Coordinated Workload Scheduling in Hierarchical Sensor Networks for Data Fusion Applications

Xiaolin Li, Hui Kang
Scalable Software Systems Lab
Department of Computer Science
Oklahoma State University
Stillwater, OK, 74078, USA
{xiaolin, huk}@cs.okstate.edu

Jiannong Cao
Internet and Mobile Computing Lab
Department of Computing
Hong Kong Polytechnic University
Hung Hom, Kowloon, Hong Kong
csjcao@comp.polyu.edu.hk

Abstract

To minimize the execution time of a sensing task over a multi-hop hierarchical sensor network, we present a coordinated scheduling method following the divisible load scheduling paradigm. The proposed scheduling strategy builds from eliminating transmission collisions and idle gaps between two successive data transmissions. We consider a sensor network consisting of several clusters. In a cluster, after related raw data measured by source nodes are collected at the fusion node, in-network data aggregation is further considered. The scheduling strategies consist of two phases: intra-cluster scheduling and inter-cluster scheduling. Intra-cluster scheduling deals with assigning different fractions of a sensing workload among source nodes in each cluster; inter-cluster scheduling involves the distribution of fused data among all fusion nodes. Closed-form solutions to the problem of task scheduling are derived. Finally, numerical examples are presented to demonstrate the impacts of different system parameters such as the number of sensor nodes, measurement, communication, and processing speed, on the finish time and energy consumption.

1 Introduction

Wireless sensor networks (WSNs) enable a new class of computing with resource-constrained tiny nodes deployed at a large scale from hundreds to thousands. Many applications using WSNs require measuring, processing, and communicating large amount of data [5, 6, 16]. It is often favorable that a given task should be completed collaborative in a minimum time so that the end system can draw useful conclusions timely, especially for time-critical applications such as fire detection. Quick response of a sensor network also results in energy saving [7, 23]. Moreover, many applications share a common operation on these sensor data, applying a synthesis function, namely data fusion [9]. The challenge is how to minimize finish time of a sensing task and provide analytical means to study the relationship between the minimum finish time and system parameters to aid the design of a WSN system architecture.

We focus on workload scheduling following divisible load theory (DLT) to minimize the finish time in WSNs. DLT has been intensively studied in the past decade [2]. It is mainly concerned with obtaining an optimal partitioning and scheduling strategy for a given task or workload such that it can be processed in the shortest amount of time. Since sensor data are independent to each other, and have no precedence relations, a sensing task can be divided and distributed to any available nodes. Hence, the features of DLT enable a tractable means to model sensing workload scheduling in sensor networks. In addition, the rapid progress in sensor networks makes sensing workload scheduling become more and more important as the advent of integrating sensor networks with the Internet and the Grid [1, 5, 16].

We consider a multi-hop hierarchical sensor network architecture, which is used in many application scenarios from small scale to large scale networks. This model supports data fusion by organizing source nodes into clusters, in which original sensor data are processed at data fusion nodes and then transmitted to the base station to meet some requirement specified by a high-level task. One base station node is used to accept the required amount of sensing task and high-level task, distribute to each cluster by the proposed scheduling strategies, and finally collect results from them. The workload is divided in two stages: intra-cluster and inter-cluster. First, inter-cluster scheduling partitions the entire task into each cluster; then the workload

1-4244-1455-5/07/$25.00 ©2007 IEEE

in a cluster is assigned to each source node by intra-cluster scheduling. The minimized finish time is achieved by coordinating the measuring, processing, and communication time in both stages involving different kinds of nodes. In addition, we investigate the energy consumption of individual sensor nodes. The proposed energy model covers various energy consumption components involved in data acquisition, fusion, and communication. The impacts of system parameters on the performance and energy consumption are evaluated by numerical simulations.

Two key contributions of this study are summarized as follows. First, we obtain the closed-form solutions of the workload distribution to minimize the finish time of a given task. A realistic and widely-used hierarchical sensor network model is used to derive the scheduling strategies. Second, we evaluate the effects of three system parameters of sensor nodes on the performance in terms of finish time and energy consumption. The results show that the proposed scheduling methodology provides a tractable means, following DLT paradigm, for modeling system behaviors of WSNs.

The rest of the paper is organized as follows. We discuss related work in Section 2. In Section 3, we first explains how the proposed sensor network model is tailored to the targeted applications. Then we formally define the multi-hop hierarchical architecture and notations. Section 4 presents the derivation of the close-form solutions for workload scheduling of intra-cluster and inter-cluster nodes. The corresponding energy consumption models for three kinds of sensor nodes are presented in Section 5. In Section 6, we present the numerical simulation results of the evaluation of the system parameters. Section 7 concludes the paper and discusses some directions for future work.

2 Related Work

Divisible load theory offers an elegant and tractable approach to obtain an optimal distribution of workload among a number of processors. It offers a tractable means to minimize task execution times and analytically study system performance [17, 19], compared to other task scheduling methods currently proposed in WSNs [18,22]. With respect to resource constraints as in sensor networks, DLT was used in a single-level tree network with processors having limited buffer size [13]. More recently it was applied to wireless sensor networks in [15]. Although the authors derived closed-form solutions to obtain the optimal finish time for three particular sensor networks, their single level (single-hop) model is not scalable to a large sensor network.

Because data transmission and reception dominate the energy consumption in wireless sensor networks, data fusion, or called data aggregation, aims to compute a smaller size of data that equals or at least represents the original data [8, 14]. Experiments showed that considerable amount of energy are saved. Existing efforts in support data fusion or aggregation using structured approaches can be found in [4, 10]. Compared to centralized schemes, where all original data are gathered at the fusion center, aggregating local data at sensor nodes not only reduces energy consumption, but also maintains performances such as accuracy and fidelity [20]. Data fusion was also validated to benefit more powerful devices used in future sensor networks [12].

Since little experimental data from sensor network are available, many of current algorithms and sensornet protocols are evaluated by some statistic data models. For example, Gaussian distribution, yet simple, is an effective data acquisition model of sensor network and widely used by many researchers [3, 12]. However, Yu *et al* [21] evaluated some classic algorithms using data sets generated by several parametric statistic models: uniform, binomial, Gaussian, etc. They concluded that the performances of some problems were sensitive to the data model they choose. Our closed-form expressions suggest that the proposed scheduling strategy does not depend on any specific probability distribution, if the mean of a data set is given.

3 Problem Formulation

3.1 Target applications

The representative applications can be medical care application in a hospital; precision agriculture, monitoring soil and crop properties. A common feature among these applications is sensor nodes are organized into clusters. Each cluster has a fusion node, which acts as a local collector of sensor data from a set of source nodes.

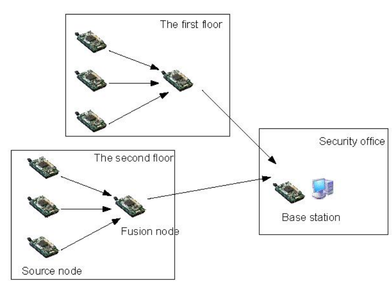


Figure 1. An example sensing application that uses the two-level hierarchical architecture.

For example, in a smart hospital building, sensors are deployed on each floor, monitoring the conditions including temperature, humidity, and patient activities. Each floor is equipped with one or more fusion node and several source nodes; the base station is set in the security office (Fig. 1).

Instead of just relaying sensor data to the base station, fusion nodes process the data using some aggregation methods and produce fused data. The fused data should meet the requirement specified by the application to complete the high-level task. Therefore, the base station should know the required amount of fused data as well. This is reasonable because the end application always requires the amount of input data to draw some conclusions.

Since most sensor nodes have only single channels, collisions happen when more than one source nodes send data to the fusion node. On the other hand, the fusion node or base station node can be at the idle status, when there is an undesirable gap between two data reports. This is also a waste of energy. Furthermore, the temporal property of these data should be considered when the base station aims to draw some time-critical conclusions on current conditions, e.g., fire detection, abnormal differences in temperature and humidity, and queries triggered by user applications.

3.2 Multi-hop hierarchical architecture

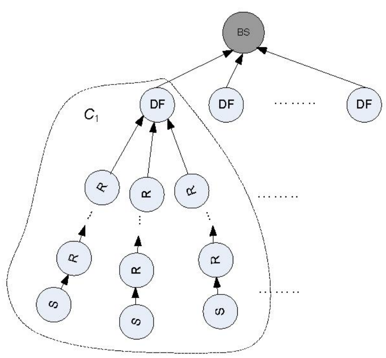


Figure 2. Multi-hop hierarchical tree architecture. BS: base station; DF: data fusion node; R: routing node; SN: source node.

Fig. 2 shows the hierarchical architecture with four kinds of nodes: base station, data fusion node, routing node, and source node. They are organized into *N* clusters with *n* source nodes in each cluster. Two kinds of data are produced in each cluster: original data measured by source nodes and fused data processed by fusion nodes. The total amount of original data (*M*) and fused data (*D*) are known as specified by the sensing task and high-level task of the application, respectively.

Following notations are used throughout the paper.

s_i: the ith source node in one cluster, $j = 1 \cdots n$.

M_i: the fraction of sensing data assigned to the ith cluster.

$m_{i,j}$: the fraction of sensed data of node j in cluster *i*.

D_i: fused data generated by the ith fusion node.

t_i: the finish time of data collection for the ith cluster.

T_i: the finish time of reporting fused data to the base station for the ith cluster.

S_c, S_m, S_p: the time taken to transmit, measure, and process a unit of workload, inversely proportional to the corresponding speeds.

The sensor network follows the realistic sequential computation and communication model (SSCM). For example, a source node can not transmit its sensor data until it completes the measuring task; a fusion node must perform processing data after all source nodes have reported their data. SSCM also means that at one time the data fusion node (base station node) receives data from one source node (one data fusion node). To formulate the scheduling models, we assume that measuring, communication, and processing time are compound measures, including all possible packet loss, synchronization overheads, and other uncertain factors.

4 Hierarchical Scheduling Strategies

4.1 Intra-cluster scheduling

One data fusion node and a set of routing nodes and source nodes constitute a working cluster, where each source node communicates with the fusion node through routing nodes in a multi-hop manner. One cluster has *b* branches from L_1 to L_b, each starting from a source node and having the same number of routing nodes, r. To complete certain amount of sensor readings in minimum finish time, the workload should be allocated to source nodes and scheduled to avoid transmission conflicts and idle time on the fusion node.

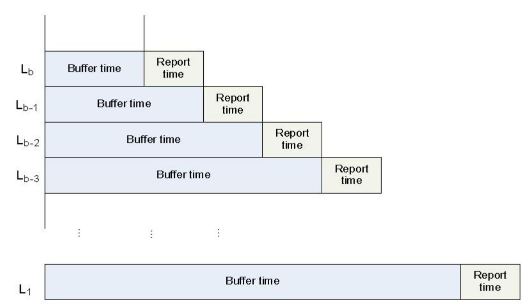


Figure 3. Timing diagram for intra-cluster scheduling. From bottom to top, source nodes take less measuring workload.

Fig. 3 illustrates the timing diagram for one cluster. From this figure, we observe that there is no time gap between every two successive branches because the divisible workload can be transferred in the cluster. Data transmission and throughput are independent to each branch until they arrive at the last routing node which is connected to the fusion node. At this time, communication should be coordinated to avoid collisions. We define the time from the source node starting to sense data until data arrive at the node next to the fusion node as buffer time. During report time the last routing node at each branch sends data to the fusion node. As a result, the proposed timing diagram minimizes the finish time by scheduling the measuring time and report time of each source node. Moreover, since the intra-cluster scheduling tries to avoid the transmission conflicts at the fusion node, energy spent on retransmission are conserved.

Furthermore, since at every active period of a large sensor network, only selected source nodes are needed to collect data; we can safely assume that *r* is much greater than *b*. From the timing diagram (Fig. 3), we derive a set of equations of the buffer time and report time of all the branches:

$$m_{i,1}S_m + m_{i,1}\sum_{i=1}^{r} S_c = m_{i,2}S_m + m_{i,2}\sum_{i=1}^{r+1} S_c \quad (1)$$

$$m_{i,2}S_m + m_{i,2}\sum_{i=1}^{r} S_c = m_{i,3}S_m + m_{i,3}\sum_{i=1}^{r+1} S_c$$

$$\cdots$$

$$m_{i,b-1}S_m + m_{i,b-1}\sum_{i=1}^{r} S_c = m_{i,b}S_m + m_{i,b}\sum_{i=1}^{r+1} S_c,$$

where i is the index of the cluster. In this model, the sensor data M is further distributed among N clusters as $M_1, M_2, \cdots, M_N$. Since all sensor data in the ith cluster sum to be M_i, the workload of source nodes in the cluster can be expressed as

$$m_{i,j} = H^{(j-1)}\frac{M_i}{1+\sum_{k=1}^{n-1} H^k}, \quad (2)$$

where $H = \frac{S_m + rS_c}{S_m+(r+1)S_c}$.

Note in the above equations, source nodes are ordered by the amount of workloads they take. It is not necessary that they are numbered in this way.

From (1) and (2), the largest workload for the source node can be solved as

$$m_{i,1} = \frac{M_i}{1+\sum_{k=1}^{n-1} H^k}. \quad (3)$$

When the node with the largest measuring data finishes transmission, the local cluster completes its assigned sensing workload. Then the finish time of collecting and transmitting data for the ith cluster is

$$t_i = \frac{M_i}{1+\sum_{k=1}^{n-1} H^k}(S_m + S_c). \quad (4)$$

For all source nodes in the network, their measurement (S_m) and transmission speed (S_c) are constants. Therefore, (4) shows that the finish time of each cluster depends on the sensing workload assigned to the cluster, which will be discussed in following sections.

4.2 Inter-cluster scheduling

The inter-cluster scheduler decides the amount of workloads to be assigned to each cluster, using the execution times of aggregation functions, transmission between fusion nodes and the base station, and in-cluster data collection derived in the previous section.

When aggregating data on the fusion node, original data are reduced to a smaller set of values, which aims to fulfill the high-level task of an application. From the perspective view of the whole sensor network, data fusion nodes can be viewed as intermediate nodes, which apply some aggregation function on the original data.

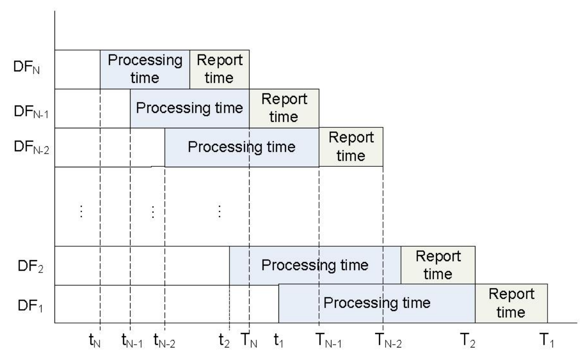


Figure 4. Timing diagram for inter-cluster scheduling. From bottom to top, data fusion nodes have less workloads.

The timing diagram of inter-cluster scheduling is shown in Fig. 4. As shown in this figure, each fusion node, DF_i, starts processing data at time t_i, which varies according to the workload assigned to each cluster. Similarly, to avoid conflicts and idle time, the inter-cluster scheduler allocates the entire task among clusters such that no gap exists between each two successive fusion nodes. Thus, the minimum finish time of a given task of the whole system is

$$T = max(T_1,\ T_2,\ \cdots,\ T_N). \quad (5)$$

We can obtain a set of equations from Fig. 4:

$$t_1 + M_1S_p = t_2 + M_2S_p + M_2u_2S_c \quad (6)$$

$$t_2 + M_2S_p = t_3 + M_3S_p + M_3u_3S_c$$

$$\cdots$$

$$t_{N-1} + M_{N-1}S_p = t_N + M_NS_p + M_Nu_NS_c.$$

Using (4), t_i $(i = 1 \cdots N)$ is expressed as

$$t_i = CM_i, \tag{7}$$

where $C = \frac{(S_m+S_c)}{1+\sum_{k=1}^{n-1} H^k}$. Then the workload assigned to each cluster is given by

$$M_i = M_1 - \frac{S_c}{C+S_p} \sum_{j=1}^{i-1} f(M_j),\ i = 2 \cdots N. \tag{8}$$

Since the total workload is assumed to be M, the summation of (8) satisfies

$$M = NM_1 - \frac{S_c}{C+S_p} \sum_{i=2}^{N} \sum_{j=1}^{i-1} f(M_j); \tag{9}$$

then we solve (9) for M_1 as follows

$$M_1 = \frac{1}{N}\left(M + \frac{S_c}{C+S_p} \sum_{i=2}^{N} \sum_{j=1}^{i-1} f(M_j)\right). \tag{10}$$

Thus, a closed-form solution for each task can be obtained by (8) for the case where the aggregation function is known. The minimized measuring, communication, and processing time of the system then is expressed as

$$T = T_1 = t_1 + M_1 S_p + f(M_1) S_c. \tag{11}$$

4.3 Data fusion model

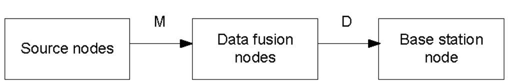


Figure 5. Data flow from source nodes to the base station. M: original data measured by source nodes; D: fused data produced by fusion nodes.

Fig. 5 illustrates the data flow from the sources nodes to the base station. As mention earlier, the fusion function results in a reduced size of output data. Thus, although the data fusion nodes in this hierarchical architecture receive all the data from sources, their outputs could be much smaller than the shared workload assigned by the inter-cluster scheduler. This results in significantly saving energy in communications. More complicated fusion functions include image, video stream, and stochastic data processing.

Based on the size of their outputs, fusion functions can be mainly classified into two categories.

Identical-size value (ISV)— The results of the fusion function for all input data have values of identical size, which can be a bit or multiple bits. For example, in a detection scenario, a fusion node indicates the appearance of a target with one or zero if not.

Multiple-size value (MSV)— These fusion functions use a smaller set of outputs with various data sizes to represent the original data by exploring the correlation of elements of the inputs. Usually, the size of fused data is proportional to that of the input data. For example, data fusion nodes can apply some lossy or lossless compression algorithms such as JPEG and Huffman coding, resulting in a compact output. Because ISV is in fact a special case of MSV, we only need to consider fusion functions of the second case.

The fusion function is formally defined as

$$f:\ M_i \rightarrow D_i, \tag{12}$$

which, for a data fusion node and the shared workload, transforms the original data to fused data. Therefore, substituting D_i in (10) yields

$$M_1 = \frac{1}{N}\left(M + \frac{S_c}{C+S_p} \sum_{i=2}^{N} \sum_{j=1}^{i-1} D_j\right). \tag{13}$$

Fused data from each cluster accumulated at the base station node sum to fulfill the requirement of the high-level task, which is

$$D = \sum_{i=1}^{N} D_i. \tag{14}$$

As mentioned earlier, using probabilistic models of sensor data is an effective method to model sensor data. Therefore, given a specific fusion function that produces a contraction expression of the original data, the fused data of sensor observations are assumed to follow *i.i.d.* probability distribution. Let D_i be a discrete random variable for the ith data fusion node that can assume any of the finite number N of different values in the set $\Gamma = \{d_1, \cdots, d_N |\ d_j < d_k,\ j < k\}$. N fusion nodes have N random variables with the same mean and variance values. We express the probability of D_i equal to d_j by the probability mass function $P(D_i)$. When N approaches ∞, we have

$$\mu = E[D_i] = \sum_{D_i \in \Gamma} D_i P(D_i) \tag{15}$$

Because the mean value gives a general impression of the behavior of some random variable without giving full details of its probability distribution, the term $\sum_{i=2}^{N} \sum_{j=1}^{i-1} D_j$ in (13) can be approximated by its mean value.

To solve (13), $\sum_{i=1}^{N}\sum_{j=1}^{i-1} D_j$ is expanded as

$$\sum_{i=2}^{N}\sum_{j=1}^{i-1} D_j = (N-1)D_1 + (N-2)D_2 + \cdots + D_{N-1}$$
$$= \sum_{i=1}^{N}(N-i)D_i. \quad (16)$$

Because the random variables $D_i,\ i = 1 \cdots N$, follow the same probability mass function, the mean value of the above equation is given by:

$$E\left[\sum_{i=1}^{N}(N-i)D_i\right] = N^2\mu - \frac{(N+1)N}{2}\mu. \quad (17)$$

Proof of (17). We rewrite (16) as

$$\Sigma_{i=1}^{N}(N-i)D_i = \sum_{i=1}^{N} ND_i + \sum_{i=1}^{N} iD_i. \quad (18)$$

Because N random variables D_i have identical probability mass function and are independent to each other. The mean value of the above equation is

$$E\left[\Sigma_{i=1}^{N}(N-i)D_i\right] = E\left[\sum_{i=1}^{N} ND_i\right] + E\left[\sum_{i=1}^{N} iD_i\right]$$
$$= N\sum_{i=1}^{N} E\left[D_i\right] + \sum_{i=1}^{N} iE\left[D_i\right] \quad (19)$$

Since we have known the mean value for each discrete random variable D_i by (15), (19) can be expressed in terms of μ as:

$$N\sum_{i=1}^{N}\mu + \sum_{i=1}^{N} i\mu. \quad (20)$$

Thus, (17) follows. □

Thus, we obtain the workload assigned to the last cluster to finish its task, as follows:

$$M_1 = \frac{M}{N} - \frac{S_c}{C+S_p}\frac{(N-1)}{2}\mu. \quad (21)$$

Then the amount of data that the rest of (N-1) clusters are assigned is solved by using the above equation in (8),

$$M_i = M_1 - \frac{S_c}{C+S_p}(i-1)\mu,\ i = 2\cdots N. \quad (22)$$

Finally, the finish time of the entire task given by (11) is

$$T = t_1 + M_1 S_p + \mu S_c, \quad (23)$$

where t_1 and M_1 are expressed by (4) and (21), respectively.

Table 1. Energy consumption models for different kinds of sensor nodes

	S	DF	R
Data acquisition	✓		
Data fusion		✓	
Communication	✓	✓	✓

5 Energy Consumption Model

In this section, we present the energy model of the multi-hop hierarchical model and derive the equations of energy consumption of individual sensor nodes in the network.

There are three kinds of energy consumption in the proposed sensor network model: data acquisition, data fusion, and communication. Table 1 lists the aspects of energy consumption of different nodes. The rest of the section describe each model in detail.

Data acquisition—In the proposed hierarchical network model, data are collecteded by source nodes equipped with some appropriate sensing devices. For example, some representative motes such as MICA2 and Tmote Sky have sensing devices for different purposes: temperature, humidity, light, vibration and so on. Optionally, users can expand the mote by installing more sensing units.

We denote the unit energy consumption as e_s for measuring one observation. We also assume that there is no energy consumption of the sensing device, when it is turned off after the node completes the sensing workload or before it starts to work. Therefore, for the jth source node in the ith cluster which is assigned the sensing workload $m_{i,j}$, the energy consumption on data acquisition will be $e_s \times m_{i,j}$.

Data fusion—Processing data at fusion nodes consumes energy of the microprocessor on the motes. A variety of fusion functions are to be used in different applications from basic arithmetic operations such as *min*, *max*, and *avg* to intensive computations like FFT and video image processing. The unit energy consumption of processing observed data is denoted by e_p.

One data fusion node collects original data from all source nodes in the same cluster. Because the collected data are the workloads assigned by the inter-cluster scheduling, the ith fusion node consumes $e_p \times M_i$ energy to process the data.

Communication—All source nodes connect to the data fusion node through a set of routing nodes; data fusion nodes connect to the base station directly. The energy for transmitting and receiving one unit of observed data is denoted by $e_{tx}d^2$ and e_{rx}, respectively; d is assumed to be the distance between the sender and the receiver. For example, when the jth source node in the ith cluster sends measured data to the fusion node, we can obtain their communication energy as

$$E_{tx}^{i,j} = e_{tx} d^2 m_{i,j};\ E_{rx}^{i} = e_{rx} m_{i,j}. \tag{24}$$

Data communication flows from the fusion nodes to the base station; if the shared workload M_i at the ith fusion node is fused to data size of D_i, the energy consumed at both sides are $E_{tx}^{i} = e_{tx} d^2 D_i$ and $E_{rx}^{BS} = e_{rx} D_i$.

So far, we have described three aspects of energy consumption for each kind of sensor nodes. The workloads shared by all source nodes and data fusion nodes are also derived, following the divisible load scheduling paradigm. Since all the sensor nodes are assumed to be homogeneous their capabilities of computing and communication and storage of battery energy are same. We outline the energy use for each kind of nodes, as follows:

Energy use for individual source nodes j in cluster i:

$$E_{i,j} = m_{i,j}(e_s + e_{tx} d^2). \tag{25}$$

Energy use for individual routing nodes:

$$Er_{i,j} = m_{i,j} e_{rx}. \tag{26}$$

Energy use for individual data fusion nodes:

$$E_i = M_i(e_{rx} + e_p) + M_i u_i e_{tx} d^2 \ i = 1 \cdots N. \tag{27}$$

6 Simulation Results

We have obtained the closed-form expressions of finish time for a given sensing task and energy use for individual sensor nodes for the hybrid hierarchical tree model. In this section, we investigate the effects of three system parameters– measurement, communication, and processing speed–on the finish time and energy consumption through numerical simulations.

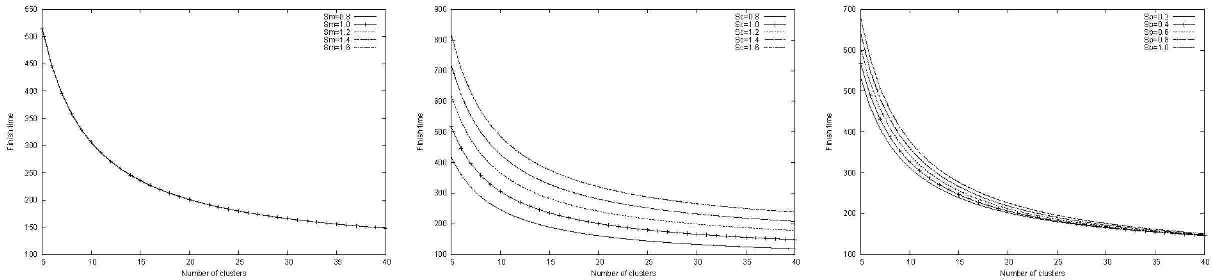
In our simulation, the amount of data all the source nodes need to measure is $M = 1000$. We assume that the required informative data after processing the original data gathered at the base station should be $L = 100$. We adopt the following energy parameters: transmitting a unit of sensor reading over a unit distance takes $e_{tx} = 200nJ$; receiving one unit of sensor reading consumes $e_{rx} = 250nJ$; measuring one unit of sensor reading needs $e_s = 100nJ$; processing one unit of observation consumes $e_p = 30nJ$; the distance between the sender and the receiver is $d = 100m$. There are 30 source nodes in each cluster; the number of routing nodes in between a source and a fusion node is 10. (Note that e_s depends on the sensors used for particular applications and could be of different orders of magnitude for different applications.)

First, the finish time against the number of clusters is plotted in Fig. 6. Since the number of clusters equals that of data fusion nodes in Fig. 2, increasing N leads to the increment of both data fusion nodes and source nodes. In Fig. 6(a), the value of S_m is chosen from 0.8 to 1.6, while S_c and S_p is fixed to 1.0. This figure shows that measurement speed almost does not affect the finish time because sensing takes a small fraction of the entire execution time. Fig. 6(b) and 6(c) show that when the communication or processing speed of sensor nodes increases, the finish time is reduced. However, the effect of S_p is not as significant as S_c. We can find that the five lines in Fig. 6(c) converge when N becomes large.

The second set of simulation is about the energy consumption of source nodes and data fusion nodes. The energy consumption of routing nodes depends on the sensing load of source nodes connected. Thus they are not considered here. We configure the system with 20 clusters, which also means it has 20 fusion nodes.

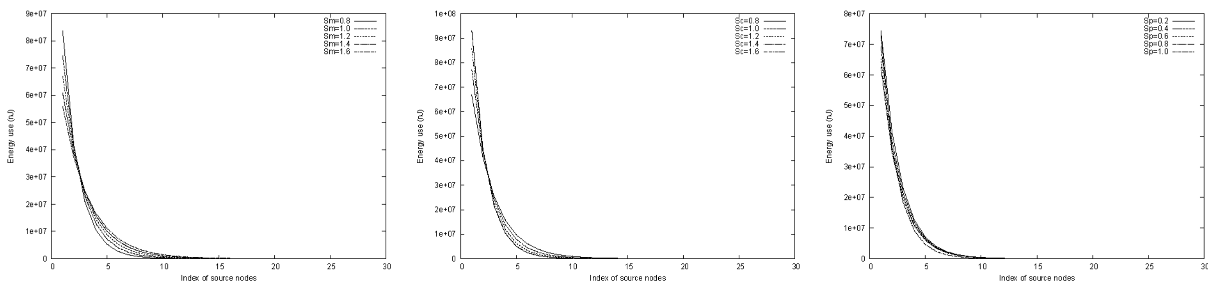
Without loss of generality, we choose the source nodes to study the energy consumption in the first cluster, which is assigned the most measuring workload as shown in Fig. 4. Fig. 7 presents the energy consumption of all the source nodes in the cluster as given by (25), where the source nodes are indexed from 1 to 30. In each case the energy consumption of source nodes monotonically decreases due to the reduced workload. In Fig. 7(a) and 7(b), the results using different S_m and S_c are illustrated. These two figures demonstrate two interesting observations. First, it is observed in both figures that there exists a threshold, which changes the effect of S_c and S_m. Another interesting observation is that when the index of source nodes increases in the cluster, higher S_m indicates less energy consumption on the nodes (Fig. 7(a)). However, in the same situation slower S_c results in less energy consumption (Fig. 7(b)). This is expected since the communication is sequential and the entire workload can be completed by less nodes with faster links. Because all source nodes are capable of measuring data simultaneously, the faster S_m is, the workload is distributed more evenly among all source nodes. As a result, each source can consume less energy as shown in Fig. 7(a). On the other hand, Fig. 7(c) shows that for the source node of the same index number, it consumes more energy if the processing speed of the fusion node is faster. This follows the fact that $m_{i,j}$ is obtained by M_1 in (2).

Another set of numerical evaluation shows that the energy consumption of data fusion nodes decreases from the first cluster to the last one as we expect from (6). The difference between each case in Fig. 8 is that S_m, S_c, and S_p have various effects. Intuitively, Fig. 8(a) shows that S_m does not change the value of energy use of individual fusion nodes. Fig. 8(b) and 8(c) illustrate that S_c and S_p have similar observed effects on the energy use of the fusion nodes to that



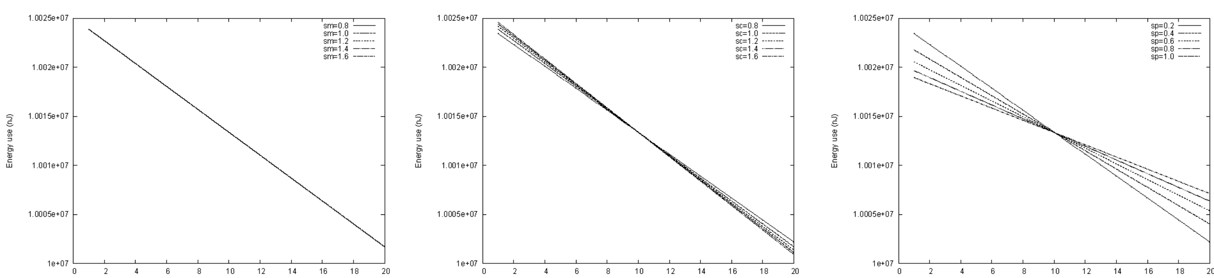
(a) Effect of measurement speed (S_m) (b) Effect of communication speed (S_c) (c) Effect of processing speed (S_p)

Figure 6. Measurement speed, communication speed, and processing speed against the finish time with the number of cluster increasing. (a) S_m varies with S_c and $S_p$fixed; (b) S_c varies with S_m and S_p fixed; (c) S_p varies with S_m and S_c fixed.



(a) Effect of measurement speed (S_m) (b) Effect of communication speed (S_c) (c) Effect of processing speed (S_p)

Figure 7. Measurement speed, communication speed, and processing speed against energy use of individual source nodes in cluster one. (a) S_m varies with S_c and S_p fixed; (b) S_c varies with S_m and S_p fixed; (c) S_p varies with S_c and S_m fixed.



(a) Effect of measurement speed (S_m) (b) Effect of communication speed (S_c) (c) Effect of processing speed (S_p)

Figure 8. Measurement speed, communication speed, and processing speed against energy use of individual fusion nodes. (a) S_m varies with S_c and S_p fixed; (b) S_c varies with S_m and S_p fixed; (c) S_p varies with S_c and S_m fixed.

of the source nodes, except for the shape of the curves.

From Fig. 7 and 8, we see that the energy consumption is not evenly spread among both source nodes in one cluster and fusion nodes. In all figures, energy consumption concentrates on the first several source nodes or fusion nodes. One solution to this problem is to employ some power-aware schemes [11], considering the residual energy at particular sensor nodes and rotating the workload scheduling sequence among sensor nodes periodically or based on an integrated utility. In this way, we not only minimize the execution time for each sensing task, but also balance the energy consumption among sensor nodes to prolong the lifetime of the entire sensor network.

7 Conclusion

We presented a coordinated sensing workload scheduling for a multi-hop hierarchical sensor network which is suitable for many monitoring applications. The proposed workload scheduling strategy builds on the divisible load theory. Two closed-form solutions were obtained to minimize finish time for intra-cluster and inter-cluster scheduling. We argue that our results are not sensitive to some probability distributions of fused data, if the mean value of the data set is fixed. The numerical simulation results showed that the proposed close-form solutions offer a tractable means to analyze the performance of a sensor network, regarding some system parameters. Our work introduces a thorough and realistic treatment of workload scheduling problems in WSNs. The results can be used as guidelines for the design of sensor networks considering the relationship between performance metrics and system parameters. An ongoing project is to implement and experimentally evaluate our scheduling strategies on real sensor platforms. The results in this paper also shed light on studying more complicated sensor network topologies, e.g., multi-level trees and arbitrary topological graphs.

References

[1] J. Cao, A. T. Chan, Y. Sun, S. K. Das, and M. Guo. A taxonomy of application scheduling tools for high performance cluster computing. *Cluster Computing*, 9(3):355–371, 2006.

[2] Y. Cheng and T. Robertazzi. Distributed computation with communication delay [distributed intelligent sensor networks]. *Aerospace and Electronic Systems, IEEE Transactions on*, 24(6):700–712, 1988.

[3] A. Deshpande, C. Guestrin, S. R. Madden, M. H. Joseph, and W. Hong. Model-driven data acquisition in sensor networks. In *Proceedings of the 30th VLDB Conference*, Toronto, 2004. http://db.lcs.mit.edu/madden/html/vldb04.pdf.

[4] K.-W. Fan, S. Liu, and P. Sinha. Scalable data aggregation for dynamic events in sensor networks. In *SenSys '06*, pages 181–194, New York, NY, USA, 2006. ACM Press.

[5] M. Gaynor, S. L. Moulton, M. Welsh, E. LaCombe, A. Rowan, and J. Wynne. Integrating wireless sensor networks with the grid. *IEEE Internet Computing*, 8(4):32–39, 2004.

[6] C. Hartung, R. Han, C. Seielstad, and S. Holbrook. Firewxnet: a multi-tiered portable wireless system for monitoring weather conditions in wildland fire environments. In *MobiSys 2006*, pages 28–41, New York, NY, USA, 2006. ACM Press.

[7] J. Hill, R. Szewczyk, A. Woo, S. Hollar, D. Culler, and K. Pister. System architecture directions for networked sensors. *SIGOPS Oper. Syst. Rev.*, 34(5):93–104, 2000.

[8] B. Hong and V. Prasanna. Optimizing a class of in-network processing applications in networked sensor systems. In *Mobile Ad-hoc and Sensor Systems, 2004 IEEE International Conference on*, pages 154–163, Oct. 2004.

[9] C. Intanagonwiwat, R. Govindan, and D. Estrin. Directed diffusion: a scalable and robust communication paradigm for sensor networks. In *MobiCom '00*, pages 56–67, New York, NY, USA, 2000. ACM Press.

[10] L. Jia, G. Noubir, R. Rajaraman, and R. Sundaram. Gist: Group-independent spanning tree for data aggregation in dense sensor networks. In *DCOSS'06*, San Francisco, 2006.

[11] H. Kang, X. Li, and P. J. Moran. Autonomic sensor networks: A new paradigm for collaborative information processing. In *DASC'06*, pages 258–268, New York, NY, USA, 2006.

[12] R. Kumar, M. Wolenetz, B. Agarwalla, J. Shin, P. Hutto, A. Paul, and U. Ramachandran. Dfuse: a framework for distributed data fusion. In *SenSys '03*, pages 114–125, New York, NY, USA, 2003. ACM Press.

[13] X. Li, V. Bharadwaj, and C. C. Ko. Divisible load scheduling on single-level tree networks with buffer constraints. *IEEE Transactions on Aerospace and Electronic Systems*, 36(4):1298–1308, 2000.

[14] S. Madden, M. J. Franklin, J. M. Hellerstein, and W. Hong. Tag: a tiny aggregation service for ad-hoc sensor networks. *SIGOPS Oper. Syst. Rev.*, 36(SI):131–146, 2002.

[15] M. Moges and T. G. Robertazzi. Wireless sensor networks: scheduling for measurement and data reporting. *Aerospace and Electronic Systems, IEEE Transactions on*, 42(1):327–340, 2006.

[16] S. Nath, J. Liu, J. Miller, F. Zhao, and A. Santanche. Sensormap: a web site for sensors world-wide. In *SenSys '06*, pages 373–374, 2006.

[17] T. G. Robertazzi. Ten reasons to use divisible load theory. *Computer*, 36(5):63–68, 2003.

[18] H. O. Sanli, R. Poornachandran, and H. Çam. Collaborative two-level task scheduling for wireless sensor nodes with multiple sensing units. In *IEEE SECON 2005 proceedings*, pages 350–361, 2005.

[19] B. Veeravalli, D. Ghose, V. Mani, , and T. G. Robertazzi. *Scheduling Divisible Loads in Parallel and Distributed Systems*. IEEE Computer Society Press, Los Almitos, California, 1996.

[20] L. Yu, L. Yuan, G. Qu, and A. Ephremides. Energy-driven detection scheme with guaranteed accuracy. In *IPSN*, pages 284–291, 2006.

[21] L. Yu, L. Yuan, G. Qu, and A. Ephremides. Energy-driven detection scheme with guaranteed accuracy. In *IPSN '06*, pages 284–291, New York, NY, USA, 2006. ACM Press.

[22] Y. Yu and V. K. Prasanna. Energy-balanced task allocation for collaborative processing in wireless sensor networks. *Mob. Netw. Appl.*, 10(1-2):115–131, 2005.

[23] F. Zhao and L. Guibas. *Wireless Sensor Networks: An Information Processing Approach*. Morgan Kaufmann Publishers, 2004.