IJDACR
International Journal Of Digital Application & Contemporary Research

International Journal of Digital Application & Contemporary research
Website: www.ijdacr.com (Volume 2, Issue 3, October 2013)

Reinforcement Learning based Efficient Routing for Wireless Mesh Network

Deepshikha Yadav
Dept of Computer Science & Engineering
Shri Satya Sai Institute of Science & Technology
Sehore, India
deepshikha.cs16@gmail.com

Gajendra Singh
Dept of Computer Science & Engineering
Shri Satya Sai Institute of Science & Technology
Sehore, India
gajendrasingh86@rediffmail.com

***Abstract*— Wireless mesh networks are a promising technology for offering ubiquitous Internet connectivity. These multi-hop wireless access networks consist of fixed and mobile nodes, which help each other relaying packets toward Internet gateways and back. This Synopsis addresses the problem of multipath routing in wireless mesh networks. We study the use of clustering algorithms to facilitate the discovery and deployment of non-interfering multipath routes in these settings. In this context we propose a novel clustering based intelligent reinforcement learning algorithm based on Markov decision process to discover and maintain routes in an efficient manner and when a mesh node should have to sleep or awake depends upon involvement of mesh node in routing, aiming at minimizing interferences between transmissions of neighbouring nodes. The work offers an interesting trade-off between the signalling costs, the time required to set up and maintain paths, and the properties of the discovered paths.**

***Keywords* —Wireless Mesh Network, Reinforcement Learning, Markov Decision Process.**

I. INTRODUCTION

A. *Wireless Mess Network*

A wireless mesh network (WMN) is a communications network made up of radio nodes organized in a mesh topology. Wireless mesh networks often consist of mesh clients, mesh routers and gateways. The mesh clients are often laptops, cell phones and other wireless devices while the mesh routers forward traffic to and from the gateways which may, but need not, connect to the Internet. The coverage area of the radio nodes working as a single network is sometimes called a mesh cloud. Access to this mesh cloud is dependent on the radio nodes working in harmony with each other to create a radio network. A mesh network is reliable and offers redundancy. When one node can no longer operate, the rest of the nodes can still communicate with each other, directly or through one or more intermediate nodes. The animation below illustrates how wireless mesh networks can self-form and self-heal. Wireless mesh networks can be implemented with various wireless technology including 802.11, 802.15, 802.16, cellular technologies or combinations of more than one type.

A wireless mesh network can be seen as a special type of wireless ad-hoc network. A wireless mesh network often has a more planned configuration, and may be deployed to provide dynamic and cost effective connectivity over a certain geographic area. An ad-hoc network, on the other hand, is formed ad hoc when wireless devices come within communication range of each other. The mesh routers may be mobile, and be moved according to specific demands arising in the network. Often the mesh routers are not limited in terms of resources compared to other nodes in the network and thus can be exploited to perform more resource intensive functions. In this way, the wireless mesh network differs from an ad-hoc network, since these nodes are often constrained by resources.

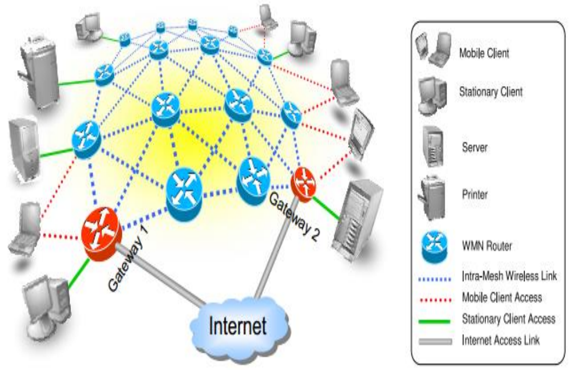

Figure 1: A wireless mesh network connecting several stationary and mobile clients to the Internet

B. *Objective of Wireless Sensor Network*

Energy awareness is critical, especially in situations where it is not possible to replace sensor node batteries so it is essential design issue in wireless sensor networks. Most sensor network

IJDACR
ISSN: 2319-4863

IJDACR
International Journal Of Digital Application & Contemporary Research

International Journal of Digital Application & Contemporary research
Website: www.ijdacr.com (Volume 2, Issue 3, October 2013)

applications aim at monitoring or detection of phenomena likes office building environment control, wildlife habitat monitoring, and forest fire detection.

The main problem is that a network of wireless sensors are low-powered, it is essential to capture as many environmental events as possible while still preserving the battery life of the sensor node.

The main aim or objective is to properly utilize the energy in sensors network so as to increase the lifetime of sensors network. These protocols aim to be energy-efficient in order to elongate the battery lifetime and network lifetime as a result. In most application scenarios the replacement of failed or depleted network nodes is not an option since they are placed in hazardous zones, thus it is extremely important that nodes consume the minimum amount of energy in order to make as long as possible the lifetime of the network, i.e. the time the application is still working properly.

For fulfilling our objective for the problem statement and to increase the lifetime of sensors network as compare with the base paper selected, we are developing an intelligent sensing algorithm based on Markov decision process (Infinite-horizon dynamic programming and Bellman's equation) which is a type of Reinforcement Learning scheme. The sleeping behavior and energy based cluster head selection (or energy based routing) will decide through our intelligent sensing algorithm.

C. Reinforcement Learning

Reinforcement learning, a sub-area of machine learning, uses a mathematical way to evaluate the success level of actions [12, 13]. Its emphasis on individual learning from the direct interactions with the environment makes it perfectly suited to distributed cognitive radio scenarios. Reinforcement learning has been considered as the most suitable learning approach for cognitive radio systems in this work. There are mainly two reasons:

- Reinforcement learning is an individual learning approach where the learning agent learns only on local observations. This is perfectly suited to cognitive radios who also work on a fully distributed fashion.
- Reinforcement learning learns on a trial-and-error basis that no environment model is required. This is also perfectly suited to cognitive radio systems which constantly interact with an 'unknown' radio environment on a trial-and-error basis.

The original reinforcement learning model [13] where agents are interacting with the environment as illustrated in figure 1.2 consists of:

- A set of possible states, represented by S
- A set of actions, A
- A set of numerical rewards R

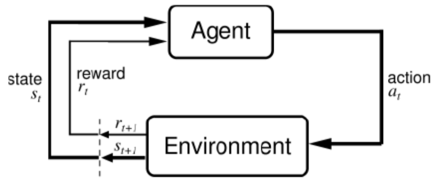

Figure 2: Standard Reinforcement Learning Model (directly reproduced from [12]).

The learner is called the agent. The outside world which it interacts with is called the environment. The learning problem can be formulated by defining:

- The state $s_t \epsilon S$. of the environment as observed by the agent, where S is the set of possible states.
- The action $a_t \epsilon A(s_t)$ chosen by the agent, where $A(t)$ is the set of actions admissible at time t.
- The probabilistic reward $r_{t+1} \epsilon R$, whose mean value is provided by the reward function $R(s_t, a_t)$. Roughly speaking, the reward function maps the state-action pair to a scalar value and it is used to measure the goodness of taking action at in state s_t.
- The state transition function $T(s_t, a_t, s_{t+1})$, which provides the probability of making a transition from state s_t to state s_{t+1} after performing action a_t. Note that $T(s_t, a_t, s_{t+1})$ captures the non-determinism of the environment, because taking the same action on the same state may result in different next states.

II. PROPOSED METHODOLOGY

A. Heterogeneous Consideration

In this section we describe our model of a wireless mesh network with nodes heterogeneous in their initial amount of energy. We particularly present the setting, the energy model, and how the optimal number of clusters can be computed. Let us assume the case where a percentage of the population of mesh nodes is equipped with more energy resources than the rest of the nodes. Let m be the fraction of the total number of nodes n, which is equipped with α times more energy than the others. We refer to these powerful nodes as advanced nodes, and the rest $(1 - m) \times n$ as normal nodes. We assume that all nodes are distributed uniformly over the mesh field.

i) Grouping through Clustering Hierarchy

We consider a mesh network that is hierarchically clustered.

The clustering protocol maintains such clustering hierarchy. In this protocol, the clusters are re-established in each "round." New cluster heads are elected in each round and as a result the

IJDACR
ISSN: 2319-4863

International Journal of Digital Application & Contemporary research
Website: www.ijdacr.com (Volume 2, Issue 3, October 2013)

load is well distributed and balanced among the nodes of the network. Moreover each node transmits to the closest cluster head so as to split the communication cost to the sink (which is tens of times greater than the processing and operation cost.) Only the cluster head has to report to the sink and may expend a large amount of energy, but this happens periodically for each node. In clustering protocol there is an optimal percentage p_{opt}(determined a priori) of nodes that has to become cluster heads in each round assuming uniform distribution of nodes in space.

If the nodes are homogeneous, which means that all the nodes in the field have the same initial energy, the clustering protocol guarantees that every one of them will become a cluster head exactly once every $1/p_{opt}$ rounds. Throughout this paper we refer to this number of rounds, $1/p_{opt}$, as epoch of the clustered mesh network.

Initially each node can become a cluster head with a probability p_{opt}. On average, $n \times p_{opt}$ nodes must become cluster heads per round per epoch. Nodes that are elected to be cluster heads in the current round can no longer become cluster heads in the same epoch. The non-elected nodes belong to the set G and in order to maintain a steady number of cluster heads per round, the probability of nodes ЄG to become a cluster head increases after each round in the same epoch. The decision is made at the beginning of each round by each node s Є G independently choosing a random number in [0, 1]. If the random number is less than a threshold T(s) then the node becomes a cluster head in the current round. The threshold is set as: Where, r is the current round number (starting from round 0.) The election probability of nodes Є G to become cluster heads increases in each round in the same epoch and becomes equal to 1 in the last round of the epoch. Note that by round we define a time interval where all cluster members have to transmit to their cluster head once. We show in this paper how the election process of cluster heads should be adapted appropriately to deal with heterogeneous nodes, which means that not all the nodes in the field have the same initial energy.

ii) Optimal Clustering

Previous work have studied either by simulation or analytically the optimal probability of a node being elected as a cluster head as a function of spatial density when nodes are uniformly distributed over the mesh field. This clustering is optimal in the sense that energy consumption is well distributed over all mesh and the total energy consumption is minimum. Such optimal clustering highly depends on the energy model we use. This optimum clustering and cluster head selection in wireless mesh network can be calculate through intelligent sensing scheme. For the purpose of this study we use similar energy model and analysis as proposed in. According to the radio energy dissipation model illustrated in Figure. in order to achieve an acceptable Signal-to-Noise Ratio (SNR) in transmitting an L-bit message over a distance *d*, the energy expended by the radio is given by:

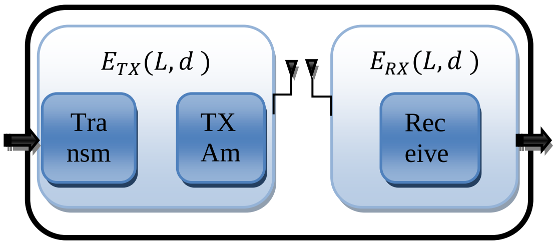

Figure 3: Radio Energy Dissipation Model

$$E_{T2}(l,d) = \begin{Bmatrix} L.E_{elec} + L.\epsilon_{fs}.d^2 \ \ if\ d \le d_0 \\ L.E_{elec} + L.\epsilon_{mp}.d^4 \ \ if\ d > d_0 \end{Bmatrix} \tag{1}$$

Here E_{elec} is the energy dissipated per bit to run the transmitter or the receiver circuit, ϵ_{fs} and ϵ_{mp} depend on the transmitter amplifier model we use, and *d* is the distance between the sender and receiver. By equating the two expressions at *d* = *d0*, we have $d_0 = \sqrt{\epsilon_{fs}/\epsilon_{mp}}$. To receive an L-bit message the radio expends$E_{Rx} = L.E_{elec}$.

B. Q-Learning Model

Q-learning is a model-free reinforcement learning technique, based on agents taking actions and receiving scalar rewards from the environment in response to those actions. It assigns Q-values to each possible action, representing the approximate goodness of the action. In the learning process, the agent selects and executes one action, then receives the reward, which it uses to update the Q-value. Over time the agent learns the real action values (costs), which it uses to select the most appropriate route. Q-learning has been widely applied in robotics, wireless ad hoc communications, etc. Its main challenge is to properly model the agent and define the Q-values.

In our cluster head routing scenario, each sensor node is an independent learning agent, and actions are routing options using different neighbours as the next hop toward the cluster head. The cluster head is defined as the cluster node with the best (lowest) routing cost to all sinks. The following provides detail for our Q-learning solution.

i) Q-values

Q-values represent the goodness of actions and the goal of the agent is to learn the actual goodness of the available actions. In our case, Q-values represent an estimate of the cost to route through a neighbour, a value composed of two parts. The first part is the broadcast hop count to reach all sinks from the agent

IJDACR
International Journal Of Digital Application & Contemporary Research

International Journal of Digital Application & Contemporary research
Website: www.ijdacr.com (Volume 2, Issue 3, October 2013)

and the second is the minimum battery level among the nodes on the route to the sinks through this neighbour. The first part accounts for energy efficiency by minimizing communication overhead. The second, the minimum battery of the nodes, allows very low powered nodes to be avoided. The two elements of the cost function are united with a hop count multiplier (hcm) value, which grows exponentially for decreasing battery levels. This means that when the batteries of the nodes are full, the routing cost of a neighbour is exactly the number of hops to reach the sinks; while with decreasing batteries this cost exponentially grows, giving preference to higher powered nodes on possibly longer routes

To initialize these values, we could use random values, as is common in many learning approaches. However, we use a more sophisticated approach that calculates an estimate based on the hop counts to individual sinks available in a standard routing table, thus speeding up the learning process. The initial Q-value for an action $a_{ni} = (ni, D)$ is:

$$Q(a_{ni}) = Q_{hops}(a_{ni}) * Q_{battery}(a_{ni}) = \sum_{d \epsilon D} hops\ d^{ni} * hcm(bat_{ni}) \quad (2)$$

Where $hops d^{ni}$ is the number of hops neighbour ni needs to reach sink d. The initial value of the battery element is set to the battery status of neighbour ni. Note that the hop-count estimation is an upper bound of the real costs, because subsequent hops are expected to be able to share routes to multiple sinks, decreasing the number of transmissions needed to reach the sinks. On the other hand, the battery element is expected to decrease, because battery levels decrease. Thus, the Q-values are expected first to drop, reflecting the learning of the real hop costs to reach the sinks, and then to slowly and constantly increase because of depleting energy on the nodes.

Updating a Q-value. To learn the real values of the actions, the agent uses the reward values from the environment. In our case, each neighbour to which a data packet is forwarded sends the reward, its best Q-value, piggybacked on its next data packet. The new Q-value of the action is:

$$(a_{ni}) = Q_{old}(a_{ni}) + \alpha\left(R(a_{ni}) - Q_{old}(a_{ni})\right) \quad (3)$$

Where $R(a_{ni})$ is the reward value and α is the learning rate of the algorithm. We use α = 1 to speed up learning and because we initialize the Q-values with non-random values. Therefore, with α= 1, the formula becomes $Q_{new}(a_{ni}) = R(a_{ni})$, directly updating the Q-value with the reward. Reward function. Intuitively the reward function is the downstream node's opportunity to inform the upstream neighbours of its actual cost for the requested action.

Thus, when calculating the reward, the node selects its lowest Q-value among all its actions and adds the real action cost:

$$R(n_{self}) = c_{ni} + \underbrace{min}_{n_i \in N} Q(a_{ni}) \quad (4)$$

Where c_{ni} is the cost of reaching node ni and is always 1 (hop) in our model. This propagation of Q-values upstream is piggybacked on usual DATA packets and allows all nodes to eventually learn the actual costs.

Exploration strategy (action selection policy). One final, important learning parameter is the action selection policy. A trivial solution is to greedily select the action with the best (lowest) Q-value. However, this policy ignores some actions that may, after learning, have lower Q-values, resulting in a locally optimal solution. Therefore, a trade off is required between exploitation of good routes and exploration among available routes. This problem has been extensively studied in machine learning. Here we chose the standard greedy strategy, which selects a random route with probability and the best route otherwise. Previous work showed that a dynamic cost function whose value changes continuously over time, such the one here based on battery level, results in implicit exploration of the routes. This is because the changing route costs force the protocol to switch to other, less costly routes, thus also learning their real costs.

C. Markov Decision Process

This scheme is not new in intelligent world; to use this technique in WMN is innovative and challenging task. It helps to obtain decisions from set of actions. Also Markov process says, the effects of an action taken in a state never depends on past state because it always depending on the present state. This model has a set of feasible states S, many probable actions A, an actual valued reward function R(s, a). Markov Model can take two type of action:

- Deterministic Actions- T: State (S) × Action (A) → State(S), for all states and actions we identify a novel state.
- Probabilistic Actions - T: State(S) × Action(A) → Prob(S), for all new states and actions we denote a probability distribution for next states , symbolized by the distribution P(s' | s, a).

Markov decision processes (MDP) provide a broad framework for modelling sequential decision making

IJDACR
ISSN: 2319-4863

IJDACR
International Journal Of Digital Application & Contemporary Research

International Journal of Digital Application & Contemporary research
Website: www.ijdacr.com (Volume 2, Issue 3, October 2013)

under uncertainty. MDP's have two sorts of variables: *state variables* $s_t$and *control variables* d_t, both of which are indexed by time t = 0, 1, 2, 3.... , T, where the horizon T may be infinity. A decision-maker or agent can be represented by a set of *primitives (u, p, β)* where $u(s_t, d_t)$ is a utility function representing the agent's preferences at time *t*, $p(s_{t+1} \mid s_t, d_t)$ is a Markov transition probability representing the agent's subjective beliefs about uncertain future states, and fit(0, 1) is the rate at which the agent discounts utility in future periods.

Infinite-horizon dynamic programming and Bellman's equation:

From the solution specified in section below, we will calculate sleeping behavior and selection probability of cluster head. Further simplifications are possible in the case of stationary MDP's. In this case the transition probabilities and utility functions are the same for all t, and the discount functions β_t (s_t, d_t) are set equal to some constant $\beta\ \varepsilon\ [0,1]$. In the finite-horizon case the time homogeneity of u and p does not lead to any significant simplifications since there still is a fundamental non-stationarity induced by the fact that remaining utility $\sum_{j=t}^{T} \beta^j u(s_j, d_j)$ depends on t. However in the infinite-horizon case, the stationary Markovian structure of the problem implies that the future looks the same whether the agent is in state s_t at time t or in state S_{t+k} at time t + k provided that $s_t = s_{t+k}$.

In other words, the only variable which affects the agent's view about the future is the value of his current states. This suggests that the optimal decision rule and corresponding value function should be time invariant, i.e. for all t ≥0 and all s ε S, $\delta_t^\infty = \delta(s)$ and$V_t^\infty = V(s)$, satisfies

$$\delta(s) = \underbrace{argmax}_{d\in D(s)} \left[u(s,d) + \beta \int V(s')p(ds'|s,d)\right] \tag{5}$$

Where, V is defined recursively as the solution to Bellman's equation, It is easy to see that if a solution to Bellman's equation exists, then it must be unique. Suppose that W(s) is another solution to above equation. Then we have

$$V(s) = \underbrace{max}_{d\in D(s)} \left[u(s,d) + \beta \int V(s')p(ds'|s,d)\right]$$

$$|V(s) - W(s)| \le \beta \int \underbrace{max}_{d\in D(s)} |V(s') - W(s')|p(ds'|s,d)$$

$$\le \beta \underbrace{sup}_{s\epsilon S} |V(s) - W(s)| \tag{6}$$

Since 0 < β< 1, the only solution to above equation is$\underbrace{sup}_{s\epsilon S} |V(s) - W(s)| = 0$.

D. Procedure of Proposed Model

1. Method is applied in a Mesh Field of Area 100×100 m.
2. The base Station is Placed at the Centre of Mesh Field INITIALLY; however we can change the Position of base Station.
3. Number of Nodes in the field is 100.
4. Advanced Node Have α time more energy than a normal node.
5. Hence Energy of Advanced Node becomes = initial Energy× (α). Total = initial Energy× (1+α).
6. Initially the dissipated energy is Zero & residual energy is the Amount of initial energy in a Node, Hence Total energy E_t also the Amount of residual energy because it is the sum of dissipated & residual energy.
7. Average distance between the cluster-head and the base station is calculated by D_{bs}= (0.765×one dimension of field)/2
8. Optimum Number Of Clusters are calculated by: K_{opt} =Calculate through our intelligence sensing algorithm.
9. The average distance between the cluster members and the cluster-head is calculated by

$$D_{ch} = \frac{\text{ONE DIMENSION OF FIELD}}{\sqrt{2 \times \pi \times Kopt}} \tag{7}$$

10. The total energy dissipated in the network during a round is calculated by:

$$E_t = bits\ data \times (2 \times n \times Etx + n \times Eda + Kopt \times Emp \times {D_{bs}}^4 + 4 \times n \times Efs \times {D_{ch}}^2) = \tag{8}$$

11. Also we calculated the average energy $E_a$of a Node after the particular round with the Knowledge of Total Energy and a particular number of round numbers.

$$E_a = E_t \times \left(\frac{1 - (r/Rmax)}{n}\right) \tag{9}$$

IJDACR
ISSN: 2319-4863

International Journal Of Digital Application & Contemporary Research

International Journal of Digital Application & Contemporary research
Website: www.ijdacr.com (Volume 2, Issue 3, October 2013)

12. We calculated the Dead Statistics before assigning a Cluster Head, and its value renewed every new round.
13. The New Expression for Optimum Probability can be calculated from Different Energy Levels and Optimum Probability Defined Earlier.
    $p(i)$ =It is the Energy Dependent value (Initial energy, average energy, total energy, residual energy) calculated from our Markov decision process.
14. The value of optimum probability will use by our reinforcement learning scheme to decide an optimum selection mechanism for routing.
15. Here, an Advanced will becomes Cluster Head, if a Temporary number assigned to it is Less than the Probability Structure Below,

$$T(s_i) = \begin{cases} \dfrac{P_i}{1 - P_i\left(r mod \dfrac{1}{P_i}\right)} & if \ \in G \\ 0 & otherwise \end{cases} \tag{10}$$

Here, Pi is come out from New Expression for Optimum Probability P_i

16. After an Advanced becomes Cluster Head, Energy Models are applied to calculate the Amount of Energy Spent by it on that Particular Round and complete the round of steady state phase.

$$E_{TX}(l,d) = \begin{cases} lE_{dec} + l\varepsilon_{fx}d^2, d < d_0 \\ lE_{dec} + l\varepsilon_{fx}d^4, d \geq d_0 \end{cases} \tag{11}$$

17. If a Node will Not an Advanced node and Discarded from the criteria above, than it goes to a Set of Normal node, and follow the behavior of normal node and complete the round of steady state phase.

III. RESULTS AND DISCUSSION

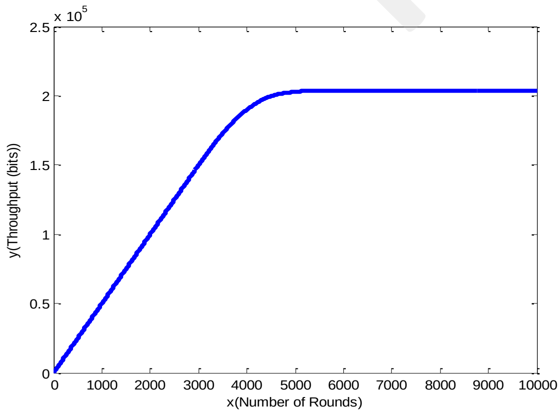

Figure 4: Throughput of Nodes with respect to Number of rounds in our protocol for 50 Nodes.

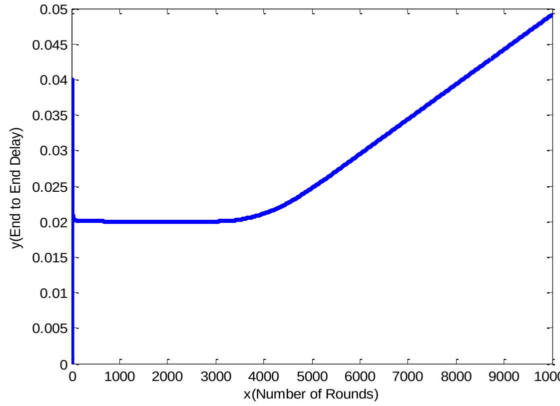

Figure 5: End to End Delay in packet Delivery with respect to Number of rounds for 50 nodes

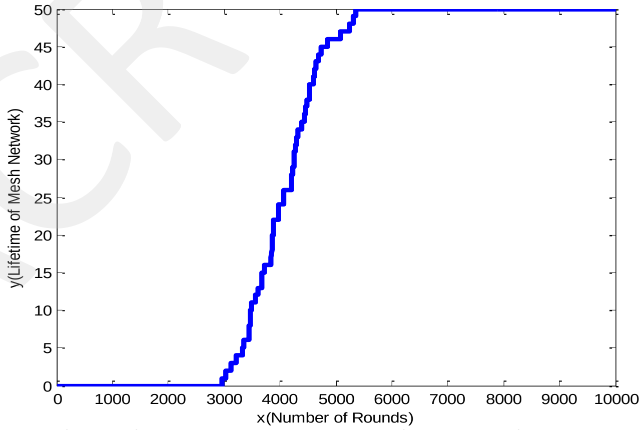

Figure 6: Lifetime of Mesh Devices in our protocol for Number of rounds (when considering 50 nodes in the network)

IV CONCLUSION

This work proposes an amend implementation on previously developed work which is further compare with the existing scenario. This protocol is used to determine the optimal probability for cluster formation in WSNs and sleeping behavior of sensor nodes through Markov Decision Scheme. Since the use of the optimal probability yields optimal energy-efficient clustering. Simulation results shows in terms of network lifetime, retained energy, data packets transmission, average information gathered, sleep strategy etc.

Our protocol successfully extends the stable region by being aware of heterogeneity through assigning probabilities of cluster-head election weighted by the relative initial energy of nodes. Proposed algorithm is implemented using MATLAB.

Real-time learning algorithm is developed to extend the lifetime of a sensor node to sense and transmit environmental events. The purpose of our new learning algorithm is to couple

IJDACR
International Journal Of Digital Application & Contemporary Research

International Journal of Digital Application & Contemporary research
Website: www.ijdacr.com (Volume 2, Issue 3, October 2013)

the sensor's sleeping behavior to the natural statistics of the environment therefore it can be in optimal synchronization with changes in the environment by sleeping with steady environment and staying awake when turbulent environment.

REFERENCES

[1] Miguel Elias M. Campista, Pedro Miguel Esposito, Igor M. Moraes, Luis Henrique M.K. Costa, Otto Carlos M. B. Duarte, "Routing Metrics and Protocols for Wireless Mesh Networks", Network, IEEE, 22(1):6-12, 2008.

[2] Maltz, David B. Johnson and David A., "Dynamic Source Routing in AdHoc Wireless Networks", In Imielinski and Korth, editors, Mobile Computing, volume 353. Kluwer Academic Publishers, 1996.

[3] Royer, E.M. Perkins, C.E.,"An Implementation Study of AODV Routing Protocol", In Wireless Communications and Networking Conference, 2000. WCNC, pages 1003 – 1008 vol.3 IEEE 2000.

[4] Yaling Yang, Jun Wang,"Design Guidelines for Routing Metrics in Multihop Wireless Networks", In IEEE INFOCOM - The 27th Conference on Computer Communications, pages 1615-1623. IEEE. 2008.

[5] Manolis Genetzakis, Vasilios A. Siris,"Contention-Aware Routing Metric for Multi-Rate Multi-Radio Mesh Networks", In 5th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks, pages 242-250. IEEE. 2008.

[6] Hongkun Li, Yu Cheng, Chi Zhou, Weihua Zhuang,"Minimizing End-to-End Delay: ANovel Routing Metric for Multi-Radio Wireless Mesh Networks", InIEEE INFOCOM the 28th Conference on Computer Communications, pages46-54. IEEE. 2009.

[7] Jonathan Guerin, Marius Portmann and Asad Pirzada,"Routing Metrics formulti radioWireless Mesh Networks", In Telecommunication Networks andApplications Conference,pages 343 - 348. 2007.

[8] Bellman Ford Algorithm. Wikipedia.org. [Online] 2010.en.wikipedia.org/wiki/Bellman-Ford_algorithm.

[9] T. M. Mitchell, Machine Learning: McGraw-Hill, 1997.

[10] S. J. Russell and P. Norvig, Artificial Intelligence: A Modern Approach: Pearson Education, 2003.

[11] Z. Ghahramani, "Unsupervised Learning," ed: University College London, September 2004.

[12] R. S. Sutton and A. G. Barto, Reinforcement learning: An Introduction: The MIT Press, 1998.

[13] L. P. Kaelbling, et al., "Reinforcement Learning: A Survey," Journal of artificial intelligence Research, vol. 4, pp. 237-285, May. 1996.

[14] R. Poor, "Wireless mesh networks", Sensors, February 2003.

[15] R. Poor, "Wireless mesh links everyday devices", Electronic Engineering Times, 5 July 2004.

[16] I.F.Akyildiz, I.H.Kasimoglu, "Wireless sensor and actor networks: research challenges", Ad Hoc Networks 2, pp.351-367, 2004.

[17] I.F.Akyildiz, O.B.Akan, "ARC for real-time traffic: ARC: the analytical rate control scheme for real-time traffic in wireless networks", IEE/ACM Transactions on Networking 12 (4), pp.634-644, 2004.

[18] I.F. Akyildiz, W. Su, Y. Sankarasubramaniam, E. Cayirci, "Wireless sensor networks: a survey", Computer Networks 38 (4) (2002) 393–422.

[19] C. Perkins, E. Belding-Royer, S. Das, "Ad hoc on-demand distance vector (AODV) routing", IETF RFC 3561, July 2003.

[20] Ritu Malik, Meenakshi Mittal, Isha Batra and Chander Kiran, "Article: Wireless Mesh Networks (WMN)", International Journal of Computer Applications 1(23):66–74, February 2010.

[21] Longjam Velentina Devi, Sheeba Parveen and Prof. Rizwan Beg, "Standard Activities of Wireless Mesh Networks", International Journal of Computer Applications 12(10), pp.12-16, January 2011.

[22] Jaydip Sen , "Secure Routing in Wireless Mesh Networks, Wireless Mesh Networks, Nobuo Funabiki (Ed.), InTech. Available from: http://www.intechopen.com/articles/show/title/secure-routing-in-wireless-mesh-networks, 2011.

[23] Jonathan Agre Fujitsu, "Secure Nomadic Wireless Mesh (Snow Mesh) 802.11 TGs ESS Mesh Networking Proposal", Laboratories of America Jonathan.agre@us.fujitsu.com Date: 2005-05-15 doc.: IEEE 802.11-05/596r1.

[24] John Paul M. Torregoza,Won-Joo Hwang, "Multi-channel Multi- Tranciever Routing Protocol for Wireless Mesh Network", in: IEEE Communications magazine, pp. 484–487, 2007.

[25] J. Jun, M.L. Sichitiu, "The nominal capacity of wireless mesh networks", IEEE Wireless Communications 10 (5)pp. 8–14,2003.

[26] S. Tierney, Mesh Networks, whitepaper of communitynetworking.org.

[27] R. Draves, J. Padhye, B. Zill, "Routing in multi-radio,multi-hop wireless mesh networks", in: ACM Annual International Conference on Mobile Computing and Networking (MOBICOM), pp. 114–128, 2004.

[28] Mesh Networking Forum, "Building the business case for implementation of wireless mesh networks", Mesh Networking Forum 2004, San Francisco, CA, October 2004.

[29] Ping Zhou, Xudong Wang, B. S. Manoj, and Ramesh Rao "An Optimizing Gateway placement for throughput in Wireless Mesh Networks" Department of Electrical and Computer engineering, University of California, USA, February, 2010.

[30] Mesh Networks Inc. Scalable routing technology. Available from: <http://www.meshnetworks.com/pages/technology/msr_atp_overview.htm>.

[31] Mesh Networks Inc; www.meshnetworks.com.

[32] Microsoft Mesh Networks. Available from: <http://research.microsoft.com/mesh/>.

[33] Theodore S. Rappaport, "Wireless Communications, Principles and Practice", 2nd edition, pp.643, 2003.

[34] T.-S. Jou, D.E. Eastlake, ESS MESH Network Study Group Meeting Minutes, May 2004.

[35] J. Beutel J. "Metrics for Sensor Network Platforms", in Proc. ACM Workshop on Real-World Wireless Sensor Networks (REALWSN'06), June2006.

International Journal of Digital Application & Contemporary research
Website: www.ijdacr.com (Volume 2, Issue 3, October 2013)

[36] A. Forster and A. L. Murphy, "Balancing Energy Expenditure in WSNs through Reinforcement Learning: A Study," in Proc. of the 1st Int. Workshopon Energy in Wireless Sensor Networks (WEWSN), 2008.

[37] Anna Forstery, Amy L. Murphyz, Jochen Schiller and Kirsten Terfloth, "An Efficient Implementation of Reinforcement Learning Based Routing on Real WSN Hardware", IEEE International Conference on Wireless and Mobile Computing, Networking and Communications, 2008.