International Journal of Computer Engineering Research Vol. 2(1), pp. 8-18, February 2011
Available online at http://www.academicjournals.org/IJCER
ISSN 2141- 6494 ©2011 Academic Journals

Full Length Research Paper

Quality of service constrained task mapping and scheduling algorithm for wireless sensor networks

Medhat H. A. Awadalla[1]* and Rania R. Darwish[2]

[1]Department of Communication, Electronics and Computers, Faculty of Engineering, University of Helwan, Egypt.
[2]Department of Mechatronics, Faculty of Engineering, University of Helwan, Egypt.

Accepted 30 November, 2010

Efficient task mapping and scheduling is a crucial factor for achieving high performance in multimedia wireless sensor networks. This paper presents Quality of Service (QoS) - constrained task mapping and scheduling algorithm for multi-hop clustered wireless sensor networks. With the objective of meeting high performance and providing real-time guarantees, the algorithm simultaneously schedules the computation tasks and associated communication events of real time applications. The proposed scheduling algorithm exploits linear task clustering, augmented with task duplication and migration approach. Thus, reduces inter-task communication costs. Meanwhile, mitigates local communication overhead incurred due to communication medium contention. Experimental results and comparisons, based on both randomly generated application graphs, as well as graphs of some real-world applications, demonstrate that the proposed task mapping and scheduling scheme significantly surpasses previous approaches in terms of both quality and cost of schedules, which are mainly presented with deadline missing ratio, schedule length, and total application energy consumption.

Key words: Multimedia wireless sensor networks, task mapping and scheduling, linear clustering, real time applications.

INTRODUCTION

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 apart from boosting the existing applications of WSNs 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 (Dai and Akyildiz, 2009; Dimokas et al., 2008; Campbell et al., 2005; Holman, 2003). 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 (Akyilidiz, 2007). Real-time, collaborative in-network processing gains recognition as a viable solution for balancing the performance and consumption in MWSN (Wang and Wang, 2007). 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 (Akyilidiz, 2007; Yick et al., 2008).

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 (Tian and Ekici, 2007). Therefore, task mapping and scheduling plays an essential role in collaborative in-network processing by solving the following problems. First, assigning tasks into sensors. Second, determining the execution sequence of tasks on sensors. Finally, scheduling communication transactions

*Corresponding author. E-mail: awadalla_medhat@yahoo.co.uk.

between sensor nodes (Gu et al., 2007; Tian et al., 2007).

This paper proposes a Quality of Service (QoS)-constrained task mapping and scheduling algorithm resilient for real-time multimedia application in MWSN. The proposed approach simultaneously exploits linear clustering algorithm augmented with task duplication and migration approach. The proposed approach aimed at increasing network lifetime, meanwhile guarantee meeting application deadline. Furthermore, task scheduling and communication scheduling in the proposed approach are carried out in parallel. Thus, resulting in a realistic schedule due to the incorporation of communication contention awareness in the task scheduling, which is critical in real time multimedia applications.

The rest of the paper is organized as follows: Discussion of most related work; details of underlying system architecture; Introduction of the proposed task mapping and scheduling algorithm; the performance evaluation results; and finally, conclusion.

BACKGROUND

Collaborative in-network processing has been widely pursued by the research community in order to achieve energy saving and network scalability objectives. Giannecchini et al. (2004) 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, CoRAI assumes that the tasks are already assigned to sensors without addressing the task mapping problem. Furthermore, energy consumption is not explicitly discussed in (Wang and Chandrakasan, 2002). Wang et al. (2002) 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 et al. (2005) proposed an Energy-balance Task Allocation (EbTA) onto a single-hop cluster of homogenous sensor nodes connected with multiple wireless channels. In this work, communication over multiple wireless channels are 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 Yu and Prasanna (2005) 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 can not satisfy the requirement of multiple wireless channels assigned in every cluster, especially in densely deployed networks. Zhu et al. (2007) 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. (2003) presented a data fusion task mapping mechanism for wireless sensor network. The proposed mechanism in Kumar et al. (2003) comprises data fusion API and a distributed algorithm for energy aware role assignment. The data fusion API enables an application to be specified as a coarse-grained dataflow graph. While, the role assignment algorithm maps the graph onto the network, and optimally adapts the mapping at run-time using role migration. Zhu et al. (2007) and Kumar et al. (2003) assumed an existing underlying communication model. Tian et al. (2007) 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. Tian et al. (2007) 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.

Apart from all these efforts, this work is motivated for addressing all the above mentioned drawbacks and developing a QoS constrained task Happing and sche@uling algorithm for multi-hop cHustered multime@ia wireless senRor networks. Th@ main idea behiJd the proposed algorithm is to group tasks that are heavily communicate with each other to be processed on the same sensor. Thereby, reducing 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. 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, guarantee meeting very strict application deadlines.

PRELIMINARIES

Network and interference model

The proposed task mapping and scheduling strategy targets multi-hop cluster-based network architectures, the following discusses the assumed network and the interference model.

i.) All sensor nodes are grouped into k-hop clusters, where k is the hop count of the longest path connecting any two nodes.
ii.) 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.
iii.) Cluster heads are responsible for creating the applications' schedules within the clusters.
iv.) Location information is locally available within clusters.
v.) 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.
vi.) Inter-cluster communication is assumed to be isolated from other clusters through time division or multiple wireless channels assignment mechanisms.

INTERFERENCE MODEL

This works 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 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, 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 RTS/CTS interference model is adopted through out 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 four nodes, that is., $s_i \neq s_j \neq s_p \neq s_q$, and s_i and s_j are not in the interference ranges of s_p and s_q, and vice versa [16].

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 (Yu and Prasanna, 2005; Zhu et al., 2007).

ENERGY CONSUMPTION MODEL

The energy consumption of transmitting and receiving *l* bit data over a distance *d* that is less than a threshold d_0 are defined as E_{tx} (*l,d*) and E_{rx}(*l*), respectively (Tian and Ekici, 2007).

$$E_{tx}(l,d) = E_{elec} \cdot l + \varepsilon_{amp} \cdot l \cdot d^2 \qquad (1)$$
$$E_{rx}(l) = E_{elec} \cdot l$$

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 (1) and (2) respectively. Also the energy consumption of executing *N* clock cycles with CPU clock frequency *f* is given as (Tian and Ekici, 2007),

$$E_{comp}(V_{dd}, f) = NCV_{dd}^2 + V_{dd}(I_o e^{\frac{V_{dd}}{nV_T}})(\frac{N}{f}), \qquad (2)$$
$$f \cong K(V_{dd} - c)$$

Where V_T is the thermal voltage, V_{dd} is the supply voltage, and C, I_o, n, K, c are processor-dependent parameters (Wang, 2008; Shih et al., 2001).

The proposed task mapping and scheduling algorithm

This section presents the proposed task mapping and allocation algorithm. The proposed algorithm comprises of two mechanisms. Linear task clustering algorithm, and sensor assignment mechanism based on a task duplication and migration scheme.

```
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
```

Figure 1. Linear clustering sequence.

TASK CLUSTERING

This work adopted linear clustering algorithm (Kwok and Ahmed, 1999) for mapping tasks onto distinct clusters. The rational behind this mapping is to reduce the overall energy consumption, as well as the schedule length of the application, since communication between tasks within the same cluster costs negligible time and energy.

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 assign 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 in Figure 1.

SENSOR ASSIGNMENT MECHANISM

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

i.) Map the obtained μ task clusters into the p physical sensors.
ii.) Determine the execution sequence of the computation tasks on sensors.
iii.) 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 worth 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 which can be expressed as (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, C_i is the energy needed to execute task cluster i, and E_k is the remaining energy of sensor k.

Figure 2 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 assign to it. Then the cluster would be assigned to the sensor node that gives the minimal normalized load.

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 3 and 4 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 constrain. Without any duplication, the algorithm first attempts to schedule the previous phase. Obviously, to calculate the task starting time on its assigned sensor ts(v_j,S), all the receiving communication transactions from v_j parents should be

```
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
```

Figure 2. Cluster mapping sequence.

```
1.  Traverse the application graph V downwards and compute Latest Finishing Time (LFT) for
    every task.
2.  Sort tasks v ∈ V into list L according to precedence constrains.
3.  For every vj ∈ V DO
// Calculate the Earliest Starting Time of vj on its assigned sensor s ts(vj, s).
4.  For each vi ∈ pred(vj) DO
5.  IF SEN(vi) ≠ s THEN
6   Determine route M=<m1,m2,…….mn> from SEN(vi) to s
7.  Process links from m1 to mn and assign to each mk the earliest free interval on
          the communication channel not causing any interference.

8.  ENDIF
1.  ENDFOR
2.  Calculate ts(vj, s)
//  Check the duplication condition
3.  If duplication condition is satisfied THEN
4.          Duplicate vcp on s
5.          Schedule vj on s
6.          ELSE
7.               IF tf(vj, s) < LFT(vj) THEN
8.               Schedule vj on s
9.             Else
10.               Migrate vj to s(vcp)
11.               Schedule vj on s(vcp)
12.              ENDIF
13.     ENDIF
14. ENDFOR
```

Duplication Conditions:

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

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

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

Figure 3. Task scheduling scheme.

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 advancing the task starting $ts(v_j)$ time, reducing the consumed energy, and meanwhile preserves the task deadline, this phase is accepted. Otherwise, it is rejected.

In some cases, the duplication mechanism fails at satisfying the task deadline constrain. In such cases, the algorithm employs a global migration process for the task, where the task under consideration began to be migrated to other sensor. To reduce this migration impact on the

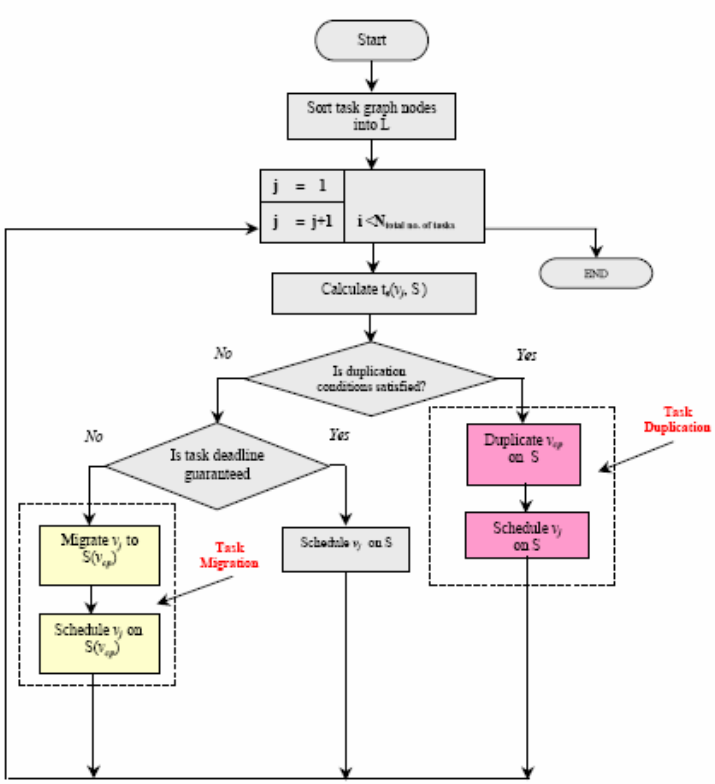

Figure 4. Task scheduling flow chart.

energy increase, the destination sensor is selected as the sensor that holds its critical parent.

SCHEDULING COMMUNICATION BETWEEN SENSOR NODES UNDER THE RTS/CTS MODEL

In order to satisfy dependencies between tasks, scheduling a task should include scheduling all its entering edges on the shared communication wireless channel. Figure 3, steps 4 - 9 stand for scheduling communication scheduler mechanism. As previously mentioned, this work concerned with multi-hop clustered wireless sensor networks. So, the sender and the receiver of a communication transaction may be one or more hops away, moreover, multiple communication transmissions could occur simultaneously. Thus, First a routing algorithm should be employed in order to determine the appropriate route $<m_1, m_2,\ldots\ldots,m_n>$ for each entering edge $e_{ij} \in$ pred (v_j) of task v_j. The basic idea here is after determining the appropriate route for all entering edges for the considered task, all these routes are sorted according to their length and the communication volume. Then, every link m_{xy} of the generated path is scheduled on the earliest free interval on the communication channel. Here, the adopted communication scheduler is interference-aware mechanism, in which scheduling a transmission between sensor x and sensor y on the interval [A, B] is valid, if and only if it will not result in a collision at either node x or node y (or any other node).

Moreover, the proposed communication scheduler makes use of the overhearing, which is a unique characteristic of wireless communication. That is, when a communication transaction generated from a certain task requested by several destinations. The sender only sends one transmission to one destination, and all other transaction destinations lying in its transmission range will receive it. Thus, prune out superfluous transmission from the source, as multiple destination can hear the same transmission with only one transmission from the source. Thus yields in shortening the communication latency, and results in significant reduction in the consumed energy.

SIMULATION RESULTS

This section presents the results of the conduced experiments analyzing many aspects of the proposed scheduling. 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.

In the evaluation experiments, the schedule length, the energy consumption and the deadline missing ratio are

Table 1. Simulation parameters.

Attribute	Value
Channel bandwidth	1Mb/s
Transmission range *r*	10 m
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

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 as in MTMS (Tian and Ekici, 2007) have been adopted. Table 1 summarizes these simulation parameters.

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 multihop clusters. For the sake of parameters as that used while evaluating MTMS (Tian and Ekici, 2007).

EFFECT OF NUMBER OF TASKS

In order to investigate 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 5 shows a comparison between the proposed scheduling mechanism, and MTMS in terms of energy consumption. It can be seen that, as the number of tasks of the application increases, the energy consumption increase in both the MTMS and the proposed scheduling. Whereas, the proposed scheduling scheme shows lower energy consumption compared to that of MTMS. On the other hand, Figure 6 depicts the deadline missing ratio of the proposed scheduling scheme and MTMS with respect to different deadlines for different number of tasks. It can be seen, that MTMS 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. Thus, the proposed scheduling scheme shows better scalability than MTMS in terms of energy consumption and application deadline.

EFFECT OF COMMUNICATION LOAD

In order to investigate the effect of varying the communication load on the proposed scheme performance, experiments were conducted on randomly generated application graphs with 40 tasks. Three different setting for application graphs were considered. Communication load uniformly distributed in [600 bit, ±10%], [800 bit, ±10%], [1,000 bit, ±10%] with fixed computation load equal to [300 KCC, ±10%] on the performance of the proposed scheduling scheme. As shown in Figure 7, 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.

REAL WORLD APPLICATIONS

In addition to randomly generated application graphs, this study also has considered application graphs of three real world problems: Gauss Jordan elimination (Jin et al., 2008), LU factorization, (Jin et al., 2008) and Real-life distributed visual surveillance example (Tian and Ekici, 2007).

For the experiments of Gauss Jordan elimination, Figure 8 gives the schedule length of both MTMS 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 both 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. As in Gauss Jordan elimination, Figure 9 presents the schedule length for LU factorization of both the proposed scheduling mechanism and MTMS. Different number of tasks are used in this experiment. The number of tasks varies between 14 tasks and 43 tasks. It could be seen that the proposed scheduling outperforms MTMS in terms of the schedule length.

Also, in order to demonstrate the effectiveness of the proposed scheduling algorithm, the same distributed visual surveillance example used in Tian and Ekici (2007) is considered. In this experiment, the performance of the proposed scheduling is compared to MTMS (Tian and Ekici, 2007), DCA (Giannecchini et al., 2004) and EbTA (Wang and Chandrakasan, 2002). Table 2 summarizes this comparison. In this set of experiments the performance of the proposed scheduling is evaluated in terms of schedule length, energy consumption, and the

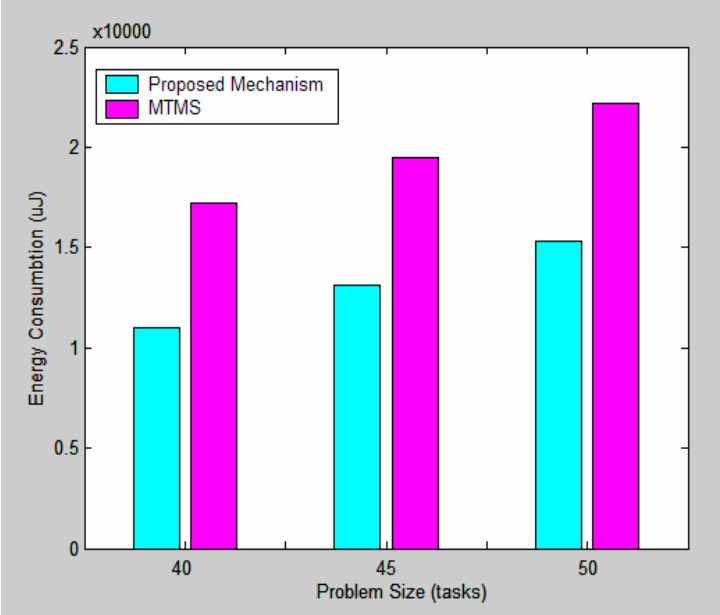

Figure 5. Energy consumption versus number of tasks.

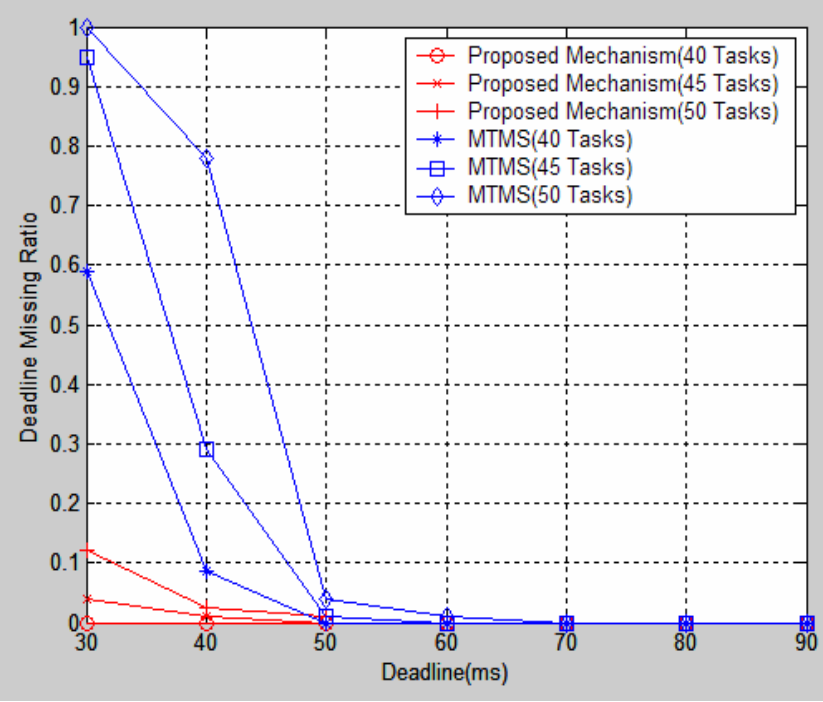

Figure 6. Deadline missing ratio versus the deadline of different number of tasks.

maximum energy consumption per-node. Regarding the schedule length, it could be seen that the proposed scheduling results in the shortest schedule length among all other algorithms MTMS, EbTA, and DCA. This is because the proposed scheduling 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? For energy consumption, the proposed schedule also produces the smallest application energy consumption between MTMS, EbTA, and DCA.

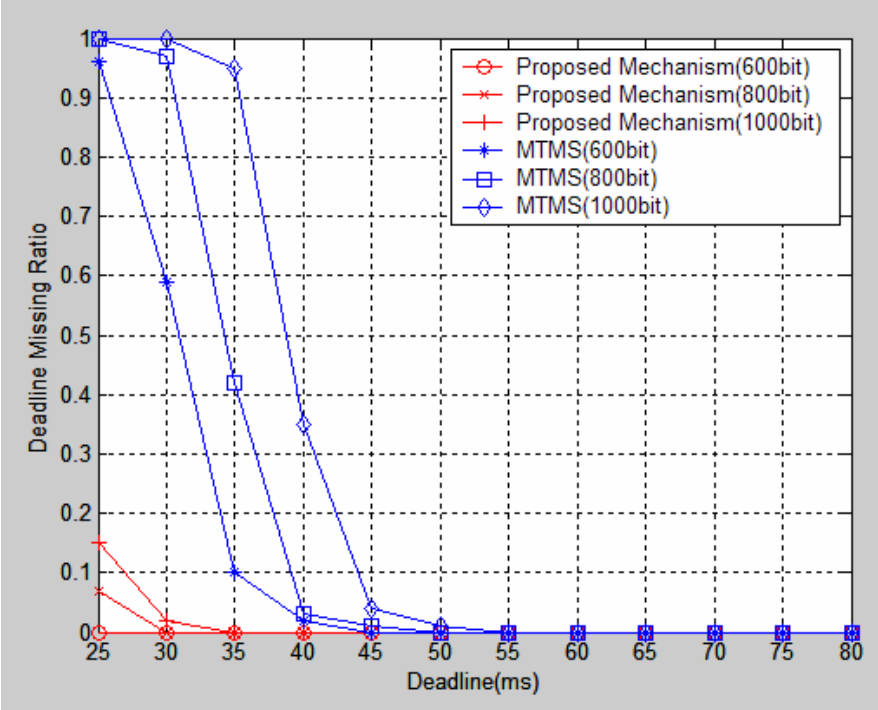

Figure 7. Deadline missing ratio versus deadline for different communication load.

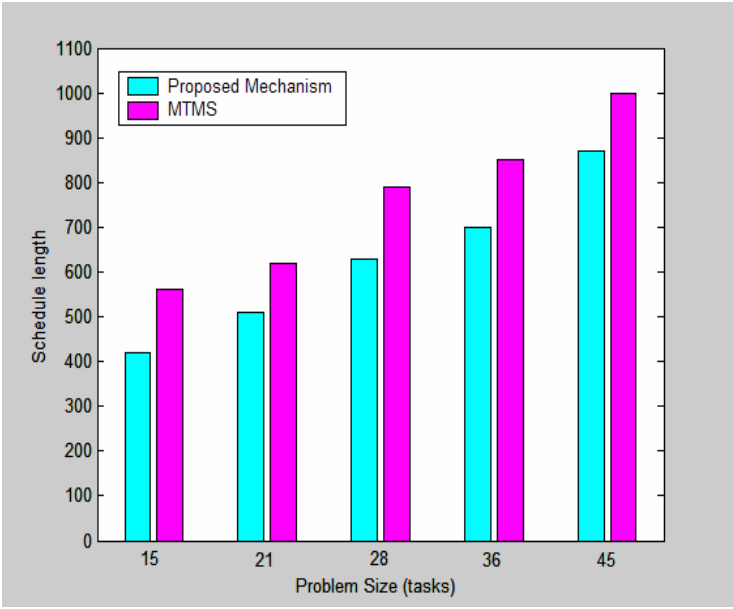

Figure 8. Scheduling length for Gauss Jordan elimination problem versus number of tasks.

Finally, the proposed scheduling results in 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.

Conclusions

This paper proposes Quality of Service (QoS)-constrained task mapping and scheduling algorithm for multimedia wireless sensor networks. The proposed algorithm adopted

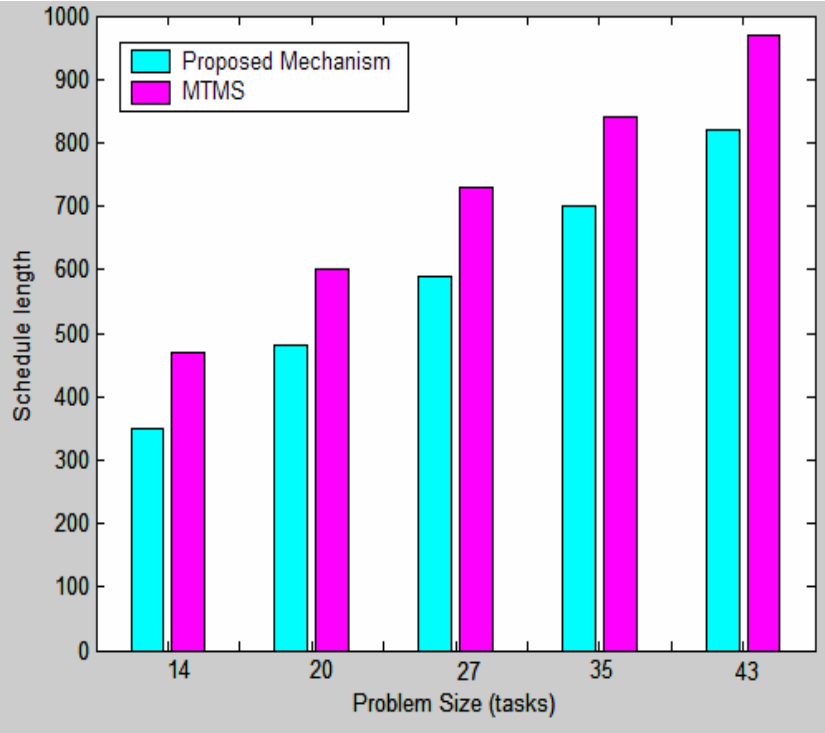

Figure 9. Scheduling length for LU factorization problem versus number of tasks.

Table 2. Simulation with the visual surveillance example.

Metrics	Proposed scheme	MTMS	EBTA	DCA
Schedule Length(ms)	2.11	3.00	3.00	5.64
Overall Energy Consumption(μJ)	1170	2194	2743	2238
Maximum Energy Consumption per node (μJ)	284	592	298	1139

linear task mapping, augmented with task duplication and migration approach. At the same time, the algorithm also takes into consideration the exact communication delay by scheduling communication transactions in parallel. The proposed algorithm judiciously duplicated the critical predecessors only if the duplication can help in conserving energy, and advancing the starting time of the succeeding tasks. Experimental results and comparisons conducted on both real–world application graphs, 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. Also, since in large-scale networks cluster heads are not in direct range of the sink, thus, our future work will investigate developing non-interfering inter-cluster multi-hop routing algorithm. Furthermore, varying the network parameters will be addressed to study its effect on the performance of the overall system.

REFERENCES

Akyilidiz IF, Melodia T, Chowdhury K (2007). “A survey on wireless multimedia sensor networks”. J. Comput. Netw., 51(4): 921-960.

Campbell J, Gibbons P, Nath S, Pillai P, Seshan S, Sukthankar R (2005). “IrisNet: an internet-scale architecture for multimedia sensors,” Proc. Of the ACM Multimedia Conference, pp. 1230-1236.

Dai D, Akyildiz IF (2009). “A spatial correlation model visual information in wireless multimedia sensor networks”. IEEE Trans. Multimed., 11(6): 1148-1159.

Dimokas N, Katsaros D, Manolopoulos Y (2008). “Cooperative caching in wireless multimedia sensor networks”. J. Mob. Netw. Appl., 13(4): 345-356.

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

Gu Y, Tian Y, Ekici E (2007). “Real-time multimedia processing in video sensor networks”. J. Image Commun., 22: 237-251.

Holman R, Stanely J, Ozkan T (2003). “Applying video sensor networks to near shore environmental monitoring”. IEEE Trans. Distributed Comput., 2(4): 765-779.

Jin S, Schiavone G, Turgut D (2008). “A performance study of multiprocessor task scheduling algorithms”. J. Supercomput., (48): 77-97.

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

Kwok Y, Ahmed I (1999). “Static scheduling algorithms for allocating directed task graphs to multiprocessors”. ACM Comput. Surv. (CSUR), 31(4): 406- 471.

Shih EC, Ickes N, Min R, Sinha A, Wang A, Chandraksan A (2001). “Physical layer driven protocol and algorithm design for energy-efficient wireless sensor networks”, Proc. of ACM MobiCom’01, pp. 272-286.

Tian Y, Ekici E (2007). “Cross-layer collaborative in-network processing in multihop wireless sensor networks”. IEEE Trans. Parallel and Distributed Syst., 6(3):297-310.

Tian Y, Ekici E, Ozguner F (2007). “Cluster-based information processing in wireless sensor networks: an energy-aware approach,” J. Wireless Commun. Mob. Comput., (7): 894-907.

Wang A, Chandrakasan A (2002). “Energy-efficient DSP’s for wireless sensor networks”. IEEE Signal Process. Mag., pp. 68-78.

Wang A, Chandrakasan A (2002). “Energy-Efficient DSPs for Wireless Sensor Networks,” IEEE Signal Process. Mag., pp. 68-78.

Wang X, Wang S (2007). “Collaborative signal processing for target tracking in distributed wireless sensor networks. J. Parallel and Distributed Comput., (67):501-515.

Wang Y. Wang W. Yang X, Song W (2008). “Interference-aware joint routing and TDMA link scheduling for static wireless networks”. IEEE Trans. Parallel and Distributed Syst., 19(12):1709-1724.

Yick J, Mukherjee B, Ghosal D (2008). “Wireless sensor network survey.” J.Comp. Netw., (52): 2292-2330.

Yu Y, Prasanna V (2005). “Energy-balanced task allocation for collaborative processing in wireless sensor networks.” ACM/Kluwer J. Mob. Netw. Appl., 10(1-2): 115-131.

Zhu J, Li J, Gao H (2007). “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.