# Impact of Data Aggregation in Wireless Sensor Networks

**Authors and Addresses:**

Bhaskar Krishnamachari
bhaskar@ece.cornell.edu
Wireless Intelligent Systems Laboratory
School of Electrical and Computer Engineering
Cornell University
Ithaca, NY 14853

Deborah Estrin
destrin@cs.ucla.edu
Laboratory for Embedded Collaborative Systems
Department of Computer Science
University of California Los Angeles
Los Angeles, CA, 90095

Stephen Wicker
wicker@ece.cornell.edu
Wireless Intelligent Systems Laboratory
School of Electrical and Computer Engineering
Cornell University
Ithaca, NY 14853

**Keywords:** Data aggregation, data-centric routing, directed diffusion, wireless sensor networks, event-based systems.

**Abstract:** Sensor networks are event-based systems that differ from traditional communication networks in several ways: sensor networks have severe energy constraints, redundant low-rate data, and many-to-one flows. The end-to-end routing schemes that have been proposed in the literature for mobile ad-hoc networks are not appropriate under these settings. Data-centric technologies are needed that perform in-network aggregation of data to yield energy-efficient dissemination. In this paper we model data-centric routing and compare its performance with traditional end-to-end routing schemes. We examine the impact of source-destination placement and communication network density on the energy costs and delay associated with data aggregation. We show that data-centric routing offers significant performance gains across a wide range of operational scenarios. We also examine the complexity of optimal data aggregation, showing that although it is an NP-hard problem in general, there exist useful polynomial-time special cases.

# The Impact of Data Aggregation in Wireless Sensor Networks

*Abstract*— **Sensor networks are distributed event-based systems that differ from traditional communication networks in several ways: sensor networks have severe energy constraints, redundant low-rate data, and many-to-one flows. The end-to-end routing schemes that have been proposed in the literature for mobile ad-hoc networks are not appropriate under these settings. Data-centric technologies are needed that perform in-network aggregation of data to yield energy-efficient dissemination. In this paper we model data-centric routing and compare its performance with traditional end-to-end routing schemes. We examine the impact of source-destination placement and communication network density on the energy costs and delay associated with data aggregation. We show that data-centric routing offers significant performance gains across a wide range of operational scenarios. We also examine the complexity of optimal data aggregation, showing that although it is an NP-hard problem in general, there exist useful polynomial-time special cases.**

## I. Introduction

THE wireless sensor networks of the near future are envisioned to consist of hundreds to thousands of inexpensive wireless nodes, each with some computational power and sensing capability, operating in an unattended mode. They are intended for a broad range of environmental sensing applications from vehicle tracking to habitat monitoring [3], [18], [23]. The hardware technology for these networks - low cost processors, miniature sensing and radio modules are here today, with further improvements in cost and capabilities expected within the next decade [3], [12], [14], [18], [19]. The applications, networking principles and protocols for these systems are just beginning to be developed [7], [8], [10], [18].

Sensor networks are quintessentially event-based systems. A sensor network consists of one or more “sinks” which subscribe to specific data streams by expressing interests or queries. The sensors in the network act as “sources” which detect environmental events and push relevant data to the appropriate subscriber sinks. For example, there may be a sink that is interested in a particular spatio-temporal phenomenon (“does the temperature ever exceed 70 degrees in area A between 10am and 11am ?”). During the given time interval all sensors in the corresponding spatial portion of the network act as event-based publishers. They publish information toward the subscribing sink if and when they detect the indicated phenomenon.

Because of the requirement of unattended operation in remote or even potentially hostile locations, sensor networks are extremely energy-limited. However since various sensor nodes often detect common phenomena, there is likely to be some redundancy in the data the various sources communicate to a particular sink. In-network filtering and processing techniques can help conserve the scarce energy resources.

*Data aggregation* has been put forward as an essential paradigm for wireless routing in sensor networks [9], [13]. The idea is to combine the data coming from different sources enroute – eliminating redundancy, minimizing the number of transmissions and thus saving energy. This paradigm shifts the focus from the traditional *address-centric* approaches for networking (finding short routes between pairs of addressable end-nodes) to a more *data-centric* approach (finding routes from multiple sources to a single destination that allows in-network consolidation of redundant data).

In this paper we study the energy savings and the delay tradeoffs involved in data aggregation and how they are impacted by factors such as source-sink placements and the density of the network. We also investigate the computational complexity of optimal data aggregation in sensor networks and show that although it is generally NP-hard, there exist polynomial special cases.

## II. Routing Models

We focus our attention on a single network flow that is assumed to consist of a single data sink attempting to gather information from a number of data sources. We start with simple models of routing schemes which use data aggregation (which we term data-centric), and schemes which do not (which we term address-centric). In both cases we assume there are some common elements - the sink first sends out a query/interest for data, the sensor nodes which have the appropriate data then respond with the data. They differ in the manner the data is sent from the sources to the sink:

**Address-centric Protocol (AC)**: Each source independently sends data along the shortest path to sink based on the route that the queries took ( “end-to-end routing” ).

**Data-centric Protocol (DC)**: The sources send data

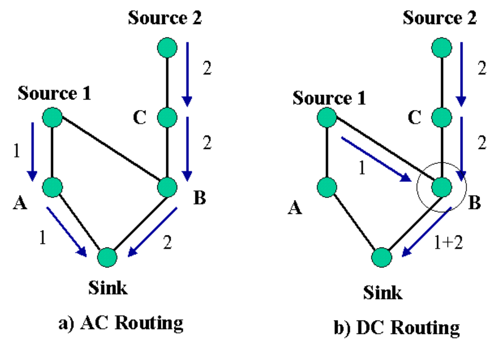


Fig. 1. Illustration of AC versus DC routing

to the sink, but routing nodes enroute look at the content of the data and perform some form of aggregation/consolidation function on the data originating at multiple sources.

Figure 1 is a simple illustration of the difference between AC and DC schemes. In the address-centric approach, each source sends its information separately to the sink (source 1 routing the data labelled "1" through node A, and source 2 routing the data labelled "2" through nodes C and B). In the data centric-approach, the data from the two sources is aggregated at node B, and the combined data (labelled "1+2") is sent from B to the sink. The latter results in energy savings as fewer transmissions are required to send the information from both sources to the sink.

## III. Data Aggregation

Data aggregation is the combination of data from different sources, and can be implemented in a number of ways. The simplest data aggregation function is duplicate suppression - in the example of figure 1, if sources 1 and 2 both send the same data, node B will send only one of these forward. Other aggregation functions could be *max*, *min*, or any other function with multiple inputs. For our modelling purposes in this paper we make a simplifying assumption - the aggregation function is such that each intermediate node in the routing transmits a single aggregate packet even if it receives multiple input packets. We will refer to the information received by the sink when it has obtained the messages transmitted by all sources in a given flow (whether or not these messages are aggregated) as a "datum".

### A. Optimal Aggregation

Say there are $k$ sources, labelled $S_1$ through $S_k$, and a sink, labelled $D$. Let the network graph $G = (V, E)$ consist of all the nodes $V$, with $E$ consisting of edges between nodes that can communicate with each other directly. With the assumption that the number of transmissions from any node in the data aggregation tree is exactly one, the data aggregation tree can be thought of as the reverse of a multicast tree: instead of a single source sending a packet to all receivers, all the sources are sending a single packet to the same receiver. It is well-known that the multicast tree with a minimum number of edges is a minimum Steiner tree on the network graph. The following can therefore be readily obtained:

***Result 1***: The optimum number of transmissions required per datum for the DC protocol is equal to the number of edges in the minimum Steiner tree in the network which contains the node set $(S_1, ...S_k, D)$.

***Corollary***: Assuming an arbitrary placement of sources, and a general network graph G, the task of doing DC routing with optimal data aggregation is NP-hard.

The latter follows from the NP-completeness of the minimum Steiner problem on Graphs [24].

### B. Suboptimal Aggregation

The following are three generally suboptimal schemes for generating data aggregation trees that we examine in this paper.

1. **Center at Nearest Source (CNS)**: In this data aggregation scheme, the source which is nearest the sink acts as the aggregation point. All other sources send their data directly to this source which then sends the aggregated information on to the sink.
2. **Shortest Paths Tree (SPT)** : In this data aggregation scheme, each source sends its information to the sink along the shortest path between the two. Where these paths overlap for different sources, they are combined to form the aggregation tree.
3. **Greedy Incremental Tree (GIT)** : In this scheme the aggregation tree is built sequentially. At the first step the tree consists of only the shortest path between the sink and the nearest source. At each step after that the next source closest to the current tree is connected to the tree.

This is by no means an exhaustive list, but is representative of some of the data aggregation tree heuristics that can be implemented.

### C. Performance measures

In exploring the gains and tradeoffs involved in data-centric protocols, we need to specify performance measures of interest. Two are examined in some detail in this paper:

- **Energy Savings**: By aggregating the information coming from the sources, the number of transmissions is reduced, translating to a savings in energy.

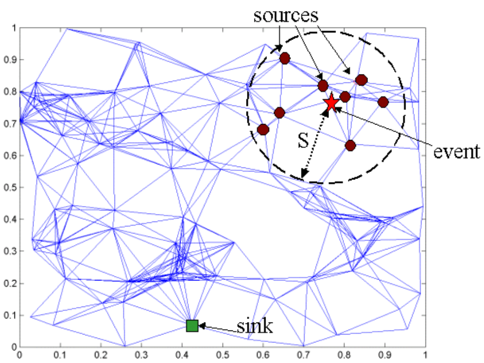


Fig. 2. Illustration of the event-radius model for source positions

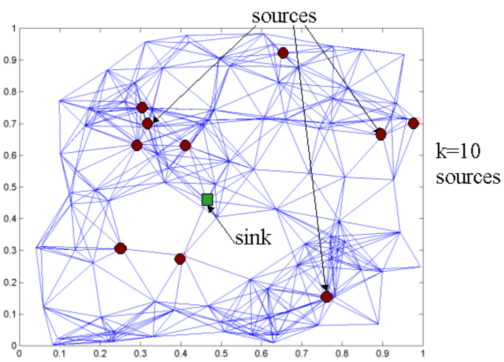


Fig. 3. Illustration of the random-sources model for source positions

• **Delay**: There is latency associated with aggregation. Data from nearer sources may have to be held back at intermediate nodes in order to combine them with data from sources that are farther away.

### D. Source Placement Models

The chief factors that can affect the performance of data aggregation methods are the positions of the sources in the network, the number of sources, and the communication network topology. In order to investigate these factors, we study two models of source placement, the event-radius (ER) model, and the random sources (RS) model. In both models, we generate a sensor network by scattering $n$ sensor nodes randomly in a unit square. One of these nodes is the data sink. All nodes are assumed to be able to communicate with any other nodes that are within some distance $R$ (the communication radius). The location of the data sources depends on the models as follows:

• **Event-Radius Model**: In this model, a single point in the unit square is defined as the location of an "event." This may correspond to a vehicle or some other phenomenon being tracked by the sensor nodes. All nodes within a distance $S$ (called the sensing range) of this event that are not sinks are considered to be data sources. The average number of sources is approximately $\pi * S^2 * n$ (somewhat less than this if we take into account boundary effects). This model is shown in figure 2.

• **Random-Sources Model**: In this model, $k$ of the nodes that are not sinks are randomly selected to be sources. Unlike in the event-radius model, the sources are not necessarily clustered near each other. This is illustrated in figure 3.

## IV. Energy Savings due to Data Aggregation

### A. Theoretical Results

We now give some analytical bounds on the energy costs and savings that can be obtained with data aggregation, based on the distances between the sources and the sink, and the inter-distances among the sources. The upshot of this section is that the greatest gains due to data aggregation are obtained when the sources are all close together and far away from the sink.

Let $d_i$ be the distance of the shortest path from source $S_i$ to the sink in the graph. Per datum, the total number of transmissions required for the optimal AC protocol in this case (call it $N_A$) is:

$$N_A = d_1 + d_2 + ...d_k = sum(d_i) \tag{1}$$

Let the number of transmissions required for the optimal DC protocol be $N_D$.

*Definition*: The "diameter" $X$ of a set of nodes $S$ in a graph $G$ is the maximum of the pairwise shortest paths between these nodes $X = max_{i,j \in S} SP(i,j)$ where $SP(i,j)$ is the shortest number of hops needed to go from node $i$ to $j$ in $G$.

***Result 2***: If the source nodes $S_1, S_2, \ldots S_k$ have a diameter $X \geq 1$. The total number of transmissions ($N_D$) required for the optimal DC protocol satisfies the following bounds:

$$N_D \leq (k-1)X + min(d_i) \tag{2}$$
$$N_D \geq min(d_i) + (k-1) \tag{3}$$

*Proof* : (2) can be obtained by a construction - the data aggregation tree which consists of $(k-1)$ sources sending their packets to the remaining source which is nearest to the sink. This tree has no more than $(k-1)X + min(d_i)$ edges, hence the optimum tree must have no more than this. (3) is obtained by considering the smallest possible Steiner tree which would happen if the diameter were 1. In this case, the shortest path from the source node at

$min(d_i)$ must be part of the minimum Steiner tree, and there is exactly one edge from each of the other source nodes to this node. □

***Result 3***: If the diameter $X < min(d_i)$, then $N_D < N_A$. In other words, the optimum data-centric protocol will perform strictly better than the AC protocol in terms of the total number of transmissions.

*Proof* :

$$\Rightarrow N_D < (k-1)X + min(d_i) < (k)min(d_i)$$
$$\Rightarrow N_D < sum(d_i) = N_A. \quad (4)$$

□

***Definition*** : Let us define the fractional savings, $FS$, obtained by using the DC protocol as opposed to the AC protocol as follows:

$$FS = (N_A - N_D)/(N_A) \quad (5)$$

$FS$ can range from 0 (no savings) to 1 (100 percent savings). The following are the lower and upper bounds on $FS$, which follow directly from (2) and (3) and the above definition.

***Result 4***: The fractional savings FS satisfies the following bounds:

$$FS \geq 1 - ((k-1)X + min(d_i))/sum(d_i) \quad (6)$$
$$FS \leq 1 - (min(d_i) + k - 1)/sum(d_i) \quad (7)$$

To clarify the matter, assume that all the sources are at the same shortest-path distance from the sink. i.e. $min(d_i) = max(d_i) = d$.

Then we have that

$$1 - \frac{((k-1)X + d)}{kd} \leq FS$$
$$\leq 1 - \frac{(d + k - 1)}{(kd)} \quad (8)$$

***Result 5***: Assume X and k are fixed, then as d tends to infinity (i.e. as the sink is farther and farther away from the sources):

$$lim_{d \to \infty} FS = 1 - 1/k. \quad (9)$$

*Proof*:

In the limit, $X << d$, and $k << d$. It suffices to show that both lower and upper bounds in (8) converge to the same right hand side value:

$$lim_{d \to \infty}\left(1 - \frac{(k-1)X + d}{kd}\right)$$
$$= lim_{d \to \infty}\left(1 - \frac{(k-1)X}{kd} - \frac{d}{kd}\right) = 1 - 1/k \quad (10)$$

and

$$lim_{d \to \infty}\left(1 - \frac{(d + k - 1)}{(kd)}\right)$$
$$= lim_{d \to \infty}\left(1 - \frac{d}{kd} - \frac{(k-1)}{kd}\right) = 1 - 1/k \quad (11)$$

□

Result 6 tells us that if the distance between the sink and the sources is large compared to the distance between the sources, then the optimal DC protocol gives k-fold savings. For example, when there are 4 sources that are close together and located far-away from the sink, then the AC protocol will have about 4 times as many transmissions, i.e. there are roughly 75 $\%$ fewer transmissions with data aggregation.

***Result 6***: If the subgraph $G'$ of the communication graph $G$ induced by the set of source nodes $(S_1, \ldots S_k)$ is connected, the optimal data aggregation tree can be formed in polynomial time.

*Proof*: The proof is constructive. Start GIT. The tree is initialized with the path from the sink to the nearest source. At each additional step of the GIT, the next source to be connected to the tree is always exactly one step away (such a source is guaranteed to exist since $G'$ is connected). At the end of the construction, the number of edges in the tree is therefore $d_{min} + (k-1)$, which is the lower bound given in relation (3). Hence the lower bound is tight and therefore optimal. The GIT construction runs in polynomial time w.r.t. the number of nodes [21]. Hence although finding the optimal data aggregation tree is NP-hard in general, in this particular situation, we have a polynomial special-case.□

***Result 7***: In the ER model, when $R > 2S$, the optimal data aggregation tree can be formed in polynomial time.

*Proof*: It is easy to see that when $R > 2S$, all sources are within one hop of each other. This is therefore a special case of result 8. Under this condition, both GIT and CNS schemes will result in the optimal data aggregation tree.□

### *B. Experimental Results*

We now present our experimental results showing the energy costs of AC and DC protocols for both the ER and RS source placement models. The experimental setup is as follows: for the ER model, 5 evenly spaced values of the

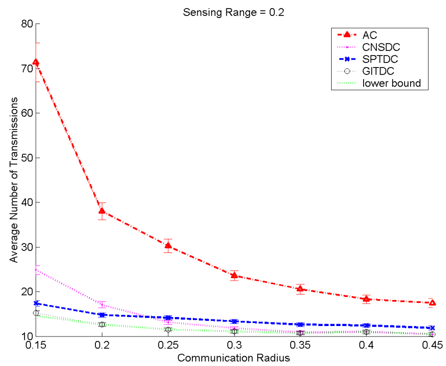


Fig. 4. Comparison of energy costs versus communication radius in event-radius model

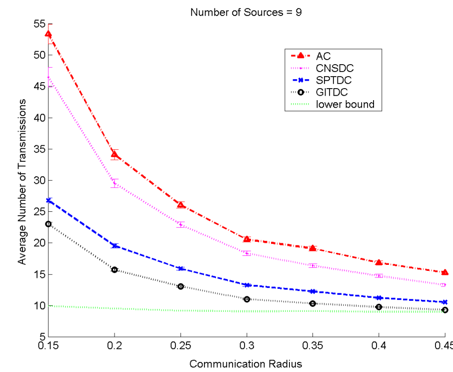


Fig. 6. Comparison of energy costs versus communication radius in random-sources model

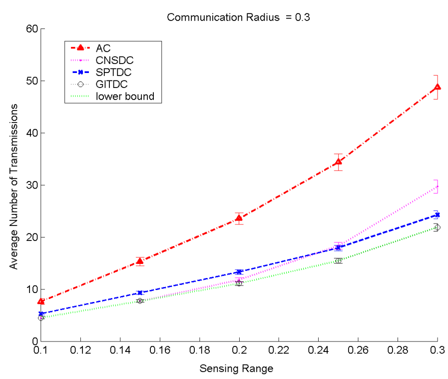


Fig. 5. Comparison of energy costs versus sensing range in event-radius model

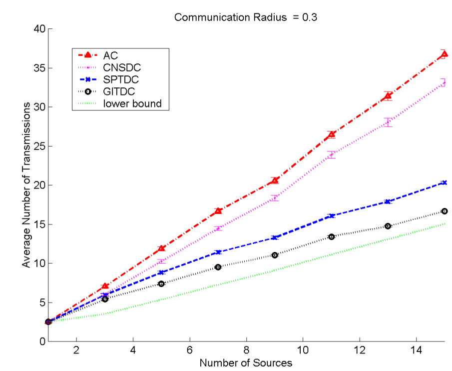


Fig. 7. Comparison of energy costs versus number of sources in random-sources model

sensing range $S$ from 0.1 to 0.3 are tested, while for the RS model the number of sources $k$ is varied 1 to 15 in increments of 2. For both models the communication radius $R$ is varied from 0.15 to 0.45 in increments of 0.05. For each combination of $S$ or $k$ and $R$ 100 experiments were run. Each experiment consists of a random placement of the $n = 100$ nodes including the sink node in a square are of unit size. In some cases (particularly when the values of $E$ or $R$ are low) a particular experiment may result in unconnected graphs or no sources; the measurements from these cases are not taken into account while computing the averages. The error-bars shown in the plots represent the standard error in the mean.

Figure 4 compares the transmission energy costs of the various protocols as the communication range is varied, keeping the sensing range constant at 0.2 (which corresponds to about 12.5 sources on average, ignoring edge-effects). At the very bottom is the lower bound on $N_D$ given in relation 3. In this figure it can be seen that the GITDC seems to coincide with the lower bound all throughout. This is because when $S$ is even of moderate length, with high probability, the subgraph which lies within the circle of radius $S$ around the event is connected, and result 7 holds. The performance of the CNSDC approaches optimal as $R$ increases, as per result 8. The SPTDC protocol also performs well all through. In all cases there is a $50 - 80\%$ savings compared to the AC protocol. Figure 6 is the equivalent plot for the Random Sources Model. The first thing to note is that the lower bound is no longer tight, since the sources are placed randomly anywhere in the network and unless the network is dense (high $R$) the sources are unlikely to be within one hop of each other. In this setting the GITDC performs the best, followed by SPTDC, CNSDC and AC, respectively. CNSDC performs poorly in this setting since the sources are far apart and it doesn't pay to always aggregate at the source nearest to the sink.

Figures 5 and 7 both show that the transmission costs increase as the number of sources is increased. In the event-radius model, it can be seen that the CNSDC protocol performs poorly when the sensing range is really large. When $S = 0.3$, nearly a third of all nodes in the experiments act as sources and for many of these sources it may be faster to route directly to the sink rather than through one particular source that is closest to the sink. Figure 7 shows that the gains due to a good data aggregation technique

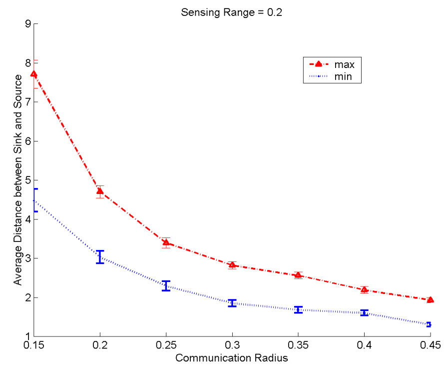


Fig. 8. Distance of sink to nearest and farthest source versus communication radius in event-radius model

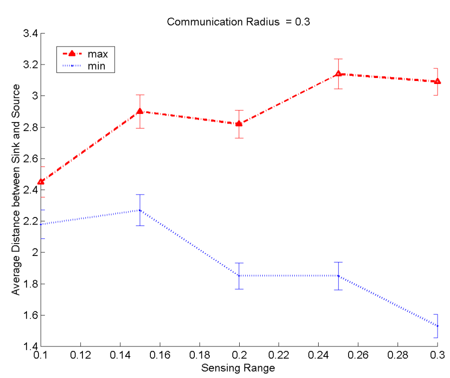


Fig. 9. Distance of sink to nearest and farthest source versus sensing range in event-radius model

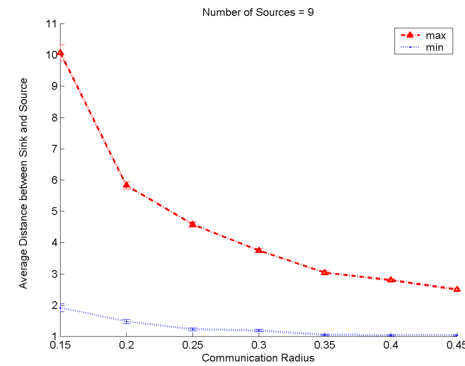


Fig. 10. Distance of sink to nearest and farthest source versus communication radius in random-sources model

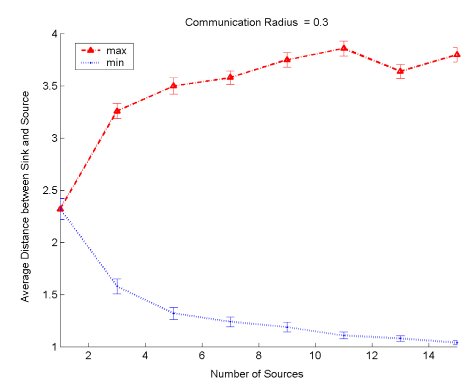


Fig. 11. Distance of sink to nearest and farthest source versus number of sources in random-sources model

(like GITDC) can be very significant when the number of sources is high.

To summarize, our experiments show that the energy gains due to data aggregation can be quite significant with SPTDC or GITDC particularly when there are a lot of sources (large $S$ or large $k$) that are many hops from the sink (small $R$).

## V. Delay due to Data Aggregation

Although data aggregation results in fewer transmissions, there is a tradeoff - potentially greater delay because data from nearer sources may have to be held back at an intermediate node in order to be aggregated with data coming from sources that are farther away. This can be seen by referring back to figure 1; in figure 1b, node B which acts as the aggregating node for sources 1 and 2, is only one hop from source 1 but is two hops from source 2. Thus if both sources transmit the data simultaneously, the data from source 1 will get to B before the data from source 2 and take longer to get to the sink than it would in the no aggregation scheme shown in figure 1a. Note that this delay depends on the aggregation function - for some simple kinds of data aggregation such as duplicate suppression, there is no need for data to be withheld at an aggregating node. For more complicated forms of data aggregation, where the output aggregated packet depends on the combination of multiple input packets this delay is an issue.

It can be seen that, in the worst case, the latency due to aggregation will be proportional to the number of hops between the sink and the farthest source. When no aggregation is employed, the delay between the time when the various sources transmit data and the sink receives the first packet is proportional to the number of hops between the sink and the nearest source. Hence one way to quantify the effect of aggregation delay is to examine the difference between these two distances. This is shown in figures 8-11. The experimental setup is the same as discussed in section IV-B. The upper curve in all these figures is representative of the latency delay in DC schemes with non-trivial aggregation functions and the lower curve is representative of the latency delay in AC schemes. The difference between these curves is greatest in both models when the communication radius is low, and when the number of sources is high. In figure 11, as the number of sources increases the two curves saturate to extreme values. The upper curve saturates to a value of about 4 which is about the maximum number of hops between the sink and any node in the network. The lower curve saturates at a value close to the minimum number of hops (1).

It should be noted that there are two other possible sources of delay that we have not taken into account - delay due to congestion and the processing delay. We chose not to model the delay due to congestion as this would depend on a number of additional details such as the MAC protocol used, and the traffic in the network which are not likely to have a differential impact on data-centric versus address-centric protocols. The processing delay at each node is a second order effect and unlikely to be a significant issue for most anticipated forms of aggregation.

## VI. Related Work

The use of sensor networks has been envisioned in a range of settings such as industrial applications [23], vehicle tracking applications [18] and habitat monitoring [3]. A number of independent efforts have been made in recent years to develop the hardware and software architectures needed for wireless sensing. Of particular note are UC Berkeley's Smart Dust Motes [14], TinyOS [12], and the PicoRadio [19] project; the Wireless Integrated Network Sensors (WINS) project [18] and PC-104 based sensors [3] developed at University of California Los Angeles; and the $\mu$AMPS project at MIT [16]. The challenges and design principles involved in networking these devices are discussed in [7], [8], and [15]. Energy-efficient medium access schemes applicable for sensor networks are presented in [5], and [25]. Techniques for balancing the energy load among sensors using randomized rotation of cluster heads are discussed in [11]. Some attention has also been given to developing localized self-configuration mechanisms in sensor networks [4].

The great majority of wireless routing protocols developed in recent years have been for mobile ad-hoc communication networks [17]. These approaches are all address-centric, in that they are focused on end-to-end routing between pairs of addressable nodes.

The application-specific nature of sensor networks leads to the alternative approach we have described in this paper as data-centric. The meta-naming of data is suggested in [10] as a means to reduce transmission of redundant data for flooding-like schemes for information dissemination. The *Directed diffusion* protocol, which is most like the data-centric routing models analyzed in this paper, is described in [13]. A physical implementation of directed diffusion with a small wireless sensor test-bed consisting of 14 nodes with 4 sources and a single sink is described in [9]. For the particular configuration described in that paper the form of data aggregation used (duplicate suppression) is observed to reduce the traffic by up to $42\%$. Our work has generalized that result by showing the performance of data aggregation for a wide range of source placement topologies and densities in a larger network. Also, the effect of data aggregation on delay has not been discussed much previously in the literature.

The use of in-network processing during routing has also been considered in other contexts such as fsActive Networks [22], and router-assist techniques for multicast on the internet [2].

Optimal data aggregation, as we have shown in this paper, requires the formation of a minimum Steiner tree, a well known NP-complete problem arising in many networking contexts [24]. The greedy incremental tree (GIT) heuristic scheme described in our paper is a well-known approximation algorithm for this problem [21] with an approximation ratio of 2. A distributed version of this algorithm is discussed in [1]. The best known approximation algorithm for the minimum Steiner tree problem has an approximation ratio of about 1.55 [20].

Finally, we mention here in passing that there is another sense in which the phrase "data-centric networking" has been used [6]; namely to describe an approach to ubiquitous computing in which human users are identified not with static computing devices but with their personalized services and data.

## VII. Conclusions

Wireless sensor networks are an important type of resource-constrained distributed event-based systems. We have modelled and analyzed the performance of data aggregation in such networks.

We identified and investigated some of the factors affecting performance, such as the number of placement of sources, and the communication network topology. The formation of an optimal data aggregation tree is generally NP-hard. We presented some suboptimal data aggregation tree generation heuristics and showed the existence of polynomial special cases.

The modelling tells us that whether the sources are clustered near each other or located randomly, significant energy gains are possible with data aggregation. These gains are greatest when the number of sources is large, and when the sources are located relatively close to each other and far from the sink. The modelling, though, also seems to suggest that aggregation latency could be non-negligible and should be taken into consideration during the design process. Data-centric architectures such as directed diffusion should support a Type of Service (TOS) facility that would permit applications to effect desired tradeoffs between latency and energy.

Our analysis has focused on the case where there is a single sink. Although this is a reasonable scenario for many applications, it is reasonable to ask what would happen if there were additional sinks. One solution is to think of the different flows in that case as a superposition of many single sink data-flows. However, this would yield an over-estimate of the energy costs, as further aggregation savings can be possible if there are redundancies in the sources and the data being requested by the various sinks. This is a topic for further study.

In-system processing of data is useful to avoid overwhelming the consumer of data notification, be it a person or a program. Thus the results we have presented in this paper for a resource-constrained event-based system might well hold important design lessons for scalable event-based systems, even if they are less constrained.

## REFERENCES


[1] F. Bauer and A. Varma. "Distributed algorithms for multicast path setup in data networks," *in Transactions on Networking*, vol. 4, no. 2, 181-191, 1996.

[2] B. Cain, T. Speakman, D. Towsley, "Generic Router Assist (GRA) Building Block Motivation and Architecture," *RMT Working Group, Internet-Draft <draft-ietf-rmt-gra-arch-01.txt>*, work in progress, March 2000.

[3] A. Cerpa *et al.*, "Habitat monitoring: Application driver for wireless communications technology," *2001 ACM SIGCOMM Workshop on Data Communications in Latin America and the Caribbean*, Costa Rica, April 2001.

[4] A. Cerpa and D. Estrin, "ASCENT: Adaptive Self-Configuring sEnsor Networks Topologies," *unpublished*.

[5] Seong-Hwan Cho and A. Chandrakasan, "Energy Efficient Protocols for Low Duty Cycle Wireless Microsensor Networks", *ICASSP 2001*, May 2001.

[6] M. Esler *et al.*"Next Century Challenges: Data-Centric Networking for Invisible Computing: The Portolano Project at the University of Washington," *ACM/IEEE International Conference on Mobile Computing and Networks (MobiCom '99)*, Seattle, Washington, August 1999.

[7] D. Estrin *et al.*"Next Century Challenges: Scalable Coordination in Sensor Networks," *ACM/IEEE International Conference on Mobile Computing and Networks (MobiCom '99)*, Seattle, Washington, August 1999.

[8] D. Estrin, *et al.* "Instrumenting the World with Wireless Sensor Networks," *International Conference on Acoustics, Speech and Signal Processing (ICASSP 2001)*, Salt Lake City, Utah, May 2001.

[9] J. Heidemann *et al.*, "Building Efficient Wireless Sensor Networks with Low-Level Naming," *18th ACM Symposium on Operating Systems Principles*, October 21-24, 2001.

[10] W.R. Heinzelman, J. Kulik, and H. Balakrishnan "Adaptive Protocols for Information Dissemination in Wireless Sensor Networks," *Proceedings of the Fifth Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom '99)*, Seattle, Washington, August 15-20, 1999, pp. 174-185.

[11] W.R. Heinzelman, A. Chandrakasan, and H. Balakrishnan "Energy-Efficient Communication Protocol for Wireless Microsensor Networks," *33rd International Conference on System Sciences (HICSS '00)*, January 2000.

[12] J. Hill *et al.*, "System Architecture Directions for Networked Sensors," *ASPLOS*, 2000.

[13] C. Intanagonwiwat, R. Govindan and D. Estrin, "Directed Diffusion: A Scalable and Robust Communication Paradigm for Sensor Networks," *ACM/IEEE International Conference on Mobile Computing and Networks (MobiCom 2000)*, August 2000, Boston, Massachusetts

[14] J. M. Kahn, R. H. Katz and K. S. J. Pister, "Mobile Networking for Smart Dust", *ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom 99)*, Seattle, WA, August 17-19, 1999

[15] W.W. Manges, "Wireless Sensor Network Topologies," *Sensors Magazine*, vol. 17, no. 5, May 2000.

[16] Rex Min *et al.* "Low-Power Wireless Sensor Networks", *VLSI Design 2000*, January 2001.

[17] C. Perkins, *Ad Hoc Networking*, Addison-Wesley, 2000.

[18] G.J. Pottie, W.J. Kaiser, "Wireless Integrated Network Sensors," *Communications of the ACM*, vol. 43, no. 5, pp. 551-8, May 2000.

[19] J. Rabaey *et al.*, "PicoRadio: Ad-Hoc Wireless Networking of Ubiquitous Low-Energy Sensor/Monitor Nodes ," *Proceedings of the IEEE Computer Society Annual Workshop on VLSI (WVLSI'00)*, Orlando, Florida, April 2000.

[20] G. Robins, A. Zelikovsky, "Improved Steiner Tree Approximation in Graphs," *Proc. of ACM/SIAM Simposium on Discrete Algorithms*, pp. 770-779, 2000.

[21] H. Takahashi and A. Matsuyama, "An approximate solution for the steiner problem in Graphs," *Math. Japonica*, vol. 24, no. 6, pp. 573-577, 1980.

[22] D.L. Tennenhouse, J.M. Smith, W.D. Sincoskie, D.J. Wetherall, and G.J. Minden, "A survey of active network research," *IEEE Communications Magazine*, vol. 35, no. 1, pp. 80-86, January 1997.

[23] J. Warrior, "Smart Sensor Networks of the Future," *Sensors Magazine*, March 1997.

[24] P. Winter, "Steiner Problem in Networks: A survey," *Networks*, vol. 17, no. 2, pp. 129-167, 1987.

[25] A. Woo and D.E. Culler, "A Transmission Control Scheme for Media Access in Sensor Networks," *ACM/IEEE International Conference on Mobile Computing and Networks (MobiCom '01)*, July 2001, Rome, Italy.