

We provide a partitioning method called *Fixed Centered Partitioning* (FCP) [4] where several fixed centers are chosen and the graph is then coarsened around these fixed centers by collapsing the heaviest or random edges around them iteratively. Different than [6], FCP does not have a matching phase, therefore iterations are much faster. FCP requires the initial marking of the fixed centers. One possible solution is to choose the fixed centers randomly so that they are all at least some bounded distance from each other. The heuristic for the bound we used is $h = 2d \,/\, p$ where d is the diameter of the network and p is the number of partitions (clusters) to be formed. The time complexity of the total collapsing of FCP is $O(n)$. FCP provided much favorable partitions than CM and RM in terms of the average edge cost, time to partition a graph and the quality of the partitions experimentaly [4].

2.2 The Hierarchical Routing Protocol

The routing protocol called the *Neighbor Protocol* for the mobile network is not fully distributed due to the existence of some privileged nodes in the network. The distributed routing architecture consists of hierarchical clusters of routing nodes and each cluster has a controller which is called the *representative*. At the highest level, one of the representatives called the *coordinator*, receives messages to update its view everytime there is an addition or deletion of a node to a cluster. Upon such changes of configuration or periodically gathering of the changes, the coordinator starts a new configuration process by partitioning the network graph into new clusters. The nodes in the cluster that have connections to other clusters are called the *neighbor nodes*. The coordinator chooses one of the neighbor nodes in each cluster as the cluster representative and sends the cluster and neighbor topology information to the representative of such a group. Each representative then distributes the local connectivity information to all of the nodes in its group which concludes the first step of the protocol. In the second step, each node performs All-Pairs Shortest-Paths (APSP) routing within its cluster. At the end of this step, the distances between all pairs of nodes in the cluster including the neighbor nodes are calculated. In the third step, only the neighbor nodes calculate APSP routes for the simplified network graph which consists of neighbor nodes only. Any route is then formed by the union of the route from the source node to its nearest neighbor, the shortest route between the source neighbor and the destination neighbor and the shortest route between the destination neighbor and the destination node.

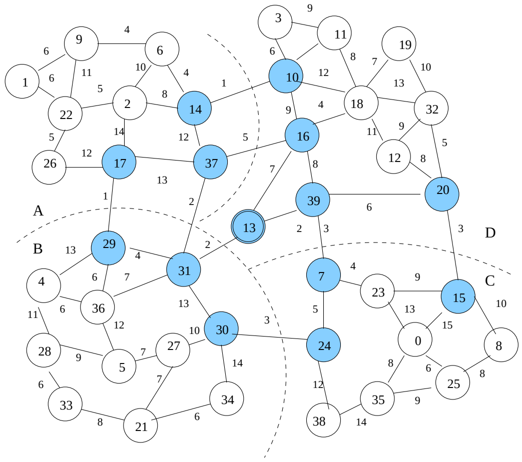

Fig. 1. The Original Network

2.3 An Example Network

An example network is depicted in Fig. 1. The initial centers allocated are 28, 8, 1 and 32. The coordinator and also the representative for cluster D is at 13. The coordinator partitions the graph using FCP as shown. Based on the partitioning information, the representatives chosen from the neighbors as 17, 30 and 15 are informed of their local connection. In the second phase, the representatives transfer this information to local nodes in their clusters in parallel. The ordinary nodes then calculate APSP in parallel, however, the neighbor nodes have to also calculate APSP for the simplified network graph which consists of the neighbor nodes only as shown in Fig. 3. Consider an example where node 26 in cluster A wants to send a message to the node 35 in cluster C. Since destination is not in its own cluster, 26 sends the message to its closest neighbor node, 17. Node 17 sends the message to node 7 which is its closest neighbor node in cluster C over 17-29-31-13-39-7. The neighbor node 7 routes the message to the destination over the shortest path which is 7-23-0-35. The total cost of this path is 49.

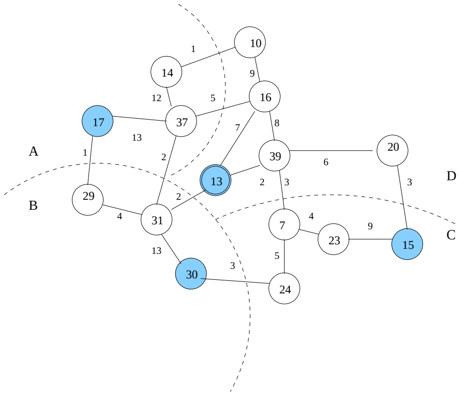

Fig. 2. The Simplified Neighbor Network

3 Analysis

The performance analysis should include the following

1. Partitioning of the network graph by FCP : O_1
2. Distribution of the cluster connectivity messages to the cluster representatives : O_2
3. Distribution of the routing information to the individual nodes by each representative : O_3
4. Intra-cluster route calculation time by the nodes within the cluster : O_4
5. Inter-neighbor route calculation by the neighbors : O_5

It was shown in [4] that the distribution of individual cluster routing information to the nodes (steps 2 and 3 above) take $O_{dist}(m)$ time where m is an upper bound on the number of nodes in a cluster. Also the total time required for intra-cluster and inter-neighbor routing algorithms is $O_{route}(m^3)$. The following theorem showed the Speedup obtained by the proposed protocol[4].

Theorem 1. *The Speedup obtained by the proposed protocol to a pure sequential all-to-all shortest paths protocol is $O(p^3)$ and to the parallel case where each node calculates all of the routes in parallel with others is $O(p^2/m)$.*

Proof. Total time for the protocol (O_{prot}) is :

$$O_{prot} = O_{part}(n) + O_{dist}(m) + O_{route}(m^3) = O(n + m^3) \tag{1}$$

and assuming a balanced partition, that is, $n = mp$

$$O_{prot} = O(n + m^3) = O(mp + m^3) \tag{2}$$

Assuming the network has p clusters and m nodes at each cluster, a serial algorithm to compute all routes of this network will take $O_{serial}((p * m)^3)$ operations. The speedup S that can be approximated with respect to pure serial case is :

$$S = O_{serial}/O_{prot} = O((p * m)^3/(mp + m^3)) \tag{3}$$

and assuming $m \gg p$

$$S = O(p^3) \tag{4}$$

For the pure parallel case where each node has all of the network connectivity information, $O_{par} = O_(p^2m^2)$ and the speedup now is :

$$S = O(p^2m^2/m^3) = O(p^2/m) \tag{5}$$

4 Experimental Results using PVM

Simulation of the network initialization and routing was performed using PVM. The simulation was performed for cluster sizes of 2, 4, 8, and 12 on a Beowulf cluster of PC's running Linux.

4.1 Initialization

To simulate the Neighbor Protocol, a central coordinator task is started to initialize the configuration of the network as follows:

1. Central coordinator task partitions graph into designated number of clusters.
2. From each cluster, the coordinator selects a neighbor node as representative and spawns a rep task on a new host.
3. Coordinator distributes local (cluster) connectivity information and neighbor connectivity to each rep task.

4. Each rep task spawns a neighbor task for each neighbor (not including itself) in its cluster on its host machine, and distributes local connectivity and neighbor connectivity to the neighbor.
5. The rep task spawns an ordinary task for each remaining non-neighbor node in the cluster and sends only local connectivity.

Every node, in parallel, calculates local routes using APSP. In addition, the neighbor tasks perform APSP for inter-cluster routes. Upon completing routing calculations, each node sends and ACK message to the representative of the cluster. Once the rep has collected ACK messages from all nodes in the cluster it sends ACK to the coordinator. When the coordinator has collected ACK's from all rep tasks, initialization is complete. The results for initialization of the network cluster configuration and routing calculations is shown in Fig. 3 for different size clusters. The times are significantly lower than the times for a non-distributed implementation that does not use the Neighbor Protocol wherein all nodes perform APSP for the entire graph as illustrated below. The measurements for normal APSP are about 5-10 times higher than the NP values and are not shown in graph.

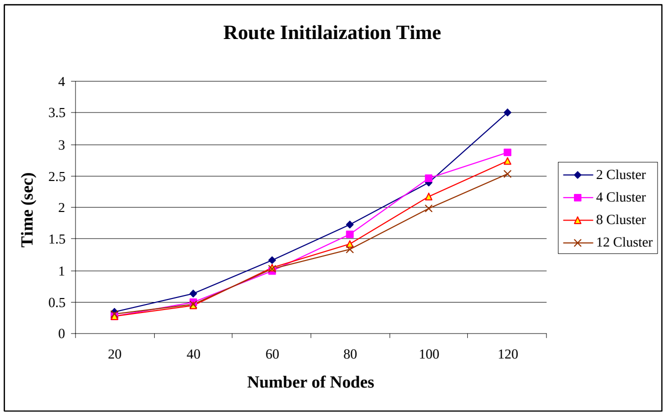

Fig. 3. Initialization Times for Clusters

4.2 Network Communication

Once a node has calculated routes it sends messages to randomly picked nodes in the network. Node u sends a message to node w by first creating a message and then looking up the next node v, in the route to w. For example, if ordinary node in cluster $d1$ wishes to send a message to a node in cluster $d2$, it would send the message to the next node in the shortest path to its closest neighbor, to be routed to cluster $d2$. When an incoming message is received, the node checks the header and either receives it or forwards it to the next node in route to destination. This procedure continues for each node to send one hundred messages and until all messages have been received. We measured communication time required for each node in the Neighbor Protocol to send one hundred messages and have them correctly routed and received. We found that run time is lower for more clusters providing less of a load per cluster as graph size increases.

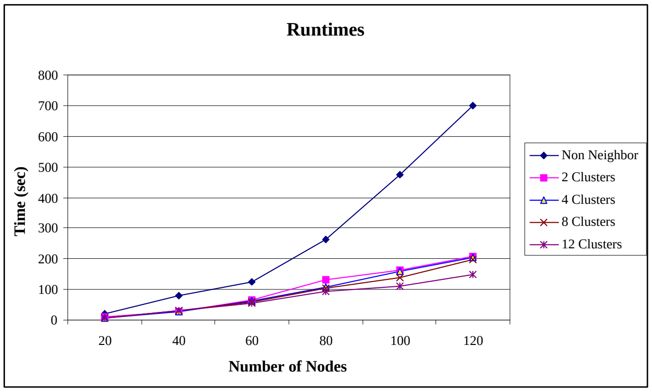

Fig. 4. Runtimes for Clusters

Fig. 4 shows that the execution time is roughly equivalent for smaller graphs of sizes 20 and 40 nodes for all cluster sizes but is improved for the larger graphs of 100 and 120 nodes for larger number of clusters such as 8 and 12. Again, the distributed Neighbor version substantially outperforms the single host implementation as shown.

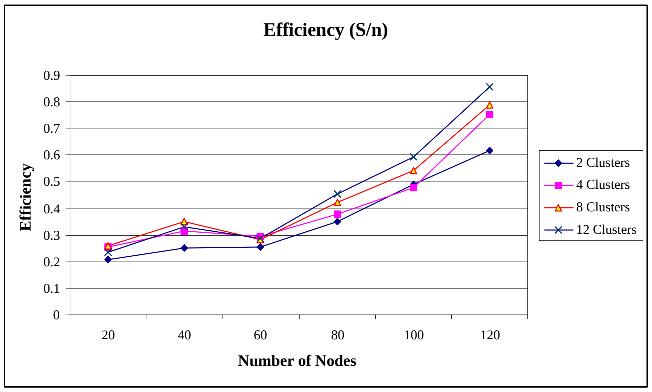

Fig. 5. Efficiency of the Neighbor Protocol

In Fig. 5, the efficiency curves for various cluster and node sizes are plotted where it can be seen that the efficiency rises as the number of clusters increases which is in accordance with the theoretical analysis (Section 3 and [4]).

5 Conclusions

We showed the simulation and further results of a proposed dynamic routing protocol initially described in [4] for a mobile network called the Neighbor Protocol using PVM. The protocol consists of three main steps of by firstly partitioning the mobile network graph, secondly delivery of the connectivity information of each cluster to the representative of the cluster which forwards this to individual nodes which calculate APSP routes within their clusters. In the final step, neighbor nodes calculate APSP routes for the simplified network. We showed that this approach improves performance considerably theoretically and the test results using PVM supported the theoretical analysis that the efficiency of the NP protocol rises as the number of clusters are inreased. The method we propose for routing in mobile networks provides *good* routes which are not necessarily the shortest paths but are comparable to shortest paths as shown by the tests. Further tests that exhibit the dymanic nature of the mobile networks to evaluate the performance of NP in terms of total control traffic against the frequency of route requests and frequency of movement in a mobile network using PVM are required. We are also looking into the fully distributed version of this protocol for mobile ad hoc networks for the case where there is no central coordinator but there are representatives and decisions on the partitioning of the graph and routing are done at the representative level by distributed agreement.

References

1. Busch, C. et al.: Analysis of Link Reversal Routing Algorithms for Mobile Ad hoc Networks. Proc. of the 15th ACM Symp. on Parallel Alg. and Arch. (2003) 210-219
2. E.W. Dijkstra: A Note on Two Problems in Connection with Graphs. Numerische Math., Vol. 1. (1959) 69-271
3. Fernandess,Y., Malki, D.: K-clustering in Wireless Ad hoc Networks, Proc. of the second ACM Int. Workshop on Principles of Mobile Computing, (2002) 31-37
4. Erciyes, K, Marshall, G, : A Cluster-based Hierarchical Routing Protocol for Mobile Networks, SV-LNCS, ICCSA(3) 2004, 528-537.
5. Z.J. Haas, M.R. Pearlman: The Zone Routing Protocol (ZRP) for Ad hoc Networks. Internet Draft, Internet Engineering Task Force. (1997)
6. Karypis, G., Kumar V.: Multilevel k-way Partitioning scheme for Irregular Graphs. Journal of Parallel and Distributed Computing, Vol. 48. (1998) 96-129
7. Kernighan, B., Lin, S.: An Effective Heuristic Procedure for Partitioning Graphs. The Bell System Technical Journal, (1970) 291-308
8. P. Krishna et al.: A Cluster-based Approach for Routing in Dynamic Networks. ACM SIGCOMM Comp. Comm. Rev., Vol. 27(2). (1997) 49 - 64
9. V.D. Park, M.S. Corson: A Highly Adaptive Distributed Routing Algorithm for Mobile Wireless Networks. Proc. IEEE INFOCOM, Vol. 3. (1997) 1405-1413