ISSN:2231-0711

A Study on ASCENT in Wireless Sensor Networks

Tummala Sasanth, Dr Syed Umar

Department of ECM,
KL University, A.P. INDIA.

***Abstract*— The nodes will have to self-configure to establish a topology that provides communication under stringent energy constraints. ASCENT builds on the notion that, as density increases, only a subset of the nodes are necessary to establish a routing forwarding backbone. In ASCENT, each node assesses its connectivity and adapts its participation in the multi-hop network topology based on the measured operating region. This paper motivates and describes the ASCENT algorithm and presents analysis, simulation, and experimental measurements. We show that the system achieves linear increase in energy savings as a function of the density and the convergence time required in case of node failures while still providing adequate connectivity.**

***Keywords*—Wireless sensor networks, adaptive topology, topology control, energy conservation.**

I. INTRODUCTION

The availability of micro sensors and low-power wireless communications will enable the deployment of densely distributed sensor/actuator networks for a wide range of environmental monitoring applications from urban to wilderness environments; indoors and outdoors; and encompassing a variety of data types including acoustic, image, and various chemical and physical properties. The sensor nodes will perform significant signal processing, computation, and network self-configuration to achieve scalable, robust, and long-lived networks [2], [8], [7]. More specifically, sensor nodes will do local processing to reduce communications and, consequently, energy costs. In this paper, we describe and present simulation and experimental performance studies for a form of adaptive self-configuration designed for sensor networks. As we argue in Section 2, these unattended systems will need to self-configure and adapt to a wide variety of environmental dynamics and terrain conditions. These conditions produce regions with non-uniform communication density. We suggest that one of the ways system designers can address such challenging operating conditions is by deploying redundant nodes and designing the system algorithms to make use of that redundancy over time to extend the systems life. In ASCENT, each node assesses its connectivity and adapts its participation in the multi-hop network topology based on the measured operating region. For instance, a node:

- . signals when it detects high packet loss, requesting

additional nodes in the region to join the network in order to relay messages, reduces its duty cycle if it detects high packet losses due to collisions, . probes the local communication environment and does not join the multi-hop routing infrastructure until it is "helpful" to do so. Why can this adaptive configuration not be done from a central node? In addition to the scaling and robustness limitations of centralized solutions, a single node cannot directly sense the conditions of nodes distributed elsewhere in space. Consequently, other nodes would need to communicate detailed information about the state of their connectivity in order for the central node to determine who should join the multi-hop network. When energy is a constraint and the environment is dynamic, distributed approaches are attractive and possibly are the only practical approach [22] because they avoid transmitting dynamic state information repeatedly across the network. Pottie and Kaiser [22] initiated work in the general area of wireless sensor networks by establishing that scalable wireless sensor networks require multi-hop operation to avoid sending large amounts of data over long distances. They went on to define techniques by which wireless nodes discover their neighbors and acquire synchronism. Given this basic bootstrapping capability, our work addresses the next level of automatic configuration that will be needed to realize envisioned sensor networks, namely, how to form the multi-hop topology [7]. Given the ability to send and receive packets and the objective of forming an energy-efficient multi-hop network, we apply well-known techniques from MAC layer protocols to the problem of distributed topology formation. In the following section, we present a sensor network scenario, stating our assumptions and contributions. Related work is reviewed in Section 3. Section 4 describes ASCENT in more detail. In Section 5, we present some initial analysis, simulation, and experimental results using ASCENT. Finally, in Section 6, we conclude.

2 DISTRIBUTED SENSOR NETWORK SCENARIO

To motivate our research, consider a habitat monitoring sensor network that is to be deployed in a remote forest. Deployment of this network can be done, for example, by dropping a large number of sensor nodes from a plane or placing them by hand. In this example and in many other anticipated applications of ad hoc wireless sensor networks [5], the deployed systems must be designed to operate under the following conditions and constraints:

. Ad hoc deployment: We cannot expect the sensor field to be deployed in a regular fashion (e.g., a linear array, 2D lattice). More importantly, uniform deployment does not correspond to uniform connectivity owing to unpredictable propagation effects when nodes, and therefore antennae, are close to the ground and other surfaces. . Energy constraints: The nodes (or at least some significant subset) will be untethered for power as well as communications and therefore the system must be designed to expend as little energy as is possible in order to maximize network lifetime.

. Unattended operation under dynamics: The anticipated number of elements in these systems will preclude manual configuration, and the environmental dynamics will

preclude design-time pre configuration. In many such contexts, it will be far easier to deploy larger numbers of nodes initially than to deploy additional nodes or additional energy reserves at a later date (similar to the economics of stringing cable for wired networks). In this paper, we present one way in which nodes can exploit the resulting redundancy in order to extend system lifetime. If we use too few of the deployed nodes, the distance between neighboring nodes will be too great and the packet loss rate will increase or the energy required to transmit the data over the longer distances will be prohibitive. If we use all deployed nodes simultaneously, the system will be expending unnecessary energy at best and, at worst, the nodes may interfere with one another by congesting the channel. In the process of finding an equilibrium, we are not trying to use a distributed localized algorithm to identify a single optimal solution. Rather, this form of adaptive self –configuration using localized algorithms is well suited to problem spaces that have a large number of possible solutions; in this context, a large solution space translates into dense node deployment. Our simulation and experimental results confirm that this is the case for our application. We enumerate the following assumptions that apply to the remainder of our work: We assume a Carrier Sense Multiple Access (CSMA) MAC protocol with the capacity to work in promiscuous mode. This clearly introduces the possibilities for resource contention when too many neighboring nodes participate in the multi-hop network. Our approach should be relevant to TDMA MACs as well because distributed slot allocation schemes will also have degraded performance with increased load.

Our algorithm reacts when links experience high packet loss. The ASCENT mechanism does not detect or repair network partitions of the underlying raw topology. Partitions are more prevalent when node density is low, and our approach is not applicable because, in general, all nodes will be needed to form an effective network. Of course, network partitions can occur even in dense arrays when a swath of nodes are destroyed or obstructed. When such network partitions do occur, complementary system mechanisms will be needed; for example, detecting partitions in the multi-hop sensor network by exploiting information from long range radios deployed on a subset of nodes and used sparingly because of the power required. We leave such complementary techniques for network partition detection and repair to future work. The two primary contributions of our design are:

. The use of adaptive techniques that permit applications to configure the underlying topology based on their needs while trying to save energy to extend network lifetime. Our work does not presume a particular model of fairness, degree of connectivity, or capacity required.

. The use of self-configuring techniques that react to operating conditions measured locally. Our work is not restricted to the radio propagation model, the geographical distribution of nodes, or the routing mechanisms used.

3 ASCENT DESIGN

ASCENT adaptively elects “active” nodes from all nodes in the network. Active nodes stay awake all the time and perform multi-hop packet routing, while the rest of the nodes remain “passive” and periodically check if they should become active.

Consider a simple sensor network for data gathering similar to the network described in Section 2. Fig. 1 shows a simplified schematic for ASCENT during initialization in a high-density region. For the sake of clarity, we show only the formation of a two-hop network. This analysis may be extended to networks of larger sizes. Initially, only some nodes are active. The other nodes remain passively listening to packets but not transmitting. This situation is depicted in Fig. 1a. The source starts transmitting data packets toward the sink. Because the sink is at the limit of radio range, it gets very high packet loss from the source. We call this situation a communication hole.

The sink then starts sending help messages to signal neighbors that are in listen-only mode—also called passive neighbors—to join the network. When a neighbor receives a help message, it may decide to join the network. This situation is illustrated in Fig. 1b.

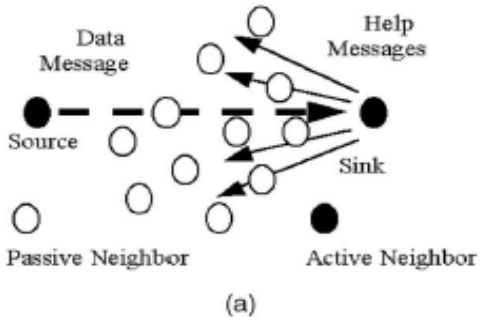
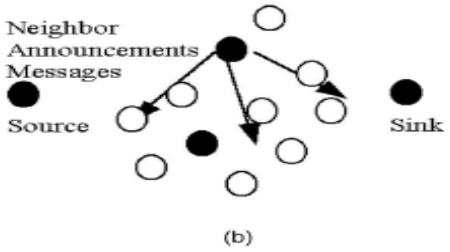
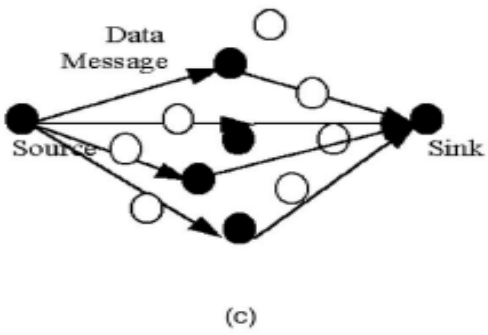


Fig. 1. Network self-configuration example.(a) Communication hole.



(b) Transition



(c) Final state.

When a node joins the network, it starts transmitting and receiving packets, i.e., it becomes an active neighbor. As soon as a node decides to join the network, it signals the

existence of a new active neighbor to other passive neighbors by sending an neighbor announcement message. This situation continues until the number of active nodes stabilizes on a certain value and the cycle stops (see Fig. 1c). When the process completes, the group of newly active neighbors that have joined the network make the delivery of data from source to sink more reliable.

The process will restart when some future network event (e.g., node failure) or environmental effect (e.g., new obstacle) causes packet loss again. In this section, we describe the ASCENT algorithm and their components. We elaborate on several design choices while we describe the scheme. Our initial analysis, simulations, and experiments in Section 5 focus only on a subset of these design choices.

3.1 ASCENT State Transitions:

In ASCENT, nodes are in one of four states: sleep, passive, test, and active. Fig. 2 shows a state transition diagram. Initially, a random timer turns on the nodes to avoid synchronization. When a node starts, it initializes in the test state. Nodes in the test state exchange data and routing control messages. In addition, when a node enters the test state, it sets up a timer Tt and sends neighbor announcement messages. When Tt expires, the node enters the active state. If, before Tt expires, the number of active neighbors is above the neighbor threshold (NT) or if the average data loss rate (DL) is higher than the average loss before entering in the test state, then the node moves into the passive state. If multiple nodes make a transition to the test state, then we use the node ID in the announcement message as a tie breaking mechanism (higher IDs win). The intuition behind the test state is to probe the network to see if the addition of a new node may actually improve connectivity. When a node enters the passive state, it sets up a timer Tp and sends new passive node announcement messages. This information is used by active nodes to make an estimate of the total density of nodes in the neighborhood. Active nodes transmit this density estimate to any new passive node in the neighborhood. When Tp expires, the node enters the sleep state. If, before Tp expires, the number of neighbors is below NT and either the DL is higher than the loss threshold (LT) or DL is below the loss threshold but the node received a help message from an active neighbor, it makes a transition to the test state. While in passive state, nodes have their radio on and are able to overhear all packets transmitted by their active neighbors. No routing or data packets are forwarded in this state since this is a listen only state. The intuition behind the passive state is to gather information regarding the state of the network without causing interference with the other nodes. Nodes in the passive and test states continuously update the number of active neighbors and data loss rate values. Energy is still consumed in the passive state since the radio is still on when not receiving packets. A node that enters the sleep state turns the radio off, sets a timer Ts, and goes to sleep. When Ts expires, the node moves into passive state. Finally, a node in active state continues forwarding data and routing packets until it runs out of energy. If the data loss rate is greater than LT, the active node sends help messages.

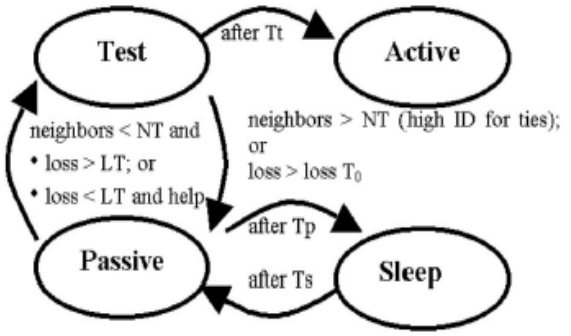


Fig. 2. ASCENT state transitions.

3.2 ASCENT Parameters Tuning:

ASCENT has some parameters that can affect its final behavior. In this section, we explain the choices made in the current ASCENT algorithm. A particular application may select different values for some parameters, for instance, trading energy savings for greater reaction time in case of dynamics. ASCENT also provides adaptive mechanisms to dynamically determine the optimal parameter values. The neighbor threshold (NT) value determines the average degree of connectivity of the network. An application could adjust this value dynamically depending on the events occurring in a certain area of the network, for example, to increase network capacity. In this study, we set this value to 4. The loss threshold (LT) determines the maximum amount of data loss an application can tolerate before it requests help to improve network connectivity. This value is very application dependent. For example, average temperature measurements from a sector of a forest will not tend to vary drastically, and the application may tolerate high packet loss. In contrast, tracking of a moving target by the sensor network may be more sensitive to packet losses. In our implementation, this value was set to 20 percent. The test timer Tt and the passive timer Tp determine the maximum time a node remains in the test and passive states, respectively. They face a similar trade off of power consumption versus decision quality. The larger the timers, the more robust the decision in the presence of transient packet losses (that also affects the neighbor determination), but the greater the power consumed with the radio on, and vice versa. Our work with SCALE [4] has shown that the final determination of these timer values should be dependent on the quality of the reception rate for each link. On the one hand, links that present very high (> 80 percent) or very low (< 20 percent) reception rates show less variability over time and consequently require less time to make an accurate determination of the link quality. On the other hand, links with intermediate reception rates show great variability over time and require more time to make better estimations. We note that it should be possible to design a mechanism that automatically determines the minimum amount of time we should measure the channel to provide some statistical bounds on the accuracy of the link quality estimation, but we have left this as future work. In our implementation, the Tp timer was set to 2 minutes and Tt to 4 minutes. Similarly, the sleep timer Ts represents the amount of time the node sleeps to preserve energy. The larger the Ts

timer, the larger the energy savings, but also the larger the probability of no node in passive state ready to react to dynamics. ASCENT uses an adaptive probabilistic mechanism in order to determine the optimal relationship between the Tp and Ts timers. This mechanism is solely dependent on the average density neighborhood estimate and the probability threshold Pt that a certain k number of nodes in the neighborhood are in the passive state at any given point in time. The details of this mechanism are explained in Section 5.2. In our implementation, the value of k was set to 2 and Pt was set to 95 percent.

3.3 Neighbor and Data Loss Determination:

The number of active neighbors and the average data loss rate are values measured locally by each node while in passive and test state. We have chosen to define a neighbor as a node from which we receive a certain percentage of packets over time. This implies having a history window function (CW) that keeps track of the packets received from each individual node over a certain period (time and/or number of messages) and a fixed or dynamic neighbor loss threshold (NLS). In ASCENT, each node adds a unitary monotonically increasing sequence number to each packet transmitted (including data and control packets). This permits neighbor link loss detection when a sequence number is skipped. In addition, we assume application data packets also have some mechanism to detect losses (data payload sequence numbers in our implementation). Additionally, the final packet loss (or its reciprocal reception rate) estimate from each neighbor node is calculated by using an exponentially weighted moving average (EWMA) of the form:

$$EWMA_{current} = \rho \cdot CW + (1 - \rho) EWMA_{previous}.$$

The value of the filter constant _ was set to 0:b3 3. This local estimate is periodically exchanged between active and test nodes (not passive nodes) by piggybacking this information in data packets or by sending hello packets in the absence of data traffic. The number of active neighbors N is defined as the number of neighbors with link packet loss smaller than the neighbor loss threshold (NLS) and with symmetrical links. In our study, we consider a link symmetrical if it has a difference in reception rate of less than 40 percent between the incoming and outgoing reception rate. We have chosen the following formula NLS:

$$NLS = 1 - \frac{1}{N},$$

with N being the number of neighbors calculated in the previous cycle.

When a node gets a neighbor's packet loss estimate larger than the NLS, it no longer considers that node as a neighbor and deletes it from its neighbor list. The intuition behind this formula is the following: As we increase the number of neighbors in the region, the likelihood of any pair of them not listening to each other (or having high losses) increases. Therefore, as we increase the number of neighbors, we should correspondingly increase the neighbor's loss threshold. Not doing so may result in getting a lower neighbor count even though nodes in the region may still interfere with each other.

4 Performance Evaluation

In this section, we report results from a preliminary performance evaluation of ASCENT. We use simple mathematical models to determine an idealized expected performance of delivery rate, latency, and energy savings as we increase node density. Since our analysis cannot capture the complexity of a full ASCENT scenario, we use simulations and real experiments to further validate the performance evaluation.

4.1 Goals and Metrics:

Our goals in evaluating ASCENT were three-fold: first, in order to validate some of the assumptions made during design of the algorithm; perform analysis, simulations, and real experiments; and conduct comparative performance evaluation of the system with and without ASCENT; second, to understand the energy savings and delivery rate improvements that can be obtained by using ASCENT; finally, to study the sensitivity of ASCENT performance to the choice of parameters.

We choose four metrics to analyze the performance of ASCENT: One-Hop Delivery Rate measures the percentage of

packets received by any node in the network. When all the nodes are turned on—we call this the Active case—the packet reception includes all nodes. In the ASCENT case, it includes all nodes but the ones in the sleep state. This metric indicates the effective one-hop bandwidth available to the nodes in the sensor network. End-to-End Delivery Rate is the ratio of the number of distinct packets received by the destination to the number originally sent by the source. It provides an idea of the quality of the paths in the network, and the effective multi-hop bandwidth. A similar metric has been used in ad hoc routing [3]. Energy Savings is the ratio of the energy consumed by the Active case to the energy consumed by the ASCENT case. This metric defines the amount of energy savings and network lifetime we gain by using the ASCENT algorithm. Finally, Average Per-Hop Latency measures the average delay in packet forwarding in a multi-hop network. It provides an estimate of the end-to-end delay in packet forwarding.

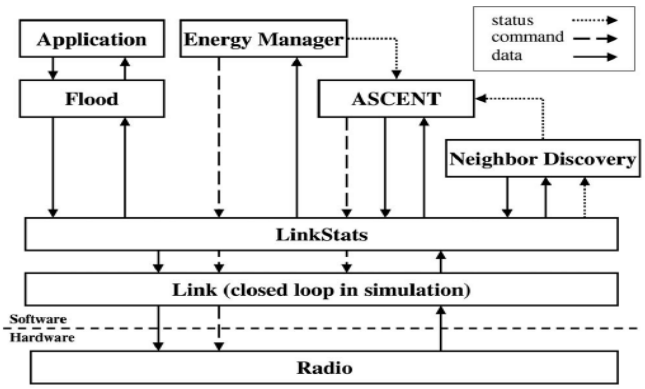


Fig. 3. ASCENT code structure. ASCENT was developed in a modular way so other developers could reuse as much functionality as possible, even when not running the ASCENT topology control algorithm.

4.2 Simulation and Experimental Methodology:

4.2.1 Implementation

ASCENT implementation was developed using the EmStar programming environment [13]. We implemented ASCENT using a number of fine-grained modules. Fig. 3 shows the diagram of the code structure. The first is the LinkStats module, which adds a monotonically increasing sequence number to each packet sent by any module on the node. It monitors packets arriving from other nodes and maintains detailed packet statistics without increasing channel use (but slightly reducing the maximum data payload). This module also implements the exponentially weighted moving average (EWMA) filter for the reception rate of each neighbor. The second module is Neighbor Discovery, which sends and receives heartbeat messages and maintains a list of active neighbors. Third, to evaluate energy usage, we created the Energy Manager module that acts as a simulated battery for each node. It counts packets sent and received, idle time, and radios powering on and off; energy is deducted from an initial supply accordingly. Finally, we created the ASCENT protocol implementation itself, which uses the information provided by the other modules.

4.2.2 Simulator

ASCENT was simulated using the built in simulator (emsim) provided by EmStar [13]. The simulator essentially runs exactly the same code base as the implementation, with no modifications. As in reality, the nodes must interact using their radios and are not allowed to share state directly. Instead of using real radios and sensors, emsim provides a channel simulator that models the behavior of the environment The channel model used in our simulations is a statistical model based on extensive radio connectivity traces gathered when developing earlier versions of ASCENT and SCALE [4]. The simulator is also able to provide CSMA style of collisions.

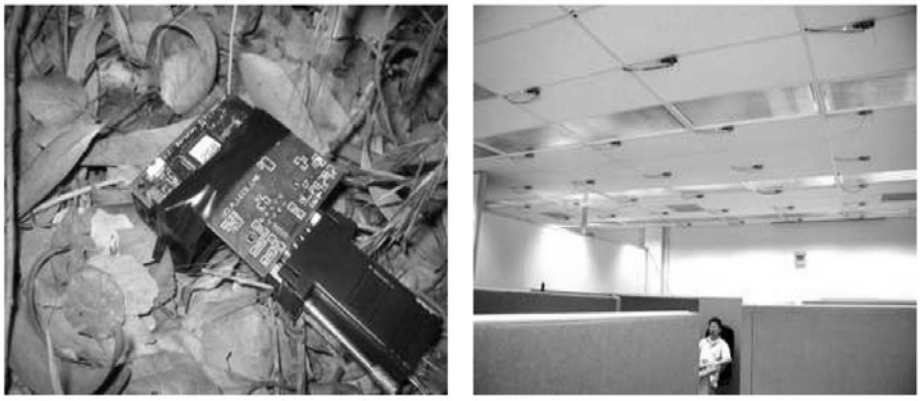
4.2.3 Experimental Test bed

Fig. 4 shows pictures of the hardware components of our Test bed. The ceiling array [13] used in the experiments is composed of various serial port multiplexors attached to a test bed PC. Fig. 4b shows an image of the ceiling array deployed in our lab. We use UTP Cat 5 cables of different lengths (up to 30 meters) and attach one end of the cable to the multiplexor and the other end to a node. A total of 55 nodes are used in the test bed. The nodes are wall powered. Fig. 4a shows a picture of the Mica 1, the node used in the experimental test bed.

5 Conclusion And Future Work

In this paper, we described the design, implementation, analysis, simulation, and experimental evaluation of ASCENT, an adaptive self-configuration topology mechanism for distributed wireless sensor networks. There are many lessons we can draw from our preliminary experimentation. First, ASCENT has the potential for significant reduction of packet loss and increase in energy efficiency. Second, ASCENT mechanisms were responsive and stable under systematically varied conditions.

Furthermore, our paper reports on results from experiments using real radios, demonstrating the importance of self-configuring techniques that react to the operating conditions measured locally. In the near future, we will evaluate the interactions of ASCENT with new MAC mechanisms and the use of robust statistical techniques to improve online link quality estimation. We will also investigate the use of load balancing techniques to distribute the energy load and explore the use of wider area links to detect network partitions. More generally, we are interested in understanding the relationships between topology control mechanisms, like ASCENT, and different routing strategies. This work is an initial foray into the design of self-configuring mechanisms for wireless sensor networks. Our distributed sensing network simulations and experiments represent a nontrivial exploration of the problem space. Such techniques will find increasing importance as the community seeks ways to exploit the redundancy offered by cheap, widely available micro-sensors, as a way of addressing new dimensions of network performance such as network-lifetime.



(a) (b)

Fig. 4. Experimental Testbed. The ceiling array is composed of a PC attached to various serial multiplexors. Several UTP cables run from each multiplexor to the deployment locations in the ceiling where a mote is attached at the end. (a) Mica 1 mote. (b) Indoor office, UCLA CENS lab ceiling array.

APPENDIX

PROOF OF THE PROBABILITY OF K PASSIVE NODES EQUATION

Given a set of n nodes that alternate between passive and sleep states with probabilities given by (6), we would like to find the probability P(k) of at least k passive nodes at any given moment in time:

$$P(\text{at least } k \text{ passive nodes}) = 1 - P(\text{at most } k-1 \text{passive nodes}),$$

for k = 1:

$$P(1) = 1 - P(0 \text{ nodes passive}) = 1 - \left(\frac{1}{\alpha+1}\right)^n,$$

for k = 2:

$$P(2) = 1 - (P(0 \text{ nodes passive}) + P(1 \text{ node passive})) = 1 - \left[\left(\frac{1}{\alpha+1}\right)^n + \left(\frac{1}{\alpha+1}\right)^{n-1} \cdot \left(\frac{\alpha}{\alpha+1}\right)\right],$$

generalizing for any k:

$$\begin{aligned} P(k) &= 1 - (P(0 \text{ nodes passive}) + P(1 \text{ nodes passive}) + \cdots + P(k-1 \text{nodes passive})) \\ &= 1 - \left[\left(\frac{1}{\alpha+1}\right)^n + \left(\frac{1}{\alpha+1}\right)^{n-1} \cdot \left(\frac{\alpha}{\alpha+1}\right) + \cdots + \left(\frac{1}{\alpha+1}\right)^{n-k+1} \cdot \left(\frac{\alpha}{\alpha+1}\right)^{k-1}\right] \\ &= 1 - \left[\left(\frac{1}{\alpha+1}\right)^n \cdot \alpha^0 + \left(\frac{1}{\alpha+1}\right)^n \cdot \alpha^1 + \cdots + \left(\frac{1}{\alpha+1}\right)^n \cdot \alpha^{k-1}\right] \\ &= 1 - \left(\frac{1}{\alpha+1}\right)^n \cdot (\alpha^0 + \alpha^1 + \cdots + \alpha^{k-1}) \\ &= 1 - \left(\frac{1}{\alpha+1}\right)^n \cdot \frac{\alpha^k - 1}{\alpha - 1}. \end{aligned}$$

REFERENCES

[1] K.M. Alzoubi, P.-J. Wan, and O. Frieder, "Message-Optimal Connected-Dominating-Set Construction for Routing in Mobile Ad Hoc Networks," Proc. Third ACM Int'l Symp. Mobile Ad Hoc Networking and Computing (MobiHoc), June 2002.

[2] B. Badrinath, J. Scholtz, M. Srivastava, K. Mills, and V. Stanford, IEEE Personal Comm., special issue on smart spaces and environments, K. Soolins, ed., Oct. 2000.

[3] J. Broch, D.A. Maltz, D.B. Johnson, Y. Hu, and J. Jetcheva, "A Performance Comparison of Multi-hop Wireless Ad Hoc Network Routing Protocols," Proc. Fourth Ann. ACM/IEEE Int'l Conf. Mobile Computing and Networking (MobiCom '98), pp. 85-97, Oct. 1998.

[4] A. Cerpa, N. Busek, and D. Estrin, "SCALE: A Tool for Simple Connectivity Assessment in Lossy Environments," Technical Report CENS Technical Report 0021, Center for Embedded Networked Sensing, Univ. of California, Los Angeles, Sept. 2003.

[5] A. Cerpa, J. Elson, D. Estrin, L. Girod, M. Hamilton, and J. Zhao, "Habitat Monitoring: Application Driver for Wireless Communications Technology," Proc. SIGCOMM Workshop Comm. in Latin America and the Carribean, Apr. 2001.

[6] B. Chen, K. Jamieson, H. Balakrishnan, and R. Morris, "Span: An Energy-Efficient Coordination Algorithm for Topology Maintenance in Ad Hoc Wireless Networks," Proc. Seventh Ann. ACM/ IEEE Int'l Conf. Mobile Computing and Networking (MobiCom), pp. 85-96, July 2001.

[7] D. Estrin, R. Govindan, J. Heidemann, and S. Kumar, "Next Century Challenges: Scalable Coordination in Sensor Networks," Proc. Fifth Ann. ACM/IEEE Int'l Conf. Mobile Computing and Networking (MobiCom), pp. 263-270, Aug. 1999.

[8] Comm. ACM, special issue on embedding the internet, D. Estrin, R. Govindan, and J. Heidemann, eds., vol. 43, no. 5, May 2000

[9] S. Floyd, V. Jacobson, C.-G. Liu, S. McCanne, and L. Zhang, "A Reliable Multicast Framework for Light-Weight Sessions and Application Level Framing," Proc. ACM Special Interest Group on Data Comm. (SIGCOMM), pp. 342-356, Aug. 1995.

[10] D. Ganesan, B. Krishnamachari, A. Woo, D. Culler, D. Estrin, and S. Wicker, "Complex Behavior at Scale: An Experimental Study of Low-Power Wireless Sensor Networks," Technical Report UCLA CSD-TR 02-0013, Center for Embedded Networked Sensing, Univ. of California, Los Angeles, and Intel Research Lab, Univ. of California, Berkeley, Feb. 2002.

[11] J. Gao, L.J. Guibas, J. Hershburger, L. Zhang, and A. Zhu, "Geometric Spanner for Routing in Mobile Networks," Proc. Second ACM Symp. Mobile Ad Hoc Networking and Computing (MobiHoc 01), pp. 45-55, Oct. 2001.

[12] J. Gao, L.J. Guibas, J. Hershburger, L. Zhang, and A. Zhu, "Discrete and Computational Geometry," Proc. Second ACM Symp. Mobile Ad Hoc Networking and Computing (MobiHoc 01), vol. 30, no. 1, pp. 45-65, 2003.

[13] L. Girod, J. Elson, A. Cerpa, N. Ramanathan, T. Stathopoulos, and D. Estrin, "Emstar: A Software Environment for Developing and Deploying Wireless Sensor Networks," Proc. 2004 USENIX Technical Conf., June-July 2004.

[14] C. Intanagonwiwat, R. Govindan, and D. Estrin, "Directed Diffusion: A Scalable and Robust Communication Paradigm for Sensor Networks," Proc. Sixth Ann. ACM/IEEE Int'l Conf. Mobile Computing and Networking (MobiCom), Aug. 2000.