

# A Scalable Scheme for Preventing Feedback Implosion in a Large-Scale Multi-Tier Sensor Network

Reuven Cohen and Alexander Landau  
Dept. of Computer Science  
Technion  
Israel

**Abstract**—We consider a huge hierarchical sensor network consisting of millions of sensors arranged in clusters for scalability and cost-performance. We address the problem of how a centralized gateway can estimate the number of sensors affected by a certain event. We propose a scheme for solving this problem in the most efficient way in terms of communication cost, and a complete mathematical analysis of the estimation error. We show that the error of the new scheme is very small even if the number of sensors experiencing an event is several million.

## I. INTRODUCTION

In sensor networks, sensors probe the surrounding environment and generate reports of the collected readings. Using wireless communication, these reports are sent to a control center, usually through a gateway deployed in the physical proximity of the sensors [21]. Although one possible approach for designing a sensor application is to deploy homogeneous sensors and program each one to perform all possible application tasks, this yields a flat, non-scalable network of homogeneous nodes. An alternative, multi-tier approach is to use heterogeneous elements. In this approach, resource-constrained, low-power elements are in charge of performing the simple tasks, such as detecting scalar physical measurements, while stronger devices, called gateways, perform complex tasks such as routing and resource management [1]. A large-scale cost-effective sensor network that consists of millions of nodes can be realized only by means of a multi-tier wireless network, where every tier employs a different wireless technology. Such an architecture allows the sensors to be controlled and alarms received quickly and reliably.

As an example, consider a 4-tier, large-scale sensor network. The bottom tier of such a network consists of millions of sensors deployed throughout a country or a continent and divided into areas. WiFi technology is used to give each sensor direct wireless connectivity with an area gateway. There are thousands of such gateways, divided into regions that are controlled by regional gateways. The area gateways can communicate with their regional gateways using WiMax. Finally, the regional gateways are connected to a centralized root gateway by means of a satellite channel.

This architecture has several advantages compared to a flat network with homogeneous devices [12], the most important

of which are as follows:

- The sensors do not need to participate in a routing protocol, which would require them to expend a lot of energy because of the network size [19].
- The routing of a message from a sensor to the root gateway is fast, reliable, and simple. In the 4-tier model considered above, the messages go through three broadcast domains: from a sensor to an area gateway, then to a regional gateway, and finally to the root gateway. If the various gateways are connected to a permanent power supply, they do not need to switch their communication module on and off. Hence, it takes less than a second for a message to reach the root gateway [24].
- This is probably the only architecture that can be used for a huge sensor network consisting of millions of devices and covering a gigantic area [2], without being restricted by topography, weather, and other constraints.

This paper addresses one of the most important problems in such a multi-tier, gigantic sensor network: collecting important real-time information from the sensors in a scalable way while preventing feedback implosion. In many sensor network applications, the root gateway needs to know not only about the occurrence of specific events, but also about their scale. For instance, the gateway is likely to be interested not only in the detection of a sudden temperature increase by the sensors, but also in the number of sensors experiencing this event. Having every sensor notify the gateway would result in feedback implosion due to the huge number of sensors. The scheme proposed in this paper informs the root gateway of the number of sensors experiencing a given event without requiring each of them to send its own notification message.

The main idea behind the proposed scheme is that only a small fraction of the sensors notify their area gateways, which then forward only some of these notification messages to the upper-level gateways. Thus, only a few messages (less than 10) arrive at the root gateway, which then processes them with our estimation algorithm and gets a precise estimate of the number of sensors experiencing the event. We show that the error of the new scheme is very small even if the number of nodes experiencing an event is several million. Moreover, we show

that the communication cost of the proposed scheme is the minimum one could achieve, because each gateway needs to send only 2 messages.

The rest of this paper is organized as follows. In Section II we present related work. In Section III we describe an existing scheme and extend it for a hierarchical sensor network. In Section IV we present a new scheme, analyze it, and show that it can reduce the communication cost by a factor of 10. In Section V we present simulation results and in Section VI we conclude the paper.

## II. RELATED WORK

In this section we present works related to the considered sensor network architecture and works related to real-time estimation schemes.

Many papers develop and study routing problems for hierarchical sensor networks. One of the two most common approaches is to designate some of the sensors as cluster heads, through which messages from the rest of the sensors will be routed to a central gateway. LEACH [8] selects these cluster heads randomly and periodically rotates this role. PEGASIS [14] builds paths from the sensors to the central gateway, as routing is done in flat sensor networks [9].

The multi-tier approach considered in this paper is different, and is more suitable for very large scale sensor networks. Here we assume that strong gateways, which are connected to a continuous power source, are located in advance in the same way that cellular base-stations are located in a cellular network. In [22], the authors provide a time synchronization algorithm for such networks. In [11], the authors employ a hierarchical sensor network for a video over IP application. In this application, home sensors detect user presence and send report messages via intermediate access points to an area head end. The latter is in charge of starting, halting, and prioritizing streaming information originating at every camera.

Habitat monitoring can also take advantage of hierarchical sensor networks. In [15], the authors spread sensors on Great Duck Island that measure environmental characteristics such as temperature, pressure and humidity. The authors of [18] extend this idea and provide fault tolerance and distributed storage in their sensor network.

In [20] the authors address the problem of gateway positioning in a hierarchical sensor network in order to increase sensor lifetime. A 3-tier hierarchical sensor network built using off-the-shelf hardware and software is presented in [5]. A model for hierarchical in-network aggregation in sensor networks is presented in [6].

A common criterion for a sensor to send its measurement is the fulfillment of a certain condition or event regarding the measured attribute. For example, sensors measuring temperature in some geographical region may send messages to a central gateway once the temperature exceeds a given threshold. These messages may be synchronous or asynchronous. In the asynchronous model [16], [17], whenever a certain condition or event takes place, a sensor sends a message. In the synchronous model [23], the central gateway or cluster head

periodically polls the sensors, and those experiencing the event send a reply.

While receiving messages from all the sensors experiencing a given event may be useful in certain circumstances, it may result in feedback implosion if a large group of sensors send their messages simultaneously. One solution to this problem is data aggregation and directed diffusion [10], in which sensors or intermediate gateways merge received messages and send the combined message to the central gateway. Our paper addresses this problem in a different way, by asking the reporting sensors to wait a random time before sending their messages, and by allowing only the first sensors to send their reports. Using an estimation scheme like the one proposed in [7] for a flat wireless network, a gateway can estimate the actual number of sensors experiencing the considered event in its region. We elaborate on this scheme in Section III.

Other estimation schemes were optimized for different application scenarios. In [3], [4] the authors estimate the number of receivers in a multicast group. To avoid feedback implosion, not all the receivers send a message to the sender. Rather, each one sends a message with a predefined probability  $p$ . The sender uses  $p$  and other parameters to estimate the number of receivers.

A timer based scheme for estimating the number of hosts contending for access to a shared ALOHA channel is given in [13]. The scheme is derived from the probability  $p_{f|s}$  of a successful access during the first transmission given a successful delivery of the packet. To this end, each contending host indicates in every transmitted packet whether this packet is being transmitted for the first time, or whether it has already experienced a collision.

## III. A HIERARCHICAL IMPLEMENTATION OF NATO!

In [7], the authors propose a scheme called NATO! for estimating the number of affected nodes in a 1-tier (flat) network. In this section we summarize this scheme and then show how it can be extended for a multi-tier network.

The scheme is optimized for a flat network where a gateway is able to broadcast a message to all the nodes, and the delays between the nodes and the gateway are equal. According to this scheme, the gateway broadcasts a  $\text{START}(t_0)$  message indicating that after time  $t_0$  it wishes to receive RPRT messages from a subset of the affected nodes. The affected nodes are those that experience the considered event. Every affected node chooses a random timer in the range  $[t_0, t_0 + T]$ , using a known probability distribution function  $F$ , and if when the timer expires, the other affected nodes have sent fewer than  $N$  RPRTs to the gateway, the considered node should send a RPRT.

As a result of this scheme, the gateway receives  $N$  RPRTs during  $[t_0, t_0 + T]$ . It then executes the following algorithm.

*Algorithm 1:* (the algorithm developed in [7] for the root gateway to estimate the number  $r$  of affected nodes from the first  $N$  RPRTs)

- A) Receive the first  $N$  RPRT messages at times  $x_1 < \dots < x_N$ .

- B) Using the Newton-Raphson method, find the greatest root  $r$  of the following equation:

$$\frac{1}{r} + \frac{1}{r-1} + \dots + \frac{1}{r-N+1} + \ln(1 - F(x_N)) = 0. \quad (1)$$

- C) The estimated number of affected nodes is  $r \cdot \frac{N-1}{N}$ .

It is shown in [7] that in a flat network (one broadcast domain) with thousands of affected nodes this algorithm yields an estimation error of 1%.

We now modify this scheme to fit a hierarchical sensor network. We consider a logical tree structure with  $L$  levels of gateways. The root is referred to as a Level- $L$  gateway and it has direct wireless connectivity with multiple Level- $(L-1)$  gateways. The latter have direct wireless connectivity with Level- $(L-2)$  gateways, and so on. At the bottom level, each Level-1 gateway has direct wireless connectivity with the sensors. Figure 1 shows an example for  $L = 2$ . Some of the sensors experience an event. Recall that these sensors are also referred to as ‘‘affected nodes.’’

Scheme 1, presented below, implements NATO! hierarchically. The main idea behind this scheme is that a Level- $i$  gateway relays towards the root the first  $N$  RPRTs it receives from its Level- $(i-1)$  gateways.

*Scheme 1:* (using NATO! hierarchically)

- A) The root gateway broadcasts a START( $t_0$ ) message, where  $t_0$  is the time after which RPRT messages can be sent from the affected nodes to the Level-1 gateways.
- B) At time  $t_0$ , each affected node  $v$  performs the following steps:
- B1) It chooses a random time  $t$  in the range  $[t_0, t_0 + T]$ , using a known probability distribution function  $F$ .
- B2) If the number of RPRTs sent before  $t$  by the other affected nodes to the gateway of  $v$  is smaller than  $N$ ,  $v$  sends a RPRT to this gateway. Otherwise, it does not send a RPRT.
- C) For every level  $i$ : By time  $t_0 + i \cdot T$ , every Level- $i$  gateway has the  $N$  times  $x_1, \dots, x_N$  associated with the first  $N$  RPRTs it has received from its downstream gateways (if  $i > 1$ ) or affected nodes (if  $i = 1$ ). For every  $j$  from 1 to  $N$ , if the number of RPRTs sent to its parent Level- $(i+1)$  gateway before  $t_0 + i \cdot T + x_j$  is smaller than  $N$ , the considered Level- $i$  gateway should send to its parent gateway a RPRT at time  $t_0 + i \cdot T + x_j$ .

In Scheme 1, by time  $t_0 + nT$ , the root gateway knows the transmission time of the first  $N$  RPRTs sent by the affected nodes and it can run Algorithm 1 to estimate the number of affected nodes.

Scheme 1 requires each Level- $i$  gateway to receive  $N$  RPRTs. This is in addition to 2 control messages, START and STOP, sent by each gateway. Thus, the communication cost

of the scheme is equal to  $|G|(N+2)$ , where  $|G|$  is the total number of gateways. Because we do not make any assumption regarding the distribution of affected nodes in the network, any estimation scheme must involve all the gateways. Moreover, in every scheme every gateway must receive a message when the scheme is initiated by the root and send a message after it acquires data. Hence, we believe that the minimum number of messages requires by any scheme is  $3|G| + C$ , where  $C$  is a constant that does not depend on the network structure. In the next section we present a new scheme that achieves this lower bound.

#### IV. THE H-NATO! SCHEME AND ITS ANALYSIS

##### A. The New Scheme

In this section we propose an alternative scheme that significantly reduces the communication cost without affecting the precision. The main idea behind this scheme is as follows. The root gateway needs to get the  $N$  smallest reporting times of the network nodes. However, this requirement should not necessarily be translated into the requirement that every Level- $i$  gateway will get the  $N$  smallest reporting times of affected nodes in its subtree. If the number of Level- $i$  gateways is much larger than  $N$ , there is a very small probability that several RPRTs of affected nodes from the subtree of the same Level- $i$  gateway will fall into the list of the  $N$  smallest RPRT transmission times to be received by the Level- $(i+1)$  gateway. Therefore, it might be sufficient for every Level- $i$  gateway, where  $1 \leq i \leq L-1$ , to get only the  $N_i < N$  smallest reporting times in its subtree. In such a case, the communication cost of the scheme is  $N + \sum_{i=1}^{L-1} N_i \cdot B_i$ , where  $B_i$  is the number of Level- $i$  gateways.

This idea is translated into the following scheme:

*Scheme 2:* (H-NATO!)

- A) The root gateway broadcasts a START( $t_0$ ) message, where  $t_0$  is the time after which RPRT messages can be sent from the affected nodes to their Level-1 gateways.
- B) At time  $t_0$ , each affected node  $v$  performs the following steps:
- B1) It chooses a random time  $t$  in the range  $[t_0, t_0 + T]$ , using a known probability distribution function  $F$ .
- B2) If the number of RPRTs sent before  $t$  to the Level-1 gateway of  $v$  is smaller than  $N_1$ , then  $v$  should send a RPRT to this gateway. Otherwise, it should not send a RPRT.
- C) For every level  $i \leq L-1$ : By  $t_0 + i \cdot T$ , every Level- $i$  gateway  $g$  has received the  $N_i$  times  $x_1, \dots, x_{N_i}$  associated with the  $N_i$  first RPRTs it has received from its downstream gateways (if  $i > 1$ ) or affected nodes (if  $i = 1$ ). For every  $1 \leq j \leq N_i$ , if the number of RPRTs sent to the parent Level- $(i+1)$  gateway of  $g$  before  $t_0 + i \cdot T + x_j$  is smaller than  $N_{i+1}$ ,  $g$  sends to its parent Level- $(i+1)$  gateway a RPRT at  $t_0 + i \cdot T + x_j$ .

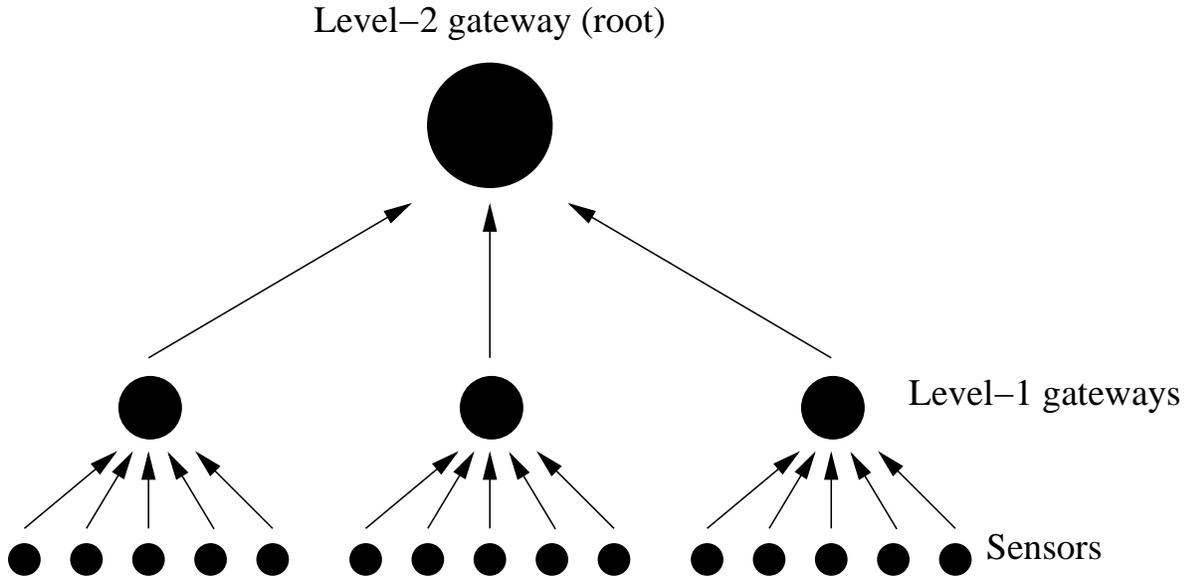


Fig. 1. An example of a 3-tier sensor network hierarchy

- D) The root gateway uses the  $N$  RPRTs it receives as input to Algorithm 2, to be described later, in order to estimate the number of affected nodes.

**Implementation Notes:**

- 1) In both Scheme 1 and Scheme 2, it was assumed that all gateways and sensors are synchronized to a common clock or that all gateways and sensors of the same level receive the START message at the same time. Such an assumption can be easily realized in the considered wireless networks (WiFi, WiMax, satellite, etc.). However, H-NATO! can also be executed in a system where there is no common clock and there is a variable delay between a gateway and its downstream gateways or sensors, provided that each gateway knows the delay  $D_j$  to/from every downstream gateway or sensor  $j$ , and that  $j$  knows the delay to its parent gateway. In such a system the following adjustment should be used. Let  $D_j$  be the delay from a Level- $i$  gateway to node  $j$ , and let  $D = \max_j(D_j)$  be the maximum delay. The Level- $i$  gateway adds the value of  $D$  to the START message it broadcasts, and each node  $j$  should wait a time period of  $D - D_j$  before running Scheme 2.
- 2) RPRT messages are subject to loss due to collisions on a contention channel or to transmission errors. To address this problem, we enhance Scheme 2 with the following reliability mechanism. Upon receiving a RPRT, a Level- $i$  gateway sends a confirmation to the sender. A RPRT sender that does not receive a confirmation within a time-out period resends its RPRT, and specifies the offset between its current local time and the time of the original RPRT.

Figure 2 depicts the new scheme for 2 levels, when  $N_1 = 1$ . In what follows we concentrate on the case where  $N_i = 1$

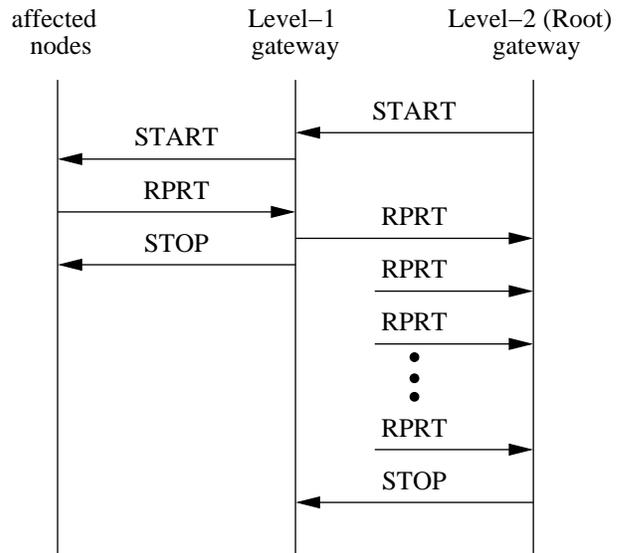


Fig. 2. An example of the new scheme with two levels

for  $1 \leq i \leq L - 1$ . In such a case this scheme reduces the communication cost of Scheme 1 by a factor of  $N$ . Moreover, the communication cost for the new scheme with  $N_i = 1$  for  $1 \leq i \leq L - 1$  is probably *the best one may achieve*, because in such a case every gateway is required to send approximately two messages, i.e., a START and a STOP.

We are now looking for an estimation algorithm to be used by the root gateway in step (D) of Scheme 2. With high probability, even if  $N_i = 1$  for every  $i \leq L - 1$ , the root receives the  $N$  almost smallest RPRT times sent by the affected nodes. This may suggest that Algorithm 1 can be used in this case as well. In Figure 3 we see the estimation error as a function of  $N$  when running Scheme 2 with Algorithm

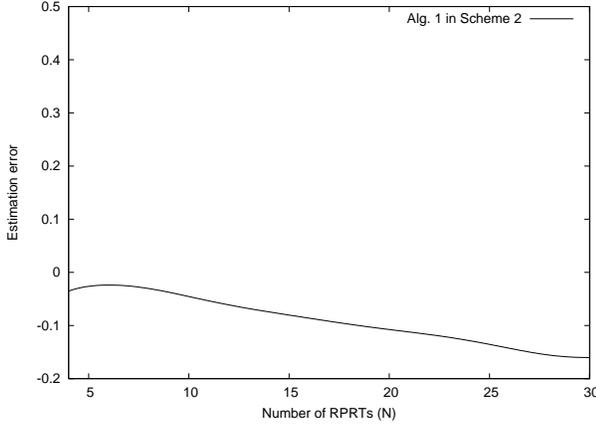


Fig. 3. Estimation error vs.  $N$  when Algorithm 1 is used in Scheme 2

1 while  $N_i = 1$  for every  $i \leq L - 1$ . The estimation error is defined by  $\frac{r_{estimated} - r_{real}}{r_{real}}$ , where  $r_{estimated}$  is the estimated number of affected nodes, and  $r_{real}$  is the real number. For this graph we simulated a network with two levels of gateways: 1 root and 100 Level-1 gateways. Each Level-1 gateway has a random number of affected nodes, uniformly distributed in the range  $[1, 1000]$ . We can see that the error is significant, and that it does not converge to 0 when  $N$  increases. There are two reasons for this error. First, each Level-1 gateway may have a different number of affected nodes. Thus, the transmission times of the RPRTs sent by these gateways to the root are not from the same distributions. Algorithm 1, however, requires that all affected nodes use the same (known) distribution when deciding the transmission time of their RPRT message [7]. Second, the  $N$  RPRTs received by the root are not guaranteed to be those with the earliest transmission time. For example, if an affected node of the first Level-1 gateway sends the first RPRT and another affected node of the same gateway sends the next RPRT, this second RPRT is not forwarded by the Level-1 gateway to the root. Consequently, the input to the algorithm is not accurate.

### B. A New Estimation Algorithm

We now develop a new estimation algorithm to be executed by the root in Scheme 2. We assume that the root knows the number  $B$  of Level- $(L - 1)$  gateways, the number  $N$  of RPRTs it receives, and the times these RPRTs were sent by the affected nodes. The algorithm requires that each Level- $(L - 1)$  gateway send at most one RPRT to the root, containing the smallest RPRT time sent by all affected nodes in its subtree. This means that for every  $1 \leq i \leq L - 1$ ,  $N_i = 1$ . Recall that with these values, the communication cost of Scheme 2 is minimized.

Let  $F$  be a uniform distribution function over the interval  $[0, 1]$ . Thus, for every  $x \in [0, 1]$ ,  $F(x) = x$  and  $f(x) = 1$ . Let  $B$  be the number of Level- $(L - 1)$  gateways. Let random variable  $X_i$  be the time a Level- $(L - 1)$  gateway  $i$  receives a RPRT from its descendants, where  $i = 1, \dots, B$ . Let  $Y_1, \dots, Y_B$  be the sequence  $X_1, \dots, X_B$  sorted in ascending

order. Thus, the times during which the root gateway receives the  $N$  RPRTs are  $Y_1, \dots, Y_N$ . Let  $m_i$  be the total number of affected nodes in the subtree of gateway  $i$ . Since  $X_i$  is the minimum of  $m_i$  independent uniform random variables, we get:

$$P(X_i > x) = P(U > x)^{m_i} = (1 - x)^{m_i},$$

where  $U$  is a uniform random variable on  $[0, 1]$ .

Therefore,

$$F_{X_i}(x) = P(X_i \leq x) = 1 - (1 - x)^{m_i}$$

and

$$f_{X_i}(x) = \frac{dF_{X_i}}{dx}(x) = m_i \cdot (1 - x)^{m_i - 1}.$$

The joint density of  $Y_1, \dots, Y_N$  is:

$$\begin{aligned} & f_{Y_1, \dots, Y_N}(y_1, \dots, y_N) dy_1 \cdots dy_N \\ &= P(\text{one } X_i \text{ is in } (y_1, y_1 + dy_1), \dots, \\ & \quad \text{one } X_i \text{ is in } (y_N, y_N + dy_N), \\ & \quad \text{all other } X_i \text{'s are in } (y_N, 1)) \\ &= \sum_{(i_1, \dots, i_N) \in I} P(X_{i_1} \in (y_1, y_1 + dy_1)) \cdots \cdots \\ & \quad \cdot P(X_{i_N} \in (y_N, y_N + dy_N)) \\ & \quad \cdot P((B - N) \text{ other } X_i \text{'s are in } (y_N, 1)) \\ &= \sum_{(i_1, \dots, i_N) \in I} m_{i_1} (1 - y_1)^{m_{i_1} - 1} dy_1 \cdots \cdots \\ & \quad \cdot m_{i_N} (1 - y_N)^{m_{i_N} - 1} dy_N \\ & \quad \cdot (1 - y_N)^{\sum_{j \in \{1, \dots, B\} - \{i_1, \dots, i_N\}} m_j}, \end{aligned}$$

where

$$I = \{(i_1, \dots, i_N) \mid \forall j : 1 \leq i_j \leq B \wedge \forall j, k : i_j \neq i_k\}$$

is the set of  $N$ -tuples of pairwise different indexes in the range  $1, \dots, B$ .

Therefore,

$$\begin{aligned} & f_{Y_1, \dots, Y_N}(y_1, \dots, y_N) \\ &= \sum_{(i_1, \dots, i_N) \in I} m_{i_1} (1 - y_1)^{m_{i_1} - 1} \cdots \cdots \\ & \quad \cdot m_{i_N} (1 - y_N)^{m_{i_N} - 1} \\ & \quad \cdot (1 - y_N)^{\sum_{j \in \{1, \dots, B\} - \{i_1, \dots, i_N\}} m_j}. \end{aligned}$$

We now use the maximum likelihood method to find the values of  $m_1, \dots, m_N$  that maximize  $f_{Y_1, \dots, Y_N}(y_1, \dots, y_N)$ , where  $y_1, \dots, y_N$  are the times when the  $N$  RPRTs are received by the root. First, note that  $f_{Y_1, \dots, Y_N}(y_1, \dots, y_N)$  is a symmetric function of  $m_1, \dots, m_N$ . In other words, any permutation of  $m_1, \dots, m_N$  yields the same value of  $f$ . Therefore, the maximum of  $f$  is achieved for  $m_1 = \dots = m_N$ . The

intersection of  $f$  with the plane defined by  $m_1 = \dots = m_N$  in the  $(N + 1)$ -dimensional space is a function

$$g(m) = \sum_{(i_1, \dots, i_N) \in I} m(1 - y_1)^{m-1} \cdot \dots \cdot m(1 - y_N)^{m-1} \cdot (1 - y_N)^{\sum_{j \in \{1, \dots, B\} - \{i_1, \dots, i_N\}} m},$$

whose maximum coincides with the maximum of  $f$ . Thus, we should find the value of  $m$  that maximizes  $g(m)$ . To this end, we now rewrite  $g(m)$  in the following way:

$$\begin{aligned} g(m) &= \sum_{(i_1, \dots, i_N) \in I} m(1 - y_1)^{m-1} \cdot \dots \cdot m(1 - y_N)^{m-1} \cdot (1 - y_N)^{(B-N)m} \\ &= |I| \cdot m(1 - y_1)^{m-1} \cdot \dots \cdot m(1 - y_N)^{m-1} \cdot (1 - y_N)^{(B-N)m} \\ &= \frac{B!}{(B-N)!} \cdot m^N \cdot [(1 - y_1) \dots (1 - y_N)]^{m-1} \cdot (1 - y_N)^{(B-N)m}. \end{aligned}$$

Since  $\ln(\cdot)$  is a monotonically increasing function,  $g(m)$  gets its maximum at the same  $m$  as  $\ln g(m)$ .

$$\begin{aligned} \ln g(m) &= \ln \frac{B!}{(B-N)!} + N \ln m \\ &\quad + (m-1) \ln[(1 - y_1) \dots (1 - y_N)] \\ &\quad + (B-N)m \ln(1 - y_N). \end{aligned}$$

We now compute the derivative:

$$(\ln g(m))' = \frac{N}{m} + \ln[(1 - y_1) \dots (1 - y_N)] + (B-N) \ln(1 - y_N).$$

Equating  $(\ln g(m))'$  to 0 and solving for  $m$  yields  $m$  equal to

$$\frac{N}{\ln[(1 - y_1) \dots (1 - y_N)] + (B-N) \ln(1 - y_N)}. \quad (2)$$

Since  $(\ln g(m))'' = -\frac{N}{m^2} < 0$  holds for every  $m$ , the above value is indeed a maximum, and the total number  $r$  of affected nodes estimated by the root is

$$\frac{N \cdot B}{\ln[(1 - y_1) \dots (1 - y_N)] + (B-N) \ln(1 - y_N)}. \quad (3)$$

### C. The Error of Eq. 3

We will now find the relative error of Eq. 3. Let random variable  $R$  be the result of substituting  $Y_i$  for  $y_i$  in Eq. 3, and let random variable  $M$  be the result of the same substitution in Eq. 2. Thus, the estimation error is

$$\frac{E(R) - r}{r} = \frac{B \cdot E(M) - Bm}{Bm} = \frac{E(M) - m}{m},$$

where

$$\begin{aligned} E(M) &= \int_{0 < y_1 < \dots < y_N < 1} \frac{-dy_1 \dots dy_N}{N} \\ &\quad \cdot \frac{1}{\ln[(1 - y_1) \dots (1 - y_N)] + (B-N) \ln(1 - y_N)} \\ &\quad \cdot \frac{B!}{(B-N)!} \cdot m^N \cdot [(1 - y_1) \dots (1 - y_N)]^{m-1} \\ &\quad \cdot (1 - y_N)^{(B-N)m}. \end{aligned}$$

Substituting  $x_i = 1 - y_i$  yields:

$$\begin{aligned} E(M) &= -m^N N \frac{B!}{(B-N)!} \int_{0 < x_N < \dots < x_1 < 1} \frac{(\prod_{i=1}^N x_i)^{m-1} \cdot x_N^{(B-N)m}}{\ln[\prod_{i=1}^{N-1} x_i \cdot x_N^{B-N+1}]} dx_1 \dots dx_N \\ &= -m^N N \frac{B!}{(B-N)!} \int_{0 < x_N < \dots < x_1 < 1} \frac{(\prod_{i=1}^N x_i \cdot x_N^{B-N})^{m-1} \cdot x_N^{B-N}}{\ln[\prod_{i=1}^{N-1} x_i \cdot x_N^{B-N+1}]} dx_1 \dots dx_N. \end{aligned}$$

Define  $t_1 = \prod_{i=1}^N x_i \cdot x_N^{B-N} = \prod_{i=1}^{N-1} x_i \cdot x_N^{B-N+1}$ ,  $t_2 = x_2, \dots, t_N = x_N$ . Then,  $x_1 = \frac{t_1}{\prod_{i=2}^N t_i \cdot t_N^{B-N}}$ . The Jacobian matrix  $\frac{\partial(x_1, \dots, x_N)}{\partial(t_1, \dots, t_N)}$  is an upper triangular matrix with the values  $\frac{\partial x_1}{\partial t_1}, \dots, \frac{\partial x_N}{\partial t_N}$  on its diagonal. For  $i \geq 2$ ,  $\frac{\partial x_i}{\partial t_i} = 1$  and  $\frac{\partial x_1}{\partial t_1} = \frac{1}{\prod_{i=2}^N t_i \cdot t_N^{B-N}}$ . Therefore, the determinant of this matrix (the Jacobian) is  $J = \frac{1}{\prod_{i=2}^N t_i \cdot t_N^{B-N}}$ .

Thus,

$$\begin{aligned} E(M) &= -m^N N \frac{B!}{(B-N)!} \cdot \\ &\quad \int_D \frac{t_1^{m-1}}{\ln t_1 \cdot \prod_{i=2}^N t_i} dt_1 \dots dt_N, \end{aligned}$$

where  $D$  is a domain in  $\mathbb{R}^N$  defined by  $D = \{(t_1, \dots, t_N) | 0 < t_N < \dots < t_2 < \frac{t_1}{\prod_{i=2}^{N-1} t_i \cdot t_N^{B-N+1}} < 1\}$ .

We solve this integral by iterated integration with the following bounds.

If  $N = 2$ ,

$$\begin{aligned} 0 &< t_1 < 1 \\ B\sqrt[B]{t_1} &< t_2 < \sqrt[B]{t_1}. \end{aligned}$$

If  $N \geq 3$ ,

$$\begin{aligned} 0 &< t_1 < 1 \\ B\sqrt[B]{t_1} &< t_2 < 1 \\ B\sqrt[B]{\frac{t_1}{t_2}} &< t_3 < t_2 \\ &\dots \\ B\sqrt[B]{\frac{t_1}{t_2 \dots t_{N-2}}} &< t_{N-1} < t_{N-2} \\ B\sqrt[B]{\frac{t_1}{t_2 t_3 \dots t_{N-1}}} &< t_N < B\sqrt[B]{\frac{t_1}{t_2^2 t_3 \dots t_{N-1}}}. \end{aligned}$$

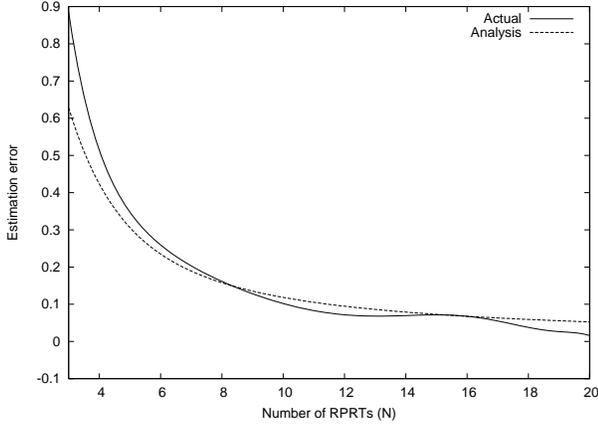


Fig. 4. The estimation error of Eq. 3 (actual error vs. our analysis)

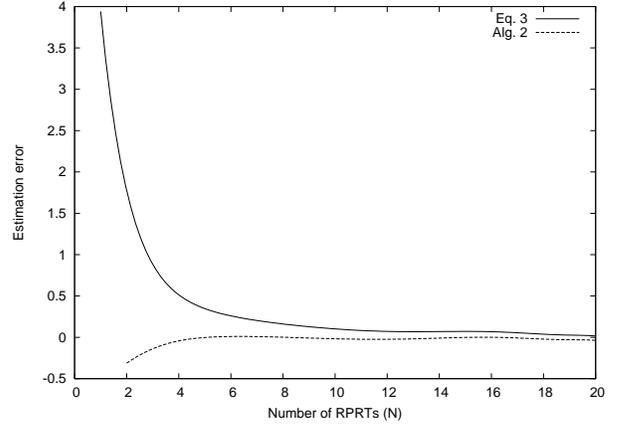


Fig. 5. The estimation error of Eq. 3 and of Algorithm 2

We can show analytically that for every practical value of  $N$  (e.g.,  $N \leq 15$ ):

$$\int_D \frac{t_1^{m-1}}{\ln t_1 \cdot \prod_{i=2}^N t_i} dt_1 \dots dt_N = \begin{cases} -\frac{1}{mB(B-1)} & N = 2 \\ -\frac{1}{(N-1) \cdot m^{N-1} \cdot (B-1)^2(B-2) \dots (B-N+1)} & N \geq 3 \end{cases}.$$

Therefore,

$$E(M) = \begin{cases} 2m & N = 2 \\ m \frac{N}{N-1} \frac{B}{B-1} & N \geq 3 \end{cases}.$$

Thus, the estimation error is

$$\frac{E(M)}{m} - 1 = \begin{cases} 1 & N = 2 \\ \frac{N}{N-1} \frac{B}{B-1} - 1 & N \geq 3 \end{cases}. \quad (4)$$

When  $B$  is large,  $\frac{B}{B-1} = 1 + \frac{1}{B-1} \approx 1$ . In this case the error is  $\frac{N}{N-1} - 1 = \frac{1}{N-1}$ .

In Figure 4 we present the results for implementing Scheme 2 with Eq. 3 in a 2-level network with  $B = 200$  Level-1 gateways. Each Level-1 gateway has a random number  $m$  of affected nodes, where  $m$  is uniformly distributed in the range  $[1, 1000]$ . The graph shows two curves: the upper one shows the actual estimation error while the lower one shows the error found by our analysis (Eq. 4). The  $x$ -axis is the number  $N$  of RPRTs received by the root and the  $y$ -axis is the estimation error. We see an excellent agreement of the curves for  $N \geq 3$ .

Since the error is always positive, we can significantly improve the precision of our estimation algorithm by multiplying the value found by Eq. 3 by  $\frac{N-1}{N}$  (as proposed in [7] for Algorithm 1). To summarize, the algorithm to be employed by the root in Scheme 2 is as follows:

*Algorithm 2:*

- 1) Receive  $N$  RPRTs from the Level- $(L-1)$  gateways at times  $y_1 < \dots < y_N$ .
- 2) Calculate the number of affected nodes by Eq. 3 and multiply the result by  $\frac{N-1}{N}$ .

## V. SIMULATION STUDY

We simulated a 2-gateway-level network with  $B = 200$  Level-1 gateways, each with a random number  $m$  of affected nodes, uniformly distributed in the range  $[1, 1000]$ . We implemented Scheme 2 with Algorithm 2 (i.e., Eq. 3 with error cancellation) and calculated the estimation error given by  $\frac{T_{estimated} - T_{real}}{T_{real}}$ .

Figure 5 depicts two curves. The top one is the estimation error before applying error cancellation (i.e., using Eq. 3), and the bottom one is the actual error of Algorithm 2 after multiplying Eq. 3 by  $\frac{N-1}{N}$ . As we can see, the error of Algorithm 2 is very close to 0 for  $N \geq 5$ , whereas the curve of Eq. 3 tends to 0 only for much greater values of  $N$ .

Figure 6 compares the estimation error for  $B = 100$  and for  $B = 500$  Level-1 gateways as a function of  $N$ . We see that for both cases, the error is within the 2-3% range, and that  $B$  has no effect on the error, as predicted by Eq. 4.

One of the most important properties of the proposed scheme is that it works very well even if the reporting nodes are not evenly distributed in the network. It is not difficult to see that with a non-uniform distribution it is impossible to estimate the number of affected nodes without having every gateway involved. Thus, our estimation scheme, which requires every gateway to send only two messages, it probably the most efficient scheme that can be designed. In Figure 7 we compare the estimation error for  $B = 200$  Level-1 gateways and two different node distributions. The top curve (“Uniform”) depicts the case when each gateway has a uniformly distributed random number of affected nodes in the range  $[1, 1000]$ , as in the previous graphs. The bottom curve (“Non uniform”) depicts the case when each Level-1 gateway  $i$  has  $i \cdot 20$  affected nodes. The figure shows that in both cases the error is very small, i.e., less than 2%.

Finally, Figure 8 depicts the estimation error as a function of the number of gateway levels in the hierarchy, with 6,250,000 affected nodes. Each Level-1 gateway has 50 affected nodes. For the 2-level case we use  $6,250,000/50=125,000$  Level-1 gateways. For the 3-level case we use 125,000 Level-1 gateways and  $6,250,000/(50)^2 = 2,500$  Level-2 gateways.

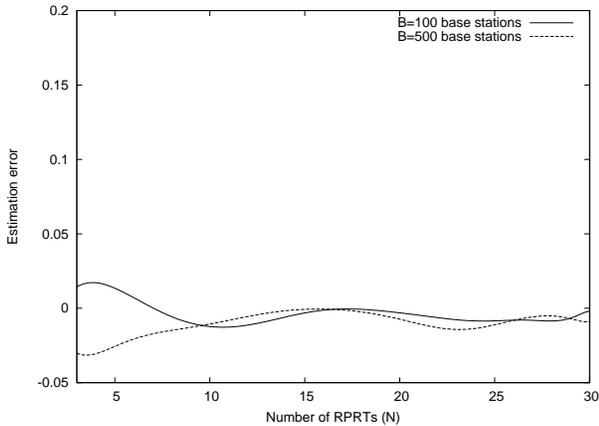


Fig. 6. Estimation error of Algorithm 2 vs.  $N$  for  $B = 100$  and 500 Level-1 gateways

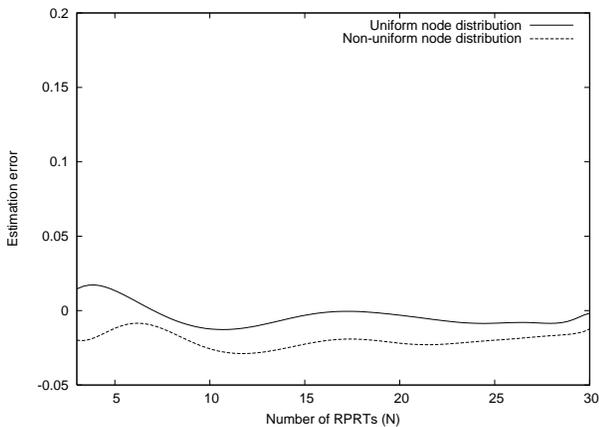


Fig. 7. Estimation error vs.  $N$  for  $B = 200$ , with a uniform and non-uniform distribution of affected nodes per Level-1 gateway

For the 4-level case we use 125,000 Level-1 gateways, 2,500 Level-2 gateways and 50 Level-3 gateways. For all cases, each non-root Level- $i$  gateway has 50 Level- $(i + 1)$  gateways or affected nodes. The top curve shows the estimation error for  $N = 5$  and the bottom for  $N = 10$ .

Several conclusions can be drawn from the figure. While  $N = 10$  leads to a lower estimation error, the improvement is not dramatic, whereas the cost is twice the cost of  $N = 5$ . Second, while the error increases with the number of levels, it is reasonable ( $< 7\%$ ) for any practical number of levels. The error and the number of levels correlate because it is more likely that two or more RPRts need to be forwarded by the same Level- $(L - 1)$  gateway when the number of levels increases, which is not possible with Scheme 2.

## VI. CONCLUSIONS

This paper addresses the problems of feedback suppression in hierarchical sensor networks. We considered a hierarchical sensor network and adjusted a scheme defined in a previous work to be used by a root gateway in order to estimate the number of sensors experiencing a given event. To reduce

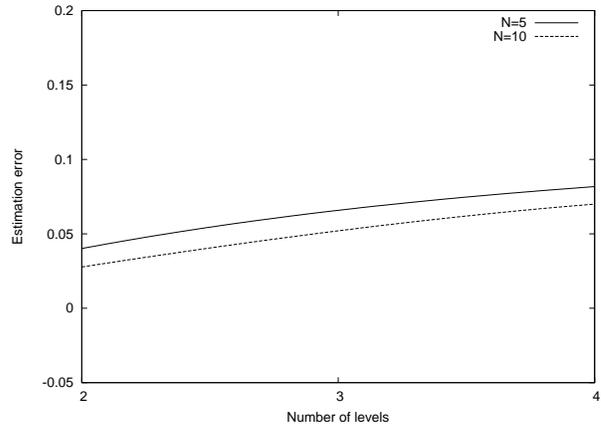


Fig. 8. Estimation error vs. number of gateway levels for 6,250,000 affected nodes and  $N = 5$  and 10

the communication cost, we developed a new scheme and a new algorithm to be employed by the root gateway. We believe that no other scheme can perform the same task with a lower communication cost. We analyzed the new scheme mathematically, calculated its estimation error, and showed that the actual error is very close to the analytical result. Using this observation, we improved the estimation algorithm and substantially reduced its error.

## REFERENCES

- [1] Ian F. Akyildiz, Tommaso Melodia, and Kaushik R. Chowdhury. A survey on wireless multimedia sensor networks. *Computer Networks*, 51(4):921–960, 2007.
- [2] Jamal N. Al-Karaki and Ahmed E. Kamal. Routing techniques in wireless sensor networks: a survey. *Wireless Communications, IEEE*, 11(6):6–28, Dec. 2004.
- [3] Sara Alouf, Eitan Altman, Chadi Barakat, and Philippe Nain. Estimating membership in a multicast session. In *SIGMETRICS*, 2003.
- [4] Sara Alouf, Eitan Altman, and Philippe Nain. Optimal on-line estimation of the size of a dynamic multicast group. In *INFOCOM*, 2002.
- [5] Urban Bilstrup and Per-Arne Wiberg. An implementation of a 3-tier hierarchical wireless sensor network. *2006 IEEE International Conference on Industrial Informatics*, pages 138–143, Aug. 2006.
- [6] Haowen Chan, Adrian Perrig, and Dawn Song. Secure hierarchical in-network aggregation in sensor networks. In *CCS '06: Proceedings of the 13th ACM Conference on Computer and Communications Security*, pages 278–287, New York, NY, USA, 2006. ACM.
- [7] Reuven Cohen and Alexander Landau. “Not All At Once!” – A generic scheme for estimating the number of affected nodes while avoiding feedback implosion. In *INFOCOM 2009 (mini-conference)*.
- [8] Wendi Rabiner Heinzelman, Anantha Chandrakasan, and Hari Balakrishnan. Energy-efficient communication protocol for wireless microsensor networks. In *HICSS '00: Proceedings of the 33rd Hawaii International Conference on System Sciences-Volume 8*, page 8020, Washington, DC, USA, 2000. IEEE Computer Society.
- [9] Wendi Rabiner Heinzelman, Joanna Kulik, and Hari Balakrishnan. Adaptive protocols for information dissemination in wireless sensor networks. In *MobiCom '99: Proceedings of the 5th Annual ACM/IEEE International Conference on Mobile Computing and Networking*, pages 174–185, New York, NY, USA, 1999. ACM.
- [10] Chalermek Intanagonwiwat, Ramesh Govindan, and Deborah Estrin. Directed diffusion: a scalable and robust communication paradigm for sensor networks. In *MobiCom '00: Proceedings of the 6th Annual International Conference on Mobile Computing and Networking*, pages 56–67, New York, NY, USA, 2000. ACM.

- [11] Yeonsik Jeong, S. Lakshmanan, S. Kakumanu, and R. Sivakumar. Cue-based networking using wireless sensor networks: A video-over-rip application. *SECON '08: 5th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks, 2008.*, pages 287–295, June 2008.
- [12] Purushottam Kulkarni, Deepak Ganesan, and Prashant Shenoy. The case for multi-tier camera sensor networks. In *NOSSDAV '05: Proceedings of the International Workshop on Network and Operating Systems Support for Digital Audio and Video*, pages 141–146, New York, NY, USA, 2005. ACM.
- [13] Richard LaMaire and Arvind Krishna. A novel feedback scheme to increase throughput in multiple access radio systems. *Wireless Networks*, 6(1):3–15, 2000.
- [14] Stephanie Lindsey and Cauligi S. Raghavendra. PEGASIS: Power-efficient gathering in sensor information systems. *Aerospace Conference Proceedings, 2002. IEEE*, 3:3–1125–3–1130 vol.3, 2002.
- [15] Alan Mainwaring, David Culler, Joseph Polastre, Robert Szewczyk, and John Anderson. Wireless sensor networks for habitat monitoring. In *WSNA '02: Proceedings of the 1st ACM International Workshop on Wireless Sensor Networks and Applications*, pages 88–97, New York, NY, USA, 2002. ACM.
- [16] Arati Manjeshwar and Dharma P. Agrawal. TEEN: A routing protocol for enhanced efficiency in wireless sensor networks. *Proceedings of the 15th International Parallel and Distributed Processing Symposium*, pages 2009–2015, Apr 2001.
- [17] Arati Manjeshwar and Dharma P. Agrawal. APTEEN: A hybrid protocol for efficient routing and comprehensive information retrieval in wireless sensor networks. *IPDPS '02: Proceedings of the 16th International Parallel and Distributed Processing Symposium*, pages 195–202, 2002.
- [18] Leo Selavo, Anthony Wood, Qihua Cao, Tamim Sookoor, Hengchang Liu, A. Srinivasan, Yafeng Wu, Woochul Kang, Jack Stankovic, D. Young, and J. Porter. Luster: wireless sensor network for environmental research. In *SenSys '07: Proceedings of the 5th International Conference on Embedded Networked Sensor Systems*, pages 103–116, New York, NY, USA, 2007. ACM.
- [19] Victor Shnayder, Mark Hempstead, Bor-rong Chen, Geoff Werner Allen, and Matt Welsh. Simulating the power consumption of large-scale sensor network applications. In *SenSys '04: Proceedings of the 2nd International Conference on Embedded Networked Sensor Systems*, pages 188–200, New York, NY, USA, 2004. ACM.
- [20] Mark Yarvis, Akishore Kushalnagar, Harkirat Singh, Anand Rangarajan, York Liu, and Suresh Singh. Exploiting heterogeneity in sensor networks. *INFOCOM 2005. 24th Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings IEEE*, 2:878–890 vol. 2, March 2005.
- [21] Mohamed Younis, Poonam Munshi, and Ehab Al-Shaer. Architecture for efficient monitoring and management of sensor networks. *Lecture Notes in Computer Science*, 2839/2003:488–502, 2003.
- [22] Yi Zeng, Bo Hu, and Shunjia Liu. Vector kalman filter using multiple parents for time synchronization in multi-hop sensor networks. *SECON '08: 5th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks, 2008.*, pages 413–421, June 2008.
- [23] Zhenghao Zhang, Ming Ma, and Yuanyuan Yang. Energy-efficient multi-hop polling in clusters of two-layered heterogeneous sensor networks. *Computers, IEEE Transactions on*, 57(2):231–245, Feb. 2008.
- [24] Suli Zhao, Kemal Tepe, Ivan Seskar, and Dipankar Raychaudhuri. Routing protocols for self-organizing hierarchical ad hoc wireless networks. In *IEEE Sarnoff Symposium*, 2003.