

Efficient Scheduling for Sensing and Data Reporting in Wireless Sensor Networks

Carlos Fernando Gamboa and Thomas G. Robertazzi *

Department of Electrical and Computer Engineering

Stony Brook University

Stony Brook, NY 11794

Email: gcarlos@ece.sunysb.edu and tom@ece.sunysb.edu*

Abstract— This paper considers an innovative scheduling strategy in which a control processor assigns a load share to be measured by each of N processors organized in a single level tree (star) wireless sensors network. Here processors begin to sense as soon as receiving their own load share assignment rather than waiting for all processors to receive their assignments as done in previous research works. This strategy has the potential to reduce solution time (make span) significantly. We find an analytical expression for optimal load assignment and finish time which is simple to compute and can be implemented in real time.

I. INTRODUCTION

Wireless sensor networks have been increasingly studied and developed during the last decade due to their potential application fields such as security, geothermal monitoring, traffic control and health care [1], [2]. Among the challenges that this technology faces are the communication constraints (limited bandwidth and transmission energy) which could be the most crucial aspects to be solved for this technology during the next few years. An interesting approach to overcome this critical aspect is to process the measured data and transmit a summarized version of the data measured when the sensor devices and its architecture allows it. Doing so would reduce the amount of data to be transmitted and consequently the energy used for transmission of the reporting data [3].

Another interesting approach was proposed in 2004 [4]. It introduced a novel scheduling strategy which considered single level tree (star) network of N processors and a control processor. In this work, each processor starts to sense when the control processor finishes distributing the entire load share assignments to all of the processors in the network. Here load share assignment means the amount of sensing load that is assigned by the control processor to each of the processors in the network.

As in [4] this paper is done in the context of Divisible Load scheduling Theory (DLT) that has been the focus of attention by researchers [5]–[8] studying data load distribution in parallel and distributed systems since 1988 [9]. Initially most of the jobs using DLT considered communication and computation as the main parameters of the system to find a optimal divisible (partitionable) load to be processed and transmitted by each processor and link in the network in a minimal amount of time. A partitionable data load is one that can be arbitrarily distributed among the processor in the network and there is no precedence relations between data.

The network architecture considered in this paper includes a control processor that distributes the load share assignment to the other N processors in the network. Thus the processors begin to sense as soon as receiving their own load share assignment rather than waiting for all processors to receive their assignments and after sensing their correspondent fraction of load share each processor returns the result to the control processor. This scheduling protocol is implemented for homogeneous and heterogeneous networks configurations with processors that have the capability to sense and processors that have the potential to sense and compute.

This paper is organized as follows: In section II, network model and parameters used in this study are presented. In section III, mathematical model and reporting time expressions for optimal allocation of load using divisible load theory are discussed and presented. In section IV, performance and evaluation results for the strategies presented in the study are showed and analyzed. Finally, the conclusions for this study are presented in section V.

II. THE NETWORK MODEL AND PARAMETERS

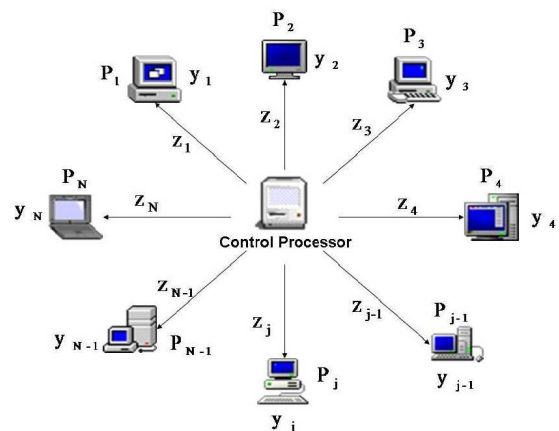


Fig. 1. Single level tree (star) network with control processor

Consider a single level tree (star) network consisting of N processors and $N-1$ links as shown in Fig.(1). There is a control processor who distributes load share assignments to the other processors in the network sequentially. As soon as each processor receives its own load share assignment it

starts to sense. The results are reported back to the control processor sequentially. In some cases, when the network topology allows it, the processor could compute measured data load before transmitting it back to the control processor.

A. Notations and Definitions

t Is the time that the control processor takes to assign the measurement instruction to each child processor.

α_j The load share fraction assigned by the control processor to the j^{th} link-processor pair to be measured.

y_j : A constant that is inversely proportional to the measuring speed of the processor j^{th} by the control processor.

w_j : The inverse of the computing speed of the j^{th} processor.

z_j : The inverse of the link speed of the j^{th} link.

T_{ms} : Measuring intensity constant: the entire load is processed in $y_j T_{ms}$ seconds by the j^{th} processor.

T_{cp} : Computing intensity constant: the entire load is processed in $w_j T_{cp}$ seconds by the j^{th} processor.

T_{cm} : Communication intensity constant: the entire load can be transmitted in $z_j T_{cm}$ seconds over the j^{th} link.

T_j : Is the total time measured from the beginning of the scheduling process up to the end of the transmission of the data measured by the j^{th} processor.

T_f : Is the time when the last processor finishes reporting.
 $T_f = \max(T_1, T_2, \dots, T_N)$

It is assumed that the fractions of load measured are normalized and their addition should sum 1 in the control processor (1).

$$1 = \sum_{j=1}^N \alpha_j \quad (1)$$

III. MATHEMATICAL MODEL AND REPORTING TIME

A. Sequential reporting time

The control processor distributes load share assignment to each processor sequentially Fig.(2). Each processor starts to measure the data as soon as it receives its load assignment but for this configuration the processor can only reports its result back sequentially. There is only one channel available for transmission. The mathematical expressions that describe this protocol in terms of the amount of finish time are:

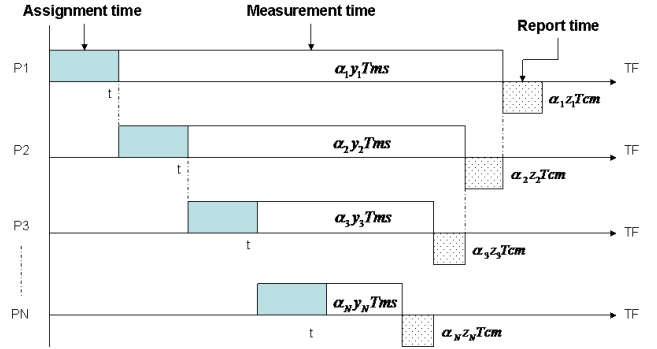


Fig. 2. Timing diagram for a single level tree network with controller and sequential reporting

$$T_1 = t + \alpha_1 y_1 T_{ms} + \alpha_1 z_1 T_{cm} \quad (2)$$

$$T_2 = 2t + \alpha_2 y_2 T_{ms} + \alpha_2 z_2 T_{cm} \quad (3)$$

$$T_3 = 3t + \alpha_3 y_3 T_{ms} + \alpha_3 z_3 T_{cm} \quad (4)$$

$$T_N = Nt + \alpha_N y_N T_{ms} + \alpha_N z_N T_{cm} \quad (5)$$

It can be seen in Fig.(2) that the time required to measure a load share assignment ($\alpha_j y_j T_{ms}$) by the j processor will be equal to the time used by the $j + 1$ processor to wait for the assignment (t) and measure ($\alpha_{j+1} y_{j+1} T_{ms}$) and report ($\alpha_{j+1} z_{j+1} T_{cm}$) its result back to the control processor. Thus,

$$\alpha_1 y_1 T_{ms} = t + \alpha_2 y_2 T_{ms} + \alpha_2 z_2 T_{cm} \quad (6)$$

$$\alpha_2 y_2 T_{ms} = t + \alpha_3 y_3 T_{ms} + \alpha_3 z_3 T_{cm} \quad (7)$$

$$\alpha_{N-3} y_{N-3} T_{ms} = t + \alpha_{N-2} (y_{N-2} T_{ms} + z_{N-2} T_{cm}) \quad (8)$$

$$\alpha_{N-2} y_{N-2} T_{ms} = t + \alpha_{N-1} (y_{N-1} T_{ms} + z_{N-1} T_{cm}) \quad (9)$$

$$\alpha_{N-1} y_{N-1} T_{ms} = t + \alpha_N (y_N T_{ms} + z_N T_{cm}) \quad (10)$$

The previous equation system can be expressed in terms of f_j and s_j as,

$$\alpha_1 = f_1 + \alpha_2 s(1) \quad (11)$$

$$\alpha_2 = f_2 + \alpha_3 s(2) \quad (12)$$

$$\alpha_{N-2} = f_{N-2} + \alpha_{N-1} s(N-2) \quad (13)$$

$$\alpha_{N-1} = f_{N-1} + \alpha_N s(N-1) \quad (14)$$

where

$$f_j = t / y_j T_{ms} \quad (15)$$

$$s_j = (y_{j+1} T_{ms} + z_{j+1} T_{cm}) / y_j T_{ms} \quad (16)$$

The equation system previously presented consists of N-1 recursive equations. Substituting recursively equation (14) into equation (13) and so on for j processor in term of α_N we have

$$\alpha_j = f_j + \sum_{m=1}^{N-j-1} f_{j+m} \prod_{t=0}^{m-1} s_{j+t} + \alpha_N \prod_{l=0}^{N-j-1} s_{j+l} \quad (17)$$

where $j=1,2,3,\dots,N-2$.

As mentioned before the normalization equation is the expression that states that the total amount of load is originated in the control processor and has to be 1. In order to have a closed solution for different load share assignments the expression (17) is used and evaluated for $j=1,2,3,\dots,N-2$. As a consequence,

$$\alpha_1 = f_1 + \sum_{m=1}^{N-2} f_{1+m} \prod_{t=0}^{m-1} s_{1+t} + \alpha_N \prod_{l=0}^{N-2} s_{1+l} \quad (18)$$

$$\alpha_2 = f_2 + \sum_{m=1}^{N-3} f_{2+m} \prod_{t=0}^{m-1} s_{2+t} + \alpha_N \prod_{l=0}^{N-3} s_{2+l} \quad (19)$$

$$\alpha_3 = f_3 + \sum_{m=1}^{N-4} f_{3+m} \prod_{t=0}^{m-1} s_{3+t} + \alpha_N \prod_{l=0}^{N-4} s_{3+l} \quad (20)$$

$$\alpha_{N-2} = f_{N-2} + \sum_{m=1}^1 f_{N-2+m} \prod_{t=0}^{m-1} s_{N-2+t} + \alpha_N \prod_{l=0}^1 s_{N-2+l} \quad (21)$$

And for $j = N - 1$ the equation is

$$\alpha_{N-1} = f_{N-1} + \alpha_N s_{N-1} \quad (22)$$

Substituting the expressions for α from equations (18-22) into the normalization equation (1) α_N can be found as,

$$\alpha_N = \frac{1 - \left[f_{N-1} + \sum_{j=1}^{N-2} f_j + \sum_{m=1}^{N-j-1} f_{j+m} \prod_{t=0}^{m-1} s_{j+t} \right]}{1 + s_{N-1} + \sum_{j=1}^{N-2} \prod_{l=0}^{N-j-1} s_{j+l}} \quad (23)$$

Consequently we obtain an expression for α_N (23) as a function of the network parameters which allows us to calculate the load share assignments α_{N-1} using equation (22) and for $\alpha_1, \dots, \alpha_{N-2}$ using equation (17).

The minimum finish time for this network configuration using this protocol can be found using equations (17) for $j=1$, (23), and substituting them into (2).

$$T_N = t + [y_N T_{ms} + z_N T_{cm}] \alpha_1 \quad (24)$$

B. Simultaneous reporting finish time

In Fig.(3) a scheduling protocol for a single level tree sensor network is considered. In this case, there is more than one channel for transmission of the data previously measured by the sensors. Consequently each sensor has an assigned channel which will be exclusively used to transmit its data measured to the control processor. Besides the multi channel availability, in this set up it is required that all processors finish at the same time. The total finish time would be described by:

$$T_1 = t + \alpha_1 y_1 T_{ms} + \alpha_1 z_1 T_{cm} \quad (25)$$

$$T_2 = 2t + \alpha_2 y_2 T_{ms} + \alpha_2 z_2 T_{cm} \quad (26)$$

$$T_3 = 3t + \alpha_3 y_3 T_{ms} + \alpha_3 z_3 T_{cm} \quad (27)$$

$$T_N = Nt + \alpha_N y_N T_{ms} + \alpha_N z_N T_{cm} \quad (28)$$

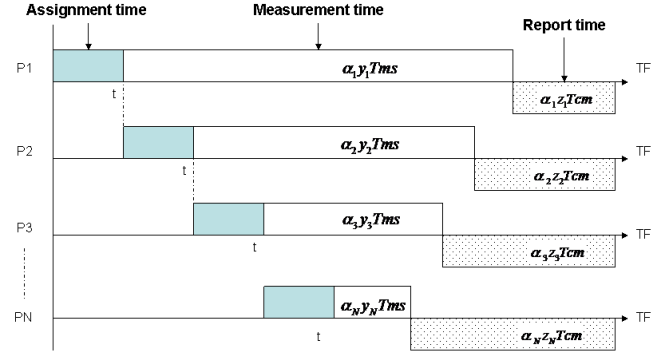


Fig. 3. Timing diagram for a single level tree network with controller and same reporting time finalization

As mentioned before the finish time would be the same for each sensor. Thus,

$$T_1 = T_2 = T_3 = \dots = T_N \quad (29)$$

Using the equation in terms of the network parameters the system would look like:

$$\alpha_1 (y_1 T_{ms} + z_1 T_{cm}) = t + \alpha_2 (y_2 T_{ms} + z_2 T_{cm}) \quad (30)$$

$$\alpha_2 (y_2 T_{ms} + z_2 T_{cm}) = t + \alpha_3 (y_3 T_{ms} + z_3 T_{cm}) \quad (31)$$

$$\alpha_{N-1} = \frac{t + \alpha_N (y_N T_{ms} + z_N T_{cm})}{(y_{N-1} T_{ms} + w_{N-1} T_{cp})} \quad (32)$$

Another way to express the last equation is:

$$\alpha_1 = h_1 + \alpha_2 g(1) \quad (33)$$

$$\alpha_2 = h_2 + \alpha_3 g(2) \quad (34)$$

$$\alpha_{N-1} = h_{N-1} + \alpha_N g(N-1) \quad (35)$$

Here

$$h_j = \frac{t}{(y_j T_{ms} + z_j T_{cm})} \quad (36)$$

$$g_j = \frac{(y_{j+1} T_{ms} + z_{j+1} T_{cm})}{(y_j T_{ms} + z_j T_{cm})} \quad (37)$$

Notice that the last set of equations (33-35) are the same obtained for the sequential reporting time protocol equations (11-14). Using the normalization equation (1) and following the same procedure as described for the sequential reporting time protocol we obtain the same expression for distribution of load to be measured for α_N (19) but instead of using f_j and s_j , the new equation would be in terms of g_j and h_j .

C. Load pre-processing, sequential reporting

As is widely known the energy used to transmit data in wireless sensors networks is higher than the amount of energy used to compute it [3] so processing the data load measured before transmitting it is an option that has to be studied. In the

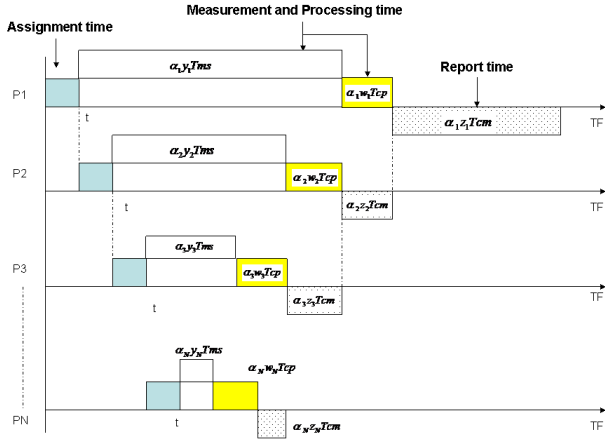


Fig. 4. Timing diagram for a single level tree network with controller and same sequencing reporting with pre-processing

following schedule protocol the same network topology used in the previous section is considered but in addition to the measuring capacity the sensors have a processing capability too. Fig.(4) shows how the control processor assigns load to be measured to the sensors in the network. As soon as the sensor receives its job starts to measure. The processing of the load measured starts when the sensor has finished to measure the whole fraction of load assigned. The results will be reported by each sensor sequentially.

For this protocol the expression that describes the finish time are:

$$T_1 = t + \alpha_1 y_1 T_{ms} + \alpha_1 w_1 T_{cp} + \alpha_1 z_1 T_{cm} \quad (38)$$

$$T_2 = 2t + \alpha_2 y_2 T_{ms} + \alpha_2 w_2 T_{cp} + \alpha_2 z_2 T_{cm} \quad (39)$$

$$T_3 = 3t + \alpha_3 y_3 T_{ms} + \alpha_3 w_3 T_{cp} + \alpha_3 z_3 T_{cm} \quad (40)$$

$$T_N = Nt + \alpha_N y_N T_{ms} + \alpha_N w_N T_{cp} + \alpha_N z_N T_{cm} \quad (41)$$

Following the same procedure from the first protocol studied we express the fraction of load in terms of the network parameters using the following expressions:

$$\alpha_1(y_1 T_{ms} + w_1 T_{cp}) = t + \alpha_2(y_2 T_{ms} + w_2 T_{cp} + z_2 T_{cm}) \quad (42)$$

$$\alpha_2(y_2 T_{ms} + w_2 T_{cp}) = t + \alpha_3(y_3 T_{ms} + w_3 T_{cp} + z_3 T_{cm}) \quad (43)$$

$$\alpha_{N-1} = \frac{t + \alpha_N(y_N T_{ms} + w_N T_{cp} + z_N T_{cm})}{(y_{N-1} T_{ms} + w_{N-1} T_{cp})} \quad (44)$$

Using the normalization equation and solving the recursive equations for α_N it is obtained the following expression,

$$\alpha_N = \frac{1 - \left[\alpha_{N-1} + \sum_{j=1}^{N-2} o_j + \sum_{m=1}^{N-j-1} o_{j+m} \prod_{t=0}^{m-1} p_{j+t} \right]}{1 + p_{N-1} + \sum_{j=1}^{N-2} \prod_{l=0}^{N-j-1} p_{j+l}} \quad (45)$$

where

$$o_j = \frac{t}{(y_j T_{ms} + w_j T_{cp})} \quad (46)$$

$$p_j = \frac{(y_{j+1} T_{ms} + w_{j+1} T_{cp} + z_{j+1} T_{cm})}{(y_j T_{ms} + w_j T_{cp})} \quad (47)$$

As in the last sections the mathematical expression obtained for α_N in term of the variables o_j and p_j will be the same as the expression obtained in terms of f_j and s_j form the first scheduling protocol. As a consequence what makes the expressions different from each other are the parameters that defined the pair (f,s), (g,h) and (o,p) for each protocol respectively.

IV. PERFORMANCE AND EVALUATION RESULTS

In order to investigate the relationship between the number of processors and the communication time in the network the expressions for minimum finish time, measuring time and reporting time were used. For each of the protocols previously presented the finish time as a function of the number of processors in the network for a fixed value of y and changing values of z and for fixed values of z and changing values of y was simulated and plotted.

A. Sequential reporting distribution

Fig.(5) shows the finish time versus number of processors variable inverse link speed z . The inverse link speed z is evaluated between 50 and 90. The y inverse measuring speed is fixed to 190. In all the simulation the values for T_{cm} , T_{cp} and T_{ms} are equal to one. It can be seen that the amount of processors needed to reached the minimum finish time will depend on z .

On the other hand in Fig.(6) the inverse link speed z is fixed to 80 and the inverse measuring speed is variated between 100 to 160. Here it can be seen that increasing the inverse measuring speed will lead us to find a better finish time for this particular example.

B. Simultaneous reporting distribution

The simulations presented in this section are shown in Fig.(7) where the inverse link speed z is evaluated between 50 and 90 and y is 190. In addition in Fig.(8) results for this schedule protocol with the inverse link speed fixed to 80 and inverse measuring speed varying between y 100 and 160, are shown.

C. Load pre-processing, sequential reporting

As mentioned before, in order to reduce the amount of energy used to report the data back to the control processor a load pre-processing can be done. In this case we include in the model the T_{cp} constant which will be fixed to 1. The inverse link speed varies from 50 and 90, inverse processor speed is fixed to be 100 and the inverse measuring speed is 190. Fig.(9) shows that for this particular example the finish time increases in comparison to the first scheduling protocol. In the case of the variation of the inverse measuring speed against the number of processor Fig.(10) the results showed for this particular experiment that introducing a pre-processing increases the finish time.

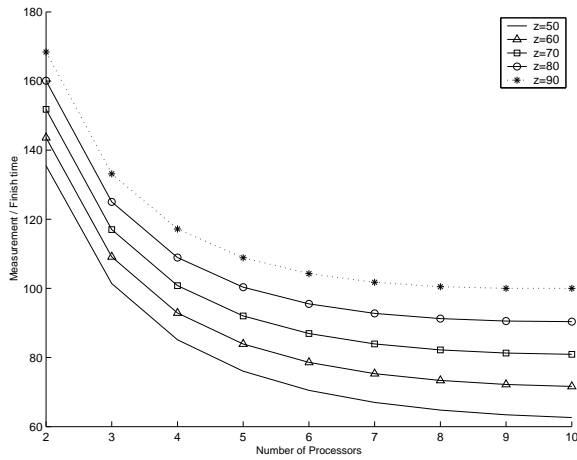


Fig. 5. Finish time versus number of processors, variable inverse link speed z and fixed load inverse measuring speed y in a single level tree network with control processor and a sequential reporting time.

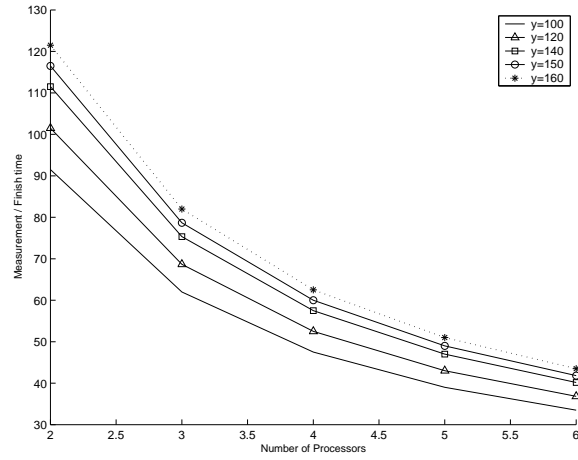


Fig. 8. Finish time versus number of processors, variable inverse measuring speed y and fixed inverse sensor speed z in a single level tree network with control processor and a simultaneous reporting time.

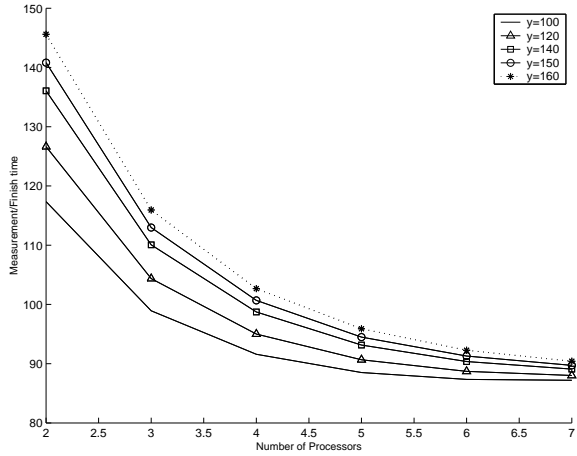


Fig. 6. Finish time versus number of processors, variable inverse measuring speed y and fixed inverse link speed z in a single level tree network with control processor and a sequential reporting time.

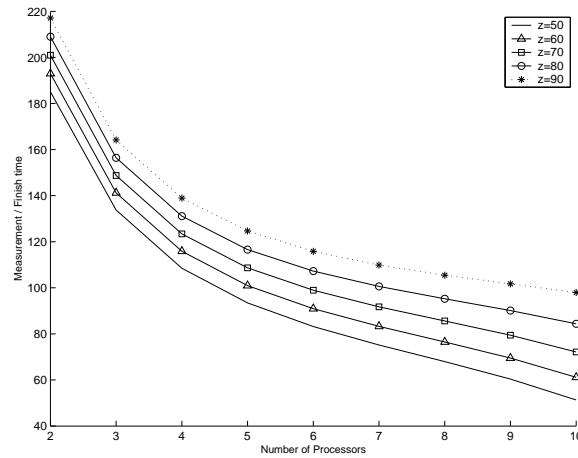


Fig. 9. Finish time versus number of processors, variable inverse link speed z , fixed inverse measuring speed y and inverse processor speed w in a single level tree network with control processor sequential reporting with pre-processing.

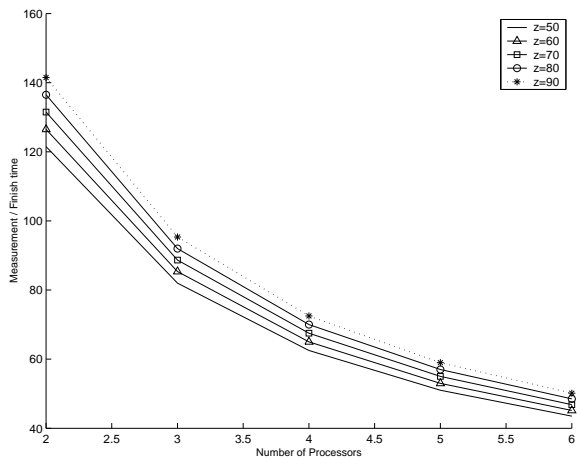


Fig. 7. Finish time versus number of processors, variable inverse link speed z and fixed inverse measuring speed y in a single level tree network with control processor and a simultaneous reporting time.

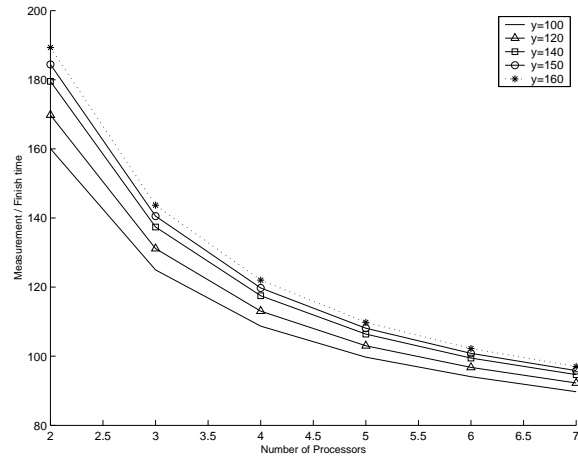


Fig. 10. Finish time versus number of processors, fixed inverse measuring speed y , variable inverse link speed z and inverse processor speed w in a single level tree network with control processor sequential reporting with pre-processing.

V. CONCLUSION

A novel closed solution for optimum finish, reporting and pre-processing time was obtained for a single level tree sensor network including immediately measuring data feature. Data and simulation were performed and analyzed for different scheduling protocols. A pre-processing scheduling strategy was proposed for a single level tree sensor network.

Future work will focus in the development of a methodology to study the energy use as a function of the network parameters used in this paper.

REFERENCES

- [1] M. Moges, L.A. Ramirez, C. Gamboa and T.G. Robertazzi, "Monetary Cost and Energy Use Optimization in Divisible Load Processing", Proc. of the 2004 Conference on Information Sciences and Systems, Princeton University, March 2004.
- [2] A. Cerpa et al., "Habitat monitoring: Application driver for wireless communications technology", 2001 ACM SIGCOMM Workshop on Data Communications in Latin America and the Caribbean, Costa Rica, April 2001.
- [3] G. J. Pottie, W. J. Kaiser, "Wireless Integrated Network Sensors", Communications of ACM, vol. 43, no. 5, pp. 551-558, May 2000.
- [4] M. Moges, and T.G. Robertazzi, "Load Scheduling for Measurement and Data Reporting in Wireless Sensor Networks", Proc. of the 2004 Conference on Information Sciences and Systems, Princeton University, March 2004
- [5] V. Bharadwaj, D. Ghose, V. Mani, "Design and Analysis of Load Distribution Strategies for Infinitely Divisible Loads in Distributed Processing Networks with Communication Delays", Dept. of Aerospace Engineering, Indian Institute of Science, Bangalore India Technical Report 422/GC/01-92, Oct. 1992.
- [6] V. Bharadwaj, D. Ghose, V. Mani and T. G. Robertazzi, "Scheduling Divisible Loads in Parallel and Distributed Systems, IEEE Computer Society Press (now distributed by Wiley), 1996.
- [7] H.M. Wong, D. Yu, B. Veeravalli and T.G. Robertazzi, "Data Intensive Grid Scheduling: Multiple Sources with Capacity Constraints", Proc. of the IASTED International Conference on Parallel and Distributed Computing and Systems (PDCS 2003), Nov. 2003.
- [8] V. Bharawaj, D. Ghose and T.G. Robertazzi, "Divisible Load Theory: A New Paradigm for Load Scheduling in Distributed Systems", in the special issue of Cluster Computing on Divisible Load Scheduling (D. Ghose and T. Robertazzi, editors), spring 2003.
- [9] Y.C. Cheng and T. G. Robertazzi, "Distributed Computation with Communication Delays", IEEE Transactions on Aerospace and Electronic Systems, vol. 24, no. 6, Nov. 1988, pp. 700-712.