2011 International Conference on Computer Communication and Management
Proc .of CSIT vol.5 (2011) © (2011) IACSIT Press, Singapore

An Intelligent Way of Defining Routing Table and Path for MANET

Anisur Rahman[+], Ashik Anower

Dept. of Computer Science & Engineering, East West University, Dhaka, Bangladesh

Abstract. The knowledge about the topology can dramatically affect the performance of Mobile Ad Hoc Networks (MANET). In pro-active routing protocols, all the nodes consume enough power to build up the routing table before starting a communication. And the routing table is always updated so that the nodes have most up-to-date information to have a communication path. Unlike destination sequence distance vector (DSDV), where all the nodes of a path need to consume processing power to generate information about the next hop to broadcast, the proposed method eliminates the requirement of processing in every node. This paper also redefined the routing table that makes the path finding mechanism more efficient; which eventually decreased end-to-end delay and increased throughput.

Keywords: broadcast storm problem, end-to-end delay, flooding, process, throughput.

1. Introduction

Routing in a mobile ad hoc network with limited power has not been achieved up to the satisfactory level. Keeping the power constraint in mind, the proposed routing solution focuses on the minimizing the processing time and maximizing the throughput. MANET are formed by wireless mobile nodes that communicate without necessarily using the pre-existing network infrastructure such as base station/access point; where each mobile node operates not only as a node but also as a router so that it can send and receive packets as well as forward packets for others. The self-configuring nature of MANETs makes them suitable for a wide variety of applications [1]. One of the applications of this type of networks is communication within groups of people with laptops and other hand-held devices as well as in the battlefield and crisis management.

In node to node communication, two cases can occur: either they are neighbors, in which case they can communicate directly; or they are too distant, in which case messages must be routed via other nodes [2]. In pro-active routing protocol, like DSDV, maintains a table in each node about the complete network. The table contains information to transmit packet to desired destination. If the destination is within its own range, it transmits the packet directly; if the destination is a distant one, it transmits the packet to the hop that is mentioned in the routing table for the desired destination. In this scheme, every node needs to search the routing table to find the next hop to transmit where a delay is associated for each packet. Eventually, for the total transmission, where number of packets in a session is too high, overall delay would be significant one.

In the proposed method, some of the shortcomings of DSDV have been identified and solved increasing throughput while decreasing end-to-end transmission delay. This achievement could be used in multimedia and real time communication where throughput and end-to-end delay are the limiting factor.

2. Related Works

The wireless ad hoc network is represented by a graph $G = (V, E)$ where V is the set of vertices (mobiles) and $E \subseteq V^2$ the set of edges between these vertices. An edge exists between two nodes if they are able to communicate to each other, that is two nodes u and v can communicate if they are in the communicating radius of each other. If all nodes have the same range R, the set E is then defined as: $E = \{(u,v) \in V^2 \mid u \neq v \wedge d(u,v) \leq R\}$, d(u,v) being the Euclidean distance between u and v. The

---

[+] Tel.: +880-2-8414652; fax: +880-2-8812336.
E-mail address: anis@ewubd.edu

neighborhood set N(u) of the vertex u is defined as: $N(u) = \{v \mid (u,v) \in E\}$ [3]. It is assumed that each node u regularly emits special short message named HELLO message, containing its id, denoted by id(u).

In DSDV, each mobile node maintains a routing table, which lists all available destinations, the metric and next hop to each destination and generated sequence number [4]. Each node updates the routing table with advertisement periodically or when significant new information is available to maintain the consistency of the routing table of the ever changing topology of the ad hoc network. To reach this consistency, the routing information advertisement must be frequent or quick enough to ensure that each mobile node can almost always locate all the other mobile nodes in the dynamic ad hoc network. Upon the updated routing information, each node has to relay data packet to other nodes upon request.

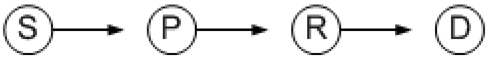

Fig. 1: A path from S to D

Fig. 1 shows an example of packet routing procedure from node S to D in DSDV. Here from S to reach D, each packet has to travel via P and R. Once node S sends a packet to P, as node D is not its neighbor; it searches its routing table to find the next node R to broadcast. Node R goes through the same procedure again until the node finds the destination. This mechanism will be repeated for each packet it receives which eventually increases the end-to-end delay and decreases throughput.

3. Proposed Method

In the proposed method, few short coming of DSDV have been identified and solved. Namely, the forwarded routing table includes lots of information to update the routing information of the nodes. Each node goes through the series of computation to figure out the next hop to relay the packet considering all other nodes of the network as destination. It has been observed that the most computed results remain unused, which eventually consumes lots of power while power in ad hoc network is one of the limiting factors. Moreover, finding the next hop for the ultimate destination for each packet consumes processing time, which eventually has negative effect in the throughput.

In the following sections, a quicker manner to find the path while consuming less power has been introduced resulting better throughput.

3.1. Neighborhood Information

At the very beginning the nodes are going to broadcast a short HELLO message containing its identification, denoted by *id(u)* to its neighbors. The nodes that are in the range would suppose to reply (same broadcast technology could be used) with another short HELLO message on reception. As HELLO messages are meant for one-hop neighbors, it is not going to create a broadcast storm problem. This is how all the nodes of the network are going to know its neighbors. Knowing the neighbors, each host is going to transmit the neighborhood set $N(u) = \{v \mid (u,v) \in E\}$ to its neighbors. Eventually, if all the nodes transmit the neighborhood set of other nodes to their respective neighbors; all the nodes of the network would come to know about all the hosts in the network and its current topology as it is done in any proactive routing algorithm. The overhead cost of transmitting the packets to know the topology is going to be less because at the very beginning before the data packets transmission, nodes would come know about the topology. As the topology will change because of nodes mobility, only the changed information will be broadcasted so that all the nodes are aware of the changes as much as possible. It is worth mentioning that network uptime at the very beginning is associated like all other pro-active protocols.

Fig. 2 shows a network where nodes B, C, D are neighbors of node A and similarly nodes A, G, H are the neighbors of B.

3.2. Building the Table

Each node of the network is going to be treated as a **process** locally. That means each process will be able to execute and return some value. Once a node knows about all other nodes in the network and who is whose neighbor, it is going to create a table. It's more like a hierarchical structure, keeping source node in the root and other associated nodes underneath. Whenever a node wishes to transmit some data to some destination, the process associated with the source node will execute and return its parent's Id. So, searching the table to find the path is no longer required according to the proposed method.

In fig. 2, node A's neighbors are {B, C, D}, B's neighbors are {A, G, H}, C's neighbors are {A, I} and D's neighbors are {A, E, F}. Once node A gets the neighborhood lists from all the nodes of the network, A is going to build the table containing nodes and its corresponding parent ID only. Table 1 shows the created table of node A for the network of fig. 2.

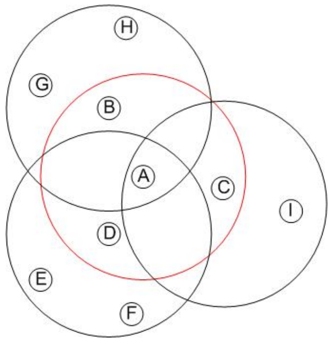

Fig. 2: Nodes and its neighbors

Source node (A)	
Nodes	Parent's Id
B	A
C	A
D	A
E	D
F	D
G	B
H	B
I	C

Table 1: Nodes and its corresponding parent ID

The convention to create the table from the neighborhood list is simple. As A is the source node, from the list of node A, A is considered as the parent of B, C, and D. Besides, from the list of node B, B is considered the parent of A, G, and H but here as node A is creating the table, node A is excluded from the table keeping only nodes G and H as the children of B. In this way, from the entire received list node A generates the table 1.

3.3. Deciding the Route

If the destination node is within the transmission range of source node, i.e. destination node is the neighbor of the source node; then it will communicate directly. But if the destination node is outside the transmission range of the source node, then source node need to find the path to destination via some other intermediate nodes.

The algorithm of the process is going to be as the following:

```
1.  Know the neighbors and all other nodes of the network
2.  Create a table of nodes and its corresponding parent Id
    // node wishes to transmit data to another node
3.  Generate a list $L_1$={destination node id}
4.  Process associated with destination node executes and return parent Id
5.  add to the list $L_1$
6.  **if** (parent Id = source Id) **then**
    6.1. path $P_1$ = reverse the list $L_1$
7.  **else**
    7.1. parent process associated with parent Id executes and return its parent Id
    7.2. add to the list $L_1$
    7.3. Go to step 6
8.  **end if**
9.  If multiple lists $L_1, L_2, L_3, \ldots\ldots, L_n$ exist
10. Choose the smallest path as the final path counting number of hops
```

The process associated with the destination node will be executed in the source node and it will return the parent node's identification. The identification of the parent node will be saved in a list. Eventually the process associated with the parent node will be executed and return the parent node's identification, which will be saved in the previous list. If the returned parent node identification is same as the source node, the execution of the further processes will stop. Otherwise, the execution of the processes will continue until it gets the source nodes identification as parent Id. The resultant list will be reversed to get the path from source to destination.

For the example of network in fig. 2, if A wants to transmit packet to F, i.e. A is the source and F is the destination; then a list L will be generated in the source node A, stating L = {F} where F denotes the destination address. Then process associated with node F is going to execute first and it will return its parent node identification D, which will be saved in the previous list L = {F, D}. As D is not the source, process associated with D is going to execute now and will return its parent node's identification, which is A and the parent Id will be saved in the same list L = {F, D, A}. The returned parent Id A is the same as source node's identification, the execution of the associated process A will stop. Now the list L = {F, D, A} will be reversed to get the path from source to destination, which is A → D → F.

The resultant list is the path from source to the destination. If there are two or more paths exist, the number of hops will be compared; the path with the minimum number of hops will be taken to transmit the data to the destination.

4. Performance Analysis

4.1. Simulation Environment

NS-2 packet level simulator (v.2.31)[5] has been used to simulate a square of 1000m by 1000m area populated with 25, 50, 75, 100, 125, 150 mobile nodes that are uniformly distributed in the region, each with a circular radio transmission range of radius 100m. This corresponds to networks consisting of multi-hops radio across while the selected mobile nodes represent the various network densities ranging from sparse to high density network [6]. The radio propagation model used in the simulation is the ns-2 default, which uses characteristics similar to commercial radio interference, Lucent's waveLAN card with 2 Mbps bit rate [7]. The simulation is allowed for 900 seconds for each simulation scenario and packet lengths are chosen with a distribution such that 30% of the packets are long packets (1024 bytes) and the remainder 70% are short packets (32 bytes). Other parameters used are show in Table 2.

Simulation Parameter	Value
Simulator	NS-2 (v.2.31)
Area	1000*1000 m^2
Transmission range	100 m
Bandwidth	2 Mbps
Interface queue length	50
Short packet size	32 bytes
Long packet size	1024 bytes
Number of nodes	25, 50, ….., 150
Simulation time	900 sec
Node speed	2 m/s…….20 m/s
Traffic type	CBR
Mobility	Random

Table 2: Simulation Parameters

4.2. Simulation Results

Fig. 3 shows that the end-to-end delays with respect to different number of nodes. The figure demonstrates that the proposed method achieved better performance while the network gets dense. With the stated packets distribution (30% longer packets and 70% short packets), though it took little longer when the number of hosts were minimal, as the number of nodes increases, the processing time of individual nodes decreases which eventually decreased the total end-to-end delay of the transmission as much as 17.3%.

Fig. 4 shows the drastic increase in throughput as the network gets denser. As the processing time to find the next hop no longer exists and end-to-end delay is less in the proposed method, the resultant throughput increase is reflected; while the offered load was 400kbps, the throughput has increased almost 14.7%.

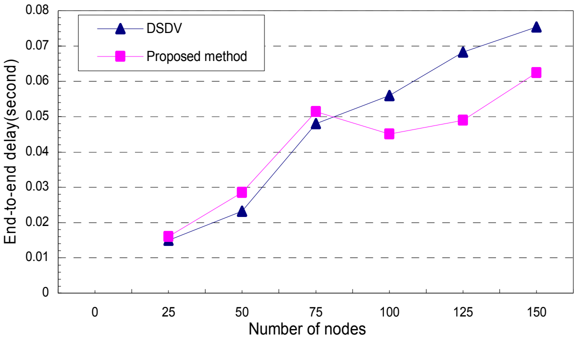

Fig. 3 End-to-end delay with respect to nodes

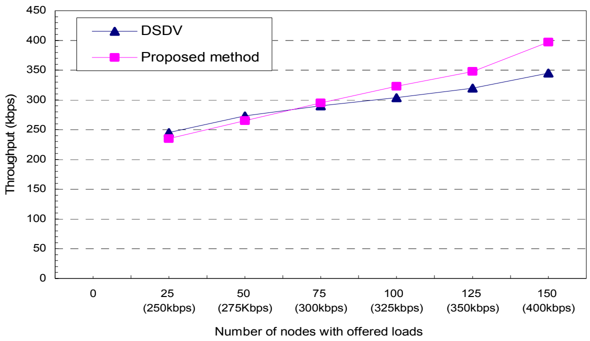

Fig. 4 Throughput with respect to loads

5. Conclusion

A mobile ad hoc network is considered in which the ad hoc communicating mode is available to increase the flexibility and mobility of users by minimizing delay and maximizing throughput. Some of the shortcomings of DSDV have been identified and solved. The proposed mechanism has minimized the processing time of individual node which is associated with finding the next hop to relay data packets. It also decides the path from source to destination from the proposed predefined routing table avoiding broadcast storm problem associated with flooding. In case of multiple paths to destination, the lowest number of hops will be chosen to transmit data. The simulation result shows that at some point the end-to-end delay has decreased as much as 17.3% and throughput increases almost 14.7%. In future, power consumption of individual nodes will be explored using this deferral scheme as it has shown better performance.

6. References

[1] F. Dai and J. Wu. Distributed dominant pruning in ad hoc networks. In Proceedings of the IEEE International Conference on Communications (ICC'03), Anchorage, AK, USA, May 2003.

[2] Williams, T. Camp, Comparison of broadcasting techniques for mobile adhoc networks. (*MOBIHOC 2002),* pages. 194–205, 2002.

[3] Framcois Ingelrest, David Simplot-Ryl, Ivan Stojmenovic. Routing and Broadcasting in Hybrid Ad Hoc Networks 2004.

[4] Perkins Charles E., Bhagwat Pravin: Highly Dynamic Destination-Sequenced Distance –Vector Routing (DSDV) for Mobile Computers, London England UK, SIGCOMM 94-8/94.

[5] The Network Simulator: NS-2: notes and documentation. http://www.isi.edu/nsnam/ns/.

[6] A. Mohammed, M. Ould, L. Mackenzie. An efficient counter-based broadcast scheme for mobile ad hoc networks, K. Wolter (Ed): EPEW 2007, pp 275-283

[7] IEEE802.11 WaveLAN PC Card – User's guide, A-1