Information Assurance in Sensor Networks

Budhaditya Deb, Sudeept Bhatnagar and Badri Nath

Rutgers University,
110 Frelinghuysen Rd,
Piscataway, NJ 08854, USA
{bdeb, sbhatnag, badri}@cs.rutgers.edu

ABSTRACT

Sensor networks are deployed to monitor the surroundings and keep the end-user informed about the events witnessed. Different types of events have different levels of importance for the user. *Information Assurance* is an ability to disseminate different information at different *assurance levels* to the end-user. The assurance level is determined by the criticality of the sensed phenomenon. Thus, information assurance capability allows a sensor network to deliver critical information with high assurance albeit potentially at a higher cost, while saving energy by delivering less important information at a lower assurance level.

In this paper, we look at the problem of efficient information assurance in sensor networks when the assurance level of information is defined as the probability of information delivery (*desired reliability*) to the sink. We propose a new scheme for information delivery at a desired reliability using hop-by-hop broadcast. We show how the wireless broadcast can be utilized to increase the packet delivery rate at each hop and attain a desired reliability at minimal cost. Finally, we derive the optimal strategy for allocation of desired reliabilities at each hop in order to attain any given desired end-to-end reliability.

Categories and Subject Descriptors

C.2.1 [**Network Architecture and Design**]: Wireless Communication

General Terms

Algorithms, Design, Reliability, Theory

Keywords

Sensor Networks, Information Assurance, Reliability, Wireless Broadcast

1. INTRODUCTION

Sensor networks have emerged as the prime candidates to provide an efficient basis for monitoring the environment. These networks consist of small sensor nodes that run on battery power, have limited memory and processing power, and are capable of wireless communication [1]. The nodes collect data by sensing the environment, process it locally, and send the information back to the user (potentially over multiple hops). Each of these tasks consumes energy which is perhaps the most critical resource at a sensor node. Thus, energy-efficiency is the most important criterion for designing any algorithm or protocol for sensor networks.

One important aspect in sensor network operations is the process of information dissemination. In general, the disseminated information would have different levels of importance to the end-user and would require different levels of guarantees in delivery. For example, the information of a potential chemical leak is more important than knowing that everything is fine (which might be the norm) and should have *higher reliability* and *lower delay* in delivery. We refer to any such guarantees in sensor data delivery as *Assurance Levels*. The assurance level for any data packet is determined by the information content that it carries. *Information Assurance* is an ability to disseminate different information at different assurance levels to the end-user.

In this paper, we look at the problem of efficient information assurance in sensor networks for the specific case where the assurance level is defined as the probability of information delivery (*desired reliability*) to the sink. The existing solutions treat the process of sending information to the end-user in two extreme ways: unreliable transmission or reliable transmission. Unreliable transmissions involve a node sending a packet to the next-hop node without worrying about the fate of the packet. At the other extreme, reliable delivery of a packet entails retransmissions of a packet until it is received successfully at the next hop (possibly using ARQ based schemes). However, we believe that reliability is *not* a boolean task in the context of sensor networks (i.e., either you have reliability or you don't is not the correct view of reliability for sensor networks). This belief arises from the following facts:

- Sensor networks could be deployed in an environment where channel errors could be very high. Thus, the overhead incurred in reliable transmission of a packet could be significant. On the other hand, unreliable transmission in such an environment would lead to a great deal of information loss. Thus, treating reliability as a boolean task has severe implications, either in terms of energy loss or in terms of information loss.
- The purpose of sensor networks is information dissemination. Thus, the loss of important information at the perceived benefit of energy-efficiency, actually inhibits the ability of a sensor network to fulfill its primary purpose.

WSNA '03, September 19, 2003, San Diego, California, USA.

- The example of monitoring chemical leak illustrates that disseminated data has different levels of importance for the user. Thus, the criticality of information is the criterion which determines the desired degree of reliability and hence the amount of energy that the network should spend in trying to attain the desired reliability. For a detailed discussion on the need to exploit the reliability-overhead trade-off in a sensor network, the reader is referred to [2].

In this paper, we evaluate various hop-by-hop schemes to deliver packets at any desired reliability. We first analyze a naive scheme in which the packets are explicitly addressed to one next-hop neighbor and derive the amount of redundancy required to attain the given reliability. We propose a hop-by-hop broadcast scheme which exploits the broadcast capability of the wireless medium to reduce the packet overhead while still attaining the desired reliability. We consider both the variations of this scheme: Hop-By-Hop Broadcast without acknowledgement (HHB) and Hop-By-Hop Broadcast with acknowledgement (HHBA). Both these schemes attain any desired degree of reliability at a minimal proportionate cost. The protocols are simple yet flexible mechanisms to implement information assurance (desired reliability).

2. PROBLEM STATEMENT

We assume that the network consists of randomly deployed sensor nodes in a given field. The links between nodes are assumed to be symmetric (which is required for hop-by-hop acknowledgment based scheme to work). There is a single *sink node* to which all the data is reported. Each sensor node collects information by sensing, processes it and sends it to the sink node. Each sensor node *i* is aware of its hop-distance, h_i, to the sink. Node *i* is also aware of the local channel error, e_i, (which it could deduce over a period of time). We assume a TDMA MAC layer [15] which avoids packet collisions. In future work, we plan to address the information assurance problem for sensor networks with other type of MAC layers like 802.11.

Each source is assumed to be aware of the desired data delivery reliability *r* (where $0<r<1$), based on the information content of the packet it is sending.[1] Desired reliability is the probability with which the source wants the packet to be delivered to the sink. The overhead of the information delivery process is given by the cumulative number of packets sent at each hop along the path from the source to sink (including retransmissions).

The nodes are assumed to have an ability to cache the packets they receive correctly from the previous hop. Thus, they can distinguish between different copies of the same packet (if the packet is retransmitted) and discard duplicate packets. If the nodes do not have caching capability, then a scheme similar to that proposed in [13] can be used. We note that the solution proposed in [13] involves end-to-end retransmissions. However, the overhead of end-to-end schemes (with or without acknowledgments) is significantly higher than hop-by-hop schemes and hence, it is useful only if the nodes are memory-constrained [13], [14]. We do not consider end-to-end schemes in this paper.

In this setting, our aim is to provide a mechanism to provide the desired reliability at a minimal overhead. We want to utilize the caching ability of sensor nodes in conjunction with the one-hop broadcast capability of the wireless medium, to design an optimal scheme to attain any desired end-to-end reliability.

3. HOP-BY-HOP UNICAST

In this section, we see how the conventional hop-by-hop unicast based forwarding scheme can be modified to accommodate different levels of reliability. We first consider the scheme where each packet is sent to a fixed next-hop node and the next-hop node does not acknowledge the packet. Later, we look at its variation where the next-hop node sends back an acknowledgment if it receives the packet correctly. The source node stops further retransmissions if it receives an acknowledgment.

3.1 Hop-By-Hop Reliability (HHR)

Hop-by-Hop Reliability (HHR) scheme involves a source sending multiple copies of a packet to *exactly one* next-hop neighbor. The next-hop node does not send any acknowledgment for any packets. This idea has been used to reduce the overhead of reliable transport protocols [10], [11]. We derive similar results in order to attain any desired degree of reliability.

Let the source be *h* hops away from the sink and *r* be the required reliability of packet delivery. Let the reliability at the i^{th} hop be r_i such that $\Pi\, r_i = r$. For ease of exposition, suppose that each $r_i = r^{1/h}$ and the channel error *e*, is constant at each hop. The sender node unicasts each packet to one (possibly randomly chosen) next-hop neighbor.

Consider the packet forwarding process at a single hop. Clearly, if $r^{1/h} \geq (1\text{-}e)$, then the sender needs to send multiple copies of the packet to attain the desired reliability. Let N_{HHR} be the number of copies required to attain a reliability of $r^{1/h}$. The probability that *at least* one of the copies of the packet is received at the next-hop node should be $r^{1/h}$. Thus,

$$r^{1/h} = 1 - e^{N_{HRR}}$$

$$\Rightarrow N_{HHR} = \frac{\log\left(1 - r^{1/h}\right)}{\log(e)} \qquad \textbf{... 1}$$

Since the probability of a packet being forwarded correctly at each hop is $r^{1/h}$, the total expected packet overhead, O_{HHR}, is given by:

$$O_{HHR} = N_{HHR} \sum_{i=0}^{h-1} r^{i/h} = \frac{\log\left(1 - r^{1/h}\right)\left(1 - r\right)}{\log(e)\left(1 - r^{1/h}\right)} \qquad \text{... 2}$$

It can be verified that O_{HHR} is significantly less than the overhead incurred in any end-to-end scheme, even though this is the most naïve form of hop-by-hop implementation of information assurance.

3.2 Hop-By-Hop Reliability with Acknowledgments (HHRA)

Next we derive the overhead incurred for hop-by-hop reliability with acknowledgments (HHRA). In HHRA, when a source sends a packet, the next-hop node sends back an acknowledgment for a

[1] We note that *information-awareness* involves classification of events, ascertaining their criticality and mapping the criticality to an assurance level. All of these are complex tasks and beyond the focus of this paper.

correctly received packet. Retransmissions stop as soon as the source receives an acknowledgment packet.

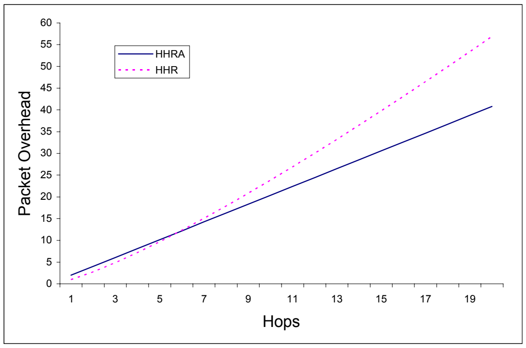

Figure 1. Effect of increasing number of hops on O_{HHR} and O_{HHRA} for r=0.7 and e=0.3

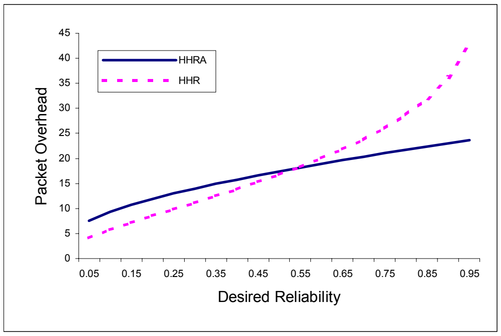

Figure 2. Effect of increasing the desired reliability on O_{HHR} and O_{HHRA} for h=10 and e=0.3

We want to find the maximum number of retransmissions at the source so that HRRA attains a reliability of $r^{1/h}$ at each hop. Clearly, N_{HHR} is also the maximum number of retransmissions after which the source should stop even if does not receive an acknowledgment. However, the number of copies of the packet transmitted by the source is different because it could stop (before sending N_{HHR} copies) if it gets an acknowledgment.

In HHRA, a successful transmission occurs when a packet is received correctly at the receiver and the acknowledgment is received correctly at the sender. In this paper, we assume that acknowledgment packets are forwarded with very high probability and are not lost. This can be achieved since acknowledgments are small in size and FEC codes with high degree of redundancy can be introduced in the packet itself.

When using HHRA, a source would retransmit a packet the i^{th} time only if all previous *i-1* copies of the packet were dropped. Hence, the expected number of transmissions, N_{HHRA}, at a hop is given by:

$$N_{HHRA} = 1 + \sum_{i=1}^{N_{HHR}} e^{i-1} = 1 + \frac{1-e^{N_{HHR}}}{1-e} = 1 + \frac{r^{1/h}}{1-e} \qquad \text{... 3}$$

Thus, using HRRA, the total overhead incurred over *h* hops, O_{HHRA}, when trying to achieve a reliability $r^{1/h}$ at each hop is:

$$O_{HHRA} = N_{HHRA}\sum_{i=0}^{h-1} r^{i/h} = \left(1+\frac{r^{1/h}}{1-e}\right)\frac{(1-r)}{(1-r^{1/h})} \qquad \text{... 4}$$

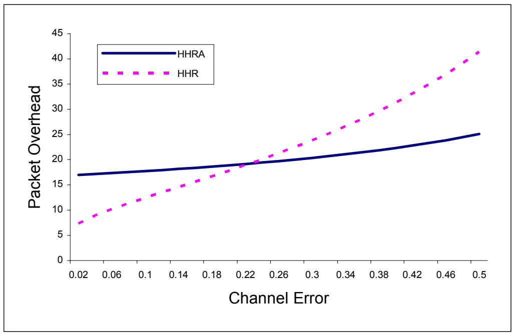

Figure 3. Effect of increasing the channel error on O_{HHR} and O_{HHRA} for h=10 and r=0.7

In Figure 1, Figure 2 and Figure 3, we plot the overhead equations 2 and 4 to evaluate the impact of hops between source and sink, desired reliability and channel error on O_{HHR} and O_{HHRA}. The figures show the expected number of packets transmitted by all the nodes in order to provide the desired end-to-end reliability to *one* packet. Figure 1 shows the effect of increasing the number of hops between the source and the sink while trying to attain a reliability of 0.7 and when the channel error is 0.3. Figure 2 shows the effect of increasing the desired reliability when the source and sink are 10 hops away and the channel error is 0.3. Figure 3 shows the effect of increasing channel errors when the source is 10 hops away from the sink and the desired reliability is 0.7. In all these figures, we see that initially HHR performs better than HHRA. This is due to the extra overhead of acknowledgment packets. However, their effect is negligible and HHRA quickly outperforms HHR as the conditions grow stringent. Thus, under high channel errors, large number of hops or high desired reliability, HHRA has lesser overhead in attaining the reliability. Consequently, it is better to use an acknowledgment-based scheme in sensor networks since it does not perform too badly in mild conditions but significantly better in testing conditions. Also, since the overhead due to acknowledgments is a small fraction of the total overhead, in subsequent derivations we ignore the overhead due to acknowledgments for simplicity.

4. HOP-BY-HOP BROADCAST

HHR and HHRA are both simple and provide the desired reliability at a reasonable overhead. However, they are constrained by their use of unicast to reach a next-hop node. Our main contribution in this paper is in showing how the use of one-hop broadcast capability of wireless medium can achieve significant reduction in overhead while providing the desired reliability. We illustrate the basic ideas of this scheme in this section.

Consider a source and sink node separated by *h* hops with a required reliability *r*. For ease of exposition, let us assume that the channel error is a constant *e* at each hop. Using the hop-by-hop methods, we require $r^{1/h}$ reliability at each hop for which we require N_{HHR} copies as given by equation 1. Our aim is to reduce the number of copies required to *less than* N_{HHR} while still attaining the desired reliability of $r^{1/h}$ at any hop.

To attain this objective, we utilize the broadcast property of the wireless medium. Since sensor networks typically have high density[2], there would be multiple nodes which are *h-1* hops away from sink. Thus, for a packet to reach the sink with reliability *r*, it is sufficient to have *any one* of these *h-1* hop nodes receive the packet with reliability $r^{1/h}$. Consider the example shown in Figure 4. In HHR and HHRA, the source sends a packet (or any copy of the packet) to one of its three next-hop neighbors *a, b* or *c*. However, if the source uses one-hop broadcast instead of unicast, then *one copy* of the packet acts as *three copies*, one sent to each of source's next-hop neighbors. Similarly, one packet sent by *c* acts like four retransmissions (because it has four next-hop neighbors *d*, *e*, *f* and *g*).

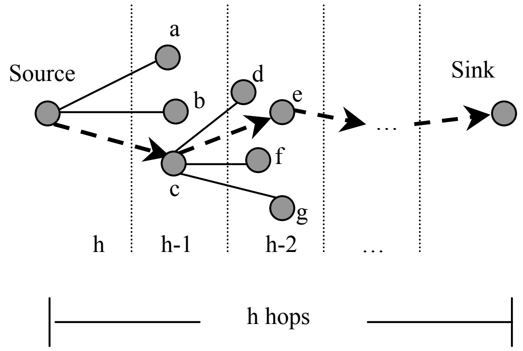

Figure 4. Illustration of the basic strategy used in hop-by-hop broadcast.

There are two key issues to be addressed in order to use hop-by-hop broadcast:

1. Determining the number of copies, N_{HHB}, of a packet required to be sent at each hop. This is easily seen to be equal to N_{HHR}/k, where *k* is the number of next-hop nodes at a source. Thus,

$$r^{1/h} = 1 - e^{N_{HHB}k}$$

$$N_{HHB} = \frac{\log\left(1 - r^{1/h}\right)}{k \log e} = \frac{N_{HHR}}{k} \le N_{HHR} \qquad \dots 5$$

2. Ensuring that *at most one* of the next-hop neighbor forwards the multiple copies of the packet. Since, more than one next-hop neighbors can receive a packet correctly, it is a waste of resources if all of them send N_{HHB} copies of a packet to the next-hop. For example, in Figure 4, if nodes *d, e* and *f* receive a packet correctly from c, then only one of them (node *e* in the figure) should forward it to nodes at *h-3* hops from sink.

   To strictly enforce the rule that exactly one node forward a packet at each hop, we would have to incur extra overhead in form of some control packets. Instead we use a probabilistic method so that we have only one node forwarding a packet at each hop, *in the expected*. Thus, each next-hop node that receives the packet, forwards the packet with a probability p_f which is given by:

$$p_f = \frac{1}{k\left(1 - e^{N_{HHB}}\right)} \qquad \dots 6$$

Thus, to use HHB to provide desired reliability, at each hop one node broadcasts N_{HHB} packets. This ensures that the packet reaches one of the next-hop nodes with a probability $r^{1/h}$. Each next-hop node that receives the packet correctly, forwards (broadcasts N_{HHB} copies) the packet with a probability p_f. Although each receiving node probabilistically forwarding the packet does not guarantee that at least one node would forward it, the probabilistic guarantee in packet delivery is still maintained.

5. HHB PROTOCOL

In this section, we integrate the hop-by-hop broadcast idea with the packet processing operations at nodes, to form the communication protocols to attain end-to-end desired reliability. Like in the previous sections, we describe two possible variations: The first one is Hop-by-Hop Broadcast Protocol (HHB) which does not use acknowledgments. At each hop, nodes which decide to forward, broadcast N_{HHB} copies of the packet. The second method uses acknowledgments and a stop and wait protocol to forward packets. This protocol is called Hop-by-Hop Broadcast with Acknowledgment (HHBA).

First we describe HHB where the source node preemptively sends the required number of copies of the packets. Later, we show the modifications that need to be done to HHB to get HHBA.

5.1 Operations at the packet source

When a source detects any sensed event, first it computes the criticality of the detected event and maps it to the required reliability *r*. The source is *H* hops away from the sink and hence the per-hop reliability required is $r^{1/H}$.

The source computes the number of packets required to provide this reliability, given by N_{HHB} in equation 5. It then generates a packet with the detected information and the following additional packet fields:

- **R**: Required packet reliability (set to $r^{1/H}$)
- **H**: Hops from the source to sink
- **H_s**: The hop distance of the *sender* to the sink
- **K**: Number of *next-hop* neighbors
- **E**: Local channel error
- **S**: The *unique* sequence number of the packet
- **N_{HHB}**: Total copies of the packet to be sent
- **C**: Copy number of the packet (C varies from 1 to N_{HHB})

Of the above values there are three new fields: H_s, *S* and *C*. The field H_s is the hop distance of the sender to the sink. Since the sender of the packet keeps on changing (e.g., intermediate node sending the packet), the sender keeps on updating this field. The packet's sequence number *S* is a unique local id generated by the source. The source-id and *S* together uniquely identify a packet.

[2] Since the communication range >> sensing range, to have a field sensor covered would amount to having a high density of the sensor network (in terms of communication coverage). Moreover, just to keep the network connected, each node requires at least 6 neighbors if the nodes are randomly deployed [12]. To add robustness, the number of neighbors for each node would be significantly higher than 6.

All the retransmitted copies of a packet carry the same sequence number. The field *C* is the copy number of the packet with sequence *S*. For each unique packet (with unique *S*), the value of *C* could range between 1 and N_{HHB}. The source populates these fields with the local values that it knows (or computes). Finally it broadcasts N_{HHB} copies of the packet with some time interval T_{delay} between consecutive copies.

5.2 Operations at an intermediate node

When a node (other than the sink) receives a packet, it first checks the packet field H_s in the packet header. If the node is one hop nearer to the sink node than the sender, i.e., it is H_s *-1* hops away from the sink, it decides to process the packet. Otherwise the packet is dropped (at the nodes that are H_s and H_s+1 hops away).

While processing the packet header, the node first checks whether the packet has been received earlier. For this, it caches the recently seen sequence numbers in a list *L* and searches this list for the value *S* in the packet header**.** If the packet's sequence number is seen for the first time, it decides to forward the packet with a probability p_f which is computed by substituting the values of various fields in the packet header, into equation 6.

If the node decides to forward a packet, the following operations are done in order:

- It waits for time delay $(N_{HHB} – C)T_{delay}$, using N_{HHB} and *C* values from the packet.
- It computes its own N_{HHB} (based on its local information).
- Updates all the packet headers, except *R* and *H*.
- Sends N_{HHB} copies of the packet with delay T_{delay} between each.
- Clears the entry for the packet from L after waiting for some duration after all the N_{HHB} copies of the packet have been sent.

We note here that T_{delay} needs to be large enough so that each packet has sufficient time to get through. If this is satisfied, then the sequence number in the packet along with the node-id uniquely determines the packet. The size of *L* is bounded by the degree of a node if each node transmits a new packet only if it has transmitted all the N_{HHB} copies of the previous packet. This is true because at each node, *L* would contain at most one entry for each of its neighbors.

5.3 Modifying HHB to HHBA

Now we describe the modifications performed on HHB to use acknowledgments and creating the HHBA Protocol. We use the same packet header fields as used for HHB. The changes are described here.

The T_{delay} has to increase so that acknowledgments for packet can be received. Thus, $T_{delay}=T_{RTT}$ where T_{RTT} is the one-hop round trip time.

When a node receives a packet, it drops the packet if its hop-distance to the sink is H_s or H_s+1. If it does not drop the packet, it decides to forward the packet with probability p_f given by

$$p_f = \frac{1}{K\left(1 - E^{\tilde{N}}\right)} \qquad \text{... 7}$$

If the node decides to forward the packet, it sends an acknowledgment as a confirmation of forwarding, back to the source. Thus, the acknowledgment serves the purpose of letting the source know that some node *which is willing to forward the packet* has already received the packet and the source should stop further retransmissions.

We note that in the computation of p_f, the value $\tilde{N}$ is used instead of N_{HHR} in equation 7. This is the expected number of packets that the source would have transmitted *till the time when a next-hop node decides to forward and sends an acknowledgment.* This is because once a next-hop node decides to forward and sends an acknowledgement, the source does not transmit the remaining N_{HHR} - $\tilde{N}$ copies.

6. EXPECTED OVERHEAD OF HHB

Now we compute the overhead of HHB and HHBA. We define the following variables used in this section:

e_i *= average channel error at* i^{th} *hop*

k_i*= expected number of next hop neighbors at* i^{th} *hop*

r = end-to-end required reliability

r_i *= per-hop reliability=* $r^{1/h}$ *(also,* $r_0=1$*)*

6.1 Expected Overhead of HHB

The number of packets to be transmitted at i^{th} hop if we use HHB, N_i, is given by:

$$N_i = \frac{\log(1 - r_i)}{k_i \log e_i}$$

Thus, the expected overhead, O_{HHB}, incurred in providing a desired end-to-end reliability of *r* using HHB is:

$$O_{HHB} = \sum_{i=1}^{h}\left(N_i \prod_{j=0}^{i-1} r_j\right) = \sum_{i=1}^{h} N_i r^{(i-1)/h} \qquad \text{... 8}$$

6.2 Expected Overhead of HHBA

We know that the maximum number of retransmissions at the i^{th} hop for a given reliability, even when using HHBA instead of HHB, is the same as N_i as computed above. Thus, after sending N_i copies of the packet, a sender at i^{th} hop can stop further retransmissions even if it does not get an acknowledgment.

As mentioned earlier, the acknowledgment packets are likely to reach correctly because of their small size and hence we do not consider them in the computation of the overhead. Thus, the sender transmits the i^{th} copy of the packet *only if* all of the previous *i-1* copies were incorrectly received at all the next-hop neighbors. In this case, the expected number of copies of the packet transmitted at the i^{th} hop, $\hat{N}_i$, is given by:

$$\hat{N}_i = \sum_{i=1}^{N_i} e^{k_i(i-1)} = \frac{1 - e_i^{k_i N_i}}{1 - e_i^{k_i}}$$

Substituting the following for N_i *:*

$$N_i = \frac{\log(1 - r_i)}{k_i \log e_i} = \frac{\log_{e_i}(1 - r_i)}{k_i}$$

$$\hat{N}_i = \frac{r_i}{1-e_i^{k_i}} \qquad \text{... 9}$$

Thus, the total end-to-end overhead using HHBA is given by:

$$O_{HHBA} = \sum_{i=1}^{h}\left(\hat{N}_i \prod_{j=1}^{i-1} r_j\right) = \sum_{i=1}^{h} N_i r^{(i-1)/h} \qquad \text{... 10}$$
$$O_{HHBA} = \sum_{i=1}^{h} \frac{r^{i/h}}{1-e_i^{k_i}}$$

7. OPTIMAL ALLOCATION OF PER-HOP RELIABILITY

The previous sections described the HHB and HHBA protocols and gave the expected overhead of the process. In the description of these schemes, we used a naïve technique to allocate desired reliabilities at each hop i as $r_i=r^{1/h}$. However, this allocation may not be optimal. In this section, we try to derive the optimal values of r_i for which the expected overhead is minimum. That is, given a desired end-to-end reliability r, what should the desired reliability at each hop be in order to minimize the total overhead?

7.1 Optimal Allocation for HHB

Let

r = required reliability for a packet.

r_i = required reliability at each hop.

k_i = expected number of next hop nodes at i^{th} hop.[3]

Using O_{HHB} computed in equation 8, we have the following constrained optimization problem for optimal packet overhead:

$$\min\left[O_{HHB} = \sum_{i=1}^{h}\left(\frac{\log(1-r_i)}{k_i \log e_i}\prod_{j=0}^{i-1} r_j\right)\right] \qquad \text{... 11}$$
$$s.t.$$
$$\prod r_i = r$$
$$r \le r_i \le 1; \quad r_0 = 1$$

The optimal solution to the above problem would give the values of reliability r_i at each hop such that the total end-to-end reliability is r and for which the total overhead is minimized. Instead of solving the above analytically, we show that under all practical network conditions, we would be using HHBA instead of HHB. Thus, getting the solution to this optimization problem is superfluous.

7.2 Optimal Allocation for HHBA

Now we look at the optimal allocation of per-hop reliabilities for the acknowledgement-based method. Again, we neglect the overhead of the acknowledgement packets. Acknowledgement packets have negligible overhead as we saw in the previous section in results for overhead incurred HHR and HHRA, when the required reliability, channel error or number of hops is large. Thus, allocating reliabilities to each hop in order to minimize O_{HHBA} reduces to the following optimization problem:

$$\min\left[O_{HHBA} = \sum_{i=1}^{h}\left(\frac{\prod_{j=1}^{i} r_i}{1-e_i^{k_i}}\right)\right] \qquad \text{... 12}$$
$$s.t \prod_{j=1}^{h} r_j = r, \qquad r \le r_i \le 1$$

In this formulation, the value of O_{HHBA} is used from equation 10. We use the following simple technique to solve the above problem. Let,

$$\alpha_i = \frac{1}{1-e_i^{k_i}}$$

Then the overhead O_{HHBA} is given by:

$$\begin{aligned} O_{HHBA} &= \alpha_1 r_1 + \alpha_2 r_1 r_2 + \cdots + \alpha_h r_1 r_2 \cdots r_h \\ &\ge \alpha_1 r_1 r_2 \cdots r_h + \alpha_2 r_1 r_2 \cdots r_h \\ O^{*}{}_{HHBA} &= R(\alpha_1 + \alpha_2 + \cdots + \alpha_h) \end{aligned} \qquad \text{... 13}$$

Thus, the minimum value of O_{HHBA} (denoted by O^{*}_{HHBA}) occurs when the coefficient of each α_i in each of the terms in the summation is R. This is achieved only when $r_1=R$ and all other r_j are 1, irrespective of the values of e_i and k_i. The optimal distribution of reliability, r^{*}, is hence given by:

$$\begin{aligned} r_1^{*} &= r \\ r_i^{*} &= 1 \quad \forall i \ne 1 \end{aligned} \qquad \text{... 14}$$

The above solution for r_i is the optimal solution of the problem. We also see that the above is a feasible solution to the constraints of the problem and hence it is the optimal solution to the optimization problem in equation 12. The optimality of this solution essentially means that we should give a reliability of r at the first hop and a reliability of 1 at subsequent hops to minimize the overhead. Thus, the protocol becomes very simple for HHBA.

We still don't have an optimal solution for HHB and it requires numerical methods to solve for the values of r_i. However, if for any values of r_i , if O_{HHBA} is less than O_{HHB}, then we don't really require HHB at all. We now show that this is indeed true.

Consider a set of allocated reliabilities r_i and compare the overhead incurred by HHB and HHBA in order to provide these reliabilities. Consider the i^{th} term in the expansions of O_{HHBA} and O_{HHB} from equations 8 and 10. Thus, O_{HHBA} is less than O_{HHB} if:

$$\frac{r_1 \cdots r_i}{1-e_i^{k_i}} \le \frac{\log(1-r_i) r_1 \cdots r_{i-1}}{k_i \log(e_i)} \qquad \forall i \in \{1,\ldots,h\} \qquad \text{... 15}$$
$$\Rightarrow \frac{r_i}{1-e_i^{k_i}} \le \frac{\log(1-r_i)}{k_i \log(e_i)} = \frac{\log(e_i^{N_i k_i})}{k_i \log(e_i)} = N_i$$

[3] One possible technique for computation of expected number of next-hop nodes for a node i hops away from the sink I is using probabilistic techniques similar to [12]. In this paper we use the average number of next hop neighbors in randomly generated topologies.

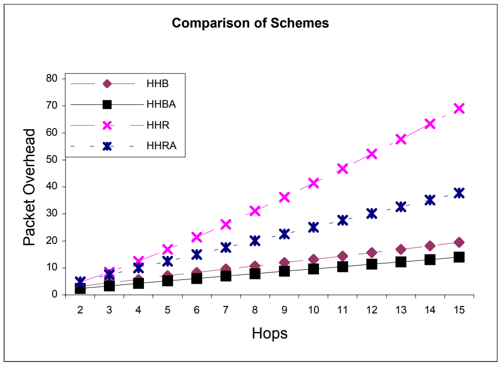

Figure 5. Comparison of schemes: Effect of number of hops on the overhead, for r=0.7 and e=0.5

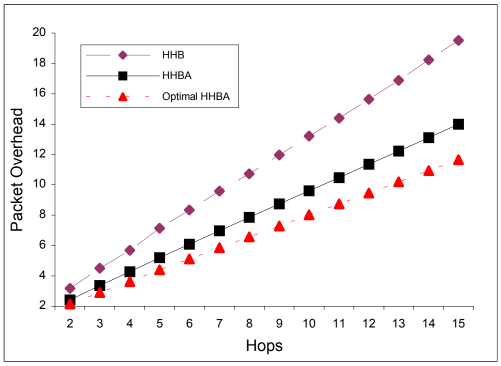

Figure 6. Comparison of HHBA and HHBA optimal, Effect of increasing number of hops, for r=0.7, e=0.5

Since

$$\frac{r_i}{1-e_i^{k_i}} = \frac{1-e_i^{k_i N_i}}{1-e_i^{k_i}} \geq 1 \qquad for \qquad N_i \geq 1$$

Hence, each of the terms in O_{HHBA} is *never more than* those in O_{HHB} whenever we need to send more than one packet *(N_i)* to get the desired reliability. In fact, this consists of all the cases for which either HHBA or HHB would indeed be used, since otherwise we may just use best effort forwarding. Since each of the i^{th} term in O_{HHBA} is less than the corresponding term in O_{HHB}, we conclude that O_{HHBA} can never be more than O_{HHB} if N_i is more than 1.

Thus, we have shown that for the optimal HHB case, we can use HHBA to have even lesser overhead and hence optimal HHBA is the minimum overhead which can be attained out of the methods described in this paper. Note that, we have not considered the overhead of the acknowledgment packets in computing O_{HHBA}. If we consider the extra overhead of one acknowledgment packet per hop, then the O_{HHBA} will be better than O_{HHB} for N_i greater than 2.

8. EVALUATION

We evaluate the four protocols described above (HHB, HHBA, HHR, HHRA) with a simple set up. We consider a network of 10000 nodes, randomly deployed in a field of dimension 800x800m^2. The sink node is assumed to be at the center of the deployment region. We set the communication range to 20m. On this sensor field, we compute the number of hops from each node to the sink and the number of next hop neighbors for each node. The mean of the number of *h-1*-hop neighbors for all nodes that are *h* hops from the sink is computed and used as the value of k_i in the overhead equations. We then use this value of k_i along with different values of *r, e* and *h*, in the formulation of the overhead equations for HHR, HHRA, HHB, HHBA and HHBA-optimal.

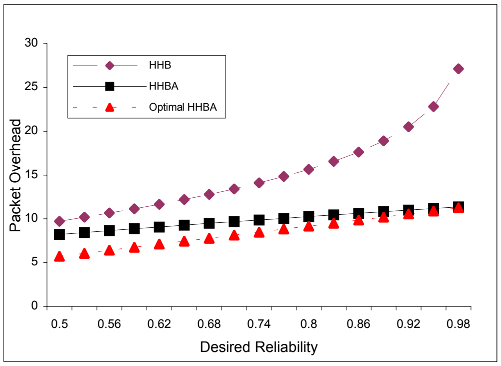

Figure 7. Effect of desired reliability on the overhead of HHB, HHBA and optimal HHBA, for h=10, e=0.5

Figure 5 shows the effect of increasing hop distance on the overhead of various schemes. The channel error was set to 0.5 and the desired reliability to 0.7. We see that HHR performs the worst and deviates rapidly as the number of hops increases. The other schemes have much smaller overhead. As expected, under the stringent conditions of high error, high desired reliability and large number of hops, HHBA is better than the other schemes.

Next we compare the overhead incurred by the broadcast based schemes HHB and HHBA with that of optimal HHBA. For these schemes the naïve reliability allocation of r_i=$r^{1/h}$ at each hop is used. Figure 6 shows that for larger number of hops, the difference between the optimal allocation and naïve HHBA starts increasing. The overhead of HHB increases much faster. Thus, it is beneficial to use the optimal allocation in extreme conditions.

Figure 7 compares the HHB, HHBA and the optimal scheme for varying degrees of reliability. The HHB scheme deviates from the other two in overhead since the number of packets significantly increases at higher reliabilities. However, we see that the naïve HHBA and the optimal HHBA schemes merge at high reliabilities. This is due to the fact the $r^{1/h}$ gets closer to *r* as the value of *r* increases to 1.

Thus, at low level of desired reliability, lower number of hops to the sink and minimal channel error, we may use HHB. However, it is worth increasing the complexity of the communication protocol by requiring per-hop acknowledgments when operating in more stringent conditions because of the significant overhead reduction achieved.

9. RELATED WORK

In [14], authors present a comparison of end-to-end and hop-by-hop schemes for reliable packet delivery. The analyses show that the overhead for end-to-end schemes is prohibitively higher than hop-by-hop schemes. However, end-to-end schemes in general have lesser protocol complexity and timeout functions and require lesser buffer at intermediate hops space to store undelivered packets. For sensor networks which are heavily constrained in memory we proposed a lightweight end-to-end protocol called ReInForM to deliver packets at desired reliability [13]. Memory less nature of ReInForM (requires no caching of packets at intermediate nodes) is suitable for memory constrained sensor nodes. In a similar memoryless network, probabilistic flooding is used to provide multiple levels of reliability and the flooding parameters are tuned using feedbacks from the sink [2]. The probabilistic flooding creates multiple paths from source to sink to provide the desired reliability in packet delivery. However, for large channel errors or hop distance between nodes, the increased overhead of such end-to-end schemes far outweighs the benefits, since sensor networks are also highly energy constrained.

In [3], multiple paths from source to sink are used in diffusion routing framework [4] to quickly recover from path failure. The multiple paths provided by such protocols could be used for sending the multiple copies of each packet. However it incurs extra overhead of multiple path formation and maintenance of path state in each node and is not adaptive to channel errors.

In [10], [11] authors study reliable transport protocols for ad-hoc and sensor networks. However, they look at reliability as an absolute. Reliable transport protocols such as in [10], [11] are designed to retransmit lost packets until a packet is successfully transmitted. Although the number of retransmissions is limited, it is still kept at a value such that packets are transmitted with high probability under any conditions. However, for data which is not critical, we may not need such a high probability of delivery even with retransmissions. In fact, we would ideally like to have different levels reliability depending on the criticality of data.

Similarly for mobile ad-hoc networks different multipath extensions to well known routing algorithms have been proposed [5], [6], [7]. However, the main purpose of these multipath routing schemes is to increase the robustness, by quickly recovering from broken paths. They are also not adaptive to local channel error.

The idea of probabilistic flooding has been extensively used in ad-hoc networks [19], [22], [23]. However the purpose is to reach all nodes with a high probability at low cost. Our purpose is to provide probabilistic guarantees in reliability of packet delivery to a specific node (the sink).

Different levels of reliability can also be provided using adaptive forwarding error control schemes [8], [9], [17]. However these require complicated dedicated hardware or computation which might not be suitable for current configuration of sensor nodes. A notion of probabilistic reliability is also given in [20] to achieve reliable multi-cast in ad-hoc networks using gossip-based algorithms.

In [21] the wireless broadcast is used to selectively forward packets to a subset of neighbors to implement multicast in ad-hoc networks. In HHB and HHBA, the broadcast acts as automatic redundancy in forwarded packet. The HHBA protocol described in this paper is also used in a subsequent work [16] as the underlying reliable transmission scheme for minimum power conservation in multi-hop wireless networks.

10. CONCLUSIONS

Different information has different level of importance to the end-user. Using the information content in the packets, sensor networks can determine the desired assurance level required for the packets and expend their resources accordingly. This ability to provide different assurance levels to packets based on their information content is termed Information Assurance and is a must for efficient functioning of sensor networks.

In this paper, we described the hop-by-hop schemes for information assurance when the assurance levels are given by the desired reaching probability. We established that using the broadcast property of the wireless channel, we could provide the desired end-to-end reliability at a minimal cost. We showed that the optimal allocation of reliabilities at each hop (such that the some desired end-to-end reliability is met), is independent of the network parameters. In fact, the optimal allocation is simply to have a first-hop reliability equal to the desired end-to-end reliability and at all subsequent hops forward any packet with a reliability of 1. Our results also demonstrate that if possible, the network should use an acknowledgment based scheme to attain reliability, over a non-acknowledgment based scheme.

We expect this preliminary investigation on the effect of various network parameters for delivering packets with multiple levels of reliability would serve as a guideline for designing data-dissemination protocols in sensor networks.

11. ACKNOWLEDGMENT

This work was supported in part by DARPA under contract number N-666001-00-1-8953 and NSF grant ANI-0240383.

12. REFERENCES

[1] G. Asada, T. Dong, F. Lin, G. Pottie, W. Kaiser and H. Marcy, “Wireless Integrated Network Sensors: Low Power Systems on a Chip”, In European Solid State Circuits Conference, The Hague, Netherlands, October 1998.

[2] S. Bhatnagar, B .Deb, and B. Nath, “Service Differentiation in Sensor Networks”, In Proc. of WPMC 2001.

[3] D. Ganesan, R. Govindan, S. Shenker and D. Estrin, “Highly-Resilient, Energy-Efficient Multipath Routing in Wireless Sensor Networks”, Mobile Computing and Communications Review (MC2R) Vol 1., No. 2, 2002.

[4] C. Intanagonwiwat, R. Govindan and D. Estrin, “Directed Diffusion: A Scalable and Robust Communication Paradigm for Sensor Networks”, In Proc. of MOBICOM, 2000.

[5] A. Nasipuri and S. Das, “On-Demand Multipath Routing for Mobile Ad Hoc Networks”, In Proc. of the 8th Annual IEEE ICCCN, Boston, MA, October 1999.

[6] M. Marina and S. Das, “On-Demand Multipath Distance Vector Routing in Ad Hoc Networks”, In Proc. of IEEE ICNP, 2001.

[7] M. Pearlman, Z. Haas, P. Sholander and S. Tabrizi, “The Impact of Alternate Path Routing for Load Balancing in Mobile Ad-Hoc Networks”, In Proc. of the ACM MobiHoc, 2000.

[8] A. Albanese, J. Blomer, J. Edmonds, M. Luby and M. Sudan, “Priority Encoding Transmission”, IEEE FOCS, 1994.

[9] A. Mohr, E. Riskin and R. Ladner, “Graceful Degradation Over Packet Erasure Channels Through Forward Error Correction”, In Proc. of DCC, 1999.

[10] F. Stann and J. Heidemann, “RMST: Reliable Data Transport in Sensor Networks”, In Proc. of International Workshop on Sensor Net Protocols and Applications, 2003.

[11] S. Banerjee and A. Misra, "Minimum Energy Paths for Reliable Communication in Multi-hop Wireless Networks", In Proc. of the ACM MobiHoc, 2002.

[12] L. Kleinrock and J. Silvester, "Optimum Transmission Radii for Packet Radio Networks, or Why Six is a Magic Number." National Telecommunications Conference, 1978.

[13] B. Deb, S. Bhatnagar and B. Nath , "ReInForm: Reliable Information Forwarding using Multiple Paths in Sensor Networks", To appear in Proc. of IEEE LCN, 2003.

[14] A. Bhargava, J. Kurose, D. Towley and G. Vanleemput, "Performance Comparison of Error Control Schemes in High Speed Computer Communication Networks", In IEEE Journal of Selected Areas in Communications, Vol.6, No. 9, 1988.

[15] K. Arisha, M. Youssef and M. Younis, "Energy-aware TDMA-based MAC for Sensor Networks", In Proc. of IEEE IMPACCT, 2002.

[16] B. Deb, B. Nath, G. Levin, S. Li, S. Samtani, "Analysis of Node Scheduling Schemes for Power Conservation in Multi-Hop Wireless Networks", Submitted for Publication.

[17] E. Shih, B. Calhoun, S. Cho and A. Chandrakasan, "Energy-Efficient Link Layer for Wireless Microsensor Networks", In Proc. of WVLSI '01, Orlando, Florida, April 2001.

[18] D. Eckhardt and P. Steenkiste, "A Trace-based Evaluation of Adaptive Error Correction for a Wireless Local Area Network", Mobile Networks and Applications, vol. 4, 1999.

[19] Y. Sasson, D. Cavin and A. Schiper, "Probabilistic Broadcast for Flooding in Wireless Mobile Ad hoc Networks", In Proc. of IEEE WCNC 2003.

[20] J. Luo, P. Eugster, and J. Hubaux, "Route Driven Gossip:Probabilistic Reliable Multicast in Ad Hoc Networks," Proc. of IEEE INFOCOM'03, April 2003.

[21] C. Chiang, M. Gerla, and L. Zhang, "Forwarding Group Multicast Protocol (FGMP) for Multihop Mobile Wireless Networks", AJ. Cluster Computing, Special Issue on Mobile computing, vol. 1, no. 2, 1998.

[22] S.Y. Ni, Y.C. Tseng, Y.S. Chen, and J.P. Sheu, "The Broadcast Storm Problem in a Mobile Ad hoc Network," Int. Conf. on Mobile Computing and Networking (MobiCom'99), pp. 151-162, 1999

[23] Z. Haas, J. Halpern, and L. Li, "Gossip-based Ad-Hoc Routing", In IEEE INFOCOM, June 2002