Performance Analysis of Data Aggregation in Wireless Sensor Mesh Networks

Haoying Liu, Jian Shen, Xiaojing Yuan, Mequanint Moges
Engineering Technology Department, University of Houston, Houston, TX 77204
hliu12@uh.edu, jshen6@uh.edu, xyuan@uh.edu, mmoges@uh.edu

Abstract

In recent times the interest in wireless sensor mesh networks has grown considerably from personal, to local and metropolitan areas deployment. These networks consist of several mesh routers with minimal mobility and mesh clients that can be either mobile or stationary. The clients may also form a client mesh network among themselves and with routers. The various nodes over the network are interconnected via wireless links which might possibly employ multiple radio interfaces. One major attribute of such networks is the presence of redundant links which removes the single point failure that is present in the classical star or tree networks. Most researches in this field focus on the study of various routing protocols while we sought to introduce the application divisible load theory to find an optimum data aggregation strategy that optimize the networks performance with respect to response time and network delay. We define data aggregation as the process of data sensing and reporting back to the sink nodes, typically routers. The performance of wireless mesh network with 25 sensor nodes is examined by varying network bandwidth and sensing power of sensor nodes. Basic recursive equations for sensing and data reporting are developed for the case of homogeneous and heterogeneous mesh networks and the performance results of two representative data sensing and reporting strategies are presented.

Keywords: Performance Analysis, Network Delay, Response Time, Data Aggregation, Mesh Network, Homogeneous and Heterogeneous

1 Introduction

In recent times the interest in wireless sensor mesh networks has grown considerably from personal, local as well as metropolitan areas deployment to list few. These networks consist of several mesh routers with minimal mobility and mesh clients which can be either mobile or stationary. The clients may also form a client mesh network among themselves and with routers. The various nodes over the network are interconnected via wireless links which might possibly employ multiple radio interfaces. One major attribute of such networks is the presence of redundant links which removes the single point failure that is present in the classical star or tree networks. Most researches in this field focus on the study of various routing protocols while we sought to introduce the application divisible load theory to find an optimum sensing job scheduling strategies for data sensing and reporting back to destination in wireless sensor mesh networks.

The traditional divisible load theory has been intensively studied over the past decade with the objective to achieve an optimal distribution of divisible jobs among a number of processors and links so that the total time required to process the job is minimized (Hernandez-Ramos, 2000; Bharadwaj, 1996, 2003). A divisible load is a load that can be arbitrarily partitioned in a linear fashion among a number of processing nodes. The model assumes that there are no precedence relations among the data. In this study we are concerned with the sensing job distribution between sensor nodes so that data sensing and eventual reporting of the sensed data is completed in the shortest possible time. We assume two network types: homogeneous and heterogeneous networks. In the first case we assume that each sensor node in the network has the same sensing power and communication bandwidth. In the later case we assume that each sensor node in the network has different sensing power and communication bandwidth. In both cases we assume that each sensor node has negligible processing capabilities which make it radically different from the traditional divisible load theory. In this situation, we assume that sensing will be done at the controller level, once all the reported data is gathered from each sensor.

A considerable amount of literature on wireless sensor network scheduling has appeared over the past several years. Some representative work is now discussed. The majority of literature on wireless sensor networks involves minimizing communication since this is the dominant operation in time (Robertazzi, 2003). Some work on energy conservation strategies have included aggregating data at nodes to shorten subsequent transmissions (Tilak, 2002), probabilistically routing traffic to spread load across network nodes and prolong stored energy (Intanagonwiwat, 2002), putting sensors to sleep when they are not needed (Shah, 2001) and activating only geographically localized wireless sensors (Schurgers, 2002). In terms of the computation efficiency of sensor networks, there has been a research on sorting (Megeurdichian, 2001) and on computational problems in distributed sensor networks (Bordim, 2002). Work also has been done on improving reliability and effective operation by integrating “intelligence” into sensor nodes (Iyengar, 2002; Yuan, 2004). Finally research in (Figueroa, 2001; Moges, 2006) has shown the integration of communication with sensing speed for single level tree networks and linear daisy chain networks respectively.

The organization of this paper is as follows. Section 2 discusses the problem formulation and the system model and some notations used in this paper. In section 3 the analysis of the sensing and data reporting strategies in mesh networks is presented. In section 4, performance evaluation of the various strategies for homogenous sensor mesh network appears. In section 5, the performance results for heterogeneous sensor mesh network from extensive simulation and analysis, comparison of different scheduling strategies and recommendation of selecting optimal strategy from them are presented. Finally the conclusion and future work appears in section 6.

2 Problem Formulations and System Model

In this section, the problem formulation as well as 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 mesh network consisting of one control/destination node and 24 communicating sensor nodes as shown in Figure 1. It will be assumed that the total sensing job considered here is of the arbitrarily divisible kind that can be partitioned into fractions of jobs to be assigned to each sensor node over the network. In this case the control processor first assigns a load share to be sensed to each of the rest sensor nodes and then receives the sensed data from each node.

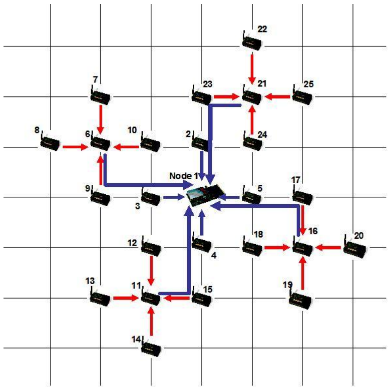

Figure 1. Example of a Mesh Network with 25 nodes

There are two ways of communication between the sensor nodes and the controller: sequential or concurrent. In the sequential communication, each sensor node is able to communicate with only one child at a time. However, in the case of concurrent communication strategy, each sensor node can communicate simultaneously/concurrently with all the child processors. In this study we consider the sequential communication strategy.

There are also different scenarios for the sensor nodes, depending whether or not they can sense and communicate at the same time. In general, we will consider two cases: with front end processors and without front end processors. In the case of sensor nodes with front end processors, it is assumed that some of the nodes in the network are equipped with front ends so that they are able to sense their own job assignment and communicate (if necessary) simultaneously. In the case of networks without front end processors, it is assumed that none of the nodes are equipped with front ends and the nodes can only sense or communicate at one time.

To describe the sensing task scheduling method, we use the notation and definitions as described in detail in (Liu, 2007 et al.). In summary, the y_i and z_i represent the sensing and communication speed of sensor node SSN^i in the network, and α_i is the fraction of sensing task assigned to it. It is assumed that every node will be assigned non-zero task, i.e., $0<\alpha_i<1$, and the task for all nodes sums to 1 ($\sum_{j=0}^{n} a_j = 1$). T_f is the total finish time or make span and $T_f = \max(T_1, T_2, \dots, T_n)$, in which T_i is the time that elapses between the beginning of the scheduling process at t = 0 and the time when SSN^i completes its reporting.

3 Proposed Strategy for Sensing and Data Reporting Time Analysis

The process of job assignment to sensor nodes and data reporting is shown through a Gantt-chart-like timing diagram in Figure 2 according to the Mesh Network with 25 Nodes (shown in Figure 1). The job scheduling strategy we follow is based on the Peters-Syska's algorithm (Peters, 1996; Liu, 2007; Drozdowski, 1999) for broadcasting in two dimensional torus network. In Peters-Syska's algorithm there are two communication phases: in the first phase of communication data is sent in a *knight move* as in the chess game. In the second phase of communication data is sent in a *cross move*. This process repeats recursively in submeshes of 25 nodes. In our case we consider a mesh network with 25 nodes only, and the process of sensing and reporting job in the mesh network is the inverse process of Peters-Syska's algorithm. During the job assignment time (T_0, represented by blue bar in time diagrams), the controller sends all the job assignments to all 25 sensors, and each sensor starts to do their sensing job after all information is completely received. Here we assume that T_0 is a short and constant task assigning time needed to start the sensing process. In the first phase of our case shown in Figure 1, after finishing sensing job, sensors 7-10, 12-15, 17-20 and 22-25 respectively and sequentially report data to the sensors 6, 11, 16 and 21 (represented by red bars in time diagrams). In the second phase, sensors 6, 11, 16 and 21 as well as sensors 2-5 sequentially report data to the controller - node 1(represented by yellow bars in time diagrams). In addition, during the data reporting time, we assume two scenarios for reporting. In the first case, Figure 2, we assume that sensors 2 – 5 which are directly connected to the controller will report to the controller before all other nodes over the network start reporting (BEFORE case). In the second case, Figure 3, we assume that sensors 2 - 5 will report to the controller after all the other sensor nodes finish their corresponding data to the controller (**AFTER** case).

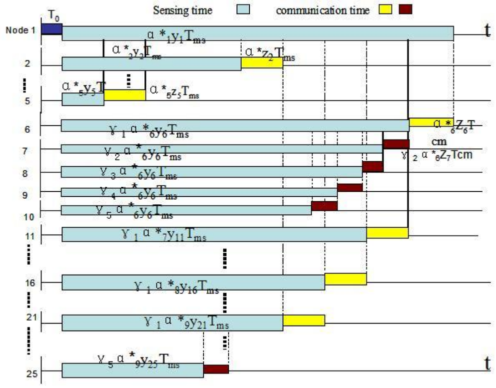

Figure 2. Timing diagram: mesh network with 25 nodes: sensor nodes 2-5 report before all other nodes start reporting

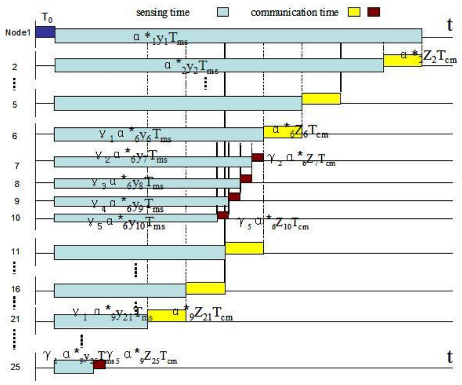

Figure 3. Timing diagram: mesh network with 25 nodes: sensor nodes 2-5 report after all other nodes finish reporting

Based on the timing diagram shown in Figure 3 one can write the following set of equations for sensor nodes 1 - 4:

$$\alpha_1^* y_1 \mathrm{T}_{\mathrm{ms}} = \alpha_2^* y_2 \mathrm{T}_{\mathrm{ms}} + \alpha_2^* z_2 \mathrm{T}_{\mathrm{cm}} \qquad (1)$$

$$\alpha_2^* y_2 \mathrm{T}_{\mathrm{ms}} = \alpha_3^* y_3 \mathrm{T}_{\mathrm{ms}} + \alpha_3^* z_3 \mathrm{T}_{\mathrm{cm}} \qquad (2)$$

$$\ldots\ldots$$

$$\alpha_4^* y_4 \mathrm{T}_{\mathrm{ms}} = \alpha_5^* y_5 \mathrm{T}_{\mathrm{ms}} + \alpha_5^* z_5 \mathrm{T}_{\mathrm{cm}} \qquad (3)$$

For sensor node 5 the expression is slightly different and is given as

$$\alpha_5^* y_5 \mathrm{T}_{\mathrm{ms}} = \gamma_1 \alpha_6^* y_6 \mathrm{T}_{\mathrm{ms}} + \alpha_6^* z_6 \mathrm{T}_{\mathrm{cm}} \qquad (4)$$

Similarly for nodes 6, 11, 16 and 21, one can write the following set of recursive equations:

$$\gamma_1 \alpha_6^* y_6 \mathrm{T}_{\mathrm{ms}} = \gamma_1 \alpha_7^* y_{11} \mathrm{T}_{\mathrm{ms}} + \alpha_7^* z_{11} \mathrm{T}_{\mathrm{cm}} \qquad (5)$$

$$\gamma_1 \alpha_7^* y_{11} \mathrm{T}_{\mathrm{ms}} = \gamma_1 \alpha_8^* y_{16} \mathrm{T}_{\mathrm{ms}} + \alpha_8^* z_{16} \mathrm{T}_{\mathrm{cm}} \qquad (6)$$

$$\ldots\ldots$$

$$\gamma_1 \alpha_{N+5}^* y_{5N+1} \mathrm{T}_{\mathrm{ms}} = \gamma_1 \alpha_{N+5+1}^* y_{5(N+1)+1} \mathrm{T}_{\mathrm{ms}} + \alpha_{N+5+1}^* z_{5(N+1)+1} \mathrm{T}_{\mathrm{cm}} \qquad (7)$$

For the rest nodes in the submesh network the following set of general recursive equation can be written as follows:

$$\gamma_i \alpha_{i+5n}^* y_{i+5n} \mathrm{T}_{\mathrm{ms}} = \gamma_{i+1} \alpha_{i+5n}^* y_{i+5n} \mathrm{T}_{\mathrm{ms}} + \gamma_{i+1} \alpha_{i+5n}^* z_{i+5n} \mathrm{T}_{\mathrm{cm}} \qquad (8)$$

In this case we define *n* as submesh group 1, 2, 3 and 4. We also define a new parameter γ showing the portion of sensing job received by sensor nodes in each submesh group.

$$g_i = \prod_{j=i+5n+1}^{i+5n+4} (f_i) / (1 + \sum_{i=1}^{5} \prod_{j=i+5n+1}^{i+5n+4} f_j) \qquad (9)$$

where

$$f_{\mathrm{i}} = (y_i + z_i \delta) / y_i \qquad (10)$$

$$\sum_{i=1}^{5} g_i = 1 \qquad (11)$$

and the parameter δ is simply the ratio of the communication intensity constant to the sensing intensity constant.

4 Performance Evaluation for a Homogenous Network

This section presents the plots of job assigned to each sensor node and plots of finish time vs. network bandwidth in a mesh network. The results are obtained by using linear programming with the objective function of minimizing the total sensing and data reporting time. In this case a homogeneous network is considered to study the effect of communication bandwidth and sensing speed variations on the total processing time. To do so, we consider the following two cases: In the first case, the task assignment to each sensor node is plotted against sensor nodes when the inverse communication speed *z* is varied/fixed and the inverse sensing speed *y* is fixed/varied. In the second case the sensing and data reporting time is plotted against network bandwidth by varying the sensing speed *y*.

The mesh network that is used to obtain the plot in Figure 4 has a homogeneous network bandwidth and sensing speed. In this case the values of T_{cm} and T_{ms} are also set to be equal to one.

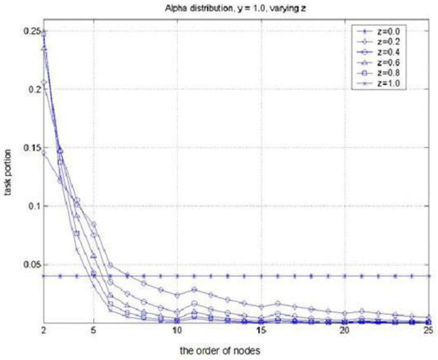

Figure 4. Load assignment when *z* is varied and *y* is fixed: sensors 2-5 report after all other nodes

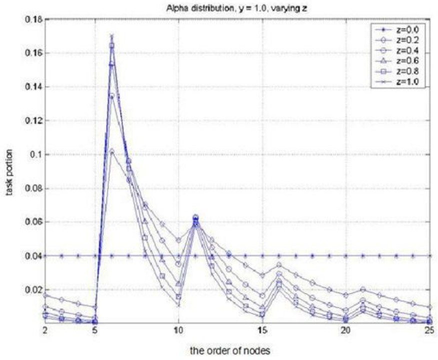

Figure 5. Load assignment when *z* is varied and *y* is fixed: sensors 2-5 report before all other nodes

The plot shown in Figure 4, presents the job assignment to each of the sensor nodes in the network for the case when the inverse communication speed *z* varies from 0.0 to 1.0. In this case sensor nodes 2 - 5 will report to the controller after all the sensors over the network finish their reporting. The result shows that much of the job is assigned to sensors 2 - 5 while the rest do little operations except the case that the total job is assigned to all nodes evenly when communication link in the network is at ideal speed (*z*=0).

Figure.5, presents similar plot however in this case sensor nodes 2 - 5 will report to the controller before all the sensors over the network start to report their data. The observation from this result is that sensors 2 - 5 in this case do little operation while sensors 6, 11, 16 and 21 get the major job assignment. In both cases the results show that as the speed of the communication link becomes slower and slower the amount of load assigned to the child processors becomes less and less. In effect this will increase the total processing time of the system since the majority of the sensing load is assigned to the first layer four sensor nodes.

On the other hand, Figure 6 and 7 show the same plot but for the case when the sensing speed *y* is varied from 0.0 to 1.0. For these parameters, the variation of the sensing speed has slight effect on the load assignment to each sensor node as compared to the communication speed variation. However the trend of the job assignment to each sensor is similar to the results obtained from the communication speed variation.

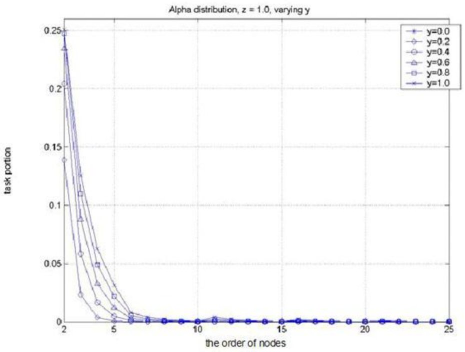

Figure 6. Load assignment when *z* is fixed and *y* is varied: sensors 2-5 report after all other nodes

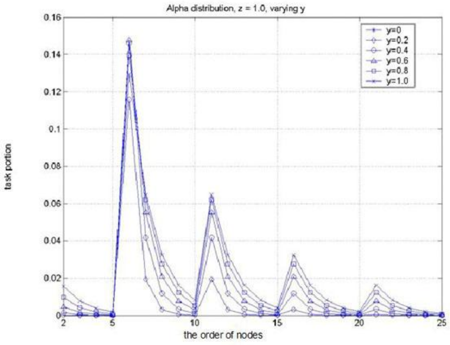

Figure7: Load assignment when *z* is fixed and *y* is varied: sensors 2-5 report before all other nodes

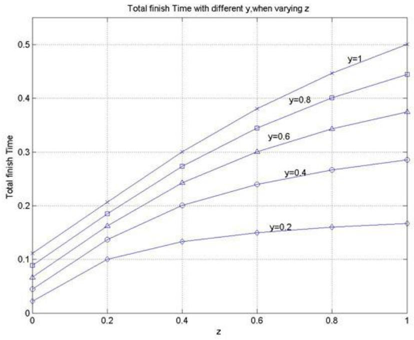

Figure 8. Finish time versus network bandwidth

In Figures 8, the finish time is plotted against the network bandwidth by using different inverse communication speeds, *z*. The sensing speed was also varied to see the effect of both the communication link and sensing speed on the total finish time.

5 Conclusions

There are many potential optimization problems involving integrating communication with sensing and/or processing. In this paper we presented the sensing job scheduling in mesh networks integrating communication with sensing. The performance analyses of two representative job scheduling strategies are examined and the effect of network bandwidth and sensing speed is studied. The total data reporting time was significantly improved as the network bandwidth increases or inverse link speed decreased. It was found out that there was a slight improvement in the total finish time as the sensing speed of the sensor nodes is increased. Another conclusion is drawn the job scheduling strategy that the sensor nodes 2-5 report after other sensor nodes should be adopted because the finish time spent in this strategy is dramatically decreased compared to another strategy.

References

Hernandez--Ramos, P., Alexander, D., Belous, A. and Behrens, J. (2000) "Changing the way we learn: How Cisco Systems is doing it" *International Workshop on*

Advanced Learning Technologies 2000, 177--179.

Bharadwaj, V., Ghose, D., Mani, V., and Robertazzi, T.G. (1996) "Scheduling divisible loads in parallel and distributed systems" *Los Alamitos, CA: IEEE Computer Society Press*

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

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

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

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

Shah, R. and Rabaey, J. (2001) "Energy aware routing for low energy ad-hoc sensor networks", 3rd *IEEE Wireless Communications and Networking Conference*, Orlando, Florida, 151-165.

Schurgers, C., Tsiatsis, V., Ganeriwal, S. and Srivastava, M. (2002) "Topology management for sensor networks: Exploiting latency and density", *ACM MobiHOC'02*, 135-145.

Megeurdichian, S., Slijepcevic, S., Karayan, V. and Potkonjak, M. (2001) "Localized algorithms in wireless ad-hoc networks: Location discovery and sensor exposure", *MobiHOC*, 106-116.

Bordim, J., Nakano, K. and Shen, H. (2002) "Sorting on a single channel wireless sensor networks", *Int. Symp. on Parallel Architectures and Networks*, 153-158.

Iyengar, S. and Wu, Q. (2002) "Computational aspects of distributed sensor networks", *Int. Symp. on Parallel Architectures and Networks*, 23-30.

Yuan, X. and Li, X. (2004) "Real time sensor fusion framework for distributed sensor network", *Proc. of FLAIRS.*

Figueroa, F. and Yuan, X. (2001) "Intutive intelligent sensor fusion with highly autonomous sensors", *Proc. of IMECE.*

Moges, M. and Robertazzi, T. (2006) , "Wireless sensor networks: scheduling for measurement and data reporting", *IEEE Transactions on Aerospace and Electronic Systems*, 42, 327-340.

Liu, H., Yuan , X. and Moges, M. (2007), "An efficient scheduling method for improved network delay in distributed sensor network", *TridentCom 2007*, Orlando, Florida.

Peters, J. and Sysks, M. (1996) "Circuit-switched broadcasting in torus networks", *IEEE Transactions on Paraller and Distributed Systems*, 7, 246-255.

Drozdowski, M. and Glazek, W. (1999) "Scheduling divisible loads in a three-dimensional mesh of processors", *Parallel Computing*, 25, 381-404.