Network Connectivity under Probabilistic Communication Models in Wireless Sensor Networks

Mohamed Hefeeda and Hossein Ahmadi
School of Computing Science
Simon Fraser University
Surrey, BC, Canada

Abstract

Several previous works have experimentally shown that communication ranges of sensors are not regular disks. Rather, they follow probabilistic models. Yet, many current connectivity maintenance protocols assume the disk communication model for convenience and ease of analysis. In addition, current protocols do not provide any assessment of the quality of communication between nodes. In this paper, we take a first step in designing connectivity maintenance protocols for more realistic communication models. We propose a distributed connectivity maintenance protocol that explicitly accounts for the probabilistic nature of communication links and achieves a given target communication quality between nodes. Our protocol is simple to implement, and we demonstrate its robustness against random node failures, inaccuracy of node locations, and imperfect time synchronization of nodes using extensive simulations. We compare our protocol against others in the literature and show that it activates fewer number of nodes, consumes much less energy, and significantly prolongs the network lifetime.

1. Introduction

Network connectivity is one of the fundamental problems in wireless sensor networks. A network is connected if every pair of nodes can communicate with each other. To study network connectivity, many previous works represent the network with an undirected unweighted graph, where network connectivity is equivalent to graph connectivity. In the graph representation, there is an edge between two nodes if they are within the communication range of each other. Furthermore, the communication range of a node is typically assumed to be a disk with radius r_c, where r_c is referred to as the communication range of a node. We refer to this kind of connectivity as the *deterministic* connectivity model. The deterministic connectivity model started in wired networks, and then used widely in wireless ad hoc and sensor networks. While it is fairly accurate in wired networks, several papers, e.g. [1, 11], argue that the deterministic connectivity model is not appropriate for wireless networks. This is because it has been experimentally shown that communication ranges of nodes are not nice regular disks. Rather, they follow probabilistic models. Therefore, two wireless nodes can not said to be 'connected' or 'disconnected' in the perfect sense. Instead, a link between a pair of wireless nodes should have a probability of data delivery between these two nodes. In addition, it is neither sufficient nor precise to state that the network is simply connected. Rather, a quantitative measure of the quality of communications between arbitrary nodes in the network is needed.

Despite the experimental evidence of the inaccuracy of the deterministic connectivity model, many current connectivity maintenance protocols in the literature, e.g., [21, 22], continue to use it. The deterministic connectivity model is used because it facilitates the design and performance analysis of the protocols. By relying on the deterministic connectivity model, current protocols may not function properly in real environments. In addition, these protocols fail to provide any assessment of the quality of communication between nodes in a wireless sensor network.

In this paper, we take a first step in designing connectivity maintenance protocols for more realistic communication models. In particular, our contributions can be summarized as follows. First, we provide a quantitative measure of the quality of communication between nodes in sensor networks by defining the probability of packet delivery between arbitrary nodes in the network. We analytically derive this probability for common node deployment schemes such as grid, triangular lattice, and uniform deployments. Second, we propose a distributed connectivity maintenance protocol to achieve a given target communication quality

1-4244-1455-5/07/$25.00

between nodes. Our protocol is simple to implement, and we demonstrate its robustness against random node failures, inaccuracy of node locations, and imperfect time synchronization of nodes using extensive simulations. We show that our protocol minimizes the number of activated nodes and consumes much less energy than other protocols in the literature. In addition, the operation of our protocol does not depend on the specifics of the adopted communication model, which enables our protocol to be used with different models and in various environments.

The rest of this paper is organized as follows. In Sec. 2, we summarize the related works. In Sec. 3, we define the probabilistic connectivity notion and derive the probability of packet delivery in three node deployment schemes. In Sec. 4, we present our connectivity maintenance protocol, and in Sec. 5, we rigorously evaluate and compare it against other protocols in the literature. Sec. 6 concludes the paper.

2. Related Work

Several connectivity maintenance protocols have been proposed in the literature. We divide these protocols into two classes. In the first class, the protocol exchanges some messages to discover the connected components in the network [5, 6, 23]. For example, SPAN [6] maintains a list of neighbor nodes based on the received hello messages. Then, each node checks whether there exists a pair of neighbors that cannot reach each other directly or via one or two hops. If this is case, the node becomes active; otherwise, it turns itself off to save energy. PEAS [23] and ASCENT [5] send probing messages. A node in PEAS uses probing messages to discover whether there are other working nodes in the probing range, and it goes to sleep if it finds any. PEAS uses the number of working nodes in the probing range to set the sleep duration. ASCENT [5] uses the probing messages to estimate the reachability between neighboring nodes by measuring the packet loss rates, and uses this information to decide on which nodes should stay on. This class of protocols suffers from high communication overhead, which consumes a nontrivial fraction of node's energy.

The second class of connectivity maintenance protocols uses information about the communication range of sensors to maintain connectivity [21,22]. For example, the Geometric Adaptive Fidelity (GAF) protocol [22] divides the area into square cells such that all nodes inside a cell can communicate with all nodes in neighboring cells. GAF, then, keeps only one node active in each cell. These connectivity maintenance protocols rely on the assumption that the communication range is a disk, which is an over-simplification of wireless nodes in real environments [1,11]. Our proposed protocol assumes that the communication ranges follow a probabilistic model, which is more realistic. In addition, our protocol is more general and can support the deterministic communication model as well. In this case, we compare our protocol versus the two best deterministic connectivity protocols in the literature: one from the first class, SPAN [6], and another from the second class, GAF [22].

Recently, there have been some efforts to develop realistic models for connectivity in wireless sensor networks. One approach employs a geometric random graph representation of the network to reflect the probabilistic behavior of wireless communications [4, 10]. In this case, there is an edge between each pair of nodes with a probability related to the distance between them. The work in [10] assumes that this probability is given by the log-normal shadowing model [15]. The work in [4] derives the probability that a node in the network is isolated based on the node deployment density. The authors also show that this node isolation probability is an upper bound on the probability of having the network connected. Unlike our work, [4,10] do not propose a distributed protocol to maintain connectivity under probabilistic communication models.

A closely related problem to connectivity is coverage, where a subset of deployed nodes are activated such that any event in the monitored area is detected by at least one sensor. Several works address deterministic coverage [20, 24] as well as probabilistic coverage [2,9,12,25]. We focus on probabilistic connectivity maintenance protocols, and in the extended version of this paper, we develop an integrated protocol for probabilistic communication and probabilistic sensing models.

3. Network Connectivity under Probabilistic Communication Models

In this section, we present a simple probabilistic connectivity model. Using this model, we can quantify the quality of communication between nodes in sensor networks. We start by defining a quantitative metric for communication quality. Then, we derive bounds for this metric in three node deployment schemes: triangular mesh, square mesh, and uniform.

3.1. Communication Quality

The main function of a sensor network is to deliver data gathered by sensors to a processing center for possible actions. Therefore, we believe that the successful data delivery between any pair of nodes in the network is a good candidate for quantifying the communication quality in a sensor network. We quantify successful data delivery from node u to another node v by the probability that v correctly receives a packet transmitted by u. We call this probability the node-to-node packet delivery rate. From the sensor network design perspective, we are interested in the minimum node-to-node packet delivery rate in the network. Thus, we

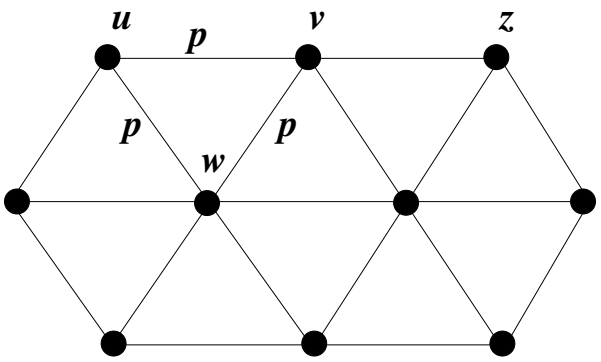

Figure 1. Probabilistic connectivity in triangular mesh.

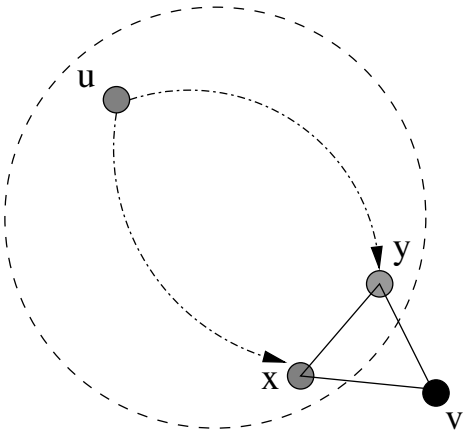

Figure 2. The triangular mesh construction process used in Theorem 1.

define the network packet delivery rate, or referred to simply as the network delivery rate, as follows.

Definition 1 (Network Delivery Rate) *The network delivery rate α of a sensor network is the minimum packet delivery rate between any pair of nodes in the network.*

Using the network delivery rate, we can define a probabilistic connectivity model for sensor networks as follows:

Definition 2 (α-connectivity) *A sensor network is said to be α-connected if the probability of delivering a packet between any arbitrary pair of nodes (i.e., network delivery rate) is at least α, where $0 \leq \alpha \leq 1$.*

In contrast to the deterministic connectivity model, the α-connectivity model provides a quantitative metric for measuring the communication quality in a sensor network. This is not only desirable, but also critical for sensor network applications that do require bounding the probability of losing a potentially important data item, such as intrusion detection systems in military applications. Furthermore, if we can determine α for a given node deployment method, we could potentially design a connectivity maintenance protocol to achieve a desired connectivity level. We derive bounds for α for common deployment methods in the following subsection. In Sec. 4, we propose a distributed protocol that achieves α-connectivity.

3.2 Computing Network Delivery Rates

We model a sensor network as a *weighted* graph $G(V, E)$, where V is the set of all nodes, and E is the set of edges between nodes. Every pair of nodes $u, v \in V$ have an edge $u \rightarrow v$ labeled with a packet delivery rate $p(u, v)$. $p(u, v)$ represents the probability of delivering packets from u to v over the *direct* wireless channel between them. Clearly, $p(u, v)$ depends on the probabilistic communication model used for the communication ranges of sensors. In addition, packets may flow between two nodes through multiple paths. We denote the total probability of delivering packets from node u to node v over all possible paths as $R(u, v)$. We refer to $R(u, v)$ as the node-to-node packet delivery rate.

The above graph representation of sensor networks is fairly general. For instance, it allows the creation of links between distant nodes. It also allows sensors to employ different communication models. It is, however, quite difficult to analytically compute the exact value of the network delivery rate α in this general setting. Therefore, we compute *lower bounds* on α under the following assumptions.

- All sensors use the same probabilistic communication model. This is not unrealistic assumption in many applications. For example, nodes in a surveillance application deployed in open areas could use the log-normal shadowing model [15], which captures path loss, shadowing effects, and Gaussian noise. Similarly, the same model could be used by nodes in a military intrusion detection system that are deployed on the ground at the same elevation. In addition, nodes in a forest fire detection system can all use a communication model that captures the characteristics of the surrounding environment such as the signal reflections from trees. Note that this assumption does not say that all nodes are deterministically identical, rather they follow the same probabilistic model. That is, the packet delivery rates over direct links have the same average $p = p(u, v)$.
- Links starting at the same sender node have independent delivery rates. For example, in Fig. 1, the packet delivery rates $p(u, v)$ and $p(u, w)$ are indepen-

dent. This assumption is needed to make the analysis tractable, otherwise, the analysis is not possible unless the nature of the dependence between links is completely specified. In our simulations, we do not assume independence and we verify that our results still hold.

- We only consider the delivery rates between immediate neighbors. For example, in Fig. 1, the *direct* delivery rate between nodes u and z is assumed to be zero. Therefore, our calculation of the network delivery rate is conservative and should be viewed as a lower bound. We notice that this is not totally unrealistic, because as the distance between nodes increases the signal fades rapidly and most wireless receivers process a signal only if its level exceeds a certain threshold.

- No retransmissions in the MAC layer. This assumption is needed to find the minimum network delivery rates regardless of the details of the employed MAC protocol, such as the maximum number of retransmissions and the random backoff scheme. This assumption actually makes our analysis more general, and therefore, our results and the proposed connectivity maintenance protocol can be used with different MAC protocols. As shown by our NS-2 simulations (Sec. 5), which are performed with MAC retransmissions, our analysis indeed provides lower bounds on the network delivery rates.

Under these assumptions, we first derive the lower bound on the network delivery rate α for nodes deployed on a triangular mesh as shown in Fig. 1. The following theorem gives this bound.

Theorem 1 *Given nodes deployed on a triangular mesh, and the average packet delivery rate between any neighboring nodes is p, the network delivery rate α is at least* $(2p-1)/p^2$.

Proof We prove this theorem by construction. First, we begin with a triangle. Then, we expand it by adding nodes one by one to make the triangular mesh as in Fig. 2. Now, we find the delivery rate between source and v at each step. There are two links connecting v to x and y. Therefore, the accumulated delivery rate at v is $1-(1-pR(u,x))(1-pR(u,y))$. Since $R(u,x)$ and $R(u,y)$ are greater than or equal to α, we get $R(u,v) \geq 1-(1-p\alpha)^2$. This result is true for every pair of nodes. Now, assume two nodes, i and j, with the least node-to-node delivery rate. By definition, we have $R(i,j)=\alpha$. On the other hand, we have $R(i,j) \geq 1-(1-p\alpha)^2$ from the above discussion. Therefore, we have $1-(1-p\alpha)^2 \leq \alpha$, or $\alpha \geq (2p-1)/p^2$.

Next, we derive the lower bound of the network delivery rate for nodes deployed on a square mesh. Due to space limitations, the proofs of the following two theorems are given in the extended version of the paper [8].

Theorem 2 *Given nodes deployed on a square mesh, and the average packet delivery rate between any neighboring nodes is p, the network delivery rate α is at least* $\min(\frac{p+p^2-1}{p^3}, p^2-2p)$.

Finally, we extend the analysis of network delivery rate to uniform random node distribution.

Theorem 3 *Given nodes deployed uniformly at random with density ρ, the network delivery rate α is at least* $\int_{x=0}^{1}(1-e^{-\rho\pi d^2}\sum_{k=0}^{3}\frac{(\rho\pi d^2)^k}{k!})^n dx$.

4. Probabilistic Connectivity Maintenance Protocol

In this section, we present a new Probabilistic Connectivity Maintenance Protocol (PCMP), which employs probabilistic communication models. We start by presenting an overview of our protocol, followed by more details.

4.1. Overview of PCMP

The goal of PCMP is to activate a subset of deployed nodes such that the probability of delivering packets between any arbitrary nodes in the network is at least α, i.e., keep the network α-connected. To achieve this goal, the protocol activates nodes to form an approximate triangular mesh. The activation process is done in a distributed manner as described below. The spacing between nodes in the triangular mesh is computed to achieve the target network delivery rate. We use the bound proved in Theorem 1 and information from the adopted communication model in computing the spacing. The details of this computation are given in Sec. 4.2. For now, let us assume that the spacing between nodes is determined to be d. We chose to activate nodes on a triangular mesh for two reasons. First, it enables us to use PCMP with the deterministic connectivity model, in addition to the probabilistic model. In this case, activating nodes on the triangular mesh has been shown to be optimal in terms of number nodes activated [3]. Second, our analysis for the triangular mesh in Sec. 3.2 provides a simpler and tighter lower bound than the analysis for the square mesh, as confirmed by our simulations. PCMP does *not* require that nodes to be deployed on a triangular mesh. It is the activated subset of them that forms a triangular mesh. Node deployment can follow any distribution. In our simulations, we deploy nodes uniformly at random.

The idea of our protocol is to start the activation process by one node, and iteratively activate other nodes until a triangular mesh-like structure is formed over the whole area. PCMP works in rounds of R seconds each, where in each round a subset of nodes are active to maintain the whole network connected and the rest of the nodes are put

in sleep mode to conserve energy. R is chosen to be much smaller than the average lifetime of sensors. In the beginning of each round, all nodes start running PCMP independent of each other. This implies that nodes need to be time-synchronized. In our simulations, we show that only coarse-grained time synchronization is needed and PCMP is quite robust to clock drifts. A number of messages will be exchanged between nodes to determine which of them should be active during the current round, and which should sleep till the beginning of the next round. The time it takes the protocol to determine active/sleep nodes is called the convergence time.

In PCMP, a node can be in one of four states: ACTIVE, SLEEP, WAIT, or START. In the beginning of a round, each node sets its state to be START, and selects a random startup timer T_s proportional to its remaining energy level. The node with the smallest T_s will become active, and broadcasts an activation message to all nodes in its communication range. The sender of the activation message is called the activator. The activation message should contain the coordinates of the activator. That is, PCMP assumes that nodes know their locations, which can be done by any efficient localization scheme such as [7, 16]. In the evaluation section, we show that PCMP is robust to inaccuracy of node locations, and thus require only light-weight localization schemes. The activation message tries to activate nodes at vertices of the hexagon centered at the activator, while putting all other nodes within that hexagon to sleep. A node receiving the activation message can determine whether it is a vertex of the hexagon by measuring the distance and angle between itself and the activator. If the angle is multiple of $\pi/3$ and the distance is d, then the node sets its state to ACTIVE and it becomes a new activator. Otherwise it goes to SLEEP state.

Nodes may not always be found at vertices of the triangular mesh because of randomness in node deployment or because of node failure. PCMP tries to activate the closest nodes to hexagon vertices in a distributed manner. Every node receiving an activation message calculates an activation timer T_a as a function of its closeness to the nearest vertex of the hexagon using the following equation: $T_a = \tau_a(d_v^2 + {d_a}^2\gamma^2)$, where d_v, and d_a are the Euclidean distances between the node and the vertex, and the node and the activator, respectively; γ is the angle between the line connecting the node with the activator and the line connecting the vertex with the activator; and τ_a is a constant. Notice that the closer the node gets to the vertex, the smaller the T_a will be. After computing T_a, a node moves to WAIT state and stays in this state till its T_a timer either expires or is canceled. When the smallest T_a timer expires, its corresponding node changes its state to ACTIVE. This node then becomes a new activator and broadcasts an activation message to its neighbors. When receiving the new activation message, nodes in WAIT state cancel their T_a timers and move to SLEEP state.

A similar node activation method was used in our previous work on coverage protocols [9]. In this paper, we extend this method to achieve probabilistic connectivity, where the spacing between nodes in the triangular mesh is determined form the adopted communication model and is based on our analysis in Sec. 3.2.

4.2. Details of PCMP

In this section, we show how the spacing between activated nodes in the triangular mesh is computed to achieve α-connectivity. We refer to this spacing as d_α.

The nodes activated by our PCMP protocol form an approximate triangular mesh. The spacing between these nodes is at most d_α. According to Theorem 1, the network delivery rate α in the triangular mesh is at least $(2p-1)/p^2$, where p is the average packet delivery rate on a link between two neighboring nodes. That is, $\alpha \geq (2p-1)/p^2$. Therefore, we need:

$$p \geq (1 - \sqrt{1-\alpha})/\alpha \tag{1}$$

to meet the target network delivery rate. p is related to the spacing d_α through the assumed communication model. Thus, we use the communication model to compute d_α to yield the required p. To illustrate, consider the log-normal shadowing model widely used in network simulators, such as NS-2 [19] and OPNET [14], and in several previous works, e.g., [10, 18]. In this model, the power of the received signal $P_r(d)$ at a distance d from a sender transmitting at power P_t is given by [15, Sec. 3.9]:

$$P_r(d) = P_t - (\overline{PL(d_0)} + 10n\log(\frac{d}{d_0}) + X_\sigma), \tag{2}$$

where X_σ is a zero-mean random variable with Gaussian distribution, n is a constant specified by the environment, and $\overline{PL(d_0)}$ is the mean path loss measured at the reference distance d_0, which is usually set to 1 m. Wireless adapters can successfully receive data if the signal strength exceeds a certain threshold, say γ. The probability that the signal strength exceeds γ is [15, Sec. 3.9]:

$$\Pr[P_r(d) > \gamma] = \frac{1}{2}[1 - \text{erf}(\frac{\gamma - \overline{P_r(d)}}{\sigma\sqrt{2}})] \tag{3}$$

Assuming that the signal strength does not significantly change during the transmission of a single packet, the average packet delivery rate p is given by $p = \Pr[P_r(d) > \gamma]$. Solving (1) and (3) for the spacing d_α, we get:

$$d_\alpha \leq d_0 e^{[P_t - \gamma - \overline{PL(d_0)} + \sigma\sqrt{2}\text{erf}^{-1}(1 - 2\frac{1-\sqrt{1-\alpha}}{\alpha})]/10n}. \tag{4}$$

Table 1. Parameters used for the communication model.

Parameter	Value
Path-loss exponent n	2.2
Shadowing standard deviation σ	4.0
Reference distance d_0	$1m$
Transmission power P_t	$1mW$
Reception threshold γ	$10^{-9}mW$

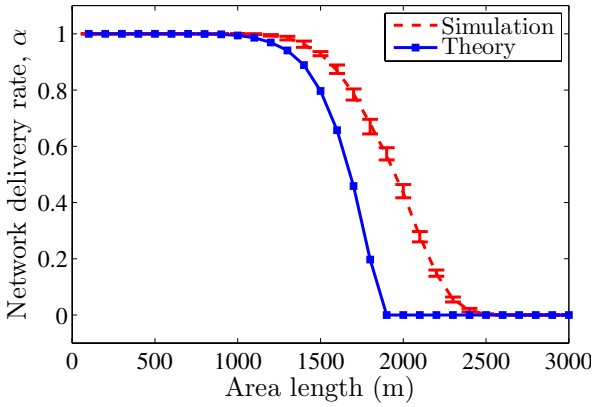

Figure 3. Network delivery rate in triangular mesh. For the simulation data, we show the minimum, average, and maximum values.

Setting the spacing between activated nodes on the triangular mesh according to (4) will achieve the target network delivery rate under the log-normal shadowing model. Computing d_α for other communication models can be done in a similar way.

We emphasize that the operation of our PCMP protocol does not depend on the adopted communication model. PCMP needs only the value of d_α, and the protocol functions the same regardless of the model. Thus, PCMP can be used with different communication models.

5. Evaluation

In this section, we evaluate our proposed protocol and compare it against others. We start by describing our experimental setup. Then, we validate our theoretical lower bounds on network delivery rate. Next, we analyze the performance of our protocol and show its robustness against node location inaccuracy, node failures, and imperfect time synchronization of nodes. Finally, we show that our protocol outperforms the best two other connectivity protocols in the literature: SPAN [6] and GAF [22].

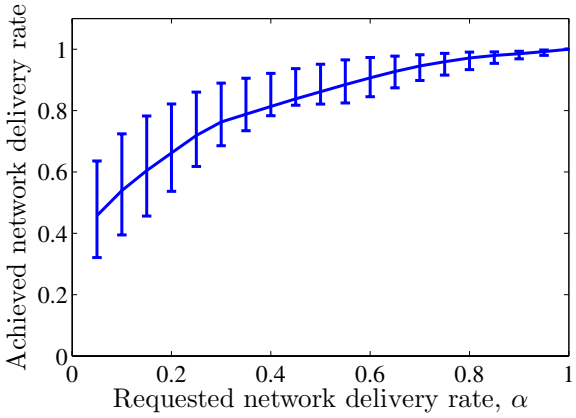

Figure 4. PCMP achieves the requested network delivery rate in all cases. The minimum, average, and maximum values are shown.

We use the following setup in all of our experiments, unless otherwise specified. We use NS-2 version 2.30 [19] in the simulation. We deploy $1,000$ nodes uniformly at random in a $1km \times 1km$ area. All nodes use the log-normal shadowing model given in Eq. (2) for radio communications. We compute the parameters for this model based on the specifications of MicaZ motes [13], and we list these parameters in Table 1. We employ the energy model specified in [23]. In this model, the power consumption for transmission, reception, idling, and sleeping are 60, 12, 12, 0.03 mWatt, respectively. We use an initial energy of 60 Jules for each node. We set the wireless channel bandwidth at 40 kbps. For our PCMP protocol, we have the following parameters. The round length R is 100 seconds, which is much smaller than the network lifetime. The average message size is 34 bytes. The maximum value for the startup timer τ_s is set to $1/E_r$, where E_r is the fraction of the remaining energy in the node.

We repeat each experiment 10 times with different seeds, and we report the average values over all of them. We also show the minimum and maximum values if they do not clutter the graph. Due to space limitations, only a sample of the results are presented. All figures and results are available in the extended version of the paper [8].

5.1. Validation of our Analysis

In the first set of experiments, we measure the network delivery rate from simulation and compare it against the analytical lower bound. To measure the network delivery rate, we randomly choose a node, and make it broadcast $1,000$ small packets of size 8 bytes each to all other nodes. Then, we record the number of received packets at each node. The

network delivery rate is the minimum number of received packets by any node divided by $1,000$.

For the triangular mesh, we activate 100 nodes (out of the $1,000$ deployed) and make them form a triangular mesh over the whole area. We vary the spacing between neighboring nodes d by varying the area over which the triangular mesh is constructed. Varying the spacing between nodes corresponds to varying the probability of delivering packets between neighboring nodes p. We run the simulation for each value of d 10 times and we measure the network delivery rate. We also compute the network delivery rate from Theorem 1 for each value of d. We repeat the experiment again, except we vary the transmission power P_t and fix everything else in the communication model. The sample results shown in Fig. 3 confirm that our lower bound is correct and conservative.

Finally, we repeat the above experiment for the square mesh and uniform deployments. Again, the results given in [8] validate our analysis.

5.2. Performance and Robustness

We first study the performance of our PCMP protocol. We run PCMP over $1,000$ uniformly deployed nodes that use the log-normal shadowing model. We vary the requested network delivery rate α between 0.1 and 1.0. For each value of α, we compute the spacing between neighboring nodes d_α from Eq. (4), and we run PCMP in the simulation with this value. We measure the achieved network delivery rate by PCMP. The results shown in Fig. 4 demonstrate that our protocol met the requested network delivery rate in all cases.

Next, we study the robustness of PCMP against inaccuracy in node locations. We use the same setup as before except that we add errors to node locations. We add random values in the interval $[-er_{max}, er_{max}]$ to both x and y coordinates of the real location of each deployed node. We vary er_{max} between 0 and $20m$. We compute the network delivery rate after the protocol converges. The results indicate that the network delivery rate is always maintained as shown in Fig. 5(a). Therefore, PCMP is robust against location inaccuracy. There is a small cost, however, with this location inaccuracy. As shown on the same figure (notice the two y-axes), the number of activated nodes slightly increases in case of inaccurate locations. There is less than 7% increase in number of activated nodes for location errors of up to $\pm 20m$.

Exact time synchronization of nodes in a large scale sensor network is costly to achieve. We study the robustness of PCMP against the granularity of time synchronization. To do this, we add random values in the interval $[0, d_{max}]$ to the clock of each node at the beginning of the simulation. We change d_{max} between 0 and $500ms$. As shown in Fig. 5(b), the network delivery rate is ensured even with high values of clock drift. In addition, the number of active nodes does not increase if the drift is less than the convergence time of the protocol (around $75ms$). This means that our protocol is robust against fairly large clock drifts, and thus, it needs only light-weight, coarse-grained, time synchronization schemes.

Finally, we show that PCMP is robust against random node failures. We choose a fraction f of all deployed nodes to be failed within the first 200 rounds of the protocol execution, and we randomly schedule them to fail. We change the fraction of failed nodes, f, and plot the network delivery rate as time progresses in Fig. 5(c). The results indicate that PCMP can ensure network delivery rate even with high failure rates.

5.3. Comparison with other Protocols

We compare our PCMP protocol against SPAN [6] and GAF [22] protocols since they are the best and widely cited other protocols in the literature. Both protocols were described in Sec. 2. We use the NS-2 code for SPAN which is published by its authors at [17]. The code for GAF is included in version 2.30 of NS-2.

First, we verify that all protocols indeed achieve the disk-model deterministic connectivity. We check this for two different node deployment densities. We set the communication range of nodes to $100m$. Based on that, the length of GAF grid cells are set to $44m$, according to the relationship presented in [22]. To measure connectivity, we run a breadth first search to find the largest connected component of nodes. We divide the size of this component by the total number of nodes. The results of this experiment show that all protocols achieve 100% deterministic connectivity.

Next, we compare the three protocols against a critical metric in sensor networks: energy consumption. We fix all parameters in the simulator and run the three protocols one at a time. We periodically collect the amount of remaining energy in every deployed node. Then, we sum these values and compute the fraction of energy remained in the network with respect to the initial energy at time 0. For each protocol, we run the simulator 10 times, and for long periods ($35,000$ seconds). The average results are shown in Fig. 6(a). As the figure shows, PCMP consumes much less energy than the other two protocols. For example, after $15,000$ seconds from the start, nodes under SPAN and GAF have less than 20% of their initial energy, while using PCMP nodes have 60% of their initial energy. The reasons behind the energy saving of PCMP over GAF is that PCMP activates much fewer number of nodes than GAF: The average number of active nodes under PCMP was always less than 70 in all cases, while GAF activated at least 160 nodes. Nodes in active mode consumes significantly more energy

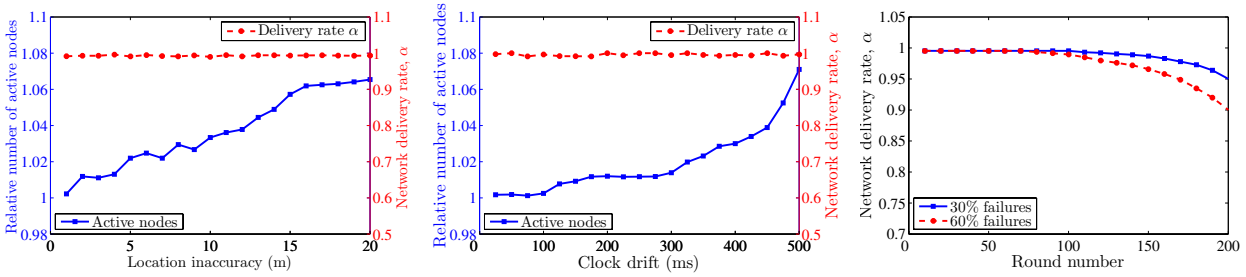

Figure 5. Robustness of PCMP against: (a) Location inaccuracy, (b) Clock drifts, and (c) Random node failures.

than nodes in sleep mode. On the other hand, SPAN activates slightly less number of nodes than PCMP, but it has much higher communication overhead due to the frequent exchange of hello messages among nodes.

Finally, we compare the network lifetime under the three protocols. Since these are connectivity protocols, we plot the average network packet delivery rate as the time progresses. The results in figure 6(b) demonstrate that our protocol extends (almost doubles) the lifetime of the network. This is because of the energy saving as described above.

6. Conclusions and Future Work

We presented a simple probabilistic connectivity model under which we could quantify the quality of communication between nodes in wireless sensor networks. We introduced the network packet delivery rate as a quantitative metric for communication quality. We derived lower bounds for this metric in three common node deployment schemes: triangular mesh, square mesh, and uniform. Based on the probabilistic connectivity model, we designed a distributed Probabilistic Connectivity Maintenance Protocol (PCMP). PCMP is a fairly general protocol that can employ different probabilistic as well as deterministic communication models, with minimal configuration. Through extensive simulations in NS-2 with nodes using the log-normal shadowing model for their radio communications, we showed that PCMP: (i) achieves the target network delivery rates; and (ii) is quite robust to several factors common in real environments such as node failures, drifts in node clocks, and errors in node locations. We compared our protocol versus two of the best connectivity maintenance protocols in the literature: SPAN [6] and GAF [22]. Our simulation results demonstrated that our protocol significantly outperforms them in several aspects, including: number of activated nodes, energy consumption, and network lifetime.

The work in this paper can be extended in several directions. One possible extension is to consider different communication models for nodes deployed in the area and forming one network. Different models are needed if heterogeneous nodes are deployed, or the environmental conditions vary significantly from one location to another. For example, some nodes could be deployed on the ground while others are deployed at different heights on a mountain.

Acknowledgment

This work is partially supported by the Natural Sciences and Engineering Research Council (NSERC) of Canada under Discovery Grant #313083 and RTI Grant #344619.

References

[1] D. Aguayo, J. Bicket, S. Biswas, G. Judd, and R. Morris. Link-level measurements from an 802.11b mesh network. In *Proc. of SIGCOMM'04*, pages 121–132, Portland, OR, August 2004.

[2] N. Ahmed, S. Kanhere, and S. Jha. Probabilistic coverage in wireless sensor networks. In *Proc. of IEEE Conference on Local Computer Networks (LCN'05)*, pages 672–681, Sydney, Australia, November 2005.

[3] X. Bai, S. Kuma, D. Xua, Z. Yun, and T. H. La. Deploying wireless sensors to achieve both coverage and connectivity. In *Proc. of ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc'06)*, pages 131–142, Florence, Italy, 2006.

[4] C. Bettstetter and C. Hartmann. Connectivity of wireless multihop networks in a shadow fading environment. In *Proc. of ACM International Workshop on Modeling Analysis and Simulation of Wireless and Mobile Systems (MSWIM'03)*, pages 28–32, San Diego, CA, September 2003.

[5] A. Cerpa and D. Estrin. ASCENT: Adaptive self-configuring sensor networks topologies. *IEEE Transactions on Mobile Computing*, 3(3):272–285, July-August 2004.

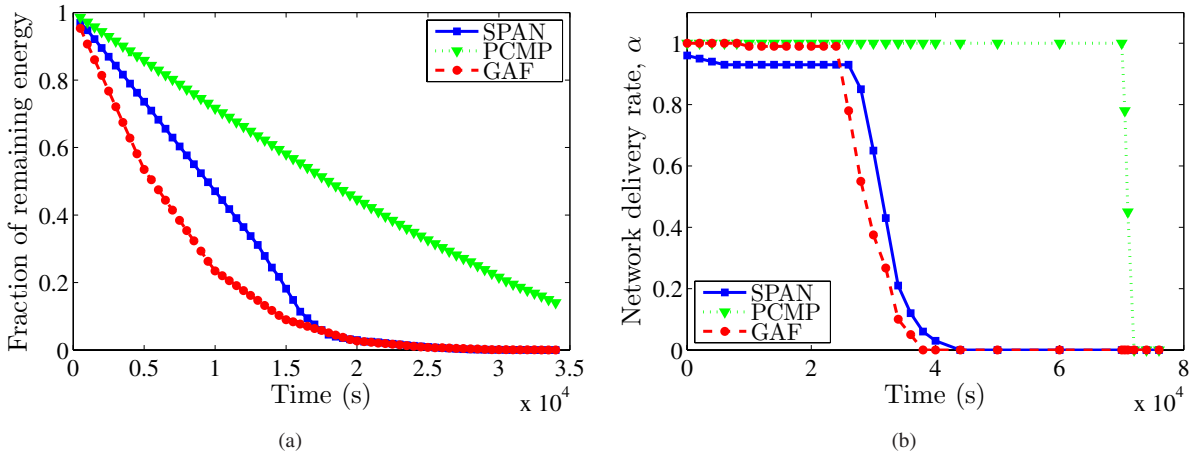

Figure 6. Comparing PCMP vs. SPAN and GAF: (a) Energy consumption, and (b) Network lifetime.

[6] B. Chen, K. Jamieson, H. Balakrishnan, and R. Morris. SPAN: An energy-efficient coordination algorithm for topology maintenance in ad hoc wireless networks. *Wireless Networks*, 8(5):481–494, September 2002.

[7] L. Doherty, L. E. Ghaoui, and K. Pister. Convex position estimation in wireless sensor networks. In *Proc. of IEEE INFOCOM'01*, pages 1655–1663, Anchorage, AK, April 2001.

[8] M. Hefeeda and H. Ahmadi. Network connectivity under probabilistic communication models in sensor networks. Technical Report 2007-10, Simon Fraser University, School of Computing Science, April 2007. Available at: http://www.cs.sfu.ca/∼mhefeeda.

[9] M. Hefeeda and H. Ahmadi. A probabilistic coverage protocol for wireless sensor networks. In *Proc. of IEEE International Conference on Network Protocols (ICNP'07)*, Beijing, China, October 2007.

[10] R. Hekmat and P. Van Mieghem. Connectivity in wireless ad-hoc networks with a log-normal radio model. *Mobile Networks and Applications*, 11(3):351–360, June 2006.

[11] D. Kotz, C. Newport, and C. Elliott. The mistaken axioms of wireless-network research. Technical Report 2003-467, Dartmouth College, Computer Science Department, July 2003.

[12] B. Liu and D. Towsley. A study of the coverage of large-scale sensor networks. In *Proc. of IEEE International Conference on Mobile Ad-hoc and Sensor Systems (MASS'04)*, pages 475–483, Fort Lauderdale, FL, October 2004.

[13] MicaZ Data Sheet. http://xbow.com/Products/Product_pdf_files /Wireless_pdf/MICAz_Datasheet.pdf.

[14] OPNET Web Page. http://www.opnet.com.

[15] T. Rappaport. *Wireless communications: principles and practice*. Prentice Hall, 1996.

[16] A. Savvides, C. Han, and M. Strivastava. Dynamic fine-grained localization in ad-hoc networks of sensors. In *Proc. of ACM International Conference on Mobile Computing and Networking (MobiCom'01)*, pages 166–179, Rome, Italy, July 2001.

[17] SPAN Reference Implementation. http://pdos.csail.mit.edu/∼benjie/span/span_ns_1.1.tar.gz.

[18] I. Stojmenovic, A. Nayak, J. Kuruvila, F. Ovalle-Martinez, and E. Villanueva-Pena. Physical layer impact on the design of and performance of routing and broadcasting protocols in ad hoc and sensor networks. *Elsevier Computer Communications*, 28(10):1138–1151, June 2005.

[19] The Network Simulator (NS-2) Web Page. http://nsnam.isi.edu/nsnam/.

[20] G. Xing, X. Wang, Y. Zhang, C. Lu, R. Pless, and C. Gill. Integrated coverage and connectivity configuration for energy conservation in sensor networks. *ACM Transactions on Sensor Networks*, 1(1):36–72, August 2005.

[21] Y. Xu, J. Heidemann, and D. Estrin. Adaptive energy-conserving routing for multihop ad hoc networks. Research Report 527, USC/Information Sciences Institute, October 2000. Available at: http://www.isi.edu/∼johnh/PAPERS/Xu00a.html.

[22] Y. Xu, J. Heidemann, and D. Estrin. Geography-informed energy conservation for ad hoc routing. In *Proc. of ACM nternational Conference on Mobile Computing and Networking (MobiCom'01)*, pages 70–84, Rome, Italy, July 2001.

[23] F. Ye, G. Zhong, J. Cheng, S. Lu, and L. Zhang. PEAS: A robust energy conserving protocol for long-lived sensor networks. In *Proc. of IEEE International Conference on Distributed Computing Systems (ICDCS'03)*, pages 28–37, Providence, RI, May 2003.

[24] H. Zhang and J. Hou. Maintaining sensing coverage and connectivity in large sensor networks. *Ad Hoc and Sensor Wireless Networks: An International Journal*, 1(1-2):89–123, January 2005.

[25] Y. Zou and K. Chakrabarty. A distributed coverage- and connectivity-centric technique for selecting active nodes in wireless sensor networks. *IEEE Transactions on Computers*, 54(8):978–991, August 2005.