Load Scheduling for Measurement and Data Reporting in Wireless Sensor Networks

Mequanint A. Moges
Department of Electrical and Computer Engr.
Stony Brook University
Stony Brook, NY 11794
e-mail: mmoges@ece.sunysb.edu

Thomas G. Robertazzi[1]
Department of Electrical and Computer Engr.
Stony Brook University
Stony Brook, NY 11794
e-mail: tom@ece.sunysb.edu

Abstract — **This paper introduces an optimal load allocation approach for measurement and data reporting in wireless sensor networks. Using divisible load theory as a starting point, results in terms of minimum finish time (make-span) are obtained for different measurement and reporting strategies. This work is novel as it introduces, for the first time, a new load scheduling strategy that considers the measurement capacity of processors and assumes negligible computation time which is radically different from the traditional divisible load scheduling research to date. Performance results in terms of finish time (make-span) for homogeneous measurement and reporting speeds are also presented.**

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 the 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 divisible load scheduling theory since the work of Cheng and Robertazzi [5] in 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 time.

In this paper, the processors are assumed to have a certain measurement capacity. The controller in the network assigns the amount of data to be measured by each processor and in turn these processors after finishing the measurement will report their measured results back to the control processor. There may be different strategies by which the controller communicates with the child processors and the child 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 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.

The organization of this paper is as follows. In section II, the system model and some notation used in this paper is discussed. Section III presents the analysis of the measurement and reporting time. Using the divisible load theory as a starting point, the analysis obtains set of recursive equations to find an optimum load distribution to processors. In section IV, a performance evaluation of the various strategies is presented. Finally the conclusion part appears in section V.

II. System Model

In this section, the various network parameters used in this paper 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 will be 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. Some of the 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 following section.

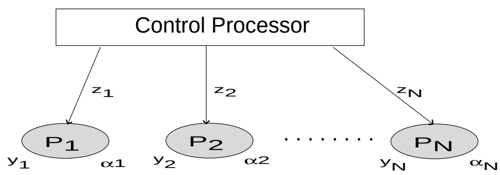

Figure 1: Single level tree network with controller.

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.

[1]This work was supported by NSF Grant CCR-99-12331.

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

T_{ms} : Measurement intensity constant. This is the time it takes the i^{th} processor to measure the entire load when $y_i = 1$. The entire load can be measured on the i^{th} processor in time $y_i T_{ms}$.

T_{cm} : Communication intensity constant. This is the time it takes to transmit all the processing load over a link when $z_i = 1$. The entire load can be transmitted over the i^{th} link in time $z_i T_{cm}$.

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, $i = 0, 1, ..., N$. 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).

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

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 originating at the control processor is assumed to be normalized to be a unit load.

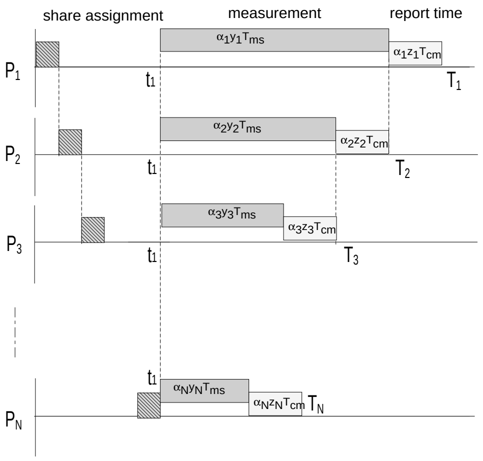

Figure 2: Timing diagram for a single level tree network with controller and sequential reporting time.

III. Measurement and Reporting Time

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. We assume only one processor may report back to the root processor at a time. (i.e, there is a single channel). 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 [6]. 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 of can be Fig. 2 written as follows:

$$T_1 = t_1 + \alpha_1 y_1 T_{ms} + \alpha_1 z_1 T_{cm} \quad (1)$$
$$T_2 = t_1 + \alpha_2 y_2 T_{ms} + \alpha_2 z_2 T_{cm} \quad (2)$$

$$T_N = t_1 + \alpha_N y_N T_{ms} + \alpha_N z_N T_{cm}. \quad (3)$$

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:

$$\alpha_1 + \alpha_2 + \alpha_3 + ... + \alpha_{N-1} + \alpha_N = 1 \quad (4)$$

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

$$\alpha_1 y_1 T_{ms} = \alpha_2 y_2 T_{ms} + \alpha_2 z_2 T_{cm} \quad (5)$$
$$\alpha_2 y_2 T_{ms} = \alpha_3 y_3 T_{ms} + \alpha_3 z_3 T_{cm} \quad (6)$$

$$\alpha_{N-2} y_{N-2} T_{ms} = \alpha_{N-1} y_{N-1} T_{ms} + \alpha_{N-1} z_{N-1} T_{cm} \quad (7)$$
$$\alpha_{N-1} y_{N-1} T_{ms} = \alpha_N y_N T_{ms} + \alpha_N z_N T_{cm} \quad (8)$$

Using the above sets of equations and the normalization equation, one can solve for α_1 for a homogeneous network as:

$$\alpha_1 (1 + s + s^2 + ... + s^{\text{N-2}} + s^{\text{N-1}}) = 1 \quad (9)$$

where $s = yT_{ms}/(yT_{ms} + zT_{cm})$, for the case where the communication link and measuring speed are assumed to be homogeneous.

The above equation can be simplified as:

$$\alpha_1 = (1 - s)/(1 - s^{\text{N}}) \quad (10)$$

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:

$$\alpha_i = \alpha_1 s^{\text{i-1}} \quad (11)$$

where i = 2, 3, 4, . . . , N.

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

$$T_f = t_1 + (yT_{ms} + zT_{cm})(1 - s)/(1 - s^{\text{N}}) \quad (12)$$

This measurement and reporting time of the network approaches $t_1 + zT_{cm}$ as N approaches ∞, which conforms to the result shown in [6,7]. 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:

$$1 - s = zT_{cm}/(yT_{ms} + zT_{cm}) \tag{13}$$

Then substituting this result back in T_f gives:

$$T_f = t_1 + zT_{cm} \tag{14}$$

Intuitively, the measurement latency is "hidden" by the reporting latency.

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 child node 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 [8]. 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.

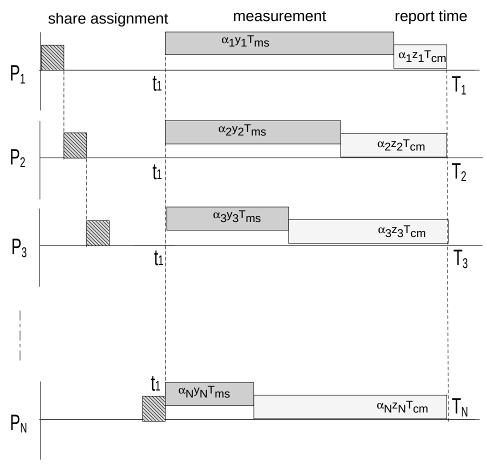

Figure 3: Timing diagram for a single level tree network with controller and simultaneous reporting termination.

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:

$$T_1 = t_1 + \alpha_1 y_1 T_{ms} + \alpha_1 z_1 T_{cm} \tag{15}$$

$$T_2 = t_1 + \alpha_2 y_2 T_{ms} + \alpha_2 z_2 T_{cm} \tag{16}$$

$$T_N = t_1 + \alpha_N y_N T_{ms} + \alpha_N z_N T_{cm}. \tag{17}$$

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

$$\alpha_1 + \alpha_2 + \alpha_3 + ... + \alpha_{N-1} + \alpha_N = 1 \tag{18}$$

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

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

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

$$\alpha_1 r_1 = \alpha_2 r_2 \tag{19}$$

$$\alpha_2 r_2 = \alpha_3 r_3 \tag{20}$$

$$\alpha_{N-2} r_{N-2} = \alpha_{N-1} r_{N-1} \tag{21}$$

$$\alpha_{N-1} r_{N-1} = \alpha_N r_N \tag{22}$$

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

$$\alpha_i = (1/r_i)/\sum_{i=1}^{N}(1/r_i) \tag{23}$$

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 more share than processors with slower combined measurement and link speeds. The minimum measurement and reporting time of the network will then be given as:

$$T_f = T_i = t_1 + (y_i T_{ms} + z_i T_{cm})(1/r_i)/\sum_{i=1}^{N}(1/r_i) \tag{24}$$

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 ... , N.

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

$$T_f = t_1 + (yT_{ms} + zT_{cm})/N \tag{25}$$

C Concurrent Measurement and Reporting Case

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 co-processor 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. The timing diagram of the network is shown in Fig. 4. Note this model is analogous to an equivalent single level tree model involving only computation and communication. In this time non-reversed 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

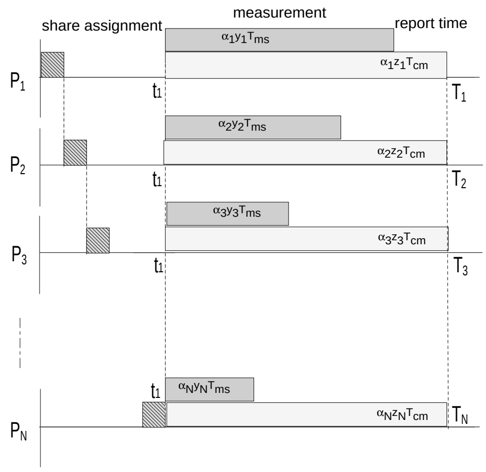

Figure 4: Timing diagram for a single level tree network with controller and concurrent measurement and reporting time.

and communication in the model of this section is equivalent to computation in the standard model [9].

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

$$T_1 = t_1 + \alpha_1 z_1 T_{cm} \quad (26)$$
$$T_2 = t_1 + \alpha_2 z_2 T_{cm} \quad (27)$$

$$T_N = t_1 + \alpha_N z_N T_{cm}. \quad (28)$$

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

$$\alpha_1 + \alpha_2 + \alpha_3 + ... + \alpha_{N-1} + \alpha_N = 1 \quad (29)$$

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

$$T_1 = T_2 = T_3 = . . . = T_N$$

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

$$\alpha_1 z_1 T_{cm} = \alpha_2 z_2 T_{cm} \quad (30)$$
$$\alpha_2 z_2 T_{cm} = \alpha_3 z_3 T_{cm} \quad (31)$$

$$\alpha_{N-2} z_{N-2} T_{cm} = \alpha_{N-1} z_{N-1} T_{cm} \quad (32)$$
$$\alpha_{N-1} z_{N-1} T_{cm} = \alpha_N z_N T_{cm} \quad (33)$$

As in the previous case, using the above set of equations, one can now write $\alpha' s$ as a function of z_i as:

$$\alpha_i = (1/z_i)/\sum_{i=1}^{N}(1/z_i) \quad (34)$$

From the above expression, one can see that that the share of 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 more share than processors with slower link speeds. The minimum measurement and reporting time of the network will then be given as:

$$T_f = T_i = t_1 + (z_i T_{cm})(1/z_i)/\sum_{i=1}^{N}(1/z_i) \quad (35)$$

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:

$$T_f = t_1 + zT_{cm}/N \quad (36)$$

IV. Performance Evaluation

The minimum measurement and reporting time expressions obtained in the previous sections will be 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 of the network. To do so, we consider the following two cases: In the first case the measurement and reporting time use is plotted against number of processors when z is varied and measurement speed y is fixed. In the second case, the measurement and reporting time use is plotted against number of processors when z is fixed and measurement speed y is varied.

A Simultaneous Measurement Start, Sequential Reporting Case

In Fig. 5, the measurement/report time is plotted against the number of processors 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 $T_{cm} = 1$ and $T_{cp} = 1$. It can be shown from the figure that the larger the communication speed, the longer 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 processing time approaches zT_{cm}, which in this case is 0.1, as N approaches ∞.

B Simultaneous Measurement Start, Simultaneous Reporting Termination

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

In Fig. 7, the measurement/report time is plotted against the number of processors for the simultaneous measurement start same 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

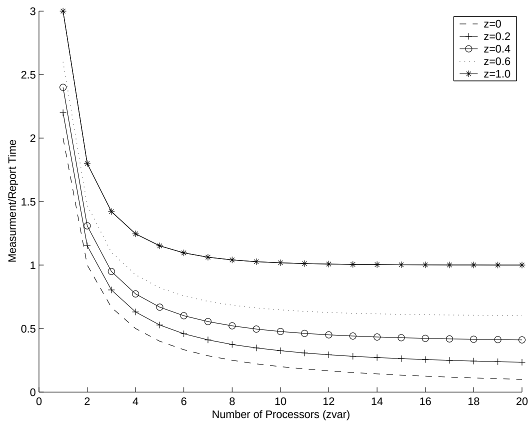

Figure 5: Measurement/report time versus number of processors and variable inverse link speed z for single level tree network with controller and sequential reporting time.

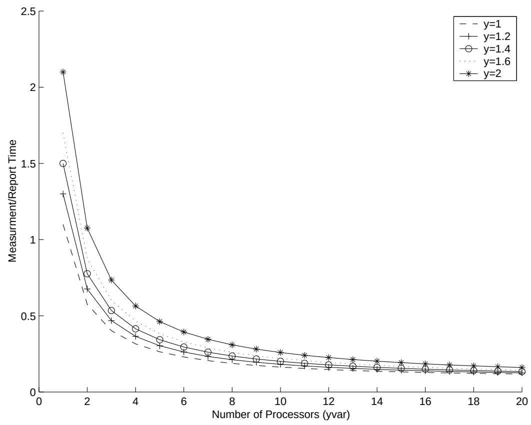

Figure 6: Measurement/report time versus number of processors and variable inverse measuring speed y for single level tree network with controller and sequential reporting time.

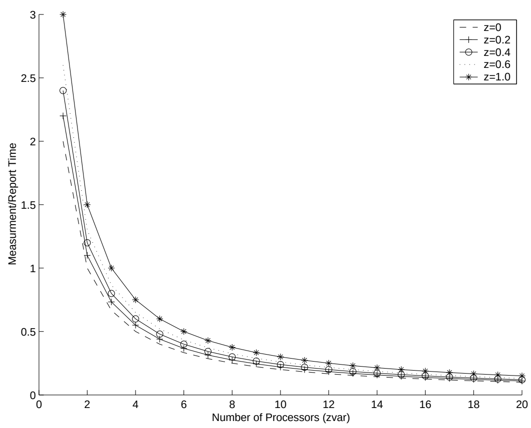

Figure 7: Measurement/report time versus number of processors and variable inverse link speed z for single level tree network with controller and simultaneous measurement start, simultaneous reporting termination case.

assumes the communication speed is fast enough to distribute the load to all the processors.

Fig. 8 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.

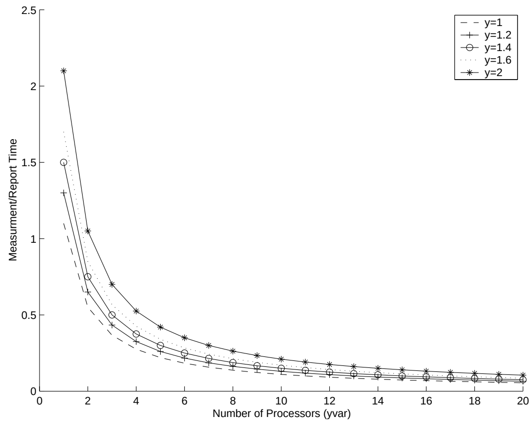

Figure 8: Measurement/report time versus number of processors and variable inverse measuring speed y for single level tree network with controller and simultaneous measurement start, simultaneous reporting termination case.

C Concurrent Measurement and Reporting Case

This section presents the performance results obtained from the concurrent measurement and reporting strategy. In this case it is assumed that the measurement time is one order of magnitude smaller than the reporting time ($y_i T_{ms} < z_i T_{cm}$), in order to allow some time for the last measured data to be reported back to the controller.

In Fig. 9, 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 measurement speed y is fixed to be 0.5.

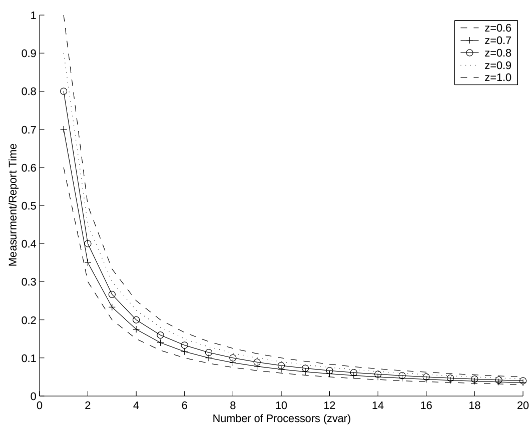

Figure 9: Measurement/report time versus number of processors and variable inverse link speed z for single level tree network with controller and concurrent measurement and reporting case.

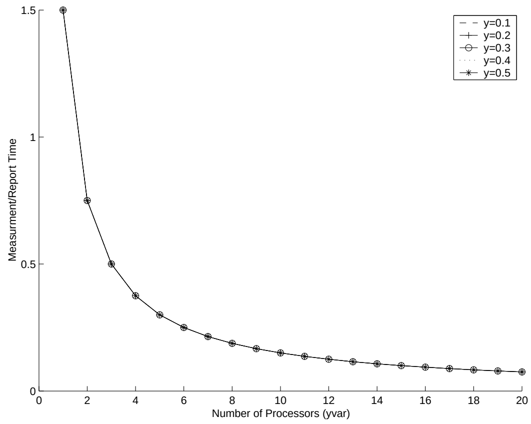

Figure 10: Measurement/report time versus number of processors and variable inverse measuring speed y for single level tree network with controller and concurrent measurement and reporting case.

Fig. 10 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.

Fig. 11 shows a comparison of the minimum finish time by the three measured data reporting strategies discussed earlier. As it can be seen from the plot, the concurrent measurement and reporting strategy will have less finish time. This is due to the fact that in the case of sequential reporting case, some of the processors in the network receive almost zero load, which effectively reduces the number of effective processors as compared to 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.

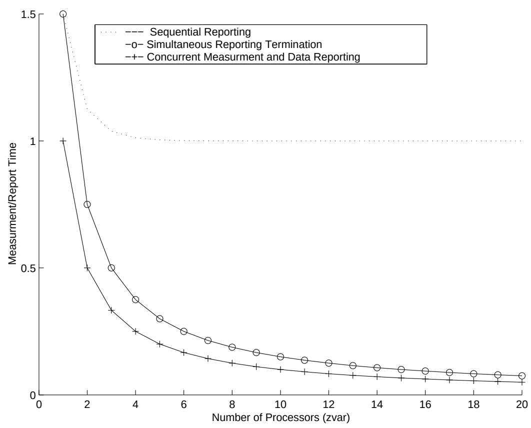

Figure 11: Energy use versus number of processors (comparison). In this case z = 1 and y = 0.5.

V. Conclusion

In this paper, closed form solutions for optimum measurement and report time are obtained for single level tree networks with three types of data reporting strategies. The performance of these strategies with respect to the optimum finish time 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.

Currently we are working on the energy use of the three strategies presented in this paper. Future research works may involve, finding optimum scheduling that accounts for both the finishing time and energy use for other network topologies that are used in divisible load theory such as linear daisy chain networks, multi level tree networks, and more complex networks including hypercubes, two dimensional meshes as well as in wireless ad hoc network applications.

References

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

[2] V. Bharadwaj, D. Ghose, V. Mani, and T.G. Robertazzi, Scheduling Divisible Loads in Parallel and Distributed Systems, IEEE Computer Society Press, Los Alamitos, CA, 1996.

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

[4] J. Blazewicz and M. Drozdowski, Distributed Processing of Distributed Jobs with Communication Startup Costs, Discrete Applied Mathematics 76 (1997) 21-41.

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

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

[7] S. Bataineh and T.G. Robertazzi, Distributed computation for a bus networks with communication delays, in Proc. 1991 Conf. Inform. Sci. Syst., The Johns Hopkins Univ., Baltimore MD (1991) 709-714.

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

[9] J.T. Hung and T.G. Robertazzi, Scalable scheduling for clusters and grids using cut through switching, Accepted for International Journal of Computers and Applications, 2004.