A STABILITY-BASED ON-DEMAND MULTICAST ROUTING IN AD-HOC WIRELESS NETWORKS

Krishna Paul*, S. Bandyopadhyay#, A. Mukherjee#, D. Saha+
* Cognizant Technology Solutions, Sector V, Saltlake, Calcutta 700 091, India
# PricewaterhouseCoopers Ltd., Sector V, Saltlake, Calcutta 700 091
+ Dept. of Computer Science, Jadavpur University, Calcutta 700 032, India

ABSTRACT

Multicast communication in the context of ad hoc wireless network is a very useful and efficient means of supporting group-oriented applications, where the need for one-to-many data dissemination is quite frequent in critical situations such as disaster recovery or battlefield scenarios. The objective of this paper is to propose and evaluate an on-demand multicasting routing mechanism for ad hoc wireless network. The mechanism is a generalization of our previously proposed stability-based unicast routing scheme that relies on determining link stability and path stability in order to find out a stable route from a source to a destination. The proposed multicast routing mechanism depends only on local state information (at source) for constructing a multicast tree. It is demand- driven in the sense that whenever a source needs to communicate with a set of destinations, it discovers the routes and creates a multicast tree dynamically. It has been shown that the proposed multicast routing scheme reduces both the control traffic and the data traffic and decreases the delivery delay considerably when compared to multiple unicasts.

1. INTRODUCTION

An ad hoc wireless network can be envisioned as a collection of mobile routers, each equipped with a wireless transceiver, which are free to move about arbitrarily. The mobility of the routers and the variability of other connecting factors results in a network with a potentially rapid and unpredictable changing topology. These networks may or may not be connected with the infrastructure such as internet, but still be available for use by a group of wireless mobile hosts that operates without any base-station or any centralized control. The basic assumption in an ad-hoc network is that two nodes willing to communicate may be outside the wireless transmission range of each other but still be able to communicate in multiple hops, if other intermediate nodes in the network are willing to forward packets from them [1, 2]. Applications of ad hoc networks include military tactical communication, emergency relief operations, commercial and educational use in remote areas, etc. where the networking is mission-oriented and / or community-based.

Multicast communication in this context is a very useful and efficient means of supporting group-oriented applications, where the need for one-to-many data dissemination is quite frequent in critical situations such as disaster recovery or battlefield scenarios [3]. Instead of sending data via multiple unicasts, multicasting reduces the communication costs by minimizing the link bandwidth consumption and delivery delay. This is especially important in the context of mobile, wireless environments where bandwidth is scarce and hosts have limited power.

However, the dynamics of ad-hoc wireless networks as a consequence of host mobility and disconnection of mobile hosts pose a number of problems in designing even unicast routing schemes for effective communication between any source and destination [2]. The conventional routing protocols that require to know the topology of the entire network is not suitable in such a highly dynamic environment, since the topology update information needs to be propagated frequently throughout the network [7,8]. On the other hand, a demand-based route discovery procedure generates large volume of control traffic. In a highly mobile environment with a large number of nodes, even if a route is discovered, a route rediscovery needs to be initiated when an intermediate node, participating in a communication between two nodes, moves out of range suddenly or switches itself off in between message transfer [1,2,4,5].

The objective of this paper is to propose and evaluate an on-demand multicasting routing mechanism for ad hoc wireless network. The mechanism is based on a stability-based unicast routing scheme[6] that relies on determining link stability and path stability in order to find out a stable route from a source to a destination. The proposed multicast routing mechanism depends only on local state information (at source) for constructing a multicast tree and is demand-driven in the sense that whenever a source needs to communicate with a set of destinations, it discovers the routes and creates a multicast tree dynamically. It has been shown that the proposed multicast routing reduces both the control traffic and the data traffic and decreases the delivery delay significantly.

2. RELATED WORK

Research in the area of routing in ad-hoc wireless network has mostly concentrated on designing effective routing schemes for unicast communication [1,2,4,5]. Those routing algorithms were not designed with multicast extensions in mind. Therefore, they do not naturally support multicast routing solutions. Since fixed network multicasting is based on state in routers (either hard or soft), it is fundamentally unsuitable for ad hoc network where topology is changing frequently due to unconstrained mobility [3]. It has been shown that the performance of both hard- and soft-state multicast tree maintenance mechanisms degrade rapidly with increased mobility. Traditional multicast approaches that rely on maintaining and exchanging multicast-related state information are not suitable in highly dynamic ad-hoc network with frequent and unpredictable changing topology [3].

Currently proposed ad hoc multicast routing schemes lie on a spectrum that spans from pure Internet multicast routing based schemes to a pure flooding scheme. Internet multicast routing schemes, as it is currently, generally require the routing nodes to maintain fairly large amount of state information for routing and to use processing power of hosts rather liberally. Feasibility of supporting continuous unlimited mobility is also a question with Internet routing schemes. Only flooding control packets may support unlimited continuous mobility. Flooding will also reduce the amount of state information kept at mobile hosts, and will provide reliable and timely delivery. FGMP [9], the Forwarding Group Multicast Protocol, proposes a scheme that is hybrid between flooding and source based tree multicast. AMRoute [10], the Ad hoc Multicast Routing Protocol, assumes the existence of an underlying unicast routing protocol and its performance is influenced by the characteristics of the unicast routing protocol being used. The Core- Assisted Mesh Protocol (CAMP) [11] generalizes the notion of core-based trees introduced for internet multicasting into multicast meshes that have much richer connectivity than trees.

Keeping in mind the dynamic nature of ad hoc network topology due to unconstrained mobility as well as the application domain of ad hoc network, we have developed a demand-driven multicast routing scheme with the following assumptions:

- The proposed multicast routing mechanism depends only on local state information at source for constructing a multicast tree dynamically and is demand- driven in the sense that whenever a source needs to communicate with a set of destinations belonging to a multicast group, it discovers the routes to the individual destinations and creates a multicast tree dynamically at source for that given group.
- Stability-based multicast routing scheme proposed here will ensure that the life-span of the multicast tree so formed will be sufficient to complete the required volume of data transfer at that instant of time.
- Each node knows its multicast group membership id(s). One node may belong to multiple multicast groups and it knows all the multicast group ids to which it belongs. Multicast group creation may be source-initiated where a source creates a multicast group-id and informs its members; or, it may be destination-initiated where a node, willing to become a member of a multicast group, collects its multicast group-id from any node belonging to the same multicast group. There is no global group-membership-management-protocol; the application domain of ad hoc network does not demand that.

3. A STABILITY-BASED FRAMEWORK FOR UNICAST ROUTING

In an ad hoc network, relationship among nodes is based on providing some kind of service, and stability can be defined as minimal interruption in that service. Most of the routing schemes discussed so far in the literature in the context of ad hoc network do not consider the user mobility pattern in order to find a more stable path rather than a shortest path. The idea of selecting stable routes within a dynamic network has been proposed in [12.13]. However, in these methods, stability is not explicitly evaluated. Hence, first we propose a notion of stability of a path and its evaluation mechanism in the context of ad-hoc network.

The network is modeled as a graph G = (N,L) where N is a finite set of nodes and L is a finite set of unidirectional links. Each node $n \in N$ is having a unique node identifier. Since in a wireless environment, transmission between two nodes does not necessarily work equally well in both direction [1], we assume unidirectional links. Thus, two nodes n and m are connected by two unidirectional links $l_{nm} \in L$ and $l_{mn} \in L$ such that n can send message to m via l_{nm} and m can send message to n via l_{mn}.

In a wireless environment, each node n has a wireless transmitter range. We define the neighbors of n, $N_n \in N$, to be the set of nodes within the transmission range R of n. It is assumed that when node n transmit a packet, it is broadcast to all of its neighbors in the set N_n. However, in the wireless environment, the strength of connection of all the members of N_n with respect to n are not uniform. For example, a node $m \in N_n$ in the periphery of the transmission range of n is weakly connected to n compared to a node $p \in N_n$ which is more closer to n. Thus, the chance of m going out of the transmission range of n due to outward mobility of either m or n is more than that of p.

Each link l_{nm} is associated with a signal strength S_{nm} which is a measurable indicator of the strength of connection from n to m at any instant of time. Due to the mobility of the nodes, the signal strengths associated with the links changes with time. When the signal strength S_{nm} associated with l_{nm} goes below a certain threshold S_t, we assume that the link l_{nm} is disconnected.

Affinity a_{nm}, associated with a link l_{nm}, is a prediction about the span of life of the link l_{nm} in a particular context. For simplicity, we assume a_{nm} to be equal to a_{mn} and the transmission range R for all the nodes are equal. To find out the affinity a_{nm} , node m samples the strength of signals received from node n periodically. Since the signal strength is roughly proportional to $1/R^2$, we can predict the current distance d at time t between n and m. If V is the average velocity of the nodes, the worst-case affinity a_{nm} at time t is (R-d)/V, assuming that at time t, the node m has started moving outwards with an average velocity V. For example, If the transmission range is 300 meters, the average velocity is 10m/sec and current distance between n and m is 100 meters, the life-span of connectivity between n and m (worst-case) is 20 seconds, assuming that the node m is moving away from n in a direction obtained by joining n and m..

Given any path p = (i, j, k, …, l, m), the **stability of path p** at a given instant of time will be determined by the lowest-affinity link (since that is the bottleneck for the path) and is defined as min[$a_{ij,}$ $a_{jk,}$ …, a_{lm}]. In other words, stability of path p between source s and destination d, $\eta^p_{sd,}$ is given by $\eta^p_{sd} = \min_{\forall i,j} a^p_{ij}$

However, the notion of stability of a path is dynamic and context-sensitive. As indicated earlier, stability of a path is the span of life of that path from a given instant of time. But stability has to be seen in the context of providing a service. A path between a source and destination would be stable if its span of life is sufficient to complete a required volume of data transfer from source to destination. Hence, a given path may be sufficiently stable to transfer a small volume of data between source and destination; but the same path may be unstable in a context where a large volume of data needs to be transferred. All these aspects needs to be considered in designing a routing scheme.

4. ON-DEMAND MULTICAST ROUTING

The mechanism for multicast routing is based on the stability-based unicast routing scheme described in [6]. The mechanism is an extension of this and comprises of four sequential steps. First, the source initiates a route discovery to get all the paths to individual destinations; next, it selects the stable routes from them and constructs a sub-graph connecting the source and the destinations; next, the source extracts multicast tree(s) from this sub-graph; finally, the source communicates the data to destinations using the multicast tree(s).

4.1 Path Finding Mechanism

A source initiates a route discovery request when it needs to send data to a set of destinations belonging to a multicast group. The source broadcast a route request packet to all neighboring nodes. Each route request packet contains source id, destination ids (or multicast group id), a request id, a route record to accumulate the sequence of hops through which the request is propagated during the route discovery, and a count n which is decremented at each hop as it propagates. When n=0, the search process terminates. The count n thus limits the number of intermediate nodes (hop-count) in a path in order to reduce unconstrained propagation of control packets. Through simulation, the optimum value of n has been estimated to be 5.

When any node receives a route request packet, it decrements n by 1 and performs the following steps:

1. If the node is one the destination nodes (i.e. belongs to the multicast group), a route reply packet is returned to the source along the selected route, as given in the route record which now contains the complete path information between source and that destination node. At the same time, the node id is appended to the route record in the route request packet and the request is re-broadcast in search of other destinations.
2. Otherwise, if n=0, the route request packet is discarded.
3. Otherwise, if this node id is already listed in the route record in the request, the route request packet is discarded (to avoid looping).
4. Otherwise, the node id is appended to the route record in the route request packet and the request is re-broadcast.

When any node receives a route reply packet, it performs the following steps :

1. If the node is the source node, it records the path from source to one of the destinations .
2. If it is an intermediate node, it appends the value of affinity and propagates to the next node listed in the route record to reach the source node.

4.2 Constructing Stable Sub-graph Connecting Source and Destinations

Let us assume that a source wants to send NUM number of packets to a set of destinations. The source initiates a route discovery request as described earlier and waits for the route reply packets from each destination belonging to the multicast group until timeout. For each path p, it computes its stability η^p . To evaluate whether the path is stable enough to complete the data transfer between s and d, the following condition needs to be checked:

$\eta^p_{sd} > NUM * t_p$,

where t_p is the average end-to-end delay per packet and depends on the bandwidth and number of hops. If this condition holds good, the path is selected. Thus, all the *stable* paths between source and destinations are selected. As a next step, a graph is constructed containing all those stable path. This is a *stable* sub-graph of the given network that contains the source, the set of destinations belonging to a multicast group and a set of intermediate nodes connecting them.

4.3 Constructing Multicast Tree at the Source

Once a stable sub-graph has been constructed at source, the source deploys a multi-directional search technique [14] to find out the connectivity among source node and destination nodes with a minimum set of intermediate nodes. Let M be the multicast group containing source node and destination nodes. Conceptually, the search technique is a multi-directional, step-by-step breadth-first search starting from all the nodes in M. It eventually finds out the connectivity among all the nodes in M through a minimal set of intermediate nodes. A distributed version of this algorithm is given in [14]; however, since the source contains the entire sub-graph, it employs a centralized version of this algorithm in order to find out the stable multicast tree at source.

The search technique spreads a set of activation tokens in the sub-graph stored in source, starting from all the nodes in M, in order to search for the connectivity among all the nodes in M through a minimal set of intermediate nodes. An activation token activates a node of the sub-graph. Each activated node modifies its internal state (initially null) and generates activation tokens further to activate all its neighbors. This continues until the search process is complete. The activation tokens carry the path information. Thus, when the completion of a search process is detected, the resultant token contains the total path information which gives the desired connectivity.

An activation token has two components: the first component gives the node-ids through which the token has already traveled and the second component indicates the target nodes in M, which are still being searched. As an example, let us assume that it is intended to find out the connectivity among nodes $\{x, y, z\} \in M$. Each of these nodes are activated with an activation token [( )(xyz)]. It indicates that the first component of the token is null i.e. it has not yet traveled through any node and the second component is (xyz) i.e. it is searching the connectivity among x, y and z. The output of x, which would activate all the neighbors of x, is [(x)(yz)]. It indicates that the token has already traveled through x in search of connectivity among y and z. The node x retains this information as its "current state" for subsequent use, as will be illustrated below. Similarly, the output of y would be [(y)(xz)] and that of z would be [(z)(xy)].

Every activated node manipulates the input token by the application of a unique search function f on its state and the input token. Thus, $O \leftarrow f(I,S)$ and $S \leftarrow O$, where, O is the output token, S is the state of the node under consideration, and I is the input token. Initially, S is null. The function extracts the commonality between the state and the input token to evaluate the extent of search. It also detects the termination in case the search is complete.

Before defining the function f, let us assume that the multicast group M contains $\{ x_1 x_2...x_n \}$ which denotes the source and set of destination nodes to be connected through minimum number of intermediate nodes. So, they are all activated, each with a token $[(\ )(x_1 x_2...x_n)]$. Let us also assume that A is a set of arbitrary intermediate nodes $\{a_1 a_2...a_m\}$ and B is a set of arbitrary intermediate nodes $\{b_1 b_2...b_p\}$. During the search process, suppose node c, an arbitrary intermediate node in the graph, receives an activation token $[(x_j A)(x_1 x_2 \ldots x_{j-1} x_{j+1} \ldots x_n)]$. It implies that the node c receives an input token which has traveled from x_j through $a_1 a_2...a_m$ in search of $x_1 x_2 \ldots x_{j-1} x_{j+1} \ldots x_n$.

The function f_c is defined as follows:

Case I.

When S_c, the state of the node c, is null, the output token O_c and the current state S_c will be the input token with the current node-id appended to it.

If $I_c = [(x_j A)(x_1 x_2 \ldots x_{j-1} x_{j+1} \ldots x_n)]$ and S_c is null, then

$O_c \leftarrow f(I_c, S_c) = [(cx_j A)(x_1 x_2 \ldots x_{j-1} x_{j+1} \ldots x_n)]$, and

$S_c \leftarrow O_c$.

If $c \in \{ x_1 x_2 \ldots x_{j-1} x_{j+1} \ldots x_n \}$, then that node id would be deleted from the second component of O_c.

Case II.

The connectivity between x_j and x_m is determined under the following condition:

If $I_c = [(x_j A)(x_1 x_2 \ldots x_{j-1} x_{j+1} \ldots x_n)]$ and $S_c = [(cx_m B)(x_1 x_2 \ldots x_{m-1} x_{m+1} \ldots x_n)]$ then

$O_c \leftarrow f(I,S) = [(cAB\ x_j\ x_m)(x_1 x_2 \ldots x_{j-1} x_{j+1} \ldots x_{m-1} x_{m+1} \ldots x_n)]$, and $S_c \leftarrow O_c$.

Thus, $(cAB\ x_j\ x_m)$ denotes connectivity between nodes $x_j\ x_m$. Proceeding in this manner, the connectivity among all members of M would be established.

Case III.

When the state contains more connectivity information than the input token or the same connectivity information as the input token, the activation is ignored. This is done to avoid looping.

Thus, if $S_c = [(cAB\ x_j\ x_m)(x_1 x_2 \ldots x_{j-1} x_{j+1} \ldots x_{m-1} x_{m+1} \ldots x_n)]$ and $I_c = [(x_j A')(x_1 x_2 \ldots x_{j-1} x_{j+1} \ldots x_n)]$, then the activation is ignored and S_c remains unchanged. The reason is that the connectivity between $x_j\ x_m$ is already established as $(cAB\ x_j\ x_m)$ and this token has already been spread out. So, there is no need to consider a new input token whose first component contains less information.

Case IV.

When the complete connectivity is known, the activation terminates.

If $S_c = [(c A' x_1 x_2 \ldots x_{n-1})(x_n)]$ and $I_c = [(A'' x_n)(x_1 x_2 \ldots x_{n-1})]$, where A′ and A″ are sets of intermediate nodes, then f (I,S) generate the required connectivity as $(c\ A'\ A''\ x_1 x_2 \ldots x_{n-1} x_n)$ and the state will be set to null.

Once the connectivity is known by the source, it traces back the sub-graph in order to find out the desired multicast tree.

5. PERFORMANCE EVALUATION

5.1 Simulation Set up

The performance of the proposed scheme is evaluated on a simulated environment under a variety of conditions. In the simulation, the environment is assumed to be a closed area of 1000 x 1000 unit in which mobile nodes are distributed randomly. We ran simulations for networks with 20 mobile hosts with range of transmission varying from 250 to 350 unit in each case. The bandwidth for transmitting data is 400 kbps.

In order to study the delay, throughput and other time-related parameters, every simulated action is associated with a simulated clock. The clock period (time-tick) is assumed to be one millisecond (simulated). For example, if the bandwidth is assumed to be 400 kbps and the volume of data to be transmitted from one node to its neighbor is 4 kb, it will be assumed that 10 time-ticks (10 millisecond) would be required to complete the task. The size of both control and data packets are same and one packet per time-tick will be transmitted from a source to its neighbors.

The pattern and speed of movement of individual node ranges from 10 units and 30 units per second. Each node starts from a home location, selects a random location as its destination and moves with a uniform, predetermined velocity towards the destination. Once it reaches the destination, it waits there for a pre-specified amount of time, selects randomly another location and moves towards that. However, in the present study, we have assumed zero waiting time to analyze worst-case scenario. Also, we have assumed that nodes are moving with a uniform velocity of 20 units / sec.

5.2 Simulation Results

5.2.1 Control Message Overhead

On-demand route discovery generates large number of control packets due to unrestricted propagation of route request packets throughout the network. It has been observed that [6] with increasing transmission range, since the network is more strongly connected (i.e. each node has more number of neighbors), the control traffic increases rapidly due to this unrestricted propagation of control packets over the network.

Figure 1 shows the average number of control packets generated for multiple unicast and multicast for different number of destination nodes in multicast group for different transmission range (R). The node mobility is assumed to be fixed at 20 units per second. As expected, multicast mechanism generates less number of control packets as compared to multiple unicast; the effect is more pronounced with increasing number of nodes in multicast group.

5.2.2 Multicast Efficiency

In ad-hoc network, data from source would reach the destination in multiple hops through intermediate node. In multiple unicast, the same data packets may need to travel through same set of intermediate nodes in order to reach different destinations. Using multicast, we intend to avoid this duplication of data traffic which, in turn, reduces the data traffic in the network. Multicast efficiency, M_E, is a measure of reduction in data traffic in the network and is defined as :

$$M_E = \frac{\sum \text{Number of intermediate nodes in all unicast routes}}{\text{Number of Intermediate nodes in Multicast Route}}$$

$M_E = 1$ indicates that there is no difference between multicast and multiple unicast. This would happen when all the unicast routes are disjoint i.e. there is no common node among the unicast routes.

Table 1 shows the average Multicast Efficiency for different number of nodes in multicast group (source + destinations). The node mobility is assumed to be uniform at 20 units per second. In all cases, multicast efficiency is more than one. The improvement is more significant at low transmission range. The reason is, at higher transmission range (350, for example), some source-destination pairs are directly connected (single-hop connectivity). However, at higher transmission range, since the network is more strongly connected (i.e. each node has more number of neighbors), the control traffic increases the congestion due to unrestricted propagation of control packets over the network. Subsequently, the delay in delivering the data packets to destinations increases. Hence, low transmission range is a preferred configuration, where the proposed multicast mechanism shows significant improvement.

5.2.3 Delay in Data Communication

Figure 2 shows the average time taken by a set of data packets (500 packets) to reach all destination nodes using a) multiple unicast and b) multicast routing schemes. We have selected different number of nodes in the multicast group and studied the data communication under a light background traffic load (at least another unicast communication is going on in the background with randomly selected source-destination pair). Since multicast reduces both control and data traffic, the delay would be substantially less compared to multiple unicast.

Table 1. Multicast Efficiency for different number of nodes in multicast group at two transmission ranges

Transmission range	No. of nodes in multicast group	Total no. of intermediate nodes in multiple unicast	Total no. of intermediate nodes in multicast	Multicast Efficiency
250	4	5	2	2.5
	6	9	4	2.25
	8	11	4	2.75
	10	15	6	2.5
350	4	2	1	2
	6	4	2	2
	8	6	4	1.5
	10	7	4	1.75

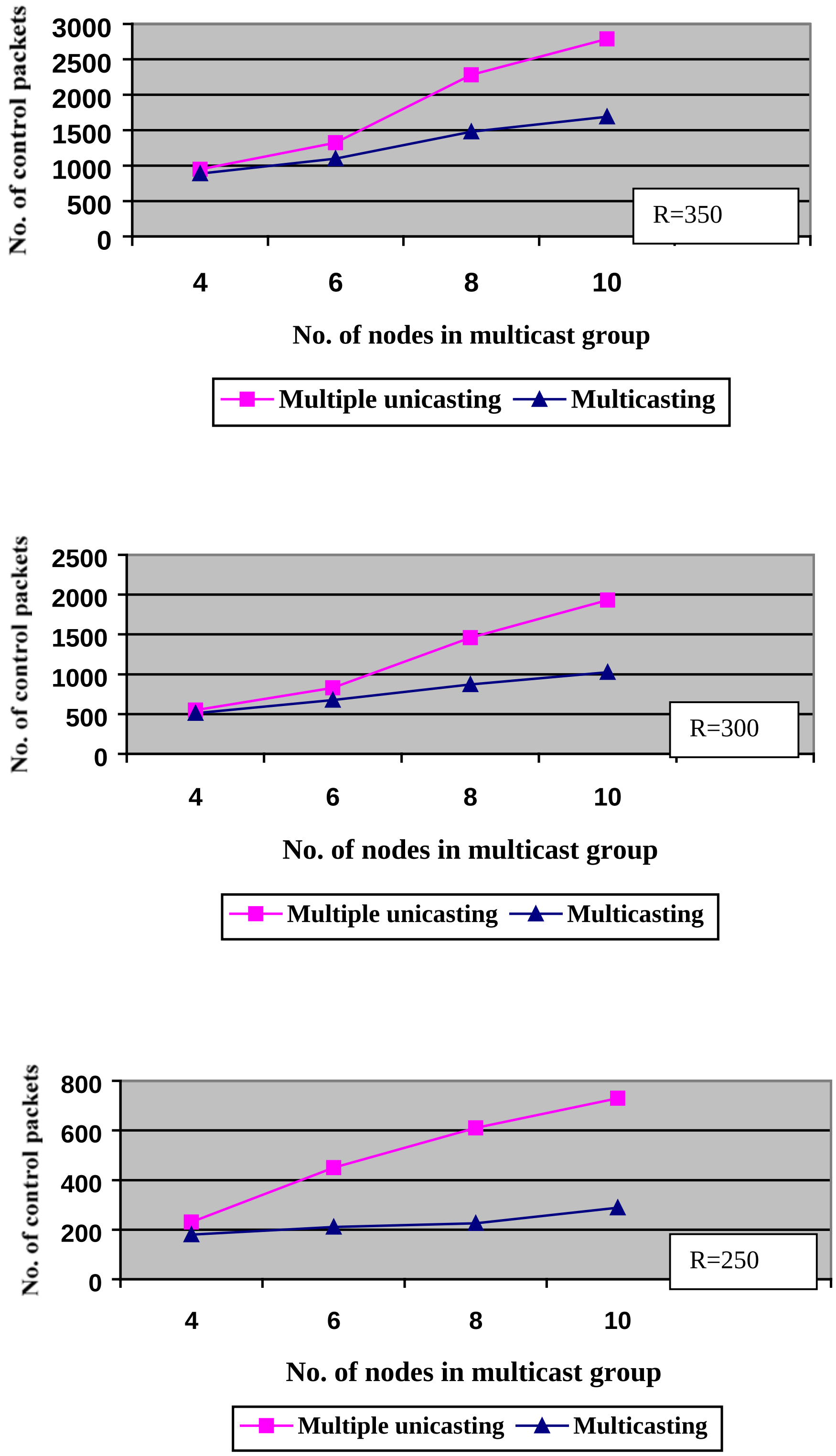

Figure 1. Number of control packets generated for different number of nodes in multicast group at different transmission range (R).

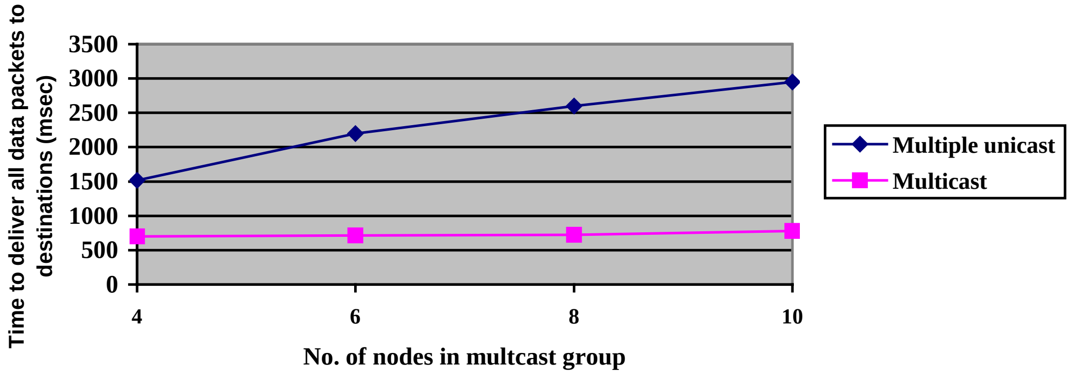

Figure 2. Time to deliver all data packets to destinations with multicast and multiple unicast

5. CONCLUSION

The proposed multicast routing mechanism is a generalized form of a stability-based unicast routing scheme that relies on determining link stability and path stability in order to find out a stable route from a source to a destination. It depends only on local state information (at source) for constructing a multicast tree and is demand-driven in the sense that whenever a source needs to communicate with a set of destinations, it discovers the routes and creates a multicast tree dynamically. It has been shown that the proposed multicast routing scheme reduces both the control traffic and the data traffic and decreases the delivery delay significantly.

REFERENCES

1. D. B. Johnson and D. Maltz, Dynamic source routing in ad hoc wireless networks, T. Imielinski and H. Korth, eds., *Mobile computing*, Kluwer Academic Publ. 1996.
2. S. Corson, J. Macker and S. Batsell, Architectural considerations for mobile mesh networking, Internet Draft RFC Version 2, May 1996.
3. K. Obraczka and G. Tsudik, Multicast Routing Issues in Ad Hoc Networks, Proc. of the IEEE ICUPC '98, October 1998.
4. Z.J.Haas, A new routing protocol for the reconfigurable wireless networks, ICUPC'97, San Diego, CA, Oct. 1997.
5. V. D. Park and M. S. Corson, A highly adaptive distributed routing algorithm for mobile wireless networks, Proc. IEEE INFOCOM '97, Kobe, Japan, April 1997.
6. K. Paul, S. Bandyopadhyay, D. Saha and A. Mukherjee, Communication-Aware Mobile Hosts in Ad-hoc Wireless Network, Proc. of the IEEE International Conference on Personal Wireless Communication, Jaipur, India, Feb. 1999.
7. C. Perkins and P. Bhagwat, Highly dynamic destination-sequenced distance-vector routing for mobile computers, Proc. of the ACM SIGCOMM, October 1994.
8. S. Murthy and J.J. Garcia-Luna-Aceves, An efficient routing protocol for wireless networks. *ACM Mobile Networks and Aplication Journal*, Special issue on Routing in Mobile Communication Networks, 1996.
9. Ching-Chuan Chiang, Mario Gerla, and Lixia Zhang, Forwarding Group Multicast Protocol(FGMP) for Multihop, Mobile Wireless Networks, ACM-Baltzer Journal of Cluster Computing: Special Issue on Mobile Computing, Vol. 1, No. 2, 1998
10. M.Liu, R.R.Talpade, A.McAuley and E.Bommaiah, AMRoute: Adhoc Multicast Routing Protocol, CSHCN TR 99-1, University of Maryland, USA.
11. J.J.Garcia-Luna-Aceves and E.Lmadruga, A Multicast Routing Protocol for Adhoc Networks, Proc. of the IEEE INFOCOM'99,New York, March 1999.
12. C-K Toh, A novel distributed routing protocol to support ad-hoc mobile computing, IEEE International Phoenix Conference on Computer & Communications (IPCCC'96).
13. R. Dube, C.D. Rais, K. Wang and S.K. Tripathi, Signal stability based adaptive routing for ad hoc mobile networks, Technical Report CS-TR-3646, UMIACS-TR-96-34, Institute for Advanced Computer Studies, Department of Computer Science, University of Maryland, College Park, MD 20742, USA, August, 1996.
14. S. Bandyopadhyay. An Architecture for Distributed Knowledge Processing", Journal of New Generation Computing Systems, Vol 4, no.1, 1991.