Wireless Sensor Networks: Scheduling for Measurement and Data Reporting

MEQUANINT MOGES, Member, IEEE
University of Houston

THOMAS G. ROBERTAZZI, Fellow, IEEE
SUNY Stony Brook

An optimal load allocation approach is presented for measurement and data reporting in wireless sensor networks with a single level tree network topology. The measurement problem investigated involves a measurement space, part of which can be sampled by each sensor. We seek to optimally assign sensors part of the measurement space to minimize reporting time and energy usage. Three representative measurement and reporting strategies are studied. This work is novel as it considers, for the first time, the measurement capacity of processors and assumes negligible computation time which is radically different from the traditional divisible load scheduling research to date. Aerospace applications include satellite remote sensing and monitoring and sensor networks deployed and monitored from the air.

Manuscript received February 3, 2005; revised August 31, 2005; released for publication August 31, 2005.

IEEE Log No. T-AES/42/1/870614.

Refereeing of this contribution was handled by M. G. Simoes.

This work was supported by the NSF under Grant CCR-99-12331.

Authors' addresses: M. Moges, Dept. of Engineering Technology, University of Houston, Houston, TX 77204-4021; T. G. Robertazzi, Dept. of Electrical and Computer Engineering, Cosine Laboratory, SUNY Stony Brook, Stony Brook, NY 11794, E-mail: (tom@ece.sunysb.edu).

I. INTRODUCTION

Because of its diverse applications, divisible load theory has been intensively studied over the past decade or so. Divisible load scheduling theory (DLT) involves the study of an optimal distribution of partitionable loads among a number of processors and links [1–4]. A partitionable data parallel load is one that can be arbitrarily distributed among the processors and links in a system. Thus there are no precedence relations among the data. There has been an increasing amount of study on DLT since 1988. Most of these studies develop an efficient allocation of load to processors over a network by considering the processing and communication time as the main parameters of the system. Thus the objective is to obtain an optimal partition of the processing load to a network of processors connected via communication links such that the entire load can be distributed and processed in the shortest possible amount of time.

There are many potential optimization problems involving integrating sensing with communication and/or computation. The type of application we envision involves measurements from a sample space where each sensor is assigned a certain nonoverlapping part of the sample space to measure. For instance, the sample space may consist of a very large frequency range and performing measurements by a single sensor may be both time and energy consuming. Our optimization problem is to allocate which part of the frequency range each sensor should measure so as to do the sensing and communication in a minimal amount of time and with a minimum amount of energy. A typical example may be the employment of sensors where each sensor in the network is made to measure a specific frequency and directional range so that the report time will be minimum. Such sensors may be carried on satellites. Another typical application may be a very large array of antennas that produces images of the radio sky at a wide range of frequencies and directions. A search for extraterrestrial signals may involve each antenna scanning a different frequency and directional range.

We emphasize here wireless sensor network applications. The processors are assumed to have a certain measurement capacity. In this situation, we assume that computation will be done at the controller level, once all the measured data is gathered form each sensor. This assumption leads us to consider only the measurement and communication times and neglect the computation time of the sensors. Another reason for neglecting computation here is that in wireless sensor network applications communication time dominates computation time. There may be different strategies by which the controller communicates with the processors and the processors measure and report their data. In this study we consider features including such choices as whether reporting is sequential or simultaneous (concurrent) across processors as well as whether the measurement and reporting processes overlap.

The study considers both heterogeneous and homogeneous networks. That is the network elements may possess different measurement capacities and link speeds or the same measurement capacities and link speeds. For homogeneous networks, one can find a closed-form equation by which one can obtain the optimal share of the load that has to be assigned to each processor in the network in order to achieve minimum measurement and report time.

Besides the minimum measurement/report time analysis, energy use is a key feature in networks such as wireless sensor networks. In sensor networks, finish time performance optimization methods will need to be extended and combined in new ways in order to provide service that meets the needs of applications with the minimum possible energy consumption. To do so, this paper considers the energy consumption by the different reporting strategies and results are compared in terms of each processor's energy use.

The organization of this paper is as follows. In Section II, the literature to date in the areas of DLT and wireless sensor networks is briefly presented. Section III discusses the system model and some notations used in this work. The analysis of the measurement and reporting time processes using divisible load theory as a starting point is presented in Section IV. The corresponding energy use models for the various measurement and data reporting strategies in single level tree networks are also presented in Section V. In Section VI, a performance evaluation of the various strategies appears. Finally the conclusion appears in Section VII.

II. LITERATURE TO DATE

A. Divisible Load Scheduling Theory

The problem of minimizing processing time and/or energy consumption of wireless sensor networks has received significant attention over the last few years. The optimization process involves a need for efficient allocation of sensor loads to processors as well as links. Divisible load theory is used to allow tractable performance analyses of systems incorporating sensing, communication and computation aspects. In divisible load theory, an arbitrarily divisible load without precedence relations is partitioned and distributed among processors in a multi-computer system so that the entire load is processed in the shortest possible amount of time.

Since the original work on algebraic solutions for divisible load scheduling models by Cheng and Robertazzi [5] in 1988, there has been an increasing amount of study on DLT. Most of these studies develop an efficient allocation of load to processors over a network by forcing all of the processors to stop processing at the same time. Intuitively, this is because the solution could be improved by transferring load if some processors were idle while others are still busy [6]. Solutions for optimal allocation of loads to links and processors using a set of recursive equations were presented for network topologies including linear daisy chains [5], bus networks [7], tree networks [8], and arbitrary networks [9]. For complex networks, such as multilevel tree networks, the concept of equivalent networks was presented in [10]. There have been further studies in terms of load distribution policies for two- and three-dimensional meshes [11] and hypercubes [12]. In [13] the concept of time-varying processor speed and link speed are introduced. In [14] the integration of monetary cost optimization and divisible load theory is presented. Scheduling policy research includes independent task scheduling [15, 16], multi-installment sequential scheduling [17], multi-round algorithms [18], fixed communication charges [4], detailed parameterization and solution reporting time optimization [19] and combinatorial optimization [20]. Recently, though divisible load theory is fundamentally a deterministic theory, a study has been done to show some equivalence to Markov chain models [21]. An important reason for using DLT is its tractability, flexibility, and realism for a large class of data intensive, data parallel, computational problems.

B. Wireless Sensor Networks

In recent years, wireless sensor networks have attracted significant attention due to their integration of wireless, computer, and sensor technology. Wireless sensor networks consist of a multiplicity of nodes that are equipped with processing, communicating and sensing capabilities, and use ad-hoc radio protocols to forward data in a multi-hop mode of operation. Each node in such a network has limited energy resources, hence, minimizing energy usage in wireless sensor networks becomes important. The majority of literature on wireless sensor networks involves minimizing communication since this is the dominant operation in time. In [22] it is reported that to send 1 bit 100 m away uses the same energy as 3000 instructions on a microprocessor. Representative work on energy conservation strategies have included aggregating data at nodes to shorten subsequent transmissions [23], probabilistically routing traffic to spread load across network nodes and prolong stored energy [24], putting sensors to sleep when they are not needed [25], and activating only geographically localized wireless sensors [26]. In terms of the computation efficiency of sensor networks, there has been research on sorting [27] and on computational problems in distributed sensor networks [28]. It is clear that the uneven characteristics of wireless

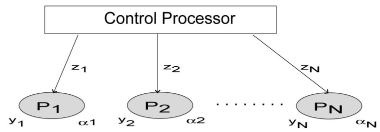


Fig. 1. Single level tree network with controller.

sensor networks functionality require a rethinking of integrated distributed processing algorithms and protocols. The following section describes a methodology for wireless sensor networks that combines both the communication and measurement aspects of the network.

III. SYSTEM MODEL DESCRIPTION

In this section, the various network parameters used here are presented along with some notation and definitions. The network topology discussed in this study is the single level tree (star) network consisting of one control processor and N communicating processors as shown in Fig. 1. It is assumed that the total load considered here is of the arbitrarily divisible kind that can be partitioned into fractions of loads to be assigned to each processor over a network. In this case the control processor first assigns a load share to be measured to each of the N processors and then receives the measured data from each processor. Each processor begins to measure its share of the load once the measurement instructions from the controller have been completely received by each processor. Naturally, other assumptions are possible, such as having each processor commence measurements as soon as receiving measurement instructions. However this is beyond the scope of this paper. We also assume that computation time is negligible compared with communication and measurement time. This is a reasonable assumption in some situations for the reasons given above.

Some of the scheduling strategies considered in this study have a time reversed dual nature with respect to standard divisible load models involving only communication and computation, and are discussed in the following section.

A. Notation and Definitions

α_i The fraction of load that is assigned to processor i by the control processor.

y_i A constant that is inversely proportional to the measuring speed of processor i in the network.

z_i A constant that is inversely proportional to the communication speed of link i in the network.

T_{ms} Measurement intensity constant. This is the time it takes the ith processor to measure the entire load when $y_i = 1$. The entire assigned measurement load can be measured on the ith processor in time $y_i T_{\mathrm{ms}}$.

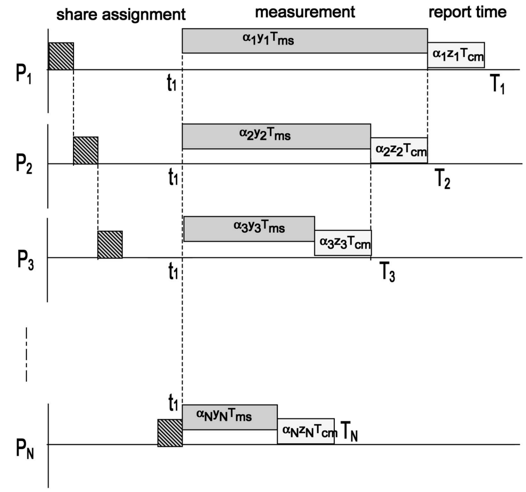


Fig. 2. Timing diagram for single level tree network with controller and sequential reporting time.

T_{cm} Communication intensity constant. This is the time it takes to transmit all of the measurement load over a link when $z_i = 1$. The entire load can be transmitted over the ith link in time $z_i T_{\mathrm{cm}}$.

T_i The total time that elapses between the beginning of the scheduling process at $t = 0$ and the time when processor i completes its reporting, $i = 0, 1, \ldots, N$. This includes, in addition to measurement time, reporting time and idle time.

T_f This is the time when the last processor finishes reporting (finish time or make-span).

$$T_f = \max(T_1, T_2, \ldots, T_N).$$

In all of the sections the same definitions are used for α_i, y_i, z_i, T_{ms}, and T_{cm} unless otherwise stated. Another convention that is followed in this case is that the load assignment originating at the control processor is assumed to be normalized to be a unit load.

IV. MEASUREMENT AND REPORTING TIME ANALYSIS

A. Simultaneous Measurement Start, Sequential Reporting

The timing diagram, Fig. 2, shows that at time $t = 0$, the processors are all idle and the control processor starts to communicate with the first processor in the network. This process of communication continues and by time $t = t_1$, each processor will receive its measurement instructions from the control processor. This may correspond to a situation where measurements shall commence at the same time at each processor. After measurements are made, we assume that only one processor may report back to the root processor at a time (i.e., there is a single channel).

The equations given below are used to find the optimal allocations of load that minimize $T_f - T_1$ in the context of this particular scheduling policy and interconnection topology. It is interesting to note that if time is reversed, the timing diagram of Fig. 2 for measurement and reporting time is equivalent to standard divisible load models of computation and communication for sequential distribution in single level tree [7]. In this case the processors receive their share of load from the root processor sequentially and start computation after completely receiving their share of load. The equations that govern the relations among various variables and parameters in the network shown in Fig. 2 can be written as follows:

$$T_1 = t_1 + \alpha_1 y_1 T_{\text{ms}} + \alpha_1 z_1 T_{\text{cm}} \tag{1}$$

$$T_2 = t_1 + \alpha_2 y_2 T_{\text{ms}} + \alpha_2 z_2 T_{\text{cm}} \tag{2}$$

$$\vdots$$

$$T_N = t_1 + \alpha_N y_N T_{\text{ms}} + \alpha_N z_N T_{\text{cm}}. \tag{3}$$

As mentioned earlier, since the total measurement load originating at the control processor is assumed to be normalized to a unit load, the fractions of the total processing load should sum to one:

$$\alpha_1 + \alpha_2 + \alpha_3 + \cdots + \alpha_{N-1} + \alpha_N = 1. \tag{4}$$

Based on the above equations and the timing diagram shown in Fig. 2, one can write the following set of equations:

$$\alpha_1 y_1 T_{\text{ms}} = \alpha_2 y_2 T_{\text{ms}} + \alpha_2 z_2 T_{\text{cm}} \tag{5}$$

$$\alpha_2 y_2 T_{\text{ms}} = \alpha_3 y_3 T_{\text{ms}} + \alpha_3 z_3 T_{\text{cm}} \tag{6}$$

$$\vdots$$

$$\alpha_{N-2} y_{N-2} T_{\text{ms}} = \alpha_{N-1} y_{N-1} T_{\text{ms}} + \alpha_{N-1} z_{N-1} T_{\text{cm}} \tag{7}$$

$$\alpha_{N-1} y_{N-1} T_{\text{ms}} = \alpha_N y_N T_{\text{ms}} + \alpha_N z_N T_{\text{cm}}. \tag{8}$$

A general expression for the above set of recursive equations can be written as

$$\alpha_i = s_i \alpha_{i-1} \tag{9}$$

where $s_i = y_{i-1} T_{\text{ms}} / (y_i T_{\text{ms}} + z_i T_{\text{cm}})$ and $i = 2,3,\ldots,N$. The above recursive equation for α_is can be rewritten in terms of α_1 only as

$$\alpha_i = \prod_{j=2}^{i} s_j \alpha_1. \tag{10}$$

Now using the above sets of equations and the normalization equation, one can solve for α_1 as

$$\alpha_1 + \sum_{i=2}^{N} \prod_{j=2}^{i} s_j \alpha_1 = 1. \tag{11}$$

The above equation can be simplified as

$$\alpha_1 = 1 \Big/ \left(1 + \sum_{i=2}^{N} \prod_{j=2}^{i} s_j\right). \tag{12}$$

The control processor will use the above value of α_1 to obtain the amount of data that has to be measured by the rest of the $N-1$ processors by using the following equation:

$$\alpha_i = \prod_{j=2}^{i} (s_j) \Big/ \left(1 + \sum_{i=2}^{N} \prod_{j=2}^{i} s_j\right) \tag{13}$$

where $i = 2,3,4,\ldots,N$.

The minimum measuring and reporting time of the network will then be given as

$$T_f = t_1 + (y_1 T_{\text{ms}} + z_1 T_{\text{cm}}) \Big/ \left(1 + \sum_{i=2}^{N} \left(\prod_{j=2}^{i} s_j\right)\right). \tag{14}$$

A closed-form solution can be obtained for the above expression, for the case of homogeneous networks. In this case one can solve for α_1 for a homogeneous network as

$$\alpha_1 (1 + s + s^2 + \cdots + s^{N-2} + s^{N-1}) = 1 \tag{15}$$

where $s = y T_{\text{ms}} / (y T_{\text{ms}} + z T_{\text{cm}})$, for the case where the communication link and measuring speed are assumed to be homogeneous.

The above equation can be simplified as

$$\alpha_1 = (1 - s)/(1 - s^N). \tag{16}$$

Similarly, the control processor will use the value of α_1 to obtain the amount of data that has to be measured by the rest of the $N-1$ processors by using the following equation:

$$\alpha_i = \alpha_1 s^{i-1} \tag{17}$$

where $i = 2,3,4,\ldots,N$.

The minimum measuring and reporting time of the homogeneous network will then be given as

$$T_f = t_1 + (y T_{\text{ms}} + z T_{\text{cm}})(1 - s)/(1 - s^N). \tag{18}$$

This measurement and reporting time of the network approaches $t_1 + z T_{\text{cm}}$ as N approaches ∞, which conforms to the result shown in [7]. This result can be proved analytically as follows. As N approaches ∞, the expression $(1 - s)/(1 - s^N)$ approaches $(1 - s)$. Now using the definition of s, one can easily obtain

$$1 - s = z T_{\text{cm}} / (y T_{\text{ms}} + z T_{\text{cm}}). \tag{19}$$

Then substituting this result back in T_f gives

$$T_f = t_1 + z T_{\text{cm}}. \tag{20}$$

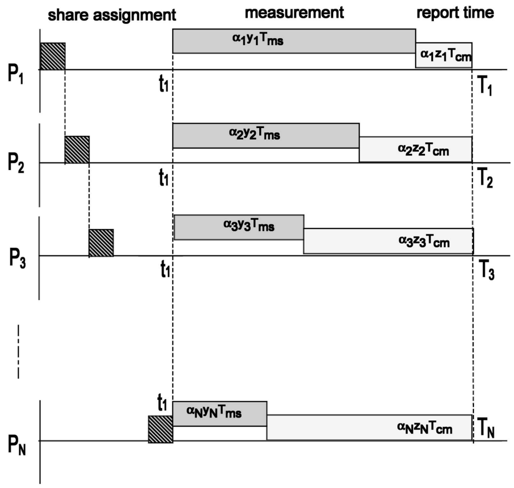


Fig. 3. Timing diagram for single level tree network with controller and simultaneous reporting termination.

Intuitively, the measurement latency is "hidden" by the reporting latency. The optimality condition discussed above is actually not always true for the general class of tree networks. For certain network parameter values, it may not be essential to utilize all the processors to achieve optimal performance. This fact is proved through a rigorous analysis in [2].

B. Simultaneous Measurement Start, Simultaneous Reporting Termination

The network topology that is presented in this section is similar to that discussed in the previous section except for the fact that each of N processors in the network finish reporting at the same time. That is, the network will have the same report finishing time for each processor. This is possible if each processor has a separate channel to the root. The timing diagram of the network is shown in Fig. 3. Again there is a time-reversed dual of this model in terms of standard models of computation and communication only. It involves concurrent distribution of load from a root in a single level tree [29]. In this case the processors receive their share of load from the root processor concurrently and start computation after completely receiving their share of load.

Note that the assignments of measurement share could be concurrent with multiple channels as long as all load share assignments are distributed by time t_1. As shown in the timing diagram each processor begins to measure its share of the load at the moment that all finish receiving their measurement instructions from the control processor.

Using the definitions and notations given earlier, the equations that relate the various variables and parameters together are given as

$$T_1 = t_1 + \alpha_1 y_1 T_{\mathrm{ms}} + \alpha_1 z_1 T_{\mathrm{cm}} \tag{21}$$

$$T_2 = t_1 + \alpha_2 y_2 T_{\mathrm{ms}} + \alpha_2 z_2 T_{\mathrm{cm}} \tag{22}$$

$$\vdots$$

$$T_N = t_1 + \alpha_N y_N T_{\mathrm{ms}} + \alpha_N z_N T_{\mathrm{cm}}. \tag{23}$$

Also the fractions of the total measurement load should sum to one:

$$\alpha_1 + \alpha_2 + \alpha_3 + \cdots + \alpha_{N-1} + \alpha_N = 1. \tag{24}$$

In this case since all processors stop reporting at the same time, we have

$$T_1 = T_2 = T_3 = \cdots = T_N.$$

Based on the above equations and the timing diagram shown in Fig. 3, one can write the following set of equations:

$$\alpha_1 r_1 = \alpha_2 r_2 \tag{25}$$

$$\alpha_2 r_2 = \alpha_3 r_3 \tag{26}$$

$$\vdots$$

$$\alpha_{N-2} r_{N-2} = \alpha_{N-1} r_{N-1} \tag{27}$$

$$\alpha_{N-1} r_{N-1} = \alpha_N r_N \tag{28}$$

where $r_i = y_i T_{\mathrm{ms}} + z_i T_{\mathrm{cm}}$, $i = 1,2,\ldots,N$. Using the above set of equations, one can now write αs as a function of r_i as

$$\alpha_i = (1/r_i) \Big/ \sum_{i=1}^{N} (1/r_i). \tag{29}$$

This is the optimal allocation of load to minimize $T_f - T_1$ in the context of this scheduling policy and interconnection topology. From the above expression, it can be easily seen that the share of each processor will entirely depend on the combined speed of the measurement and communication of that processor. That is, intuitively, processors with faster combined measurement and link speeds receive a larger share than processors with slower combined measurement and link speeds. The minimum measurement and reporting time of the network will then be given as

$$T_f = T_i = t_1 + (y_i T_{\mathrm{ms}} + z_i T_{\mathrm{cm}})(1/r_i) \Big/ \sum_{i=1}^{N} (1/r_i). \tag{30}$$

For the case of a homogeneous network of measurement and link speeds, the simultaneous reporting time strategy allows each processor in the network to share the load equally. That is, $\alpha_i = 1/N$, for $i = 1,2,3,\ldots,N$.

In this case the minimum measuring and reporting time of the network will then be given as:

$$T_f = t_1 + (yT_{\mathrm{ms}} + zT_{\mathrm{cm}})/N. \tag{31}$$

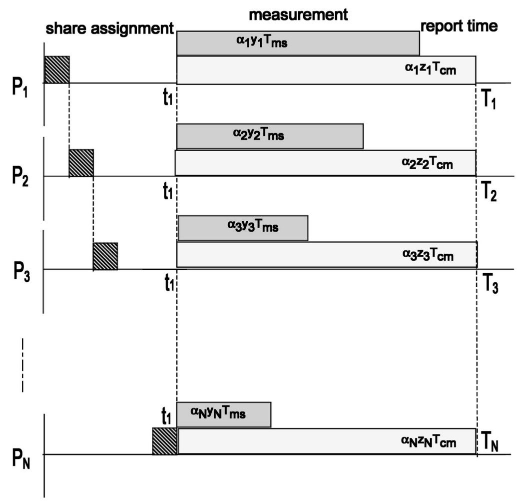


Fig. 4. Timing diagram for single level tree network with controller and CMR time.

C. Concurrent Measurement and Reporting

The network topology that is presented in this section is similar to that discussed in the previous section except for the fact that each of the N processors in the network contains a coprocessor so that the processors may be able to measure and report data at the same time. Thus, each processor after receiving its measurement instructions immediately begins reporting back to the control processor while measuring its share of the load. In this case it is assumed that the measurement time is about one order of magnitude smaller than the reporting time ($y_i T_{\mathrm{ms}} < z_i T_{\mathrm{cm}}$), in order to allow some time for the last measured data to be reported back to the controller. The timing diagram of the network is shown in Fig. 4. Note that this model is analogous to an equivalent single level tree model involving only computation and communication. In this time nonreversed model load is distributed concurrently on all links and computation starts as soon as load begins to be received. For this dual, measurement is equivalent to communication in the standard model and communication in the model of this section is equivalent to computation in the standard model [30].

In a similar way as in the previous sections, the equations that relate the various variables and parameters together are given as

$$T_1 = t_1 + \alpha_1 z_1 T_{\mathrm{cm}} \tag{32}$$

$$T_2 = t_1 + \alpha_2 z_2 T_{\mathrm{cm}} \tag{33}$$

$$\vdots$$

$$T_N = t_1 + \alpha_N z_N T_{\mathrm{cm}}. \tag{34}$$

Also the fractions of the total measurement load should sum to one:

$$\alpha_1 + \alpha_2 + \alpha_3 + \cdots + \alpha_{N-1} + \alpha_N = 1. \tag{35}$$

In this case since all processors stop reporting at the same time, we have

$$T_1 = T_2 = T_3 = \cdots = T_N.$$

Based on the above equations and the timing diagram shown in Fig. 4, one can write the following set of equations:

$$\alpha_1 z_1 T_{\mathrm{cm}} = \alpha_2 z_2 T_{\mathrm{cm}} \tag{36}$$

$$\alpha_2 z_2 T_{\mathrm{cm}} = \alpha_3 z_3 T_{\mathrm{cm}} \tag{37}$$

$$\vdots$$

$$\alpha_{N-2} z_{N-2} T_{\mathrm{cm}} = \alpha_{N-1} z_{N-1} T_{\mathrm{cm}} \tag{38}$$

$$\alpha_{N-1} z_{N-1} T_{\mathrm{cm}} = \alpha_N z_N T_{\mathrm{cm}}. \tag{39}$$

As in the previous case, using the above set of equations, one can now write αs as a function of z_i as

$$\alpha_i = (1/z_i) \Big/ \sum_{i=1}^{N} (1/z_i). \tag{40}$$

Again this is the allocation of load that minimizes $T_f - T_1$ in the context of this particular scheduling policy and interconnection topology. From the above expression, one can see that the share of load for each processor in this case will entirely depend only on the speed of the communication of that processor. That is, intuitively, processors with faster link speeds receive a larger share than processors with slower link speeds. The minimum measurement and reporting time of the network will then be given as

$$T_f = T_i = t_1 + (z_i T_{\mathrm{cm}})(1/z_i) \Big/ \sum_{i=1}^{N} (1/z_i). \tag{41}$$

As can be seen from the above set of equations, the processors will share the load equally when the network is homogeneous. This result is similar to the result obtained from the previous strategy, however, the measurement and reporting time in this case is given as

$$T_f = t_1 + z T_{\mathrm{cm}}/N. \tag{42}$$

V. WIRELESS ENERGY USE

Over the last few years the problem of minimizing the energy use of wireless networks has witnessed a great deal of interest on the part of many researchers. Some of the studies based on the problem of minimum energy broadcasting in wireless networks include [31, 32]. In these studies each node in a network is able to adjust its transmission power by selecting routes that optimize performance. This approach maximizes the overall network lifetime by distributing energy consumption fairly.

In this section the energy model used by each load measurement and reporting strategy discussed

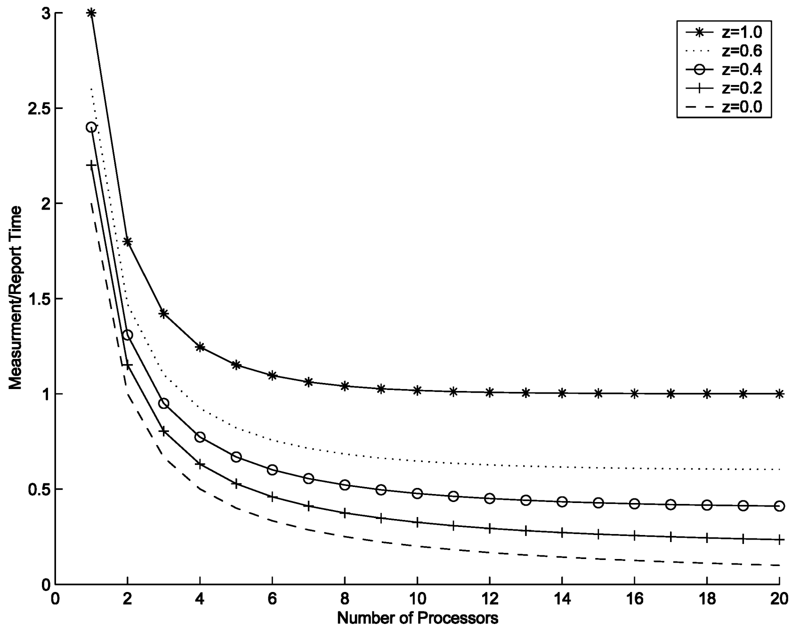


Fig. 5. Measurement/report time versus number of processors and variable inverse link speed z for single level tree network with controller and sequential reporting time.

previously is presented. The model assumes a first-order radio model [33] which considers different assumptions about the radio characteristics, including energy dissipation in the transmit and receive modes. To see this effect, we consider the following parameters used in a simple model radio design:

E_{elec}: transmitter/receiver electronics. This is the energy dissipated to run the transmitter/receiver circuitry. Usually, $E_{\text{elec}} = 50$ nJ/bit.

ε_{amp}: transmit amplifier. This is the energy dissipated by the transmit amplifier to achieve an acceptable signal-to-noise ratio. Usually, $\varepsilon_{\text{amp}} = 100$ pJ/bit/m^2.

Thus, the energy dissipated by processor i in order to transmit an α_i-bit data message to a distance of d units is given as

$$E_{Tx} = E_{\text{elec}} * \alpha_i + \varepsilon_{\text{amp}} * \alpha_i * d^2. \tag{43}$$

Similarly, the energy dissipated to receive the same amount of data from a distance d units is

$$E_{Rx} = E_{\text{elec}} * \alpha_i. \tag{44}$$

In this case each processor is assumed to have a direct radio connection with the control processor. When reporting the measured data, each processor sends its data directly to the control processor.

Based on the above equations, an energy use comparison of the three load measurement and reporting strategies discussed earlier is discussed below. To do so, in each case the energy use was obtained by considering only the reporting time and neglecting communication time utilized when the control processor informs each of the processors their load assignment. This is reasonable as load assignment information is relatively concise. In this case the energy use comparison takes into account both the total energy consumption of the network as well as energy consumption of each processor.

VI. PERFORMANCE EVALUATION

The minimum measurement and reporting time and energy use expressions obtained in the previous sections are used to study the effect of the communication link speed, the measurement speed, and the number of processors in the network on the minimum measurement and reporting time and energy use of the network. To do so, we consider the following two cases. In the first case the measurement and reporting time/energy use is plotted against the number of processors when z is varied and measurement speed y is fixed. In the second case, the measurement and reporting time/energy use is plotted against the number of processors when z is fixed and measurement speed y is varied. The energy use plots are given for the simultaneous measurement start, sequential reporting (SMS^{2}R) strategy only. In the simultaneous measurement start, simultaneous reporting termination (SMS^{2}RT) and concurrent measurement and reporting (CMR) strategies, the energy use of individual processors is the same for a homogeneous network.

A. SMS^{2}R Strategy

In Fig. 5, the measurement/report time is plotted against the number of homogeneous processors

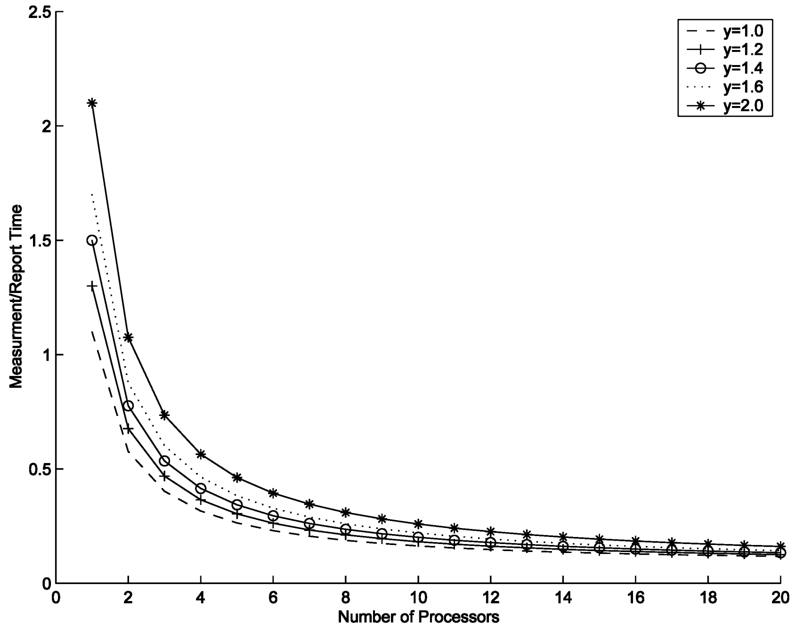


Fig. 6. Measurement/report time versus number of processors and variable inverse measuring speed y for single level tree network with controller and sequential reporting time.

when the value of the communication speed z is varied from 0 to 1 and the value of measurement speed y is fixed to be 2. In all cases $T_{\mathrm{cm}} = 1$ and $T_{\mathrm{ms}} = 1$. It can be shown from the figure that the faster the communication speed, the smaller the measurement/report time. It is also shown that the measurement/report time levels off after a certain number of processors for each performance curve.

Fig. 6 on the other hand shows for the case when the inverse measuring speed y is varied from 1 to 2 and the inverse link speed z is fixed to be 0.1. The result confirms, as mentioned earlier, that the measurement time approaches zT_{cm}, which in this case is 0.1, as N approaches ∞.

Figs. 5 and 6 are similar in shape to curves of solution time versus number of processors appearing in [2] and [5] where only communication and computation speeds are considered. It points to a law of diminishing returns for performance (i.e., measurement/reporting time) in adding additional processors for this protocol.

The energy use by each node in the simultaneous measurement start, sequential reporting case is shown in Fig. 7. The value of the inverse link speed z is varied from 0 to 1.0 while the inverse measuring speed y is fixed to be 2. We have also assumed the distance $d = 100$ m for each sensor as our main objective is to see the effect of variation of communication link and measuring speeds on the reporting time and energy usage. As shown in the plot, the energy use of the individual nodes is unevenly distributed as it depends mainly on the amount of load assigned to each node. Fig. 7 clearly shows that as the speed of the communication link is decreased the total number of nodes which are doing substantial processing is reduced, while the rest are not assigned any load or are simply assigned very small amounts of data measurements. This is expected as the communication is sequential and with a slow communication link effectively the number of processors needed to finish processing the load becomes less. The straight line in the plot is the case for high speed communication where all the processors will be able to participate in the job processing. In this situation the energy usage will be the same for all processors since the processors share the same amount of load.

These results demonstrate uneven energy usage for this protocol that can lead to some sensors (the first to receive load) becoming depleted in energy before others. This can be mitigated by randomizing the load distribution order from load to load.

Similarly, Fig. 8 shows that as the inverse measurement speed y is decreased, since the total load is partitioned among only few of the processors, the energy use by those active processors will be very high as compared with the others which receive very small or no load fraction. However as the inverse measurement speed y is increased, this effectively increases the number of participating processors and hence the energy use is distributed fairly among all participating processors. In this case the inverse measuring speed y is varied from 1 to 2 and the inverse link speed z is fixed to be 0.1.

B. SMS^{2}RT Strategy

In this section, results of the measurement/report time for the case of simultaneous reporting termination is presented.

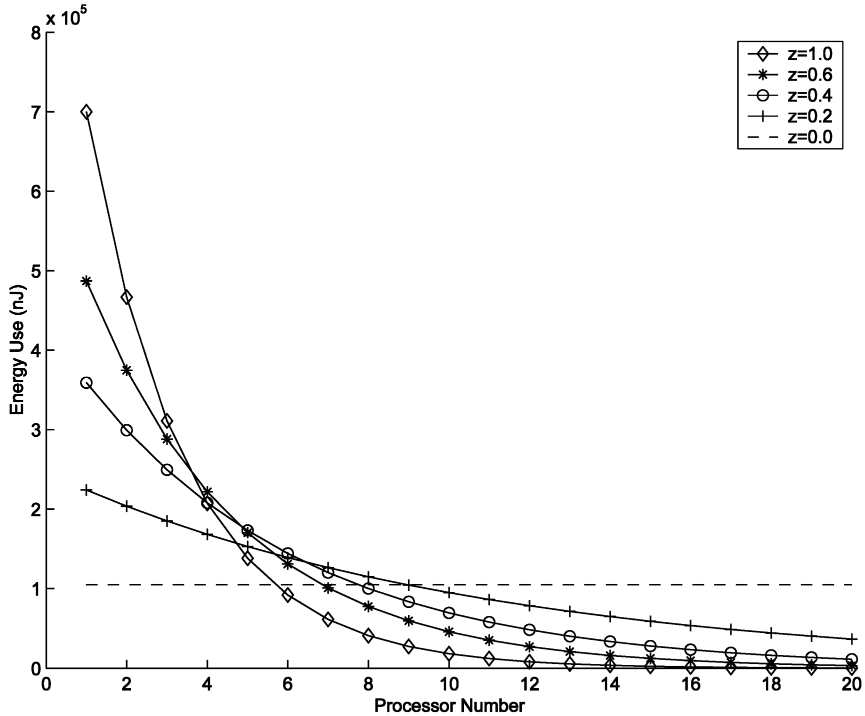


Fig. 7. Individual node energy use with variable inverse link speed z for single level tree network with controller and SMS^{2}R case.

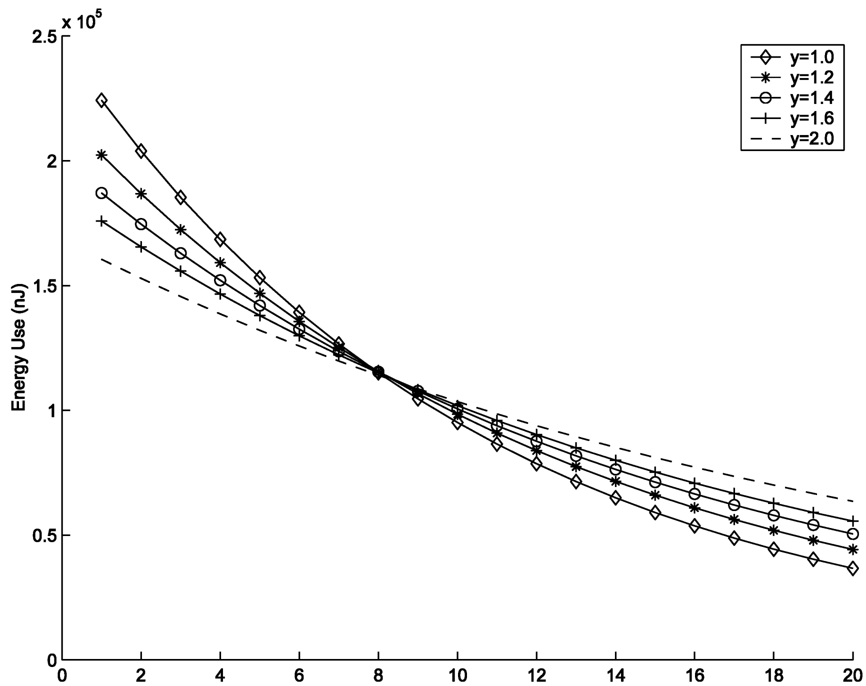


Fig. 8. Individual node energy use with variable inverse measuring speed y for single level tree network with controller and SMS^{2}R case.

In Fig. 9, the measurement/report time is plotted against the number of processors for the simultaneous measurement start simultaneous reporting case. The value the inverse link speed z is varied from 0 to 1 while the inverse measuring speed y is fixed to be 2. In this case the minimum finish time decreases as the number of processors in the network is increased. This assumes that the communication speed is fast enough to distribute the load to all the processors. We see here that time performance saturates beyond using a certain number of processors.

Fig. 10 on the other hand shows for the case when the inverse measuring speed y is varied from 1 to 2

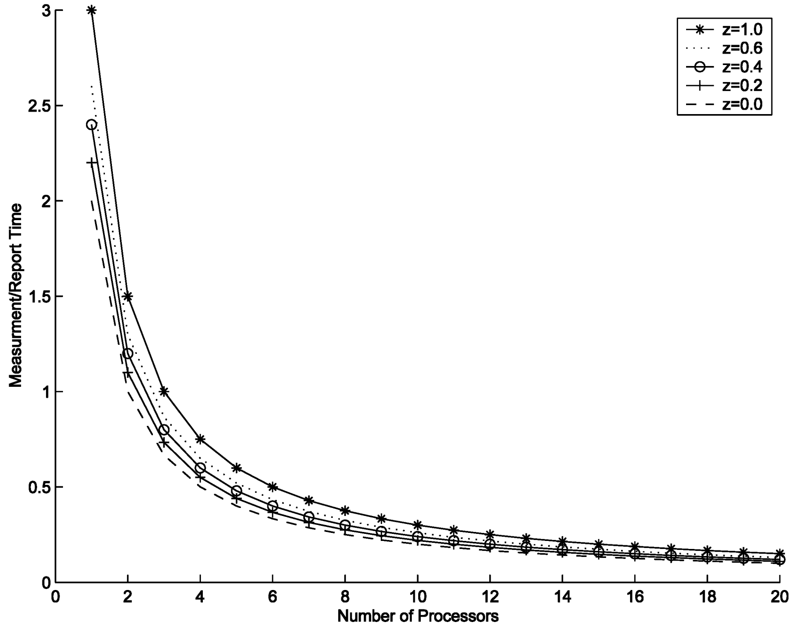


Fig. 9. Measurement/report time versus number of processors and variable inverse link speed z for single level tree network with controller and SMS^{2}RT case.

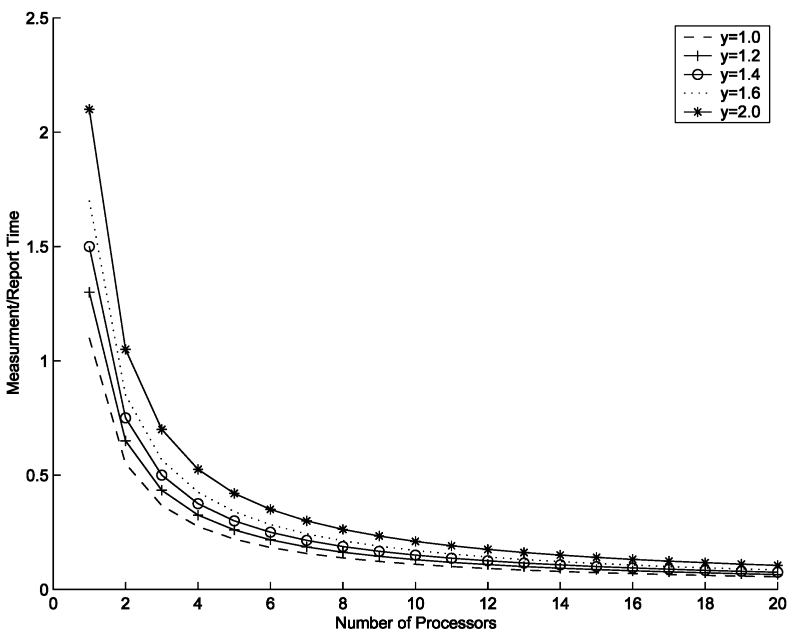


Fig. 10. Measurement/report time versus number of processors and variable inverse measuring speed y for single level tree network with controller and SMS^{2}RT case.

and the inverse link speed z is fixed to be 0.1. The monotonically decreasing behavior of [2] and [5] for only communication and computation is, again, repeated.

In terms of the energy use in the case of simultaneous reporting termination and a homogeneous network, one would expect that each processor's energy use is the same. This is because for a homogeneous network, the processors share the load equally ($\alpha_i = 1/N$).

C. CMR Strategy

This section presents the performance results obtained from the CMR strategy. As mentioned earlier, here it is assumed that the measurement time is about one order of magnitude smaller than the reporting time ($y_i T_{\text{ms}} < z_i T_{\text{cm}}$), in order to allow some time for the last measured data to be reported back to the controller.

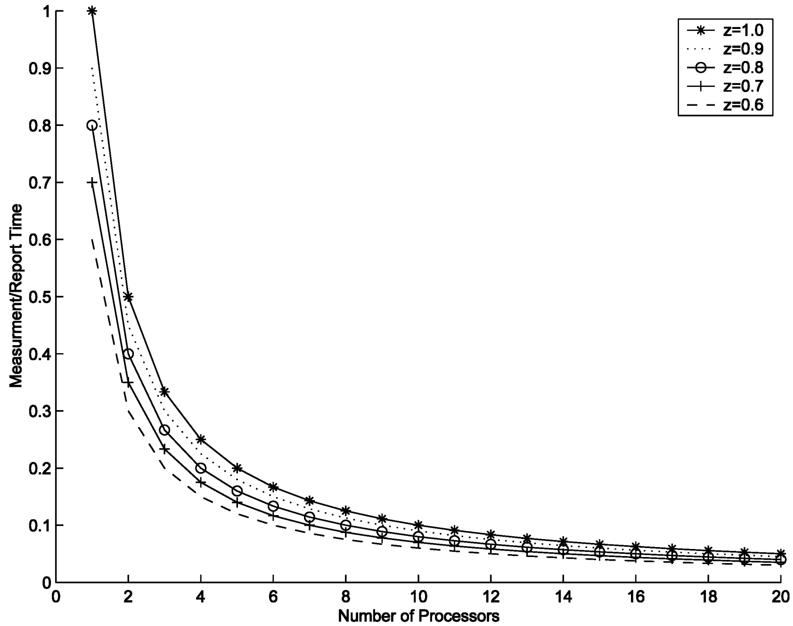


Fig. 11. Measurement/report time versus number of processors and variable inverse link speed z for single level tree network with controller and CMR case.

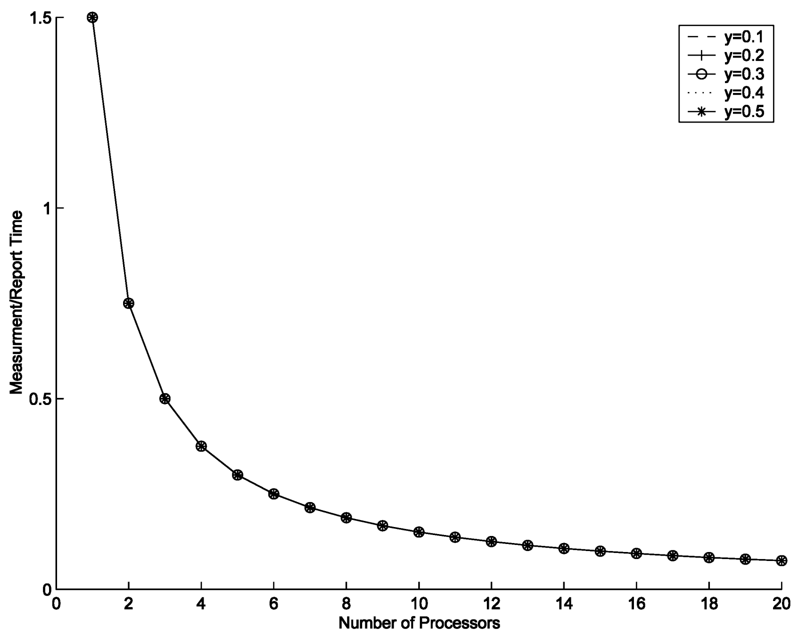


Fig. 12. Measurement/report time versus number of processors and variable inverse measuring speed y for single level tree network with controller and CMR case.

In Fig. 11, the measurement/report time is plotted against the number of processors when the value of the communication speed z is varied from 0.6 to 1 and the value of the measurement speed y is fixed to be 0.5. The typical saturation in time performance when adding processors is again illustrated.

Fig. 12 on the other hand shows for the case when the inverse measuring speed y is varied from 0.1 to 0.5 and the inverse link speed z is fixed to be 1.5. The result clearly shows that the minimum finish time is only dependent on the communication link speed for this specific strategy as the communication and measurement occur concurrently. Again, Figs. 11 and 12 demonstrate the monotonically decreasing behavior of [2] and [5].

The energy use for a homogeneous network using CMR is the same as for a homogeneous network using SMS^2RT. This is because for a homogeneous network, in both cases, the processors share the load equally ($\alpha_i = 1/N$).

Fig. 13 shows a comparison of the minimum finish time by the three measured data reporting

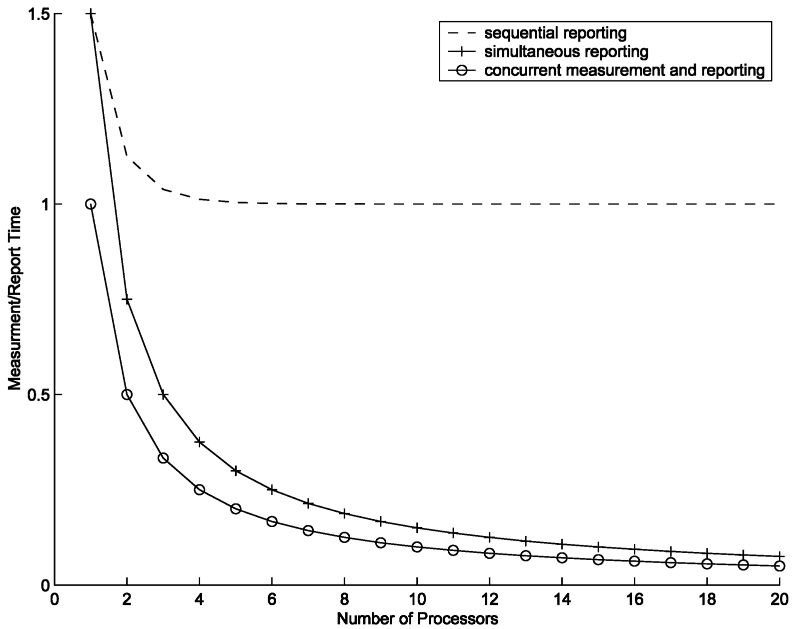


Fig. 13. Minimum finish time versus number of processors (comparison). In this case, $z = 1$ and $y = 0.5$.

strategies discussed earlier. As it can be seen from the plot, the CMR strategy will have a smaller finish time. This is due to the fact that in the case of the sequential reporting case, some of the processors in the network receive almost zero load, which effectively reduces the number of effective processors as compared with the concurrent reporting case where all processors receive a reasonable amount of load. The comparison is shown for the case where the value of the communication speed z is 1, the value of measurement speed y is 0.5, and T_{cm} and T_{cp} are set to be one.

VII. CONCLUSION

In this paper, closed-form solutions for optimum measurement and report time are obtained for single level tree networks with three types of representative data reporting strategies. Naturally, many scheduling variations are possible. Besides the finish time analysis, the energy use of the corresponding strategies is also examined. The performance of these strategies with respect to the timing and energy use and the effect of link and measurement speed is studied. The measurement and reporting time can be improved by increasing the number of processors in the network only to some extent before saturation in the sequential SMS^2R strategy. In terms of the energy use, unlike in the SMS^2R strategy, the energy use is equally distributed among all processors in the case of homogeneous networks using SMS^2RT and CMR strategies.

There is a growing body of work on integrating computation and communication. This paper is a step on integrating sensing/measurement with computation and communication. We note that with some extra complexity, substantial computing time could be included in the models discussed here. Future work will explore the intersection of sensing, computing and communication: the three key properties of wireless sensor networks.

REFERENCES

[1] Bharadwaj, V., Ghose, D., and Robertazzi, T. G.
Divisible load theory: A new paradigm for load scheduling in distributed systems.
Cluster Computing, **6** (2003), 7–18.

[2] Bharadwaj, V., Ghose, D., Mani, V., and Robertazzi, T. G.
Scheduling divisible loads in parallel and distributed systems.
Los Alamitos, CA: IEEE Computer Society Press, 1996.

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

[4] Blazewicz, J., and Drozdowski, M.
Distributed processing of distributed jobs with communication startup costs.
Discrete Applied Mathematics, **76** (1997), 21–41.

[5] Cheng, Y. C., and Robertazzi, T. G.
Distributed computation with communication delays.
IEEE Transactions on Aerospace and Electronic Systems, **22** (1988), 60–79.

[6] Sohn, J., and Robertazzi, T. G.
Optimal divisible load sharing for bus networks.
IEEE Transactions on Aerospace and Electronic Systems, **32** (1996), 34–40.

[7] Bataineh, S., and Robertazzi, T. G.
Bus oriented load sharing for a network of sensor driven processors.
IEEE Transactions on Systems, Man and Cybernetics, **21** (1991), 1202–1205.

[8] Cheng, Y. C., and Robertazzi, T. G.
Distributed computation for a tree network with communication delays.
IEEE Transactions on Aerospace and Electronic Systems, **26** (1990), 511–516.

[9] Yao, J., and Veeravalli, B.
Design and performance analysis of divisible load scheduling strategies on arbitrary graphs.
Cluster Computing, **7** (2004), 191–207.

[10] Robertazzi, T. G.
Processor equivalence for a linear daisy chain of load sharing processors.
IEEE Transactions on Aerospace and Electronic Systems, **29** (1993), 1216–1221.

[11] Blazewicz, J., Drozdowski, M., Guinand, F., and Trystram, D.
Scheduling a divisible task in a 2-dimensional mesh.
Discrete Applied Mathematics, (1999) , 35.

[12] Blazewicz, J., and Drozdowski, M.
Scheduling divisible jobs on hypercubes.
Parallel Computing, **21** (1996), 1945–1956.

[13] Sohn, J., and Robertazzi. T. G.
Optimal time-varying load sharing for divisible loads.
IEEE Transactions on Aerospace and Electronic Systems, **34** (1998), 907–923.

[14] Sohn, J., Robertazzi, T. G., and Luryi, S.
Optimizing computing costs using divisible load analysis.
IEEE Transactions on Parallel and Distributed Systems, **9** (1998), 225–234.

[15] Beaumont, O., Carter, L., Ferrante, J., Legrand, A., and Robert, Y.
Bandwidth-centric allocation of independent tasks on heterogeneous platforms.
In *Proceedings of International Parallel and Distributed Processing Symposium* (IPDPS'2002), IEEE Computer Society Press, 2002.

[16] Beaumont, O., Legrand, A., and Robert, Y.
Optimal algorithms for scheduling divisible workloads on heterogeneous systems.
Presented at the 12th Heterogeneous Computing Workshops (HCW'2003), 2003.

[17] Bharadwaj, V., Ghose, D., and Mani, V.
Multi-installment load distribution in tree networks with delays.
IEEE Transactions on Aerospace and Electronic Systems, **31** (1995), 555–567.

[18] Yang, Y., and Casanova, H.
UMR: A multi-round algorithm for scheduling divisible workloads.
In *Proceedings of the International Parallel and Distributed Processing Symposium* (IPDPS'03), Nice, France, 2003.

[19] Rosenberg, A. L.
Sharing partitionable workloads in heterogeneous NOWs: greedier is not better.
Cluster Computing, (2001), 124–131.

[20] Dutot, P. F.
Divisible load on heterogeneous linear array.
In *Proceedings of the International Parallel and Distributed Processing Symposium*, (IPDPS'03), Nice, France, 2003.

[21] Moges, M., and Robertazzi, T. G.
Optimal divisible load scheduling and Markov chain models.
In *Proceedings of the 2003 Conference on Information Sciences and Systems*, The Johns Hopkins University, Baltimore, MD, 2003.

[22] Tilak, S., Abu-Ghazaleh, N. B., and Heinzelman, W.
A taxonomy of wireless micro-sensor network models.
Mobile Computing and Communications Review, **9** (2002), 28–36.

[23] Intanagonwiwat, C., Estrin, D., Govindan, R., and Heidemann, J.
Impact of network density on data aggregation in wireless sensor networks.
In *Proceedings of the 22nd International Conference on Distributed Computing Systems* (ICDCS'02), 2002, 414–415.

[24] Shah, R. C., and Rabaey, J. M.
Energy aware routing for low energy ad-hoc sensor networks.
In *Proceedings of the 3rd IEEE Wireless Communications and Networking Conference*, Orlando, FL, 2001, 151–165.

[25] Schurgers, C., Tsiatsis, V., Ganeriwal, S., and Srivastava, M.
Topology management for sensor networks: Exploiting latency and density.
ACM MobiHOC'02, 2002, 135–145.

[26] Megeurdichian, S., Slijepcevic, S., Karayan, V., and Potkonjak, M.
Localized algorithms in wireless ad-hoc networks: Location discovery and sensor exposure.
MobiHOC, 2001, 106–116.

[27] Bordim, J. L., Nakano, K., and Shen, H.
Sorting on a single channel wireless sensor networks.
In *Proceedings of the International Symposium on Parallel Architectures and Networks*, 2002, 153–158.

[28] Iyengar, S. S., and Wu, Q.
Computational aspects of distributed sensor networks.
In *Proceedings of the International Symposium on Parallel Architectures and Networks*, 2002, 23–30.

[29] Hung, J., Kim, H. J., and Robertazzi, T. G.
Scalable scheduling in parallel processors.
In *Proceedings of the 2002 Conference on Information Sciences and Systems*, Princeton University, Princeton, NJ, 2002.

[30] Hung, J. T., and Robertazzi, T. G.
Scalable scheduling for clusters and grids using cut through switching.
International Journal of Computers and Applications, **26** (2004), 147–156.

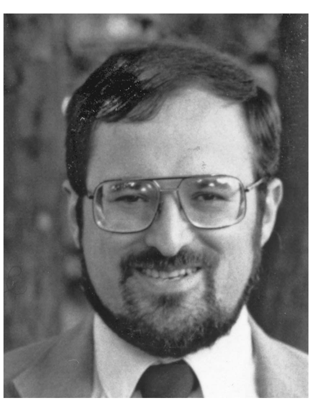
[31] Jones, C. E., Sivalingam, K. M., Agrawal, P., and Chen, J. C.
A survey of energy efficient network protocols for wireless networks.
Wireless Networks, **7** (2001), 343–358.

[32] Chang, J., and Tassiulas, L.
Energy conserving routing in wireless ad-hoc networks.
In *Proceedings of IEEE INFOCOM 2000*, 2000.

[33] Heinzelman, W., Chandrakasan, A., and Balakrishnan, H.
Energy efficient communication protocol for wireless micro sensor networks.
In *Proceedings of the 33rd Annual Hawaii International Conference on System Sciences*, 2000, 3005–3014.

Mequanint A. Moges (M'05) received his undergraduate degree from Addis Ababa University, Ethiopia, and master's degree from the University of New South Wales, Australia. He received the Ph.D. degree from the Department of Electrical and Computer Engineering at Stony Brook University in 2005. He is an assistant professor in the College of Technology at the University of Houston, Houston, TX, a position he has held since August 2005. His research interests include the design and optimization of wireless sensor networks, performance evaluation and optimization of computer and communication systems, and job scheduling in parallel and distributed systems and computational grids.

Dr. Moges is a recipient of a Presidential Fellowship from the State University of New York at Stony Brook, NY, and an AUSAID Scholarship from the University of New South Wales, Sydney, Australia.



Thomas G. Robertazzi (S'75—M'77—SM'91—F'05) received the Ph.D. from Princeton University, Princeton, NJ, in 1981 and the B.E.E. from the Cooper Union, New York, NY in 1977.

He is presently a professor in the Deptartment of Electrical and Computer Engineering at Stony Brook University, Stony Brook, NY. In supervising a very active research area, he has published extensively in the areas of parallel processor and grid scheduling, ad hoc radio networks, telecommunications network planning, ATM switching, queueing, and Petri networks.

Dr. Robertazzi authored, coauthored, or edited four books in the areas of performance evaluation, scheduling, and network planning.