SAFE: A Data Dissemination Protocol for Periodic Updates in Sensor Networks

Sooyeon Kim*, Sang H. Son†, John A. Stankovic†, Shuoqi Li†, Yanghee Choi*

*School of Computer Science and Engineering
Seoul National University
{sykim, yhchoi}@mmlab.snu.ac.kr

†Department of Computer Science
University of Virginia
{son, stankovic, sl7q}@cs.virginia.edu

Abstract

In sensor networks, it is crucial to design and employ energy-efficient communication protocols, since nodes are battery-powered and thus their lifetimes are limited. This paper studies data dissemination in two-tiered networks comprised of stationary sensor nodes and mobile data users who request periodic sensor data updates. We propose a protocol called SAFE (sinks accessing data from environments) which attempts to save energy through data dissemination path sharing among multiple data sinks. Simulation results show that the proposed protocol is energy-efficient as well as scalable to a large data sink population.

1. Introduction

Advances in embedded system technologies motivate the deployment of sensor networks which consist of a large number of sensor nodes scattered over a spacious area. Each sensor node has a processor, memory, and a short-range radio communication facility. These distributed sensing systems enable remote monitoring and event detection in a geographically large region or an inhospitable area. For example, in an explosion area rescuers equipped with handheld devices can be notified of the nearest survivor's location detected by sensor nodes thrown over the area.

Sensor nodes are scattered in a physically spacious area and accordingly powered by batteries instead of being tethered to durable power sources. Generally nodes are assumed to be revoked rather than replenished when they exhaust all the battery power. Previous empirical studies show that the larger portion of power is consumed by communication between nodes [2, 11, 14]. Therefore, in order to expand overall system lifetime, it is crucial to design energy-efficient communication protocols for sensor networks.

In this paper, we investigate data dissemination in a two-tiered network which is comprised of stationary sensor nodes and mobile data users as shown in Figure 1. For example, in an emergency rescue, rescuers might need to monitor a specific area that they are supposed to search, while approaching that area. Desired data updates would be periodic to keep data fresh, and an area of interest might overlap with another. Such data dissemination applications suggest protocol design criteria as the following:

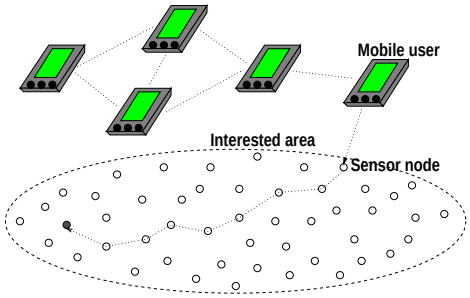


Figure 1. Data dissemination services in sensornets: a two-tiered example.

- Immediate deployment: The protocol should be designed not to require a long-term startup (e.g., network topology construction) after sensor node placement, to get ready for the actual sensor data dissemination.
- Adaptability: The protocol should be scalable to both the number of data sources and the data sink populations, and allow the diversity of user requests in terms of desired update rates and service durations.
- Fast response to data requests: It is desirable that users do not experience a substantial amount of delay after they request sensor data updates.
- Energy-efficiency: Given data update demands, the protocol should be able to satisfy them with lower energy dissipation and ultimately extend the network lifetime.

Although data dissemination protocols such as TTDD [16] and Directed Diffusion [10] have been proposed, they do not pay much attention to the cases where data sources of concern are not known a priori and the actual data updates

should be promptly triggered by data requests. When a large number of nodes have potential to be data sources, it might be a heavy load to construct grid networks per data source as suggested in [16], and infeasible to let every potential data source keep flooding their measurement before any explicit user requests as proposed in [10]. Also, a complicated setup phase like the grid construction for each source [16] and a long-term comparison between multiple data delivery paths [10] might not be suitable when a fast reaction to a new data update request is required.

We introduce SAFE (sinks accessing data from environments), a data dissemination protocol for wireless sensor networks. Using SAFE, individual sensor observations are disseminated to data sinks that explicitly present their interests by sending data requests. Each data sink is allowed to specify its own desired data update rate, and SAFE finds a subscription point through which the sink gets updated, trying to minimize the message transfers in the entire network. Simulation results show that SAFE provides energy-efficiency and fast response, without severe degradation as the number of data sources and data sinks increases.

The rest of this paper is organized as follows. Section 2 describes the environmental model. Then in Section 3 we introduce the proposed data dissemination protocol SAFE, and in Section 4 simulation results are given. Section 5 places our work in the context of related work, and Section 6 concludes this paper.

2. Environmental Model

This section describes the environmental premises that we rely on throughout this paper. First, we introduce the state-of-the-art sensor specification and describe our assumptions on how the sensor network is formed and operated. Then the application scenario is described.

A sensor node is commonly powered by batteries, and equipped with a processor, memory, a radio transceiver, and one or more sensors. An example of such sensor nodes is the Berkeley Motes that employ a lithium battery as the power source, ATMEL 90LS8535 processor, 8 KB flash as the program memory, 512 byte SRAM as the data memory, RF Monolithics 916.50 MHz tranceiver, and options of photo and temperature sensors [7]. Sensor nodes are location-aware, with support of either a GPS (global positioning system) receiver mounted on the sensor node itself or a pseudo-GPS system aided by more powerful nodes. This does not mean that the proposed protocol requires location service, but we believe that such location-awareness expands the applicability of sensor networks.

Using the radio facility, sensor nodes form a wireless ad hoc network based on short range hop-by-hop communication. Routing protocols such as the greedy geographic forwarding [13] and SPEED [4] that exploit the location-awareness of sensor nodes are deployed. We presume that the service availability such as what types of sensor readings are available and which geographic regions can be monitored is known a priori as suggested in [10], that is, there is no need for service advertisements mentioned in [6, 16]. The density of the sensor networks is assumed to be enough such that given a location or an area one or more sensor nodes fall reasonably close to that location/area or inside of the area. An arbitration mechanism is provided for the cases where two or more nodes exhibit the same level of appropriateness, which alleviates mapping a location or an area to an actual sensor node. When the query imposes a consensus on a certain group of sensor nodes (e.g., when a user requests the average temperature of a region), we suppose that the representative node responsible for the data provision on behalf of that group is determined statically or dynamically (an overlay network is one possibility [12]). Thus, hereafter in this paper, it is assumed that each data request arises with a concern to a single sensor node.

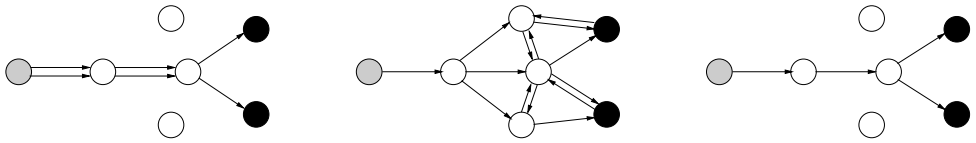
Consider a communication scenario for rescue in an explosion area. In the beginning, an adequately large number of sensor nodes are spread over the entire area. Then rescuers are committed to the area, equipped with a portable communication device such as a PDA (personal digital assistant). Now each rescuer needs to monitor aftershocks and locate survivors in his/her assignment area; A rescuer's query might be like "Keep me updated on carbon monoxide measurement in the parking lot" or "Let me know if a survivor is detected in 300m around my current location".

For that kind of application, a conceivable communication pattern would be that stationary sensor nodes discover the desired data on behalf of mobile users in their vicinity, as previously shown in Figure 1. This paradigm still works even when the geographical distribution of mobile nodes does not retain the soundness of network connectivity or neighboring mobile nodes are not cooperative to relay data. Also, this two-tiered approach is advocated by previous work [6] which points out that sink nodes can be used to improve remote access to sensor data by connecting sensor networks to other networks through themselves.

Although secure communication between nodes is essential in certain applications, this paper does not deal with security issues. We suppose that the communication protocol of the sensor nodes is carefully tuned to bear a sufficient level of security in advance of node placement, considering the level of security threat. Finally, this paper assumes that any message transferred from one node to another can be overheard by any intermediate nodes between them, without any security violation.

3. The Proposed Protocol

Figure 2 illustrates the intended data dissemination path sharing between multiple data sinks, comparing with unicast and flooding. The grey node is the data source of in-



(a) Unicast. (b) Flooding. (c) Intended.

Figure 2. Data dissemination path sharing between multiple data sinks.

```
recvQuery (q)
1  if isRecentlyDealtWith (q)
2     then return
3  saveQueryAsRecentOne (q)
4  if isSource (q)
5     then sendPathSetup (sender(q))
6  else if isJunction (q)
7     then sendJunctionInfo (sender(q))
8  else if isApproachingToSource (q)
9     then forwardQueryToNextHop (q)
```

Figure 3. How to deal with a query arrival: a functional description.

terest, and the black nodes are data sinks that need sensor data updated by that grey node. Instead of updating data on a per-node basis or flooding an update message to the entire network, we attempt to aggregate data delivery paths for a group of sinks close to one another. This section explains the details of the proposed protocol by describing two major phases: query transfer and dissemination path setup.

3.1 Query Transfer

A *query* describes an interest in a certain series of sensor data measured at a remote location, by specifying the location, the sensor data type, the desired data update rate, and possibly the service duration. For instance, the aftershock monitoring mentioned in Section 2 might be specified as:

```
area = [1850, 2150, 60, 900]
attribute = carbon monoxide
interval = 0.5 second
duration = 70 seconds
```

When a sensor node receives a query from a user about a remote location, the sensor node transfers the query to its neighbors via one-hop broadcast. Such a node that works as a representative of actual data consumers is referred to as *data sink* or simply *sink*.

Every node is supposed to maintain the recent query information and a data management table where each entry is identified by a tuple of (*location*, *data type*) as illustrated in Table 1. Receiving a query transferred from another node, a sensor node executes a function described in Figure 3. At first, the node checks if the same query has recently been dealt with. If so, the new query arrival is simply ignored to avoid wasting resource. Otherwise, the node saves the query into its recent query information repository, and then does appropriate things depending on its status regarding the requested data.

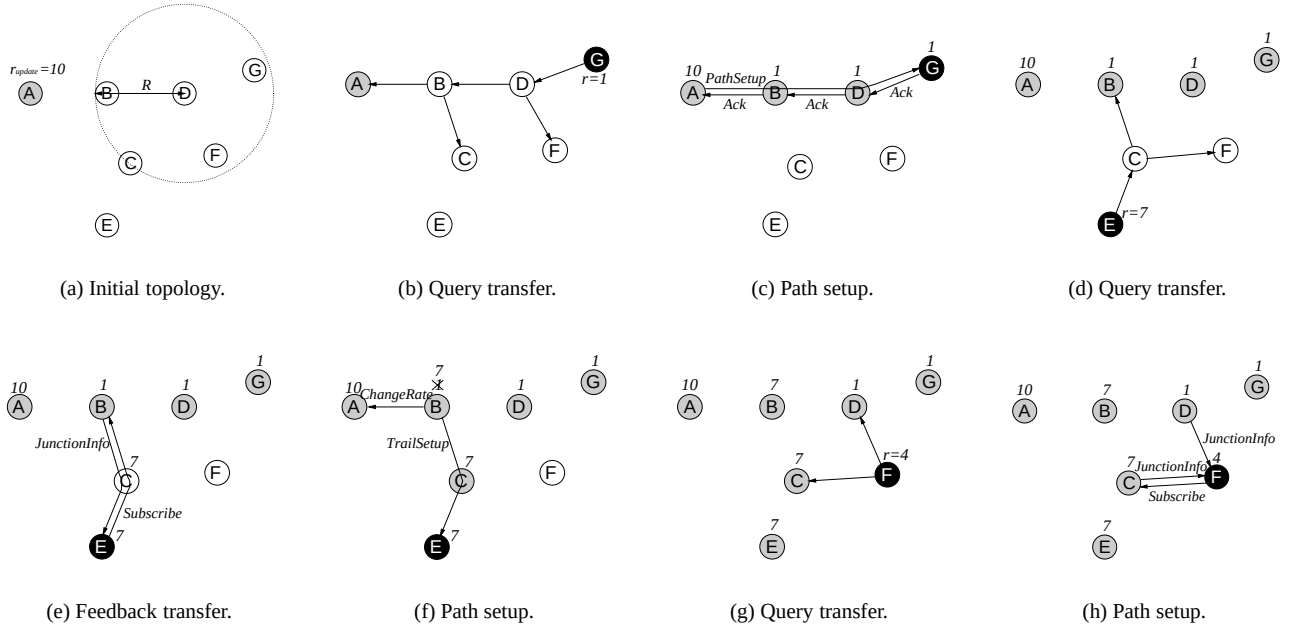
When the corresponding entry in the table shows that the node is the data source, the node sends a *PathSetup* message to the inquiring node via unicast. Figure 4 illustrates an example. Nodes G, E, and F request sensor data originating from node A. When node G sends a query, the query eventually reaches the data source, node A, since there are no dissemination paths previously established. Then node A sends a PathSetup message to node G. If the node is not the source but on a dissemination path, like node B in Figure 4(e), it sends a *JunctionInfo* message to the sink via unicast. We call such nodes *junctions*, which are being fed with the desired sensor data, but not the data source. When the node is neither the data source nor a junction, it forwards the query to the next hop, as long as it is not farther away from the queried location than the last hop sender. In Figure 4(b), nodes C and F do not forward G's query because they are not closer to the data source A than the last hop sender B and D, respectively. The hop sender information might be extracted from the packet header filled by the routing protocol in use, or injected by this data dissemination protocol before forwarding a query.

3.2 Dissemination Path Setup

While a PathSetup message is delivered to the data sink via unicast, all the intermediate nodes on the route overhear the PathSetup message and follow the steps described in Figure 5. When an intermediate node does not have the corresponding entry in its data management table, it creates a new entry with the hop sender information. We refer to the hop sender as the *progenitor*, because it is in charge of conveying the requested sensor data to this node when the path is activated. Once having an entry for the PathSetup message that has just passed by, each intermediate node starts a timer that waits for an *Ack* message from its descendant, which confirms the path is activated. The duration of the

Table 1. Data Management Table Maintained at Each Sensor Node.

location	type	value	timestamp	available interval	service interval	state	source?
[70, 80, 450, 460]	temperature	80°F	41.0 sec	1.0 sec	-	DATA_FED	YES
[100, 110, 60, 70]	temperature	315°F	40.5 sec	3.0 sec	-	DATA_FED	NO
[30, 40, 40, 50]	temperature	-	-	-	-	QUERY_SENT	NO
[230, 240, 0, 10]	acoustic	20dB	41.2 sec	0.1 sec	0.5 sec	SERVING	NO



(a) Initial topology. (b) Query transfer. (c) Path setup. (d) Query transfer.

(e) Feedback transfer. (f) Path setup. (g) Query transfer. (h) Path setup.

Figure 4. How the proposed data dissemination protocol SAFE establishes data delivery paths.

timer is set as a pessimistic estimation of round trip time to the data sink, say, several times the network diameter. This timer prevents memory waste at irrelevant sensor nodes, by releasing the memory occupied by that entry when the timer expires.

In contrast to PathSetup messages, JunctionInfo messages do not build dissemination paths while being transferred to data sinks. This is based on a presumption that given that a junction happens to be in the vicinity of a data sink, there might also exist other junctions in that area. The dissemination path from a junction to a data sink is established only after the data sink subscribes to that junction. A data sink compares every feedback (either a JunctionInfo or a PathSetup), during a certain amount of time [1] after the first feedback received. When we attempt to minimize message exchanges over the network, the best subscription locus for a sink is one that can update the sink with the smallest number of extra messages. We define the messaging overhead as *subscription cost*, and with an approximation that figures out the cost C of a junction j when a data sink m who wants sensor data updates from s through j

$$C(m,s,j) = \begin{cases} d(s,j)\cdot(r_m - r_j) + d(j,m)\cdot r_m & \text{if } r_m > r_j \\ d(j,m)\cdot r_m & \text{otherwise} \end{cases}$$

where $d(a,b)$ quantifies the hop distance from node a to b, and r_a denotes the sensor data update rate requested by and thus available to node a.

When the timer has expired, a data sink subscribes to the node that sent the best feedback until then. If the best one is a junction, the sink sends a *Subscribe* message to that junction. Otherwise, when the data source is eventually the best subscription point, the sink sends an Ack to its progenitor and every progenitor acknowledges its progenitor in turn until the data source gets an Ack message. If a junction receives a Subscribe message from a data sink two or more hops away from itself, then the junction sends a *TrailSetup* message to that sink and establishes the dissemination path. This path enforcement procedure has two

[1] For example, in our simulations introduced in Section 4, this value was set as five times the network diameter.

```
recvPathSetup (p)
 1  if destination (p) ≠ myAddr
 2    then if noEntryInDataManTable (p)
 3           then e ←createEntry (p)
 4                waitForAckFromSink (e)
 5  else /* if the PathSetup p is destined for this node */
 6    then e ← findEntry (p)
 7         if currState (e) = QUERY_SENT
 8           then sendAck (hopSender (p))
 9                changeMyState (e, SUBSCRIBE_SENT)
10         else if currState (e) = FEEDBACK_RCVD
11           then if bestFeedbackCost (e) > cost (p)
12                then saveAsBestFeedback (p)
```

Figure 5. How to deal with a PathSetup arrival: a functional description.

purposes. First, it considers the asymmetry of low-power wireless communication [3], establishing the data dissemination path in the direction of actual data updates instead of simply using the upstream path the Subscribe message has traveled along. Second, it makes the subscription status management at each node more scalable, because a node's potential subscribers are restricted to its immediate neighbors.

A *ChangeRate* message is transferred to the progenitor when the rate of data updates that a node receives should be increased or decreased. As an extreme, a ChangeRate message with the requested update rate of zero means unsubscribe from the data dissemination service.

4. Performance Evaluation

This section presents some simulation results that evaluates the performance of SAFE.

4.1 Metrics and Methodology

We implemented the SAFE protocol in GloMoSim [1]. Our simulations use the 802.11 MAC layer that GloMoSim implements, and SPEED [4], a variant of geographic forwarding, as the routing protocol. We adopt the radio energy model of an actual sensor prototype [7], which states that the energy dissipation is 1μJ for a single bit transmission and 0.5μJ for one bit reception.

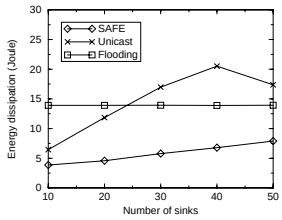
Two simple protocols are implemented as baselines to compare with SAFE: *unicast* and *flooding*. Using unicast, every data source serves each data sink on a per-node basis. With flooding, any queried data source broadcasts the desired data to its neighbors with the update rate of the maximum desired update rate specified by the data sinks, and each node who receives a new data update relays it to the next hop. Note that in this work we modify classic flooding to forward the received data, only when the data is fresher than ever, which averts the implosion problem mentioned in the literature [6].

We employ four metrics for the performance evaluation of SAFE. *Overall energy dissipation* is the total energy dissipation of the entire network. Note that in this work we focus on the power expenditure induced by communication, and thus all the dissipation data to be shown do not include the amount of energy consumed by data processing, memory accesses, etc. *Energy dissipation per effective data update* measures the ratio of total dissipated energy over the whole network to the total number of distinct and meaningful data update messages received by data sinks. We consider that a new data update message arrival is meaningful and effective, when the new message is fresher than the most recent one in terms of sensor observation time at the origin. *First turnaround time* quantifies the elapsed time between query transfer and the first feedback arrival. This indicates the responsiveness of the data provision mechanism, which is significant when the user mobility is high and accordingly a late data provision might be useless. Finally, *data update success rate* measures the ratio of the number of effective data updates received by data sinks, to the total number of data updates expected by data sinks.

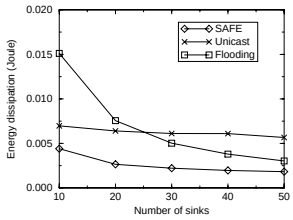
In the following simulations, 100 sensor nodes form a grid network over a 2500m$\times$2500m sensor terrain. The distribution of inter-arrival times follow the exponential distribution (μ=1-5 sec). The size of a message is 64 bytes, and every point plotted is the average of 10 runs with the 90% confidence intervals ranging from 0% to 16% of the mean.

4.2 Simulation Results

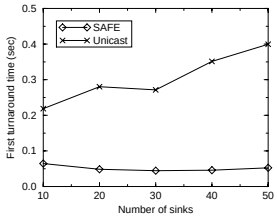
Figure 6 depicts a simulation result with one data source and different numbers of data sinks from 10 to 50. Desired update intervals are fixed at 3 seconds. As expected, flooding shows a constant level of overall energy expenditure regardless of the data sink population as depicted in Figure 6(a), and unicast also keeps even in terms of per-update energy consumption as shown in Figure 6(b). Flooding is profitable with a large data sink population, but extravagant with a smaller number of sinks, and unicast never consumes too much, but does not scale well as the number of sinks increases. For all the data sink populations tested, SAFE always outperforms the two baselines in total and per-update energy dissipation at the same time, and spends less energy per data update with larger sink populations due to dissemination path sharing. SAFE uses a factor of 1.8 to 3.6 less overall energy than flooding in Figure 6(a), and a factor of 1.6 to 3.4 less energy per update than unicast in Figure 6(b). Figure 6(c) introduces how fast a data sink receives the first data update. Flooding is not depicted, because with flooding all the sinks except the first one experience zero response time and the first turnaround time on



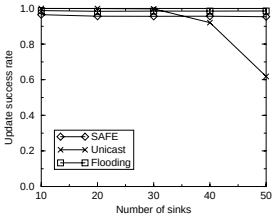
(a) Overall energy dissipation.



(b) Energy dissipation per update.

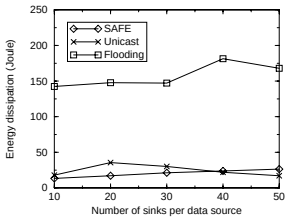


(c) First turnaround time.

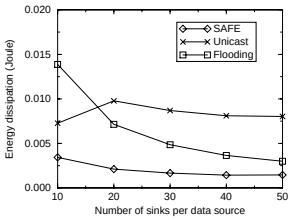


(d) Update success rate.

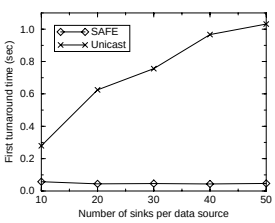
Figure 6. Single data source, varied number of data sinks, fixed desired update intervals.



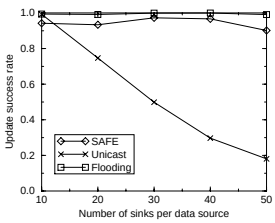
(a) Overall energy dissipation.



(b) Energy dissipation per data update.



(c) First turnaround time.



(d) Update success rate.

Figure 7. Two data sources, varied number of data sinks, varied desired update intervals.

average is not a meaningful factor. SAFE always exhibits fast response time regardless of background traffic volume due to distributed query processing at junctions. Also SAFE retains a reasonable level of update success rate, while unicast fails to maintain the success rate when the number of sinks reaches 50.

The results from the second set of simulation with two active data sources are presented in Figure 7, where desired update rates are varied between 1 second and 5 seconds. Figure 7(a) shows that SAFE reduces overall energy consumption by a factor between 6.3 and 10.6 over flooding, and Figure 7(b) presents that SAFE lowers energy dissipation per update by a factor from 2.1 to 5.7 over unicast. Note that the low overall energy dissipation of unicast in Figure 7(a) results from severe update failures due to network congestion depicted in Figure 7(d). In terms of the first turnaround time, the effect of traffic volume increases is ignorable when using SAFE, while unicast causes the response time to drastically increase.

5. Related Work

Communication protocols in sensor networks have been extensively developed in recent years. Previous work on data dissemination [6, 9, 16] encourages us to investigate sensor data provision of a large population. The DataSpace [9] points out that the physical space monitoring using networked sensors contrasts with traditional databases, in a sense that a database locally stores information about remote physical objects while in sensor networks data are inherently dispersed with the physical object and retrieved via queries transferred to the object through the network. It also envisions that the querying and monitoring the physical space may rely on multicast mechanisms, which is consented by Directed Diffusion [10]. SPIN [6] studies efficient data dissemination that delivers each individual sensor observation to all the nodes in the network, and proposes the use of meta-data (high-level data descriptor) to avoid redundant data provision.

Directed diffusion [10] and TTDD [16] study more realistic problems that only a certain subset of nodes are interested in specific sensor data. An event detection architecture is introduced in [10], which consists of two phases, a low-rate interest flooding and the actual data feed. TTDD [16] considers sink mobility, by constructing grid networks for each data source and selecting a grid node as the communication portal of mobile data sinks.

While previous research on data dissemination in sensor networks deals with such prominent issues, the proposed protocol SAFE is distinguished in that it is totally data consumer initiated, that is, there will be no communication overhead imposed without needs to monitor remote locations. We believe that this feature is crucial, when a huge selection of sensor data is provided and the data dissemination service should be available throughout a fairly

large area. Another point that makes SAFE differ from previous work is that SAFE considers service differentiation between data sinks, allowing each data sink itself to specify the desired data update rate. This aspect entails multiple-level provision of data freshness, possibly according to user importance (e.g., commanding center of a rescue team) or subscription classes of different service charges.

Data diffusion [5, 10], geographical adaptive fidelity (GAF) [15], sentry service [8] suggest that power conservation can be achieved by excluding extraneous nodes from data relaying. We believe that this kind of service should be done as the groundwork preceding actual data communications unlike per-source basis grid network construction proposed in [16], to lessen the overhead of maintaining the network topology information.

6. Conclusions and Future Work

This paper introduces a data dissemination protocol that attempts data delivery path sharing between multiple data sinks whose desired data updates might be different. Simulation results show that the proposed protocol achieves energy efficiency as well as scalability, both of which are crucial for large-scale battery-powered sensor networks.

Currently we are investigating the *timeliness* of data updates, which would be a significant aspect where data freshness is an issue. One of the future extensions of SAFE would be data aggregation that accumulates multiple data provision into a single hop-by-hop transfer. We envision that it would be beneficial for heavy traffic cases with a large number of active data sources, since the data aggregation might reduce the network contention and thus lower the energy expenditure and the first turnaround time while retaining the update success rate acceptable.

Acknowledgements

This work was supported in part by the Brain Korea 21 project of Korea Ministry of Education, the National Research Laboratory project of Korea Ministry of Science and Technology, DARPA grant F33615-01-C-1905, and NSF grants IIS-0208758, EIA-9900895, and CCR-0098269.

References

[1] L. Bajaj, M. Takai, R. Ahuja, K. Tang, R. Bagrodia, and M. Gerla. GloMoSim: A scalable network simulation environment. UCLA Computer Science Department Technical Report 990027, May 1999.

[2] B. Chen, K. Jamieson, H. Balakrishnan, and R. Morris. Span: An energy-efficient coordination algorithm for topology maintenance in ad hoc wireless networks. In *Proc. of the Seventh Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom 2001)*, Rome, Italy, July 2001.

[3] D. Ganesan, B. Krishnamachari, A. Woo, D. Culler, D. Estrin, and S. Wicker. An empirical study of epidemic algorithms in large scale multihop wireless networks. Technical Report IRB-TR-02-003, Intel Research, March 2002.

[4] T. He, J. Stankovic, C. Lu, and T. Abdelzaher. SPEED: A stateless protocol for real-time communication in sensor networks. In *Proc. of the 23rd International Confernece on Distributed Computing Systems (ICDCS-23)*, Providence, RI, USA, May 2003.

[5] J. Heidemannn, F. Silva, C. Intanagonwiwat, R. Govindan, D. Estrin, and D. Ganesan. Building efficient wireless sensor networks with low-level naming. In *Proc. of the Symposium on Operating Systems Principles*, Lake Louise, Banff, Canada, October 2001.

[6] W. Heizelman, J. Hill, and H. Balakrishnan. Adaptive protocols for information dissemination in wireless sensor networks. In *Proc. of the Fifth Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom '99)*, Seattle, WA, USA, August 1999.

[7] J. Hill, R. Szewczyk, A. Woo, S. Hollar, D. Culler, and K. Pister. System architecture directions for networked sensors. In *Proc. of International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS-IX)*, Cambridge, MA, USA, November 2000.

[8] J. Hui, Z. Ren, and B. Krogh. Power management component proposed by CMU. http://www.andrew.cmu.edu/∼zren/nest_challenge_power_management.htm, 2002.

[9] T. Imielinski and S. Goel. DataSpace: Querying and monitoring deeply networked collections in physical space. *IEEE Personal Communications*, 7(5):4–9, October 2000.

[10] C. Intanagonwiwat, R. Govindan, and D. Estrin. Directed diffusion: A scalable and robust communication paradigm for sensor networks. In *Proc. of the Sixth Annual ACM/IEEE International Conference on Mobile Computing and Networks (MobiCom 2000)*, Boston, Massachusetts, August 2000.

[11] O. Kasten. Energy consumption. http://www.inf.ethz.ch/∼kasten/research/bathtub/energy_consumption.html, 2001.

[12] S. Li, S. H. Son, and J. Stankovic. Event detection services using data service middleware in distributed sensor networks. In *Proc. of International Workshop on Information Processing in Sensor Networks (IPSN'03)*, Palo Alto, CA, USA, April 2003.

[13] M. Mauve, J. Widmer, and H. Hartenstein. A survey on position-based routing in mobile ad hoc networks. *IEEE Network*, 15(6):30–39, November/December 2001.

[14] M. Stemm and R. H. Katz. Measuring and reducing energy consumption of network interfaces in hand-held devices. *IEICE Transactions on Commmunications*, E80-B(8):1125–1131, August 1997.

[15] Y. Xu, J. Heidemannn, and D. Estrin. Geography-informed energy conservation for ad hoc routing. In *Proc. of the Seventh Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom 2001)*, Rome, Italy, July 2001.

[16] F. Ye, H. Luo, J. Cheng, S. Lu, and L. Zhang. A two-tier data dissemination model for large-scale wireless sensor networks. In *Proc. of the Eighth ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom 2002)*, Atlanta, Georgia, USA, September 2002.