Joint Design of Dijkstra's Shortest Path Routing and Sleep-Wake Scheduling in Wireless Sensor Networks

Susan Thomas
PG Scholar
Department of Electronics and Communication
Federal Institute of Science and Technology, Angamaly
Ernakulam, India

Gayathri I. K., Aiswariya Raj
Assistant Professors
Department of Electronics and Communication
Federal Institute of Science and Technology, Angamaly
Ernakulam, India

Abstract—**Wireless Sensor Network (WSN) are geographically disseminated self-sufficient sensors, used for tracking ecological or physique conditions. Since wireless sensors are battery powered the energy is considered as the important factor in the WSN. Sleep-wake schedule a systematic method to curtail energy consumption in the network by effectively arranging sensor network to sleep when transmission or reception is not required. Here supplementary delay may occur in every node through the path to sink node. So to minimize this problem Dijkstra's shortest path routing is used. In this method shortest path is chosen and it is able improve the network lifetime. Hence, a joint design of Dijkstra's shortest path routing and sleep-wake scheduling, called DSRSS, to increase the network lifetime is proposed. Simulation outputs demonstrate that DSRSS efficiently attains network lifetime maximization by reducing the energy consumption compared with other routing scheme.**

Keywords—Wireless communication, Sensor network, Sleep-wake scheduling, Dijkstra's shortest path routing, Energy consumption.

I. Introduction

Wireless sensor network possess a unique ability to remotely sense the conditions of the environment, it will not be easy to energize the batteries of massive number of wireless sensors as the networks are placed in faraway areas. Wireless sensors are used in different types of applications, for example, in [1] it is used for discover the activity of volcano and in [2] for observing the 70-meter tall redwood tree lifetime. Because of treacherous wireless links and restricted power supply, the routing should be cautiously designed to conserve energy for increasing the network lifetime, which is termed as the time till the first node completely depletes its energy [3], [4]. As sleep-wake scheduling organize sensor nodes to sleep mode when information transmission or reception is not required, it is an effective method to maximize the lifetime of a network. The sleep wake schedule used here can cause supplementary delay in every node through the route of the destination node. So to minimize this problem Dijkstra's shortest path routing packet is used. In the method it selects the shortest path and dispatches the data through the nodes in that path.

For the destination node to receive its corresponding transmission signal it had to wait for long time in the case of sleep-wake schedule method. In this scenario energy is not efficiently used. So to ameliorate lifetime of the network, efficient energy utilization, and to obtain high packet delivery ratio a shortest path based sleep-wake scheduling is used.

II. Related work

In ancient deterministic routing techniques, every node selects a particular node or entity to send information depending on the given metric. Techniques for improving the lifetime of the network is used in [3], [4]. The amount of residual energy of every nodes is pondered in [5], [6] for obtaining the routing path. In the time of heavy traffic nodes with high quantity of residual energy is used to send the information to the given destination. Another routing method titled as opportunistic routing is proposed [7], [8]. The method [9] involves more than one forwarders in which every dispatching nodes will receive the data and only the best node will dispatch [10] the information to the next node. In [11] EFFORT (energy efficient opportunistic routing technology in WSN) is introduced, here two benefits of this method is used i.e. transmission reliability and path trustworthy. The geographical information like Euclidean distance of the entities or the nodes is pondered [12] and depending on this a precedency queue is created and path is selected according to the precedency queue [13]. The clock contemporized sleep wake schedule is used in [14], [15]. Due to the contemporized clock, delay can be occur. An anachronous sleep-wake schedule is used in [16], [17], which reduces the delay in the network.

III. System Model

Here a wireless network composed of sensor nodes, in which every sensor node have the duty of both discovering events and imparting packets is considered. When a sensor node identifies an event, it convert data into several packets and transmits these packets to the destination by multiple-hop dispatching. Moreover, it is alleged that target nodes are re-energized; [18] so, the target nodes can be deemed as having infinite energy. The notations and symbols used is summarized in Table 1.

For the efficacious and the eminent usage of wireless sensor energy, a joint design of Dijkstra's shortest Path Routing and Sleep-wake Scheduling, named DSRSS is studied. When none of the packet to send or relay sensor nodes can veer into sleep mode so energy consumption can be curtailed by employing sleep-wake scheduling. Dijkstra's smallest path routing approach is able to both maximize lifetime of the sensor

978-1-5386-1887-5/17/$31.00 ©2017 IEEE

network and to reduce the consumption of energy by every nodes in the network.

A. Sleep-Wake Scheduling

In the method of sleep-wake scheduling, pictured in Fig. 1, the sensor node shifts betwixt sleep and active modes. An example is shown in the Fig. 1, if a wireless sensor node *u* have information to send to other nodes in the network, [18] the sleep-wake schedule is removed and node wakes up suddenly for a time T_b+T_s titled as the probing time or probing period. While the probing node *u* proclaim beacons for T_b time then waits for a time T_s interval for their ACK riposte. The transmission can be scheduled if a node successfully receives ACK riposte from any of the node in its forwarding set. If the node doesn't receive any ACK riposte or if there is no node in the forwarding set wakes up for receiving data, node returns to sleep mode and then resend beacons till at least one node in the forwarders set wakes up. And if there is no information to deliver for a sensor node, it enter into sleep mode for random length of intervals T_{sleep}. Based on the network configuration the time interval for sleep period is chosen arbitrarily. Time interval for the network with wireless sensor nodes is chosen small and for large network, the time interval can be increased.

In the method, every wireless sensor node with sleep-wake scheduling wakes up for an interval length Δ_{wake} titled as wake period and then it shifts to sleep mode of length T_{sleep}. Furthermore, the scheduling of every sensor node is independent. This premise is sensible and simple to execute in a wireless sensor network, because it need not contemporize clocks of sensor nodes and interchange extra information, which results in more communication overhead and intakes more energy. Let the wake-up event trails a Poisson process [18] with a given rate. In Poisson process, the value of time interval T_{sleep} is an exponentially arbitrary variable with norm $1/\lambda$. The memory less property is the advantage of Poisson scheduling, which makes computation simple. Assume that total number of information packets formed per unit time, is already known.

1) Formulation of Scheduling

In scheduling overhead, mainly there are two component, E_{prob} is afore the information transmission the energy used up for probing period and the wake periods energy usage is E_{wake}. Surmise that there are N sensor nodes in the network. The Eq. (1) shows the E_{prob} calculation and e_{tx} is the energy required for transmission, e_{rx} is the energy needed for reception and e_{idle} indicates the energy intake while idle state per unit time. Afore the transmission of information the number of expected probing period is $E[n_{prob}]$, which is showed in the Eq (2). That is, [18] T_b and T_s time units is needed by the sender node to transmit beacon in every probing period, and for waiting ACK riposte. Also at final probing period the sender node takes more time units for ACK reception.

$$\mathrm{E_{prob}} = \mathrm{E}\big[\mathrm{n_{prob}}\big].\,(\mathrm{T_b}.\,\mathrm{e_{rx}}{+}\mathrm{T_s}.\,\mathrm{e_{idle}}){+}\mathrm{T_s}.\,\mathrm{e_{rx}} \qquad (1)$$

$$\mathrm{E}\big[\mathrm{n_{prob}}\big] = \frac{1}{1-\mathrm{e}^{-\lambda.\mathrm{F}.\Delta_{\mathrm{wake}}}} \qquad (2)$$

Table 1

Notation	Description
T_b	Beacon transmission or reception time
T_a	ACK transmission or reception time
T_s	For awake forwarders time required to wait ACK
N	Sensor nodes total number
Δ_{wake}	Wake period time or length
λ	Poisson process wake up rate
γ	Data packets formed total number
e_{tx} or e_{rx} or e_{idle}	Sending or receiving or idle state energy consumption
$E[n_{prob}]$	Afore the information transmission the number of expected probing period.
E_{prob}	Afore the information transmission the energy consumption for probing period
E_{wake}	Wake periods consumption of energy

The energy usage for wake period is denoted by E_{wake}, which is expressed in Eq. (3), it comprises the energy cost for wake periods and for reception of beacon or signal, together with transmission of ACK.

$$\mathrm{E_{wake}} = \mathrm{E}[\mathrm{n_{probe}}].\,\Delta_{\mathrm{wake}}.\,\frac{\Delta_{\mathrm{wake}}.\,\mathrm{e_{idle}}}{\frac{1}{\lambda}+\Delta_{\mathrm{wake}}}$$

$$+\gamma.\,(T_b.\,e_{rx} + T_a.\,e_{tx} + T_s.\,e_{idle}) \qquad (3)$$

If a receiver node is in wake mode and if it receives a probing signal, in Eq. (3) i.e. $T_b.e_{rx}$, had to riposte with an ACK and need T_s time units to wait for the information transmission.

B. Dijkstra's Shortest Path Routng

Dijkstra's algorithm is known as the sole source smallest path [19]. It is also designated as the sole source smallest path problem. It calculates distance of the smallest path from the source node to every leftover nodes in the sensor network. The working of Dijkstra's algorithm is by calculating the smallest path from the source node to its neighbouring nodes which are close to the source. The Dijkstra's algorithm works adequately in the weighted directed graph and the link cost or edges should not be lower than zero. If the edge value is smaller than zero then the correct smallest path cannot be determined. At the ending level, a group of entities or nodes which are close to the source and the nodes that laid outside the group are obtained and new group is formed. Also it holds the smallest distance calculated and keeps the estimated distance from the given source node to the destination node. Dijkstra's algorithm calculate the adjacent node by keeping the new groups of nodes in a precedency queue. The algorithm works by keeping the smallest distance of node *v* from the source to the target node. From the source to itself the smallest distance is zero. The distance is set as infinity for all the other nodes to specify that those nodes are not processed yet.

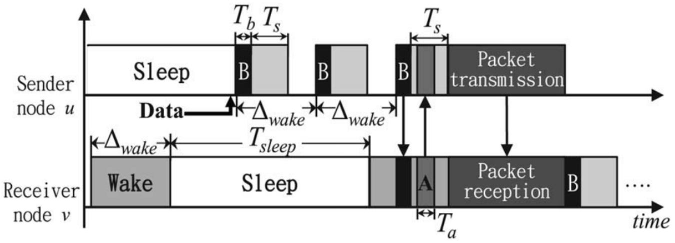

Fig. 1. Sleep-wake Scheduling Illustration

After finishing the processing of every nodes using the algorithm the smallest distance from subject to object node is obtained.

1) Algorithm

Consider the initial node is termed as the source node and the distance of a node *u* be the length from the starting node or source node to *u*. In Dijkstra's algorithm [19] some random distance values are assigned.

1. For every node in the network a distance value is assigned and for the initial or source node the distance is zero and for other nodes set infinity.
2. The uninspected nodes are marked after setting the starting node as the current node. A group of all such nodes are termed as uninspected group or set is formed.
3. The spacing betwixt the current node and all the uninspected neighbouring nodes of the current node is calculated. Then this distance obtained is compared with the values assigned formerly and discard the large distance value and hold the small value.
4. After evaluating every neighbour of current node it is marked as inspected node and eliminate it from the uninspected group. Evaluation of an inspected node is not required.
5. If destination or target node is in the inspected group or the smallest path of nodes in the uninspected set is infinity the algorithm is ceased.
6. Else, a new current node is chosen from the uninspected group which is having smallest distance, then repeat from step 3 onwards.

To acquire the smallest path from the source and the destination node the Dijkstra's algorithm firstly tag the spacing from the initial or primary node to whole other node on the network with infinity. This is not to imply that there is an unbounded or an infinite spacing, but to imply that those nodes are to be inspected. Then, at every loop, a current node is elected [19]. For the primary loop, the beginning point is chosen as the current node, and the distance to it is assigned to zero. For successive repetitions after the first, the present node will be the closest uninspected node to the beginning point. The distance is updated to every uninspected node from the current node that is directly joined to it. This is done by obtaining the sum of the spacing betwixt the value of the current node and the value of the uninspected nodes, and if that value is less than its current value then the uninspected node is relabelled with the sum value. If the formerly recognized route is bigger than the route to it via the current or present node the node is remarked with the smaller distance value. The current node is marked as the inspected node after updating the distances to each neighbouring node, and the uninspected node with lowest spacing from the primary node is picked as the present node. The nodes will not be checked again if they are marked as inspected and are tagged with the smallest route from the beginning point. The process is pursued till the neighbouring nodes with the smallest spacing is obtained, then marking the current node as inspected and moving onto the closest uninspected node till the destination is marked as inspected. The algorithm could be ceased either the smallest path to the target is determined from the primary node i.e. beginning point or destination is marked as inspected node.

2) Pseudo Code

In Dijkstra's algorithm, there is a group of uninspected nodes in set R. In the code the line "*u* ← node in R with minimum distance[u]", hunts for a node u in the node group R which have the less distance[u] value. The spacing betwixt two neighbours *u* and *v* is given by *length (u, v).* The variable *cost* represent the measure of the route from the source node to the adjacent node *v* through *u* [19]. The current route is interchanged with new *cost* path, if path obtained is smaller than the current smallest path marked for *v*. The pseudo code for Dijkstra's shortest path routing is given below. Pseudo code is the miniature working of the main coding part of the algorithm.

```
Function Dijkstra(Network, subject node):
    Form node group R
    // Initializing
    For every node v in Network:
        // Distance from subject node to node v is not known
        Distance[v] ← INFINITY
        // From the subject node, previous or former node in ideal path
        Previous[v] ← UNDEFINED
        // Initial case all nodes are in group R, uninspected group
        Move v to R
    // Subject node to itself the distance is zero
    Distance[subject node] ← 0
    While R is not a null group:
        // Select the node with minimum distance
        u ← node in R with minimum distance[u]
        Eliminate u from group R
        // For v is in group R
        For every neighbour v of u:
            Cost ← distance[u] + length (u, v)
            // Smallest route to node v is obtained
            If cost< distance[v]:
                Distance[v] ← cost
                Previous[v] ← u
    Return distance[], previous[]
```

The above pseudo code manifest the procedure to obtain the shortest path using the Dijkstra's algorithm.

IV. PERFORMANCE EVALUATION

To affirm the proposed routing method DSRSS, which collectively ponders Dijkstra's shortest path routing and sleep-wake scheduling, can efficaciously emend the sensor network lifetime. By comparing ASSORT with DSRSS i.e., opportunistic Routing with sleep-wake scheduling via NS 2.35 simulator [20]. In the method all the forwarders of the sending node is set to awake. If node *u* is not sending any packets its forwarders are set to sleep mode. The forwarding set of node *u* are the neighboring nodes. The neighbors of node *u* are the nodes which are at a distance less than the average communication extent of the node *u*. The average communication extent of a wireless sensor is 250meter and the distance is calculated by using the Euclidean distance equation.

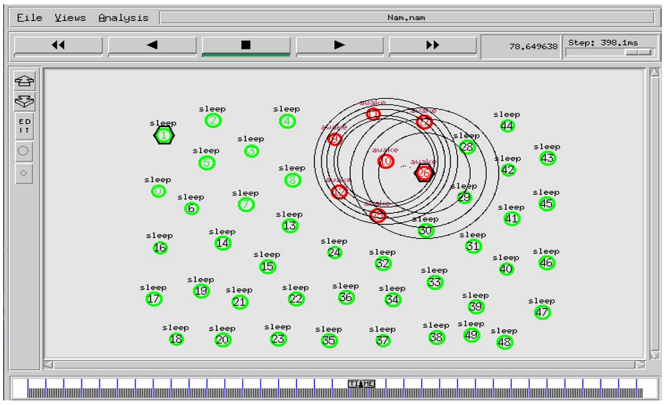

Fig. 2. Network of 50 nodes

The Poisson process with wake up event rate λ normal value is 0.1. So the norm information generation rate γ is 1-20 packets per second. All the sensor nodes dispatch the packets towards the target node, and if a packet is redeemed to the destination the transmission is successful. Other network setup used in the simulation are as follows. The deployment area is 1670mx970m and the transmission extent of the sensor is 250meter. The information packet size is 46 bytes and the channel capacity is 250Kbps. The initial energy of the node is 50J and the total number of sensors is 10-50.

The Fig. 2 is the network with 50 wireless sensor nodes. The transmission is visible in the network. The sleep node and awake nodes are labelled in the network. The source node is node 1 and the target node is node 26. The route from node 1 to node 26 is chosen by using the Dijkstra's shortest path routing method. After obtaining the shortest path the sleep wake scheduling is implemented. Then make the node awake either by random or all the forwarders of the sending node is set to awake. In both the ways the energy intake can be reduced and thereby the network lifetime can be ameliorated.

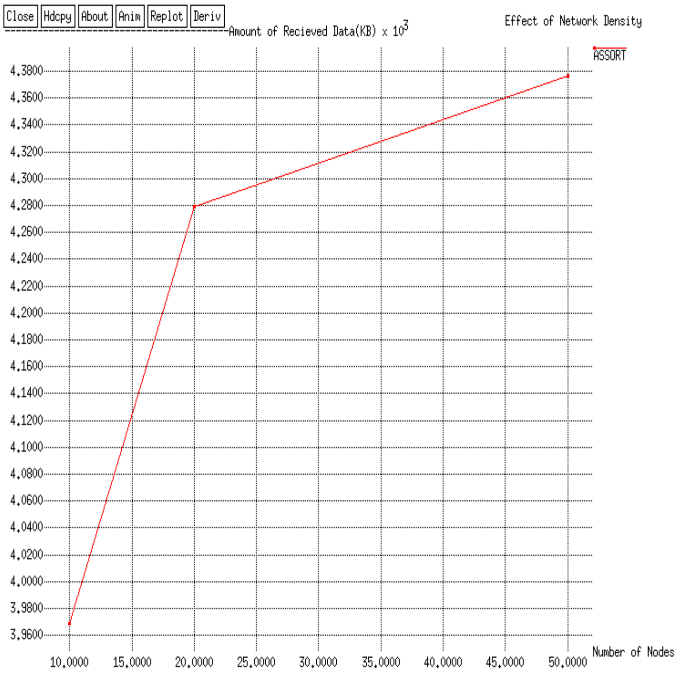

Fig. 3. Effect of Network Density in ASSORT

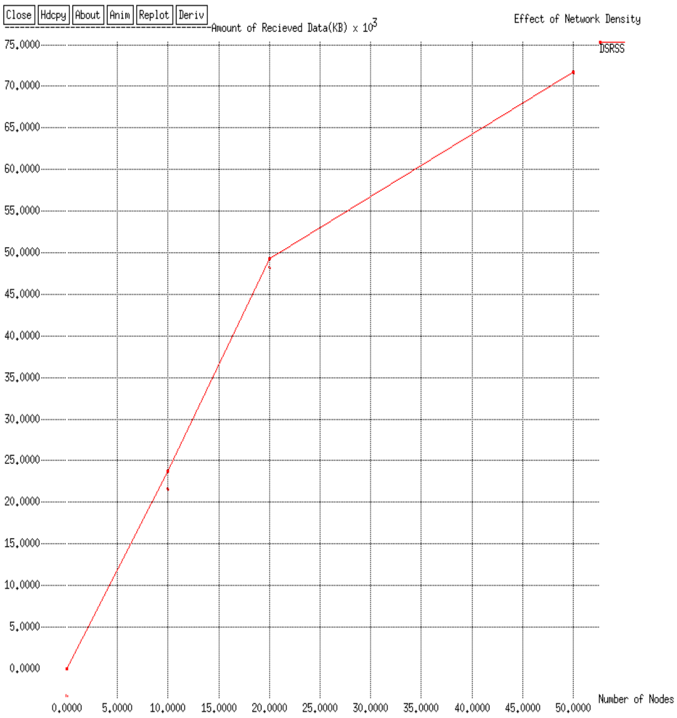

Fig. 4. Effect of Network Density in DSRSS

The Fig. 3 is the effect of network density in ASSORT i.e. opportunistic routing with sleep-wake schedules. When the node density increases the amount of data received also increases. From the graph it is clearly visible that the maximum amount of data received is 4.37KB. But in the Fig 4 the network density effect in DSRSS is plotted where maximum amount of data received is 73KB. DSRSS method have better performance in any network population.

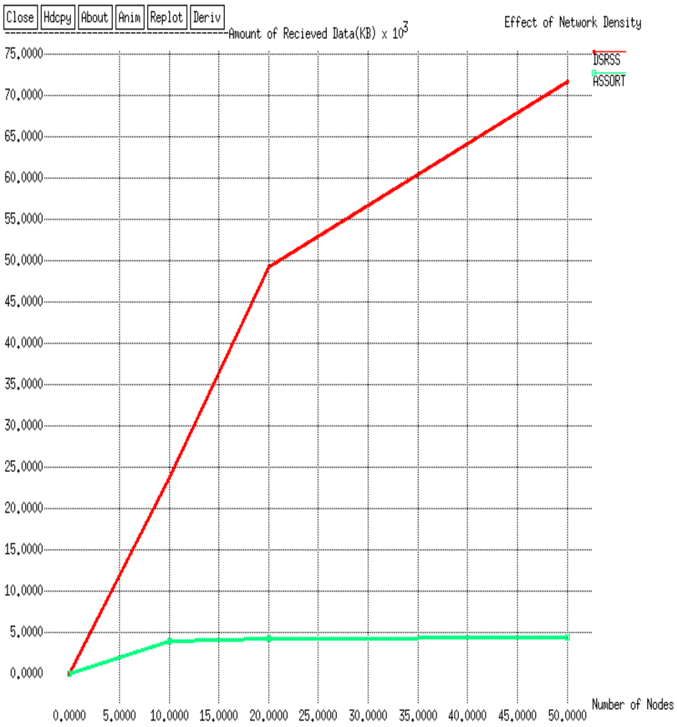

Fig. 5. Comparison of Network Density

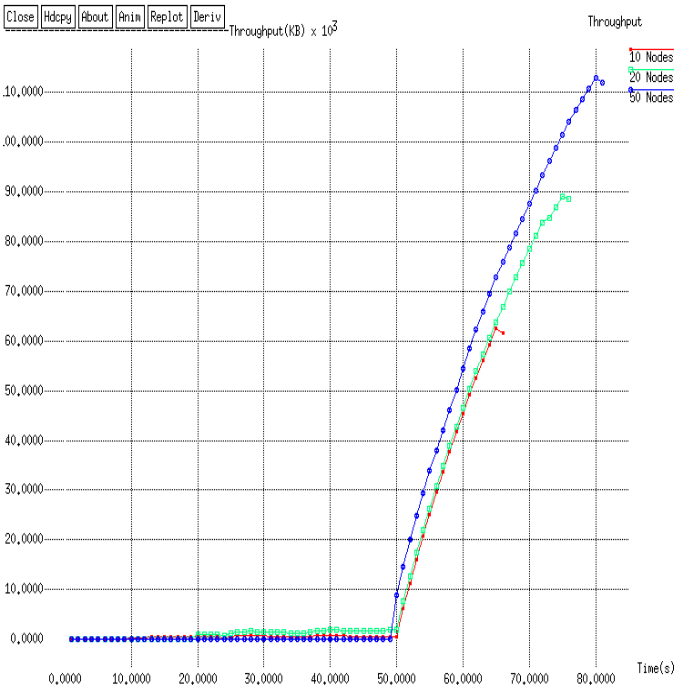

Fig. 6. Throughput of DSRSS for different Network Density

The Fig. 5 exhibits the comparison of effect of network density betwixt DSRSS and ASSORT. The density of 10, 20 and 50 nodes scenarios are considered. From the graph plotted the amount of data received for DSRSS is more than ASSORT. So DSRSS is preferred over ASSORT in high density scenarios. From the Fig. 6 the throughput of DSRSS for various network density is pictured. it is visible that as the network density increases from 10 nodes to 50 nodes the throughput increases. So the rate of successful message delivery increases with time.

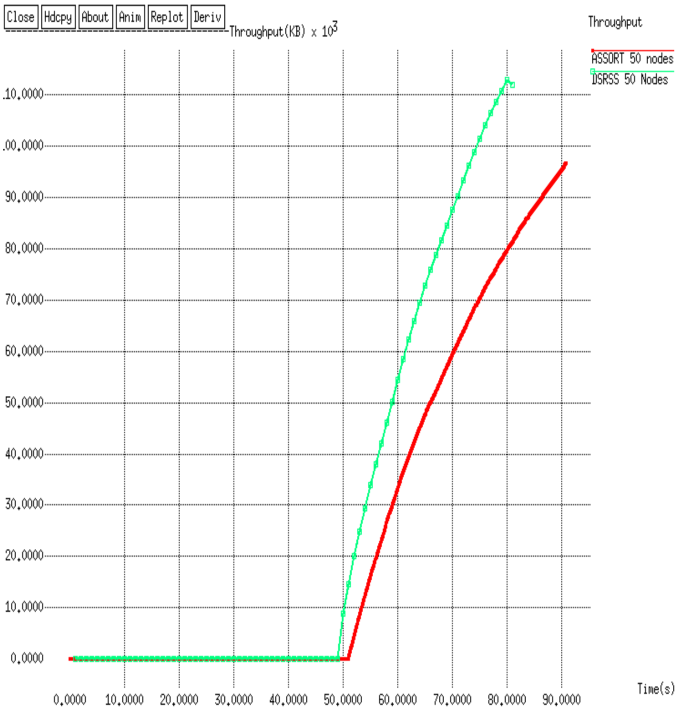

Fig. 7. Throughput of ASSORT and DSRSS of 50 Nodes

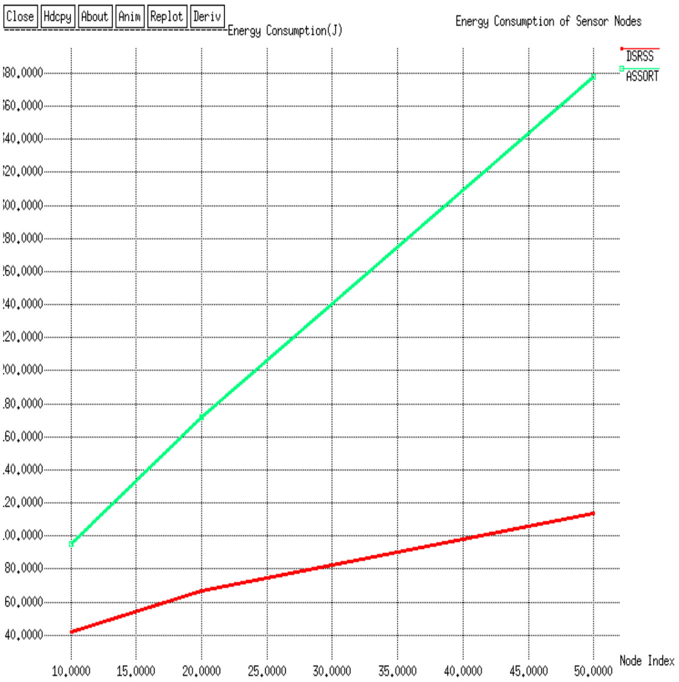

Fig. 8. Comparison of Total Energy Consumption

The Fig. 7 depicts the throughput graph of both ASSORT and DSRSS. Throughput is given by the ratio of amount of data received to the time. In a communication channel it is the rate of successful message delivery. In the case of DSRSS first the throughput attains maximum and then decreases. The throughput is increasing in the case of ASSORT, but the maximum value is less than DSRSS. The throughput of DSRSS is better i.e. Dijkstra's shortest path routing provides the successful message delivery than ASSORT.

The Fig. 8 shows the comparability of total energy intake of ASSORT and DSRSS. In order to investigate the energy consumption further, the energy consumption under different node numbers is recorded. The energy consumption for the DSRSS method is very low when compared to the ASSORT. In small network density the energy consumption is less and it increases as the number of nodes in the network increases. By using the Dijkstra's shortest path routing and sleep-wake schedules the total energy consumption can be reduced and thereby the network lifetime can be increased. So the energy can be utilized efficiently in this technique.

V. CONCLUSION AND FUTURE SCOPE

DSRSS, which is a joint design of Dijkstra's shortest path routing and sleep-wake scheduling, to meliorate network-lifetime for wireless sensor networks is proposed. The functioning of DSRSS is based on determining the shortest path for the information transmission and thereby execution of sleep-wake schedules after getting the shortest path. Simulation results show that DSRSS attains network lifetime betterment likened with ASSORT. Also the energy consumption in the networks is reduced. DSRSS also provides the quick delivery of the message.

The current performance evaluations are carried out by using NS 2.35 simulator, so one of the future scope is to execute DSRSS on real sensor devices, like MICAz. Even though DSRSS uses sleep-wake scheduling, which is carried out locally without extra message interchange, there are distinct potential defies that need to be resolved. Also the dispatching node can be picked based on the link standard, which is the ratio of number of retransmitted packets to the total number of packets formed.

REFERENCES

[1] G. Werner-Allen, K. Lorincz, J. Johnson, J. Lees, and M. Welsh, "Fidelity and yield in a volcano monitoring sensor network," in Proc. USENIX Symp. Oper. Syst. Design Implementation (OSDI), 2006, pp. 381–396.

[2] G. Tolle, J. Polastre, R. Szewczyk, D. Culler, N. Turner, K. Tu, S. Burgess, T. Dawson, P. Buonadonna, D. Gay, andW. Hong, "A Macroscope in the Redwoods," in Proc. SenSys, 2005, pp. 51–63.

[3] J. Park and S. Sahni, "An online heuristic for maximum lifetime routing in wireless sensor networks," IEEE Trans. Comput., vol. 55, pp. 1048–1056, 2006.

[4] Y. Wu, S. Fahmy, and N. Shroff, "On the construction of a maximum-lifetime data gathering tree in sensor networks: NP-completeness and approximation algorithm," in Proc. IEEE Conf. Comput. Commun. (INFOCOM), 2008, pp. 1013–1021.

[5] J.-H. Chang and L. Tassiulas, "Maximum Lifetime Routing In Wireless Sensor Networks," IEEE/ACM Trans. Netw., vol. 12, pp. 609–619, Aug. 2004.

[6] K. Kar, M. Kodialam, T. Lakshman, and L. Tassiulas, "Routing for network capacity maximization in energy-constrained ad-hoc networks," in Proc. IEEE Conf. Comput. Commun. (INFOCOM), Mar. 3/Apr. 2003, vol. 1, pp. 673–681.

[7] S. Biswas and R. Morris, "ExOR: Opportunistic multi-hop routing for wireless networks," in Proc. SIGCOMM Internet Meas. Workshop, 2005, pp. 133–144.

[8] S. Chachulski, M. Jennings, S. Katti, and D. Katabi, "Trading structure for randomness in wireless opportunistic routing," in Proc. SIGCOMM Internet Meas. Workshop, 2007, pp. 169–180.

[9] K. Zeng, W. Lou, and H. Zhai, "On end-to-end throughput of opportunistic routing in multirate and multihop wireless networks," in Proc. IEEE Conf. Comput. Commun. (INFOCOM), 2008, pp. 816–824.

[10] J. Wu, M. Lu, and F. Li, "Utility-based opportunistic routing in multi-hop wireless networks," in Proc. Int. Conf. Distrib. Comput. Syst. (ICDCS), 2008, pp. 470–477.

[11] M. C.-C. Hung, K. C.-J. Lin, C.-F. Chou, and C.-C. Hsu, "EFFORT: Energy Efficient Opportunistic Routing Technology In Wireless Sensor Networks," Wireless Commun. Mobile Comput., vol. 13, no. 8, pp. 760–773, 2013.

[12] K. Zeng, W. Lou, and Y. Zhang, "Multi-rate geographic opportunistic routing in wireless ad hoc networks," in Proc. Military Commercial Commun. Conf. (MILCOM), 2007, pp. 1–7.

[13] K. Zeng, W. Lou, J. Yang, and D. Brown, "On geographic collaborative forwarding in wireless ad hoc and sensor networks," in Proc. Int. Conf. Wireless Algorithms Syst. Appl., 2007, pp. 11–18.

[14] W. Ye, J. Heidemann, andD. Estrin, "Medium access control with coordinated adaptive sleeping for wireless sensor networks," IEEE/ACMTrans. Netw., vol. 12, no. 3, pp. 493–506, Jun. 2004.

[15] T. van Dam and K. Langendoen, "An Adaptive Energy-Efficient MAC Protocol For Wireless Sensor Networks," in Proc. SenSys, 2003, pp. 171–180.

[16] J. Kim, X. Lin, N. Shroff, and P. Sinha, "On Maximizing The Lifetime Of Delay Sensitive Wireless Sensor Networks With Anycast," in Proc. IEEE Conf. Comput. Commun. (INFOCOM), 2008, pp. 807–815.

[17] J. Polastre, J. Hui, P. Levis, J. Zhao, D. Culler, S. Shenker, and I. Stoica, "A unifying link abstraction for wireless sensor networks," in Proc. SenSys, 2005, pp. 76–89.

[18] C. C. Hung, M. S. Kuo, S. C. Wang and C. F. Chou, "Joint Design Of Asynchronous Sleep-wake Scheduling And Opportunistic Routing In Wireless Sensor Networks", IEEE Transactions On Computers, Vol. 63, No. 7, JULY 2014.

[19] A.R.Welekarand and S.S. Dorle, "Adoption of Shortest Path (Dijkastra) Algorithm for Distributed VANET using Grid Computing," in Proc. IEEE Conf. Info. Commun. (ICICT), Dec 2011.

[20] Neha Singh and Vinita Mathur, "Network Simulator NS2-2.35", IJARCSSE International Journal of Advanced Research in Computer Science and Software Engineering, Vol 2, No. 5, May 2012.