Title
Residual Energy Scans for Monitoring Wireless Sensor Networks

Authors
Zhao, Yonggang
Govindan, Ramesh
Estrin, D

Publication Date
2002-03-17

Peer reviewed

Residual Energy Scan for Monitoring Sensor Networks

Yonggang Jerry Zhao
USC/ISI
4767 Admiralty Way, Suite 1001
Marina del Rey, CA 90292

Ramesh Govindan
ICSI
1947 Center Street, Suite 600
Berkeley, CA 94704

Deborah Estrin
UCLA/LECS
Boelter Hall Suite 3440
Los Angeles, CA90095

***Abstract*— It is important to have continuously updated information about network resources and application activities in a wireless sensor network after it is deployed in an unpredictable environment. Such information can help notify users of resource depletion or abnormal activities. However, constrainted by the low user-to-node ratio, limited energy and bandwidth resources, it is infeasible to extract state of each individual node. In this paper, we propose an approach to construct *abstracted scans* of sensor network health by applying *in-network aggregation* of network state. Specifically, we design a residual energy scan which approximately depicts the remaining energy distribution within a sensor network. Simulations show that our approach has good scalability and energy-efficiency characteristics, compared to continuously extracting the residual energy level individually from each node.**

***Index Terms*—Wireless Sensor Network, Aggregation, Network Instrumentation**

I. Introduction

Wireless sensor networks have been attracting increasing research interest given the recent advances in miniaturization, low-cost and low-power design. Such networks will consist of a large collection of small wireless, low-power, unattended sensors and/or actuators [1][2][3][4]. Sensor networks can enable "smart environments" which can monitor ambient conditions such as temperature, movement, sound, light, location and others. One important feature that distinguishes sensor networks from traditional distributed systems is their need for *energy efficiency*. Many nodes in the emerging sensor systems will be *untethered*, having only finite energy reserves from a battery. The requirement for energy-efficiency pervades all aspects of the system design [5]. Another important feature is their unattended and ad hoc nature. Because of their compact form factor and potential low cost, nodes might be autonomously deployed in an unplanned fashion. The working environment for those sensor nodes might be unpredictable and thus affect the performance of the network dramatically. The high node-to-human ratio also makes it infeasible to maintain individual nodes constantly.

Given their unattended nature and their complexity, it is critical that the users be continuously updated of the sensor network *health* indications, i.e., explicit knowledge of the overall state of the sensor network. We propose sensor network *scans* as such indications of network health. A scan can provide early warning of system failure, aid in incremental deployment of sensors, or help testing sensor collaboration algorithms. For example, knowing the remaining energy resource distribution within a sensor field, a user may be able to determine if any part of the network is about to fail in the near future due to depleted energy. Similarly, given the practical difficulties in precisely planning sensor field deployments, network scans can guide incremental deployment of sensors. By examining the distribution of node density, communication quality and other resources in the sensor field, additional sensors can be placed selectively on those regions short of resources. Finally, a sensor scan can be designed to depict the overall response of the sensors to some known stimulus. Such information is helpful for validating expected sensing functionality or fine-tuning detection algorithms.

However, continuously monitoring wireless sensor networks leads to different challenges compared to existing diagnosis protocols for distributed systems [6][7][8][9][10], or other systems such as telecommunication networks or power generation systems [11]. The large number of nodes in a sensor field makes it infeasible, given energy and communication constraints, to collect detailed state information from each individual sensor node and then process centrally. In this paper, we propose an efficient monitoring infrastructure for sensor networks. Analogous to weather map or air traffic radar images, our *sensor network scans* describe the geographical distribution of network resources or activity of a sensor field. We design and evaluate a mechanism for collecting a *residual energy scan* (eScan). A eScan depicts an aggregated picture of the remaining energy levels for different regions in a sensor field, and may look like Figure 1. Instead of the detailed information of residual energy at individual sensors, this scan provides an abstracted view of energy resource distribution.

Our proposed approach to construct an eScan applies localized algorithms in sensor networks for energy-efficient *in-network aggregation* of local representations of scans. Rather than collect all local scans centrally, this technique builds a composite scan by combining local scans piecewise. At each step of aggregation, these partial scans are *auto-scaled* by varying their resolutions. The information content of the overall scan scales well with network size. We also propose to apply *incremental updates* to scans. When the state of a node changes, it should not need to continuously re-send its entire scan. Rather, it only sends a partial update to a scan only when its local state has changed significantly. Furthermore, that update only traverses up the aggregation hierarchy if it radically impacts some aspect of the overall representation. An aggregated scan may lose detailed information such as the residual energy level at each individual node. However, the compactness of such an abstracted representation can reduce the communication and processing cost significantly. As we show in this paper, the trade-off between reduced fidelity and increased lifetime is acceptable.

We evaluate the performance of our design by simulation. To calibrate the performance of distributed scan collection, we compare our scheme to centralized collection of node residual energy. It might seem that this comparison is somewhat trivial—distributed scan collection will obviously be more energy-efficient than centralized collection. However, distributed scan collection and aggregation can introduce error in observed node residual energy. The performance question we ask is: Can distributed scan collection provide significant energy savings while introducing little error? We show that there exist reasonable models of node energy dissipation for which scanning can result in an order of magnitude energy savings while introducing less than 10% error in observed residual energy.

The rest of the paper is organized as follows. In Section II, we give a brief summary of related work. Section III describes the design of residual energy scan collection. In Section IV, our preliminary simulations results show that sensor network scanning is energy-efficient and scales well with network size, compared to collecting all residual energy information centrally. Section V concludes with some future work directions.

II. Related Work

To our knowledge, there exists no ongoing or previous work that has attempted continuous monitoring of large-scale distributed sensor networks. In this section, we review peripherally related areas: wireless sensor networks, debugging and diagnosis protocols for parallel and distributed systems, monitoring other industrial systems and recent work on coverage problem in sensor networks.

Wireless sensor networks can potentially support a variety of high profile applications [4][12]. Researchers have been addressing various aspects of the design of sensor networks [1][13][14]. In data fusion [15][16], information gathered from various knowledge sources and sensors are combined to provide a better understanding of the phenomenon being observed. An energy-efficient paradigm (directed diffusion) for the design of sensor network protocols is proposed in [13]. One important feature of directed diffusion are localized interaction. The control decisions are made based solely on the interactions with neighbors or nodes with some vicinity, which is intended to avoid the high energy cosumption on data delivery over long distance but be more scalable, robust and energy-efficient.

Distributed diagnosis protocols [10][8][17][18] have been designed either for multiprocessor computers or for wired computer networks. Particularly, instrumentation on the growing Internet shares similar scalability challenges [7][19][20].

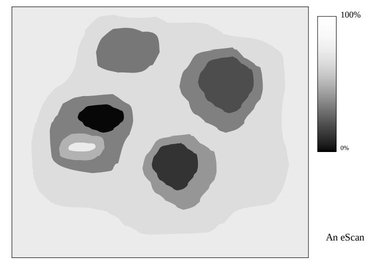


Fig. 1. An Example of Residual Energy Scan

For example, SCAN [7] provides a multicast-based continuous monitoring infrastructure with good scalability and robustness by sharing information between routers. Engineers have been deploying monitoring systems such as SCADA(Supervisory Control and Data Acquisition) [11] for power plants, gas companies, telephony systems and others.

When instrumenting a sensor network, lots of ideas can be borrowed from those techniques. However, none of them is *directly* applicable to continuously monitoring sensor networks which imposes different challenges in scalability and energy-efficiency. For example, states are intended to be extracted from the whole network as a background activity of low priority. The sum of states is accumulatively very large and light-weight in-network aggregation of those states are necessary. More recently, the coverage problem in wireless sensor networks is studied in [21]. Given the locations of sensor nodes, these techniques detect maximal breach path and maximal support path along which there is poorest and best coverage of sensors, respectively. These approaches can detect one or more specific network vulnerabilities. Sensor network scans are complementary in that it can provide an approximate indication of when to invoke these other tools.

III. Residual Energy Scan

A *residual energy scan* (or *eScan*) depicts the remaining energy levels of sensor nodes(Figure 1). Different regions of the sensor field are shaded differently, depending on the average energy resources within that region. An eScan can help users to decide where new sensor nodes be deployed to avoid energy depletion. It may also help verifying the behavior of energy-aware adaptive routing protocols [22].

A. System Model and Assumptions

Without loss of generality, the network we intend to monitor consists N sensor nodes on a m by m square plane. Each node is immobile but has symmetric communications to other nodes within certain range. Location on the plane can be obtained at each node, by GPS with fair accuracy [23], or other ranging localization systems [24]. Nodes are powered by batteries with normalized capacity of 100%. The residual energy level can be measured by interface similar to APM or ACPI. Each node executes one or more sensing tasks**,** consuming energy by inter-node communication and local signal processing. We emphasize the high energy cost of communication compared to computation. For example, reference [5] shows the energy consumed in transmitting a 1 kilobit packet over 100m is approximately the same as performing 3M CPU instructions on prototype wireless sensor nodes. For this reason, sensor nodes will prefer to perform significant local collaborative processing of data, rather than transmit data over long distances.

We also assume that there are one or more *user gateway*s at the network edge, from where users will collect residual energy scans. We intend to design network communication and aggregation mechanisms for delivering energy scans to a user gateway with good *scalability*, *robustness* and *energy efficiency* characteristics.

B. Collecting Residual Energy Scans

The process of constructing a eScan of a sensor field can be briefly described as follows:

Determining Local eScans: Each node constructs its local scan with its residual energy level and its location, and it only reports when the energy level drop significantly since last time it reported its eScan.

Disseminating eScans: Local eScans are disseminated across the network to compute a composite eScan of the entire network. For this to happen, the user at a *gateway* expresses a special INTEREST message. This INTEREST message propagates throughout the network by flooding. Upon receiption of INTEREST message, each node sets the sender as its parent node leading toward the user. An aggregation tree is constructed with root as the user gateway. Local eScans is sent along the tree towards the user.

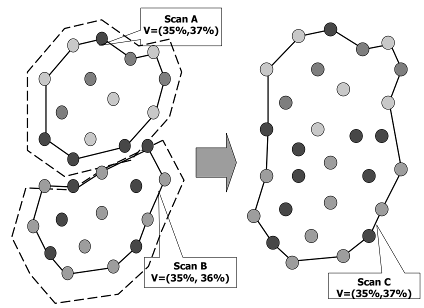
Aggregating eScans: Those nodes that receive two or more eScans may aggregate eScans if the eScans are topologically adjacent and have the same or similar energy level. The *aggregated* scan is a tuple consisting of a polygon that describes a collection of nodes, and the range of residual energy levels at those nodes, which reduces the messaging cost by losing little critical information content in the scans.

There are several interest problems for us to explore in the design space, for example: What is the proper compact representations for scans? How do we aggregate scans to reduce messaging cost? Are there other ways to organize the aggregation paths in terms of energy-efficiency and robustness? Which network characteristics are proper and which are not? In this paper, we focus on designs of representation and aggregation schemes for eScans.

C. Abstracted Representation

A sensor network *scan* represents an *abstracted* view of a particular network characteristics. More precisely, we can define a *eScan* as a collection of (VALUE, COVERAGE) tuples.

Fig. 2. Representation and Aggregation of eScans



VALUE is the quantitative representation of the network state we are monitoring. It may have a more complex form than a single scalar value. In eScan, we use VALUE=(min, max), where min, max are the minimum and maximal residual energy level of the nodes, respectively. For example in Figure 2(a), the eScan VALUE in scan A is (35%,37%).

COVERAGE denotes the region that VALUE describes. In eScan, COVERAGE of a scan is discribed by a polygon, which covers those nodes with energy levels falls in the range of VALUE.min and VALUE.max. The vertices of COVERAGE polygon are the locations of those boundary nodes. The polygon is not necessarily convex, but not self-overlapping. In Figure 2(a) , the coverage polygon of the eScan is shown using a solid line.

The representation of eScan leads to energy savings on messaging cost. Combining locations of the nodes with a region, the polygon representation is more compact. Intuitively, if all N nodes within a square have similar values, instead of a list of N locations, the polygon representation uses a list proportional to $\sqrt{N}$. Information of residual energy at each node and the locations for the interior nodes will be lost. However, it is still very helpful for a user as an indication of energy resource distribution. A user is not necessarily interested in the individual energy levels when monitoring a network in a large number of nodes. The eScan are enough in most case to identify the near-depleted regions or discover energy consumption patterns.

The VALUE and COVERAGE representation for eScan is reasonable for another reason. Over long time scales, the energy consumption pattern is expected to show *spatial locality*. We anticipate that all the nodes within a certain neighborhood participate in the processing of similar events, thereby spend similar energy on the sensing task. If all nodes start out with comparable energy levels, spatial locality can results in good compressibility. There exist alternative representation schemes for eScan and other scans. For example, scans that show ambient sensing activities can be described in a manner similar to isothermal curves.

D. In-network Aggregation of eScans

1) Aggregation Rules: One of the most important characteristics of scans is multiple scans can be combined. For example, eScan A and eScan B can be aggregated if

A.VALUE and B.VALUE are similar

$$\frac{max\{A.max, B.max\} - min\{A.min, B.min\}}{Avg\{A.min, A.max, B.min, B.max\}} \leq T$$

And, A.COVERAGE and B.COVERAGE are adjacent:

$$Distance(A.COVERAGE, B.COVERAGE) < R$$

where T (*tolerance*) denotes the maximum allowed relative error of residual energy value by aggregation. R(*resolution*) decides when two regions are adjacent. Function Distance gives the minimum distance between any pair of points, each from one of the coverage sets. When both conditions are met, the aggregated eScan C can be obtained in the form of

$C.min = min\{A.min, B.min\}$

$C.max = max\{A.max, B.max\}$

$C.COVERAGE = Merge(A, B, R)$

where Merge(P,Q,R) combines two polygons with resolution R. Obviously, $size(C) < size(A) + size(B)$, because Merge only removes vertices from COVERAGE but never adds new ones. An example of aggregation operation is shown in Figure 2(b,c), and the scan size is reduced by removing the location information for 5 nodes.

T and R control how deeply we aggregate scans. They also decide the "fidelity" of outcome scans. In eScan, we usually assign a fixed value to R such as radio communication range or sensing range. The operation to test and aggregate two eScans costs $O(N^3)$ time, where N is the total number of points in two scans. Though aggregation consumes energy on local CPU processing, but such cost is much lower than delivering unaggregated data across the network.

2) Incremental Update: From an energy perspective, incremental updates are necessary for continuously monitoring a sensor network over long time scale. The aggregation operation introduces error to eScans. If the value change is within the aggregation tolerance T, it is not necessary to relay the update along the aggregation hierarchy to the user. Each node maintains a finite eScan cache. New eScan updates are first compared with cached scans. A update will be droped, if it is covered by an old eScan and their values are similar with tolerance T. Otherwise it will be forwarded to the user and invalidate any old scan that shares the same region.

E. Discussion

Section III-C and III-D describes the core of eScan construction: abstracted representation and its corresponding aggregation operations. There are some additional design issues that we have currently been investigating.

Complementary Tools: Critical information may be lost due to abstraction and aggregation. Complementary to sensor network scans is *drill down*, where scans can be used as hints to more thourough diagnosis protocols to identify particular network problems within particular region. The mechanisms for drill down are similar to those for query distribution and response aggregation described in [13].

Aggregation Tolerance Adaptation: To balance the savings in aggregation and the loss of accuracy in scans, each node adaptively adjusts its aggregation operation locally. For example, if a node keeps receiving scan updates, it can increase the aggregation tolerance value to reduce the size of resulted scan. If the node only receives a few escan updates or those updates are very similar to each other, it can reduce the the aggregation tolerance value to generate more detailed scans of residual energy.

Aggregation Path Maintenance: Node failure may partition the aggregation tree thus some nodes are not able to send eScan updates to the user. We propose that tree is maintained by "soft-states" and the user periodically refresh INTEREST messages to adapt node failure and network dynamics. Similar protocols have been well studied in IP routing, IP multicast and others.

All the issues above are important and can not be ignored. However, our experiments in next section shows the key benifits of sensor network scans comes from abstracted representation shceme, in-network aggregation and incremental updates.

IV. Experimental Results

We compare the performance of eScans to centralized collection of individual residual energy information from each node. Our experiments serve more or less a sanity check to reveal the benefits of abstraction and aggregation for large wireless sensor networks. Distributed scan collection with aggregation can introduce error in observed node residual energy. We intend to verify if the communication cost saving is worth by introducing acceptable error. We use a stand-alone C++ package to simulate the eScan construction process. In this section, we present our results and discuss their implications and possible applications.

A. Metrics

The key performance criterion for eScan is the energy consumed in communicating to the user gateway and the error introduced by abstraction and aggregation. There are other important metrics for the performance of network scans. For example, an eScan update arrived at user gateway is actually a delayed view of the sensor network. The latency distortion is another metric of accuracy. We have not studied those metrics yet but leave them for future work.

Messaging Cost: Sensor network applications consume energy from time to time. After each sensing activity *event*, the energy dissipation may be significant enough to invoke an eScan update. We define the messaging cost E_c for continuous monitoring as

$$E_c = \frac{\sum_{\forall n} C_n}{N \times M} \ (bit/node/event)$$

where M is the number of events happening per node, C_n is the sum of messaging cost on collecting eScan updates during this time frame, and N is the network size. We define E_o as the cost of centralized collection of residual energy information without any in-network processing. Compared with E_0, E_c is expected to reveal the potential savings by in-network aggregation. We quantify such savings as **cost ratio** $R = \frac{E_0}{E_c}$: larger R indicates more significant savings.

Relative Distortion: Aggregation introduces error into eScans. We quantify the "fidelity" of an eScan snapshot by the relative root mean square error between perceived residual energy values in an eScan and the actual values:

$$D = \sqrt{\frac{\sum_{\forall n} |\frac{E_n - E'_n}{E'_n}|^2}{N}}$$

where E_n is the estimated residual energy of node n in the eScan and E'_n is the real value. N is the size of the network.

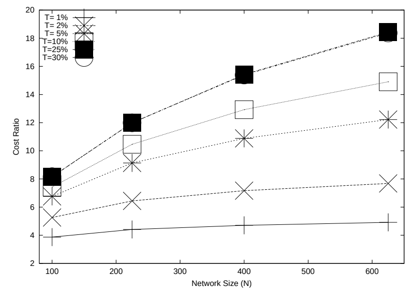
B. Energy Dissipation Model

There is one missing piece when we define those metrics: E_c and D may be sensitive to the energy consumption patterns. How do we model the energy dissipation in a sensor network? To our knowledge, there is no realistic model or empirical data on energy dissipation of large-scale sensor networks. We propose two energy dissipation models that emphasize different types of sensing application activity.

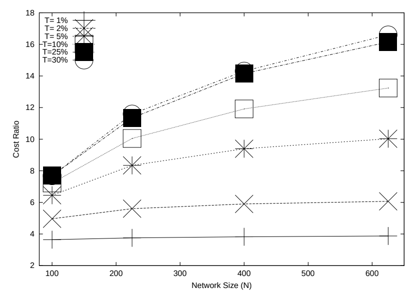
We propose a UNIFORM DISSIPATION model, to capture uniformly distributed sensing activity. More precisely, during a sensing event, each node n in the network has a probability p of initiating a local sensing activity, and every node within a circle of r centered at n consumes fixed amount of energy e. This latter feature of our model is inspired by collaborative sensing algorithms, where triggered by physical activities on the sensor field, the nodes within some vicinity send their data to a particular head node.

In the UNIFORM DISSIPATION model, the residual energy at all nodes decreases at approximately the same rate. In a realistic environment, different regions in a sensor field have different energy dissipation rates. To capture this, we propose a HOTSPOT DISSIPATION model. In this model, there are h hotspots uniformly distributed in random on the sensor field but their locations are fixed during the simulation. Each node n has a probability of $p = f(d)$ to initiate a local sensing activity, and every node within a circle of r centered at n consumes fixed amount of energy e; where f is a density function and $d = min_{\forall i}\{|n - h_i|\}$ is the distance from n to nearest hotspot. We use two density functions: $f(x) = ae^{-ax}$ is the density function for exponential distribution, where the hotspot effect drops quickly with increasing distance; and the Pareto density $f(x) = a/(x+1)^{a+1}$, where the impact of the hotspot falls off more gradually than an exponential distribution with the same value of a.

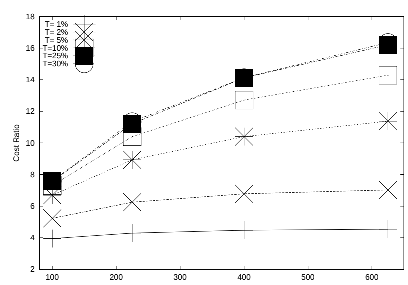
Fig. 3. Messaging Cost Ratio for eScans



(a) Cost Ratio for Uniform Model (p=0.1)



(b) Cost Ratio for Hotspot Model (Exponential, a=0.5)

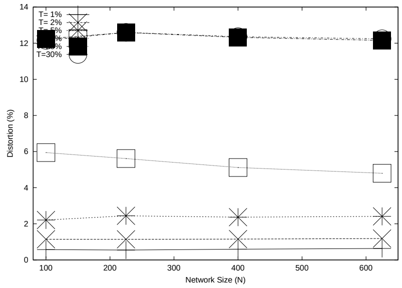


(c) Cost Ratio for Hotspot Model (Pareto, a=0.5)

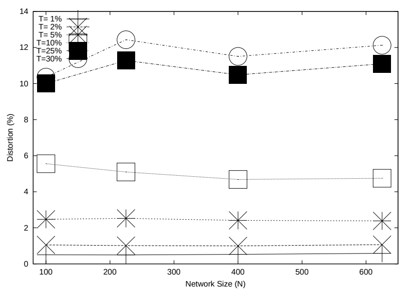
C. Settings

Following the assumptions in Section III-A, the user gateway is placed at the upper left corner of the grid. We also assume a perfect MAC layer, i.e. there is no loss or overhead due to contention or environment changes. The eScan algorithm in the experiments uses 16 bits to represent the residual energy and 32 bits for every node location. An eScan is represented as a collection of a (min, max) value segment plus a coverage polygon. The estimated residual energy in a eScan for a node covered by a polygon is $\frac{min+max}{2}$ in the corresponding value segment. The messaging cost only includes the data size for eScans but ignores other overhead such as packet headers.

Fig. 4. Relative Distortion for eScans

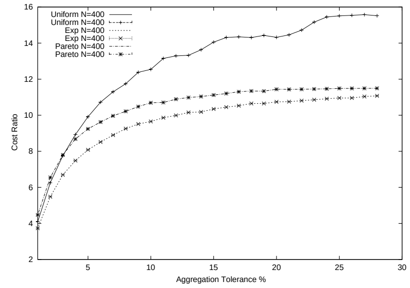


(a) Distortion Uniform(p=0.1)

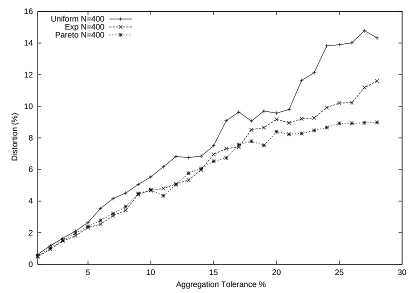


(b) Distortion for Hotspot Model (Exponential, a=0.5)

Fig. 5. Cost Ratio and Distortion v.s. Aggregation Tolerance



(a) Cost Ratio v.s. Aggregation Tolerance



(b) Distortion v.s. Aggregation Tolerance

Each node aggregates eScans with tolerance T. Incremental update is supported with a cache of size 30. eScan updates are triggered if the energy drops more than 0.1%.

Each run of our experiment corresponds to one choice of random aggregation tree and parameters for one of the energy dissipation models. To compute continuous monitoring cost E_c, we also run energy dissipation simulation for $M_1 = 100$ events before starting eScan process. We continue to simulate energy dissipation for another $M = 200$ events in the same time of eScanning. We then stop energy dissipation simulation but continue eScan process until all updates arrive at user gateway. We then compute the cost E_0, E_c and relative distortion D for the final snapshot. For each set of experiments , the number of runs were adjusted to obtain acceptable 95% confidence intervals.

D. Results

Figures 3 plot R of different aggregation tolerance T, as a function of the network size N for different energy dissipation models and Figures 4 plot the corresponding distortion for each set of experiments. The energy-efficiency of eScans can be observed when we evaluate cost ratio together with distortion introduced by aggregation and incremental update. For example, for a network of 400 nodes using the uniform dissipation model ($p = 0.1$), aggregation with $T = 10\%$ can save messaging costs by a factor of 12.5, but only introduces ~5% distortion. Beyond a certain tolerance, however, the gains are almost the same. For example, Figure 3(a) shows T=25% and T=30% are not distinguishable. At these levels, almost the entire network is aggregated into one polygon. Results for the two hotspot dissipation models shows the same trend.

Given a fixed aggregation tolerance, the cost ratio increases with the network size, which indicates that eScan collection with aggregation and incremental update has better scalability than centralized collection with no aggregation. This trend holds for all three energy dissipation models. However, for smaller aggregation tolerances, these results seem to be less dependent on network size. This confirms one of our design principles that given the high cost for communication, it is preferable to process data locally to aggregate data rather than disseminating raw data over long distance.

The cost ratio R can be plotted as a function of T in Figure 5(a). From this graph, the cost ratio increases sharply for larger aggregation tolerance. However the curve converges gradually when T increases. Figure 5(b) shows distortion v.s T, where distortion is roughly a linear function of aggregation of tolerance with slope of around $\frac{1}{2}$. Combined with (a), it can provide guideline to choose proper aggregation tolerance.

In Figure 5(a) the two curves of Hotspot-Pareto and Hotspot-exponential dissipation models roughly show the same trend because they share the exact same number of hotspots and locations. However the curve for Pareto model outperform the case of exponential distribution. Given the same shape factor of $a = 0.5$, the Pareto model of energy dissipation tends to have less localized impact, leading to a smaller deviation of residual energy across nodes, which increases the aggregatability of the scan.

We have also studied the performance sensitivity to different simulation parameters such as the number of hotspots in energy dissipation model, probability function shape factor a. Our experiments show that small perturbations in those parameters does not change the experimental results significantly.

The models of energy consumptio in simulations are reasonable but may appear in favor to our aggregation scheme. The practical energy consumption might be totally different. For example, an extreme case might be that residual energy levels are highly diverse between every neighboring nodes, thus our proposed aggregation scheme with fixed tolerance does not perform well and the saving in communication is limited. Our argument for this highly diverse case is that actually there are hardly any schemes can perform well in this case. Furthermore, the definition with min/max value serves more as a simple but proof-of-concept design. Applying more sophisticated statistical/modeling tools in the definition of VALUE in scans, we expect our scheme can be more robust to those unfavorable cases to provide more meaningful abstraction of network states.

V. Concluding Comments

Continuously monitoring resource distribution and network activity will be a integral component for future sensor networks. To our knowledge, there is no other ongoing or previous work on continuous monitoring large-scale distributed sensor networks. Our design of residual energy scans provides an overall abstracted view of residual energy in an energy-efficient manner. Instead of collecting the raw residual energy data from individual nodes, we apply in-network aggregation to composite residual energy scans. Our design is an instance of trading off local processing cost against the savings in communicating raw data over long distance. Simulation results show that our approach has good scalability and energy-efficiency characteristics, compared to extracting the residual energy individually from each node. To some extent, residual energy scanning itself is a unique application on sensor networks. Our research on this topic will also enrich understanding of sensor network design in general.

We will continue to refine and evaluate our design, especially those issues stated in Section III-E. A residual energy scan is only one kind of abstracted indication of sensor network state. Different network resource or performance metrics may require different techniques to achieve good energy-efficiency, scalability and robustness characteristics. We intend to investigate other sensor network scans and their impacts in the design of monitoring mechanism for sensor networks.

References

[1] G. Asada, T. Dong, F. Lin, G. Pottie, W. Kaiser, and H. Marcy, "Wireless integrated network sensors: Low power systems on a chip," in *European Solid State Circuits Conference*. October 1998, The Hague, Netherlands.

[2] D. Estrin, R. Govindan, J. Heidemann, and S. Kumar, "Next Century Challenges: Scalable Coordination in Sensor Networks," in *Proceedings of the ACM/IEEE International Conference on Mobile Computing and Networking*, Seattle, WA, USA, Aug. 1999, ACM.

[3] J. M. Kahn, R. H. Katz, and K. S. J. Pister, "Next Century Challenges: Mobile Networking for "Smart Dust"," in *Proceedings of the ACM/IEEE International Conference on Mobile Computing and Networking*, Seattle, WA, USA, Aug. 1999, ACM.

[4] Deborah Estrin, Ramesh Govindan, and John Heidemann, "Embedding the Internet," *Communications of the ACM*, vol. 43, no. 5, pp. 39–41, May 2000, (special issue guest editors).

[5] Gregory J. Pottie and William J. Kaiser, "Embedding the internet: wireless integrated network sensors," *Communications of the ACM*, vol. 43, no. 5, pp. 51–58, May 2000.

[6] A. Bagchi and S. L. Hakimi, "An Optimal Algorithm for Distributed System Level Diagnosis," in *Proc. 21st Int. Symp. on Fault-Tolerant Computing (FTCS-21)*, Montreal, Que., Canada, 1991, pp. 214–21, IEEE Computer Society Press.

[7] A. Reddy, D. Estrin, and R. Govindan, "Large Scale Fault Isolation," *IEEE Journal of Selected Areas in Communication*, Feb. 2000, Special Issue on Network Management.

[8] P. Bates, "Debugging heterogeneous distributed systems using event-based models of behavior," *ACM Transactions on Computer Systems*, vol. 13, no. 1, 1995.

[9] A. Tarafdar and V.K. Garg, "Predicate Control for Active Debugging of Distributed Programs," in *Symposium on Distributed and Parallel Debugging*, 1998.

[10] Douglas M. Blough and Hongying W. Brown, "The Broadcast Comparison Model for On-Line Fault Diagnosis in Multicomputer Systems," *IEEE Transactions on Computers*, vol. 48, no. 5, pp. 470–493, 1999.

[11] S. Boyer, *SCADA: Supervisory Control and Data Acquisition*, Instrument Society of America, 1993.

[12] David Tennenhouse, "Embedding the Internet: proactive computing," *Communications of the ACM*, vol. 43, no. 5, pp. 43–43, May 2000.

[13] Chalermek Intanagonwiwat, Ramesh Govindan, and Deborah Estrin, "Directed diffusion: A scalable and robust communication paradigm for sensor networks," in *Proceedings of the ACM/IEEE International Conference on Mobile Computing and Networking*, Boston, MA, USA, Aug. 2000, pp. 56–67, ACM.

[14] Wendi Rabiner Heinzelman, Joanna Kulik, and Hari Balakrishnan, "Adaptive Protocols for Information Dissemination in Wireless Sensor Networks," in *Proceedings of the ACM/IEEE International Conference on Mobile Computing and Networking*, Seattle, WA, USA, Aug. 1999, pp. 174–185, ACM.

[15] Pramod K. Varshney, "Multisensor data fusion," in *Industrial and Engineering Applications of Artifi cial Intelligence and Expert Systems*, 2000, pp. 1–3.

[16] P. Varshney and C. Burrus, *Distributed detection and data fusion*, New York, Springer, 1997.

[17] Christopher Welles and Walter Milliken, "Loader Debugger Protocol," *Internet Request for Comments 909*, 1984.

[18] J. Case, M. Fedor, M. L. Schoffstall, and C. Davin, "Simple Network Management Protocol," RFC 1908, Internet Request For Comments, Nov. 1996.

[19] J. Mahdavi, M. Mathis, and V. Paxson, "Creating a national measurement infrastructure (nimi)," http://www.psc.edu/networking/nimi, May 1997.

[20] R. Caceres, N. G. Duffi eld, J. Horowitz, and D.Towsley, "MINC: Multicast-Based Inference of Network-Internal Loss Characteristics," in *Proceedings of the IEEE Infocom*. 1999, IEEE Computer Society Press.

[21] Seapahn Meguerdichian, Farinaz Koushanfar, Miodrag Potkonjak, and Mani Srivastava, "Coverage Problems in Wireless Ad-hoc Sensor Networks," in *Proceedings of the IEEE Infocom*, 2001.

[22] Ya Xu, John Heidemann, and Deborah Estrin, "Geography-informed energy conservation for ad hoc routing," in *Proceedings of the ACM/IEEE International Conference on Mobile Computing and Networking*, Rome, Italy, July 2001, pp. 70–84, ACM.

[23] E. Kaplan, *Understanding GPS: Principles and Applications*, Artech House, 1996.

[24] N. Bulusu, J. Heidemann, and D. Estrin, "GPS-less Low Cost Outdoor Localization for Very Small Devices," *Proceedings of the IEEE Personal Communications Magazine*, vol. 7, no. 5, pp. 28–34, Oct. 2000.