An Efficient Task Scheduling Method for Improved Network Delay in Distributed Sensor Networks

Haoying Liu, Xiaojing Yuan, Mequanint Moges
Engineering Technology Department
University of Houston
Texas, United States
xyuan@uh.edu

***Abstract*—One of the challenges in developing smart sensor networks is the minimization of network delay or at the very least be able to have upper and lower boundaries of network delay when sensor nodes respond to higher level applications. In this paper, we present a highly efficient task scheduling method based on linear programming that integrates both sensing and networking communication delay. The objective is to minimize the total response time and global power consumption of the network with respect to the total number of sensor nodes in the network. Simulation results based on closed-form solutions for the task scheduling problem are presented for two scenarios with homogeneous and six scenarios with heterogeneous sensor nodes using single level tree-network topology. Specifically, for the heterogeneous scenarios, responding sequence that results in global optimum total respond time has also been found.**

Keywords- response time and network delay optimization, smart sensor network, linear programming

I. TASK SCHEDULING OF SMART SENSOR NETWORKS

In recent years, sensor networks have attracted significant attention due to their integration of computation, sensor technology, networking and communication. With the advances of large scale wired and wireless sensor networks, each sensor node possesses greater functionality, better sensing, processing and storage capabilities, and use radio frequency communication protocols to forward data in a multi hop mode of operation. With increasing complexity of the sensor network applications, the sensor nodes are not only responsible for gathering and blindly relaying information upward, but also expected to have some "intelligence" enabling the interaction between sensors and systems to achieve improved reliability and effective operation of a specific application [1-3].

With a grant from NASA, the ISGRIN (Intelligent Sensor GRid and INformatics) research lab in the University of Houston is working with NASA engineers to develop a Testbed of Smart Sensors (UH-ToSS). The project focuses on smart sensor nodes compatible with IEEE 1451 standards with applications for space exploration in mind. There are a number of general purpose (wireless) sensor network hardware testbeds such as the SmartDust project at UC Berkeley [4], the μAmps project at MIT [5], the MoteLab project at Harvard [6], and the GNOMES project at Rice [7]. Unlike the previous researches, the UH-ToSS focuses on providing an open platform to develop, validate, and test hardware and software components from analog and digital sensors, to communication and networking, to higher level decision support systems.

Scheduling for general-purpose sensor network is often performed to facilitate resource management of sensing, on-board computation, communication, and power consumption. Researchers have developed algorithms to schedule the idle, sleep, and wake-up time for radio communication links and the microprocessor of the sensor nodes. All these researches have been done for underlying applications with limited measurements from external environment. Thus, to conserve power, the sensor nodes wake up at certain time to communicate the raw measurement upward.

Unlike other existing scheduling algorithms for wireless sensor network testbeds, our task scheduling algorithm is designed with data-centric applications in mind. For instance, instead of sensor networks that acquire time series data at low sampling frequency, we are interested in applications dealing with sound, images, video streams, and other multi-media measurement at higher sampling frequency. The challenge here is the scheduling of arbitrary application portion assignment to each smart sensor node that optimizes the utilization of sensing, computation, communication, and power resources in the whole sensor network with minimal of network delay.

Network delay is defined here as the total response time from central node assigning sensing tasks to each sensor, sensor nodes accomplish their sensing task and complete on-board computation/processing, and reporting back to the central node. It is the one quantity defined as the summation of traditional network delay parameters such as network setup time, propagation time, and transmission time, but also the application completion time, and reporting time. From application perspective, the most important performance parameter is how fast the application can complete, and provide some decision options for the decision makers.

The design of our scheduling algorithm is based on the traditional divisible load theory [8-10] that minimizes the processing time of extensive computational jobs originating from single root processors and being processed in a multiplicity of nodes. Unlike the computational jobs, which

The project is partially supported by UH-GEAR, and NASA grant under CAN agreement no. NNS06 ZBA001C to Dr. Yuan.

can be processed in parallel and strives to end at the same time to minimize the idle time for each processor, sensing tasks are not multi-tasking in nature. The most up-to-date smart sensor node can perform one sensing task at a time, and cannot accomplish multiple tasks via multi-tasking mechanism. In addition, while the computational jobs terminate upon completion, sensing tasks do not stop when the measurements are acquired. Typically, depending on the application, sensing tasks assigned will require the smart sensor node to perform the sensing task for specific duty cycles and reporting the measurements back for further processing.

This study tackles the problem of optimally scheduling and distributing an arbitrarily portion of a data centric partitionable sensing application in a homogeneous and heterogeneous smart sensor network using single level tree network topology. The homogeneous condition assumes that each sensor node has the same sensing capacity and communication link speed. The heterogeneous condition assumes that each smart sensor node has different sensing capacity and communication link speed.

The organization of this paper is as follows. In section II, we discuss the single level tree smart sensor network model and some notations used in this paper. Section III formulates the task scheduling problem and details of the derivation of the closed form solution for optimal sensing application assignment in terms of number of sensor nodes, minimal response time, and the sensing and communication speed. In section IV, a performance evaluation of our proposed task scheduling strategy for two generic cases is presented. We reach the conclusion in section V.

II. Single level tree smart sensor network model and notations used

In this section, we present the single level tree smart sensor network model and the various network parameters used in our study.

The network structure presented in this paper is a single level tree network consisting of one controller/destination node with more computational and power resource, and N-1 smart sensor nodes with limited computational and energy resource, as shown in Fig.1. The task scheduling will be performed at the controller/destination node and partitioned sensing tasks will then be distributed among the smart sensor nodes in the network. For any application, the controller node will schedule the portion of sensing and computation task for each smart sensor node, SSN_i, distributes the tasks, and gathers the responses back. It is assumed that the total sensing task is considered to be of the arbitrary divisible kind so that there are no precedence relations among the partitions. The approach is particularly suited to the processing of very large multimedia data files that might be collected from sensor nodes.

Any partitionable application has the property that the application can be partitioned into any number of sub-tasks without any precedence relations among sub-tasks. Each sub-task can then be assigned and distributed independently along different communication links and sensor nodes. On the other hand non partitionable tasks can not be further subdivided and are required to be processed in their entirety on one processor. The problem of scheduling of such jobs is generally referred to as bin-packing problem, and has been proven to be NP-complete under several constraints [3].

When designing an integrated scheduling algorithm, it is necessary to consider the types of scheduling strategies, task distribution policies and network topologies. Research has shown [11] that the characteristics of these strategies and policies have significant effect on the performance of the network in terms of the total sensing, processing and communication time as well as the power consumption.

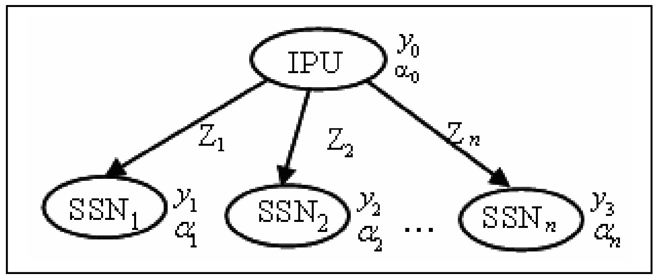


Figure 1. Schematic diagram of a single level tree network.

To describe the sensing task scheduling method, we summarize the notation and definitions used in the set of linear equations as follows.

α^*_i: The fraction of sensing task that is assigned to sensor node by the controller node. 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 ($\sum_{j=0}^{n} a^*_j = 1$).

α_i: The optimal α^* that results in minimum total respond time. $\alpha=\{\alpha_1, \alpha_2, \ldots \alpha_n\}$ is the optimal task distribution.

y_i: A constant that is inversely proportional to the sensing speed of sensor node SSN_i in the network.

z_i: A constant that is inversely proportional to the communication speed of the communication $link_i$ in the network.

T_{ms}: Sensing intensity constant. This is the time it takes for the i^{th} sensor node to accomplish the whole sensing task when $y_i = 1$. The entire assigned sensing task can be sensed on the i^{th} sensor node in time $y_i T_{ms}$.

T_{cm}: Communication intensity constant. This is the time it takes to transmit the entire sensing task over a link when $z_i = 1$. The entire sub-task 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 $sensor_i$ completes its reporting, $i = 1, \ldots, N$. This includes sensing time, responding time, and idle time. The responding time, i.e., the network delay, is the time used to transmit all data from the source node to the destination node.

T_r^*: The time when the last sensor node finishes responding (finish time or make-span). $T_r^* = \max(T_1, T_2, \ldots, T_n)$.

T_r: The minimum T_r^*, i.e. T_r^* when the task distribution is α.

For any given network, the task assigning strategy from the controller/destination to the terminal sensor nodes may be *concurrent* or *sequential*. In the *concurrent* task assigning strategy, the controller can communicate simultaneously with all the smart sensor nodes (SSN) over the network. This can be implemented with a controller which has a processor that loads an output buffer for each output link of the network. In the *sequential* task assigning strategy, the controller has one communication channel and can only distribute tasks sequentially to relevant SSNs. The concurrent task assigning strategy is simpler than the sequential task assigning strategy because it does not consider the effect of the sequence that each node receives their sensing task on its sensing and communication speed. The challenge is much more severe in a heterogeneous smart sensor network than in a homogeneous smart sensor network, as shown in our models and simulation results presented in section III and IV.

Similarly, we have *concurrent* and *sequential* responding strategy for a smart sensor network. In the *concurrent* responding strategy, the controller utilizes more than one communication channels to accept incoming reporting data from different smart sensor nodes at the same time. In the *sequential* responding strategy, the controller uses one communication channel and can only accept incoming reporting data sequentially from relevant sensor nodes.

Thus, for a smart sensor network, we need to consider both concurrent and sequential task assigning and responding strategy. In reality, in a smart sensor network designed for data-centric application, the task distribution requires much less bandwidth than the responding.

In this study we assume concurrent task assignment strategy, where every sensor node in the network will receive their sensing task then start sensing at the same time. In reality, this can be easily implemented by task multi-installment, that is, the sensor node starts its sensing task as soon as it receives a portion of it. On the other hand, when sensors report their measurement to the controller, we assumed sequential responding strategy due to the bandwidth required to receive multi-media data at the controller side. That means only one sensor node reports to the controller at one time.

III. Smart Sensor Network Sechduling Formulation and Closed Form Solution for Single Level Tree Networking Topology

Fig. 2 shows the timing diagram for a Simultaneous Sensing Start, Sequential Reporting (S4R) strategy [9]. In this timing diagram the communication link speed is shown above the x-axis whereas the sensing speed is shown below the x-axis for each smart sensor node. It can be seen in Fig. 2 that all smart sensor nodes begin to perform their sensing task at t = T_0, where T_0 is a constant task assigning time needed to start the sensing process.

To reduce the idle waiting time of the sensor nodes that are allocated to report later, the optimum scheduling algorithm should assign enough sensing task for them to keep sensing while waiting for its turn to report, as shown in Fig. 2. Those sensor nodes that finished reporting will be hibernating, thus reserve the power it has. For the next application, the task will again be assigned based on the sensing speed, communication speed and power resources available for each sensor. As the result, global optimization is achieved for the whole sensor network with respect to power consumption.

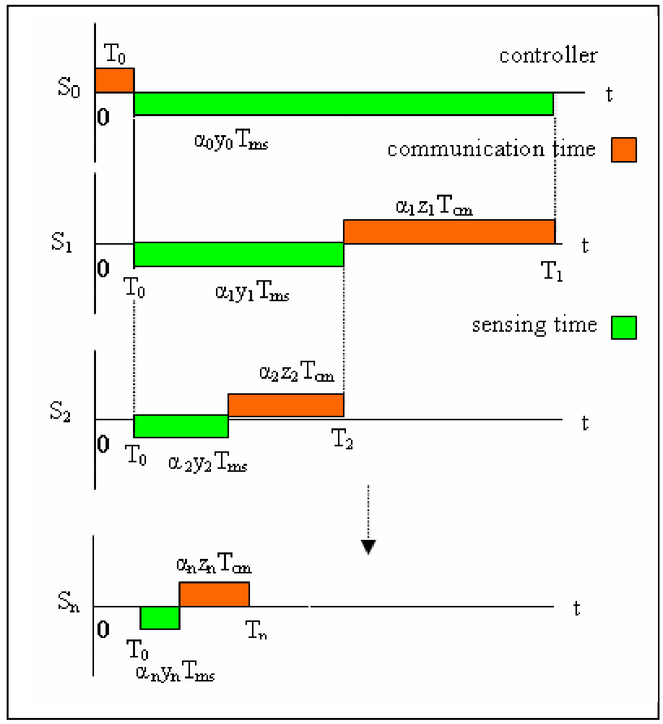


Figure 2. Time diagram for an N-node single level tree smart sensor network with simultaneous sensing and sequential reporting.

In the following section, we derive the closed form solution for the optimal task assignment α that result in minimum T_{rm}, and the value of T_{rm}, for both homogeneous and heterogeneous sensor networks.

A. *Simultaneous Sensing Start, Sequential Responding (S4R) Strategy for Homogeneous Smart Sensor Network*

The formal definition of a homogeneous smart sensor network is given below.

Definition 1: A smart sensor network is *homogeneous* if and only if the communication links (z_i, i=1,2,…,n) are the same for all the sensor nodes; and the sensing speed (y_i, i=1,2,…,n) are the same for all the sensor nodes (Equation 2).

$$y_i = y_j = y$$
$$z_i = z_j = z \qquad (2)$$

Because of the property associated with the homogeneous smart sensor network (i.e., their communication links are the same), the responding sequence does not matter. For example, node SSN_i can respond before or after SSN_j without showing any difference for the controller, i.e., no change in the task assignment. In the derivation of closed form solution, we assume node S_n responds to controller first. After it finishes reporting, SSN_{n-1} reports to the controller, and so on, until S_1 reports to the controller node (Fig. 2).

From Fig. 2. the set of linear equations for the homogeneous sensor network with simultaneous sensing starting, sequential reporting (S4R) strategy are listed in equation 3.

$$\begin{aligned}
&a_0 y_0 T_{ms} = a_1 y_1 T_{ms} + a_1 z_1 T_{cm} \\
&a_1 y_1 T_{ms} = a_2 y_2 T_{ms} + a_2 z_2 T_{cm} \\
&\ldots \\
&a_{n-1} y_{n-1} T_{ms} = a_n y_n T_{ms} + a_n z_n T_{cm}
\end{aligned} \qquad (3)$$

Solving the optimum task allocated to N sensor nodes based on the given network, communication, and sensing parameters, we get:

$$a_k = a_{k+1} \frac{y_{k+1} + z_{k+1} d'}{y_k}, \text{ k=0, 1, …, n-1} \qquad (4)$$

In which $d' = \frac{T_{cm}}{T_{ms}}$. δ′ is the coefficient constant that determines the tradeoff between the communication and sensing capability of a smart sensor node.

Let $f_k = \frac{y_k + z_k d'}{y_{k-1}}$, together with equation (2), we have:

$$f_k = f = 1 + \frac{z}{y} d' \qquad (5)$$

Using the assumption that task fraction allocated to all sensor nodes are summed to 1, i.e., $\sum_{j=0}^{n} a_j = 1$, we will get,

$$a_k = \frac{\prod_{j=k+1}^{n} f_j}{1 + \sum_{i=1}^{n} \prod_{j=i}^{n} f_j} = \frac{f^{n-k}}{1 + \sum_{i=1}^{n} f^{n-i+1}}, \quad \text{k=0, 1, ..., n-1}$$
$$a_n = \frac{1}{1 + \sum_{i=1}^{n} f^{n-i+1}} \qquad (6)$$

From the closed form solutions shown above, the task distribution among n smart sensors can be calculated given f_k (equation 5).

The total completion time T_r can then be calculated as follows:

$$T_r = T_0 + \frac{\prod_{j=1}^{n} f_j}{1 + \sum_{i=1}^{n} \prod_{j=i}^{n} f_j} y T_{ms} = T_0 + \frac{f^n}{1 + \sum_{i=1}^{n} f^{n-i+1}} y T_{ms}, n > 0 \qquad (7)$$
$$T_r = T_0 + y T_{ms}, n = 0$$

From equation 5, we have: $\begin{cases} \frac{1}{f} = 1, z = 0 \\ \frac{1}{f} < 1, z \neq 0 \end{cases}$

So, for ideal communication (z=0), $\lim_{n\to\infty} T_r = T_0$. This is intuitive. Since the communication does not take any time, and when we have unlimited number of sensors, each sensor has the same sensing speed, and the task can be partitioned arbitrarily into portions approaching zero. In this case, the total completion time only depends on the time needed to communicate the task to each sensor, T_0, which is approaching zero.

When the communication is not ideal, i.e., $z \neq 0$, $\lim_{n\to\infty} T_r = T_0 + \frac{f-1}{f} y T_{ms}$. When the number of sensor nodes is approaching infinity, the response time depends on the sensing speed and communication speed. When the communication speed decreases, and sensing speed remains the same, the total response time increases linearly with respect to the increase of the number of sensors. Similarly, when sensing speed decreases, while the communication speed remains the same, the total response time increases linearly with the increase of the number of sensors.

B. Simultaneous Sensing Start, Sequential Responding (S4R) Strategy for Heterogeneous Smart Sensor Network

The formal definition of a heterogeneous smart sensor network is described as follows.

Definition 2: A smart sensor network is *heterogeneous* if the communication links (z_i) and the sensing speed (y_i) are not the same for all the sensor nodes.

In contrast to homogeneous sensor network, in heterogeneous sensor network, the responding sequence **does** affect the network delay of the whole sensor network for optimum responding time. The goal here is to find the optimal responding sequence which yields minimum responding time. To demonstrate the effect of the responding sequence to the performance of the whole sensor network, we designed six experiments which will be reported in the next section. In these situations, S_1 and S_n in Fig. 2 represent the smart sensor nodes that report last and first respectively. For example, α_1 stands for the task portion assigned to the node which has the worst communication link if the reporting sequence is from the best communication link to the worst one. Thus we define the index for sensor node as the subscript, which will be fixed for the sensor always, e.g., SSN_1. On the other hand, the logic order of the task assignment (e.g., α_2) and its corresponding sensor node (e.g., SSN^2) are shown as superscript for each sensor node. The subscript and superscript notations are not always the same with each other. For example, if SSN_1 is the 2nd sensor that reports back, then its task assignment is α_2, and it is denoted as $SSN_1^{\,2}$.

The set of linear equations for heterogeneous sensor network and the optimal task fraction allocated to each sensor node can be calculated using equations 4 and 5. However, in the heterogeneous sensor network, equation 2 does not satisfy anymore. Now using equations 4 and 5 with the assumption $\sum_{j=0}^{n} a_j = 1$, we can derive the solution for optimal task distribution for all the sensor nodes in heterogeneous sensor network, as shown in equation 8.

$$a_k = \frac{\prod_{j=k+1}^{n} f_j}{1+\sum_{i=1}^{n}\prod_{j=i}^{n} f_j}, \quad k=0, 1, ..., n-1$$
$$a_n = \frac{1}{1+\sum_{i=1}^{n}\prod_{j=i}^{n} f_j} \qquad (8)$$

Then, the total completion time T_r can be calculated as:

$$T_r = T_0 + \frac{\prod_{j=1}^{n} f_j}{1+\sum_{i=1}^{n}\prod_{j=i}^{n} f_j} y_0 T_{ms}, n>0$$
$$T_r = T_0 + y_0 T_{ms}, n=0 \qquad (9)$$

IV. SIMULATION RESULTS AND PERFORMANCE ANALYSIS

In all of the experiments, we assume that $T_{cm}= T_{ms}= 1$, and $T_0 = 0$, in order to clearly show the effects of y_i, z_i, and reporting sequence on the total response time Tr.

A. *Optimal task assignment for homogeneous smart sensor network based on S4R strategy*

Graphical representative solutions for α_i for a smart sensor network with 10 sensor nodes are shown in Fig. 3. Fig. 3 shows the task assignment α_i versus ordinal of sensor nodes with variable communication speed z_i=z and fixed sensing speed y_i (y_i=y=1.0). When the communication is ideal, i.e., z = 0, every sensor node gets the same proportion of the task, and contributes to the improved performance of the senor network. With the decrease of the communication speed, i.e., increase in z, the task assignments become more and more skewed among the first and last reporting nodes. For example, the task assignment for the last-reporting node increases from 10% with ideal communication link to 60% with very limited communication bandwidth. On the other hand, the task assignment for the 9th and 10th sensor nodes, i.e., the nodes that respond to the controller node the second and the first, respectively, decreases from 10% with ideal communication link to almost zero when inverse communication speed goes to one or more.

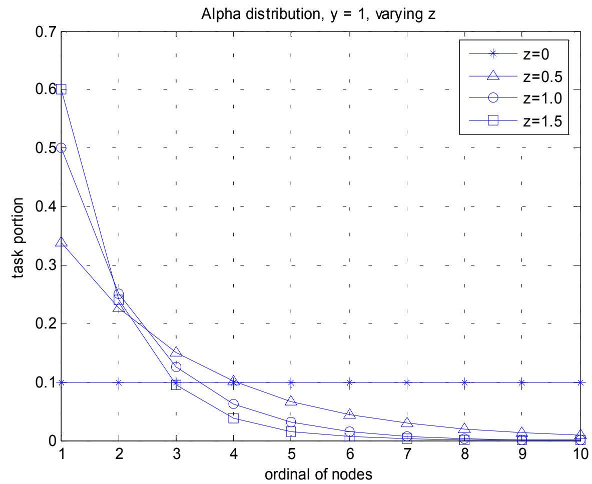


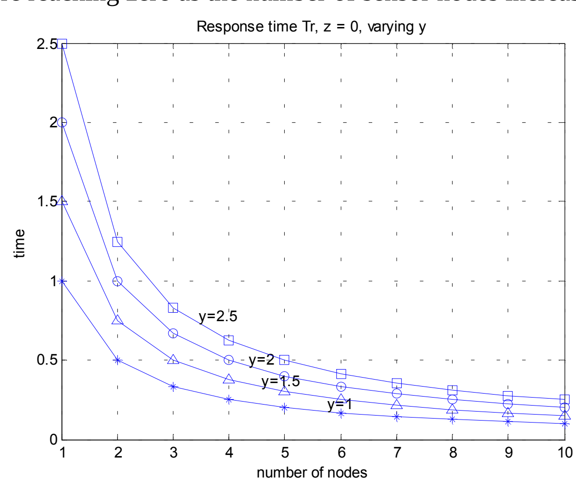
Figure 3 Optimal task assignment for 10-node sensor network. X-axis represents the ordinal of sensor node.

B. *Total completion time for S4R strategy of homogeneous sensor network*

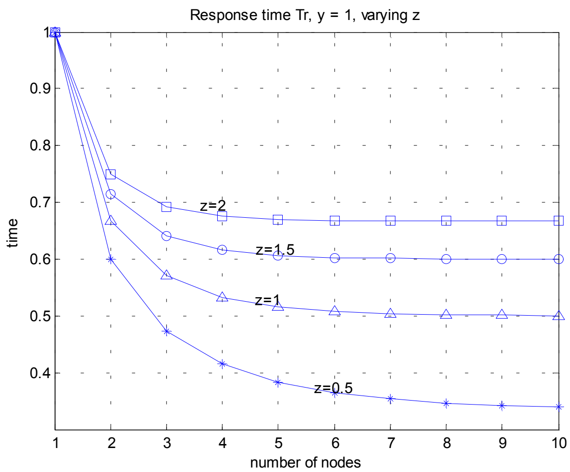
The total completion time is plotted against the number of sensor nodes in Fig. 4 (a) when the communication speed (z_i=z) is fixed and the sensing speed (y_i=y) is varied. We have the following observations when the communication is ideal.

(1) T_r decreases as the number of sensor nodes increases.
(2) The response time will go to zero when n à ∞.
(3) When the numbers of sensor nodes are the same, T_r decreases as the inverse sensing speed decreases.
(4) When the sensor network only has one controller node, the response time will be determined by the sensing time.

When the communication link is not ideal ($z \neq 0$), it has similar results. However, the response time will get saturated before reaching zero as the number of sensor nodes increases.



(a) Fixed communication speed, variable sensing speed



(b) Fixed sensing speed, variable communication speed

Figure 4 Response time versus number of sensor node for single level tree homogeneous smart sensor network

Fig. 4(b) shows the total response time versus number of sensor nodes in a smart sensor network with fixed sensing speed and varied communication speed. The response time improves as the inverse communication speed decrease. However, when the communication speed decreases, the response time reaches saturation point more quickly, resulting in lesser improvement of the response time by adding more sensor nodes.

C. *Optimal responding sequence that results in global minimum T_r for S4R of heterogeneous sensor network*

Six representative experiments were designed to demonstrate the effect of the data responding sequence on the performance of the whole sensor network. The first set of three scenarios was designed to study the effect of sensing speed on the total completion time and the task assignment among different sensor nodes. Similarly, the second set of three scenarios was designed to study the effect of communication speed on the total completion time and the task assignment distribution.

Without lose of generality, we fixed the communication speed to z_i=z=1.5 and studied three responding sequences based on their sensing speed:

Case 1.1: The sensor nodes were sorted in descending order based on their inverse sensing speed. The sensor node with the worst sensing speed responded first, and the node with the best sensing speed responded last, i.e., responding sequence from the sensor node with the largest y_{max} to the sensor node with the smallest y_{min}.

Case 1.2: The sensor nodes were sorted in ascending order based on their inverse sensing speed. The sensor node with the best sensing speed responded first, and the node with the worst sensing speed responded last, i.e., responding sequence from the sensor node with the smallest y_{min} to the sensor node with the largest y_{max}.

Case 1.3: Random response sequence, independent of y.

In these three cases, we assumed that the communication link between each sensor node and the controller node was identical, i.e., z_i=z=1.5. Their corresponding sensing speed, y_i, increased from 1, with a step of 0.5, as the number of sensor nodes increased. For example, when there were two nodes in the network, there was one controller node and one smart sensor node. Then the sensing speed y_i took either y_i=1, or y_i=1.5 (i=1,2). When we had 8 nodes in the sensor network, then each sensor node had sensing speed within the set, i.e., $y_i \in \{1, 1.5, 2, 2.5, 3, 3.5, 4, 4.5\}$.

Fig. 5 shows the total response time for the above three cases. It is clear that the response time decreased when the number of sensor node increased. More interestingly, the optimal total completion time for all three responding sequences follows the same curve.

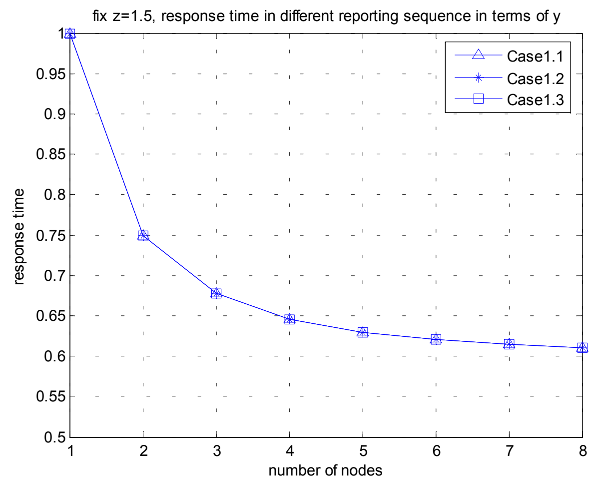


Figure 5 Total response time decreases as the number of sensor nodes increases in a heterogeneous sensor network for all three responding sequences based on sensing speed.

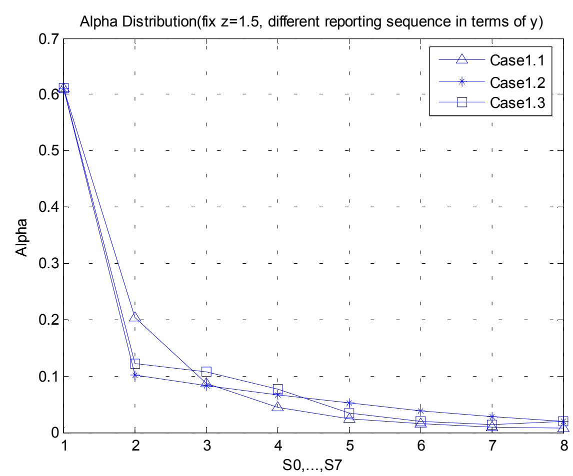


Figure 6 Task distribution for a 8-node heterogeneous sensor network with fixed communication speed and three reporting sequences based on their sensing speed. X-axis represents the ordinal of sensor node.

Fig. 6 shows the task assignment distribution for a 8-node heterogeneous sensor network that has one controller and seven smart sensors. The order for each sensor to responds to the controller node is denoted as the nodes' superscript, i.e.,

SSN^i. Thus, for the 8-node heterogeneous sensor network we are studying, the seven smart sensor nodes are ordered by their sensing speed and denoted as $\{SSN^1, SSN^2, SSN^3, \ldots, SSN^7\}$ with corresponding sensing speed as {4.5, 4, 3.5, 3, 2.5, 2, 1.5} for Case 1.1; {1.5, 2, 2.5, 3, 3.5, 4, 4.5} for Case 1.2; and random ordering for Case 1.3.

From Fig. 6, we observe that the controller node had always been assigned the largest portion of sensing task (60%) for all three responding sequences. Also, for most of the sensor nodes, the task distribution for Case 1.1 and Case 1.2 define the boundary for that of Case 1.3 (random responding sequence). For the 3rd and 4th node, the random responding sequence actually performs better than either extreme case defined in Case 1.1 and Case 1.2.

The next three experiments results demonstrate how the sensor network behaves when we fix the sensing speed y_i=y=1.5 and varying the reporting sequence based on inverse individual communication speed z_i. Similar to Case 1.1 ~ Case 1.3, we define the three scenarios below:

Case 2.1: The sensor nodes are sorted in descending order based on their communication link, z_i. The sensor node with the worst communication link responds first, and the node with the best (or approaching ideal) communication link responds last. That is, responding sequence from the sensor node with the largest z_{max} to the sensor node with the smallest z_{min}.

Case 2.2: The sensor nodes are sorted in ascending order based on their communication link, z_i. The sensor node with the best communication link responds first, and the node with the worst communication link responds last. That is, responding sequence from the sensor node with the smallest z_{min} to the sensor node with the largest z_{max}.

Case 2.3: Random response sequence, independent of z.

In these three cases, we assumed that the sensor nodes have identical sensing speed, y_i=y=1.5. The communication link speed, z_i, increased from 1, with a step of 0.5, as number of sensor nodes increased. For example, for a 2-node network, we have one smart sensor node and one controller node. Then the communication speed z_i took either z_i=1 or z_i=1.5 (i=1,2). For an 8-node network, each smart sensor can have communication speed within the set of {1, 1.5, 2, 2.5, 3, 3.5, 4, 4.5}.

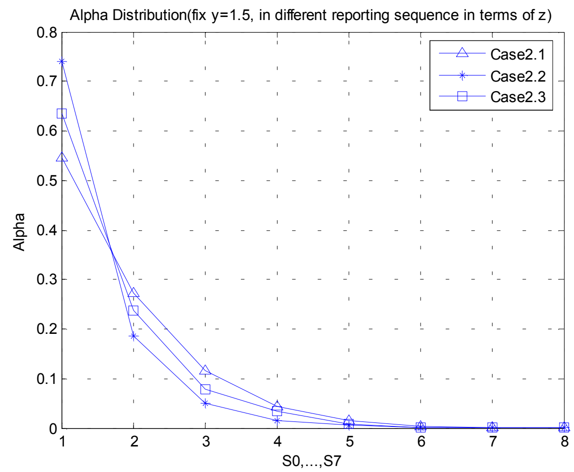


Figure 7 Task distribution for heterogeneous sensor network with fixed sensing speed and three reporting sequences. X-axis represents the ordinal of sensor node.

Fig. 7 shows the task distribution among the controller, from the last-reporting node to the first-reporting node for the above three cases. We observed that the task assignment for Case 2.3 always fell between the task assignment boundaries defined by Case 2.1 and Case 2.2. Unlike the Case 1.1~Case 1.3, the controller node was assigned different task portion depends on the reporting sequences because the communication channel within the controller could be less than ideal. Still, the controller node was always assigned the largest portion of the sensing task.

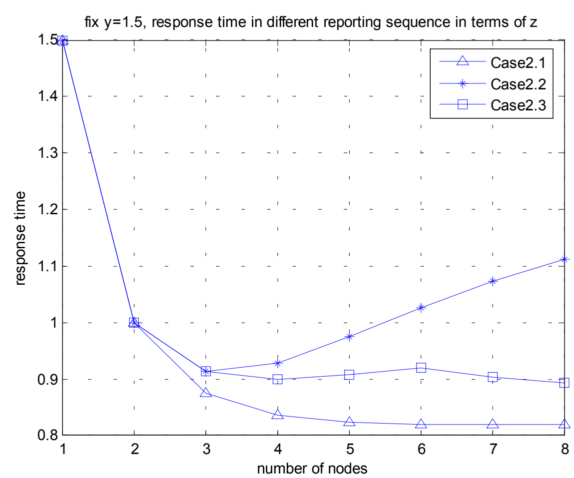


Figure 8 Total response time for heterogeneous sensor network with three reporting sequences and different number of sensor nodes.

Fig. 8 shows the total response time versus number of sensor nodes for the three reporting sequences defined in Case 2.1~2.3. It is interesting to observe that only in Case 2.1, where the reporting sequence corresponds to the order from the worst communication link to the best communication link, the increase in the number of sensors resulted in better total response time. For the other two cases, the communication speed dominates the response time, thus resulting in higher response time when the network has more than three smart sensor nodes. The zig-zag curve shown for Case 2.3 is due to the randomness of the reporting sequence, poor or good. Since Case 2.3 includes every possible respond sequence after extensive experiments, it can be concluded that Case2.1, which is reporting from largest z to smallest z, is the best reporting sequence in terms of communication link. Case 2.2 defines the upper boundary (worst case scenario) with respect to the total response time.

We can see from simulation results for Case1.1~1.3 that, the optimal respond sequence is independent of y. From simulation results for Case 2.1~2.3, we have the conclusion that Case 2.1 yields minimum T_r. As a result, it can be concluded that the optimal respond sequence is from the node which has the largest z to the node which has the smallest z, regardless what sensing speeds they have.

We can also reach the conclusion that the communication speed heterogeneity in the whole sensor network has more influence on the total response time than the sensing speed heterogeneity by comparing experiments results shown in Fig. 5 and Fig. 8. That is, different respond sequence in terms of z results in lager variation in T_r than that resulted from different respond sequence in terms of y.

V. DISUCSSION, CONCLUSION, AND FUTURE WORK

This paper reports on a new integrated scheduling algorithms we have developed and the simulation results we did for both homogeneous and heterogeneous sensor network using single-level tree network topology. For both types of networks, the simulation results show that the communication link has more influence on the total response time than the sensing speed. However, for homogeneous sensor network, the total response time improves (i.e., decrease) with increase of the number of smart sensors in the network. However, this is not true for heterogeneous sensor network. We designed six experiment cases for heterogeneous sensor networks to study the effect and trade-off between sensing, communication speed, and total response time. These six experiments explored the Ad Hoc property of the sensor network in terms of both their communication link and sensing speed.

For real-world applications of the ad hoc sensor network, the communication and sensing speed of each smart sensor node are not only independent, but also change over time. We plan to extend the research results presented in this paper and develop scheduling algorithm for ad hoc mesh network protocol. In addition, the task scheduling algorithm will be implemented and validated on a real smart sensor network testbed, UH-ToSS, currently under development in the ISGRIN research lab at University of Houston.

REFERENCES

[1] X. Yuan, X. Li, "Real-time sensor fusion framework for distributed sensor network," Proceedings of FLAIRS 2004, May, 2004, Florida.

[2] F. Figueroa, X. Yuan, "Intuitive Intelligent Sensor Fusion with Highly Autonomous Sensors," Proceedings of IMECE 2001.

[3] J. Sohn and T.G. Robertazzi, Optimal divisible load sharing for bus networks, *IEEE Transactions on Aerospace and Electronic Systems*, 32(1), 34-40, 1996.

[4] J.Khan, R. Katz, K.Pister, "Next Century Challenges: Mobile Networking for Smart Dust," in proceedings of the 5th Annual ACM/IEEE International Conference on Mobile Computing and Networking, pp. 271-278, 1999.

[5] R. Min, M. Bhardwaj, S. Cho et al, "An Architecture for a Power-Aware Distributed Microsensor Node," in IEEE Workshop on Signal Proc. Systems, pp. 581-590, Oct. 2000.

[6] G. Werner-Allen, P. Swieskowski, W. Welsh, "Motelab: A Wireless Sensor Network Testbed," in Proceedings of the 4th International conference on Information Processing in Sensor Networks (ISPN '05), Special Track on Platform Tools and Design Methods for Network Embedded Sensors (SPOTS), pp. 73-78, April 2005.

[7] E. Welsh, W. Fish, J.P. Frantz, "GNOMES: A Testbed for Low Power Heterogeneous Wireless Sensor Networks," IEEE International Symposium on Circuits and Systems (ISCAS), Bangkok, Thailand, 2003.

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

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

[10] T.G. Robertazzi,"Ten Reasons to use Divisible Load Theory" Computer Journal, 36(1), pp. 63-68, 2003.

[11] M. A. Moges and Thomas G. Robertazzi, "Wireless Sensor Networks: Scheduling for Measurement and Data Reporting", IEEE Transactions on Aerospace and Electronic Systems, Vol. 42, No. 1, pp. 327-340, January 2006.