International Journal of the Physical Sciences Vol. 7(5), pp. 769 - 775, 30 January, 2012
Available online at http://www.academicjournals.org/IJPS
DOI: 10.5897/IJPS11.1662
ISSN 1992 - 1950 ©2012 Academic Journals

Full Length Research Paper

Learning automata based multicast routing algorithm for wireless mobile ad-hoc networks

Parastoo Jameshourani[1*], Javad Akbari Torkestani[2], Mohammad Reza Meybodi[3]

Department of Computer Engineering, Islamic Azad University, Arak branch, Iran.
Young Researchers Club, Arak Branch, Islamic Azad University, Arak, Iran.
Department of Computer Engineering and IT, Amirkabir University of Technology, Tehran, Iran

Accepted 9 January, 2012

A wireless mobile ad-hoc network is a set of wireless mobile nodes that forms a temporary network with the capability of reconfiguration. Nodes in these networks can move freely and without dependence on any fixed connecting infrastructure. Due to their independence from a fixed structure as well as their easy reconfiguration, these networks have various applications in everyday life. Multicasting plays an important role in many applications of mobile ad-hoc networks. It can significantly improve the performance of these networks. This paper offers a distributed algorithm based on learning automata using the definition of Steiner connected dominating set problem for multicast routing in wireless mobile ad-hoc networks. Proposed algorithm is compared with existing leading ones and simulation results indicate that the proposed multicast routing algorithm works better in terms of packet delivery ratio and end to end delay.

Key words: Steiner connected dominating set, multicast routing, learning automa, mobile ad hoc networks.

INTRODUCTION

A mobile ad-hoc network (MANET) is a set of digital terminals equipped with radio senders/receivers in a way that they can connect one another without any infrastructures through exchanging data packets on a public channel. Lack of any infrastructure, such as an array of base stations makes ad-hoc networks different from other wireless networks. Since connections of a mobile terminal do not take place via a base station as conducted in a configured network like cellular networks, a mobile terminal in ad-hoc networks can directly connect with other nodes that it finds within its radio range. In order to connect with a node that is located outside radio range, a multihop transmission method is used (Ram Murthy and Manoj, 2004; Sobeih, 2002; Agrawal and Zeng, 2003). Nodes can play the role of a sender, a receiver or a relay. As ad-hoc networks do not make use of any router for routing mechanism, every host plays the role of a router. When a host tries to communicate with other hosts, the hosts in between the source and the destination(s) should be involved to make the connection. Beside their specific features, these networks inherit the common features of wireless networks. MANET users connect with each other via wireless connections and encounter the effects of radio communications including noise, overlap and loss. Regardless of the user type, a MANET needs distributed algorithms for organizing, scheduling and routing. Since MANET is a set of mobile nodes, their topology changes rapidly. These nodes use wireless channels with low bandwidth (Luo and Fang, 2003). Theoretically, we can describe a MANET as a unit diskgraph $G = (V, E)$, where the node set V represents a set of wireless mobile hosts and the edge set E represents a set of Links between the neighboring hosts, assuming all hosts have the same transmission range r. Regardless of the application, MANETs need efficient distributed algorithms to determine network organization, link scheduling and routing.

Multicast is a communication process for sending messages only to a group of receivers that have a common interest in receiving a certain data. The backbone infrastructure plays an important role in wireless networks for routing, connectivity management,

*Corresponding Author. E-mail: iau_akbari@yahoo.com.

broadcasting and multicasting. In a MANET, the backbone infrastructure is not a physical backbone, so we need to create a virtual backbone (VB), but a virtual one that always changes when the topology of the network changes. Ad-hoc network establishes some form of connected distribution structures often called as virtual multicast backbone (VMB). VMB spans all multicast members and contains all forwarding nodes that are responsible for forwarding multicast packets. Based on the structure of VMB that was used, the current multicast protocols can be mainly classified into three categories: Tree based, mesh based and hybrid. In wired networks, the optimal multicast backbone is defined as a Steiner tree (ST) problem in graphs, and multicast packets are forwarded along the tree edges. ST is not a suitable model for the ad-hoc networks because the goal of Steiner tree is to minimize total weight of all the links but our goal is to minimizethe number of the forwarding nodes group. On the other hand, the Steiner tree contains all multicast nodes and all forwarding nodes but some multicast hosts are only receivers not forwarder.

This article offers an algorithm based on learning automata in defining of Steiner connected dominating set for multicast routing in MANET. In this paper, our objective is to construct the optimal or near optimal VMB. The proposed multicast routing algorithm is compared with the best multicasting methods, and experimental results show that the proposed algorithm is superior to the existing methods in terms of the packet delivery ratio and end-to-end delay. The rest of the article is organized as follows: This study reviews Steiner connected dominating set, multicast routing protocols and learning automata, after which it explains the proposed algorithm based on learning automata. Furthermore, the proposed algorithm is compared with those by on-demand multicast routing protocol (ODMRP), Guo proposed protocol and SMRP, before it was finally concluded.

RELATED WORK

Here, we give a brief overview of the Steiner connected dominating set, multicast routing in MANET and learning automata.

Steiner connected dominating set

The problem of dominating set (DS) in graph $G(V,E)$ with vertices set of V and edges set of E is a subset of vertices like $S \subseteq V$ in a way that if there is no any vertex in this set, there should be at least one in the proximity of one of the nodes of this set (Santi, 2006). Set S is called the dominating set and the other nodes are called dominated. Connected dominating set (CDS) is a dominating set whose vertices are connected to each other (Santi, 2006). Steiner connected dominating set (SCDS) problem (Guha and Khuller, 1998) is a generalization of connected dominating set in which only one subset of the incoming graph nodes like $R \subseteq V$ are dominated. This problem was first introduced by Guha and Khuller (1998). It is considered as one of the nondeterministic polynomial (NP)-hard problems in general graphs and even in unit disk graphs (UDG). UDG are crossover graphs of same-size circles on the disk (Clark et al., 1990). One of the most important applications of SCDS problem is the creation of multicast virtual backbone in mobile ad-hoc networks (Wu et al., 2004). In order to solve this problem, Guha et al. (1998) introduced a greedy algorithm. The proposed algorithm uses set covering definition to find a dominating set and then applies Steiner tree approximation algorithm to connect the newly-found DS. The approximation rate for the found set is at most

$$((c+1)H(\delta)+c-1)\times Opt.$$

In this relationship, δ is the value of the biggest subset of R, c is the approximation rate of Steiner tree (currently $c \approx 1.55$ (Min et al., 2006)) and H is the harmonic function. Wu et al. (2004) introduced an algorithm based on maximal independent set (MIS) (Abello et al., 2001). This algorithm first creates a MIS like I on the nodes of set R and then applies Steiner tree algorithm on the vertices of this set. Since set I, is a MIS, all nodes of set R become dominated. Muhammad (2006) offered a two-phase algorithm. In the first phase, a MIS is computed for the vertices of set R and in phase two, a Steiner tree is used to connect the found MIS.

Multicast routing protocols in mobile ad-hoc networks

In multicast zone routing protocol (MZRP) (Devarapalli et al., 2001), a delivery tree rooted in the source node is created and includes two parts: An active protocol that is executed in every zone, and a multicast tree to send data to group members. In differential destination multicast (DDM) (Ji and Corson, 2001) protocol, the source node has the permission to control the members of multicast group. The source node encodes the address of the multicast receiver using a certain data header in multicast data packets. In every node, a forwarding set (FS) exists for every multicast session and keeps the destinations to which the node should send data. Ad-hoc multicast routing protocol utilizing increasing id-numberS (AMRIS) (Wu and Tay, 1999) protocol is an on-demand protocol that creates a multicast delivery tree to support senders and receivers inside a multicast session. Multicast ad-hoc on-demand distance vector (MAODV) (Royer and Perkins, 1999) is an extension of ad-hoc on-demand distance vector (AODV) (Perkins and Royer, 1997) unicast protocol. The first member of a multicast group is considered its leader and is responsible for keeping the multicast sequential number and distributing it among the multicast group. Associativity-based ad-hoc multicast (ABAM) (Toh et al., 2000) protocol is an on-demand method in which the sender starts the configuration of the multicast tree with multicast broadcast query (MBQ) packet flooding and through the ID of the node, every node in the network receives the MBQ packet. The receiver node collects all the packets of the multicast group which it wishes to connect with. Then, it selects the route with highest endurance and sends the MBQ reply packet to the sender node via the selected route. Wu and Tay (1999) extended multicast version optimized link state routing (OLSR) (Jacquet et al., 1998) protocol. In multicast optimized link state routing (MOLSR), (Laouiti et al., 2003) multicast processes are divided into three categories: making the tree, keeping the tree and disconnecting the tree. Every change in the network topology will lead to a change in the multicast tree. This algorithm proposes the minimal connection between multicast sender and receiver nodes. ODMRP (Gerla et al., 1999) uses the concept of the forwarding group (FG). Group membership and multicast routes are created by the source node as needed. In forwarding group multicast protocol (FGMP) (Chiang et al., 1998) protocol, the concept of the forwarding group is used to detect the nodes involved in transmission and for every multicast group like S, a forwarding set is considered. Core assisted mesh protocol (CAMP) (Garcia-Luna-Aceves and Madruga, 1999) is a mesh based multicast routing protocol. In mesh structure, at least one route is provided from each sender node to each receiver node. Progressively adapted sub-tree in dynamic mesh (PAST-DM) (Gui and Mohapatra, 2003) protocol is an overlay multicast routing protocol that creates a virtual mesh from all members of the multicast group. In neighbor-supporting multicast protocol (NSMP) (Lee et al., 2000), mesh structure is used to increase the flexibility for the node mobility. This protocol works independent of the unicast protocol. In AMRoute (Liu et al., 1999), a common tree is created using senders and receivers as the nodes of the tree. This protocol consists of two main parts: creating mesh and building tree. Guo and yang (2008) offered two algorithms in which the definition of maximum lifetime multicast (MLM) problem is used to increase the lifetime of multicast tree in static ad-hoc networks. In the first algorithm that we called Guo1, a multicast tree is created at the beginning of the session and the same tree is used during the multicast. Every node has the ability to control the power. In other words, sender can have access to nodes of different distances by adjusting the level of transmission power. Sensor mesh-based relocation protocol (SMRP) (Mustafa and Labiod, 2003) is a mesh based protocol and uses the concept of FG. In this protocol using route discovery process and select FG nodes are found several paths to multicast group. mobility-based hybrid multicast routing (MHMR) (An and Papavassiliou 2003) is a hybrid multicast routing protocol. It creates clusters then creates the trees. MHMR creates a mesh based on cluster-head.

Learning automata

Learning automata (Narendra and Tahtahchar 1989) is an abstract model with limited number of action and probability of choosing action. Each chosen action is evaluated by a stochastic environment and a response is sent back to the environment. Learning automata uses this response to choose an action for its next level (Unsel et al., 1999). Learning automata have been found to be useful in systems where incomplete information about the environment, in which those systems operate, exists. Learning automata are also proved to perform well in dynamic environment of the wireless, ad-hoc and sensor networks.

A distributed learning automaton (DLA) (Meybodi and Beigy, 2003) is a network of learning automata that cooperate to solve a problem. The number of action of automata in a DLA is equal to the number of the learning automata attached to it. Officially, DLA can be defined with graph $G=(V,E)$ in which V is the set of learning automata and $E\subset(V\times V)$ is the set of graph edges. Edge (i,j) demonstrates action j by the learning automata LA_i. Learning automata LA_j will be active when action j of the learning automata is chosen. An automata with variable action set (Tahtahchar and Harita, 1987) is one that the number of its action is variable in every moment. Such that the automata chooses its actions in moment n only from a non-empty subset $V(n)$ of actions which are called active actions. This choice is conducted by an extrinsic element and in a random manner. Automata works as follows: in order to choose an action in time n, learning automata computes first the sum of probability of its active actions $K(n)$ and then vector $P^{\wedge}(n)$ according to the following relationship:

$$P_i^{\wedge}(n)=prob[\alpha(n)\,|\,V(n)\ is\ set\ of\ active\ actions,\alpha_i\in V(n)]=\frac{P_i(n)}{K(n)} \tag{1}$$

Then, the automata randomly choose an action from its active actions in accordance with probability vector $P^{\wedge}(n)$ and apply it to the environment. If the chosen action is α_i, the automata updates probability vector $P^{\wedge}(n)$ of its actions as follows after receiving the response of the environment.

If the environment response is desirable:

$$P_i^{\wedge}(n+1) = P_i^{\wedge}(n) + a(1 - P_i^{\wedge}(n)) \quad \alpha(n) = \alpha_i$$
$$P_j^{\wedge}(n+1) = (1-a)P_j^{\wedge}(n) \qquad \alpha(n) = \alpha_j \quad \forall j,\ j \neq i \tag{2}$$

If the environment response is not desirable:

$$P_i^{\wedge}(n+1) = (1-b)P_i^{\wedge}(n) \qquad \alpha(n) = \alpha_i$$
$$P_j^{\wedge}(n+1) = \frac{b}{r^{\wedge}-1}(1-b)P_j^{\wedge}(n) \quad \alpha(n) = \alpha_j \quad \forall j,\ j \neq i \tag{3}$$

Then, the automata updates actions probability vector $P^{\wedge}(n)$ as follows:

$$P_j^{\wedge}(n+1) = P_j^{\wedge}(n+1) \qquad \forall j, \alpha_j \notin V(n)$$
$$P_j^{\wedge}(n+1) = P_j^{\wedge}(n+1)K(n) \quad \forall j, \alpha_j \in V(n) \tag{4}$$

In standard learning automata, one task is selected in every step and then, given the desirability or undesirability of the environment response, the task probability vector will be updated. In the new type of automata, k tasks are selected instead of one task in each step. This automata was first introduced by Abolhasani et al. (2009) in order to present a new algorithm for the fault-tolerant routing in sensor networks. Abolhasani introduces three learning algorithms for this type of automata. Here, the third learning algorithm known as "Orderly Linear Learning Algorithm" is applied as follows.

Assume that the h^{th} selected task is at the n^{th} step of task α_i. If the environment response is desirable:

$$\hat{p}_i^{h+1}(n) = \hat{p}_i^{h}(n) + a[1 - \hat{p}_i^{h}(n)]$$
$$\hat{p}_j^{h+1}(n) = (1-a)\hat{p}_j^{h}(n) \qquad \forall j \quad j \neq i \tag{5}$$

If the environment response is not desirable:

$$\hat{p}_i^{h+1}(n) = (1-b)\hat{p}_i^{h}(n)$$
$$\hat{p}_j^{h+1}(n) = (b/r_s^h - 1) + (1-b)\hat{p}_j^{h}(n) \quad \forall j \quad j \neq i \tag{6}$$

Then

$$M = \sum_{j=1}^{r_s^h} p_j^h(s) \quad \forall j \ \ sush \ \ that \ \ \alpha_j \in \alpha^h(s)$$

$$\hat{p}_i^h(s) = \frac{p_i^h(s)}{M} \quad \forall i \ \ sush \ \ that \ \ \alpha_i \in \alpha^h(s) \tag{7}$$

In the foregoing relation, a is the reward and b the penalty parameters. For the automata to return to typical condition and select $h+1^{th}$ task, the automata probability vector will be updated given Relation 8.

$$p_i^{h+1}(s) = \hat{p}_i^{h+1}(s) \times M \quad \forall i \ \ sush \ \ that \ \ \alpha_i \in \alpha^h(s)$$
$$p_i^{h+1}(s) = p_i^h(s) \quad \forall i \ \ sush \ \ that \ \ \alpha_i \quad \alpha^h(s) \tag{8}$$

DPDLA ALGORITHM

Here, we explained the proposed algorithm that we called DPDLA. In decisional partial discrete logarithm assumption (DPDLA), we used Steiner connected dominating set (SCDS) definition to create multicast virtual backbone. In order to map the MANET network to the learning automata, distributed learning automata is used. All of the automata allocated to the nodes are learning automata with variable action sets except for the automata of the sender node which is an automata with K selective actions, where K is a random number between 1 and n/5, so that n is the number of sender's neighbors. Every automata is a neighbor to another if it falls within the radio range of the host. The number of actions of each automata is equal to the number of hosts that exist in its neighborhood. Primarily, the probability of selecting every action in all of the hosts is equal and calculated by 1 divided by the number of neighboring hosts. We used learning automata repeatedly to create multicast routes and update the action probability vector until they find a near-optimal solution to the problem, whereas in our algorithm, we use LA. The probability of choosing stable routes increases whiles the probability of choosing unstable routes decreases so that the probability of choosing the most stable multicast route converges to one. The reward and penalty parameters of our algorithm are set to 0.15 and 0.04, respectively.
The action probability vector of the activated automata is updated using a L_{RP} reinforcement scheme. Every host needs to have the following data structures for moving and routing:

1. JQUE packet: in order to find the route, this packet is sent from one node to another and includes the ID of the sender node of the group, ID of the multicast group, list of the receiver nodes, list of the so-far selected nodes, number of iteration and the sequential number of the packet.
2. FinDes packet: this packet is sent to the sender node of the group when a node identifies one or more receiver nodes in its neighborhood. It includes: the ID of the sender node of the group, ID of the multicast group, list of the so-far selected nodes, ID of the identified receiver nodes and number of the identified receiver nodes.
3. EmpDes packet: sender of the group sends this packet to notify it has found a route to all of the receiver nodes of the group and inform automata not to select any other node.
4. Pnz packet: sender of the group sends this message and penalizes the selected nodes (through reducing their probability).

This message includes: the ID of the sender node of the group, the considered route that has been called from an entry of NbrTbl table and the packet must pass through it and also a flag that clarifies the packet as reward so that it is not mistaken with penalty packet. After receiving this packet, node X examines the route identified in the message and rewards the element, like Y, that follows in the route (this in fact is the select action by automaton X in the stage of finding the route). The amount of penalty and reward is determined overall.

```
Procedure Select _Neighbor ( )
Begin
   Current _ Node _ Neighbor_ list =Current _ Node _
Neighbor_ list – Parent _ Node
If (Current _ Node _ Neighbor_ list ∩ Receiver _List ≠ ø)
   Delete common node(s) from Receiver _List
   Send FinDes packet to Sender
   End if
   if (Receiver _List ≠ ø )
           Add Current _ Node to the JQUE packet
           Select Random Neighbor node and Send JQUE
packet
    End if
End
Algorithm DPDLA
Input: Sender, Receivers
Output: multicast Routes
Begin
  Best _ Size = number of all nodes
  Repeat
      Each node Finds its Neighbor_ List with sending Echo
packet to its radio range
     if (Sender _ Node _ Neighbor_ list ∩ Receiver _List  ≠ ø)
          Delete common node(s) from Receiver _List
     End if
if (Receiver _List ≠ ø )
          K= Random ( )
          for i :=1 to K
               Create JQUE packet
               Send JQUE packet to selected nodes
           End for
      End if
       While (Receiver _List ≠ ø)
           Select _Neighbor ( )
       End While
       Sender broadcasts EmpDes   packet
       Sender compute the   Path _ Size
       if (Path _ Size < = Best _ Size)
           Sender Send Rwd packet for all selected Nodes
           Best _ Size = Path _ Size
     else
          Sender Send Pnz packet for all selected Nodes
     End if
 Until (Stop condition)
```

Figure 1. Pesudo code of the DPDLA algorithm.

5. Rwd packet: this message is sent by the sender of the group and rewards the selected nodes (through increasing their probability). It includes the ID of the sender node of the group, the considered route that has been called from an entry of NbrTbl table and the packet must pass through it, and also a flag that clarifies the packet as penalty so that it is not mistaken with reward packet. After receiving this packet, node X examines the route identified in the message and penalizes the element, like Y, that follows in the route (this in fact is the select action by automaton X in the stage of finding the route). The amount of penalty and reward is determined overall.

When the sender is about to start a session or begin a new round, the automata of the sender is activated and compares its action list with the list of receiver nodes. If it finds common elements, it eliminates them from the list of receiver nodes. In case the list is not completely empty, it randomly calculates the amount of K and randomly selects k actions from its action list and then creates k JQUE packets. Sender in every created packet adds its own ID to the field "selected nodes list" of the packet, adds one unit to the number of iteration, places the receiver nodes list in the field "receiver nodes list" and sends the packet to its selected neighboring node.

Step 1: Having received JQUE packet, if it has not received it before, each node first eliminates the ID of the node from which it received the packet from its own action list and then compares its action list with the receiver nodes list that exists in the received message. If it finds common node(s), it eliminates them from the field "receiver nodes list" and sends a FinDes message to the sender of the group via the route saved in the field "selected nodes" from the received JQUE message.
Step 2: If the receiver nodes list is not empty, the activated automaton randomly selects an action from its action list, adds its own ID to the filed "selected nodes list" and sends the JQUE packet to the selected node.
Step 3: The operation of selecting a node and sending a JQUE packet continues until either the receiver nodes list runs out of JQUE messages or an EmpDes message is received from the sender.
Step 4: When a FinDes packet is received, for every node that exists in the field "identified nodes" in the packet, the sender node of the group checks the corresponding entry in the NbrTbl table. If the entry is empty, it values it with the amount that exists in the field "identified receiver nodes" and considers it as the route to the considered receiver node.
Step 5: The sender examines to see if all of the entries of the table are valued (that is, a route has been identified to all of the receiver nodes), it continues to step 6; otherwise, step 2 is repeated.
Step 6: The sender distributed an EmpDes message in the network which means the end of an iteration cycle of the algorithm. Then, the sender of the group calculates the route to the receivers in an unrepeated manner based on the entries of the table. If the obtained amount is less than that of the current route, it rewards the nodes on the route by sending an Rwd packet; otherwise, it panelizes them by sending a Pnz packet. When receiving a reward or a penalty packet, every node adds its eliminated actions to the action list again. Sender of the group examines the number of iteration. If it has not reached the maximum level, it starts the algorithm again; otherwise, it picks the selected routes as the best route and sends data packets through these routes.
Step 7: The process of activating and selecting nodes and executing the algorithm continues until the number of iteration reaches the pre-defined one or the threshold term is met.

After rewarding the chosen action, the selected action probability vector must be updated once again by enabling all the disabled actions. Note at the end of each round, DPDLA finds a multicast route. Figure 1 shows the pseudo code of the proposed algorithm for multicast routing in MANET.

SIMULATION RESULTS

The proposed algorithm was compared with ODMRP, SMRP and Guo1 algorithms. Applying Glomosim simulator, simulation characteristics are as follows: capacity of each channel is 2 mb/s; diffusion range of all of the hosts is 250 m; the area of network is 2000 × 2000 m. We study the impact of the reward/penalty parameters and stop condition on the performance of the proposed algorithm. To do so, we changed the learning parameters

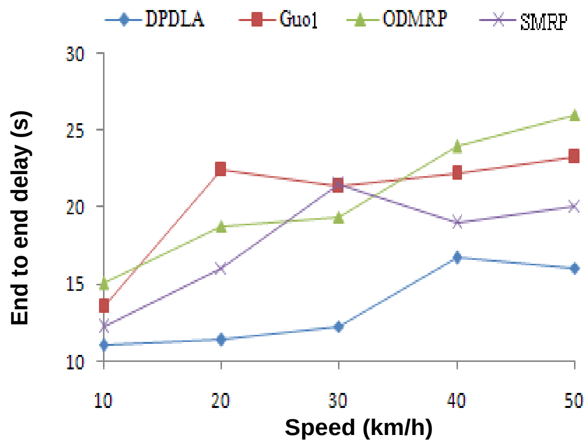

Figure 2. End-to-end delay as a function of the mobility speed.

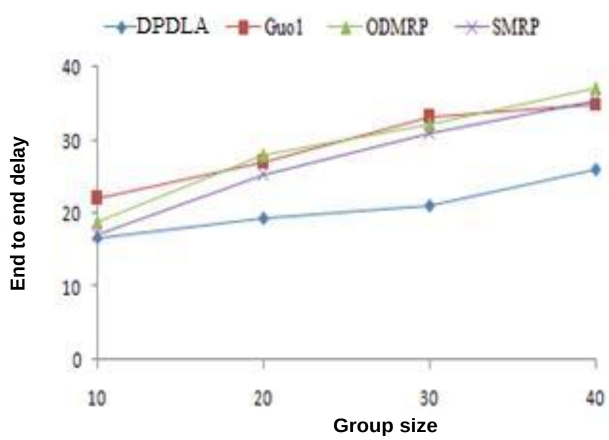

Figure 3. End-to-end delay as a function of the multicast group size.

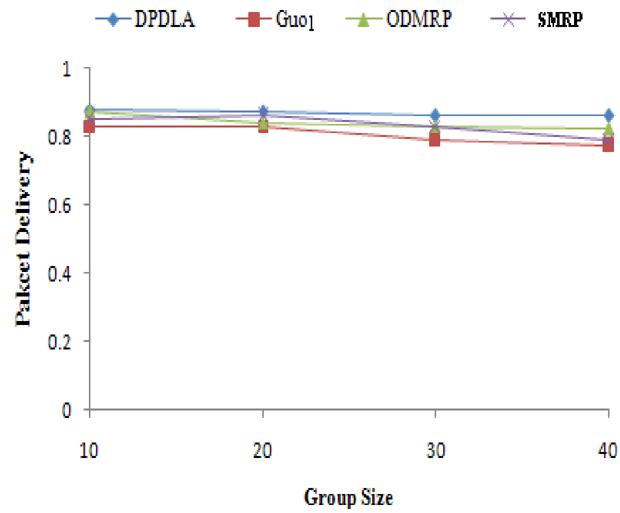

Figure 4. Packet delivery ratio as a function of the multicast group size.

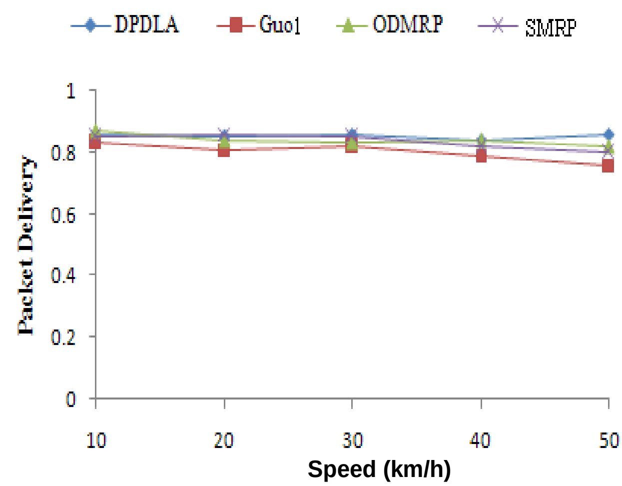

Figure 5. Packet delivery ratio as a function of the mobility speed.

to 0.01, 0.04, 0.05, 0.1, 0.15, 0.2 and 0.25, and measured the algorithm performance. The experiments show that the best value for reward and penalty parameters is 0.15 and 0.04, respectively. The following diagrams are obtained from networks with 100 nodes. In these experiments, 100 vertices are randomly distributed in the simulation area and we varied the mobility speed from 10 to 50 km/h. These diagrams examine and demonstrate end to end delay, packet delivery ratio based on parameters of hosts' speed and the size of multicast group.

Figures 2 and 3 showed the end to end delay as a function of the mobility speed and the multicast group size. The packet delivery ratio of the multicast routing algorithm as a function of the mobility speed and multicast group size is shown in Figures 4 and 5.

Conclusion

In MANET, the mobility of hosts results in path breaks and less routing information. This paper offers an algorithm, called DPDLA, for multicast routing in MANET based on learning automata using a L_{RP} reinforcement scheme. We tested the proposed algorithm with different learning parameters and stopping condition also. The reward and penalty parameters of our algorithm are set to 0.15 and 0.04, respectively. The proposed algorithm was compared with ODMRP, SMRP and Guo1 algorithms. Simulation results demonstrate the advantages of the proposed algorithm both in end to end delay and packet delivery ratio.

REFERENCES

Abello J, Butenko S, Pardalos PM, Resende MGC (2001). Finding independent sets in a graph using continuous multivariable polynomial formulations. J. Global Optim., 21(2) : 111-137.

Abolhasani M, Esnaashari M, Meybodi MR (2009). LADIT: A Learning Automata Fault Tolerant Routing Protocol for Sensor Networks. Proceedings of the Third Joint Congress on Fuzzy and Intelligent Systems, University of Yazd, pp. 15-17.

Agrawal DP, Zeng QA (2003). Introduction to Wireless and Mobile Systems, Brooks/Cole.

An B, Papavassiliou S (2003). MHMR: mobility-based hybrid multicast routing protocol in mobile ad-hoc wireless networks. Wireless Commun. Mobile Comput., 3: 255–270.

Chiang CC, Gerla M, Zhang L (1998). Adaptive shared tree multicast in mobile wireless networks. Proceedings GLOBECOM. 3: 1817–1822.

Clark BN, Colbourn CJ, Johnson DS (1990). Unit Disc Graphs. Discrete Math., 86: 165-177.

Devarapalli V, Seluck AA, Sidhu D (2001). Mzr: A multicast protocol for mobile ad hoc networks. Internet Draft, <draftvijay- manet-mzr-01.txt>.

Garcia-Luna-Aceves JJ, Madruga EL (1999). The Core Assisted Mesh Protocol. IEEE Journal on Selected Areas in Communications, Special Issue on Ad-Hoc Networks, 17: 1380-1394.

Gerla M, Lee SJ, Su W (1999). On-demand multicast routing protocol (odmrp) for ad hoc networks**.** Proceedings of the wireless communications and networking conference (WCNC), pp. 1298-1302.

Guha S, Khuller S (1998). Approximation algorithms for Connected Dominating Sets. Algorithmica, 20(4): 374-387.

Gui C, Mohapatra P (2003). Efficient Overlay Multicast for Mobile Ad Hoc Networks. IEEE Wireless Communications and Networking (WCNC), 2: 1118 – 1123.

Guo S, Yang O (2008). Maximizing multicast communication lifetime in wireless mobile ad hoc networks. IEEE Trans. Vehicular Technol., 75(4): 2414 – 2425.

Jacquet P, Muhlethaler P, Qayyum A (1998). Optimized Link State Routing Protocol. Internet Draft, draft-ietf-manetolsr-O.

Ji L, Corson MS (2001). Differential destination multicast – a manet multicast routing protocol for small groups. In INFOCOM, 2: 1192–2002.

Laouiti A, Jacquet Pminet P, Viennot L, Clausen T, Adjih C(2003). Multicast Optimized Link State Routing. Instıtut Natıonal De Recherche EnInformatıque Et En Automatıque (INRIA).

Lee SJ, Su W, Hsu W, Gerla M, Bagrodia R(2000). A performance comparison study of ad hoc wireless multicast protocols. Proceedings of IEEE INFOCOM, 2: 565–574.

Liu M, Talpade R, McAuley A Bommaiah E (1999). Amroute: Ad hoc multicast routing protocol. Mobile Netw. Appl., 7, pp. 429-439.

Luo W, FangY (2003). A Survey of Wireless Security in Mobile Ad hoc Networks: Challenges andAvailable Solutions, Ad hoc wireless networking. Dordrecht: Kluwer Academic Publishers, pp. 319–64.

Meybodi MR, Beigy H (2003). *Solving Stochastic Shortest Path Problem Using Monte Carlo Sampling* Method: A Distributed Learning Automata Approach. Proceedings of the sixth Int. Confer. Neural Networks Soft Comput.. pp. 626-632.

Min M, Du H, Jia X, Huang CX, Huang SCH, Wu W (2006). Improving Construction for Connected Dominating Set with Steiner Tree in Wireless Sensor Networks. J. Global Optim., 35(1) : 111-119.

Muhammad RB (2006). Distributed Steiner Tree Algorithm and its Application in Ad-hoc Wireless Networks. Proceedings International Conference on Wireless Networks, pp. 173-178.

Mustafa H, Labiod H (2003). A performance comparison of multicast routing protocols in ad hoc networks. Personal, Indoor and Mobile Radio Communications, 1: 497 – 501.

Narendra KS, Tahtahchar MAL (1989). Learning Automata: An Introduction, Prentice Hall.

Perkins CE, Royer EM (1997). Ad hoc On Demand Distance Vector Routing (AODV). Processings of th 2nd IEEE workshop on mobile computing system and applications, pp. 90-100.

Ram Murthy CS, Manoj BS (2004). Ad hoc wireless networks architectures and protocols. Prentice Hall.

Royer EM, Perkins CE (1999). Multicast ad hoc on-demand distance vector (maodv). Proceedings Int. Confer. Mobile comput. Netw. (MobiCom), pp. 207-218.

Santi P (2006). Topology Control in wireless Ad hoc and sensor network. New York. Wiley.

Sobeih A (2002). Reliable multicasting in wireless mobile multihop ad hoc network. M.Sc thesis, Faculty of Engineering, Cairo University.

Tahtahchar MAL, Harita BR (1987). Learning automata with changing number of actions.In IEEE Trans. System, Man Cybernetics. 17(6).

Toh CK, Guichala G, Bunchua S (2000). Abam: On demand *associativity* -based multicast routing for ad hoc mobile networks. Proceedings of IEEE VTC, 3: 987–993.

Unsel C, Kachroo P, Bay JS (1999). Multiple Stochastic Learning Automata for Vehicle Path Control in an Automated Highway System. IEEE Trans. Systems, Man Cybernetics, 3: 29-45.

Wu CW, Tay YC (1999). Amris: A multicast protocol for ad hoc wireless networks. MILCOM, 1: 25–29.

Wu YF, Xu YL, Chen GL, Wang K (2004). On the construction of virtual multicast backbone for wireless ad-hoc networks. IEEE International Conference on Mobile Ad-hoc and Sensor Systems, pp. 294-303.