Task Mapping and Scheduling in Wireless Sensor Networks

M. H. A. Awadalla

Abstract— **Collaborative processing among sensors to fulfill given tasks is a promising solution to save significant energy in resource - limited wireless sensor networks. Quality of Service such as lifetime and latency is largely affected by how tasks are mapped to sensors in a network. Due to the limitations of wireless sensor networks, existing algorithms cannot be directly used. This paper presents an efficient allocating algorithm that allocates a set of real-time tasks with dependencies onto a sensor network. The proposed algorithm comprises linear task clustering algorithm and sensor assignment mechanism based on a task duplication and migration scheme. It simultaneously schedules the computation tasks and associated communication events of real time applications. It reduces inter-task communication costs and moderates local communication overhead incurred due to communication medium contention. Performance is evaluated through experiments with both randomly generated Directed Acyclic Graph (DAG) and real-world applications. Simulated results and qualitative comparisons with the most related literature, Multi-Hop Task Mapping and Scheduling (MTMS), Distributed Computing Architecture (DCA), and Energy-Balance Task Allocation (EBTA), demonstrated that the proposed scheme significantly surpasses the other approaches in terms of deadline missing ratio, schedule length, and total application energy consumption.**

Index Terms*—*wireless sensor networks, task scheduling, clustering, real time applications, task duplication and migration

I. Introduction

Wireless sensor networks (WSNs) have recently come into prominence because they hold the potential to revolutionize a wide spectrum of both civilian and military applications, including environmental monitoring, scene reconstruction, motion tracking, motion detection, battlefield surveillance, remote sensing, global awareness, etc. [1]. WSNs consist of hundreds to thousands of tiny, inexpensive, and battery-powered wireless sensing devices which organize themselves into multi-hop radio networks [2-3]. The availability of inexpensive hardware such as low cost small-scale imaging sensors, CMOS cameras, and microphones, has immensely funneled the emergence of a new class of wireless sensor networks, known as Multimedia Wireless Sensor Networks (MWSN). These sensor networks will create a new wave of applications that interface with the real world environment, for example, multimedia surveillance sensor networks, traffic monitoring, and environmental monitoring [4].

For most of these applications, it might be beneficial for the sensor network paradigm to be rethought in view of the need for energy efficient multimedia algorithms with tight Quality of Service (QoS) expectations [5]. Real-time, collaborative in-network processing gains recognition as a viable solution for balancing the performance and consumption in MWSN [6]. These algorithms allow the extraction of semantically relevant information at the edge of the sensor network. Applying these algorithms assists at increasing the system scalability by reducing the transmission of redundant information, along with merging data originated from multiple views, on different media, and with multiple resolutions [5,7].

Collaborative in-network processing partitions applications into smaller tasks executed in parallel on different sensor nodes. Dependencies between tasks are maintained through the exchange of intermediate results between sensor nodes [8]. Therefore, task mapping and scheduling play an essential role in collaborative in-network processing by solving the following problems. First, it assigns tasks onto sensors. Second, it determines the execution sequence of tasks on sensors. Finally, it schedules communication transactions between sensor nodes [9-10].

This paper proposes a task mapping and scheduling algorithm resilient for real-time tasks in WSN. The proposed approach simultaneously exploits linear clustering algorithm augmented with task duplication and migration approach. The proposed approach aimed at increasing network lifetime provided that timing constraints are guaranteed. Furthermore, task scheduling and communication scheduling in the proposed approach are carried out in parallel, resulting in a realistic schedule due to the incorporation of communication contention awareness in the task scheduling, which is critical in real time applications.

The rest of the paper is organized as follows: Section 2 discusses the most related work to the theme of this paper. Section 3 details the underlying system architecture. Section 4 introduces the proposed task mapping algorithm. Section 5 shows the performance evaluation results. Finally, section 6 concludes the paper.

II. Background

Collaborative in-network processing has been widely pursued by the research community in order to achieve

Manuscript received April 30, 2013; revised November 6, 2013.

M. H. A. Awadalla is with the Electrical and Computer Engineering Department, University of Sultan Qaboos, Oman, on leave from the Communications, Electronics and Computer Department, Faculty of Enginnering, Helwan University, Egypt. (e-mail: medhatha@squ.edu.om).

energy saving and network scalability objectives.

Tian et al. [11] proposed an online task scheduling mechanism (CoRAI) to allocate the network resources between the tasks of periodic applications in wireless sensor networks in an iterative manner: the upper-bound frequencies of applications are first evaluated according to the bandwidth and communication requirements between sensors. The frequencies of the tasks on each sensor are then optimized subject to the upper-bound execution frequencies. However, CoRAl assumes that the tasks are already assigned to sensors without addressing the task mapping problem. Furthermore, energy consumption is not explicitly discussed in [11]. Authors in [12] proposed a Distributed Computing Architecture (DCA) which executes low-level tasks on sensing sensors and offloads all other high-level processing tasks to cluster heads. However, processing high-level tasks can still exceed the capacity of the cluster heads' computation power. Furthermore, the application-specific design of DCA limits its implementation for generic applications. Yu and Prasanna [13] proposed an Energy-Balance Task Allocation (EBTA) onto a single-hop cluster of homogenous sensor nodes connected with multiple wireless channels. In their work, communication over multiple wireless channels is first modeled as additional linear constraints of an Integer Linear Programming (ILP) problem. Then, a heuristic algorithm is presented to provide a practical solution. However, the communication scheduling model in [13] does not exploit the overhearing property of wireless communication, which can conserve energy and reduce schedule length. Furthermore, the small number of available orthogonal channels cannot satisfy the requirement of multiple wireless channels assigned in every cluster, especially in densely deployed networks. Zhu et al. [14] exploited divide-and-conquer technique in order to allocate tasks for heterogeneous sensor networks. The tasks are first grouped into task partitions, and then optimal execution schedule based on the optimal schedules of the tasks partitions is generated.

Kumar et al. [15] presented a data fusion task mapping mechanism for wireless sensor network. The proposed mechanism comprises a data fusion API and distributed algorithm for energy aware role assignment. The data fusion API enables an application to be specified as a coarse-grained dataflow graph. Meanwhile, the role assignment algorithm maps the graph onto the network, and optimally adapts the mapping at run-time using role migration. The authors assumed an existing underlying communication model.

Gu et al. [10] proposed EcoMapS algorithm for energy constrained applications in single-hop clustered wireless sensor networks. EcoMapS aimed at mapping and scheduling communication and computation simultaneously.

EcoMapS aims to schedule tasks with minimum schedule length subject to energy consumption constrains. However, EcoMapS does not provide execution deadline guarantees for applications. Authors also presented Multi Hop Task Mapping and Scheduling (MTMS) for multi-hop clustered wireless sensor networks. This work simultaneously addressed computation and communication scheduling. Further, the task mapping is maintained through adopting Min-Min task scheduling algorithm. However, MTMS shows a very low capacity to meet strict applications deadline.

Phung et al., [16] developed multichannel communication protocols to alleviate the effects of interference and consequently improve the network performance in wireless sensor networks requiring high bandwidth. In their work, it is proposed a contention-free multichannel protocol to maximize network throughput while ensuring energy-efficient operation. Arguing that routing decisions influence to a large extent the network throughput, they formulate route selection and transmission scheduling as a joint problem and propose a reinforcement learning based scheduling algorithm to solve it in a distributed manner. The proposed solution not only provide a collision-free transmission schedule but also minimize energy waste, which makes it appropriate for energy-constrained wireless sensor networks.

Dai et al., [17] presented an energy-aware workload dispatching simulator that assists data center administrators capturing the probable energy usage profiles with various dispatching algorithms and workload patterns. The simulator imitates the behavior of real-world workload dispatching to a heterogeneous cluster. A model is built to represent the key characteristics of a heterogeneous cluster. A set of emulated workloads based on real-world traffic traces is used to test this simulator. The result shows that the simulator produces a power usage profile that is very similar to the real-world data.

Shigei et al., [18] presented battery-aware algorithms taking into account the node's battery levels. For mobile relay, given a sequence of relaying nodes, their algorithms determine the movement of relaying nodes according to not only the total cost of movement and communication but also their battery levels. Further, in their paper, they proposed battery-aware algorithm for initial route construction. Initial route construction is needed for determining the sequence of relaying nodes, which is provided to mobile relay algorithms. The simulation results show that, for most cases, there is an improvement in the performance in terms of network lifetime.

Liu and Xu [19] focused on sensor scheduling and information quantization issues for target tracking in wireless sensor networks (WSNs). To reduce the energy consumption of WSNs, it is essential and effective to select the next tasking sensor and quantize the WSNs data. In existing works, the goals of sensor scheduling include maximizing tracking accuracy and minimizing energy cost. In their paper, the integration of sensor scheduling and quantization technology is used to balance the tradeoff between tracking accuracy and energy consumption. A real tracking system platform for testing the novel sensor scheduling and the quantization scheme is developed. Energy consumption and tracking accuracy of the platform under different schemes are compared finally.

Apart from all these efforts, this work is motivated for addressing all the above mentioned drawbacks and developing a constrained task mapping and scheduling algorithm for multi-hop clustered wireless sensor networks. The main idea behind the proposed algorithm is to group tasks that are heavily communicating with each other to be processed on the same sensor. Thereby, it will reduce the number of inter-task communication operations. Furthermore, the proposed algorithm tries to redundantly allocate some of the application tasks on which other tasks critically depend to the same sensor, which in turn yields at significant reduction in the start times of waiting tasks and eventually improves the overall schedule length of the application. Thus, it guarantees meeting very strict application deadlines.

III. Preliminaries

The proposed task mapping and scheduling strategy targets multi-hop cluster-based network architectures, and the following sections discuss the assumed network, interference, application and energy consumption models.

A. Network Model

1. All sensor nodes are grouped into k-hop clusters, where k is the hop count of the longest path connecting any two nodes.
2. Each cluster is assumed to execute a specific application, which is either assigned during the network set up time or remotely distributed by the base station during the network operation.
3. Cluster heads are responsible for creating the applications' schedules within the clusters.
4. Location information is locally available within clusters.
5. Intra-cluster communication is assumed to be handled over a single common channel, which results in further constrains on the scheduling problem arises from the contention taking place in the shared communication channel, because of sensor competing on the shared communication channel.

B. Interference Model

This work assumes that communication within each cluster is handled over a single common channel. In other words, the communication channel is shared by all sensors within each cluster. Thus, one of the major problems that will arise is the reduction of capacity due to the interference caused by simultaneous transmissions. So, in order to achieve a robust and collision free communication, a careful interference-aware communication schedule should be constructed.

In this paper, we assume that the time is slotted and synchronized, and to schedule two communication links at the same time slot, we must ensure that the schedule will avoid the interference. Two different types of interference have been studied in the literature [20], namely, primary interference and secondary interference. Primary interference occurs when a node transmits and receives packets at the same time. Secondary interference occurs when a node receives two or more separate transmissions. Here, all transmissions could be intended for this node, or only one transmission is intended for this node. Thus, all other transmissions are interference to this node. Several different interference models have been used to model the interferences in wireless networks. However the commonly used RTS/CTS interference model is adopted throughout this work. In this model, all nodes within the interference range of every pair of either the transmitter or the receiver cannot transmit. Thus, for every pair of simultaneous communication links, say m_{ij} and m_{pq}, it should satisfy that they are four distinct nodes, i.e., $s_i \neq s_j \neq s_p \neq s_q$, and s_i and s_j are not in the interference ranges of s_p and s_q, and vice versa [20].

C. Application Model

Directed Acyclic Graph (DAG) can represent applications executed within each cluster. A DAG T = (V, E) (consists of a set of vertices V representing the tasks to be executed and a set of directed edges E representing communication dependencies among tasks. The edge set E contains directed edges e_{ij} for each task $v_i \in V$ that task $v_j \in V$ depends on. The computation weight of a task is represented by the number of CPU clock cycles to execute the task. Given an edge e_{ij}, v_i is called the immediate predecessor of v_j and v_j is called the immediate successor of v_i. An immediate successor v_j depends on its immediate predecessors such that v_j cannot start execution before it receives results from all of its immediate predecessors. A task without immediate predecessors is called an entry-task and a task without immediate successors is called an exit-task. A DAG may have multiple entry tasks and one exit-task. If there are more than one exit-tasks, they will be connected to a pseudo-exit-task with computation cost equal to zero [13]. Models of randomly generated, Gaussian elimination, LU factorization [21] and real-life distributed visual surveillance [8] task graphs are considered in this paper. Some of these task graphs are illustrated in figures 1 and 2.

D. Energy Consumption Model

The energy consumption of transmitting and receiving l bit data over a distance d that is less than a threshold d are defined as E_{tx} (l,d) and E_{rx}(l), respectively [8]:

$$E_{\alpha}(l,d) = E_{elec}.l + \varepsilon_{amp}.l.d^2 \quad (1)$$

$$E_{rx}(l) = E_{elec}.l \quad (2)$$

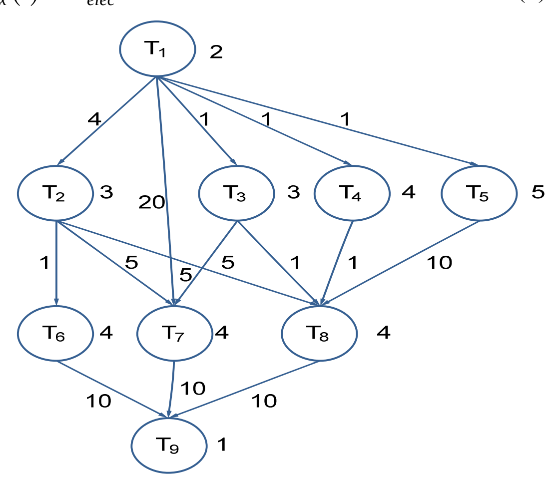

Fig1. Task graph for randomly generated application

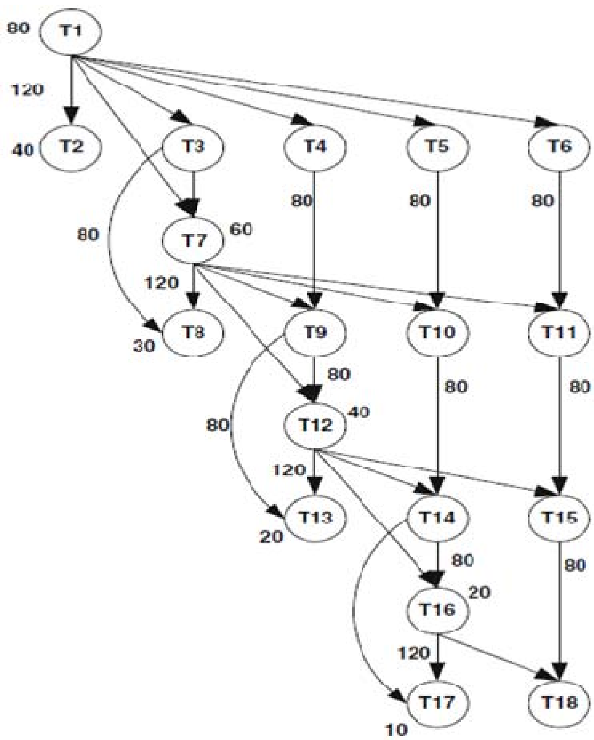

Fig2. Gaussian elimination with 18 tasks

where E_{ele} is the energy dissipated to run the transmit or receive electronics, and ε_{amp} is the energy dissipated by the transmit power amplifier. In the proposed communication scheduling algorithm, the energy consumption incurred due to sending and receiving a data packet can be expressed as in equation (1) and equation (2) respectively.

Also the energy consumption of executing N clock cycles with CPU clock frequency f is given as [8]:

$$E_{comp}(V_{dd}, f) = NCV_{dd}^2 + V_{dd}\left(I_o e^{\frac{V_{dd}}{\eta V_\tau}}\right)\left(\frac{N}{f}\right) \qquad (3)$$

$$f \cong K(V_{dd} - C) \qquad (4)$$

where $V\tau$ is the thermal voltage, V_{dd} is the supply voltage, and C, Io, n, K, c are processor-dependent parameters [20, 22].

IV. 4. The Proposed Task mapping and scheduling algorithm

This section presents the proposed task mapping and allocation algorithm. The proposed algorithm comprises two mechanisms: Linear task clustering algorithm and sensor assignment mechanism based on a task duplication and migration scheme. First, the proposed algorithm starts with partitioning the application directed acyclic graph using linear task clustering algorithm. This partitioning aims at mitigating the communication overhead such that the heavily communicated nodes are assigned to the same sensor node. Second, a sensor assignment mechanism is applied. That phase starts with mapping each partition (cluster) into an actual sensor. Finally, a task scheduling algorithm based on task duplication and migration is applied. The following sections describe the proposed algorithms in full details.

A. *Task Clustering*

This phase assumes an unlimited number of sensors, implying that the number of clusters is also unlimited. Linear clustering first determines the set of nodes constituting the critical path, then assigns all the critical path nodes to a single cluster at once. These nodes and all edges incident on them are then removed from the directed acyclic graph. The linear clustering algorithm is outlined below in figure 3.

1. Initially mark all edges as unexamined
2. WHILE there is an edge unexamined DO
3. Determine the critical path composed of unexamined edges only.
4. Create a cluster by putting the communication load equal to zero on all the edges on the critical path.
5. Mark the entire edges incident on the critical path and the entire edges incident to the nodes in the cluster as examined.
6. ENDWHILE

Fig.3. Linear clustering sequence

B. *Sensor Assignment Mechanism*

The obtained task clusters from the previous step are scheduled on the actual sensors through the following steps:

1. Map the obtained μ task clusters into p physical sensors.
2. Determine the execution sequence of the computation tasks on sensors and schedule the communication between the sensor nodes.

Cluster mapping

In this phase, the obtained task clusters from the previous step are mapped into the actual sensor nodes. As the main concern in this paper is proposing an energy-aware scheduling algorithm, this mapping takes into account the remaining energy level of the sensor nodes. This means that, the sensor node with higher remaining energy level will be assigned more working load than that having less remaining energy. It is worthy to be noted that multiple task clusters can be mapped to the same sensor node. First the load of each task cluster is computed. Then the normalized load of each sensor node is computed using equation (5), in which the sum of all loads of all task clusters assigned to the sensor is normalized by the sensor remaining energy.

$$L_k = \frac{\sum_i C_i}{E_k} \qquad (5)$$

where, Ci is the energy needed to execute task cluster i, and E_k is the remaining energy of sensor k.

Figure 4 depicts the pseudo code of this phase. Initially, all task clusters are sorted in non-increasing order of their load. Then for each cluster, the normalized load of each sensor node is calculated as if it is assigned to it. Then the cluster would be assigned to the sensor node that gives the minimal normalized load.

1. Sort the list Π containing all unmapped task clusters
2. WHILE Π is not empty DO
3. Select the first element π in Π
4. Calculate the normalized load for each sensor node
5. Assign π to the sensor node that gives the minimal normalized load
6. Update the normalized load of the sensor
7. Remove π from Π
8. ENDWHILE

Fig. 4. Cluster mapping sequence

C. *Task scheduling*

In this step, determining the execution sequence for the tasks on the sensors is carried out. This step comprises two components: task scheduling with duplication, and global task migration. Figures 5 and 6 outline the pseudo code and the flow chart for the proposed scheduling algorithm. Initially, all tasks are sorted into a list L in which tasks are ordered according to the bottom level priority and precedence constraint. Without any duplication, the algorithm first attempts to schedule the task under consideration to the assigned sensor. Obviously, to calculate the task starting time on its assigned sensor $t_s(v_j,s)$, all receiving communication transactions from v_j parents should be scheduled on the wireless channel. The task critical parent v_{cp} which has the heaviest communication and the latest arrival time is identified. Then duplicating the task critical parent v_{cp} is investigated. If this duplication helps in advance the task starting $t_s(v_j)$ time, reducing the consumed energy, meanwhile preserves the task deadline, this phase is accepted. Otherwise, it is rejected.

In some cases, the duplication mechanism fails to meet the deadline of some tasks. In such cases, the algorithm employs a global migration process for the task, where the task under consideration began to be migrated to another sensor. To reduce this migration impact on the energy increase, the destination sensor is selected as the sensor that holds its critical parent.

1. Traverse the application graph V downwards and compute Latest Finishing Time (LFT) for every task.
2. Sort tasks v ϵ V into list L according to precedence constrains.
3. For every v_j ϵ V DO
// Calculate the Earliest Starting Time of v_j on its assigned sensor s // ts (v_j, s)
4. For each v_i ϵ pred(v_j) DO
5. IF SEN(v_i) $\neq$ s THEN
6 Determine route M < m_1, m_2,…,m_n> from SEN(v_i) to s
7. Process links from m_1 to m_n and assign to each m_k the earliest free interval on the communication channel not causing any interference.
8. ENDIF
9. ENDFOR
10. Calculate $t_s(_{vj}, s)$
// Check the duplication condition
11. If duplication condition is satisfied THEN
12. Duplicate v_{cp} on s
13. Schedule v_j on s
14. ELSE
15. IF $t_f(v_j, s) < LFT(v_j)$ THEN
16. Schedule v_j on s
17. Else
18. Migrate v_j to s(v_{cp})
19. Schedule v_j on s(v_{cp})
20. ENDIF
21. ENDIF
22. ENDFOR

Duplication Conditions:

$$t_f\,(v_i,s)_{with\ duplication} < LFT(v_i)$$

$$t_s\,(v_i,s)_{with\ duplication} < t_s\,(v_i,s)_{with\ no\ duplication}$$

$$Cost(v_i)_{with\ duplication} < Cost(v_i)_{with\ no\ duplication}$$

Fig. 5. Task scheduling scheme

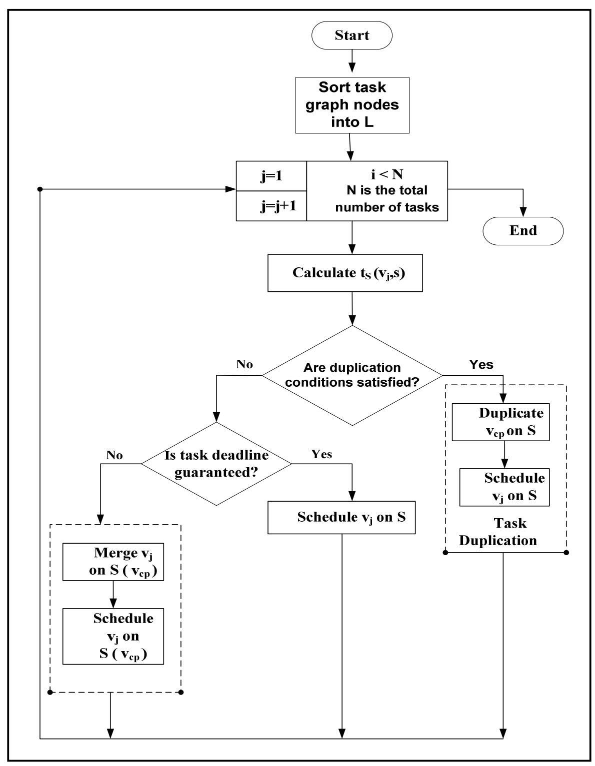

Fig. 6. Task scheduling flow chart

V. SIMULATION RESULTS

This section presents the results of the conducted experiments analyzing many aspects of the proposed scheduling scheme. The objective is to investigate the energy efficiency, and applications deadline guarantees of the proposed model compared to recently proposed models. For this purpose, an experimental evaluation on real world applications, along with randomly generated application graphs is carried out. The schedule length, the energy consumption, and the deadline missing ratio are observed. The schedule length is defined as the finish time of the exit task of an application. The energy consumption includes the communication and computation expenses of all sensors. The deadline missing ratio is defined as the number of schedules with schedule lengths larger than the application deadline. For the sake of comparison, the same parameters of real-life distributed visual surveillance, Gauss Jordan elimination and LU factorization models have been adopted. Table 1 summarizes these simulation parameters.

Table I: Simulation Parameters

Attribute	Value
Channel bandwidth	1Mb/s
Transmission range *r*	10 meters
E_{elec}	50 nJ/b
ε_{amp}	10 pJ/b/m^2
V_T	26 mV
C	0.67 nF
I_o	1.196 mA
n	21.26
K	239.28 MHz/V
c	0.5V

A. *Randomly generated application graphs*

In order to evaluate the effectiveness of the proposed scheduling mechanism, simulations were first conducted on randomly generated application graphs. The randomly generated application graphs were scheduled on randomly created multi-hop clusters. For the sake of comparison, this study uses the same related graph parameters.

Effect of number of tasks

In order to address the effect of varying the number of the application tasks on the total energy consumption and deadline missing ratio, experiments were conducted on three sets of randomly generated applications with 40, 45, 50 tasks. Figure 7 shows a comparison between the proposed scheduling mechanism, and the most related algorithms, MTMS, ETBA and DCA, in terms of energy consumption. It can be seen that as the number of tasks of the application increases, the energy consumption increases in both the related algorithms and proposed scheme. However, the proposed scheduling scheme shows lower energy consumption compared to them. On the other hand, figure 8 depicts the deadline missing ratio, it can be noticed that MTMS which is better than ETBA and DCA is dramatically affected by increasing the number of tasks while the proposed scheduling scheme shows better capacity to meet application deadline even in very strict ones.

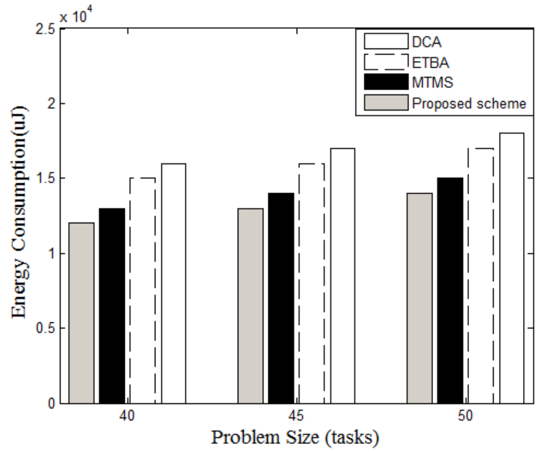

Fig. 7: Energy consumption versus number of tasks

Effect of communication load

In order to investigate the effect of varying the communication load on the proposed scheme performance, experiments conducted in [9] on randomly generated application graphs with 40 tasks with the same three different setting for application graphs are repeated. Communication load uniformly distributed in [600 bit, ±10 percent], [800 bit, ±10 percent], and [1,000 bit, ±10 percent] with fixed computation load equal to [300 KCC, ±10 percent] on the performance of the proposed scheduling scheme. As shown in figure 9, the performance of MTMS is highly affected by varying the communication load. As the communication load increases the deadline missing ratio of MTMS increases. Whereas, the proposed scheduling scheme is less likely to be affected by varying the communication load.

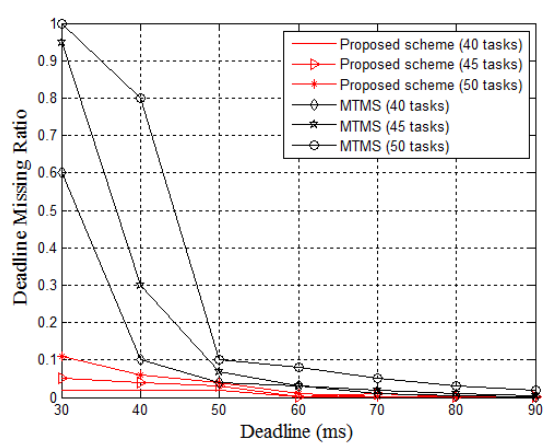

Fig. 8. Deadline missing ratio versus the deadline of different number of tasks

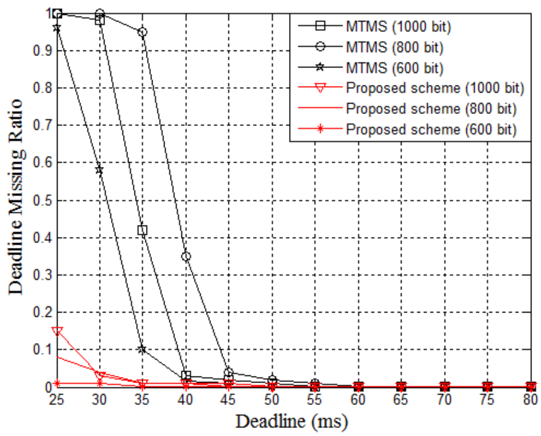

Fig. 9. Deadline missing ratio versus deadline for different communication load

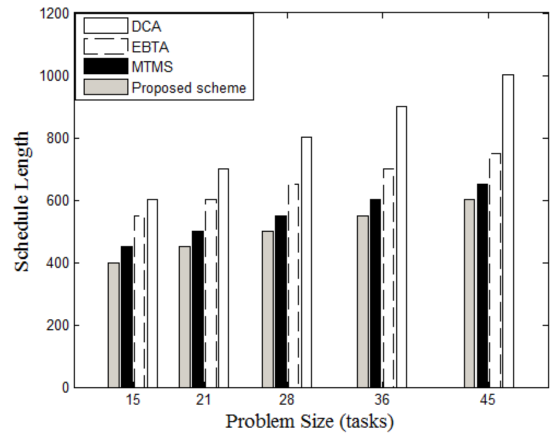

Fig. 10. Scheduling length for Gauss Jordan elimination problem versus number of tasks

B. Real world applications

In addition to randomly generated application graphs, this study also considered application graphs of three real world problems: Gauss Jordan elimination [22], LU factorization, [20], and Real-life distributed visual surveillance example [8]. For the experiments of Gauss Jordan elimination, figure 10 gives the schedule length of related schemes and the proposed scheduling scheme at various numbers of tasks. The smallest size graph in this experiment has 15 tasks and the largest one has 45 tasks. In all algorithms, the obtained schedule length increases as the number of tasks increases. However, in all cases the proposed scheduling scheme results in shorter schedule length. Figure 11 presents the schedule length for LU factorization. Different numbers of tasks are used in this experiment. It could be seen that the proposed scheduling algorithm outperforms in terms of the schedule length. Figure 12 presents the schedule length for Real-life distributed visual surveillance problem. Again, the proposed scheme outperforms the others. Regarding the schedule length, It obvious that the proposed scheduling results in the shortest schedule length among all other algorithms MTMS, EBTA, and DCA because the proposed algorithm mitigates channel contention through redundantly duplicating some of the graph tasks in which other tasks critical depend. Thus resulting in shorter schedule lengths which in turn enables the proposed scheduling scheme to satisfy very strict application deadlines.

In order to demonstrate the effectiveness of the proposed scheduling algorithm. Gauss Jordan elimination, LU factorization and Real-life distributed visual surveillance examples are again considered. Tables (3-5) summarize the achieved results. In these set of experiments, the performance of the proposed scheduling scheme is evaluated in terms of energy consumption and the maximum energy consumption per node. For energy consumption, the proposed schedule produces the smallest application energy consumption compared with MTMS, EBTA, and DCA. Finally, the proposed schedule has the smallest maximum energy consumption per node. Thus, employing our proposed scheduling algorithm yields in a fair energy consumption balance across the cluster sensor nodes.

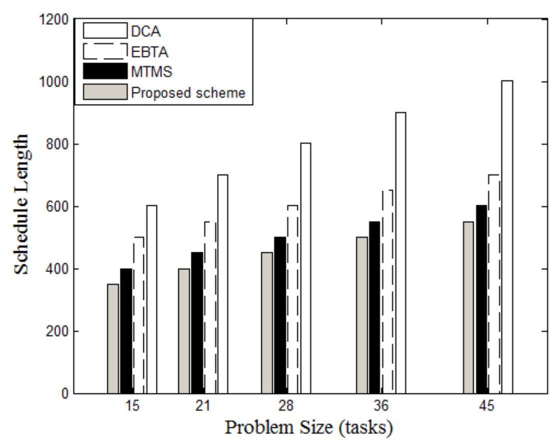

Fig. 11. Scheduling length for LU factorization problem versus number of tasks

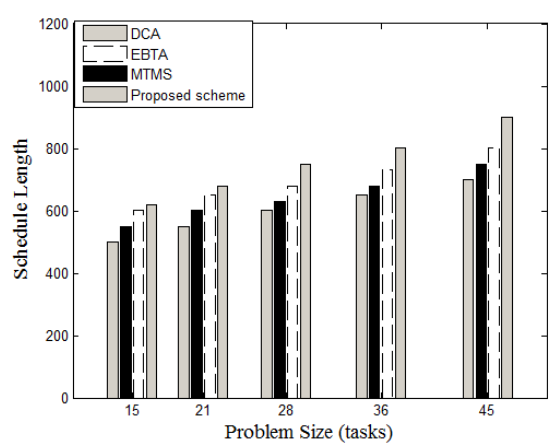

Fig. 12. Scheduling length for Real-life distributed visual surveillance problem versus number of tasks

Table II: Simulation results with the Visual Surveillance Example

Metrics	Proposed Scheme	MTMS	EBTA	DCA
Overall Energy Consumption(μJ)	1170	2194	2743	2238
Maximum Energy Consumption per node (μJ)	284	592	298	1139

Table III: Simulation results with LU factorization Example

Metrics	Proposed Scheme	MTMS	EBTA	DCA
Overall Energy Consumption(μJ)	1570	3213	4321	2983
Maximum Energy Consumption per node (μJ)	347	719	409	1845

Table IV: Simulation results with Gauss Jordan elimination Example

Metrics	Proposed Scheme	MTMS	EBTA	DCA
Overall Energy Consumption(μJ)	1823	2822	3542	3012
Maximum Energy Consumption per node (μJ)	276	432	350	2108

VI. CONCLUSIONS

This paper discussed the problem of allocating a set of real-time tasks with dependencies into heterogeneous sensor network. It presented an energy-efficient tasks scheduling scheme that minimizes the execution energy while meeting deadline. The proposed method adopted linear task mapping, augmented with task duplication and migration approach. it duplicated the critical predecessors only if the duplication can help in conserving energy, and advance the starting time of the succeeding tasks. Experimental results and comparisons conducted on both real–world and randomly generated application graphs, revealed that the proposed scheduling algorithm outperforms previous scheduling algorithms in terms of schedule length, energy consumption, and deadline missing ratio.

In our future work, recovering functionality from sensors failure will be handled. Furthermore, varying the network parameters will be addressed to study its effect on the performance of the overall system.

REFERENCES

[1] D. Ucin´ski, “Sensor network scheduling for identification of spatially distributed processes” J. Appl. Math. Computer. Sci. , vol. 22, No. 1, 2012, pp. 25–40.

[2] M. Junchao and L. Wei, “Interference-Free Wakeup Scheduling with Consecutive Constraints in Wireless Sensor Networks” International Journal of Distributed Sensor Networks, vol. 1, 2012, pp. 1-17.

[3] M. A. Ameen, S. M. Islam and K. Kwak, “Energy saving mechanisms for MAC protocols in wireless sensor networks,” International Journal of Distributed Sensor Networks, vol. 1, 2010, pp. 1-16.

[4] D. Dai, I. F. Akyildi, “A spatial correlation model visual information in wireless multimedia sensor networks”, IEEE Trans. On Multimedia, vol. 11(6), 2009, pp. 1148-1159.

[5] N. Dimokas, D. Katsaros and Y. Manolopoulos, “ Cooperative caching in wireless multimedia sensor networks”. Journal Of Mobile networks and applications, vol. 13(4): 2008, pp. 345-356.

[6] I. F. Akyilidiz, T. Melodia, and K. Chowdhury, “A survey on wireless multimedia sensor networks”. Journal of Computer Networks, vol. 51(4), 2007, pp. 921-960.

[7] X. Wang and S. Wang, “Collaborative signal processing for target tracking in distributed wireless sensor networks”. Journal of Parallel and Distributed computing, vol. 67, 2007, pp. 501-515.

[8] J. Yick, B. Mukherjee, and D. Ghosal, “Wireless sensor network survey”. Journal of Computer Networks, vol. 52, 2008, pp. 2292-2330.

[9] Y. Tian, and E. Ekici, “Cross-layer collaborative in-network processing in multihop wireless sensor networks”. IEEE Trans. on Parallel and Distributed Systems, Vol. 6(3), 2007, pp. 297-310.

[10] Y. Gu, Y. Tian and E. Ekici “Real-time multimedia processing in video sensor networks”. Journal of Image Communication, vol. 22, 2007, pp. 237-251.

[11] Y. Tian, E. Ekici and F. Ozguner, “Cluster-based information processing in wireless sensor networks: an energy-aware approach”. Journal of Wireless Communication and Mobile Computing, Vol. 7, 2007, pp. 894-907.

[12] S. Giannecchini., M. Caccamo and S. Shih, “Collaborative Resource Allocation in Wireless Sensor Networks”. Proc. Euromicro Conf. Real-Time Systems (ECRTS ’04), 2004, pp. 35-44.

[13] Y. Yu and V. Prasanna, “Energy-balanced task allocation for collaborative processing in wireless sensor networks”. ACM/Kluwer Journal of Mobile Networks and Applications, vol. 10(1), 2005, pp. 115-131.

[14] J. Zhu, J. Li and H. Gao, “Tasks allocation for real-time application in heterogeneous sensor networks for energy minimization”. Proc. Of 8th ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and parallel/Distributed Computing, 2007, pp. 1732-1738.

[15] R. Kumar, M. Wolenetz, B. Agarwalla and J. Shin “A framework for distributed data fusion”. Proc. Of ACM Conference on Embedded Networked Sensor Systems (SenSys’03), 2003, pp. 114-125.

[16] K. Phung, B. Lemmens, M. Mihaylov and T. Steenhaut, “Adaptive Learning Based Scheduling in Multichannel Protocol for Energy-Efficient Data-Gathering Wireless Sensor Networks”. Int. Journal of Distributed Sensor Networks, vol. 2013, 2013, pp. 1-11.

[17] M. Dai, C. Tang and C. Chuang, “An Energy-Aware Workload Dispatching Simulator for Heterogeneous Clusters”. Proceedings of the International MultiConference of Engineers and Computer Scientists 2013 vol. I, IMECS, Hong Kong 2013, pp. 374-379.

[18] N. Shigei, I. Fukuyama, H. Miyajima, and Y. Yudo, “ Battery-Aware Algorithms for Mobile Relay and Route Construction on Wireless Sensor Network”. IAENG International Journal of Computer Science, vol. 39:3, 2013, pp. 321-328.

[19] G. Lin and B. Xu, “Novel sensor scheduling and energy-efficient quantization for tracking target in wireless sensor networks”. Journal of Control Theory and Applications February 2013, vol. 11, Issue 1, 2013, pp 116-121.

[20] Y. Wang, W. Wang, X. Yang and W. Song, “Interference-aware joint routing and TDMA link scheduling for static wireless networks”. IEEE Trans. on Parallel and Distributed Systems, vol. 19(12), 2008, pp. 1709-1724.

[21] M. Conard, M. Marrakchi, Y. Robert and D. Trystram “ Parallel Gaussian elimination on an MIMD computer”. Parallel Comuputing, vol. 6, 1988, pp. 275–295.

[22] E. Shih, S. Cho, N. Ickes, R. Min, A. Sinha, A. Wang, and A.

Chandraksan, “Physical layer driven protocol and algorithm design for energy-efficient wireless sensor networks”. Proc. of ACM MobiCom’01, 2001, pp. 272-286.

Medhat H. A. Awadalla works at the Electrical and Computer Engineering Department, University of Sultan Qaboos, Oman, on leave from the Communications, Electronics and Computer Department, Faculty of Engineering, Helwan University, Egypt.

He obtained his PhD from university of Cardiff, UK, 2005. MSc and B.Sc. from Helwan university, Egypt. His research interest includes cloud computing, sensor networks, high performance computing and real time systems.