Non-uniform Information Dissemination for Sensor Networks

Sameer Tilak, Amy Murphy, and Wendi Heinzelman
University of Rochester

July 31, 2003

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

To motivate applications with non-uniform information granularity requirement, 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.

2 Design Goals

In this section we will describe the design goals of the protocols for non-uniform information dissemination.

- *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 the 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 from sensors that are far away, whereas sensors that are close by should have highly accurate information about each other.

- *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. For example, a protocol that requires a given sensor to have knowledge of the topology of the entire network at every point in time will require a lot of communication and will not scale well with an increase in the number of nodes in the network.

3 Dissemination Protocols

This section introduces the mechanisms of several protocols that perform non-uniform information dissemination.

3.1 Traditional Flooding

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

If we ignore possible data loss (due to collisions or congestion), every node has essentially the same high accuracy data from every other node 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. This massive data replication requires active participation from every sensor in the network, and thus nodes can quickly run out of battery power.

3.2 Deterministic Protocols

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, specified as a parameter to the protocol, n. 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 node keeps a count of the total number of packets it has received so far from each source, $source_{cnt}$. A node forwards a packet that it receives from $source$ only if $(source_{cnt}\ mod\ n) == 0$; then the node 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 re-disseminated decreases due to the cascading effect of the filtering frequency at each subsequent node.

While this reduces the total number of transmissions compared to flooding, the state information maintained at each node increases. Specifically, each node must maintain a list of all the sources it has encountered from the start of the application and the 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

Randomized Filtercast is a variant of Filtercast, where the filtering frequency n is still the same for all nodes, but each node generates a random number r between $0 \ldots n-1$, and retransmits a packet if $(source_{cnt}\ mod\ n) - r == 0$. Intuitively, this means that each source node considers a window of size n, and will transmit only one of the packets 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 nodes that choose $r = 1$ will transmit the first of the two packets while the nodes 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.

3.3 Randomized protocols

These protocols are non-deterministic as their forwarding decisions are based on coin tosses. To elaborate further on this, when a node receives a packet, it tosses a coin and then decides whether to forward the packet or not based on the outcome of the coin toss. We next evaluate two probabilistic protocols whose decision making about forwarding packets is determined by a random number generated upon the arrival of each packet, essentially a coin toss.

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 100%) and thus it can be energy efficient.

3.3.2 Biased Protocol

For the biased protocol, the forwarding probability is inversely proportional to the distance the packet has traveled since leaving the source sensor. In other words, if a node 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 nodes, a sensor examines the TTL (time-to-live) field contained in the packet. If we assume all nodes use the same initial TTL, we can use the current TTL to adjust the forwarding probability for this packet.

Similar to the unbiased protocol, this biased protocol requires no additional storage overhead and the protocol itself is completely stateless, as each node does a TTL lookup per packet and no state information regarding the source of a packet is ever stored at any node.

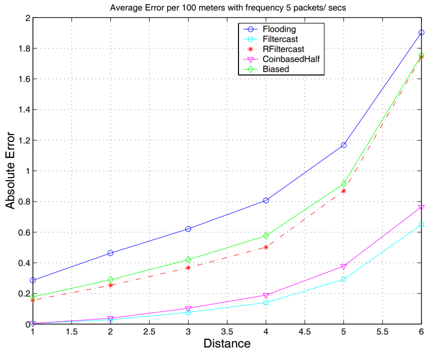

Figure 1: Grid: Average error as a function of distance with data rate 5 packets/sec.

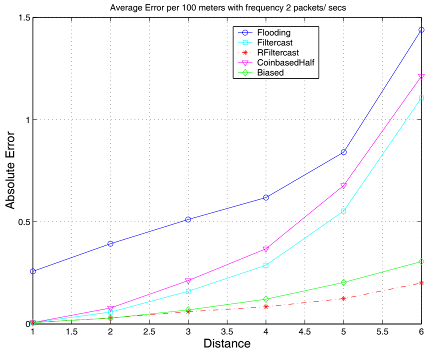

Figure 2: Grid: Average error as a function of distance with data rate 2 packets/sec.

4 Experimental Study

In order to analyze the protocols described above, we have developed an evaluation environment within the ns-2 discrete event simulator [1] and implemented the protocols described in the previous section. For details of the simulation settings, please refer to [2]. We consider uniform (grid-like) and random sensor deployment strategies.

In these 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. To calculate accuracy we find the difference between a sensor's local view of another sensor's data and the actual value of that sensor's data. A *view* is essentially the latest data point that one node knows about another node. This view is then normalized based on distance, so 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.

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.

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 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 pack-

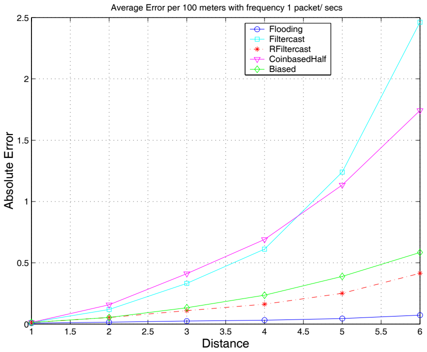

Figure 3: Grid: Average error as a function of distance with data rate 1 packets/sec.

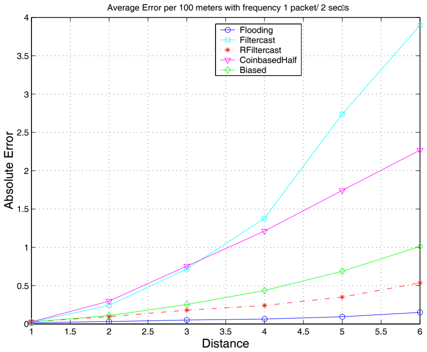

Figure 4: Grid: Average error as a function of distance with data rate 1 packets/2 sec.

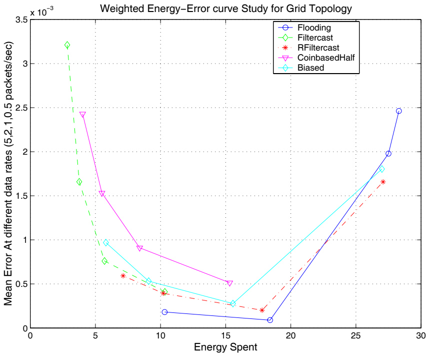

Figure 5: Grid: Energy-accuracy tradeoff.

ets/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.

When the sending frequency is lowered to 1 packet/sec, as shown in Figure 3, then even flooding does not exceed network capacity and it 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.

The interesting point about these results is the oscillatory phenomenon in energy-error trade-off, as shown in Figure 5. 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 life time of the network. When the data is below network capacity, then there is 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 nodes 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.

Due to space restrictions, we are not adding the results with random deployment, but they have a similar trend to that of grid topology.

5 Conclusion

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. We believe that randomized protocols can be attractive alternatives to flooding when 100 % distribution of information is not needed by the application.

References

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

[2] Tilak, S., Murphy, A., and Heinzelman, W. Non-uniform information dissemination for sensor networks. In *Proc. ICNP '03* (2003).