International Journal of Grid and Distributed Computing
Vol. 9, No. 2 (2016), pp.255-272
http://dx.doi.org/10.14257/ijgdc.2016.9.2.22

An Energy Aware Cellular Learning Automata Based Routing Algorithm for Opportunistic Networks

Feng Zhang*, Xiaoming Wang, Peng Li and Lichen Zhang

School of Computer Science, Shaanxi Normal University
Xi'an, Shaanxi 710062, China
zhangfeng@snnu.edu.cn

Abstract

Message transmission in opportunistic networks is accomplished via the encounters of mobile nodes while moving around. The distributing of nodes greatly impacts the performance of message delivery ratio due to their sparse encounter opportunities. Nodes with exhaust energy can't participate in message transfer process. So it is very meaningful to make nodes energetic and balance the energy consumption between nodes. In this paper, a novel dynamic irregular cellular multiple learning automata (DICMLA) model and the corresponding routing algorithm are proposed to optimize the energy consumption of nodes. The proposed routing algorithm utilizes the characteristics of cellular learning automata to reduce the energy consumption of nodes and improve the delivery ratio of message transmission. The simulation results show that the proposed algorithm can obviously balance energy consumption of nodes and thus prolong the lifetime of the network.

***Keywords**: Opportunistic networks, Routing algorithm, Cellular learning automata (*CLA*), Energy efficient*

1. Introduction

Opportunistic networks are composed of a number of mobile nodes with the ability of wireless communication in a short range. Due to the movement of nodes, most of the connections between nodes are temporary and unpredictable. The traditional routing protocols are not suitable for this kind of environment because they normally assume that the end-to-end connection from the source to the destination is stable during message transmission[1]. However, different from the conventional networks, the unreliable connections and the unpredictable contacts make routing become a great challenge in the opportunistic networks. To deal with these issues, a new message transmission fashion of store-carry- forward is applied.

Unlike the traditional networks, devices in the opportunistic networks may be very small, and have a very limited processing and storage capability. Moreover, they have limited energy resource and can't be charged up all the time[2]. Usually, devices with energy exhausted cannot participate in message transmission any more, so they are considered as dead nodes. With the increase number of dead nodes, encounters between nodes become more difficult. This will lead to a decline of the message delivery ratio. Thus energy consumption is an important factor for routing in the opportunistic networks, especially in certain scenarios such as disasters in which the electrical infrastructure is damaged, faults in the electrical grid, travel through remote areas like mountains or deserts where the access to an electrical infrastructure is rare or completely lacking [3].

Nodes in opportunistic network have to consume energy to send, receive and store messages, so does performing computation[4]. It is energy efficient to reduce the transmission time during the routing phase. Consequently routing strategies can optimize energy consume by using energy-limited nodes sparingly[5]. So it is a good way to balance the energy consumption and messages delivery probability, and take the energy consumption of all the nodes into account for messages routing.

The remainder of the paper is organized as follows: Section 2 describes related works. A dynamic irregular cellular multiple learning automata(*DICMLA*) model is presented in section 3. Section 4 discusses the proposed algorithm. Simulation results are explained in section 5, and section 6 shows the conclusions of this work.

2. Related Works

Recently, much research in opportunistic networks has focused on routing. To deal with the unpredictability of connections between nodes and the network partitions, the existing routing protocols can be classified as forwarding based and flooding based [6].

In the forwarding based routing protocols, there is only one copy of message in the network, such as Direct Delivery routing protocol. How to utilize the contacted nodes to get better delivery ratio is the main scope of these protocols. These strategies require more information about the network than the other. The disadvantage of forwarding based routing protocols is that they usually have lower deliver ratio and cannot adapt to different networks or conditions. So they may not make optimal decisions. At the other end of the spectrum, a node might need to know the complete future schedule of every contact in the network [7].

On the contrary, in the flooding based routing protocols, there are many replications of message in the network, so much more energy consumed. How to reduce the number of replications of message is a challenge topic in this type of routing protocol. It is clear that the more copies of message can increase the delivery ratio and decrease the delay time. However it also requires much more energy consumption as well as bandwidth and storage resource. In the Epidemic routing protocol [8], messages are copied to every node which it contacts. So the number of copies is maximized. This is very helpful for optimal delivery ratio and delivery latency, but huge resource consumed. Different from the Epidemic routing protocol, messages are only sent to the destination in the Direct Delivery routing protocol [9]. There is only one copy of message in the network and the resource utilization is the maximum. This is a clear trade-off between cost and performance in the Spray and Wait[10] an PRoPHET [11] routing protocols. Spray and Wait is a combination of Direct Delivery and Epidemic in which the copies of messages is limited to a fix value. In the ProPHET, messages are only copied to neighbor nodes only if they have a higher value of delivery probability to destination.

Recently, energy performance evaluation and comparison have been performed, investigated and compared for various already existing routing protocols in terms of energy consumption [12] [13]. Yet there are little works focusing on the energy efficiency in these routing protocols. In this paper, we introduce cellular learning automata to deal with energy efficiency of nodes through continuously messages exchange between neighbors. Messages are only replicated to the preferred neighbors for routing based on considering the delivery probability to the destination and the energy consumption of surrounding neighbors. The message forwarding probability is updated according to the learning atuomata for reward and penalty. At the end of this paper, we simulate our routing algorithm and other three classic flooding based routing protocols. The results show that our algorithm gets the better performance in the energy consumption than the three protocols.

3. Dynamic Irregular Cellular Multiple Learning Automata (DICMLA)

In this section we propose the DICMLA model based on the analysis on the learning automaton (LA), cellular learning automaton (CLA), irregular CLA (ICLA) and dynamic irregular CLA (DICLA).

3.1 Learning Automata

Learning Automata (*LA*) is an adaptive decision-making mathematical model which operates on an unknown random environment. A learning automata has a finite set of actions that could be chosen at each stage and the choice (action) depends upon an action probability vector. For each chosen action, the environment gives a reinforcement signal with probability distribution. The automaton then updates its action probability vector depending upon the reinforcement signal at that stage, and evolves to some final desired behavior. A class of learning automata is called variable structure learning automata and is represented by quadruple (a, b, p, T) in which $a = \{a_1, a_2, \ldots, a_r\}$ represents the action set of the automata, $b = \{b_1, b_2, \ldots, b_r\}$ represents the input set, $p = \{p_1, p_2, \ldots, p_r\}$ represents the action probability set, and $p_{(n+1)} = T[a_{(n)}, b_{(n)}, p_{(n)}]$ represents the learning algorithm. Let a_i be the action chosen at time n, then the recurrence equation for updating p is defined as:

$$\begin{gathered} p_{\mathrm{i}}(n+1)= p_i(n)+a*\big(1-p_i(n)\big) \\ p_j(n+1)= p_j(n)-a*p_j(n)) \quad \forall j \quad j\neq i \end{gathered} \tag{1}$$

for favorable responses, and

$$\begin{gathered} p_i(n+1)=(1-b)*p_i(n)) \\ p_j(n+1)=\frac{b}{r-1}-((1-b)*p_j(n)) \quad \forall j \quad j\neq i \end{gathered} \tag{2}$$

for unfavorable ones, where a and b are reward and penalty parameters respectively. If $a = b$, learning algorithm is called $L_{R\text{-}P}$, if $a << b$, it is called $L_{R\varepsilon P}$, if b=0, it is called $L_{R\text{-}I}$ [14].

3.2 Cellular Learning Automata

Cellular learning automata (*CLA*) is a combination of cellular automata (*CA*) and learning automata (*LA*)[15]. It is a powerful mathematical model for dynamic complex system. The basic idea of the *CLA* is to utilize learning automata to adjust the state transition of *CA*. The learning automata residing in a particular cell determines its action on the basis of its action probability vector. The neighbor *LA*s of any specific *LA* constitute the local environment of that cell in which that *LA* resides. Like *CA*, there is a local rule that the *CLA* need to comply with. The local rule of the *CLA* and the actions selected by the neighbor LAs of any specific *LA* determine the reinforcement signal to the *LA* residing in a cell. The local environment of a cell is non-stationary because the action probability vectors of the neighbor *LA*s vary during evolution of the *CLA*. *CLA* has been applied in many wireless sensor networks fields, such as energy based routing algorithm in WSN [16], deployment strategy for mobile wireless sensor networks [17].

3.3 Irregular CLA

Irregular cellular learning automata (*ICLA*)[18] is a cellular learning automata (*CLA*) in which the restriction of the rectangular grid structure in traditional *CLA* is removed. This generalization is expected because there are applications which cannot be adequately modeled with regular grids, such as wireless sensor networks, opportunistic networks etc. An *ICLA* is defined as an undirected graph in which, each vertex represents a cell which is equipped with a learning automata. Like *CLA*, there is a rule that the *ICLA* operates under. The neighbor *LA*s of any particular *LA* constitute the local environment of the cell in which that *LA* resides. The rule of the *ICLA* and the actions selected by the neighbor *LA*s of any particular *LA* determine the reinforcement signal to that *LA*. The operation of *ICLA* is identical to the operation of *CLA*. *ICLA* is recently used as a learning model in a clustering algorithm for wireless sensor networks [19]. Despite its irregular structure, *ICLA* operation is equivalent to that of *CLA*. *ICLA* has found a number of applications in wireless ad hoc and sensor networks.

In some cases the adjacency matrix of the underlying graph of the *ICLA* can be changed over time. This dynamicity is necessary in many applications such as mobile ad hoc and sensor networks [20]. And this kind of *ICLA* is called as the dynamic irregular cellular learning automata (*DICLA*).

3.4 Dynamic Irregular Cellular Multiple Learning Automata

The above models of learning automata use only one *LA* per cell. In some case every cell needs to be equipped with several *LA*s [21]. For instance, the packet forwarded in opportunistic networks may be sent to several destination nodes, each of which can be adapted by a *LA*. So the model of *DICLA* with multiple *LA*s(*DICMLA*) is proposed for this scenario.

We define *DICLA* with multiple *LA*s in each cell (*DICMLA*) as an undirected graph in which, each vertex represents a cell and multiple learning automaton assigned to every cell (vertex). A finite set of interests is defined for the *DICMLA*. For each cell of *DICMLA* a tendency vector is defined whose *j*th element shows the degree of tendency of the cell to the *j*th interest. In *DICMLA*, the state of each cell consists of two parts: the actions selected by the learning automaton and the tendency vector. Two cells are neighbors in *DICMLA* if the distance between their tendency vectors is smaller than or equal to the neighborhood radius.

Like *DICLA,* there is a local rule under which *DICMLA* operates. The rule of *DICMLA* and the actions selected by the neighbor learning automata of any particular learning automaton LA_i residing in cell c_i determine the followings:

●The reinforcement signal to the learning automata $LA_{i..}$

●The restructuring signal to the cell c_i in which *LAi* resides. The restructuring signal is used to update tendency vector of the cell.

Dynamicity of *DICMLA* is the result of modification made to the tendency vectors of its constituting cells. Figure.1 gives a schematic of *DICMLA*. A *DICMLA* is formally defined as follows.

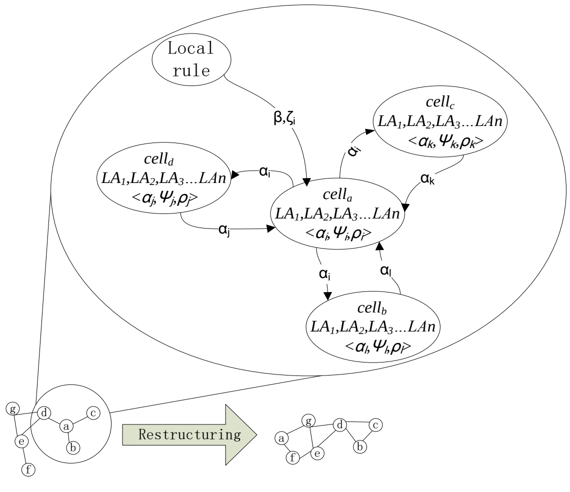

Figure 1. Dynamic irregular cellular multiple learning automata

Definition 1: A dynamic irregular cellular multiple learning automata (*DICMLA*) with *s LA*s in each cell is a structure $A = (G<E, V>, \Psi, A_s, \Phi<\alpha, t_\Psi>, \tau, F, Z, \rho)$ where

1. *G* is an undirected graph, with *V* as the set of vertices and *E* as the set of edges. Each vertex represents a cell in *DICLA*.

2. Ψ is a finite set of interests. Cardinality of Ψ is denoted by $|\Psi|$.

3. A_s is the set of LA_s assigned to *DICMLA*, where A_i is the vector of *LA*s assigned to cell *i*, *s* is the number of *LA*s assigned the cell *i* ;

4. $\Phi(\alpha_{j,i}, t_{\Psi,i})$is the cell state. State of a cell c_i (Φ_i) consists of two parts; 1). $\alpha_{j,i}$ which is the action selected by the *j*th learning automata of that cell *i*, and 2). A vector $t_{\Psi,i} = (t_{1,i}, t_{2,i},\ldots, t_{|\Psi|,i})^T$ which is called the tendency vector of the cell *i*. Each element $t_{k,i} \in [0,1]$ in the tendency vector of the cell *ci* shows the degree of tendency of c_i to the interest $\psi_k \in \Psi$.

5. τ is the neighborhood radius. Two cells c_i and c_j of *DICMLA* are neighbors if $\| t_{\Psi,i} - t_{\Psi,j} \| \le \tau$. In other words, two cells of *DICMLA* are neighbors if the distance between their tendency vectors is smaller than or equal to τ .

6. $F: \Phi_{sxi} \rightarrow <\beta, [0,1]^{|\Psi|}>$ is the local rule of *DICMLA* in each cell c_i, where $\Phi_{sxi} = \{\Phi_{sxj} | \| t_{\Psi,i} - t_{\Psi,j} \| \le \tau\} + \{\Phi_{sxi}\}$ is the set of states of all neighbors of c_i, β is the set of values that the reinforcement signal can take, and $[0, 1]^{|\Psi|}$ is a $|\Psi|$-dimensional unit hypercube. From the current states of the neighbor cells of each cell c_i, the local rule performs the following: 1). Gives the reinforcement signal to the learning automata LA_i residing in c_i, and 2). Produces a restructuring signal ($\zeta = (\zeta_1, \zeta_2,, \ldots, \zeta_{|\Psi|})^T$) which is used to change the tendency vector of cell c_i. Each element ζ_i of the restructuring signal is a scalar value within the close interval [0,1].

7. $Z : [0,1]^{|\Psi|} \times [0,1]^{|\Psi|} \rightarrow [0,1]^{|\Psi|}$ is the restructuring function which modifies the tendency vector of a cell using the restructuring signal produced by the local rule of the cell.

8. ρ is an $|\Psi| \times s$-dimensional vector called as activation probability vector, where $\rho_{i,j}$ is the probability that the LA_j in cell *i* (for $i = 1,\ldots,|\Psi|$ and $j = 1,\ldots,s$) is to be activated in each stage.

The operation of *DICMLA* takes place as the following iterations process. At iteration *k*, each learning automata in the cell activated with probability $\rho_{i,j}$, and the activated *LA*s choose one of their actions. The activated automata use their current actions to execute the rule to compute the reinforcement signal. The actions of neighbor cells of an activated cell are their most recently selected actions. The reinforcement signal is produced by the application of local rule. The local rule computes the reinforcement signal for each *LA* based on the actions selected by the neighbor *LA*s. Then, each learning automata receives a reinforcement signal. Let $\beta_i \in \beta$ be the reinforcement signal received by LA_i. This reinforcement signal is produced by the application of the local rule $F\ (\Phi_{sxi}) \rightarrow <\beta, [0,1]|^{\Psi}|>$, then every *LA* updates its action probability vectors on the basis of the supplied reinforcement signal and the action chosen by the cell. Next, each cell c_i updates its tendency vector using the restructuring function *Z* (Eq. (3)). And the process repeats.

$$t_{\Psi,i}(n+1) = Z(t_{\Psi,i}(n), \zeta(n)) \tag{3}$$

The cells may be activated in either a time-driven or event-driven manner. In time-driven asynchronous *DICMLA*, each cell is assumed to have an internal clock which wakes up the *LA*s associated to that cell while in event-driven asynchronous *DICMLA* a cell is selected in fixed or random sequence.

4. Proposed Routing Algorithm

In this section, we will propose corresponding novel routing algorithm based on the DICMLA model.

4.1 Network Model

In the proposed routing algorithm, an event-driven asynchronous *DICMLA*, which is isomorphic to the opportunistic network topology, is created. For an opportunistic network with *N* nodes, each node is considered as a cell c_i . The number of neighbors of the cell c_i is the interest set of nodes. The neighborhood radius (τ) of node c_i is equal to the communication range of nodes. Every node has *N*-1 learning automaton that are used to construct the message forwarding probability to other *N*-1 destination nodes via its neighbors. The learning automata in each cell c_i referred to as $LA_{i,j}$, (j=0,1,2,….n-1), in which *j* is the number of destination nodes of messages buffered on this node. The number of actions of each $LA_{i,j}$ is equal to the number of neighbors of the cell; The actions which learning automata can take means to duplicate message to its neighbors, and the action probability is called as forwarding probability is combined with the delivery probability and energy level of all neighbors of the node. When acknowledge packet received from its neighbors, the energy level of neighbors enclosed in the acknowledge packet will be used to update the new value of forwarding probability according the learning automate principle. The probability of selecting each action is initialized and updated, and the process will be described in the following section.

According to the definition of *DICMLA*, values of tendency level are used along with the neighborhood radius of *DICMLA* to specify the neighbor cells of a cell. But in the opportunistic network, the tendency of cells can't be estimated and calculated due to the movement of nodes in the opportunistic network is randomly. This does not cause any problems since neighbor cells of each cell are implicitly specified according to the topology of network for a *DICMLA* model which is mapped into opportunistic network.

4.2 Proposed Routing Algorithm

This algorithm is specific for the topology of opportunistic networks in which one node moves into its communication scope, and its neighbor nodes can be organized as an irregular cell automata. Next we introduce two useful definitions used in the algorithm.

Definition 2: Cell Neighbors List. The list is used to record the neighbors of node and exists in every node of the network. The number of neighbors dynamically changes along with the movement of nodes. The list also stores the node identification and current energy of each neighbor.

Definition 3: Forward Probability Table. This table stores the forward probability of node to other destination nodes in the network via its neighbors. Each entry also stores the update time of this forward probability.

For an opportunistic network with *n* nodes, node *a* has four neighbors *b*,*c*,*d*,*e*, so there are five entries in the table include itself. And each entry contains *n* values for other nodes. The forward probability table on node *a* can be demonstrated as shown in Table 1.

Table 1. Forwarding Probability Table

	a	*b*	*c*	*d*	*e*	………	*n*
a	*N/A*	F_{ab}	F_{ac}	F_{ad}	F_{ae}	………	F_{an}
b	F_{ba}	*N/A*	F_{bc}	F_{bd}	F_{be}	………	F_{bn}
c	F_{ca}	F_{cb}	*N/A*	F_{cd}	F_{ce}	………	F_{cn}
d	F_{da}	F_{db}	F_{dc}	*N/A*	F_{de}	………	F_{dn}
e	F_{ea}	F_{eb}	F_{ec}	F_{ed}	*N/A*	………	F_{en}

In the proposed routing algorithm each node in the network is equipped with multiple learning automata. The number of automata in each node is N-1, for opportunistic network is consist of N nodes. Each learning automata determines the probability path from this node to the other destination nodes. Furthermore each learning automata has r actions {*α1, α2, …, αr* }, where r is the number of neighbor nodes. Each action has a probability which is calculated by learning automata to change the value of the forward table for this node.

We have used four different packet types namely ASK packet, ENERGY packet, DATA packet and ACK packet, and three phase named *Cell Construction Phase* **(CC),** *Learning and Routing Table Construction Phase* **(LR),** *Forwarding and Learning Phase* **(FL).**The ASK and ENERGY packets are used only in second phase, ACK and DATA packets are needed in *Forwarding and Learning Phase*. These phases are described as follows:

Cell Construction Phase

In this phase, all nodes in the network are moved according to their patterns. Each node in the network is considered as a cell of DICMLA. Nodes will form neighbors when they move into their communication range and setup connections of neighbors. After connection established between the neighbors, nodes can exchange information，send messages and receive responses from their neighbors. The connections between nodes are dynamically changed and this character is very similar to the reconstruct of cell in the *DICMLA*. So dose the topology of network changes along with continuous movements of nodes in the opportunistic network.

Learning and Routing Table Construction Phase

There is a local forwarding table established by each node. This table contains the forwarding probability to the destination node if message relay through each neighbor of the node. For each destination node, there is a learning automata equipped and the forwarding probability updated by the leaning automata according to the acknowledgements of its neighbors. All automata are activated at the same time as described in the following parts.

Whenever two nodes meet, the connection is established between them. After the cell construction phase, the initial forwarding probability for the encounter node calculated as follows:

1. Node a and node b are encounter when they are moving in their manner.

2. Node a adds node b as an element in its cell neighbor list, and calculates the delivery predictions for this host as Eq. (4) as follows.

$$P_{(a,b)}=P_{(a,b)old}+（1-P_{(a,b)old}）*P_{init} \tag{4}$$

Where P_{init} is an initial constant.

3. Node a updates the transitive delivery predictions for this host, Transitive

affection is shown in Eq. (5), where β is a scaling constant that decides how large impacts the transitivity should have on the delivery predictability.

$$P_{(a,c)} = P_{(a,c)\ old} + (1 - P_{(a,c)\ old}) * P_{(a,b)} * P_{(b,c)} * \beta \tag{5}$$

4. Node a sends an ASK packet to its neighbor node b, requesting the energy information and forwarding table of node b.

5. On receiving the ASK packet from node a, node b sends an ENERGY packet to node a, including the residual energy and the forwarding information of node b.

6. Node a updates the energy of node b and updates local forwarding probability table of node b.

7. Node a updates the forwarding probability for other nodes if the updated probability is more recent than that in node a.

8. The forwarding probability $F_{(a,b)}$ of node a is recalculated according to the residual energy of neighbors as Eq. (6).

$$F_{(a,b)} = P_{(a,b)} + \left(1 - P_{(a,b)}\right) \frac{(1-p_{(a,b)})EnergyLevel_i}{\sum_{j=1}^{n} EnergyLevel_j} \qquad n = 1,2,\dots r \tag{6}$$

Where $P_{(a,b)}$ is delivery probability calculated in Eq. (4), and n is the number of neighbors of node a.

Forwarding and Learning Phase

Every node has constructed its forwarding table after the above two phrases. When there are messages needed to be relayed, the neighbor nodes which have the greater value of probability to destination are referred as the next hop of the message. When the transfer finished, the node gets an ACK message from these neighbors, and one of the following steps is performed:

●If the value of energy level in received ACK packet is higher than or equal to the average energy of all of the neighbors shown in Eq. (7), this action is rewarded and the action vector probability in the routing table value is updated by all the learning automatons residing in the node according to the $L_{R\text{-}P}$ learning algorithm.

$$EnergyLevel_{ACK} \geq EnergyAvg_{\ i} \qquad i = 1,2,\dots n \tag{7}$$

●If the value of energy level in the ACK packet is lower than a threshold, Eq. (8), the action selected by the sender node is not an appropriate action and therefore must be penalized.

$$EnergyLevel_{ACK} < EnergyAvg_{\ i} \qquad i = 1,2,\dots n \tag{8}$$

In Eq. (7) and (8) $EnergyLevel_{ACK}$ is the energy level of ACK packet carries and *EnergyAvg* is the average of energy from other nodes in routing table that has computed from Eq. (9) where n is the number of nodes of current *DICMLA* environment.

$$EnergyAvg_i = \frac{\sum_{j=1}^{n} EnergyLevel_j}{n-1} \qquad j \neq i \tag{9}$$

The acknowledgement packets received from other nodes during this phase help the system to learn the best route in energy consumption. The packets are routed according to the forwarding probability in the forwarding tables and nodes continuously learn through this process.

5. Simulation

In this section, we compare our proposed routing algorithm with other three flooding based routing protocols by simulation focusing on the energy consumption and end to end delay. The three protocols are PRoPHET, Epidemic, and SprayAndWait respectively.

5.1 Simulation Scenarios and Parameter Settings

We use the Opportunistic Network Environment (ONE) [22] [23] software for the simulation. The EnergyLevelReport available in ONE is used to present the energy related results for all the protocols in this work [24]. We use the shortest path map based movement mobility models and four group of nodes respectively. The parameters of all four groups are listed in Table 2. The energy settings of all groups are the same and described as the following in Table 3.

Table 2. Simulation Environment Parameters

Simulation parameters	Simulation values			
	Group1	Group2	Group3	Group4
Map size	4500m x 3400m			
Node movement	Shortest path map based movement			
Node speed	0.5-1.5 m/s	1.5-5 m/s	5-10m/s	7-13 m/s
Paused time	0-120 sec.	0-120 sec	10-30 sec.	10-30 sec
Buffer size	5MB	5MB	50MB	50MB
Transmit speed	250kBps (2Mbps)			
Transmit range	10m			
Message size	500KB- 1MB			
Message TTL	300m			
Simulation time	12h			

Table 3. Energy Settings

Energy parameter	Value(units)
initialEnergy	5000 units
scanEnergy	0.1 units
transmitEnergy	0.2 units
scanResponseEnergy	0.1 units
baseEnergy	0.01 units

5.2 Metrics

We focus on the following performance metrics in our simulation:

a) Average residual energy: This is the average of the nodes energy left after the completion of the simulation.

b) Variance of residual energy among nodes: This is shown by the dispersion of residual energy of nodes after the completion of the simulation.

If the residual energy of nodes becomes almost zero i.e. below 50 units, we can consider them as a dead node, since they cannot forward any message in the network.

5.3 Simulation Results and Analysis

5.3.1. Performance under different network size： According to figure 2 when the number of nodes increases in the network, energy consumption increases，and the average residual energy of nodes decreases. This is due to the fact that the increase of nodes result in the increase of messages generated and delivered, and the number of transmissions and scans of nodes also increases. The rate of decrease is higher in the cases of Epidemic, and PRoPHET routing protocol as compared to the Spray& Wait and our algorithm. Our algorithm has the highest average residual energy among all the protocols. This is because that in our algorithm, we take the average energy of the neighbor nodes into account before nodes want to transmit messages. Spray& Wait limits the copy of message, thus a small number of scans and transmissions with other nodes take place which results in low energy consumption.

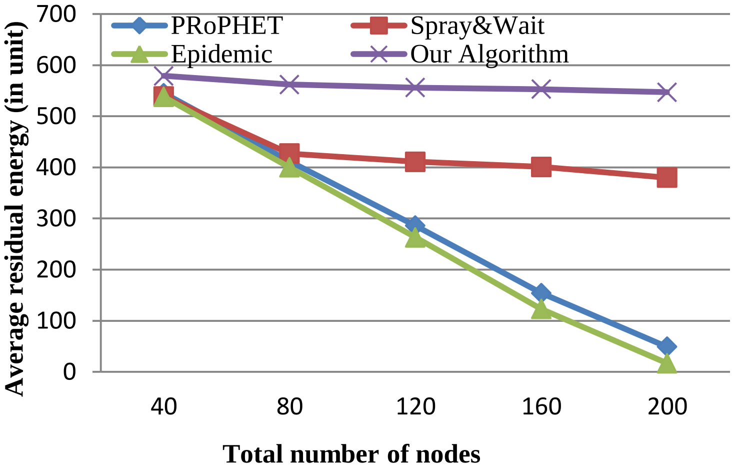

Figure 2. Average Residual Energy vs. Total Number of Nodes

In figure 3, it can be observed that as the number of nodes increases, the variance of residual energy of nodes in the network also increases. The rate of increase of, Epidemic, ProPHET and Spray& Wait routing protocol is much higher than our algorithm. This is because that the encounter and messages transfer between two nodes is not evenly distributed. The nodes with higher encounter the energy consumes is more, and the variance of residual energy becomes large. There exist dead nodes (the Residual Energy is almost zero) after simulation is done, when the number of nodes achieve 200, in case of Epidemic, PRoPHET and Spray& Wait routing protocol, variance of residual energy decline. Basically the variance of energy highlights the spread of energy from the average energy of the protocols. Since the variance is low, residual energy of most of the nodes will have their residual energy closer to the average energy. This ensures that the life time of the nodes will approximately be the same, and the network connectivity can be maintained for a longer period.

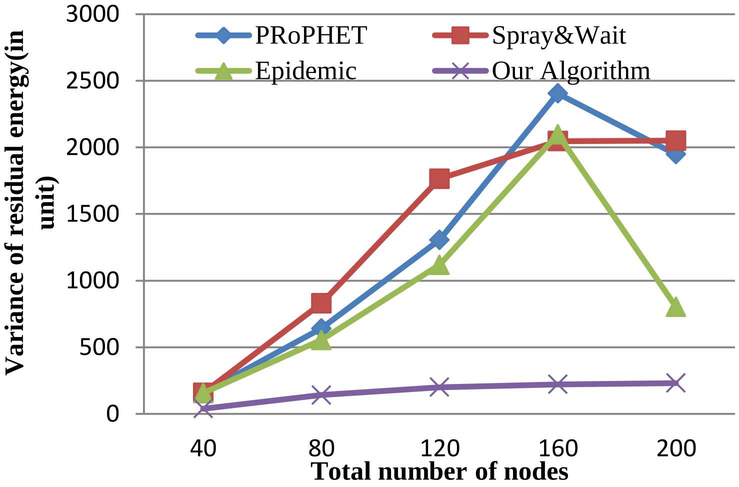

Figure 3. Variance of Residual Energy vs. Total Number of Nodes

5.3.2. Performance under different message size: According to figure 4, it is very clear that the average residual energy decreases when the message size increase. This is because the fact that with the increase in message size more number of bytes gets transmitted which consumes more energy of the participating nodes. The rate of decrease is more in the cases of Spray & Wait, but the average residual energy of Epidemic and ProPHET is less than the other two. And it can be seen that our routing algorithm has the highest average residual energy level compared to the other three routing protocols.

From figure 5, it is clearly seen that variance of residual energy between nodes are all increased as the message size increases. The three existing routing protocols have variance which is almost an order of magnitude higher than our algorithm. The main reason for small variance of our algorithm is that our protocol ensures that whenever message transmits occur for a node, both delivery ratio and remaining energy are considered. If remaining energy of specific node exceed the average energy of neighbor nodes, some other nodes takes up its responsibility of forwarding the packets to the destination.

5.3.3. Performance under different message generation interval: In figure 6, it is observed that the value of average residual energy increases with an increase in the message generation interval. This is justified by the fact that with increase in the message generation interval the total number of messages flowing in the network decreases. This result in a lesser number of scans and lesser number of messages transferred between the nodes, and hence less energy gets consumed. The rate of increase is highest for the Spray & Wait and lowest for the Epidemic. The value of residual energy is maximal in our algorithm protocol and minimal in Epidemic. Any other protocols performance is in between these protocols.

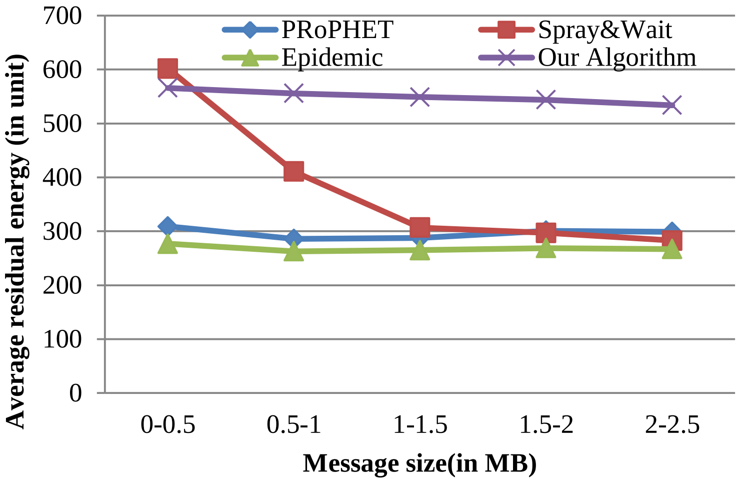

Figure 4. Average Residual Energy vs. Message Size

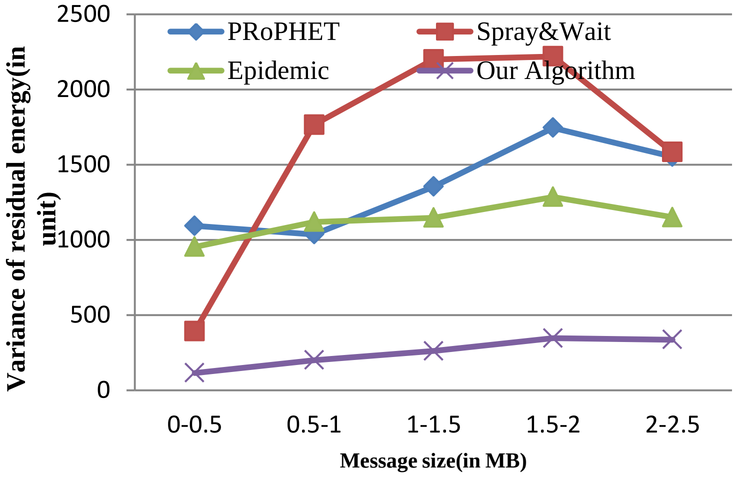

Figure 5. Variance of Residual Energy vs. Message Size

Another encouraging trend observed in figure 7 is that the variance of residual energy decreases in our algorithm with increase in the message generation interval. And other routing protocol is increases. This is due to the fact that with a decrease in the message generation, the whole number of messages in the network is decrease, the lesser nodes will be active to participate in the message relay and the number of messages transmission is decrease. The residual energy discrepancy between nodes become large in the traditional routing protocols while our algorithms balances the energy consume of all the nodes in the network, so the variance of residual energy between nodes declines with the messages generation interval increase.

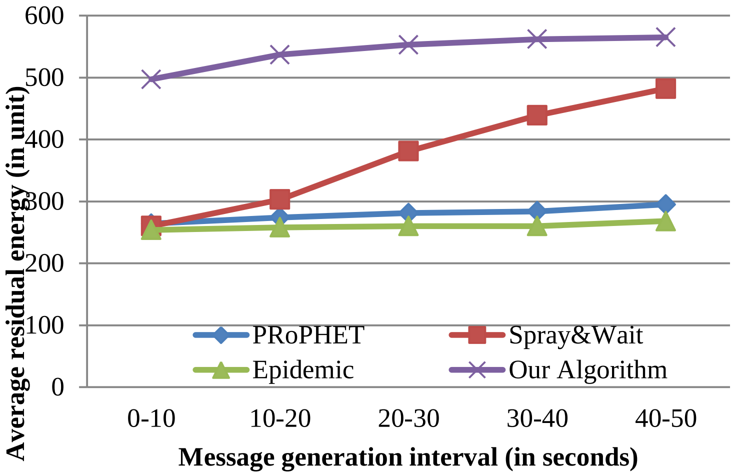

Figure 6. Average Residual Energy vs. Message Generation Interval

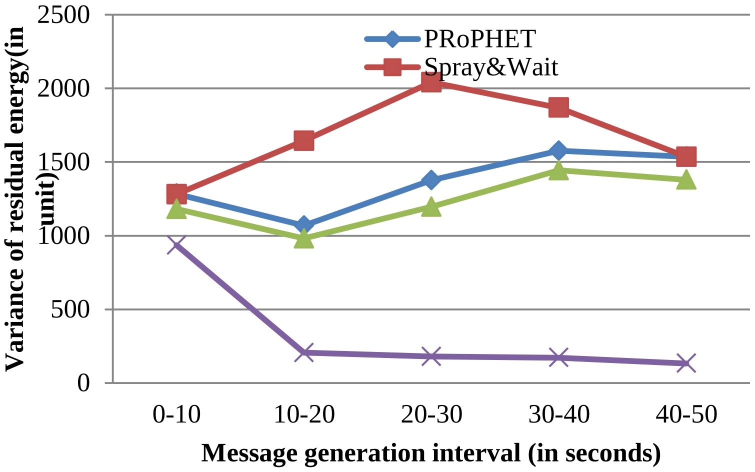

Figure 7. Variance of Residual Energy vs. Message Generate on Interval

5.3.4. Performance of average end to end delay: Figure 8 shows the performance of average end to end delay with the message size is set between 0.5MB and 1MB, the message generation interval is fixed between 25 sec and 35 sec. It is clear that average end to end delay for our algorithm is slightly lower or almost equal to PRoPHET and Epidemic in the various numbers of nodes. They are all flooding based routing strategies.

Overall, in all simulation environments mentioned above, it can be concluded regarding the number of nodes, message size and message generation intervals, our proposed algorithm can get higher average residual energy and lower variance of residual energy than the other three routing protocols with a tolerable average end to end delay.

6. Conclusions

In this paper, we introduce a new model of *DICMLA*, and propose a routing algorithm based on the *DICMLA* model. The routing algorithm can balance the energy consumption and delivery ratio of nodes in the opportunistic network. In an opportunistic network, each node is equipped with multiple learning automata, and each automata agent learns the best next hop node for the specific destination node of its neighbors. We conduct sufficient simulation to compare our algorithm with other three routing protocols in the energy consumption and the variance of residual energy. The results show that our algorithm is better energy efficient than the others. Thus the *DICMLA* and the corresponding algorithm can be used to prolong the life time of nodes in the opportunistic networks on the premise of reasonable end to end delay of message transmit.

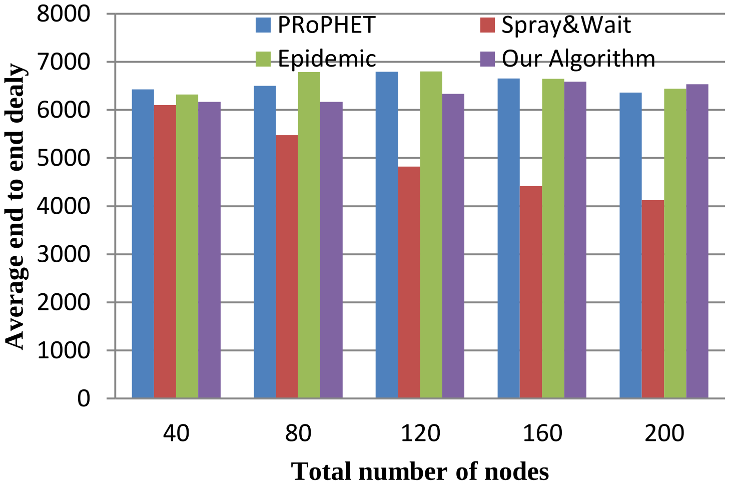

Figure 8. Average End to End Delay vs. Total Number of Nodes

Acknowledgments

This work is supported by National Natural Science Foundation of China under Grants no. 61373083, 61402273, and the Natural Science Basis Research Plan in Shaanxi Province of China under Grant Nos.2014JQ8353 and 2015JQ6238.

References

[1] C.-M. Huang, K. chan Lan, and C.-Z. Tsai. "A survey of opportunistic networks". In Proc. of the 22nd International Conference on Advanced Information Networking and Applications - Workshops, IEEE, Okinawa, Japan, **(2008)**, pp. 1672–1677.

[2] S. He, J. Chen, F. Jiang, D.K.Y. Yau, G. Xing and Y. Sun. "Energy provisioning in wireless rechargeable sensor networks", Mobile Computing, IEEE Transactions on, vol.12, no.10, **(2013)**, pp.1931-1942.

[3] C. Chilipirea, A. Petre and C. Dobre. "Energy-Aware Social-based Routing in Opportunistic Networks", Advanced Information Networking and Applications Workshops (WAINA), 2013 27th International Conference on. IEEE, **(2013),** pp.791-796.

[4] R. Sheikhpour, and S. Jabbehdarin, "An Energy Efficient Chain-based Routing Protocol for Wireless Sensor Networks", KSII Transactions on Internet and Information Systems, vol.7, no.6, **(2013)**, pp.1357-1378.

[5] E.P.C Jones and P.A.S Ward, "Routing strategies for delay-tolerant networks", ACM Computer Communication Review , **(2008)**.

[6] S. Jian, S. Moh, and I. Chung, "Routing protocols in delay tolerant networks: A comparative survey", The 23rd International Technical Conference on Circuits/Systems, Computers and Communications, **(2008)**, pp.1577-1580.

[7] R. S Mangrulkar and M. Atique, "Routing protocol for delay tolerant network: A survey and comparison", Communication Control and Computing Technologies, IEEE International Conference on, **(2010)**, pp. 210-215.

[8] Y. Wu, S. Deng and H. Huang, "Energy-efficient Joint Control of Epidemic Routing in Delay Tolerant Networks", KSII Transactions on Internet and Information Systems, vol.7, no.2, **(2013)**, pp.234-252.

[9] T. Spyropoulos, K. Psounis and C. S. Raghavendra, "Single-copy routing in intermittently connected mobile networks", In Proc. of the 1st Annual IEEE Communications Society Conference on Sensor and Ad Hoc Communications and Networks, California, USA, **(2004)**, pp. 235-244.

[10] T. Spyropoulos, K. Psounis and C. S. Raghavendra, "Spray and wait: an efficient routing scheme for intermittently connected mobile networks", In Proc. of the 2005 ACM SIGCOMM Workshop on Delay-Tolerant Networking, Pennsylvania, USA, **(2005)**, pp. 252-259.

[11] A. Lindgren, A. Doria and O. Schelén, "Probabilistic routing in intermittently connected networks", ACM SIGMOBILE Mobile Computing and Communications Review, vol.7,no.3, **(2003)**, pp. 19-20.

[12] S. K. Dhurandher, D. K. Sharma and I. Woungang, "Energy-based Performance Evaluation of Various Routing Protocols in Infrastructure-less Opportunistic Networks", Journal of Internet Services and Information Security (JISIS), vol.3, no.1-2, **(2013)**, pp. 37-48.

[13] H. Jun, M. H. Ammar and E. W. Zegura, "Power management in delay tolerant networks: a framework and knowledge-based mechanisms", Sensor and Ad Hoc Communications and Networks, IEEE SECON 2005, vol.5, **(2005)**, pp. 418-429.

[14] H. Beigy, M. R. Meybodi, "A Mathematical Framework for Cellular Learning Automata", Advances in Complex Systems, vol.7, no.3&4, **(2004)**, pp. 295-319.

[15] M. R. Meybodi, H. Beigy and M. Taherkhani, "Cellular learning automata and its applications", Sharif Journal of Science Technology, vol.19, no.25, **(2003)**, pp. 54-77.

[16] A. H. F. Navid, and H. H. S. Javadi, "An energy aware cellular learning automata based routing algorithm for sensor networks", Communications and Information Technology, 9th International Symposium on, **(2009)**, pp. 81-86.

[17] M. Esnaashari and M. R. Meybodi, "Deployment of a mobile wireless sensor network with k-coverage constraint: a cellular learning automata approach", Wireless networks, vol.19, no.5, **(2013)**, pp. 945-968.

[18] M. Asnaashari, and M. R. Meybodi, "Irregular Cellular Learning Automata and Its Application to Clustering in Sensor Networks", Proceedings of 15th Conference on Electrical Engineering, Tehran, Iran, **(2007)**, pp. 14-28.

[19] M. Kalantary and M. R. Meybodi, "Mobile sensor network deployment using cellular learning automata approach", Ultra Modern Telecommunications and Control Systems and Workshops, 2010 International Congress on IEEE, **(2010)**, pp. 976-980.

[20] M. Esnaashar and M. R. Meybodi, "A cellular learning automata-based deployment strategy for mobile wireless sensor networks", Journal of Parallel and Distributed Computing, vol.71, no.7, **(2011)**, pp. 988-1001.

[21] H. Beigy and M. R. Meybodi, "Cellular learning automata with multiple learning automata in each cell and its applications", Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on, vol.40, no.1, **(2010)**, pp. 54-65.

[22] A. Keranen, "Opportunistic network environment simulator", Special Assignment report, Helsinki University of Technology, Department of Communications and Networking, **(2008)**.

[23] A. Keränen, J. Ott and T. Kärkkäinen, "The ONE simulator for DTN protocol evaluation", Proceedings of the 2nd international conference on simulation tools and techniques, **(2009)**.

[24] The ONE. http://www.netlab.tkk.fi/tutkimus/dtn/theone/.

Authors

Feng Zhang, He received the M.S. degree in computer science from Xidian University, Xi'an, China, in 2007. He is currently a Ph.D. candidate in Shaanxi Normal University, Xi'an, China. His main research interests include wireless sensor networks and opportunistic networks.

Xiaoming Wang, He received the Ph.D. degree in computer science from Northwest University, Xi'an, China, in 2005. He is currently a professor and Ph.D. supervisor in Shaanxi Normal University, Xi'an, China. His main research interests include wireless sensor networks, opportunistic networks.

Peng Li, He received the Ph.D. degree in Computer Applications from Beijing Normal University, Beijing, China, in 2010. He is currently a lecturer in Shaanxi Normal University, Xi'an, China. His main research interests include wireless sensor networks and opportunistic networks.

Lichen Zhang, He received the M.S. degree and Ph.D. degree in computer science from Shanxi Normal University, Xi'an, China, in 2005 and 2012, respectively. He works as a visiting scholar in Georgia State University, USA, from October 2008 to October 2009. His main research interests include wireless sensor networks and opportunistic networks.