

# Maximize the Lifetime of WSN Using New Backbone Scheduling Based Algorithm

S.Alagumuthukrishnan, Dr.K.Geetha

**Abstract**—In this paper, a new algorithm called New Backbone Scheduling (NBS) was proposed which enables sleep scheduling at random amount of time. NBS schedules multiple overlapped backbones at random time so that the network energy usage is evenly distributed among all sensor nodes. In this way, the energy of the entire sensor nodes in the network is fully utilized which in turn increases the network lifetime. Wireless Sensor Networks are key for various applications such as military for battlefield surveillance and many other industrial and consumer applications, such as industrial process monitoring and control, machine health monitoring etc. In these applications, sensor nodes use batteries as the only energy source. Therefore, energy efficiency becomes vital. So NBS is designed for WSNs has redundant sensor nodes. NBS forms multiple overlapped backbones which work alternatively to increase the network lifetime. In NBS, traffic is only forwarded by backbone sensor nodes, and the rest of the sensor nodes turn off their radios to save energy. The rotation of multiple backbones makes sure that the energy consumption of all sensor nodes is balanced, which fully utilizes the energy and achieves a longer network lifetime compared to the existing techniques. The simulation result achieves higher packet delivery ratio and low end to end delay and thus increases the network lifetime.

**Index Terms**— New Backbone Scheduling, Wireless Sensor Networks, multiple overlapped backbones, Network life time, redundant nodes, packet delivery ratio, end to end delay

## 1 INTRODUCTION

### 1.1 Wireless Sensor Network:

A wireless sensor network (WSN) consists of spatially distributed autonomous sensors to cooperatively monitor physical or environmental conditions, such as temperature, sound, vibration, pressure, motion or pollutants. The development of wireless sensor networks was motivated by military applications such as battlefield surveillance. They are now used in many industrial and civilian application areas, including industrial process monitoring and control, machine health monitoring, environment and habitat monitoring, healthcare applications, home automation, and traffic control. In addition to one or more sensors, each node in a sensor network is typically equipped with a radio transceiver or other wireless communications device, a small microcontroller, and an energy source, usually a battery.

A sensor node might vary in size from that of a shoe box down to the size of a grain of dust although functioning "motest" of genuine microscopic dimensions have yet to be created. The cost of sensor node is similarly variable, ranging from few hundreds to few thousands, depending on the size of the sensor

network and the complexity required of individual sensor nodes. Size and cost constraints on sensor nodes result in corresponding constraints on resources such as energy, memory, computational speed and bandwidth [1]. A sensor network normally constitutes a wireless ad-hoc network, meaning that each sensor supports a multi-hop routing algorithm. i.e. (several nodes may forward data packets to the base station). In computer science and telecommunications, wireless sensor networks are an active research area. Sensor networks have emerged as a promising tool for monitoring (and possibly actuating) the physical worlds, utilizing self-organizing networks of battery-powered wireless sensors that can sense process and communicate. In sensor networks, energy is a critical resource, while applications exhibit a limited set of characteristics. The requirements and limitations of sensor networks make their architecture and protocols both challenging and divergent from the needs of traditional Internet architecture [2].

#### **Sensor node:**

This is a mobile node moving freely to monitors the physical environment. Once it detects its physical target, it generates a data packet and sends it to the sink node via the wireless channel [3]. The processor in the sensor node may be set the threshold value to compare with the detected data before it generates and sends a data packet.

#### **Sink node:**

This node collects all data packets from sensor nodes and uses them to analyze their targets.

#### **Node Structure:**

- S.Alagumuthukrishnan, Associate Professor, Department of Information Technology, R.V.S.College of Engineering and Technology, Coimbatore. Tamilnadu India  
E-Mail:alagumuthukrishnan@gmail.com
- Dr.K.Geetha, Professor, Department of Information Technology, Karpagam Institute of Technology, Coimbatore, Tamilnadu India  
E-Mail:geetha.arulmani@gmail.com

A sensor node can be divided into four basic modules: transducer, processor, communications and power. The transducer module contains the physical sensing device and an analog-to-digital converter (ADC) [4],[5],[6]. The sampled data is then passed to the processor, where it is stored in A sensor node can be divided into four basic modules: transducer, processor, communications memory. Some applications merely require streaming of raw data while other applications require periodic sampling of the data.

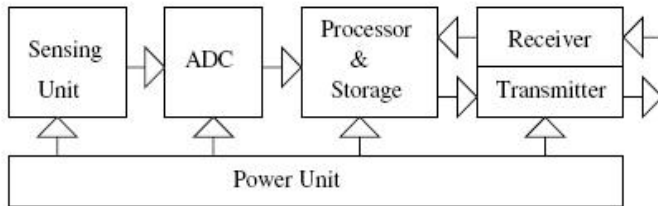


Figure: 1.1.Node Structure

However sophisticated applications require preprocessing of the data to extract important information so that transmission bandwidth can be preserved by simply transmitting the essential information (e.g., alerting the operator of a critical event). Local processing capability is also important for applications in which the sensor supports bidirectional communication. In these cases, the users can query the sensor either for status or for a history of previous samples of data. The communication module consists of a short-range radio transceiver. The power module is used to house the battery and provides energy to the other modules. The functions of all four modules partially depend on the role of the sensor node. A sensor node can operate in one of the three roles: data collector, cluster head, or data relay. If a node is a data collector, the transducer module directly passes the sampled data to the communication module for transmission. A cluster-head node gathers the sensed data from the cluster members and performs data processing to aggregate multiple signals into one signal. If a node works as a relay, [4] [5]it receives the data from nearby nodes and transmits the data to other nodes or the base station.

### Energy Constraints:

The design of each system component can be optimized to minimize energy consumption. Energy consumption occurs in three aspects: sensing, communication, and data processing. Algorithmic modifications can often result in significant energy savings. Usually the communication of data consumes much more energy than sensing and data processing.

Energy is the scarcest resource of WSN nodes, and it determines the lifetime of WSNs[6]. WSNs is meant to be deployed in large numbers in various environments, including remote and-hostile regions, with ad-hoc communications as key. Therefore the algorithms and protocols are designed with the following features such as Lifetime maximization, Robustness and fault tolerant, Self-configuration.

### Energy Efficiency:

Sensors have limited energy supply and are usually

deployed in a hostile environment. Recharging is almost impossible during the operation. Therefore, long-term applications require energy-efficient solutions.

### Scalability:

Large-scale WSNs usually consist of tens of thousands of sensor nodes at least two orders of magnitude more sensors per router than conventional wireless networks. Highly localized and distributed solutions are required.

### Goals of WSN:

- Determine the value of physical variables at a given location
- Classify a detected object, and
- Track an object.

### Requirements of WSN:

- Use of a large number of sensors,
- Attachment of stationary sensors,
- Low energy consumption,
- Self organization capability,
- Collaborative signal processing

## 2 LITERATURE SURVEY:

### 2.1 Constructing K-Connected K-Dominating Set in Wireless Sensor Networks

An important problem in wireless sensor networks is to select a few nodes to form a virtual backbone that supports routing and other tasks such as area monitoring. The construction of a k-connected k-dominating set (k-CDS) as a backbone to balance efficiency and fault tolerance

### 2.2 An Extended Localized Algorithm for Connected Dominating Set Formation in Wireless Networks

Efficient routing among a set of mobile hosts is one of the most important functions in wireless networks. Routing based on a connected dominating set is a promising approach, where the search space for a route is reduced to the hosts in the set. A set is dominating if all the hosts in the system are either in the set or neighbors of hosts in the set[3]. The efficiency of dominating set- based routing mainly depends on the overhead introduced in the formation of the dominating set and the size of the dominating set. A localized formation of a connected dominating set called marking process and dominating-set-based routing. A dominant pruning rule to reduce the size of the dominating set. This dominant pruning rule (called Rule k) is a generalization of two existing rules (called Rule 1 and Rule 2 respectively) that the vertex set derived by applying Rule k is still a connected dominating set [2].

### 2.3 Optimal Sleep Scheduling in Sensor Networks for Rare-Event Detection

Lifetime maximization is one key element in the design of sensor-network-based surveillance applications. A protocol for node sleep scheduling that guarantees a bounded-delay sensing coverage while maximizing network lifetime[5]. Our sleep scheduling ensures that coverage rotates such that each point in the environment is sensed within some finite interval of time, called the detection delay[8]. The framework is optimized for rare event detection and allows favorable compromises to be achieved between event detection delay and lifetime without sacrificing (eventual) coverage for each point.

## 2.4 Energy and Latency Control in Low Duty Cycle MAC Protocols

Several MAC protocols such as S-MAC and T-MAC have exploited scheduled sleep/wakeup cycles to conserve energy in sensor networks. Two new algorithms to control and exploit the presence of multiple schedules to reduce energy consumption and delay. The first one is the global schedule algorithm (GSA)[10]. GSA is a fully distributed algorithm that allows a large network to converge on a single global schedule to conserve energy. Secondly, demonstrate that strict schedules incur a latency penalty in a multi-hop network when packets must wait for the next schedule for transmission.

To reduce latency in multi-hop paths, the fast path algorithm (FPA). FPA provides fast data forwarding paths by adding additional wake-up periods on the nodes along paths from sources to sinks. Sleep/wakeup MAC protocols establish and maintain a schedule about when nodes listen for possible transmissions and when they sleep. When a new node joins the network, it listens for and tries to adopt an existing schedule. Sleep durations and energy savings are maximized when all nodes are on the same schedule. Multiple schedules can occur in large networks even though the protocols are biased to promote a single schedule[9]. Such nodes naturally generate different schedules because they must choose independently. Finally, if nodes are moving, then they can easily move between different parts of the network with different schedules. Sleep/wakeup protocols must be prepared to detect and track neighbors operating on different schedules to maintain global network connectivity. Since multiple schedules cannot be prevented in a large distributed system we next describe the global schedule algorithm (GSA), which allows all nodes to converge on a common schedule. The schedule age indicates how long the schedule has existed in the network. When a node originates a new schedule, it records the time when the schedule is generated. When it later advertises the schedule, it puts the schedule age into the packet [10].

## 3 EXISTING WORK

There are algorithms which maximize the network lifetime

for Wireless Sensor Networks (WSNs) that involve long-term and low-cost monitoring. The Connected Dominated Partition (CDP) based algorithm increases the network lifetime by combining the Backbone Scheduling (BS) and Duty Cycle (DC). In order to perform the packet transmissions the Backbone node should be in the frequency range that forms the Connected Dominated Partition of the disjoint Backbones. The same redundancy is used for the multihop communication because of the light traffic load and the stable wireless links[7].

### 3.1 Drawbacks

- The WSN applications require redundant sensor nodes to achieve fault tolerance and quality of service of the sensing.
- The scheduling problem is formulated as the Maximum Lifetime Scheduling problem that fails for infinite change in the network.
- The Connected Dominating Set (CDS) which forms the backbone, however the single backbone does not prolong the network lifetime.

## 4 PROPOSED WORK

- New Backbone Scheduling (NBS) schedules multiple Connected Dominating Set (CDS) at random amount of time so that the network energy consumption is evenly distributed among all sensor nodes [4].
- NBS supports for WSNs redundant sensor nodes.
- NBS forms multiple overlapped backbones at some time period which work alternatively to prolong the network lifetime.
- In NBS, traffic is only forwarded by backbone sensor nodes, and the rest of the sensor nodes turn off their radios to save energy.

## 5 PROBLEM DEFINITION

The connectivity between the sensors nodes in the network are based on the energy. The amount of energy consumed by any sensor node in the network at the end of the lifetime does not exceed its initial value. The lifetime of a schedule is the lifetime of the network using this schedule to turn on and off the radio of the sensor nodes. That is the MLBS problem is to find that schedule that achieves the maximal network lifetime and hence the backbones can be overlapped.

## 6 FORMATION OF SENSOR NODES

Sensor nodes are randomly placed in the field and are immobile thereafter. A battery is the sole energy source of the sensor nodes. All sensor nodes have an identical communication range. The power consumption of a sensor node is comprised of three parts: sensing, computing, and radio. For a typical sensor node, the radio is the most power-consuming part and may even dominate the energy

consumption. Therefore, it only considers the scheduling of the radio. Sensor nodes are duty-cycled and have the same working cycle. Sensor nodes follow a periodic active/sleep cycle, and synchronize to reduce the transmission delay called Duty-Cycle. At the beginning of each round, a backbone is selected to work in duty-cycling. Nodes that are not in the backbone will turn off their radios. The lifetime of a sensor node is the time span from when it starts working to when its energy is depleted. The lifetime of a network is the minimum lifetime of the entire sensors in the network[5]. Because backbones rotate after each round, the lifetime is counted in rounds. All sensor nodes have the same transmission range. The number of sensor nodes is varied to model different network densities and scales. Two configurations are used: identical initial energy and imbalanced initial energy. Sensor nodes are deployed in a 500 X 500 area. The transmission range is fixed to 100 so that all of the networks generated are fully connected. Since the area of the network is fixed, these settings vary the density of the sensor nodes. All sensor nodes have 100 units of initial energy.

## 7 CENTRALIZED SYSTEM

### 7.1 Approximation Algorithms:

First centralized approximation algorithm is based on a new concept called Schedule Transition Graph (STG). A STG is used to model a schedule in a WSN. The Figure 5.1 horizontal axis represents the time scale, counted in rounds. In each round, possible states are listed vertically, which are represented by ellipses. The number of possible states for each round is equal to the number of backbones. The state and the backbone have a one-to-one mapping. An initial state is placed at round 0 and is connected with all states in the first round to represent a starting point. Undirected transition edges connect states in one round to those in the next round. No backward edges are allowed. Each edge represents the time elapse of 1 round. Since energy is used in each round, each edge also represents the consumption of energy. Backbone consume a fixed amount of energy in each round, all edges represent the same amount of energy consumption[5].

The residual energy of all nodes is obtained by subtracting this value from the starting state of each transition edge. No transition is allowed if the energy of any sensor node of a state is depleted.

The length of the horizontal direction of an STG is the maximum number of rounds that the network can run without depleting the energy of any sensor node, which is denoted as C. Given a network with a fixed topology and a finite amount of initial energy in each sensor node, the maximum round number is derived by dividing the sum of the initial energy of all nodes by the minimum amount of energy consumed in each round. It is necessary to enumerate all possible backbones of the network in order to find the optimal solution. Instead, a polynomial number of backbones are constructed in our algorithm[1][2]. In order to obtain better results, more backbones should be constructed[3].

Each sensor node consumes a fixed amount of energy "in each round when working as a backbone node. A virtual node that corresponds to a sensor node as a node that contains energy". The original node is called the ancestor. The virtual nodes of the same ancestor form a virtual group. Virtual nodes in the same virtual group are indexed. Two virtual groups are neighbors if their ancestors are neighbors in the original graph. The virtual nodes that have the same indexes are connected. A virtual node is isolated if it does not connect with any virtual node of other virtual groups round. Virtual Scheduling Graph (VSG) preserves the connectivity of the original graph. The reason to use this approach is that it will be biased toward nodes with more energy. Nodes with more energy will cause "isolated" virtual nodes. In this way, the CDS construction algorithm is forced to pick virtual nodes with more energy.

## 8 DISTRIBUTED SYSTEM:

A distributed implementation of VBS called Iterative Local Replacement (ILR). ILR lets each backbone sensor node find replacement nodes to form a new CDS that preserves the connectivity of the network. Each sensor node of the backbone sensor only needs local information to do this. Execution time, in each round, a backbone node that decides to switch its status collects or updates the information of its h-hop neighbors to find replacement nodes. The time used to perform these operations should be minimized. Quality of the results, generally, more information yields better results, which achieves a longer lifetime. However, it increases the message overhead and prolongs the execution time.

First, topology information does not need updating because sensor nodes are static. Second, energy consumption can be estimate according to the working statuses of sensor nodes. These two techniques avoid the costly message exchange of the ILR. If all backbone nodes start the replacement simultaneously, many sensor nodes may contend the shared channel, which causes packet collision and loss. There is a Pswitch function a switching probability, to each backbone node to stop the replacement when the residual energy (Er) is low. Replacement of nodes becomes too expensive when there is not much energy left in the sensor nodes and the value will be zero. When the residual energy is greater than the threshold (ET) the algorithm starts replacing the nodes and the value will be one. At the end of each round, backbone nodes switch statuses according to this probability [7].

$$P_{switch} = \begin{cases} 1 - \frac{E_r}{E_T}, & \text{if } \overline{E_r} \geq E_T \wedge E_r \leq \overline{E_r}, \\ 0, & \text{if } \overline{E_r} < E_T \vee E_r > \overline{E_r}. \end{cases}$$

The replacement becomes too expensive when there is not much energy left in the sensor nodes. The line labeled "original" represents the results of no sleep-scheduling. Rules 1 and 2 and Rule K are used to construct backbones. The STG-based algorithm produces the best results. The inferior performance of ILR is because it uses only local information.



## 9 SYSTEM IMPLEMENTATION

### 9.1 Enhanced Schedule Transition Graph (ESTG)

The approximation algorithm is based on dynamic programming. Its pseudo code is listed in Algorithm 1. The search starts from the initial state. After a backbone transition, the state's energy levels are computed from those of the starting state of the transition. Each state keeps the larger energy levels. A path terminates when its associated energy level is zero. When all paths terminate, the longest path is found. In searching for the longest path in the ESTG[7], we need to record the energy levels of each state. The energy levels produced by this function are invalid because strictly lower energy levels cannot produce a longer lifetime [1]. These energy levels are discarded. In order to reduce the complexity, we select the energy level that has the largest minimum value in the tuple as the associated energy level of each state. If there are still multiple energy levels associated with a state, the one that has the largest summation of all of the values of the energy level is kept.

Algorithm 1. ESTG-based algorithm

```
1: int AT_PRESENT_ROUND = 0;
2: repeat
3: for each state S do
4: Get the linked node energy levels of S;
5: Reduce the resultant energy levels using the low()
Function;
6: Select the energy level with the high low energy value.
7: Set S's energy level to the energy level with the high
summation among the resultant energy levels;
8: end for
9: CUR_ROUND = CUR_ROUND + 1;
10: until all the energy levels of the states in CUR_ROUND
are zero;
11: Return the schedule represented by the path ending in
CUR_ROUND.
```

#### 9.1.1 Time Span of ESTG:

The maximum number of rounds in a STG is derived by dividing the sum of the initial energy of all nodes by the minimum energy consumed in each round. Backbone nodes consume a fixed amount of energy in each round. The backbone has more nodes than MCDS, suppose that the size of the MCDS is  $n$ , then the minimum energy consumption in each round. In order to reduce the complexity the energy level that has the largest minimum value in the tuple as the associated energy level of each state. If there are still multiple energy levels associated with a state, the one that has the largest summation of all of the values of the energy level is kept minimum[6].

#### 9.2 Non Real Scheduling Graph (NRSG)

Second centralized approximation algorithm is based on the concept called Non Real Scheduling Graph. It Uses the Marking Process (MP) for constructing the CDS A Non Real node that corresponds to a sensor node as a node that contains energy. The Non Real nodes of the same ancestor form a virtual group. The non real nodes in the same virtual group are indexed each ancestor is replaced by a group of virtual nodes. The size of the group corresponds to the energy of the ancestor Nodes of two neighboring virtual groups are connected with an increasing index order until one group's virtual nodes are all connected.

Algorithm 2. NRSG-based algorithm

```
1: S = {};
2: Construct the NRSG Gs (V', E') of G (V, E);
3: repeat
4: Apply the marking process on Gs (V', E');
5: Apply Rules 1 and 2 and Rule K on the induced graph;
6: Construct the PMCDS C' from the resultant CDS C;
7: Remove the highest indexed Non Real nodes of the
ancestors whose virtual nodes are in C' from Gs (V', E');
8: Find the corresponding CDS Ci of C' in G;
9: S = S U {(Ci, Ti)};
10: until any ancestor's Non Real nodes are all eliminated
from Gs (V', E');
11: return S.
```

Nodes with more energy will cause "isolated" nodes the ESTG-based scheduling; the sink is added into each backbone. A nice property of NRSG is that nodes with more energy tend to have isolated Not Real nodes. The original node is called the ancestor. The Non Real nodes of the same ancestor form a virtual group. Non Real nodes in the same virtual group are indexed. Two virtual groups are neighbors if their ancestors are neighbors in the original graph. The virtual nodes that have the same indexes are connected. A virtual node is isolated if it does not connect with any virtual node of other virtual groups round. A NRSG preserves the connectivity of the original graph. The reason to use this approach is that it will be biased toward nodes with more energy. Nodes with more energy will cause "isolated" virtual nodes. In this way, the CDS construction algorithm is forced to pick virtual nodes with more energy.

#### 9.3 Repetitive Local Replacement (RLR)

A distributed implementation of NBS called Repetitive Local Replacement (RLR). RLR lets each backbone sensor node find replacement nodes to form a new CDS that preserves the connectivity of the network. Each sensor node of the backbone sensor only needs local information to do this. Execution time, in each round, a backbone node that decides to switch its status collects or updates the information of its h-hop neighbors to find replacement nodes. The time used to perform these operations should be minimized. Quality of the results, generally, more information yields better results, which achieves a longer lifetime. However, it increases the message overhead and prolongs the execution time. These two issues are varied.

### Algorithm 3. Repetitive local replacement

- 1: loop
- 2: At the beginning of each round;
- 3: Sensor node N computes the switching probability  $P_{switch}$  using 2;
- 4: if Decide to switch then
- 5: Collect or update the h-hop information of N;
- 6: Apply the marking process on the subgraph;
- 7: Apply Rules 1 and 2 and Rule K on the induced graph using the residual energy as the priority;
- 8: R = The IDs of sensor nodes have more residual energy and can form a new CDS by replacing N;
- 9: Notify each sensor node  $N \in R$ ;
- 10: end if
- 11: end loop

Various algorithms are available in the literature, and they should be investigated carefully to choose the most appropriate one. First, topology information does not need updating because sensor nodes are static. Second, energy consumption can be estimated according to the working statuses of sensor nodes. These two techniques avoid the costly message exchange of the RLR. If all backbone nodes start the replacement simultaneously, many sensor nodes may contend the shared channel, which causes packet collision and loss. This situation may cause an increased execution time and more energy consumption.

## 10 PERFORMANCE METRICS AND RESULTS

### NETWORK LIFETIME

The result of the network lifetime is achieved by the proposal algorithms. Here identical initial energy and imbalanced initial energy sensor nodes are employed in 500\*500 areas. The transmission range is fixed to 100 so that all of the networks generated are fully connected.

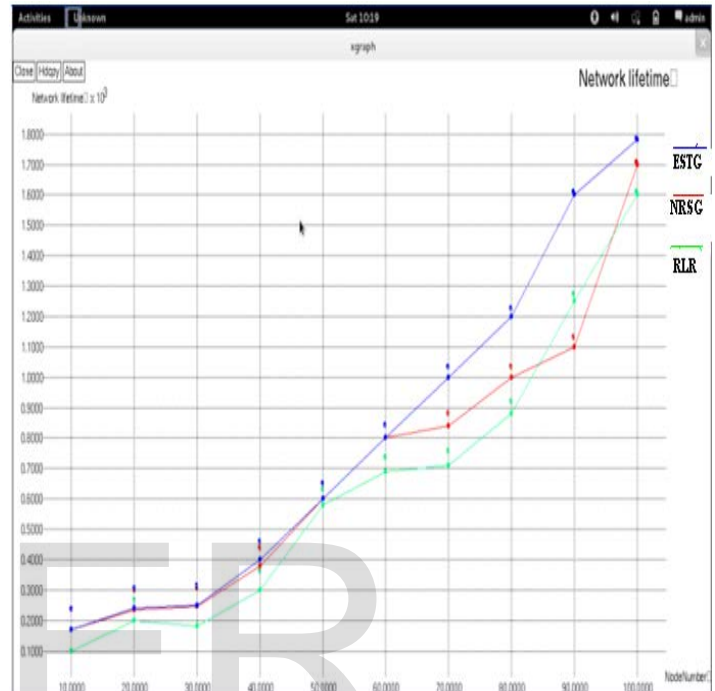
### ENERGY CONSUMPTION

Each sensor node is assigned an initial energy drawn uniformly from [50; 100] because the lifetime is determined by the node with the minimum energy, the achieved lifetime when all nodes work is nearly halved, as shown in the line labeled "original." The lifetimes of all schemes in the assessment decrease drastically. However, proposed schemes still achieve much longer lifetimes.

### END-END DELAY

End-End Delay measures the average time delay that it takes to route a data packet from the source node to the destination.

## New Backbone Scheduling



## 11 CONCLUSION AND FUTURE ENHANCEMENT

WSNs require energy-efficient communication to be able to work for a long period of time without human intervention. MLBS, a schedule in NBS is a set of backbones working sequentially in each round that satisfy connectivity and energy constraints. The lifetime of a schedule is the lifetime of the network. In this paper, two centralized approximation algorithms with different complexities and performances are implemented. Additionally, an approximation algorithm called RLR, an efficient distributed implementation of New Backbone Scheduling (NBS) is designed. VBS formulates the rotation of multiple backbones that makes sure the energy consumption of all sensor nodes is balanced, which fully utilizes the energy and achieves a longer network lifetime.

The paper can be further implemented by using Minimum Energy Scheduling (MES) algorithm which consumes minimum amount of energy in each node of the WSN. The energy efficiency attained by the MES algorithm acquires minimum loss of throughput-optimality. The proposed NBS algorithm significantly reduces the overall energy consumption that uses a

small network area and has a fixed topology design; in case of using MES algorithm in larger network the network lifetime can be maximized. In a large network area random topology design can be implemented.

## REFERENCES

- [1] V. Shnayder, M. Hempstead, B.-r. Chen, G.W. Allen, and M. Welsh, "Simulating the Power Consumption of Large-Scale Sensor Network Applications," Proc. Second Int'l Conf. Embedded Networked Sensor Systems (SenSys '04), pp. 188-200, 2004.
- [2] C. Misra and R. Mandal, "Rotation of CDS via Connected Domatic Partition in Ad Hoc Sensor Networks," IEEE Trans. Mobile Computing, vol. 8, no. 4, pp. 488-499, Apr. 2009.
- [3] W. Ye, J. Heidemann, and D. Estrin, "An Energy-Efficient MAC Protocol for Wireless Sensor Networks," Proc. IEEE INFOCOM, pp. 1567-1576, 2002.
- [4] Q. Cao, T. Abdelzaher, T. He, and J. Stankovic, "Towards Optimal Sleep Scheduling in Sensor Networks for Rare-Event Detection," Proc. ACM Fourth Int'l Symp. Information Processing in Sensor Networks (IPSN '05), pp. 20-27, 2005.
- [5] A. Keshavarzian, H. Lee, and L. Venkatraman, "Wakeup Scheduling in Wireless Sensor Networks," Proc. Seventh ACM Int'l Symp. Mobile Ad Hoc Networking and Computing (MobiHoc '06), pp. 322-333, 2006.
- [6] R. Cohen and B. Kapchits, "An Optimal Wake-Up Scheduling Algorithm for Minimizing Energy Consumption while Limiting Maximum Delay in a Mesh Sensor Network," IEEE/ACM Trans. Networking, vol. 17, no. 2, pp. 570-581, Apr. 2009.
- [7] J.W. Hui and D.E. Culler, "IP is Dead, Long Live IP for Wireless Sensor Networks," Proc. Sixth ACM Conf. Embedded Networked Sensor Systems (SenSys '08), pp. 15-28, 2008.
- [8] L. Doherty, W. Lindsay, and J. Simon, "Channel-Specific Wireless Sensor Network Path Data," Proc. Int'l Conf. Computer Comm. and Networks (ICCCN '07), pp. 89-94, 2007.
- [9] "Wireless Routing Simulation Suite," <http://sourceforge.net/projects/wrssl/>, 2012.
- [10] F. Dai and J. Wu, "On Constructing K-Connected K-Dominating Set in Wireless Ad Hoc and Sensor Networks," J. Parallel Distributed Computing, vol. 66, no. 7, pp. 947-958, 2006.

<http://doi.ieeecomputersociety.org/10.1109/TPDS.2011.305>  
<http://www.ieee.org/telecommunication.com/>  
<http://www.ieee.org/groups/sensornetworks.html>