A Routing Algorithm based on Cellular Automata for Mobile Ad-hoc Networks

Azadeh Ghalavand[1], Ahmad khademzadeh[2], Arash Dana[3] and Golnoosh Ghalavand[4]

[1] Dept. of Computer. Eng, South Tehran Branch, Islamic Azad University Tehran, Iran

[2] Education and International Scientific Cooperation Dept, Iran Telecommunication Research Center

[3] Dept. of Elect. Eng. ,Central Tehran Branch, Islamic Azad University, Tehran, Iran

[4] Dept. of Computer. Eng, Science And Research Branch, Islamic Azad University, Tehran, Iran

Abstract

Mobile ad hoc networks (MANETs) are self organizing, adaptive and infrastructure less networks. Analyzing these networks is a complex task due to the high mobility and rapid topology change. Routing in MANETs is one of the challenging tasks. Although a lot of researches have done in this area, the usage of cellular automata in routing for MANETs has not been explored. The proposed routing algorithm is a new cellular automaton based routing algorithm for mobile ad hoc networks, which finds a route that not only has least number of hops but also supports Quality of service**.** The solution presented here, is based on selecting a delay constrained shortest path between source and destination as a best route by using cellular automata. A simulator has been developed to evaluate a routing protocol and the obtained results indicate that the efficiency of the proposed protocol especially in satisfying QoS requirements.

Keywords: *Mobile Ad-hoc Networks, Routing, Quality of Service, Cellular Automata.*

1. Introduction

Mobile ad hoc networks (MANETs)[1] are networks composed of a set of mobile nodes able to spontaneously communicate with each other over a wireless medium without any fixed infrastructure. The topology of such networks changes dynamically as the nodes are free to move about arbitrarily. MANETs are useful in many applications as virtual classrooms, military communications, emergency search, rescue operations, at airport terminals for workers to share files, and etc [2, 3] .

One of the challenging aspects in the study of these networks is the routing algorithm for data transmission between the nodes. Any routing algorithm has to take into account the highly dynamic nature of the mobile nodes in the network and the limited resources of such networks. Many routing protocols for MANETs has been proposed, such as: Ad hoc On Demand Vector (AODV) [4], Associativity-Based Routing (ABR) [5], Dynamic Source Routing (DSR)[6], Temporally Ordered Routing Algorithm (TORA)[7], and etc. They are mostly designed for best effort transmission without any guarantee of quality of Service. Quality of Service (QoS) models in mobile ad hoc networks become more and more required because more and more real time applications are implemented on the network. For considering QoS, extensions, often given in terms of bandwidth or delay, can be added to the messages used during the route discovery process. Several works have been done to provide QoS routing algorithm such as Quality of Service for Ad-hoc on Demand Distance Vector (QoS-AODV) [8] which is based on AODV and creates routes according to the QoS requirements of the applications, a brief survey about this issue can be found in[9,10].

Nowadays, many areas of research have been benefited with the theories related to the cellular automata (CA)[11], mainly the areas that need to deal with systems that are constantly changing or have a random behavior as physics (mainly thermodynamics), ecology, medicine ,computer science, and chemistry. Cellular Automata (CA) consist of n-dimensional lattice of cells with states that evolve according to a set of rules. These rules are defined by the current state of the cell and the possible states of it's adjacent cells . These discrete time decentralized systems are useful and efficient as a computation tool for describing complex systems like physical systems.[12, 13].

Real ad-hoc networks infrastructures are still very expensive. Therefore, most of the evaluations of new protocols are being made through simulation tools and Different modeling, [14,15].Recently, different authors [16,19,17,18] decided to solve the problems in ad hoc networks using CA. Subrata and Zomaya [16] address the location tracking/management problem. They use CA combined with a genetic algorithm to create an evolving parallel reporting cells planning algorithm. The genetic algorithm is used to discover the 'best' CA transition rules for the application. Their results showed that the

discovered CA rules describe near-optimal solutions to the reporting cells problem. In a related application, Kirkpatrick and Van Scoy [17] investigate the use of CA to model message broadcasting in highly mobile ad hoc networks. They point out some strengths of the CA approach (discrete time steps useful in defining rapidly changing systems, and the approach is conducive to the design of a GUI), and derive an upper bound on the time required by a broadcast algorithm to distribute a message across a network. A paper by Cunha et al. [18] shows the applicability of CA to simulate the topology control problem in sensor networks. They concluded that CA can be a worthwhile simulation tool for large WSNs, because it allows for the fast and objective verification of network properties such as degree of coverage and degree of connectivity. The approach presented in[19]is modeling some features of ad-hoc networks such as mobility, network coverage and routing in wireless ad-hoc networks using Cell-DEVS which is an extension to Cellular Automata in which each cell in the system is considered a DEVS model. Their research shows that routing protocols, such as AODV can be successfully mapped onto Cell-DEVS.

Although several researchers have attempted to use CA for modeling some aspects of ad hoc networks, they don't consider the problem of routing with adequate QoS in MANET. The objective of this work is to verify the applicability of using cellular automata to solve this problem in mobile ad-hoc networks. The solution presented is based on a modified version of classical Lee's Algorithm [13] to find out the shortest path that satisfies the delay constraint as a QoS metric between two communicating nodes on a network plane. In response to this need, we add an extension to the routing request message in AODV protocol to define delay constraints which must be met by nodes during the discovery procedure of route, so the route with the least number of hops that satisfied delay constraint is chosen as a best route. We define our protocol and also mobility model with the concept of the CA, then we evaluate the performance of it in comparison with AODV and QoS for AODV routing protocols (QoS-AODV) in terms of average delay, and packet delivery fraction. Simulations are presented in different delay constraints and mobility speeds .

The remainder of the paper is structured as follows. The main concept of cellular automata is presented in section 2, including their formal definition and some practical applications. Details of our routing protocol are given in Section 3. Section 4 describes the simulation environment and results. Finally, our concluding remarks and future work are presented in Section 7.

2. Cellular Automata Concept

Cellular automata (CA)[11] are interesting computation systems to study because of their simplicity and inherently parallel operation. Such systems have the potential to model the behavior of complex systems in nature. For this reason CA and related architectures have been studied extensively in the natural sciences, mathematics, and in computer science. They have been used as models of physical and biological phenomena, such as fluid flow, galaxy formation, earthquakes, and biological pattern formation. They have been considered as mathematical objects about which formal properties can be proved. They have been used as parallel computing devices, both for the high-speed simulation of scientific models and for computational tasks such as image processing.[21]

CA can be formally defined as a 4-tuple (L, S, N, f) where[18]:

- L is a regular grid. The elements that compose this grid are called cells.
- S is a finite set of states, S ={0,1….,s-1}.
- N is a finite set (of size $|N|$ = n) of neighborhood index, such that $\forall c \in N$ and $\forall k \in L : k + c \in L$
- $f : S^n \longrightarrow S$ is a transition function *where* n is the size of the neighborhood.

The grid can be in any finite number of dimensions (e.g. a line or a cube of cells) and the time basis of the system is synchronous (t = 0, 1, 2,…). A configuration $C_t : L \rightarrow S$ is a function that associates one state with one cell of the grid. The task of this function is to change the configuration C_t to a new configuration C_{t+1} (see Equation1).

$$C_{t+i}(k) = f\ (C_t(i)\ |\ i \in N(k)) \qquad (1)$$

where $N(k)$ is the set of all neighbors of the cell k.

$$N(k) = i \in L\ |\ k - i \in N \qquad (2)$$

Generally, the CA starts at time t = 0 with each of the cells in one of their n possible states. This global state is known as the initial configuration. All cells have their own state transition function, also called the local rule, whose input is the states of their neighbors, and output is one of the possible states. At time step t, each cell uses the states of its neighbors as input to the state transition function, and the output of the function is the cell's new state at time t + 1.The rule for updating the state of cells is the same for each cell and does not change over time, and the entire CA updates synchronously in this fashion, though exceptions are known. Fig.1 shows the evolution of a one-dimensional, binary-state cellular automaton with infinite cells. As you see the neighborhood of cells is radious-1 (r = 1)which refers the neighbors of a cell are one cell from

it . The two fundamental types of neighborhood in CA systems are *von Neumann* neighborhood and *Moor* neighborhood, these two standard radius-1 neighborhoods in a two-dimensional cellular grid are depicted in Figure 2. The Moore neighborhood consists of the 8 surrounding cells of the cell depending on whether or not the central cell is counted, and the von Neumann neighborhood consisting of the 4 cell array - defined as north, south, east and west; that are strictly adjacent to the central cell.

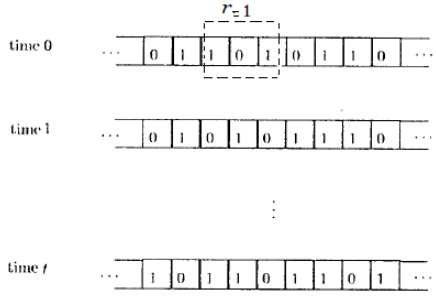
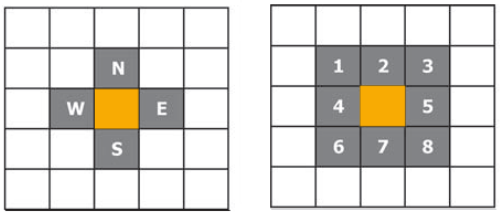


Fig.1. Evolution of 1-d, binary-state, nearest-neighbor (r =1) CA



(a) (b)

Fig. 2. von Neumann neighborhood (a) & Moor neighborhood (b) in Two- Dimensional CA.

2.1. Lee Algorithm

A very well known approach to routing problems is the Lee algorithm [20]. Its purpose is to find an optimal path between two points on a regular grid. The algorithm finds the path on the grid with the lowest sum of weights. By adjusting the weights of the grid points the user has some control over what is supposed to be an optimal path.

At a first glance this algorithm looks like it perfectly fits onto a cellular automaton. Unfortunately the number of states required to perform the algorithm is related to the longest path or more precisely to the largest accumulated weight that may occur. Thus in [13] the writers decided to develop a version of the algorithm which has a constant number of states. Hochberger and Hoffman [13] use a cellular processing model to simulate the Lee algorithm for the routing of connections on printed circuit boards. A modification was made to the Lee algorithm so that in a CA implementation it only requires 14 states. They postulate that mapping classical algorithmic problems onto the CA model may lead to new ideas in their theory and implementation.

The brief description of their algorithm is as follows:

The algorithm works in two phases and with fourteen states. At the beginning of the first phase all cells, except the start and the end point, are in the **free** state. During the first phase, all cells check whether there is any cell in the neighborhood that already has a wave mark. If a wave mark is found, the cell itself becomes a wave mark towards the already marked cell for forming a reverse path to the starting point. Fig.3 shows a sample grid at the end of phase one where S and D represent respectively, a **start** and end points. The **wave** marks are symbolized by small arrows and the black squares are obstacles which had state **used** .

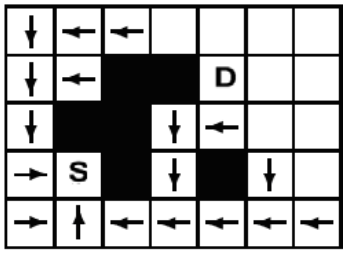


Fig.3. Sample grid at the end of phase 1

Phase one ends when the end point is reached by the wave. Now the path is built backward towards the start point along the wave marks**.** If a cell is one of the **wave** states and it sees a neighbor cell that is a path towards this cell, then this cell becomes a **path** in the direction of its previous mark (Fig.4).

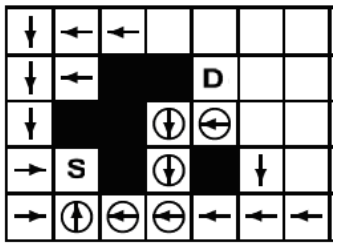


Fig.4. Forming the path along the wave marks

All other wave marks which didn't become part of the path, have to be cleared in order to allow subsequent routing passes**.** For this purpose all cells that see a neighbor cell which is a path not pointing towards this cell are cleared. Also all cells are cleared, that see a neighbor in the clear state. A cell that is in the **clear** state will become free in the next generation**.** Since the building of the path moves along the shortest path, it is impossible that a cell in **clear** state can reach a cell which will be in the path but is not yet part of it.

The algorithm terminates when the **start** cell sees one if its neighbors become part of the path. So the **start** cell changes its state to **ready** and signals that the path is complete**.**

3. MODELING the Routing problem using CA

In this section, we are going to discuss the methodology used to model the problem of finding route which has minimum hop counts and satisfies end to end delay requirement in cellular automata. We use the idea of QoS-AODV, a QoS version of AODV, which was proposed in 2000 by Perkins et al[8]. AODV is one of the first ad-hoc on demand routing algorithms chosen by IETF, that just find the shortest path without supporting any QoS attributes, but QoS-AODV establishes a route between source and destination by specifying Quality of Service parameters, namely maximum delay or minimum bandwidth .To give a more accurate definition of our objective, we first describe , an abstract description of this famous routing protocol.

3.1. Quality of Service for AODV

The AODV [4], a well known ad-hoc routing protocol, creates routes on demand. Whenever a node needs to communicate with another one, it broadcasts a Route Request message (RREQ) to its neighbors. They re-broadcast the message and set up a reverse path pointing towards the source. When the intended destination receives a RREQ message, it replies by sending a Route Reply (RREP) that travels along the reverse path set up by RREQ. AODV is based on the shortest path between two nodes in an ad hoc network plane and does not support QoS features. In contrast, QoS-AODV by considering quality-of-service (QoS), finds the best route that satisfies the end to-end QoS requirement. Infact, QoS-AODV modifies the route discovery and maintenance mechanisms of AODV to provide QoS assurance.

A QoS extension for AODV routing packets was proposed in [8]**.** This QoS object extension includes the Minimum bandwidth or Maximum delay parameters of each application, and it also has a "session *ID"* which is used to identify each QoS flow that is established according to the application. The extensions are added to the route messages (RREQ, RREP) during the phase of route discovery. A node which receives a RREQ with a quality of service extension must be able to meet the service requirement in order to either rebroadcast the RREQ, or unicast a RREP to the source. The maximum delay used in RREQ indicates the maximum time allowed to be used for a transmission from the current node to the destination and the minimum bandwidth extension specifies the minimum amount of bandwidth that must be made available along an acceptable path from the source to the destination. When a node receives RREQ, the maximum delay will be compared with the NODE TRAVERSAL TIME (should include queuing delays, interrupt processing times and transfer times). The NODE TRAVERSAL TIME could have a different value. If the maximum allowable delay is smaller than the NODE TRAVERSAL TIME, the RREQ will not be broadcasted further, otherwise, the RREQ will be broadcasted and before the broadcasting, traversal time of this node must be subtracted from the maximum delay field in RREQ message. In RREP message, the maximum delay extension field will be filled with zeroes, and delays are cumulative from the destination along the route that the RREP come from to the source node, and the source node will choose the one satisfying the delay requirement. Similarly, the minimum bandwidth extension can be also appended to a RREQ by a requesting node. In this case, before forwarding the RREQ, an intermediate node must compare its available link capacity to the bandwidth field in the extension. If the requested amount of bandwidth is not available, the node must discard the RREQ and not process it any further. Otherwise, the node continues processing the RREQ as specified in [8].When the destination generates a RREP in response to a RREQ with a minimum bandwidth extension; the bandwidth field in the RREP is initially infinity. Each node forwarding the RREP compares the bandwidth field in the RREP and its own link capacity and maintains the minimum of the two in the bandwidth field of the RREP before forwarding the RREP. This value also goes in the route table entry for the corresponding destination (as per the destination IP address in the RREP) and indicates the available minimum bandwidth to the destination. It enables the intermediate node to respond to a later RREQ with a minimum bandwidth extension by comparing the requested minimum bandwidth and the available bandwidth recorded in the route table entry.

3.2. 1. Delay constraint Assumption

As a RREQ may include a maximum delay extension or a minimum bandwidth extension, for simplicity in this work, we only consider the delay constraint. So by giving the delay requirement as *Dmax*, the goal is to detect a path from source node ***S*** to destination node ***D*** such that the delay on the path does not exceed *Dmax*.

To control dissemination of RREQ with this QoS requirement (delay) the following assumptions are considered:

First, we assume 40 millisecond for NODE_TRAVERSAL_TIME .

Second, If the NODE_TRAVERSAL_TIME is greater than or equal to the maximum delay in delay field of RREQ, the intermediate node must discard the RREQ.

Third If the NODE_TRAVERSAL_TIME is less than the maximum (remaining) delay in message by subtracting from its value of the NODE_ TRAVERSAL_TIME, the

intermediate node should send a RREP to the source or must continue broadcasting the RREQ .

4.3. Proposed Model

In order to find a path with adequate QoS between two communicating nodes in the network we consider only delay as the QoS constraint. we assume a two dimensional network plane consisting of a number of mobile nodes spread randomly . There is no hierarchy between nodes, and The network plane is homogeneous (all nodes have the same properties). Data movement between two cells represents one hop. Every node represents a cell of the cellular automaton. Each cell has a state which corresponds to the status of the node that occupies it. The neighborhood of each cell for finding the route, is the range of communication of each node defined as a Von-Neumann neighborhood governed by a circle of radius 1 around the cell .The network plane also contains *dead cells* through which communication cannot take place. These cells represent physical obstacles (such as a high-rise building) or simply the absence of a communication link. Two nodes with a dead cell between them cannot communicate directly to each other .As network nodes can move randomly to any of the neighboring cells the Moor neighborhood will be used for mobility. By using the modified Lee's Algorithm [13] with some variances, we define our routing method which satisfies the assumptions discussed in section 3.2.1.

In our method, for discovering a QoS path that satisfies the delay-constraint as the QoS parameter, the source node broadcasts QRREQ message with the delay requirement of the connection request (maximum delay(*Dmax*)), to its communicating neighbors which includes, all the nodes on its top, down, left and right side. According to lee algorithm, at first, the nodes which are in neighborhood of the source node become wave . But for providing a delay- constraint Dmax must be first subtract from NODE-TRAVERSAL-Time(NTT) in every intermediate node before becoming as a *wave* node. Therefore two conditions maybe happen:

1. If the NTT is greater than the maximum(remaining) delay in delay field of QRREQ, the intermediate node must drop the QRREQ, and don't become a wave node.
2. If the NTT is less than the maximum delay in message, by subtracting from its value of the NODE_ TRAVERSAL_TIME, the intermediate node will become wave node and continue broadcasting the RREQ.

The wave nodes re-broadcast the message to their neighbors, and set up a reverse path to the sender, which is represented with pointing arrows in the Fig.4. These nodes further if provide delay constraint and have a wave node in their neighbors, become wave and re-broadcast this message and set up a reverse path to the nodes from which they received the message.

This process continues until the message reaches the destination node or the delay experienced by the packet exceeds the limit *Dmax.* Since there are more than one path from the sender to the destination, the destination may receive multiple QRREQ message for the same sender. However, the route through which the destination node receives the QRREQ message first is the shortest path between the sender and the destination which guarantees the Quality assurance, thus the destination replies to the first QRREQ message by sending a QRREP message using the reverse path set up when the QRREQ messages are forwarded. All the wave nodes that lie on this route between the source and the destination become the *path* nodes (represented with circles containing arrows in the Fig.4). All communications between the source and the destination from this point onwards takes place using this path until the topology of the network changes. All other wave nodes that don't see a path node in their neighbor pointing towards them, are sent a clear state message to move them from the *wave* state to a *clear* state.

The local states of each node are defined by adding one state as mentioned in [19] to the 14 states of the lee algorithm. These states are as follows:

Init (initial state of each node except source and destination), **Dead** (Dead Cell), **Init_S** (Initial State of the Source node), **Init_D** (initial state of the destination), **DR** (Destination Ready; state of the destination node after it has received a RREQ message), **WU** (Wave Up), **WD** (Down), **WR** (Right), **WL** (Left), **PU** (Path Up), **PD** (Path Down), **PR** (Path Right), **PL** (Path Left), **Clear** (final state of the node that received a wave message but is not going to become a path node)and **Found** (route found; final state of the Source node).With these set of states we consider Von-Neumann neighborhood (see Fig.2) in which each cell can communicate to all the cells on its top, down, left and right. The local transition rules for each cell based on these neighborhoods are defined according to the routing mechanism.

For implementing mobility we assume that nodes can move only to neighboring cells. A Moore neighborhood is chosen because it allows movement in more different directions than von Neumann neighborhood ,and as we are using a two-dimensional grid, each node can have no

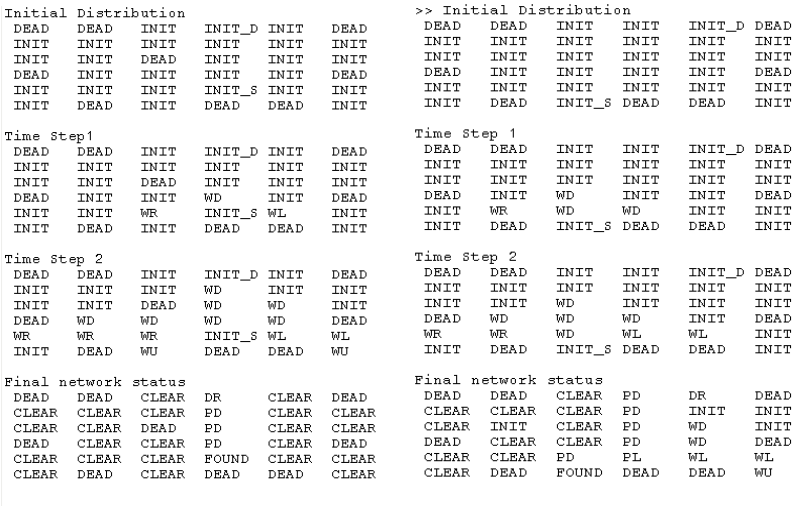


Fig.5*: Illustration of the 6*6 grid before mobility changes (a) and after movement of nodes (b).*

more than 8 nodes as neighbors (Fig.2). So in this experiment, a node can with deterministic speed, move around in random fashion to one of each surrounding cells, by selecting any direction of North, North-East, North-West, East, West, South-West, South-East and South. As cellular automata are discrete event time systems, in each time step, nodes move to one of neighboring cells and then routing algorithm is being performed. The execution of mobility rules is considered to be higher than routing rules, it means that first topology is changed then the process of finding the route between source and destination is performed.

Using this specification as a base, the model was implemented in mat lab[22]. Fig.5, shows the procedure of our method after 4 steps before the topology changes(a), and after that (b). As you see in the figure, the destination node (INIT_D) and the source node (INIT_S) will move to the east and south-east, respectively. It's obvious that dead nodes which are represented as DEAD notations, are static and didn't change their location after the topology have changed. The figure also shows that the routing mechanism is performed well as the path with adequate route is discovered.

Our proposed routing protocol with the help of cellular automata doesn't need to store accumulated delay for calculating the delay in phase of unicasting RREP instead of QoS-AODV, by defining only constant set of states .This leads to simply model and evaluate the behavior of routing algorithms in MANET.

4. Simulation and Results

To evaluate the performance of our method, we simulated our proposed method via mat lab software[22].The model will be tested by having different initial distributions of the node and checking whether the algorithm successfully determines the delay-constraint path between two nodes (if one exists).

We consider a rectangular network of 500m×300m of mobile nodes placed randomly in the area. Our simulation has been conducted at different pause times and speeds. We varied the pause time from 0 to 600 seconds which is chosen to be between high mobility rate (pause time 0 seconds) and no mobility (pause time 600seconds), and studied its effect on the performance of the routing protocols. The average node speed in this group of simulations is chosen to be 5 m/s. we also studied the effect of mobility. Node speeds are 5, 10, 15, 20 m/s. The pause time used for these scenarios was 60seconds. For both scenarios our protocol is introduced with and without QoS delay constraint. Delay constraints were chosen to be 0.1 and 0.3 seconds. Our method's performance is compared with AODV and The original QoS-AODV protocol.

5.1. Performance metrics

Two performance metrics, which are used for evaluating the performance of the routing protocols, are listed below.

- **Average end-to-end Delay**

This is the average end-to-end delay of talking parties in the simulation and it includes all possible delays caused by buffering during route discovery latency, queuing at the

interface queue, retransmission delays at the MAC, propagation and transfer times.

$$D = \frac{1}{S}\sum_{i=1}^{S} r_i - s_i$$

D: Average end-to-end delay S: Number of successfully received packets. i: Unique packet identifier. r_i : Time at which a packet with unique identifier i is received. Si: Time at which a packet with unique identifier i requests a route to be send.

- **Packet Delivery Fraction**

The fraction of successfully received packets which will be survived while finding their destination.Successful packet delivery is calculated such that all data packets with unique identifier leaving the source MAC are counted and defined as originating packets. Received packet identifiers are compared to collected transmission data and each unique packet is counted once to ensure prevention of counting excess receptions, which are mainly caused by multiple paths as a result of mobility. The result is the average of the ratio of uniquely received and all uniquely transmitted packets as seen in the following equation.

$$F = \frac{1}{C}\sum_{f=1}^{C} \frac{R_f}{T_f}$$

F: Fraction of successfully delivered packets. ***C*** : Total number of flows, connection. ***f*** : Unique flow id. R_f : Count of unique packets received from flow ***f***. T_f : Count of packets transmitted to flow ***f***.

Fig.6 shows the average end to end delay under various pause times. According to the graph, the average delay of all the protocols decreases as the time increases. When there is no delay constraint, the performance of our proposed method and QoS-AODV protocol is better than AODV with little bit differences. By Using delay constraints both QoS protocols have always better delay than AODV because they force the network to satisfy certain delay constraints, so the delay achieved is always less than or equal to the delay required even for high delay constraints (low delay bound 0.1 seconds) .For both protocols on the average, the delay achieved is half that required, but the delay of our method at higher pause times is much lower than the QoS-AODV routing protocol. The low end to end delay of packets ensures the on time transmissions required by real time traffic transmissions. As is shown in Fig.7 the packet delivery fractions are almost the same for AODV protocol, QoS-AODV, and the proposed routing protocol without delay constraints and with high delay bounds (0.3 seconds).QoS-AODV routing protocol and our method with low delay bound, have low performance by having a low packet delivery fraction, however, our method has better ratio (about 1.2%). Indeed, with delay constraints especially in delay bound 0.1 seconds, more packets are being dropped because the routes available for them do not satisfy the QoS requirements.

Simulation results also show that when mobile nodes moved at higher mobility speeds, our method with no delay constraint has average 95% packet delivery fraction, which is almost similar to the QoS-AODV (with no delay constraint) and AODV (Fig.8). The performance of our method and QoS-AODV with delay constraints also drops to provide QoS assurance. The proposed method outperforms slightly QoS AODV with 0.1 and 0.3 delay bound.

Fig.9 shows the average delay in different speeds. From Fig.9 we notice that, higher mobility leads to a higher delay for all protocols. Although the average delay increases as the mobility changes, the delay achieved is still better than required for both our protocol and QoS-AODV with different delay bounds.

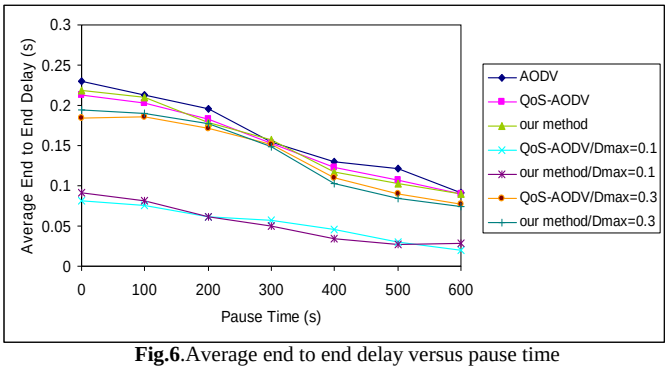


Fig.6.Average end to end delay versus pause time

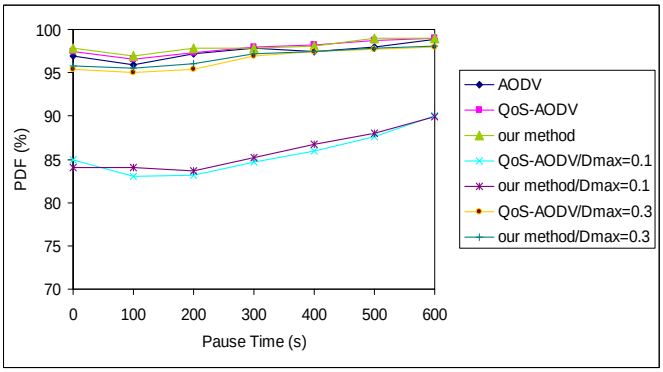
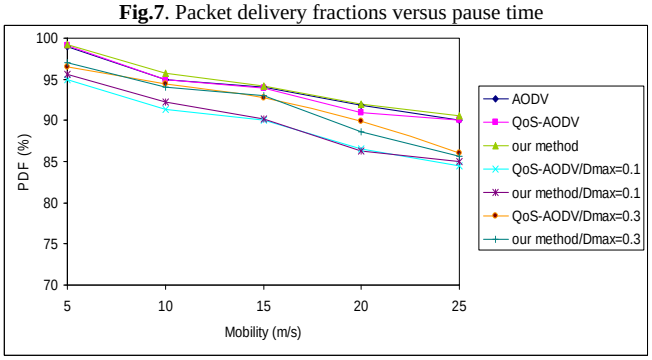


Fig.7. Packet delivery fractions versus pause time



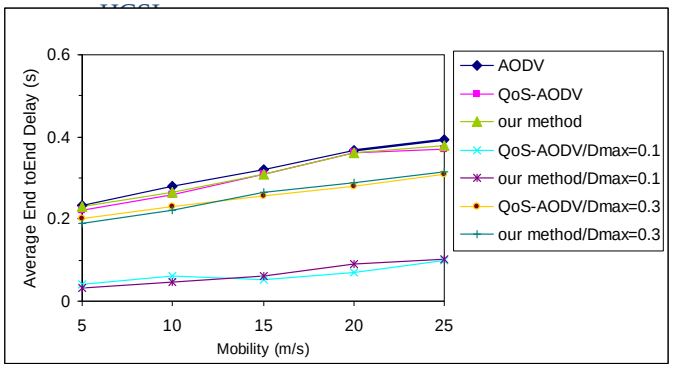


Fig. 8. Packet delivery fractions versus mobility

Fig.9. Average data packet delay versus mobility

7. Conclusion

Number of problems can be solved if they are described as cellular automata. One of the NP-complete problems is the problem of providing a shortest route with high quality of service among two communicating mobile nodes in MANET. In this paper we presented a new routing algorithm in mobile ad-hoc networks using cellular automata by taking delay into account as the QoS parameter. The variant of Lee routing algorithm for finding the shortest path between two points was modified into a more efficient form, which could find the on demand delay constraint path like QoS AODV but with lower consuming resources only by defining some infinite states. The Two following performance metrics are used to compare the performance of the protocols: (a) average end-to-end delay, (b) packet delivery fraction, in various pause times and mobility. The obtained results indicate that cellular automata can be used with success to simulate mobile ad-hoc networks and the proposed protocol performs well in supporting the QoS feature.

For further work, another extension like a bandwidth extension can be used to satisfy minimum bandwidth requirements and other optimization can be taken into consideration like reliability of the route.

References

[1] T.S. Rappaport. “Wireless Communications: Principles and Practice”, Second Ed., Prentice Hall, 2002

[2] Chen Niansheng,Li Layuan. Research on the Basis of QoS Routing Protocol of Ad Hoc Network. DCABES 2004Proceedings. Wuhan:Hubei Science and Technology Press,2004:206-210.

[3] Niansheng Chen, Layuan Li, Zongwu Ke. A Multicast Routing Algorithm of Multiple QoS for Mobile Ad Hoc Networks. International Symposium on Distributed Computing and Applications to Business, Engineering and Science, DCABES 2007, PROCEEDING: 301-305.

[4] C. Perkins, E. Belding-Royer, S. Das, "Ad-hoc On Demand Distance Vector (AODV) Routing", IETF Network Working Group, RFC 3561, July 2003.

[5] C.K. Toh, A novel distributed routing protocol to support ad-hoc mobile computing, Proceedings of the fifteenth IEEE Annual International Phoenix Conference on Computers and Communications, March, 1996 pp. 480–486.

[6] D.B. Johnson, D.A. Maltz, Dynamic Source Routing in Ad-hoc Wireless Networks, Kluwer, 1996.

[7] V. Park, M.S. Corson, A highly adaptive distributed routing algorithm for mobile wireless networks, Proceedings of the 1997 IEEE INFOCOM, Kobe, Japan, April, 1997 pp. 1405– 1413.

[8] C. Perkins and E. Royer, "Quality of service for ad hoc on-demand distance vector routing," in Zntemet draft, draft-perkins-manet-aodvqos-02.txt, Oct.2003. [Online]. Available:http://people.nokia.net/ harliep/txt/aodvid/qos.txt

[9] A.M. Abbas, O. Kure, "A Quality of Service in Mobile Ad-hoc Networks: A Survey", International Journal of Ad hoc and Ubiquitous Computing (IJAHUC), vol. 6, no. 2, pp.75- 98, June 2010.

[10] G.Snthi and A.Nachiappan, "A Survey of QoS Routing Protocols for Mobile Ad hoc Networks", International Journal of Computer Science and Information Technology (IJCSIT),vol.2,No.4, pp.125-135, August 2010.

[11] S. Wolfram "A new kind of science". Wolfram Media,Inc. 2002.

[12] B. Chopard and M. Droz. Cellular Automata Modeling of Physical Systems. New York: Cambridge Univ. Press 1998.

[13] C. Hochberger; R. Hoffmann. "Solving routing problems with cellular automata", in Proceedings of the Second conference on Cellular Automata for Research and Industry, Milan, Italy. 1996.

[14] L. Bajaj, M. Takai, R. Ahuja, K. Tang, R. Bagrodia, and M. Gerla. "GloMoSim: A Scalable Network Simulation Environment". UCLA Computer Science Department Technical Report 990027,May 1999.

[15] Kevin Fall, Kannan Varadhan, Eds. "The ns Manual (formerly ns Notes and Documentation)", http://www.isi.edu/nsnam/ns/. Accessed February 2007.

[16] R. Subrata and A.Y. Zomaya. "Evolving Cellular Automata for Location Management in Mobile Computing Networks". IEEE Trans. on Parallel and Distributed Systems 14:1 (Jan 2003), 13- 26.

[17] Michael Kirkpatrick and Frances L. Van Scoy. Using cellular automata to determine bounds for measuring the eficiency of broadcast algorithms in highly mobile ad hoc networks. In ACRI, pages 316-324, 2004.

[18] R.O. Cunha, A.P. Silva, A.A.F. Loreiro, and L.B. Ruiz. “Simulating large wireless sensor networks using cellular automata”, Simulation Symposium, in Proceedings of the 38th Annual, pages 323-330, 4-6 April 2005.

[19] U. Farooq, G. Wainer, and B. Balya, " DEVSmodeling of mobile wireless ad hoc networks", In Simulation Modelling Practice and Theory 15 of Elsevier, pp.285–314, December2007.

[20] C. Y. Lee, "An algorithm for path connections and its applications", in IRE Transaction on Electronic Computers, pp. 345-365, Sep. 1961.

[21] P. SARKAR, "A Brief History of Cellular Automata", ACM ACM Computing Surveys,Vol.32, No.1, pp.80-107,March 2000.

[22] The Matlab Simulator, http://www.mathworks.com.