A Game Theory Approach for Optimal Routing: In Wireless Sensor Networks

Article · September 2010

DOI: 10.1109/WICOM.2010.5601119

2 authors:

Babak Arisian
Sharif University of Technology

Kourosh Eshghi
Sharif University of Technology

A Game Theory Approach for Optimal Routing

in Wireless Sensor Networks

Babak Arisian
Department of Industrial Engineering
Sharif University of Technology
Tehran, Iran
b_arisian@alum.sharif.edu

Kourosh Eshghi
Department of Industrial Engineering
Sharif University of Technology
Tehran, Iran
eshghi@sharif.edu

In this paper, a "Game Theory" approach for finding an optimal path in a "Wireless Sensor Network" is discussed. WSN is one of the most interesting research fields in the communication networks, and is the center of attention in recent years.

In our model, a pricing and payment technique is presented to obtain an optimal path in a WSN by considering reliability, energy and traffic load. The proposed algorithm is able to find a path which improves network lifetime, load distribution and path reliability.

Key Words: Wireless Sensor Networks-Game Theory-Routing

I. INTRODUCTION

Wireless Sensor Networks is a new technology which is used in a vast majority of applications. This network is a graph which consists of a large number of sense nodes. These nodes are able to gather data, process it and send it to the relevant destinations. The sensors have some special characteristics such as small dimension and low power consumption. Because of these special characteristics that these sensors have, they could be used in different fields such as military, agricultural, Industrial, and biomedical applications. Furthermore, they could easily be used in different environments such as inaccessible or dangerous regions. Since there is no need to use a large amount of wire and complicated configuration and installation for these sensors in the network, we could use them with lower cost in comparison with old wired networks.

According to the variety applications of this new technology and the need for special capability in different areas, there are a lot of different areas in the field of WSN that needs research and practical work in order to improve them. Routing is one of these fields in WSN. Routing means finding the best possible way for data transmission from source node to the destination node in the network by considering networks parameters (e.g. stability, consumed and remained power, data transmission speed, and etc). Network Lifetime is one of the important factors. As a matter of the fact that sensors are commonly small in size and mostly used in inaccessible regions, it is not possible to put a greater power supply or replace their power supply (battery). Therefore, these networks must be designed and used in a way to optimize the power consumption and life time of the network.

The other important factor that must be considered in the network is Load Traffic Distribution. Usually the traffic load in wireless sensor networks is unbalance. For example, sensors which are nearer to the source have more data load. Therefore, optimization of load distribution, called Load Balancing, is one of the important factors for improving the efficiency of the networks. Optimization of load traffic distribution in WSN could increase the lifetime of the network. Since, there is more power consumption in nodes with more traffic load then the data transaction in the network could be optimized.

Nowadays use of Game Theory in the vast majority of science and applications increase considerably. Usage of this theory for routing in different kinds of communication networks is one of them. In this paper, by using Game Theory approach for WSN, optimal route in WSN is found. In this approach, routing and sensor nodes are assumed to be the game and players respectively. All players want to increase their benefit. So we use a pricing and payment model as well as profit and loss calculation for each player. In this model, the destination node pays a credit to the source node for each data packet successfully reception. Moreover, the source node pays a portion of this credit to each intermediate node that participates in data packet transaction. Furthermore, each node sustains a cost for each data packet transaction to other node. This cost is called Transmission Cost and related to different parameters. Also each node transmits the received data packet to the next hope with the probability, calculated by the reliability of the node. This parameter is depending on several items, e.g. failure probability, sleep cycling, etc. In this routing game, one of the indicators for optimal route is path reliability. Therefore, profit of each node is equal to the difference between incoming of each node. The cooperation for transmission data packet and the cost of this operation will affect the incoming of each node. Thus, each node is incentive to the cooperative game and its location on the path, if its profit is positive. If the profit of one node is negative, it is not considered in the game, because the profit of the node will be remained at least zero, instead of negative.

For modeling the cost of transmission, three parameters at each node are considered: distance, the remained energy and the load traffic. The model can optimize the power consumption (proportional to distance), network lifetime and network load distribution. For example, if the distance between nodes or traffic load of one node is increased or the energy in a node is decreased, then the transmission cost will be increase. This will be caused to selection an alternate path, with less cost since the profit of all nodes will be positive. One of the most important things in routing problem is the stability of a path which can be modeled by using the concept of Nash equilibrium.

In this paper, after a quick review of the related works, we explain our model and introduce an algorithm for finding path. Then, our model is tested and a sensitivity analysis is discussed.

II. Literature Review

The routing problem in WSN is one of most interesting research fields in the communication network and there are lots of papers published recently in this field. Among of these works, we can mention the methods that introduce in [1-5], e.g. Flooding Routing, Energy-Aware Routing, Sensor Protocol for Information Negotiation (SPIN), Directed Diffusion Routing, Rumor Routing, Low Energy Adaptive Clustering Hierarchy (LEACH), Geographic Adaptive Fidelity (GAF), etc. There are different methods to optimize the various parameters in the networks, e.g. the methods for optimizing the traffic load distribution (load balancing) discussed in [6].

One of the methods used for routing in WSN is Game Theory. The game theory approach in routing is usually based on the reputation and punishment model, or pricing and payment model. The functionality of pricing and payment model [7-12] is described that for each successfully delivered data packet from a source to a destination, the destination node pays a credit to source node, and all intermediate nodes that participate in routing game. However, each data packet transmission has a cost for each node that participates in the route (which is depending on the various parameters). These nodes, want to maximize their profit. Therefore, each node is located on the path if its profit is not negative.

In a quick view of references, [7, 8] are reviewed the game theory in networks, and introduced game-theoretic models for WSN. [8-11] are introduced a pricing and payment model for WSN with the path reliability. Consequently, [12] is introduce a model that is added source and destination to it, and available for all players to pick a null strategy. With this model, we can simply find a path with Nash equilibrium. Also, in this paper introduce an algorithm for finding optimal path with usage of Dijkstra algorithm [13], and additional stage for checking the utility function of nodes. This algorithm can found a path with positive utility function for all nodes that participate in it, and maximizing the reliability of path.

In this paper, whit usage of [7-12], a model that optimizes by the network lifetime and network load distribution is introduced simply by adding the remained energy and traffic load in nodes. Also, a modified Dijkstra algorithm is developed for finding optimal path.

III. The Mathematics Model

These networks are modeled as a graph G such that $G=(V,E)$, where $v_i \in V$ is shown each node, and $l_{ij}=(v_i,v_j)\in E$ is shown each link. Each node v_i is associated with a probability value p_i, which is the probability of forwarding each packet that is sent to it, and each link l_{ij} is associated with a weight of link C_{ij}, which is the cost of the communication between v_i and v_j nodes, and depends on the various parameters used in the model.

In these networks, we have three different types of nodes: source node (v_s), destination node (v_q), and intermediate nodes $(v_i \in V \mid \{v_s, v_q\})$ that can be cooperate with themselves to carry the data packets from a source to a destination.

Each node can transmit/receive data packets to/from another node if the distance between them is less than communication range, denoted by *R*.

There are various parameters for these networks. One of them is the remained energy of each sensor. Each node v_i has an energy value E_i that will reduce by cycle operation. If the initial value of this parameter is $E_{\max}$, then we always have $0 \le E_i \le E_{\max}$. If this value is less than a threshold, which is shown by $E_{\min}$, then this node can not transmit any data packet to other nodes. Another parameter is the load traffic of each node, denoted by T_i. This parameter is the number of data packets that node v_i receives and sends it to another nodes. The total number of data packets that are transact in a network is defined by *T* and the density of load for each node is defined as $\tau_i = T_i / T$.

In these networks, each sensor is defined as a node in a graph, and two nodes are linked together if they are located in transmission range of each other. Therefore, if D_{ij} is the distance between v_i and v_j, then:

$$\forall v_i, v_j \in V, D_{ij} \le R_{ij} \Rightarrow (v_i, v_j) \in E$$

Also, the neighbor set of a number of nodes, is defined as if $K \subset V$, $N(K) \subset V$ then *N(K)* is the set of neighbors of *K*, if and only if:

$$\forall v_i \in N(K), v_i \notin K, \exists v_j \in K \mid (v_i, v_j) \in E.$$

A. Assumptions

The general assumptions of our model are:

- The reliability of source or destination is set to one.
- All of the intermediate nodes have the same specifications and limited energy and their initial energy is equal to $E_{\max}$.
- The threshold value of energy for all intermediate nodes is the same and equal to $E_{\min}$.

- The energy of source or destination is infinite.
- The transmission range of all nodes is equal to *R*.
- The load traffic of source or destination is infinite.

B. A Game Theory Approach for The Model

The structure of network is described as a pricing and payment model: the destination node offers a payment amount M to the source node if it receives a data packet successfully. The source node offers a payment amount Q_i to each intermediate node v_i if it is cooperating in data transmission. For simplicity, we assume that this value is the same for all intermediate nodes and equal to *Q*. Therefore, each node is cooperating in the game and is locating in transmission path if it has a positive value of profit. It means that the benefit that this node is earned for data packet transmission is more than the cost of transmission of this packet.

As a game-theoretic formulation, we define a game F as a triple $F = <I, A, U>$, where I is the set of players. $A = \{a_i \mid i \in I\}$ is the set of actions that available for each player and $U = \{u_i \mid i \in I\}$ is the set of utility functions.

In a network with *n* nodes (include source and destination), the game strategy for each node (unlike destination, which we can ignore its strategy in this game) is a *n-1* tuple $A_i = (a_{i1}, a_{i2}, ..., a_{ii-1}, a_{ii+1}, ..., a_{ij}, ..., a_{in})$, where $a_{ij} \in \{0,1\}$, and $v_i \in V \mid \{v_q\}$, $v_j \in V \mid \{v_i\}$. If v_j is the next hope of v_i, then $a_{ij} = 1$, otherwise $a_{ij} = 0$. In each strategy tuple, at most one of the a_{ij} can be one and the others must be zero. If all of the a_{ij} are zero, this means that the v_i node will not cooperate in the game and it is not located on the path.

The utility function for each node is defined as:

$$u_i = \psi_i (B_i - C_{ij}) \tag{1}$$

where $\psi_i \in \{0,1\}$. If the v_i node is located on the path, then $\psi_i = 1$, otherwise $\psi_i = 0$. Each path between source, v_s, and destination, v_q, in the network can be defined as $O = (v_s, v_1, v_2, ..., v_h, v_q)$, where *h* is the number of intermediate nodes on the path.

Also, the benefit of each node on the path is defined as:

$$B_i = b_i . \prod_{v_i}^{v_h} p_i \tag{2}$$

where b_i for source node is equal to:

$$b_s = M - h.Q \tag{3}$$

and for the other intermediate nodes on the path is equal to:

$$b_i = Q \tag{4}$$

Then, we define a model for cost of transmission for each link, include distance, energy, and load traffic. Therefore, we assume that the cost function, C_{ij}, consists of three independent parts. Each part is depending on one of the parameters.

For the part that depend on the distance, $C_{D_{ij}}$, according to the prevalent models, we have:

$$C_{D_{ij}} = \alpha.(\frac{D_{ij}}{\delta_{ij}})^2$$

where $\delta_{ij} \in \{0,1\}$, if v_i and v_j nodes are located in transmission range of each other, or $d_{ij} \le R$, then $\delta_{ij} = 1$, otherwise, $\delta_{ij} = 0$, and this means that the cost of link is infinite and v_j will not be the next hope of v_i on the path.

The next part, which is depending on the remained energy in the nodes, $C_{E_{ij}}$, is defined as:

$$C_{E_{ij}} = \alpha'.(\frac{E_{\max}}{E_j.\lambda_j})$$

where $\lambda_i \in \{0,1\}$, if the remained energy in v_j node is not less than the threshold value, or $E_j \ge E_{\min}$ then $\lambda_j = 1$. Otherwise $\lambda_j = 0$. This means that the cost of link is infinity, therefore, v_i will not pick the v_j node as its next hope on the path. For the part that proportion with load traffic, $C_{T_{ij}}$, the relation is defined as:

$$C_{T_{ij}} = (1 + \gamma.\tau_j)$$

where γ is the coefficient value for growth of τ_j. Thus, the total cost function is defined as:

$$C_{ij} = \beta.C_{D_{ij}}.C_{E_{ij}}.C_{T_{ij}}$$

According to above relations, and with assumption of $\beta = \alpha.\alpha'$, we have:

$$C_{ij} = \beta.(\frac{D_{ij}^2}{\delta_{ij}}).(\frac{E_{\max}}{E_j\lambda_j}).(1 + \gamma\tau_j) \tag{5}$$

where β is the coefficient value of transmission cost.

C. Model Structure

In this section, we explain the structure of model to optimize the network life-time and network load distribution. For this purpose, we assume that we have a network like shown in Figure 1, that consists of one source node v_{s_1} and one destination node v_q. Data packets should be transmit from source to destination. We assume that the path from the source to the destination is found as $O = (v_{s_1}, v_1, v_2, v_3, v_4, v_5, v_6, v_q)$. If the energy of node v_4 is decreased after some data transmission, because this node requires more energy than the others, due to cost function, the cost of transmission between v_3 and v_4 will increase. If the increasing is so much that the utility function of v_3 for selecting the v_7 as its next hope is more than this value for selecting v_4 for next hope of v_3, then v_7 is selected by v_3 as its next hope. Therefore, the path $O' = (v_{s_1}, v_1, v_2, v_3, v_7, v_8, v_9, v_q)$ is replaced with *O*. This change causes decreasing the rate of reducing energy in v_4. This manner for all of the nodes in the network will increase the total lifetime of the network.

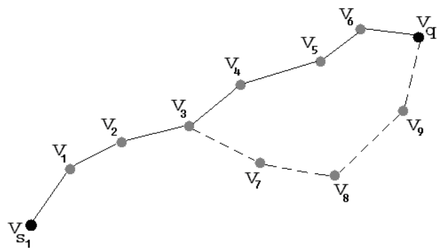

Figure 1. III Changes in the path because of energy reduction in v_4 node

Furthermore, for showing the optimization of network load distribution, we assume again that there is a network with the path between v_{s_1} and v_q is founded as $O = (v_{s_1}, v_1, v_2, v_3, v_4, v_5, v_6, v_q)$. Now assume that another source node is added to this network, like v_{s_2} and it wants to convey some data packets to destination node v_q, like Figure 2. So, if we want to find a path between v_{s_2} and v_q, without attention to the last founded path, this path will be $O_2 = (v_{s_2}, v_{10}, v_{11}, v_{12}, v_4, v_5, v_6, v_q)$. There are a few nodes in common on the two founded paths. Therefore, with attention to data packet transmission between v_{s_1} and v_q, the load density for the nodes on this path will increase, e.g. for v_4 node, that is cooperate in both paths, this density will be increased and this will cause to increase the transmission cost between v_{12} and v_4.

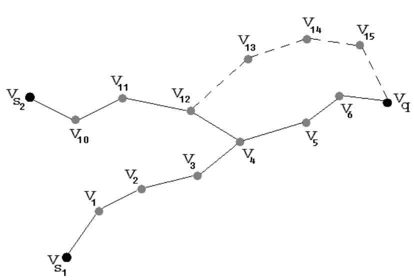

Figure 2. Changes in the path because of increase load traffic in v_4 node

Now, if this increment causes that the transmission cost between v_{12} and v_4 to be less than this value for v_{12} and v_{13}, then v_{12} will select v_{13} as its next hope on the path. By continuing this manner by all of the nodes in the network, especially when the number of source nodes is huge, the load traffic in the network will be balanced more than before.

IV. THE ALGORITHM OF PATH FINDING

To obtain an algorithm for finding a path between the source and the destination via intermediate nodes in our model, we use the utility function. Therefore, with attention to this condition that utility function must be positive for all of the nodes on the path, and according to (1), we have:

$$u_i \geq 0 \Rightarrow B_i - C_{ij} \geq 0$$

in according to (2) and (5), we have:

$$[b_i.\prod_{k=i}^{h} p_k] - [\beta.(\frac{D_{ij}^2}{\delta_{ij}}).(\frac{E_{\max}}{E_j.\lambda_j}).(1+\gamma\tau_j)] \geq 0$$

therefore:

$$\prod_{k=i}^{h} p_k - [(\frac{\beta}{b_i}).(\frac{D_{ij}^2}{\delta_{ij}}).(\frac{E_{\max}}{E_j.\lambda_j}).(1+\gamma\tau_j)] \geq 0 \tag{6}$$

We define C'_{ij} as:

$$C'_{ij} = (\frac{\beta}{b_i}).(\frac{D_{ij}^2}{\delta_{ij}}).(\frac{E_{\max}}{E_j.\lambda_j}).(1+\gamma\tau_j) \tag{7}$$

where this equation for source node, according to (3) is equal to:

$$C'_{s1} = (\frac{\beta}{M - h.Q}).(\frac{D_{s1}^2}{\delta_{s1}}).(\frac{E_{\max}}{E_1.\lambda_1}).(1+\gamma\tau_1)$$

and for other intermediate nodes, according to (4) is equal to:

$$C'_{ij} = (\frac{\beta}{Q}).(\frac{D_{ij}^2}{\delta_{ij}}).(\frac{E_{\max}}{E_j.\lambda_j}).(1+\gamma\tau_j)$$

Therefore, with attention to (6) and (7), we have:

$$\prod_{k=i}^{h} p_k - C'_{ij} \geq 0 \tag{8}$$

This formulation guarantees that all nodes on the path have positive utility function. Therefore, to obtain an algorithm for finding a path with minimum transmission cost, that utility function of all nodes on the path must be positive and we use Dijkstra's algorithm for minimizing C'_{ij}. Moreover, we add a stage to each phase of Dijkstra algorithm for checking the utility function to be positive, according to equation 13-4. By this additional stage, if a link caused that the utility function of one node is negative, it will be deleted. The pseudo-code of this algorithm is shown in Figure 3.

This algorithm is Polynomial Time, because it is based on Dijkstra's algorithm. We now that the Dijkstra algorithm is run in $O(n^2)$. For each additional stage for checking utility function or link deletion, in worst case, we need an extra time $O(n)$. Therefore, this algorithm, in worst case, will be run in $O(n^3)$.

```
inputs : G = (V, E), v_s, v_q
outputs : optimal path between v_s and v_q

begin
  // Initializing
  W = Null;  // W is set of labeled nodes
  L(v_q) = 0;
  M(v_q) = 1;
  - for all v_i ∈ V \ {v_q}
     L(v_i) = Infinity;
     Previous(v_i) = Undefined;
     M(v_i) = 1;
  - end for;

  // Main Loop
  - while( (v_s ∉ W) & (Neighbor(W) ≠ Null) )
     Select v_j ∈ V , such that ∀v_k ∈ V \ {v_j}, L(v_j) ≤ L(v_k);
     remove v_j from V;
     add v_j to W;
     - for all v_i ∈ V such that (v_i, v_j) ∈ E
        x = L(v_j) + C'_ij;
        - if (L(v_i) > x)
           L(v_i) = x;  // label of v_i node
           M(v_i) = p_i . M(v_j) ;  // equal to ∏ p_k
           - if ( (M(v_i) - C'_ij) < 0)   // Check the utility function
              delete edge (v_i, v_j) from E
           - else
              Previous(v_i) = v_j ;
           - end if;
        - end if;
     - end for;
  - end while;
end.
```

Figure 3. Psuedo-code for finding path algorithm

This algorithm is starting from the destination node, and it continues until the source node will be labeled. This means the path between the source and the destination is found. This algorithm is terminated if the neighbor set of the labeled nodes will be null. This means that source node and destination node are located in two separated sections of graph, and as a result there is no path between the source and the destination.

We assume that all of the nodes that do not located on the path will picked a null strategy. Therefore, the other nodes will not incentive to pick them as their next hope on the path.

Theorem: the final path founded by this algorithm is a Nash equilibrium.

Proof: we assume that the algorithm is found the path $O = (v_1, v_2, ..., v_i, v_j, ..., v_k, ..., v_n)$ from the source to the destination. By using the proof of contradiction, we assume that this path has not Nash equilibrium, and one of the nodes in the path, e.g. v_i node, wants to change its next hope. According to the last assumptions, all of the nodes that are not located on the path pick a null strategy, and other nodes will not incentive to pick them as their next hope on the path. So, the v_i node will not incentive to pick them as its next hope. Therefore, this node can only switch its next hope with a neighbor node that located on the path. This change will not be occurred, because for example if v_i wants to change its next hope from v_j to v_k, the new path $O' = (v_1, v_2, ..., v_i, v_k, ..., v_n)$ is also a path with positive utility function for all the nodes on the path. In this path the number of nodes is reduced, since $(0 \le p_i \le 1, \forall i)$, the path reliability for the nodes before v_i will increase and for the nodes after v_i (include v_i), there is no change. As there is no change in link cost value between all nodes on the path except between v_i and v_k link, the utility function of all nodes except v_i is remained positive. For v_i, if utility function of v_i for selecting v_k as its next hope is negative, then v_i is not selected v_j for its next hop. So, the utility function of v_i is positive. Therefore, O' is a path with positive utility function for each node, and is an available path. Due to the deletion of some nodes and links in O', the sum of C'_{ij} in O' is less than this value in *O*, therefore, algorithm must be obtained O' instead of *O*. Hence, the path that is found by algorithm has a Nash equilibrium. □

V. EXPERIMENTAL RESULTS

In this section, for showing behavior of our model and algorithm, and its effect on network lifetime, network load distribution, and path reliability, we run two tests, and then compare and analyze the results of them.

A. *Terms of Conditions for Simulations*

These simulations are running for 100 pieces of sensors, in 50m×50m area. In this simulation, the transmission range of nodes is 10m, and reliability of each sensor is randomly is picked up from interval [0.5, 1]. Initial value of energy for all nodes is set to 200 and minimum required energy for each sensor is 20. All data packets are the same, and power consumption for receiving data packet is negligible. For transmitting each data packet, power consumption is equal to:

$$E_{Tr_i} = \eta . D_{ij}$$

where D_{ij} is the distance between v_i and v_j, and η is the coefficient value of power consumption per each data packet. In this simulation, we set this value to 0.3 per each data packet. By the way, the remained energy in each node after transmitting one data packet is equal to:

$$E'_i = E_i - E_{Tr_i} = E_i - \eta . D_{ij}$$

Therefore, with these terms of conditions, we run two simulations. In first simulation, we use the Dijkstra algorithm to find the path with only minimum distance between source and destination, without checking utility functions of nodes. For simplicity, we name this simulation “No. 1”, and showing the results of this simulation in curves with “SR1”. In the second simulation, we use our model and algorithm, and for simplicity we name this simulation “No. 2”, as well as showing the results of this simulation in curves with “SR2”. The result of these simulations and their analyses are described as the following items.

B. Network Lifetime

The network lifetime for each simulation is showed in Figure 4. These curves are showing that in simulation No. 1, after 400 rounds, about 80% of nodes in the network are died, but in simulation No. 2, after 550 rounds the network is arrived to this point. So the network lifetime is increasing about 37% with using of our model and algorithm.

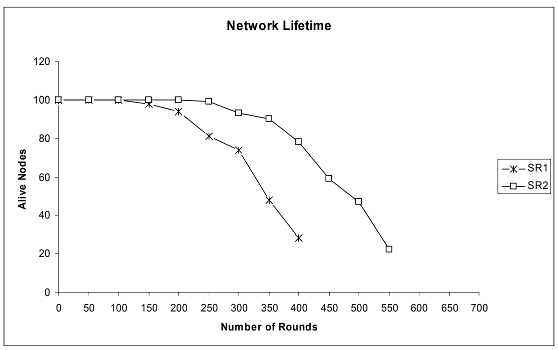

Figure 4. Network lifetime

Also, the time that first node died, for simulation No. 1 is between rounds 100 to 150, and for simulation No. 2 is between rounds 200 to 250. It means that this parameter is improved for our model and algorithm.

C. Network Load Distribution

The network load distribution for each simulation is showed in Figure 5. We know that in wireless sensor networks, usually the nodes that are located near the destination have more density of load. We can see this problem in the curves of two simulations.

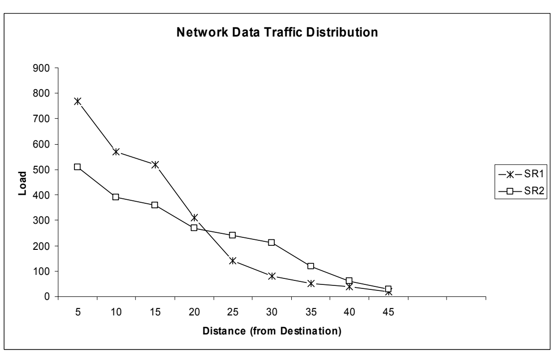

Figure 5. Network load distribution

However, in the simulation No. 1, the load value for the nodes that are located near the destination is about 800, when the distance to the destination is increased, the load value quickly decreases. In the simulation No. 2, the load value for the nodes that located near the destination is about 500, and with increase of distance to the destination, the load value decreases but the slop of decreasing is less than simulation No. 2. Therefore, in our model and algorithm, the network load distribution is improved.

D. Path Reliability

In Figure 6, the path reliability for each simulation (in relation with number of nodes) is shown.

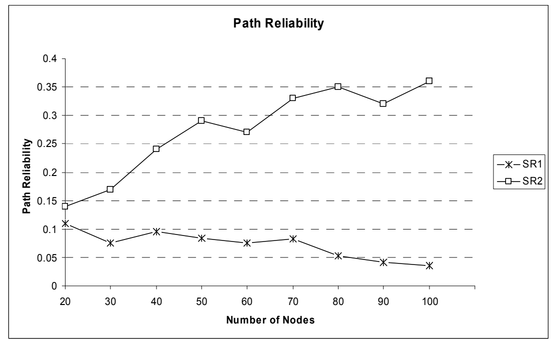

Figure 6. Path reliability

In these curves, we can see that the path reliability for simulation No. 1 is random, but the path reliability for simulation No. 2, increases with increment of node numbers. When the number of nodes increases to above 70 nodes, the path reliability is more than 0.3, so the path reliability for our model and algorithm is improved.

VI. CONCLUSION

The results of simulation and their analyze shows that our model and algorithm increases the network lifetime, network load distribution, and path reliability. In the future works, we have been comparing this model and algorithm with the other most important ones. Also, we will be optimizing the algorithm to find the path with maximum utility function of all nodes that cooperate in path. Furthermore, we will be modeling the transmission cost with other methods, to include data link layer of network cost.

REFERENCES

[1] C. F. Garcia-Hernandez, P. H. Ibarguengoytia-Gonzalez, J. Garcia-Hernandez, and J. A. Perez-Diaz, “Wireless Sensor Networks and Applications: A Survey”, IJCSNS International Jornal of Computer Science and Network Security, Vol.7, No.3, March 2007.

[2] D. Culler, D. Estrin, M. Srivastava, “Overview of Sensor Networks”, IEEE Computer Society, August 2004

[3] I. Akyildiz, W. Su, and Y. Sankarasubramaniam, “A Survey on Sensor Networks”,IEEE Communication Magazine, August 2002, pp. 102-114.

[4] J. N. Al-Karaki, A. E. Kamal, “Routing Techniques in Wireless Sensor Networks: A Survey”, IEEE Wireless Communications, 2004

[5] W. R. Heinzelman, A. Chandrakasan, H. Balakrishnan, “Energy Efficient Communication Protocol for Wireless Microsensor Networks”,

Proceedings of the Hawaii International Conference on System Sciences, Maui, Hawaii, January 2000.

[6] L. Popa, A. Rostamizadeh, R. Karp, C. Papadimitriou, and I. Stoica, “Balancing Traffic Load in Wireless Networks with Curveball Routing”, in ACM Mobihoc, July 2007.

[7] L. DaSilva, V. Srivastava, “Node Participation in Ad Hoc and Peer-to-Peer Networks: A Game-Theoritic Formulation”, Workshop on Games and Emergent Behavior in Distributed Computing, 2004.

[8] R. Kannan, S. Sarangi, S. S. Iyenger, “Game-Theoretic Models for Reliable Path-Length and Energy-Constrained Routing with Data Aggregation in Wireless Sensor Networks”, IEEE Journal on Selected Areas in Communication, Vol. 22, August 2004.

[9] R. Kannan, S. Sarangi, S. S. Iyenger, ”A Simple Model for Reliable Query Reporting in Sensor Networks Information Fusion”, Proceedings of the Fifth Information Conference, July 2002.

[10] R. Kannan, S. Sarangi, S. S. Iyenger, “Sensor Centric Quality of Routing in Sensor Networks”, IEEE Infocom, 2003.

[11] R. Kannan, S. Sarangi, S. S. Iyenger, “Sensor-Centric Energy-Constrained Reliable Query Routing for Wireless Sensor Networks”, Journal of Parallel Distributed Computing, Vol. 64, pp. 839-852, 2004.

[12] H. Liu , B. Krishnamachari, “A Price-Based Reliable Routing Game in Wireless Networks”, IEEE/ACM International Conference, Vol. 199, October 2006

[13] E. W. Dijkstra, “A Note on two Problems in Connexion with Graphs”, Numerische Mathematik 1, pp. 269-271, 1959.