This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMSCS.2017.2672553, IEEE Transactions on Multi-Scale Computing Systems

Load Balanced Coverage with Graded Node Deployment in Wireless Sensor Networks*

Punyasha Chatterjee, Sasthi C. Ghosh and Nabanita Das, *Senior Member, IEEE*

Abstract—In this paper, to gather streams of data in static wireless sensor networks, a novel graded node deployment strategy is proposed that generates minimum traffic, just sufficient for coverage. Based on this node distribution, a distributed, nearly load-balanced data gathering algorithm is developed to deliver packets to the sink node via minimum-hop paths that also in turn helps to limit the network traffic. An average case probabilistic analysis is done based on perfect matching of random bipartite graphs to establish a theoretical lower bound on the number of nodes to be deployed. Analysis and simulation studies show that the proposed model results huge enhancement in network lifetime that significantly overrides the cost due to over deployment. Hence, this technique offers an excellent cost-effective and energy-efficient solution for node deployment and routing in large wireless sensor networks to operate with prolonged lifetime.

Index Terms—Wireless sensor network (WSN), load balancing, lifetime, coverage, data gathering, graded node distribution.

1 Introduction

Recent technological advances have made feasible the deployment of tiny inexpensive sensor nodes over a region of interest to collect ground data, to process it and to route it to a sink node for aggregation, that as a whole comprises a *Wireless Sensor Network (WSN)*. In fact, *WSN*'s have numerous applications in weather monitoring, disaster management, inventory tracking, smart spaces, precision agriculture, habitat monitoring, target tracking, surveillance and many more. In a multi-hop *Wireless Sensor Network (WSN)* with continuous traffic, each sensor node acts as a *data originator* that introduces a data packet into the network at regular intervals, to send its data to the sink node. At the same time, each node also acts as a *router* to forward others data packets to the sink node via multi-hop paths. In many applications, the sensor nodes are battery-powered, and without any recharging facility. Most of the energy of the sensor nodes is depleted in the process of data communication. When the battery power is exhausted, a node fails to operate and conventionally this ends the *lifetime* of the network which is the time duration of network operation until the first node fails, mainly due to energy shortage. Hence for energy-efficiency, it is a fundamental issue to reduce the total number of packet transmissions in the network. The network load, i.e. the total number of packets to be delivered to the sink node is lower bounded by the coverage constraint. Given that bound, it is obvious that, each packet should follow the path with minimum hop to reduce the total number of packet transmissions. But, if nodes always forward their packets to the sink node via minimum hop paths, nodes nearer to the sink will carry heavier traffic and will deplete their energy faster, creating *energy holes* [15], [16] around the sink. Hence, in a multi-hop *WSN*, it is a challenging issue to exploit the energy of all the nodes uniformly so that the network *lifetime* [9], [10], [18] is maximized.

Data gathering with load balancing in terms of power demand at each individual node may be an efficient approach to enhance the *lifetime* of the network. A plethora of work has been reported recently on this issue. But in most of the earlier works, load balancing has been achieved at the expense of routing packets via longer distances or more hops which is not efficient from energy point of view [5], [4], [6], [3], [8]. Here, nodes take decision to forward packets either directly or via multi-hop paths in order to guarantee energy balance. Moreover, even for a very simple network, computing the most balanced data gathering routes is an *NP-hard* problem [11], [12]. For some simple regular sensor network topologies, in [7], the upper and lower bounds on functional lifetime are derived. In [2], for *2D* grid networks, authors propose a heuristic strategy with *transmission power control* that ensures maximum network *lifetime* by balancing the traffic load as equally as possible. In all these approaches load-balancing is targeted at the cost of increased energy consumption in routing via non-optimal paths.

With routing via shortest paths, in [1], authors formulated the *δ-Bounded Load Balanced Tree Problem (B-LBTP)*, to keep the nodes load-balanced. By their proposed algorithms, the value of the load balancing factor δ varies

**A preliminary version of this paper appeared in Proceedings of the 1st International Conference on Applications and Innovations in Mobile Computing, AIMoC-2014, Kolkata, India, pp. 126 - 132, February 2014, under the title "Coverage Constrained Non-Uniform Node Deployment in Wireless Sensor Networks for Load Balancing".*

- *Punyasha Chatterjee is with School of Mobile Computing and Communication, Jadavpur University, Kolkata, India. E-mail: punyasha.chatterjee@gmail.com*
- *Sasthi C. Ghosh and Nabanita Das are with Advanced Computing and Microelectronics Unit, Indian Statistical Institute, Kolkata, India. E-mail: {sasthi, ndas}@isical.ac.in*

between 3-14, which is not satisfactory. In some works, load balancing solutions are proposed using *multiple data sinks* [13], *or relay nodes* [9]. However, general rules for the exact distribution of the sinks, or relays are yet to come up. In some applications with high data correlation, routing with data fusion [10], [14], [19] has emerged as a useful paradigm for load balancing though it cannot resolve the *energy hole* problem fully. At the same time, it requires complex computation to prevent loss of vital information.

Hence, graded or non-uniform node distribution emerges as a better option for load-balancing [17], [27], [24], [25], [26], [28], [29], where more nodes are deployed nearer the sink to share the increased load instead of compromising energy-efficiency in routing via longer paths.

In most of the works, on non-uniform node distribution, in spite of the fact that the region is over deployed, it was simply assumed that each node generates data packets at regular intervals. But surely it results large number of redundant data packets in the network, depleting the energy of the nodes unnecessarily. However, to satisfy the coverage and connectivity issues, a sufficient number of nodes should always remain active to sense and generate data.

A lot of research has been reported recently to find energy-efficient solutions to satisfy both coverage and connectivity issues [37], [38], [43], [39], [40]. It is to be noted that coverage maximization using minimum number of nodes is important to conserve energy by limiting the network load, i.e., the number of packets to be delivered to the sink. But, unless it is combined with energy-efficient routing, *lifetime* can not be maximized.

Taking into consideration all these issues, this paper proposes a novel graded node distribution strategy to keep the network traffic just sufficient to cover the area, and at the same time to route packets via minimum hop paths with load balance. It is to be noted that routing packets via minimum-hop paths, in fact, keeps the network traffic, i.e., total number of packet transmissions in the network minimum, and at the same time, load balance distributes this network traffic uniformly across nodes to prolong network lifetime. Hence, the proposed model achieves ultimate energy efficiency at the expense of over deployment of nodes but at the same time avoiding redundant data in order to limit network traffic.

For the proposed node distribution strategy over a circular area, an elegant probabilistic average case analysis is done to establish a lower bound on the number of nodes to be deployed to satisfy both coverage and load balance with routing via minimum hop paths. From graph-theoretic point of view, we formulate the problem of selecting nearly load-balanced data-routing paths via minimum hops as the *bipartite perfect matching problem*. Finally, we present a distributed heuristic algorithm that computes the data gathering tree based on the *hop-count* information of neighborhood only. Extensive simulation studies show that the proposed strategy ensures coverage with load balance, routing packets via minimum hop paths that results significant enhancement in network *lifetime*. Also, it shows that the huge enhancement in *lifetime* overrides the increase in cost due to over deployment. For an example, with 2.4 times increase in the number of nodes compared to the uniform node distribution-based strategy [4], the *lifetime* is enhanced by 19 times.

The rest of the paper is organized as follows: Section 2 presents some preliminaries, Section 3 explains the coverage constrained node distribution strategy. The load-balanced data gathering procedure is proposed in Section 4. Section 5 includes the probabilistic coverage analysis and Section 6 describes the proposed distributed algorithm. Section 7 presents the simulation results and Section 8 concludes the paper.

2 Preliminaries

We consider a WSN consisting of a set of n homogeneous static sensor nodes with fixed transmission range T and sensing range S, distributed over a *2-D* region under consideration. Though in some works, authors assume that nodes have transmission power control ability depending on the distance of the receiver, but estimating the distance and hence to control the power, increases the node complexity and is not practical.

Definition 1. *Given a WSN with a set of n sensor nodes $V = \{1, 2, .., n\}$ and a sink node $(n+1)$, distributed over a 2-D region, the network topology is represented by a graph $G(V, E)$, where $E = \{(i, j) | i, j \in V\}$, if the distance between nodes i and j, $d(i, j) \leq T$, the transmission range. $G(V, E)$ is defined as the topology graph of the given WSN.*

Remark 1. *The links in $G(V, E)$ are assumed to be bidirectional so that $G(V, E)$ is an undirected graph.*

Definition 2. *In $G(V, E)$ the number of hops along the shortest path from any node i to the sink node $(n + 1)$ is termed as the hop-count of node i.*

Definition 3. *The set of nodes $X_i \subseteq V$ is called the 1-hop neighbors of node i, if in $G(V, E)$, each node $j \in X_i$ is adjacent to node i.*

Definition 4. *The operational lifetime of the sensor network is defined to be the maximum time duration during which all nodes in the network are alive, i.e. the time until the first node dies [9].*

2.1 Energy Model

Each sensor node starts with an initial energy E that is depleted at each time the node transmits or receives. Following the same model as proposed in earlier works [36], [1], [25], [5], [7], [2], [17], we assume that sensing and computing require negligible amount of energy, and can be ignored. For example, Mica2 motes [41] in a cluster use only 6% of the total energy for sensing

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMSCS.2017.2672553, IEEE Transactions on Multi-Scale Computing Systems

and computing takes even less energy depending on the computation load. Also SENSEnuts [44] and DZ50 [45] are few recent motes that consume significantly less power in sensing compared to that used for communication. Moreover, these two parts of energy consumption will remain invariant for a specific problem solution. So, for load balance across nodes, we take into account the energy consumption due to data communication only.

We assume the simple *First Order Radio Model* [19], where the energy consumed by a sensor node in receiving an l-bit packet is

$$R_x = \epsilon_{elec}.l,$$

and the energy consumed to transmit an l-bit packet is

$$T_x = R_x + \epsilon_{amp}.l.T^{\alpha}.$$

Here ϵ_{elec} is the energy required by the transmitter or receiver circuit and ϵ_{amp} is that for the transmitter amplifier to transmit a single bit and T is the transmission radius of a node and α is the path-loss component, $2 \leq \alpha \leq 6$.

In our model, we assume that all sensor nodes are homogeneous with a fixed transmission radius T and also the packet length l is constant. Hence, per packet transmission, each node consumes the same amount of energy.

This energy model clearly shows that for sensor nodes, transmission is the most expensive operation in terms of energy. Hence to conserve energy the number of packet transmissions per node is to be optimized.

2.2 Data Originators and Routers

In our proposed scheme, a node may act either as a data originator, or as a router, as defined below.

Definition 5. *A data originator node senses data and forwards it towards the sink node, at a fixed time interval (T_{round}), termed here as a round. Whereas a router node only forwards others' packets when it receives one.*

Definition 6. *A cycle consists of m number of rounds i.e. $T_{cycle} = m.T_{round}$, and $T_{cycle} >> T_{round}$, where m is an integer.*

It is assumed that at the beginning of each cycle, a set of *data originators* is selected afresh that covers the given area. The rest of the nodes are free to act as *routers*, if required. Hence, in successive cycles, a node may change its role. This, in turn, helps to achieve better load balance as has been explained later.

2.3 Traffic Model and Load

This paper adopts a slightly modified version of the *continuous traffic model*, where instead of each node, only the *data originator* nodes generate a single data packet in a round. Under this traffic model, during the data gathering process, the minimum load (in terms of energy) of a *data originator* node is only T_x, if it just transmits its own packet in each round. On the contrary, the minimum possible load on a *router* node is $(R_x + T_x)$ when it receives exactly one packet per round and forwards it to another router towards the sink.

If this strategy can be followed for data gathering, the load will be minimum and nearly balanced on all nodes, resulting a nearly balanced energy depletion across the nodes. It also clarifies that role change in successive *cycles* will improve the load balance further.

2.4 Minimum-Hop Path

In our proposed strategy of routing, not only the load on each node is kept minimum, but also packets are always forwarded via minimum-hop path. So far, routing techniques for load balance, mostly assume packet forwarding via non-optimal paths, either by longer distances, or by hop-stretches. Here, to achieve ultimate energy-efficiency, load balance is maintained by routing packets via minimum-hop paths. It is important to note that, minimum-hop paths not only helps to minimize delay, it also minimizes the number of packet relays, hence the total number of communications to deliver a packet from the *data originator* to the sink. Essentially, this keeps the total volume of network traffic minimum which is finally distributed uniformly across the nodes to result prolonged network *lifetime*.

3 Coverage Constrained Node Distribution (CCND)

Firstly, let us consider a circular deployment region with the sink node at the center, the area being divided into say, P number of coronas $C_1, C_2, \ldots, C_{P-1}, C_P$, each with width $R \leq T$, as shown in Fig. 1. The corresponding *topology graph* is constructed assuming homogeneous nodes with fixed transmission range T, as shown in Fig. 2. Nodes are deployed over the area such that the *data originator* nodes, exactly cover the area, and along with the *router* nodes, create the load balanced data gathering paths.

Let N_P^G denote the number of *data originator* nodes in the outer-most corona C_P, which is sufficient to cover the corona C_P. The area covered by a sensor node is modeled as a disc of radius S, centered at the node itself [22]. In each round, each *data originator* node in C_P senses the environment and generates a data packet that is forwarded to a unique *router* node in the next adjacent corona C_{P-1} towards the sink, to follow minimum hop path.

Now, the area of the corona C_P is:

$$\pi(PR)^2 - \pi((P-1)R)^2 = \pi R^2(2P-1).$$

Hence, N_P^G nodes are deployed in the area $\pi R^2(2P-1)$. The area of corona C_{P-1} is:

$$\pi((P-1)R)^2 - \pi((P-2)R)^2 = \pi R^2(2P-3).$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMSCS.2017.2672553, IEEE Transactions on Multi-Scale Computing Systems

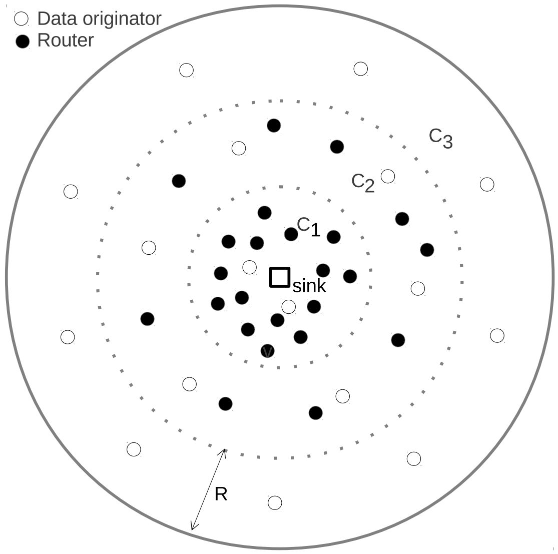

Fig. 1: Sink-centric coronas with non-uniform node distribution (*P=3*)

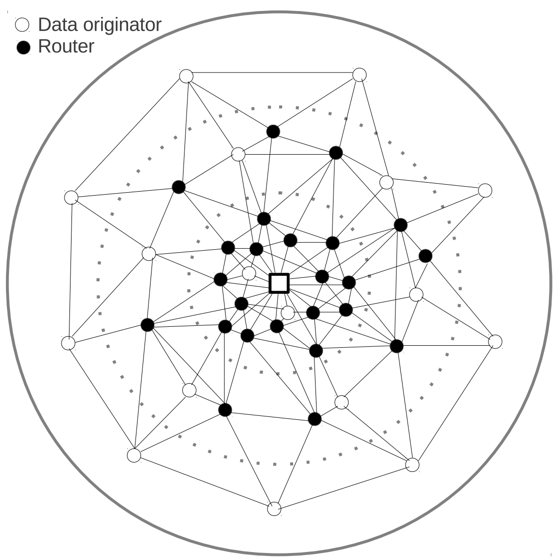

Fig. 2: The *topology graph G(V, E)* for the node deployment of Fig. 1

So, the proportionate number of *data originators* required to cover corona C_{P-1} is:

$$N^G_{P-1} = N^G_P \frac{(2P-3)}{(2P-1)}. \quad (1)$$

In fact, N^G_{P-1} is less than N^G_P as $\frac{2P-3}{2P-1} \leq 1$ for any $P \geq 2$. Asymptotically, $\frac{2P-3}{2P-1}$ tends to 1.

Now, in corona C_{P-1}, N^G_{P-1} *data originator* nodes will be deployed to cover the corona C_{P-1}. In addition, to forward N^G_P packets, generated in C_P, N^G_P *routers* are distributed in C_{P-1} such that each *router* receives just a single packet from a node in C_P and forwards it to a unique *router* in C_{P-2} to keep the load of each individual *router* node minimum, i.e., $(R_x + T_x)$ only. Hence, total number of nodes in C_{P-1} including both *data originator* nodes and *router* nodes is:

$$N_{P-1} = N^G_{P-1} + N^G_P.$$

For any corona C_K, $1 \leq K \leq P$, the total number of nodes including N^G_K *data originator* nodes to cover C_K and the *router* nodes to forward just a single packet each to keep the load minimum is given by:

$$N_K = \sum_{i=K}^{P} N^G_i = \frac{N^G_P}{2P-1}(P^2 - (K-1)^2). \quad (2)$$

The total number of *data originator* nodes across all the coronas which just transmits one packet per round generated by itself is:

$$\mathcal{N}_G = \sum_{i=1}^{P} N^G_i = N_P \frac{P^2}{(2P-1)}, P \geq 1, \quad (3)$$

where $N_P = N^G_P$ is the number of nodes in the outermost corona C_P.

Total number of *router* nodes forwarding just one packet per round is:

$$\mathcal{N}_R = N_P \frac{P(P-1)(4P+1)}{6(2P-1)}. \quad (4)$$

Total number of nodes deployed across all the coronas is:

$$N = N_P \frac{P(P+1)(4P-1)}{6(2P-1)}. \quad (5)$$

Fig. 3 shows the variation of the number of *router* and the *data originator* nodes varying the number of coronas in a network. It is clear that as the number of coronas increases, the percentage of *router* nodes grows rapidly and the percentage of *data originator* nodes becomes insignificant. Less number of *data originator* nodes restricts the number of packets generated in the network and hence enhances the network *lifetime* significantly. Also, it is evident that networks with more coronas will be better load-balanced.

4 Load Balanced Data Gathering

In this section, we propose the node distribution strategy *CCND* by considering both deterministic and random placement of nodes.

4.1 Deterministic Node Distribution

As described in Section 3, if N_K nodes are placed deterministically in corona C_K as given by equation (2), $1 \leq K \leq P$, such that N^G_K number of *data originator* nodes cover corona C_K, and each node in corona C_K can find a unique *router* node in corona C_{K-1}, $2 \leq K \leq P$ to forward its packet, the load balanced data gathering problem becomes trivial. It is obvious that by this strategy the energy consumption of each *router* node is $(T_x + R_x)$ and that of each *data originator* is only T_x and packets always follow the minimum hop paths to reach the sink without any hop-stretch as shown in Fig. 4.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMSCS.2017.2672553, IEEE Transactions on Multi-Scale Computing Systems

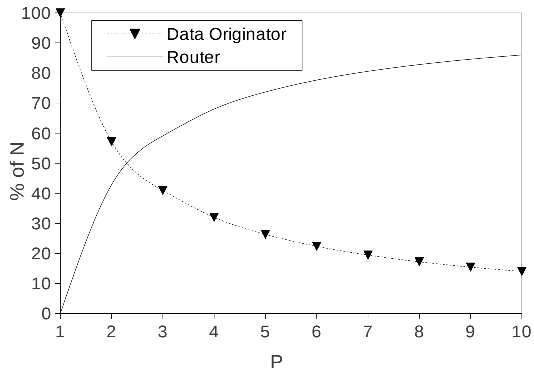

Fig. 3: Distribution of different types of nodes with P

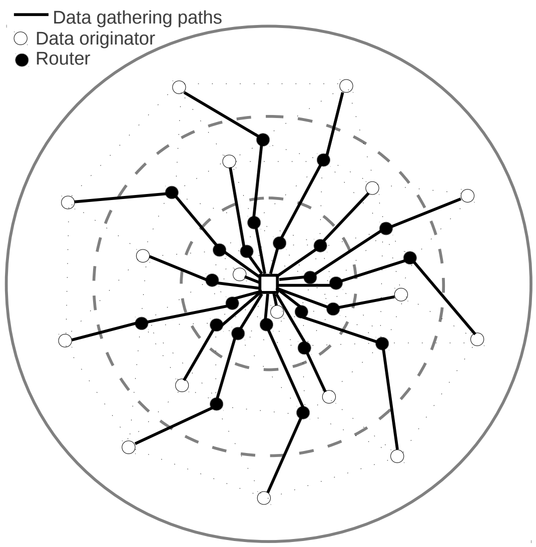

Fig. 4: A nearly load-balanced routing tree for *G(V,E)* (Fig. 2) in ideal case

It is to be noted that in large networks, since the total number of *data originator* nodes is much less than the total number of *router* nodes, as it is evident from Fig. 3, the network is nearly load-balanced. Moreover, by our proposed strategy, discussed in the following subsection, a node will change its role in successive cycles, i.e., a node acting as a *data originator* in one cycle, may operate as a *router* node in the next cycle. Hence the load balancing is improved further resulting enhanced *lifetime*.

However, in case of random node distribution in each corona, it is not sure whether such a nearly load-balanced data gathering tree really exists or not.

4.2 Random Node Distribution

In this case, it is assumed that in each corona C_K, $1 \leq K \leq P$, N_K number of nodes are deployed following a uniform random distribution.

4.2.1 Selection of Data Originators

Since in WSN, the area is generally over deployed, lot of research has been reported to select a subset of nodes sufficient to cover the area. In this paper, we assume that, following some coverage algorithm [21], [43], K disjoint connected subsets of nodes are selected such that each subset ensures coverage. For example, in [21], a number of leader nodes are selected randomly. Each leader node initiates to grow a cover around it. At each iteration, all the nodes of a cover select a *1-hop neighbor* that maximizes the coverage and forwards the information to the leader. Among all these nodes leader selects the one resulting maximum coverage and broadcasts it. Each leader terminates when its partition results the required coverage, or no connected neighbor of the partition is left to be included. Finally successful leaders forward their partitions to the sink.

The selected subsets represent the set of *data originators*. At each cycle, one subset will be activated in a round robin schedule. Nodes other than the *data originators* may act as *router* nodes, whenever required. Once, the set of *data originator nodes* is identified, each node knows its role in that cycle, and they generate the nearly load-balanced data gathering tree for that cycle as has been described below.

4.2.2 Selection of Routers

From the outermost corona C_P, each *data originator* will select a unique *router* node in corona C_{P-1}. Next each node either a *data originator* or a *router* in corona C_K will select a unique router in corona C_{K-1}, $2 \leq K \leq P$, terminating the process at the sink node. The problem can be formulated from the graph theoretic point of view as the *maximum matching problem in Bipartite graphs*.

For deterministic node deployment, it is a case of *perfect matching* where for every node in C_K, there exists a unique *router* node in C_{K-1}. Here it is guaranteed that each data packet generated by a *data originator* will successfully reach the *sink* via unique *router* nodes in successive coronas keeping the network nearly load-balanced.

For random non-uniform node distribution, it is not guaranteed that perfect matching will always exist. By the maximum matching algorithm, if a node in a corona C_K fails to find a match in corona C_{K-1}, its packet is to be dropped that may cause some region of the area to remain uncovered.

The next section presents a probabilistic coverage analysis in terms of packet dropping for the proposed coverage constrained non-uniform node distribution.

5 Probabilistic Coverage Analysis

For the probabilistic coverage analysis with random *CCND*, to simplify the mathematical expressions, the corona width is assumed to be $R = \frac{2}{3}T$, where T is the transmission range. However, it can be generalized easily for any $R \leq T$. Also, it is assumed that both *data originators* and *routers* follow random uniform distribution within each corona.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMSCS.2017.2672553, IEEE Transactions on Multi-Scale Computing Systems

5.1 Packet Dropping Probability

Given the non-uniform node distribution strategy following *CCND*, to guarantee a very low probability of packet dropping to ensure coverage, bipartite graphs are constructed with N_K nodes of corona C_K in one partition and the N_K *router* nodes of corona C_{K-1} in another, for $2 \leq K \leq P$. Given a high probability for the existence of perfect matching for each successive corona pairs, a lower bound is established for N_P, and hence for the total number of nodes. Here, we apply a well-known result of Erdos et al [23] for perfect matching in random bipartite graphs as given below:

Result 1. *For a random bipartite graph $B_{n,m}$, with $n+n$ vertices and $m = n(ln\ n + c_n)$ random edges,*
$$\lim_{n\to\infty} Pr(B_{n,m}\ has\ a\ perfect\ matching) = \begin{cases} 0, & if\ c_n \to -\infty, \\ e^{-2e^{-c}}, & if\ c_n \to c, \\ 1, & if\ c_n \to \infty. \end{cases}$$

Here $c_n = c + \frac{1}{n}o(n)$ and c is an arbitrary real constant.

Given a random uniform distribution of nodes following *CCND* over a *2-D* region divided in coronas as has been explained in Section 3, it is evident that the bipartite graphs between two successive coronas are basically subgraphs of *Random Geometric Graphs* following a *Unit Disc Graph* model, assuming the transmission range to be unity. Hence these bipartite graphs may be considered as a subset of random bipartite graphs, and the result of Erdos can be applied to this case. However, this is the most generalized version. The lower bound can be improved further in case a better graph model can be identified for its representation.

5.2 Lower Bound on Number of Nodes

Let us consider a WSN with P number of coronas. At the outer-most corona C_P, there are N_P number of nodes, all are *data originator* nodes which is sufficient to cover the corona. According to *CCND*, in corona C_{P-1}, there are N_P *router* nodes and proportionate number of data originator nodes to cover C_{P-1}.

Now, at every round, any node i in corona C_P, generates a single packet and forwards it to one of its *1-hop neighbor router*s in corona C_{P-1}. For average case analysis, let us consider a node i at distance $(P-0.5)R$ from the sink node. The *1-hop neighbor*s of node i lie within the area A_{P-1} in corona C_{P-1}, as shown in Fig. 5, for $P=4$.

Now, given two circles of radii r and r' respectively with centers at a distance d, from simple geometry, the area of intersection A is given by:

$$A = r'^2 cos^{-1}\frac{d^2+r'^2-r^2}{2dr'} + r^2 cos^{-1}\frac{d^2+r^2-r'^2}{2dr} - \alpha \quad (6)$$

where

$$\alpha = \frac{1}{2}\sqrt{(-d+r'+r)(d+r'-r)(d-r'+r)(d+r'+r)}. \quad (7)$$

Hence,

$$A_{P-1} = D_{P-1}R^2, \quad (8)$$

where

$$D_{P-1} = \frac{9}{4}cos^{-1}\theta_p + (P-1)^2 cos^{-1}\theta'_p - \sqrt{2P^2-3P}. \quad (9)$$

Here

$$\theta_p = \frac{2P+3}{6P-3} \quad (10)$$

and

$$\theta'_p = \frac{2P^2-3P-1}{2P^2-3P+1}. \quad (11)$$

With uniform random distribution, the number of router nodes within the area A_{P-1} is,

$$\frac{D_{P-1}N_P}{(2P-3)\pi}.$$

Similarly, any node j of corona C_{P-1}, can receive packets from its *1-hop neighbor*s in corona C_P. All the *1-hop neighbor*s of node j can lie within the area B_P of corona C_P, as shown in Fig. 5 for $P=4$. From equation (6), $B_P = E_{P-1}R^2$, where

$$E_{P-1} = \frac{9\pi}{4} - (\frac{9}{4})cos^{-1}\beta_p - (P-1)^2 cos^{-1}\beta'_p + \gamma. \quad (12)$$

Here

$$\beta_p = \frac{7-2P}{6P-9}, \quad (13)$$

$$\beta'_p = \frac{2P^2-5P+1}{2P^2-5P+3} \quad (14)$$

and

$$\gamma = R^2\sqrt{2P^2-5P+2}. \quad (15)$$

Hence, total number of nodes within area B_P is,

$$\frac{E_{P-1}N_P}{(2P-1)\pi}.$$

Therefore, number of edges between the *data originator* nodes in corona C_P and the *router* nodes in corona C_{P-1} is given by :

$$m_P = (\frac{D_{P-1}}{(2P-3)\pi} + \frac{E_{P-1}}{(2P-1)\pi})\frac{{N_P}^2}{2} = J_{P-1}\frac{{N_P}^2}{2}, \quad (16)$$

where J_{P-1} is a constant, for a given P.

Now, with N_P nodes in corona C_P and the same number of *router* nodes in corona C_{P-1}, a bipartite graph B_{N_P,m_P} is generated, which is a subgraph of the original *topology graph G(V,E)*. For random uniform distribution of nodes, since $G(V,E)$ is a *Random Geometric Graph*, it is evident that the bipartite graphs B_{N_K,m_K}, $P \geq K \geq 2$ belong to the set of all possible random graphs. Hence from [23], a lower bound on the probability of finding a *perfect matching* in B_{N_K,m_K} can be found. Here, the objective is to find the value of N_P such that

$$\mathcal{P}_{P-1} = Pr(B_{N_P,m_P}\text{has a perfect matching}) \quad (17)$$

is almost unity.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMSCS.2017.2672553, IEEE Transactions on Multi-Scale Computing Systems

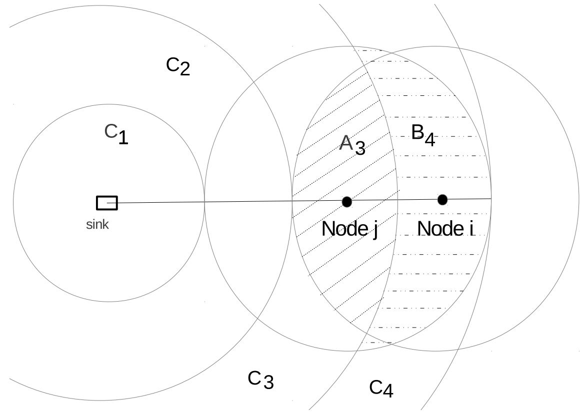

Fig. 5: Nodes in adjacent coronas C_4 and C_3 and the areas within transmission range

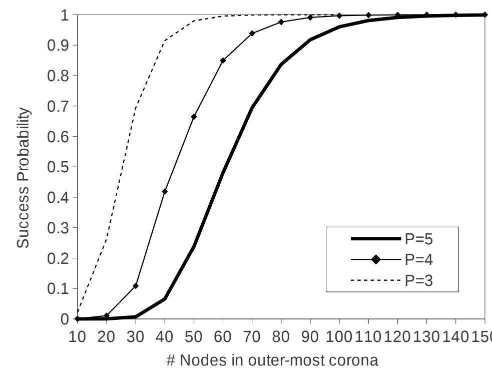

Fig. 6: Probability of successful packet delivery

For an instance, let us assume that $\mathcal{P}_{P-1} = 0.9999$. From the result of [23] by solving the equations

$$m_P = n(\ln\ n + c), \tag{18}$$

and

$$e^{-2e^{-c}} = 0.9999, \tag{19}$$

the number of *data originator* nodes in the outer-most corona C_P can be obtained for any P. Thus, the number of *data originator* nodes required in the outer-most corona C_P to ensure that there exists a perfect matching between the *data originator* nodes in corona C_P and the router nodes in corona C_{P-1} with probability 0.9999 can be found.

Similarly, all nodes in corona C_{P-1} will forward their packets to their *1-hop neighbor router*s in corona C_{P-2} and a bipartite graph $B_{N_{P-1},m_{P-1}}$ is generated, where N_{P-1} is the number of nodes in corona C_{P-1} and m_{P-1} is the number of edges between the nodes in corona C_{P-1} and *router* nodes in corona C_{P-2}. Now, keeping N_P fixed, the bipartite graph $B_{N_{P-1},m_{P-1}}$ will have a perfect matching with probability, say $\mathcal{P}_{P-2}$.

Finally the probability that a packet from the outer-most corona C_P will reach the sink is

$$\mathsf{P}_S^P = \mathcal{P}_{P-1}.\mathcal{P}_{P-2}.....\mathcal{P}_2.\mathcal{P}_1. \tag{20}$$

Therefore, depending on the number of coronas P the probabilities for successful packet forwarding in successive coronas are to be determined to achieve a high probability of P_S^P. However, it should be noted that as the number of nodes increases in coronas towards the sink, $\mathcal{P}_1 \geq \mathcal{P}_2 \geq \mathcal{P}_3 \geq ... \geq \mathcal{P}_{P-1}$. Therefore, it is evident that the probability of successful packet delivery at the sink will be the worst for the packets generated in the outermost corona. Hence it is sufficient to decide on N_P based on the value of P_S^P only.

The variation of successful packet delivery probability P_S^P with the number of nodes at the outer-most corona is depicted numerically in Fig. 6. It shows that with more coronas, to achieve the same probability of successful packet delivery more nodes are to be deployed which

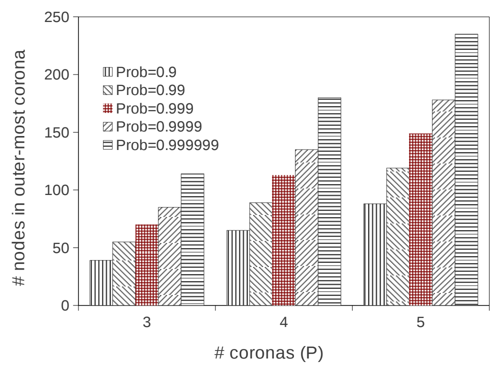

Fig. 7: Variation of N_P for different successful packet delivery probability

is expected. Also, it is true from coverage criterion as well. Fig. 7 shows the variation of N_P for different successful packet delivery probability P_S^P. This lower bound guarantees the load balanced data gathering with a specified packet dropping probability.

5.3 An Example with $P = 4$

Let us consider a *WSN* having four coronas only. At the outer-most corona C_4, there are n number of nodes, all are *data originator* nodes and in corona C_3, there are n *router* nodes. Now, at every round, any node i in corona C_4, will generate a single packet and forward it to a unique *1-hop neighbor router* in corona C_3. All the *router*s in corona C_3 within the area A_3, as shown in Fig. 5 are the possible candidates.

Using simple trigonometrical calculation, we can compute the area A_3 as

$$A_3 = D_3 R^2 \tag{21}$$

where,

$$D_3 = \frac{9}{4}cos^{-1}(\frac{11}{21}) + 9cos^{-1}(\frac{19}{21}) - 2\sqrt{5}. \tag{22}$$

Now, total number of *router* nodes in A_3

$$= D_3 R^2 \frac{n}{5\pi R^2} = \frac{nD_3}{5\pi}.$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMSCS.2017.2672553, IEEE Transactions on Multi-Scale Computing Systems

Again, any node j at corona C_3, can receive packets from its *1-hop neighbor*s in corona C_4. All the *1-hop neighbor*s of node j can reside within the area B_4 of corona C_4 as shown in Fig. 5. We can compute $B_4 = E_3R^2$ where,

$$E_3 = \frac{9\pi}{4} - (\frac{9}{4})cos^{-1}(-\frac{1}{15}) - 9cos^{-1}(\frac{13}{15}) + \sqrt{14}. \quad (23)$$

Hence, total number of nodes in area B_4

$$= E_3R^2\frac{n}{7\pi R^2} = \frac{nE_3}{7\pi}.$$

Therefore, on an average, the number of edges between the *data originator* nodes in corona C_4 and the *router* nodes in corona C_3 is:

$$m_4 = \frac{\frac{nD_3}{5\pi} \times n + \frac{nE_3}{7\pi} \times n}{2} = J_3\frac{n^2}{2} \quad (24)$$

where

$$J_3 = \frac{D_3}{5\pi} + \frac{E_3}{7\pi}. \quad (25)$$

To achieve a successful packet delivery probability, say $\mathcal{P}_3 = 0.99$, applying the result of [23], the value of $N_4 = n$ can be found out.
Given

$$\mathcal{P}_3 = Pr(B_{N_4,m_4}\ has\ a\ perfect\ matching) = 0.99, \quad (26)$$

we solve

$$m_4 = n(\ln\ n + c) \quad (27)$$

and

$$e^{-2e^{-c}} = 0.99, \quad (28)$$

that results

$$n = 89. \quad (29)$$

Thus, the number of *data originator* nodes required in the outer-most corona C_4 is $n = 89$ to ensure that there exists a perfect matching between the *data originator* nodes in corona C_4 and the *router* nodes in corona C_3 with probability 0.99.

Similarly, each node of corona C_3 will forward a single packet to one of its *1-hop neighbor router*s in corona C_2 and a bipartite graph B_{N_3,m_3} is generated, where N_3 is the number of nodes in corona C_3 and m_3 is the number of edges between the *data originator* nodes in corona C_3 and the *router* nodes in corona C_2. Note that the values of N_3 and m_3 are calculated with $n = 89$. Our objective is to find c' such that the equation

$$m_2 = N_3(\ln\ N_3 + c') \quad (30)$$

is satisfied. The probability that the bipartite graph B_{N_3,m_3} will have a perfect matching can be computed as

$$e^{-2e^{-c'}}.$$

For this example this value is found to be $\mathcal{P}_2 = 0.9999$.

Again, between corona C_2 and corona C_1 the perfect matching probability becomes $\mathcal{P}_1 \approx 1$, with $n = 89$.

Hence, the probability that a packet from the outermost corona C_4 will reach the sink is

$$\mathsf{P}_S^4 = \mathcal{P}_3.\mathcal{P}_2.\mathcal{P}_1 = 0.99 * 0.9999 * 1 = 0.9899. \quad (31)$$

Therefore, the probability of packet dropping is

$$1 - \mathsf{P}_S^4 \approx 0.0101 \quad (32)$$

only.

It is obvious that for packets generated at any other corona C_K, $2 \leq K \leq P$, the packet dropping probability will be lower.

5.4 Combined Effect of Coverage and Load Balancing

In Section 3, assuming that N_P^G number of nodes are sufficient to cover the outermost corona, the expression for total number of nodes N is derived (eqn. (5)) that is necessary for load balancing and coverage. In case of deterministic node deployment, this number is sufficient for load balanced routing without packet drop and therefore it ensures coverage as well. However, for random node deployment within each corona, in this section, a lower bound is established on the number of nodes n in the outermost corona C_P to achieve a given probability of successful packet delivery. Hence, finally, with random node deployment in each corona, to satisfy both coverage and load balancing the number of nodes to be deployed in the outermost corona is $N_P = max\{N_P^G, n\}$.

It is to be noted that N_P^G is determined by the parameter S, the sensing range, whereas n is governed by the number of edges existing across the nodes in successive coronas, i.e., by the transmission range T. In literature, many results have been published so far assuming $S = T$. But it is not at all a very realistic assumption. Depending on the type of the application, the sensing range may vary widely, whereas the commercial wireless cards used for communication operates within a specific range of transmission. As for example, the CISCO IEEE 802.11a wireless card when transmits with full power may cover a transmission range of up to $300\ m$ in outdoor. Whereas for sensors like motion detector the sensing range is about 10-15 m only. In most practical cases, $T >> S$, and coverage with random deployment will impose much higher lower bound on the number of nodes in the outermost corona than that imposed by load balancing criterion, i.e., $N_P^G >> n$.

Since, coverage is an essential condition for any WSN, one significant merit of the proposed strategy of load balancing is that in most of the cases, it does not need additional nodes to achieve an acceptable probability of packet delivery.

5.5 For Square Regions and Arbitrary Sink Positions

So far, for the ease of analysis, the deployment region has been assumed to be circular with the sink node at the

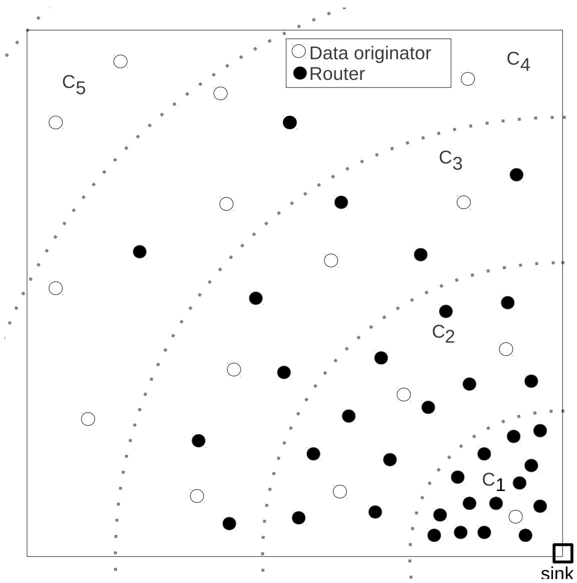

Fig. 8: Non-uniform node distribution in square deployment region with sink at a corner ($P = 5$)

center. However, the analysis can be easily extended to any regular shaped deployment region with an arbitrary location of the sink node. Fig. 8 shows a square deployment region with the sink node at a corner. Knowing the node density sufficient to cover an area, computation of the area of each corona within the deployment region can easily give an estimate on the number of *data originator* nodes. So for deterministic node deployment, it is easy to find out the number of nodes to be placed in each corona. However, for random node deployment, the calculations for average case analysis may be a bit complicated, but tractable.

6 Distributed Algorithm for Data Gathering

In this section, we present a distributed greedy heuristic to construct the load balanced data gathering tree rooted at the sink node.

6.1 Next-Node-Selection Algorithm

Given that N_K nodes are randomly distributed over a corona C_K, $1 \leq K \leq P$, the nodes know their roles in a particular cycle. Each individual node in corona C_i, having a data packet, attempts to select a unique *router* node in corona C_{i-1}, $2 \leq i \leq P$ to create the load-balanced routing tree as explained above. To make the algorithm feasible without the knowledge of the physical positions of the individual nodes in terms of corona, the proposed algorithm is based on the *hop-count* of a node. During the initialization of the network, it is assumed that each sensor node individually discovers its *1-hop neighbors* and also finds the *hop-counts* of itself and its *1-hop neighbors*. With this information, each node with *hop-count h*, maintains an *Option* list for its *router* neighbors with *hop-count* $(h-1)$ that stores the pair {*node-id*, *number of children*}. A node can select its *next node* to forward its data, when all of its neighbors with higher *hop-count* have completed their *next node selection procedure*. In its turn, a node i selects a single node from its *Option* list having minimum number of children. If there is a tie, the node with minimum number of higher *hop-count* neighbors yet to complete their *next node selection* is chosen. Finally, nodes with *hop-count* equal to 1 select the sink as their *next node* when the procedure terminates.

In the next phase i.e. the data gathering phase, nodes follow this data gathering tree generated by the *Next-Node-Selection* procedure and forward their packets to the *next node*s. If more than one node select the same *next-node*, in each round, a single packet is forwarded dropping the others in a round robin fashion. Hence, in each round, each node other than the *data originator* nodes, receives at most one packet and also transmits at most one packet. Thus it ensures a nearly load balanced data gathering at the same time the dropping of extra packets in round robin fashion guarantees that no area is left uncovered, if any, for a long time.

Algorithm 1: Next-Node-Selection Algorithm

```
Input: h_i, S(h_i + 1), S(h_i − 1)
Output: next-node(i)
for each node i do
    Option ← φ;
    Q_i ← 0;
    L(i) ← φ;
    count(i) ← |S(h_i + 1)|;
    if S(h_i + 1) = φ then
        if h_i > 1 then
            broadcasts REQ msg;
        else
            broadcasts SELECTED msg with next-node(i)=sink;

    if receives REQ(j) msg from j ∈ S(h_i + 1) then
        sends REPLY(Q_i, count(i)) msg to j;
    if receives REPLY msg from all j ∈ S(h_i-1) then
        Option ← Option ∪ {j, Q_j};
        Selects next-node(i)=k from Option with minimum (Q_k), in case
        of tie with minimum count(k));
        Broadcasts SELECTED msg with next-node(i);
    if receives SELECTED msg from j ∈ S(h_i + 1) then
        if h_j = (h_i + 1) then
            S(h_i + 1) ← S(h_i + 1) \ j;
            count(i) = count(i) − 1;

        if i= next-node(j) then
            Q_i ← Q_i + 1;
            L(i) ← L(i) ∪ j;

if sink receives SELECTED msg from all of its 1-hop neighbors then
    sink broadcasts TERMINATE msg;
```

The symbols, used in Algorithm 1 are listed in Table 1.

6.2 Correctness Proof and Complexity

Theorem 1. *The Next-Node-Selection procedure always results a tree rooted at the sink node and finally terminates.*

Proof. Given a random node distribution of N_K nodes within corona C_K with a predefined set of *data originator* nodes, $1 \leq K \leq P$, the nodes with highest *hop-count* h_{max} among its neighbors will start the *Next-Node-Selection* procedure. An intermediate node is allowed to

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMSCS.2017.2672553, IEEE Transactions on Multi-Scale Computing Systems

TABLE 1: Symbols used in Algorithm 1

Symbol	Definition
h_i	*hop-count* of node i
next-node(i)	Selected *router* node for node i
$S(h_i + 1)$	Set of neighbors of node i with *hop-count* $(h_i + 1)$
$S(h_i - 1)$	Set of *router* neighbors of node i with *hop-count* $(h_i - 1)$
$count(i)$	Number of neighbors in $S(h_i + 1)$ yet to select *next node*
Q_i	Number of nodes, that forwards their packets to node i
$L(i)$	Queue of nodes, whose packets are to be forwarded by node i
$Option$	$\{(j, Q_j)\}, \ \ \forall j \in S(h_i - 1)$

select a single *next node*, only when all its neighbors with higher *hop-counts* have finished the process. Hence it is evident that there will be no loop since the paths always are directed from a node at hop h to a node at hop $(h-1)$.

In each step, a node with *hop-count* $h > 1$, always selects the *next node* from its neighbors at lower *hop-count* which always exist(s). Finally, when all nodes with *hop-count = 1*, complete the procedure then the sink node terminates the process. Therefore it is evident that the *Next-Node-Selection* procedure always results a tree rooted at the sink node and finally terminates.

Message Complexity : During the procedure, each node, in its turn, broadcasts one request message *(REQ)* to its neighbors for *next node selection*. Also, in response to each request message of its neighbors at higher hop, a node replies *(REPLY)*. Therefore, the message complexity per node is $O(\delta)$, where δ is the maximum node degree.

Computation Complexity : It is evident that the procedure terminates in $O(h_{max})$ rounds, where h_{max} is the maximum *hop-count* in the network. In each round, in the worst case, a node may have $O(\delta)$ computation. Hence the total complexity is $O(h_{max}\delta)$. Here it is assumed that collision free message communication is guaranteed by the *MAC* layer.

7 Performance Evaluation

In this section, for performance evaluation of the proposed distributed algorithm *Next-Node-Selection*, extensive simulation studies have been done by C/C++ programming on randomly generated connected *topology graphs* with random uniform distribution of N_K nodes within each corona C_K, $1 \leq K \leq P$, following *CCND*. To compare the performance, we have chosen the *q-switch* [17] and *LBR* [27] algorithms which follow non-uniform node distribution, and the algorithms proposed in [4] and [43] that consider uniform node distribution. The values of the default simulation parameters are the same as in [17] and presented in Table 2. In all graphs, points represent average values with 95 percent of confidence from 100 different topologies.

Total Number of Nodes Deployed : Fig. 9 shows the comparison of total number of nodes deployed following uniform node density [4], [43], (node density is the same as in the outermost corona C_P), non-uniform distribution of [17] and [27], and the proposed strategy *CCND* (equation (5)) respectively. It is evident that, *CCND* requires more nodes compared to uniform strategies. But in case of non-uniform distribution of [17], total number of nodes increases more rapidly with number of coronas. As an example, for a network with $P = 5$, total number of nodes in *CCND* is reduced by almost 34% of that required by the non-uniform node distribution of [17].

Total Number of Packets Generated : Fig. 10 shows the comparison of total network traffic in terms of number of packets generated in the network. It shows that as P increases, the difference in network traffic for these cases grows rapidly. For example, with $P = 5$, the number of data packets generated in [17] is more than

TABLE 2: Simulation Parameters

Parameter	Value
Transmission radius (T)	12 - 120 (m)
Sensing radius (S)	8 (m)
Corona width (R)	8 - 80 (m)
Sensor field	Circular
Initial energy of a node (E)	0.1 (Joules)
Packet length (l)	400 (bits)
ϵ_{elec}	50 (nJ/bit)
ϵ_{amp}	0.0013 (pJ/bit/m^4)
Path-loss component (α)	4
Total number of coronas (P)	2-7
Number of nodes in outer-most corona (N_P)	60 - 220
Cycle (T_{cycle})	10 rounds

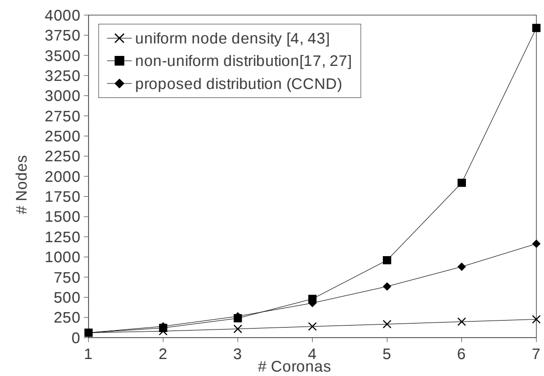

Fig. 9: Comparison of total number of nodes deployed ($N_P = 60$)

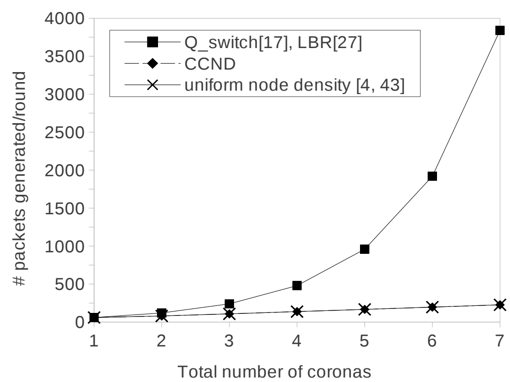

Fig. 10: Comparison of total number of packets generated ($N_P = 60$)

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMSCS.2017.2672553, IEEE Transactions on Multi-Scale Computing Systems

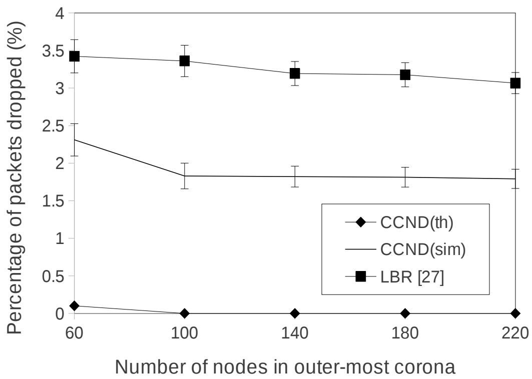

Fig. 13: Comparison of packet drop $(P = 3)$

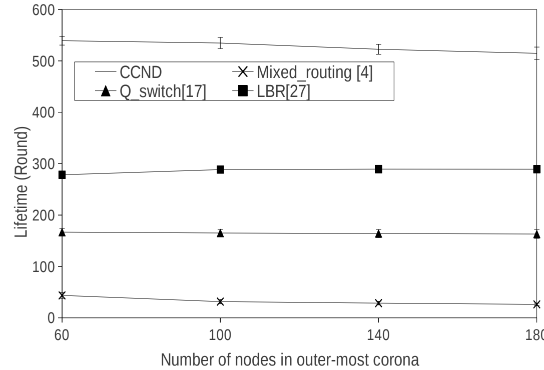

Fig. 14: Comparison of *lifetime (P=3)*

5.7 times of that in *CCND*. It is evident that this will save huge energy and will enhance the network *lifetime* manifold.

Residual Energy Distribution : Considering a square deployment region of *44 m* $\times$ *44 m* and corona width *(R)= 8 m*, with deterministic node distribution of N_K nodes within corona C_K, $1 \leq K \leq P$, Fig. 11 shows the distribution of residual energy of nodes after 1249 rounds of operation with a fixed set of data originators. The two distinct energy levels of Fig. 11 represent the residual energy levels of a *data originator* (higher energy level) and that of a *router* (lower energy level).

For random uniform distribution of nodes within coronas, to identify the set of *data originator*s, the algorithm of [21] is executed to achieve the disjoint sets of data originators to cover the area, one set is made active in a cycle. Fig. 12 shows the residual energy of nodes after the same number of rounds as in Fig. 11 using two partitions for coverage. It achieves better load balance. Also, it is to be noted that most of the nodes have more residual energy in the later case indicating longer *lifetime* of the network. Moreover, due to the presence of multiple partitions, the network remains alive till at least one partition is operational.

Packet dropping percentage : Simulation studies also compare the number of packet drops. In Fig. 13, comparison of packet drop is shown for *CCND* from both simulation (*CCND(sim)*) and theoretical analysis (*CCND(th)*), with [27], varying N_P from 60 to 220. It shows that *CCND* performs better compared to [27]. It is to be noted that as the other strategies [4], [43], [17] do not permit packet drops, they are not considered here.

Remark 2. *Note that CCND(th) is reported based on Result 1, which is an asymptotic result. It is straightforward to estimate the actual probability p_n when n is finite. For that, we generate sufficiently large number of random bipartite graphs $B_{n,m}$ and check how many of them contain a perfect matching by using Ford-Fulkerson algorithm for maximum flow [42]. If t out of T random bipartite graphs contain a perfect matching, $\frac{t}{T}$ gives a fair estimate of p_n. For the range of n considered here $60 \leq n \leq 220$, the difference between the theoretical probability and the probability obtained by simulation varies from 0.004 to 0.007 i.e. of the order of 10^{-3} only.*

Network lifetime : Fig. 14 shows the comparison of *lifetime* of a network resulted by the proposed routing algorithm and the other three corona-based approaches presented in [27], [17] and [4], whereas Fig. 15 shows the comparison with [43], which is not corona-based.

It has been found that the proposed algorithm enhances the *lifetime* almost by 2-3 times over that of [27] and [17]. This improvement, in fact, is achieved by keeping the number of packets generated in the network limited just to satisfy the coverage constraint only. It has also been found that the proposed algorithm enhances the *lifetime* almost by 10 times over that of the routing algorithm [4]. Here, the transmission radius of a node is $120m$, area radius is $240m$ and corona-width (R) is $80m$. In the algorithm of [4] with uniform distribution, the transmission radius (short-range) is taken as $120m$, transmission radius (long-range) will vary depending on the nodes' distance from the sink. The total number of nodes is the same as the total number of data originators in *CCND*.

Fig. 15 shows the comparison of *lifetime* resulted by the proposed routing algorithm, and the algorithm of [43]. Since the node distribution followed in [43] is not corona-based, we compare the lifetime against the total number of nodes instead of the number of nodes in the outer-most corona as has been shown in Fig. 14. It has been found that the proposed algorithm enhances the *lifetime* almost by 2.5 times.

Cost efficiency : Cost efficiency is measured here as the ratio of *lifetime* and the number of nodes deployed. It measures how much improvement in *lifetime* is achieved with respect to cost compared with other approaches. It is defined as :

$$C_e = \frac{L/L_{ref}}{N/N_{ref}},$$

where L is the *lifetime* of *CCND*, L_{ref} and N_{ref} are the *lifetime* and number of nodes respectively in some exist-

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMSCS.2017.2672553, IEEE Transactions on Multi-Scale Computing Systems

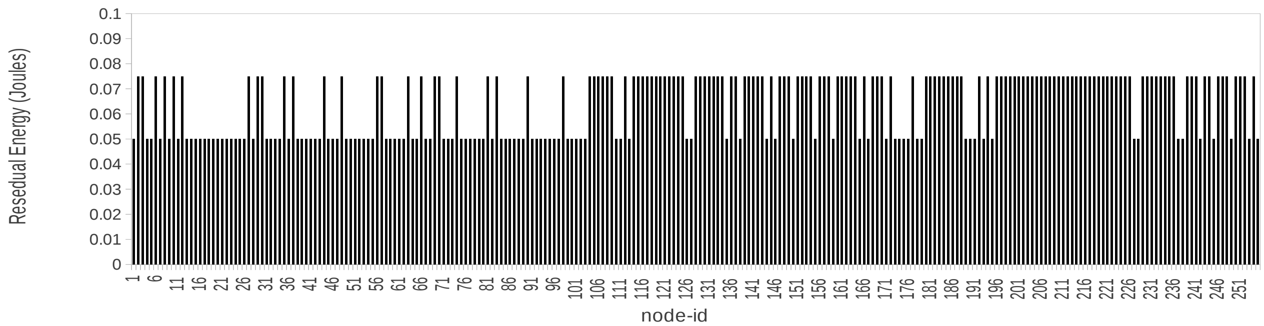

Fig. 11: Residual energy distribution for deterministic node distribution with a fixed set of *data originators* ($N_P = 60$, $P = 3$)

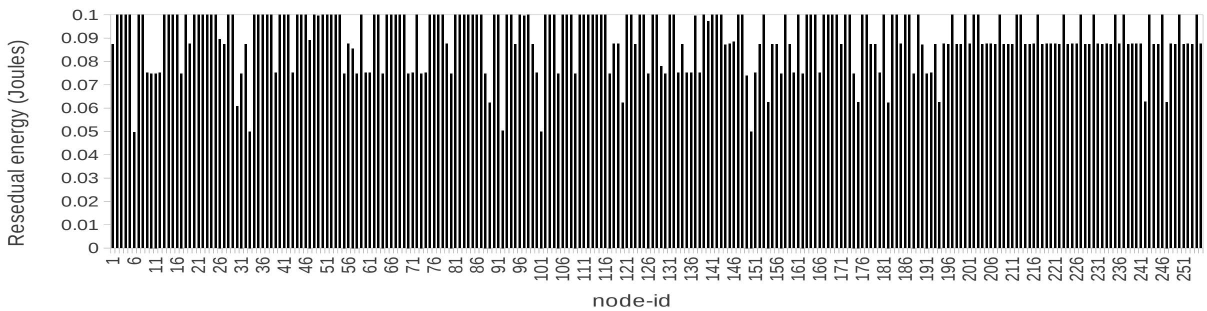

Fig. 12: Residual energy distribution for random node distribution with 2 partitions ($N_P = 60$, $P = 3$)

ing approach, with whom the cost-efficiency of *CCND* is compared. Hence, cost-efficiency $(C_e) > 1$, means that the improvement in *lifetime* in *CCND* overrides the cost of over deployment. In Fig. 16, we have compared the cost-efficiency of CCND with other existing approaches [17], [27] (non-uniform node distributions) and [4] (uniform node distribution) and found that our scheme is significantly better. As for example, with only 2.4 times increase in the number of nodes, compared to [4] (Fig. 14), CCND enhances the *lifetime* almost by 19 times. So, the cost-efficiency C_e is approximately 8. The simulation results show that CCND improves the cost-efficiency by a factor of 1.6 - 8 over other existing approaches. Hence the proposed technique is not only energy-efficient but cost-efficient as well.

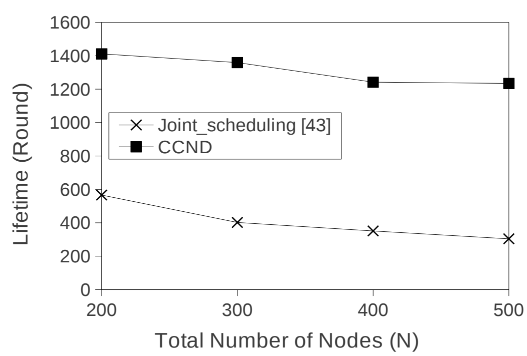

Fig. 15: Comparison of *lifetime (P=3)*

8 Conclusion

In this paper, a novel non-uniform node distribution strategy is proposed where the number of nodes grows in coronas towards the sink to cope up with increased load, in such a way that within each corona, data packets are generated by a limited number of nodes sufficient to cover the area, and some additional nodes simply act as *routers* to make the nodes load-balanced. From probabilistic average case analysis for random node distribution in each corona, lower bounds are established on the number of nodes. Finally, with this non-uniform node distribution, a distributed algorithm is proposed for constructing a load-balanced data gathering tree rooted at the sink node to forward packets via minimum hops. Analysis and simulation studies show that compared to the earlier works with non-uniform node distribution, the proposed scheme performs significantly better in terms of total number of nodes, total network traffic, load on individual nodes, and finally the *lifetime* of the network. Most importantly, the proposed model results a huge enhancement in network *lifetime* that significantly overrides the increase in cost due to over-deployment. For an example, compared with the uniform node distribution strategy [4], with 2.4 times increase in the number of nodes, the *lifetime* is enhanced by 19 times.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMSCS.2017.2672553, IEEE Transactions on Multi-Scale Computing Systems

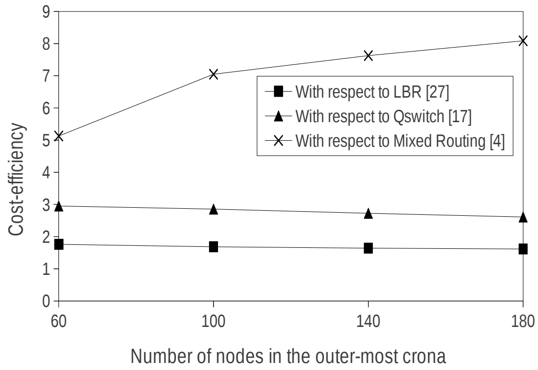

Fig. 16: Comparison of cost-efficiency ($P = 3$)

Thus the proposed scheme of over deployment of nodes coupled with load balanced routing via minimum-hop path, offers an elegant cost-effective solution to achieve maximum possible network lifetime with guaranteed coverage.

REFERENCES

[1] S. K. A. Imon, A. Khan, M. D. Francesco and S. K. Das, *RaSMaLai: A Randomized Switching Algorithm for Maximizing Lifetime in Tree-based Wireless Sensor Networks*, Proc. IEEE INFOCOM, pp. 2913-2921, 2013.

[2] R. Kacimi, R. Dhaou and A.-L. Beylot, *Load balancing techniques for lifetime maximizing in wireless sensor networks*, Ad Hoc Networks Journal, vol. 11, no. 8, pp. 2172-2186, 2013.

[3] A. Jarry, P. Leone, O. Powell and J. Rolim, *An optimal data propagation algorithm for maximizing the lifespan of sensor networks*, Proc. IEEE International Conference on Distributed Computing in Sensor Systems, pp 405-421, June 2006, San Francisco, USA.

[4] A. Jarry, P. Leone, S. E. Nikoletseas and J. D. P. Rolim, *Optimal data gathering paths and energy-balance mechanisms in wireless networks*, Ad Hoc Networks Journal, vol. 9, no.6, pp.1036-1048, 2011.

[5] J.C. Dagher, M.W. Marcellin and M.A. Neifeld, *A theory for maximizing the lifetime of sensor networks*, IEEE Transactions on Communications, vol. 55, no. 2, pp.323-332, 2007.

[6] C. Efthymiou, S. E. Nikoletseas and J. D. P. Rolim, *Energy balanced data propagation in wireless sensor networks*, Wireless Networks, Vol. 12, Issue 6, pp. 691 - 707, November 2006.

[7] A. Giridhar and P.R. Kumar, *Maximizing the functional lifetime of sensor networks*, Proc. 4^{th} international symposium on Information processing in sensor networks, pp. 5-12, 2005, Los Angeles, California.

[8] P. Casari, M. Nati, C. Petrioli and M. Zorzi, *ALBA: An Adaptive Load-Balanced Algorithm For Geographic Forwarding in Wireless Sensor Networks*, IEEE MILCOM 2006, pp. 1-9.

[9] S. C. Ergen and P. Varaiya, *Optimal Placement of Relay Nodes for Energy Efficiency in Sensor Networks*, Proc. IEEE International Conference on Communications (ICC), 2006, pp. 3473-3479.

[10] S. Olariu and I. Stojmenovic, *Design Guidelines for Maximizing Lifetime and Avoiding Energy Holes in Sensor Networks with Uniform Distribution and Uniform Reporting* , Proc. IEEE INFOCOM, pp. 1-12, April 2006.

[11] J. Gao and L. Zhang, *Load-Balanced Short-Path Routing in Wireless Networks*, Proc. of IEEE INFOCOM 2004, pp. 1098-1107, Mar. 2004.

[12] G. Chakraborty, *A Note on Cluster Size of WSN*, CODEC 2009, pp. 1-6, December 2009.

[13] M. Kalantari and M. Shayman, *Design Optimization of Multi-Sink Sensor Networks by Analogy to Electrostatic Theory*, Proc. IEEE Wireless Communications and Networking Conference (WCNC), 2006, pp. 431-438.

[14] P. Chatterjee and N. Das, *A Distributed Algorithm for Load-Balanced Routing in Multi-hop Wireless Sensor Networks*, Proc. ICDCN 2008, LNCS 4904, pp. 332-338.

[15] J. Li and P. Mohapatra, *An Analytical Model for the Energy Hole Problem in Many-to-One Sensor Networks*, Proc. 62^{nd} IEEE Vehicular Technology Conference, pp. 2721-2725, 2005.

[16] J. Li and P. Mohapatra, *Analytical Modeling and Mitigation Techniques for the Energy Hole Problems in Sensor Networks*, Pervasive and Mobile Computing, vol. 3, no. 8, pp. 233-254, 2007.

[17] X. Wu, G. Chen and S. K. Das, *Avoiding Energy Holes in Wireless Sensor Networks with Nonuniform Node Distribution*, IEEE Transactions on parallel and distributed systems, vol. 19, no. 5, pp.710-720, May 2008.

[18] A. Chamam, *On the Planning of Wireless Sensor Networks: Energy-Efficient Clustering under the Joint Routing and Coverage Constraint*, IEEE Transactions on Mobile Computing, vol. 8, no. 8, pp.1077-1086, August 2009.

[19] W. R. Heinzelman, A. Chandrakasan and H. Balakrishnan, *Energy-Efficient Communication Protocols for Wireless Microsensor Networks*, Proc. of Hawaii International Conference on Systems Sciences, pp. 1-10, January 2000.

[20] H. Alt, N. Blum, K. Mehlhorn and M. Paul, *Computing a Maximum Cardinality Matching in a Bipartite Graph in Time*, Information Processing Letters, Vol. 37, Issue 4, February 1991, pp. 237-240.

[21] N. Pervin, D. Layek, N. Das, *Localized Algorithm for Connected Set Cover Partitioning in Wireless Sensor Networks*, 1^{st} International Conference on Parallel, Distributed and Grid Computing(PDGC-2010), pp.229-234.

[22] A. Gallais, J. Carle, D. Simplot-Ryl and I. Stojmenovic, *Localized Sensor Area Coverage with Low Communication Overhead*, Fourth IEEE International Conference on Pervasive Computing and Communications (PerCom'06), pp.328-337.

[23] P. Erdős and A. Rnyi, *On the Existence of a Factor of Degree One of a Connected Random Graph*, Acta Mathematica Academiae Scientiarum Hungaricae 17 (1966), pp.359368.

[24] Y. Liu, H. Ngan and L.M. Ni, *Power-Aware Node Deployment in Wireless Sensor Networks*, International Journal of Distributed Sensor Networks, vol.3, pp. 225-241, 2007.

[25] Y. Yun and Y. Xia, *A Method for Deciding Node Densities in Non-Uniform Deployment of Wireless Sensors*, 11th International Symposium and Workshops on Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks (WiOpt), May 13-17, 2013, pp. 264-271.

[26] D. Wang, Y. Cheng, Y. Wang, D. P. Agrawal, *Lifetime Enhancement of Wireless Sensor Networks by Differentiable Node Density Deployment*, IEEE International Conference on Mobile Adhoc and Sensor Systems (MASS), Oct. 2006, pp. 546-549.

[27] P. Chatterjee and N. Das, *Distributed Data Gathering with Graded Node Distribution in Sensor Networks to Maximize Lifetime*, Proc. of the 25^{th} International Conference on Advanced Information Networking and Applications (AINA-2011), IEEE, pp. 435-442, Singapore, March, 2011.

[28] X. Zhang, Z. D. Wu, *A Non-Uniform Node Deployment Approach for Event Detection Sensor Networks*, IEEE 29^{th} International Performance Computing and Communications Conference (IPCCC) 2010, pp. 42 - 49.

[29] S. Liu, *A Lifetime-Extending Deployment Strategy for Multi-Hop Wireless Sensor Networks*, Proceedings of the 4^{th} Annual Communication Networks and Services Research Conference (CNSR) 2006, pp. 53-60 .

[30] Z. Abrams, A. Goel and S. Plotkin,*Set k-cover algorithms for energy efficient monitoring in Wireless Sensor Network*, Proc. 3^{rd} International Symposium on Information Processing in Sensor Networks, ACM 2004, pp.424-432.

[31] S. Slijepcevic and M. Potkonjak,*Power Efficient Organization of Wireless Sensor Networks*, Proc. IEEE ICC, June 2001, pp. 472-476.

[32] S. Bhattacharjee, N. Das, *Distributed Time Slot Assignment in Wireless Ad Hoc Networks for STDMA*, Proc. 2^{nd} International Conference, ICDCIT 2005, Bhubaneswar, India, December 22-24, 2005, pp 93-104.

[33] T Herman and S Tixeuil, *A distributed TDMA slot assignment algorithm for wireless sensor networks*, Proc. of the 1st Int. Workshop on Algorithmic Aspects of Wireless Sensor Networks, ALGOSENSORS 2004, Springer LNCS 3121, pp. 45-58, July 2004.

[34] S. S. Kulkarni and U. Arumugam, *Transformations for Write-All-With- Collision Model*, Proc. of the 7th Int. Conf. on Principles of Distributed Systems (OPODIS), Springer LNCS 3144, pp. 184-197, December 2003.

[35] S. Ramanathan and E. L. Lloyd, *Scheduling algorithms for multi-hop*

radio networks, IEEE/ACM Transactions on Networking, vol. 1, no. 2, pp. 166-177, 1993.

[36] M. N. Halgamuge, M. Zukerman, K. Ramamohanarao and H. L. Vu, *An estimation of sensor energy consumption*, Progress In Electromagnetics Research B, vol. 12, pp. 259 - 295, 2009

[37] Qianqian Yang, Shibo He, Junkun Li, Jiming Chen and Youxian Sun, *Energy-Efficient Probabilistic Area Coverage in Wireless Sensor Networks*, IEEE Transactions on Vehicular Technology, vol. 64, no. 1, pp. 367-377, January 2015

[38] Hamid Mahboubi and Fabrice Labeau, *Deployment Algorithms for Coverage Improvement in a Network of Mobile Sensors with Measurement Error in the Presence of Obstacles*, IEEE Sensors, pp. 1-4, November 2015

[39] S. Indhumathi and D. Venkatesan, *Improving Coverage Deployment for Dynamic Nodes using Genetic Algorithm in Wireless Sensor Networks*, Indian Journal of Science and Technology, vol. 8(16), pp. 1-6, July 2015

[40] Shibo He, Jiming Chen, Xu Li, Xuemin (Sherman) Shen, and Youxian Sun, *Mobility and Intruder Prior Information Improving the Barrier Coverage of Sparse Sensor Networks*, IEEE Transactions on Mobile Computing, vol. 13, no. 6, pp. 1268-1282, June 2014

[41] MICA2 mote datasheet. (2004). [Online]. Available: http://www.xbow.com/Products/Product-pdf-files/Wireless-pdf/MICA2-Datasheet.pdf

[42] Ford, L. R., Fulkerson, D. R., *Maximal flow through a network*, Canadian Journal of Mathematics, pp. 399 - 404, 1956.

[43] Jie Tian, Xiaoyuan Liang, Tan Yan, Mahesh Kumar Somashekar, Guiling Wang, and Cesar Bandera, *A Novel Set Division Algorithm for Joint Scheduling and Routing in Wireless Sensor Networks*, Springer Wireless Networks, vol. 21, Issue 5, pp. 1443-1455, July 2015.

[44] SENSEnuts mote datasheet. (2016). [Online]. Available: http://www.sensenuts.com

[45] Abdelraouf Ouadjaout, Noureddine Lasla, Miloud Bagaa, Messaoud Doudou, Cherif Zizoua, Mohamed Amine Kafi, Abdleouahid Derhab, Djamel Djenouri, and Nadjib Badache, *DZ50: Energy-Efficient Wireless Sensor Mote Platform for Low Data Rate Applications*, 5th International Conference on Emerging Ubiquitous Systems and Pervasive Networks (EUSPN-2014), Procedia Computer Science 37 (2014), pp. 189 - 195.