Non-uniform Information Dissemination for Sensor Networks

Sameer Tilak, Amy Murphy and Wendi Heinzelman

Computer Science Department
University of Rochester
Rochester, NY 14627
{sameer,murphy,wheinzel}@cs.rochester.edu

Abstract

Future smart environments will be characterized by multiple nodes that sense, collect, and disseminate information about environmental phenomena through a wireless network. In this paper, we define a set of applications that require a new form of distributed knowledge about the environment, referred to as non-uniform information granularity. By non-uniform information granularity we mean that the required accuracy or precision of information is proportional to the distance between a source node (information producer) and current sink node (information consumer). That is, as the distance between the source node and sink node increases, loss in information precision is acceptable. Applications that can benefit from this type of knowledge range from battlefield scenarios to rescue operations. The main objectives of this paper are two-fold: first, we will precisely define non-uniform information granularity, and second we will describe different protocols that achieve non-uniform information dissemination and analyze these protocols based on complexity, energy consumption, and accuracy of information.

1. Introduction

Smart environments are characterized by a large number of distributed sensors that collect environmental information in a distributed manner and distribute that information across wireless links. Typically, sensor networks focus on collecting this information in a single place for analysis, taking into consideration optimizations from local aggregation (e.g., LEACH [3]) or processing data en route to a central location (e.g., MagnetOS [9]). While such central collection is important for many applications, it does not match the requirements of all applications that can exploit sensor networks.

For example, consider a military application with sensors distributed throughout an area collecting information about passing vehicles, air contaminants, land mines, and other environmental data. We assume the sensors can communicate with one another, and a soldier that moves throughout the region can contact any nearby sensor to find out both the state of that sensor, as well as any other information it has collected from the other networked sensors. For this soldier, clearly the events occurring in the immediate neighborhood are most important. For example, it is more critical to know about a land mine nearby than a temperature increase several miles away that may indicate a fire. Nonetheless, it is still important that the soldier have a general overview of the area in order to plan and make appropriate decisions. Similarly, consider a rescue scenario where a team of fire fighters is working to rescue trapped victims. In this case, the fire fighter requires precise information about the immediate surroundings in order to make decisions about using resources to make progress, as well as some global knowledge to plan the path to the victims and the reverse path.

These applications differ from usual sensor network applications in two critical ways. First, the information is not collected centrally, but instead is utilized at several places in the network (e.g., the locations of the individuals). While some sensor network applications accomplish this in a query driven manner, asking a central source for the latest collected information, these applications require continuous updates. Second, the information required at each point in the network is different. Specifically, the necessary precision of information is proportional to the distance between an information producer and an information consumer. In other words, as the distance between the source node and sink node increases, loss in information precision is acceptable. We refer to this as a *non-uniform information* requirement, a new concept we introduce here.

This paper introduces and analyzes several protocols that perform non-uniform information dissemination. As these protocols are intended to run on wireless sensor networks, they must abide by the requirements of that environment, namely to be energy efficient and to have low complexity. The distinguishing feature of this new application class is that we can trade accuracy of disseminated information for energy. Our experimental results clearly show this trade-off for a variety of protocols.

The remainder of this paper begins with a precise characterization of the requirements for non-uniform information dissemination protocols. Section 3 describes the details of several protocols. In section 4, we discuss the implementation details of the protocols within the ns-2 simulator and then present our experimental results. Section 5 describes related work and section 6 presents future work. Finally, we present conclusions in section 7.

2. Design Goals

We propose using the following design goals for sensor network protocols.

- *Energy efficiency*. As sensor nodes are battery-operated, protocols must be energy-efficient to maximize system lifetime.
- *Accuracy*. Obtaining accurate information is the primary objective of a sensor network, where accuracy is determined by the given application. There is a trade-off between accuracy, latency and energy efficiency. In the applications we target, it is acceptable to have information with low accuracy about sensors that are far away, whereas sensors that are close by should have highly accurate information about each other. Because of non-uniform information dissemination, a given sensor will not have all the information from all other sensors at every point in time. Consider a case where sensor S_1 receives every n^{th} packet from another sensor S_2. In this case, S_1 receives the i^{th} packet from S_2 at time t_1 and the $(i+n)^{th}$ packet from S_2 at time t_2, ($n > 0$). Thus, in the interval t_1—t_2, the information S_1 has about S_2 is not accurate. We express accuracy in terms of the difference between actual information and stale information.
- *Scalability*. Scalability for sensor networks is also a critical factor. For large-scale networks, protocols should be distributed. A protocol should be based on localized interactions and should not need global knowledge such as current network topology.

With these design goals in mind, in this paper we present simple deterministic protocols (Filtercast and RFiltercast) and non-deterministic protocols (unbiased and biased protocols) to achieve non-uniform information dissemination. Compared to flooding, these protocols reduce the cost of communication by reducing the number of packet transmissions and receptions. At the same time, these protocols are designed to operate within the application-specific tolerance in terms of accuracy. Our results indicate that these protocols outperform flooding in terms of energy efficiency—the trade-off between energy and accuracy is resolved in favor of energy while keeping the accuracy acceptable by the application. The next section describes the details of each of these protocols.

3. Dissemination Protocols

This section introduces the mechanisms of several protocols that perform non-uniform information dissemination. Similar to traditional sensor networks, every node in the network serves as a source of information to be spread throughout the network. Unlike traditional networks where a specific node serves as the *sink* node, every node in our system receives and stores some information about the status of the other nodes in the system.

We begin our protocol discussion with a traditional flooding algorithm. Flooding achieves *uniform* information dissemination, and serves as a baseline of comparison for the rest of our protocols. Following this, we introduce two new deterministic protocols and analyze two non-deterministic protocols [10].

3.1. Traditional Flooding

In flooding, a sensor broadcasts its data, and this is received by all of its neighbors. Each of these neighbor sensors rebroadcasts the data, and eventually each sensor in the network receives the data. Some memory of packets is retained at each sensor to ensure that the same packet is not rebroadcast more than once. If each sensor broadcasts its data, then with this flooding protocol, every sensor in the network will receive data from every other sensor. Thus, ignoring distribution latency, which is the amount of time required for a packet to travel from the source to the farthest sensor in the network, every sensor has an identical view of the network at every point in time.

While it is possible for some number of packets to be lost due to collisions or transmission errors, the rebroadcasting of every packet by all sensors introduces a certain degree of redundancy to the system, making it robust to lost packets. If we ignore this possible data loss, every sensor has essentially the same high accuracy data from every other sensor in the network. Furthermore, the protocol itself is simple and straightforward to implement.

Unfortunately the simplicity and high accuracy come at the price of high energy expenditure. The massive data replication requires active participation from every sensor in the network, and thus sensors can quickly run out of energy.

3.2. Deterministic Protocols

In analyzing the flooding algorithm, it is apparent that to achieve non-uniform information dissemination, the first approach is to simply transmit fewer packets. The two protocols we introduce here, Filtercast and RFiltercast, achieve just that by deterministic means.

3.2.1. Filtercast As the name suggests, Filtercast filters information at each sensor and does not transmit all the information received from other sensors in the network. Filtercast is based on a simple idea of sampling information received from a given source at a certain rate n, specified as a parameter to the protocol. The lower the value of n, the more accurate the information disseminated by the protocol. When $n = 1$, Filtercast behaves identically to flooding. During protocol operation, each sensor keeps a count of the total number of packets it has received so far from each source, $source_{cnt}$. A sensor forwards a packet that it receives from $source$ only if $(source_{cnt}\ mod\ n) == 0$, then increments $source_{cnt}$. We refer to the constant $1/n$ as the filtering frequency. The intuition is that as the hop count between a source and a sink node increases, the amount of information redisseminated decreases due to the cascading effect of the filtering frequency at each subsequent sensor.

While this reduces the total number of transmissions compared to flooding, the state information maintained at each sensor increases. Specifically, each sensor must maintain a list of all the sources it has encountered from the start of the application and a count of the number of packets seen from each of these sources. As this increases linearly with the size of the network, it may pose some scalability problems.

3.2.2. RFiltercast One potential problem with Filtercast is the synchronization of the packets transmitted by the neighbors. Our intuition is that if we can remove this redundancy, we may be able to increase the accuracy of the protocol without increasing the energy expended.

This idea leads directly to a new protocol we refer to as RFiltercast, or Randomized Filtercast. In this variant of Filtercast, the filtering frequency $\frac{1}{n}$ is still the same for all sensors, but each sensor generates a random number r between $1 \dots n$ and retransmits a packet if $(source_{cnt}\ mod\ n) - r == 0$. Intuitively, this means that each sensor considers a window of size n and will transmit only one of the packets from a given source in this window. So, for a window of size 2, half of the packets will be selected for re-transmission, but instead of always retransmitting the first of the two packets, the sensors that choose $r = 1$ will transmit the first of the two packets while the sensors that choose $r = 0$ will transmit the second of the two packets.

While our intuition was that the same energy would be expended by RFiltercast as for Filtercast, this turns out not to be true. In fact, RFiltercast transmits more packets than Filtercast, but fewer than Flooding, putting its energy expenditure in between the two. While RFiltercast has more transmissions, increasing its energy expenditure, it also has improved accuracy over Filtercast. The crucial point to extract is that RFiltercast should, on average, propagate information faster than Filtercast, leading to more accurate data throughout the network, but RFiltercast will require less energy than flooding.

3.3. Randomized protocols

Even though both RFiltercast and Filtercast are lightweight protocols, they still have some overhead, such an keeping some history of previously seen sources. However, their deterministic nature makes them easy to analyze and their results both predictable and regular. We next describe several probabilistic protocols. In these protocols, when a sensor receives a packet, it chooses a random number and then decides whether to forward the packet or not based on the number chosen. The probabilistic protocols can be classified into two categories, namely biased and unbiased protocols. In the biased protocol, sensors bias their decision about whether to forward a packet based on the location of the source, where packets from close sensors are more likely to be forwarded than packets from distant sensors. In the unbiased protocol, all packets are forwarded with equal probability.

3.3.1. Unbiased Protocol The notion of using probabilities to flood packets throughout a network has been studied previously, but to the best of our knowledge, no studies exist that explore its applicability to applications with non-uniform information granularity requirements. Similar to the deterministic protocols, the unbiased protocol also takes a parameter that affects the accuracy of the forwarding. In this case, the parameter specifies the probability that a packet should be forwarded. In the case of unbiased protocols, this value is the same for each incoming packet.

The main advantage of this protocol is its simplicity and low overhead. As every packet is forwarded only with a certain probability, the protocol results in less communication compared to flooding (when the forwarding probability is less than 1) and thus it can be energy efficient. Also, the protocol has less overhead compared to those previously described in this paper because it does not require source information to be maintained. Also, the packet forwarding criteria is very simple and requires no complex operations. Therefore, this protocol has the potential to scale very well.

To adjust the accuracy of the information throughout the network, the forwarding probability can be tuned according to the application needs. The primary tradeoff, however, is energy for accuracy. In general, as the forwarding probability increases, the behavior converges toward flooding. While our current study considers only constant probabilities, in the future we will look at the possibility for probabilities to be adjusted dynamically to adapt to the current network traffic and the application needs.

3.3.2. Biased Protocol For biased protocols, the forwarding probability is inversely proportional to the distance the packet has traveled since leaving the source sensor. In other words, if a sensor receives a packet from a close neighbor, it is more likely to forward this than a packet received from a neighbor much farther away. To estimate distance between sensors, a sensor examines the TTL (time-to-live) field contained in the packet. If we assume all sensors use the same initial TTL, we can use the current TTL to adjust the forwarding probability for each packet.

The motivation for using TTL is two-fold. First, it indicates the distance between two sensors in terms of a hop count. In most scenarios, this can be used as a rough approximation for physical distance, and therefore is a valid metric for biasing our forwarding approach. Second, estimating distance using TTL has negligible overhead. The computation is straightforward, and because a sensor must decrement the TTL field before forwarding the packet in any case, there is no additional overhead to extract the TTL information from the packet. However, note that TTL does not always indicate the exact distance between two sensors. For example, consider a source node S and a destination node D. It is possible that either due to congestion or collisions, a packet gets dropped along the shortest path and another packet reaches node D via a longer route. In that case, the TTL would give a false estimate of distance. However, in a static network, node D can always maintain its current estimate of the TTL to node S. If it ever receives a packet from S with a higher TTL (meaning a shorter path), it can update its existing value. However, this approach will not work in networks with mobile nodes. We used the TTL-based approach for the biased protocol mainly for its simplicity and energy efficiency.

Similar to the unbiased protocol, this biased protocol requires no additional storage overhead and the protocol itself is completely stateless (note, however, that this does not eliminate the caching of recently seen packets in order to avoid re-broadcasting the same packet multiple times). Therefore, this protocol scales very well.

4. Experimental Study

In order to analyze the protocols described in the previous section, we have developed an evaluation environment within the ns-2 discrete event simulator [7] and implemented all of the protocols. Table 1 lists the major parameters used during our simulations.

Simulation area	$800 \times 800\ m^2$
Transmission range	$100\ m$
Initial-Energy	$10000\ J$
MAC Protocol	802.11
Bandwidth	$1\ Mbps$
Transmit Power	$0.660\ W$
Receive Power	$0.395\ W$
Idle Power	$0.0\ W$
Number of Nodes	100

Table 1. Simulation parameters.

We consider two sensor deployment strategies: uniform and random. In a uniform deployment strategy, sensors are distributed with some regular geometric topology (e.g., a grid). With random deployment, sensors are scattered throughout the field with uniform density. For a battlefield-like scenario, random deployment might be the only option, but with applications such as animal tracking in a forest, sensors may be deployed in a deliberate, uniform fashion.

In order to simulate sensor readings, we divide the simulation into an initialization phase and a reporting phase. During the initialization phase, each sensor chooses a random number between 0 and 100 to serve as its initial sensor reading. During the reporting phase, each sensor increments its reading by a fixed amount at fixed intervals. In the real world, due to correlation among physically co-located sensors, sensors might have a slightly different reading pattern; however, this simulation does provide us with valuable information about the behavior of our protocols under various conditions.

4.1. Traffic Load Study

This study focuses on evaluating the effect of a change in traffic load for both grid and random topologies. In the first set of experiments, we study the effect of varying traffic loads systematically from 5 packets/sec to 1 packet/ 2 sec. The goal of these experiments is to understand the relationship between accuracy, reporting rate, and network capacity for both uniform and non-uniform dissemination scenarios.

Note that in order to calculate accuracy, we find the difference between a sensor's local view of another sensor and the actual value of that sensor. A *view* is essentially the latest data point that one sensor knows about another sensor. This view is then normalized based on distance. The error matrix em_i for a given sensor S_i is given as:

$$em_i = \Sigma_{r=1}^{ranges} 1/n\, \Sigma_{k=1}^{n} abs(R(S_i) - R(S_k)) * w_{ik} \tag{1}$$

$$w_{ik} = 1/eucdist(S_i, S_k) \tag{2}$$

The fist equation shows that for a given sensor we calculate weighted average error across all ranges. We vary the range in terms of distance with a step size of 100 meters. That is, first we calculate the weighted average error for a given sensor with respect to all sensors within d meters from it, where d is varied from 100 to 600 meters. Also note that Euclidean distance is used as the weighing factor so that the higher the distance, the smaller the contribution of error toward overall error. This error calculation describes our non-uniform data dissemination requirement by giving higher weight to errors for data that originated in a close neighborhood and lower weight to errors for data that originated from a distant sensor. It is worth noting that although we refer to this as *error*, because the value of the data at the source increases linearly, it also represents the accuracy of the data.

Our results indicate that with flooding, congestion is a severe problem, and other protocols are less prone to the congestion problem. From the energy-error tradeoffs study, we can see that flooding is the least energy efficient protocol and has the highest error if the network is congested. RFiltercast and the biased protocol are more energy efficient than flooding, and provide low error in most cases. Filtercast and the unbiased protocol are the most energy efficient protocols, but their accuracy is good (low error) only at higher sending frequencies.

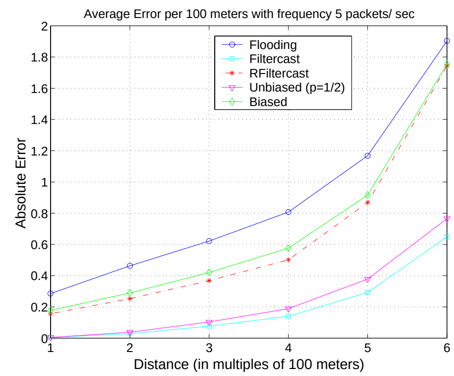

Figure 1. Grid: Average error as a function of distance with data rate 5 packets/sec.

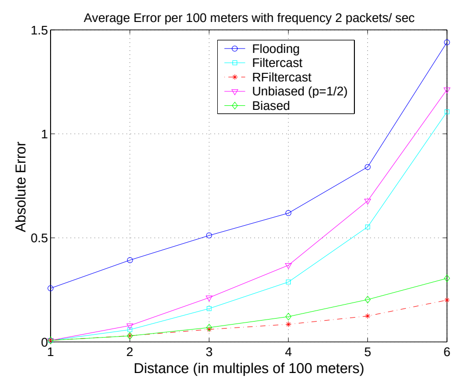

Figure 2. Grid: Average error as a function of distance with data rate 2 packets/sec.

4.1.1. Grid Topology Figures 1, 2, 3, and 4 show the performance of Flooding, Filtercast, RFiltercast, and the biased and unbiased randomized protocols under various traffic loads for the grid topology. In these graphs, distance is varied across the X-axis (in steps of 100 meters) and the Y-axis shows mean error.

From Figure 1, where the data rate is 5 packets/sec, we can see that even though theoretically flooding should have no error, due to congestion, flooding has the highest error. This is due to the fact that if the total traffic exceeds the network capacity, congestion causes packets to be dropped and this gives rise to loss of information and high error. At the same time, high traffic results in higher collisions. In this situation, even RFiltercast and the biased randomized protocol result in high traffic load and thus they have high error as well. However, both Filtercast and the unbiased randomized protocol (with forwarding probability of 0.5) perform well in this case

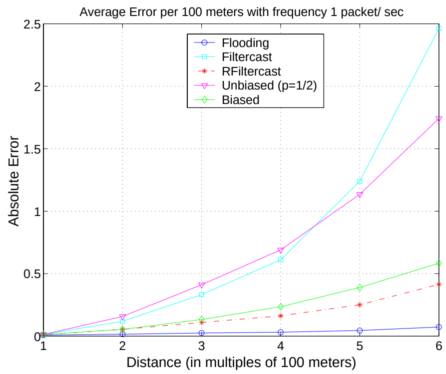

Figure 3. Grid: Average error as a function of distance with data rate 1 packet/sec.

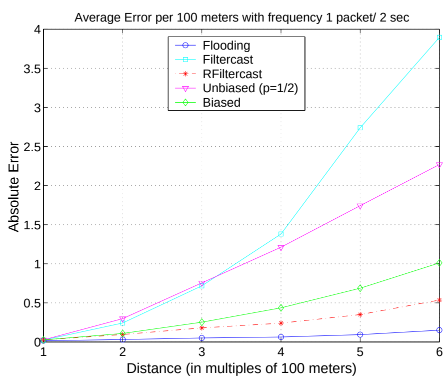

Figure 4. Grid: Average error as a function of distance with data rate 1 packet/2 sec.

because the traffic load does not exceed the available network capacity. As expected, for all protocols the error increases as the distance from the source increases, resulting in non-uniform information across the network.

When the sending frequency is changed to 2 packets/sec, as shown in Figure 2, flooding the network still causes congestion and thus flooding has high error. However, now for both RFiltercast and the biased protocol, the load does not exceed the network capacity and their performance is better than in the previous case. Also, note that now these two protocols perform better in terms of error rate than the unbiased protocol and Filtercast because of the fact that they disseminate more information yet the information disseminated does not exceed the network capacity.

When the sending frequency is lowered to 1 packet/sec, as shown in Figure 3, then even flooding does not exceed network capacity. Since the network is no longer a bottleneck, flooding disseminates the maximum information successfully and clearly has the lowest error. Both the biased and RFiltercast protocols perform better than the unbiased protocol and Filtercast. The unbiased protocol and Filtercast have the highest error in this case because they do not disseminate as much information as the other protocols. The same trend continues even for the lowest sending frequency, shown in Figure 4.

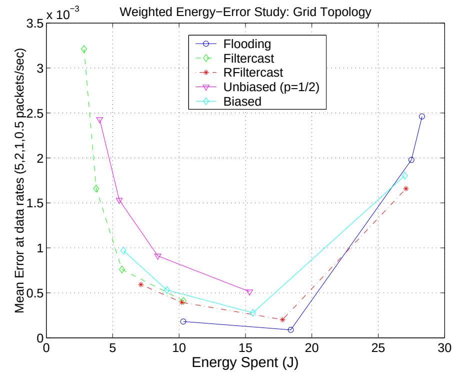

Figure 5. Grid: Weighted energy-accuracy tradeoff.

The interesting point about these results is the oscillatory phenomenon in energy-error trade-off. To elaborate further on this, if the total data exceeds network capacity, then any further data on the channel will increase congestion and decrease overall lifetime of the network. When the amount of data transmitted is below network capacity, then there is a trade-off between energy spent and accuracy observed. This is because as long as the total data does not exceed network capacity, sending more data will improve accuracy at the cost of energy spent in communication. However, with non-uniform information granularity, accuracy between two sensors is proportional to distance between them. Therefore, RFiltercast and Filtercast try to achieve this by filtering packets and randomized protocols try to achieve this by probabilistically forwarding packets.

Figure 5 shows the trade-off between energy and weighted error, using the weighted error calculation method described in Eqns. 1 and 2. In Figure 5, the X-axis indicates the energy spent in Joules and the Y-axis shows mean error.

For flooding, when the reporting frequency is highest, the energy spent is maximum. However, as mentioned earlier, congestion and collisions cause high error. As the sending frequency decreases to the point that total traffic does not exceed network capacity, the error also decreases. Flooding performs the best in terms of accuracy (minimum error) when the sending frequency is 1 packet/sec; after this rate, the error starts increasing due to the fact that not enough information is propagated. This is an interesting phenomenon, where the error oscillates between these two bounds. The upper bound is a function of the network capacity, whereas the lower bound is a function of the application-specific accuracy. Previous research has also shown this phenomenon [11].

Based on the energy-error trade-off, we can say that at high sending frequency, flooding performs the worst by spending high energy while not providing accurate information (high error). RFiltercast and the biased protocol start performing better than flooding at high rates. There is a considerable difference between energy and error for RFiltercast and the biased protocol compared to flooding at the sending frequency of 2 packets/sec. As one can anticipate, flooding performs better than all other protocols when the sending frequency is 1 packet/sec, but note that there is not much difference between flooding, RFiltercast and the biased protocol even when the network is operating in the non-congested mode. Filtercast and the unbiased protocol perform best in terms of energy and error at high sending frequencies and their performance starts to degrade as the sending frequency is reduced.

The desirable mode of operation for a protocol is in the region where minimum energy is spent and low error is observed. Note that the desired mode of operation for a protocol depends on factors such as network density, transmission range of the radios, etc. In our future work, we will perform an analytical study to address this issue. From Figure 5, this zone lies around sending frequency 2 packets/sec to 1 packet/sec for RFiltercast and the biased protocol, whereas for flooding it lies at sending frequency 1 packet/sec.

One might ask the question: with flooding at sending frequency 1 packet/sec, if less energy is spent and lower error is achieved than with RFiltercast and the biased protocol at frequencies 2 packets/sec or 1 packet/sec, then why not just use the flooding at the low sending frequency? The answer is that this choice is application-specific, as in that region the energy/error difference between flooding and RFiltercast and the biased protocol is not much, so the decision is left up to the application designer to make the right protocol choice. Also, as a passing note, we should point out that as the network size increases, flooding can pose severe problems in terms of scalability and energy efficiency. Therefore, randomized protocols should be considered as viable alternatives in these cases. In our experimentation we had a network of 100 sensors, but with a network of thousands of sensors we believe that randomized protocols will perform much better than flooding.

With randomized protocols, the biased protocol performs the best by spending moderate energy and getting high accuracy. Our results show that randomized protocols can achieve high energy savings while at the same time achieving acceptable accuracy with almost no overhead. Also note that RFiltercast and the biased protocol have almost equivalent error curves while the biased protocol is more energy efficient and has negligible overhead.

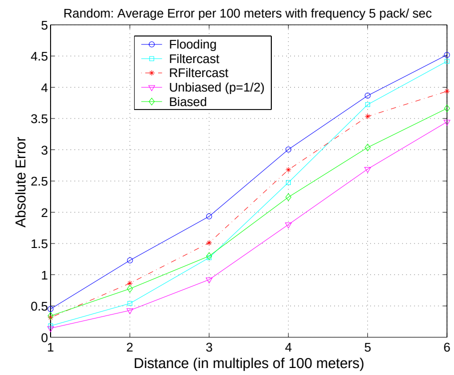

Figure 6. Random: Average error as a function of distance with data rate 5 packets/sec.

We believe that a randomized protocol with very little overhead can be seen as an attractive alternative for flooding or other protocols with large overhead. However, in all cases, the issue of acceptable accuracy should be resolved in an application-specific way.

4.1.2. Random Topology Figures 6, 7, and 8 show our results with a random topology and the same traffic loads as before. Due to space restrictions, we are not adding the results with sending frequency 2 packets/ sec and the energy-error trade-off study, but they show a similar trend to that of the grid topology. It is not clear whether regular deployment will offer advantages over uniformly distributed random deployment; if it does not, random deployment is preferable because of its low cost.

4.1.3. Transmission Range We now change the transmission range of a sensor from the original value of 100 meters to 150 meters. We use the 10x10 grid topology in this experiment. The effect of the increase in transmission range corresponds to an increase in the degree (connectivity) of a sensor. This results in decreasing the capacity of the network, meaning congestion occurs even at low sending frequencies. Intuitively, this will make the overall situation worse if the network is operating in a congested mode. This can be seen from our results, in Figures 1 and 9, as there is an increase in overall error for Flooding, RFiltercast and the randomized protocols.

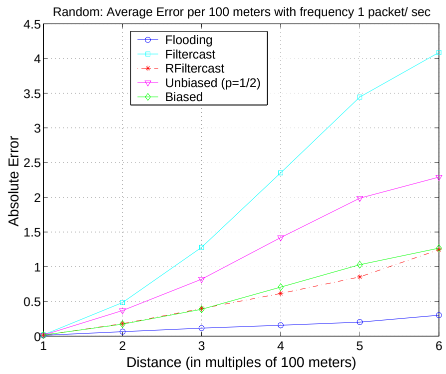

Figure 7. Random: Average error as a function of distance with data rate 1 packet/sec.

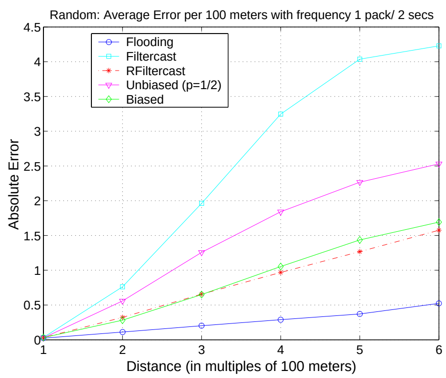

Figure 8. Random: Average error as a function of distance with data rate 1 packet/2 sec.

In this case, even at the low sending frequency of 1 packet/sec shown in Figure 11, flooding does not perform well due to network congestion. Previously, when the transmission range was 100 meters, flooding performed well at this sending frequency (see Figure 3). However, when the network is not congested, then due to the higher connectivity and shorter average hop length, the average error decreases. For example, for RFiltercast and the biased protocol, when the sending frequency is 1 packet/sec, then the maximum absolute error values with a transmission range of 100 meters are 0.4 and 0.6 respectively, as shown in Figure 3. With the transmission range changed to 150 meters, Figure 11 shows that the maximum absolute error for RFiltercast and the biased protocol changes to 0.24 and 0.24 respectively.

Similarly, all the protocols have low error values at a sending frequency of 1 packet/ 2 sec when the transmission range is 150 meters, as shown in Figure 12, compared to the ones with 100 meters transmission range, shown in Figure 4.

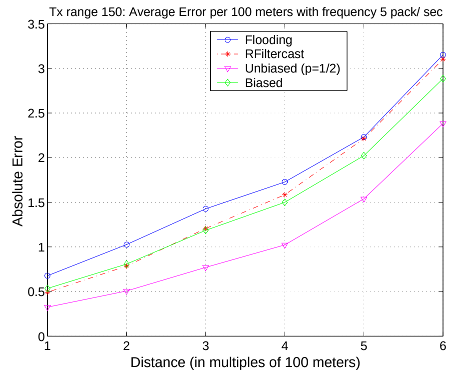

Figure 9. Tx=150 m: Average error as a function of distance with rate 5 packets/sec.

Overall from these results, we can conclude the following: in the case of applications that can exploit non-uniform information, protocols can be designed to make efficient use of the available bandwidth while providing the necessary level of accuracy. Generally, RFiltercast outperforms Filtercast when the network is not congested. Also, naive, randomized protocols such as the unbiased protocol, outperform specialized protocols such as Filtercast. This is because in general with the randomized protocols or the deterministic protocols, the total data that is transmitted remains under network capacity even for high sending frequencies and at the same time these protocols transmit data by dropping packets for far away sensors. In our setting, since accuracy is a function of distance, the errors for far sensors count less and overall these protocols perform well. The biased randomized protocol has comparable performance to that of RFiltercast. Note that both Filtercast and RFiltercast have some overhead to maintain the source list and the count of how many packets a given source node has transmitted. On the other hand, the randomized protocols require no such maintenance. We believe that randomized protocols with intelligent adjustments of forwarding probabilities can be considered as the most efficient alternative for non-uniform data dissemination.

5. Related Work

Recently, sensor networks have drawn a considerable amount of attention from the research community. A number of routing/data aggregation approaches have been proposed [2, 3, 5, 6]. Most of this existing work

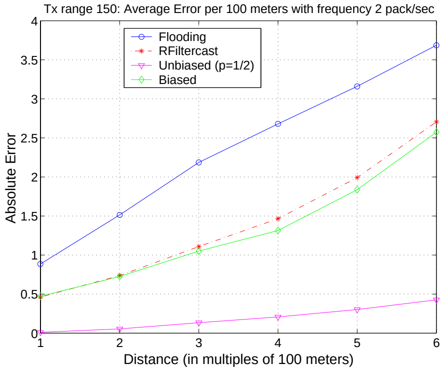

Figure 10. Tx=150 m: Average error as a function of distance with rate 2 packets/sec.

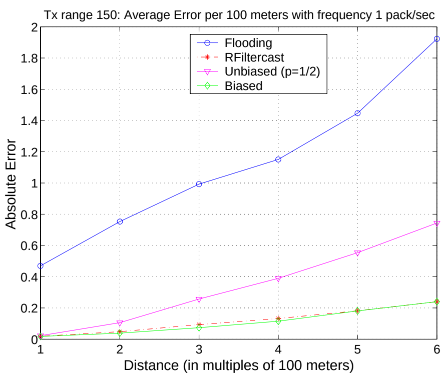

Figure 11. Tx=150 m: Average error as a function of distance with rate 1 packet/sec.

focuses on two primary cases: (1) Sensors send their data toward a centralized base station that has infinite power and is responsible for all data processing, and no *in-network* processing is done. (2) Sensors do some in-network data processing such as data fusion and this high level data is sent to the central base station. In our case, however, we do not assume the presence of any such base-station and the sensors disseminate information among themselves so that the user can connect to any of the sensors to extract network information. This comes from our target application domain described previously.

Other studies considered specific sensor network applications and their implication on protocol design. Cerpa et al. [1], have considered habitat monitoring and have designed protocols to match the application need. Heinzelman et al. [3], described adaptive protocols for information dissemination. In this

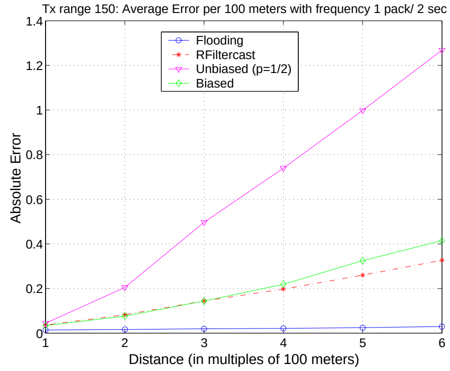

Figure 12. Tx=150 m: Average error as a function of distance with rate 1 packet/2 sec.

work, to save energy, sensors send out advertisements for data they have, and they only send the actual data if it is requested by one or more nodes. Tilak [10] described probabilistic flooding alternatives. However, the main goal of this research was congestion avoidance rather than non-uniform information dissemination. The primary difference between our work and existing work is the application requirement. In our study, we focus on a new application requirement, non-uniform information, and try to analyze protocols for this class of applications. To the best of our knowledge, this is the first paper to focus on this issue.

In the DREAM [8] routing protocol, routing tables are updated based on the distance between two nodes and the mobility rate of a given node. While this work has a similar flavor to our work, exploiting non-uniform information needs, it is limited to only adjusting routing tables and does not apply to the actual data that is exchanged between two nodes.

In this paper we have considered flooding as one of the alternatives for data dissemination in sensor networks. However, flooding and its alternatives have also been explored in the context of mobile ad hoc networks. Perkins et al. describe IP Flooding in ad-hoc networks [4]. While this paper considers probabilistic flooding protocols for sensor networks, Sasson et al. have studied probabilistic flooding for ad hoc networks [12] and used the phase transition phenomenon as a basis to select the broadcasting probability.

6. Future Work

At present, our simulation results are limited to static networks. It would be interesting to evaluate the performance of these protocols in networks with mobile sensors, which result in a continuously changing topology. The interesting question is: can a stateful, deterministic protocol adapt quickly to network dynamics? If not, then how well do the stateless, randomized protocols behave with dynamic networks? We would also like to develop a priority-based protocol where a source marks all its outgoing packets with certain probability to indicate the importance of information contained in the given packet. Any forwarding node can consider priority in the packet when making its forwarding decision. It would be interesting to compare this protocol with the protocols evaluated here.

At present, the data model consists of increasing values of samples. In our future work, we would like to use a model where sample values are random variables. With this randomized model, we can explore other interesting metrics such as reactiveness to events.

In this paper, the distance between two nodes is used as a parameter for non-uniform data dissemination, and we present applications such as a battlefield scenario as a motivating application. However, in our future work, we would like to focus on a broad range of applications with a non-uniform information dissemination requirement, where factors other than distance between two nodes can be used.

7. Conclusion

The main objectives of this paper are to precisely define non-uniform information granularity, and to both describe and analyze a set of protocols that achieve non-uniform information dissemination. We have evaluated the performance of both deterministic (Filtercast and RFiltercast) and non-deterministic (unbiased and biased) protocols under various traffic loads and transmission ranges. With flooding, congestion appears to be a limiting constraint and further, flooding is not generally energy-efficient. Our results indicate that the performance of RFiltercast and the biased randomized protocol is almost equivalent. RFiltercast requires each sensor to maintain some extra state information, whereas the biased randomized protocol is completely stateless and has negligible overhead. Also, we note that the performance of Filtercast and the unbiased randomized protocol is almost equivalent. Again, the randomized protocol is completely stateless and has negligible overhead, while Filtercast requires each sensor to maintain some state information.

We believe that randomized protocols can be attractive alternatives to flooding when 100% distribution of information is not needed by the application. Overall, randomized protocols have less complexity and are very energy-efficient. Furthermore, for randomized protocols one can adjust the forwarding probabilities in application-specific ways to make the energy-accuracy trade-off acceptable to the application.

References

[1] A. Cerpa, J. Elson, D. Estrin, L. Girod, M. Hamilton, and J. Zhao. Habitat monitoring: Application driver for wireless communications technology. In *Proc. ACM SIGCOMM Workshop on Data Communications in Latin America and the Caribbean*, Apr. 2001.

[2] W. Heinzelman, A. Chandrakasan, and H. Balakrishnan. Energy-efficient routing protocols for wireless microsensor networks. In *Proc. 33rd Hawaii International Conference on System Sciences (HICSS '00)*, Jan. 2000.

[3] W. Heinzelman, J. Kulik, and H. Balakrishnan. Adaptive Protocols for Information Dissemination in Wireless Sensor Networks. In *Proceedings of the Fifth Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom '99)*, pages 174–185, Aug. 1999.

[4] IETF MANET Working Group Internet Draft– IP Flooding in Ad hoc Mobile Networks draft-ietf-manet-bcast-00.txt. http://www.ietf.org/proceedings/01dec/I-D/draft-ietf-manet-bcast-00.txt, 2001.

[5] C. Intanagonwiwat, R. Govindan, and D. Estrin. Directed diffusion: A scalable and robust communication paradigm for sensor networks. In *Proc. 6th ACM International Conference on Mobile Computing and Networking (Mobicom'00)*, Aug. 2000.

[6] S. Lindsey, C. Raghavendra, and K. Sivalingam. Data gathering in sensor networks using energey-delay metric. In *Proc. International Workshop on Parallel and Distributed Computing Issues in Wireless Networks and Mobile Computing*, Apr. 2001.

[7] Network Simulator. http://isi.edu/nsnam/ns.

[8] B. S., I. Chlamtac, V. R. Syrotiuk, and B. A. Woodward. A distance routing effect algorithm for mobility (dream). In *Fourth Annual ACM/IEEE International Conference on Mobile Computing and Networking*, 1998.

[9] E. G. Sirer, R. Barr, T. D. Kim, and I. Y. Y. Fung. Automatic code placement alternatives for ad hoc and sensor networks., 2001.

[10] S. Tilak. Sensor, 2002. M.S. thesis- Computer Science department, SUNY Binghamton.

[11] S. Tilak, N. Abu-Ghazaleh, and W. Heinzelman. Infrastructure tradeoffs for sensor networks. In *First ACM International Workshop on Wireless Sensor Networks and Applications*, 2002.

[12] A. e. S. Yoav Sasson, David Cavin. Probabilistic broadcast for flooding in wireless mobile ad hoc networks, 2002.