

# TSLBS: A time-sensitive and load balanced scheduling approach to wireless sensor actor networks

Morteza Okhovvat and Mohammad Reza Kangavari\*

*School of Computer Engineering, Iran University of Science and Technology, Tehran, Iran*  
E-mail: [morteza.okhovvat@gmail.com](mailto:morteza.okhovvat@gmail.com)

Existing works on scheduling in Wireless Sensor Actor Networks (WSANs) are mostly concerned with energy savings and ignore time constraints and thus increase the make-span of the network. Moreover, these algorithms usually do not consider balance of workloads on the actor nodes and hence, sometimes some of the actors are busy when some others are idle. These problem causes the actors are not utilized properly and the actors' lifetime is reduced. In this paper we take both time awareness and balance of workloads on the actor in WSANs into account and propose a convex optimization model (TAMMs) to minimize make-span. We also propose a protocol called LIBP to improve load balancing that allocates tasks to actors according to their measured capabilities in such a way to enhance balances of workloads on the actors. Finally, by combination of TAMMs and LIBP, a time-sensitive and load balanced scheduling approach (TSLBS) is proposed. TSLBS considers both local and global tasks and the distribution requirements of WSANs (i.e. WSANs with hybrid architecture). The results of simulations on typical scenarios shows that TSLBs is more efficient in terms of both the make-span and load balancing compared to stochastic task scheduling algorithm (STSA). We also show that TSLBs performs significantly better than STSA in terms of actor's lifetime.

Keywords: Load balancing, make-span, actors' lifetime, task scheduling, wireless sensor actor network

## 1. INTRODUCTION

Advances in smart micro-devices and wireless communications provided the great opportunity to tackle an actual problem: sensing, monitoring and remote control of complex and harsh environments [1, 2]. Development of sensor networks alongside the need of effectively reactions to environmental events lead to the emergence of Wireless Sensor Actor Networks (WSANs).

A typical WSAN is comprised of a group of densely spread sensor nodes along with a group of sparsely spread actor nodes connected via wireless links. Sensors are responsible for collecting environmental information and actors make appropriate actions on the environment based on the sensory information. An actor can act on the environment by means of one or several actuators and as a network entity, it is able to perform networking-related functionalities, i.e., receive, transmit, process, and relay data. Sensor nodes are able to sense, process and communicate information about a wide diversity of physical phenom-

ena in a broad spectrum of applications ranging from wildlife and habitat monitoring to health care or battle field surveillance wherein reactions to the events should be done quickly [3, 4]. Hence, to make effective use of WSANs capabilities, employing suitable time-sensitive scheduling algorithms is necessary. Furthermore, balance of workloads on actors should be considered by the scheduling algorithms as a critically desired parameter in WSANs to prevent network partitioning. Because ignore of load balancing in a scheduling may cause some actors to be busy while some others are idle and hence, the busy actors will lose their energies and die soon.

Depending on the strategies pursued by the actors to send commands [5-7], there are principally three different architectures: semi-automated, hybrid, and automated. In automated architecture, the network goes in a fully distributed way with the actors that autonomously accept and perform the proper actions upon receiving sensory information. In semi-automated architecture, sensors collect and transmit environmental information to a singleton network sink, and the sink determines the proper

\*Corresponding Author. E-mail: [kangavari@iust.ac.ir](mailto:kangavari@iust.ac.ir)

actions that actors should perform in response and allocates these actions (tasks) to proper actors. In hybrid architecture [9, 10] sensors may transmit sensing data to the sink or to actors in a single hop or multiple hops. Hybrid architecture (Figure 1) deals with two groups of tasks, global tasks and local tasks. Global tasks are determined by the sink, but local tasks are determined by the actors. These tasks are figured out based on the gathered information by the sensors and actors are the responsible nodes to perform the tasks.

However, how to allocate tasks to available actors is a critical issue influencing make-span, workloads of actors and hence, energy consumptions and network lifetime. To properly distribute tasks among the actors, it is absolutely necessary to consider the architecture of WSN [11, 12].

There are a relatively great number of existing task scheduling algorithms for distributed systems in general that try to reduce the make-span of the system [13, 14]. These are however not applicable in for WSN task scheduling since they do not consider the constraints of WSNs. In addition, there are also some scheduling algorithms for distributed systems that try to achieve their load balancing objective and improving the lifetime of the system, but do not consider time constraints of the system [15, 16]. It is worthy to note that although consideration of load balancing may lead to reduction in make-span of the system, but in large scale distributed systems like WSNs, the load balancing objective alone may lead to poor make-span. Accordingly, there is a need for superior task scheduling algorithms that consider both the load balancing objective and the time constraints into account in order to shorter the make-span and upper the load balancing objective. However, to minimize make-span, we proposed a convex optimization model called *TAMMs* based on queuing theory. We also proposed a protocol called *LIBP* to improve load balancing and hence, enlarging network life time. At last we proposed an algorithm called *TSLBS*, with the above two mentioned objectives for allocation of tasks to actors. The algorithm is modeled with queuing networks and simulated for validation.

The rest of paper is organized as follows. Section 2 presents notable related works. Section 3 describes our assumptions. Section 4 presents *TSLBS*. Section 5 presents the experimental results and section 6 concludes the paper.

## 2. RELATED WORKS

Due to the challenging features and constraints of WSNs, such as their dynamicity and resource constraints, the existing general-purpose scheduling algorithms are usually inappropriate to WSNs. There are however various research works on scheduling algorithms for wireless sensor actor networks. The applicability of notable scheduling algorithms is discussed here.

Farias et al. [17] presented an algorithm to schedule tasks in WSNs to enhance the energy efficiency and hence, enlarging the network lifetime. To achieve this goal, the characteristics of applications is tried to be utilized with common tasks and unnecessary repeat of tasks is avoided. However, their algorithm may raise the total remaining energies of actuators, but neither service reliability nor make-span has been considered by their approach.

Sharifi and Okhovvat [5] presented a starvation free, time and energy aware scheduling algorithm for WSNs. Their algorithm supports concurrent executions of any mix of small and large tasks and yet precludes likely starvation of tasks. Decreasing the tasks' completion time and enhancing the remaining energies of actors were the dual objectives of their algorithm. The main disadvantage of their algorithm is that it does not guarantee the execution deadline for applications.

Shu et al. [13] proposed a scheduling algorithm to maximize the network lifetime while making firm sensing guarantees in the WSN. In order to validate their algorithm, they carried out an in-depth evaluation of its performance via large-scale simulations and reported an average of 39.2% enhancement of network lifetime over the baseline method. The main disadvantage of their work is that time as an important parameter in WSNs was not considered in their algorithm.

Yang and Lee [18] presented a fault-tolerant assignment method to tolerate failures in more than one actuator. Their method operates based on the redundancy to tolerate failures. The main drawback of their method is that time constraint of applications was not considered in their method.

Okhovvat and Kangavari [19] proposed a task allocation model to maximize total utilization of actor nodes in WSNs. They showed the task allocation policy applied by the sink with a Generalized Stochastic Petri Net (GSPN) model. Then, to maximize actors' utilization, the arrival rate of tasks for each actor is determined based on the capability of that actor. Steady state analyzing of the GSPN model and with the computation of the task dispatching weight at each actor, task arrival rates is calculated. All the above research works try to schedule tasks in sensor networks while do not consider both local and global tasks since they match one of the semi-automated or full automated architecture. However, in this paper we propose a time-sensitive and load balanced approach to schedule tasks in order to provide an appropriate tradeoff between make-span and balance of workloads on the actors in WSNs with hybrid architecture.

## 3. ASSUMPTIONS

Task scheduling is defined as the allocation of one or more time slots to one or more resources [20, 21]. The scheduling problem in WSNs amounts to the mapping of a set of tasks to a set of actors based on determined QoS parameters such as make-span minimization, expanding actors' lifetime, residual energy maximization and etc. Since the scheduling is an NP-complete problem [13, 17, 20], consideration of multi QoSs to get an optimized scheduling scheme is however impractical [7, 15].

We study a WSN with hybrid architecture (Figure 2) wherein  $m$  actors  $A_j$  ( $j = 1, \dots, m$ ) would carry out  $n$  global tasks  $T_i$  ( $i = 1, \dots, n$ ) and  $n'$  local tasks  $T'_i$  ( $i = 1, \dots, n'$ ). The execution time of task  $i$  by actor  $j$ ,  $E_{ij}$ , is defined as the total expected time taken by actor  $j$  to execute the task with no load at the time of assignment. The expected completion time of task  $i$  by actor  $j$  ( $C_{ij}$ ) is defined as the time interval in which  $A_j$  finishes task  $i$  after completing any remaining previous assigned tasks.

Let  $C_i$  denotes the completion time of task  $i$  which is equal to  $C_{ij}$  where actor  $j$  is actually assigned to perform  $T_i$ . The overall completion time of all tasks in the network is called make-span.

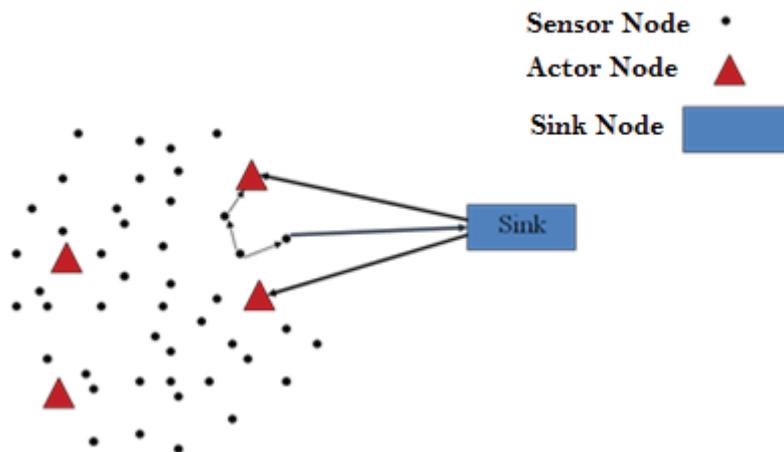


Figure 1 Hybrid architecture of WSAN.

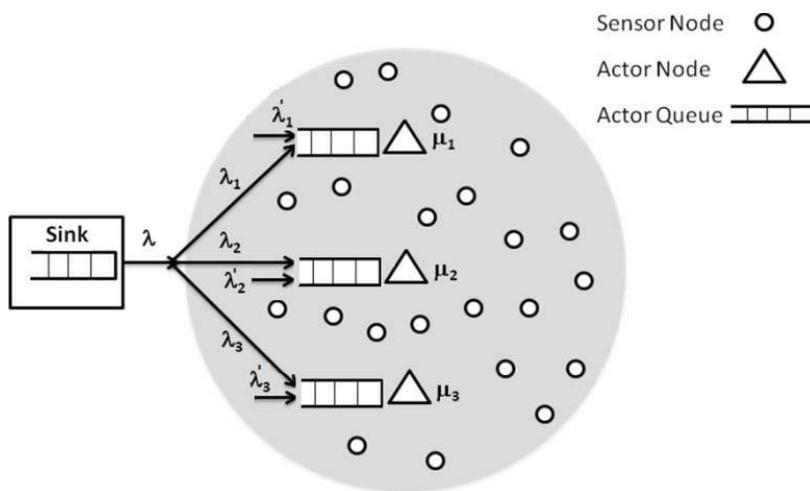


Figure 2 A queuing network model of WSAN.

Accordingly, the make-span of a WSAN with  $m$  actors for the set of  $i$  tasks can be computed by Eq. (1).

$$Make - span = \max_i \{C_{ij}\} \quad (1)$$

We defined the workload of each actor as the time to be spent by the actor to perform all tasks assigned to that actor. We assumed that tasks are independent and sensors gather information from the environment and determine the related local tasks or send them to the sink to determine the related global tasks.

We also assumed that the sink and also each actor have their own independent queue. Global tasks are initially inserted in the queue of the sink and later put in the queue of the actor chosen by the sink to perform the task. Local tasks are assigned to closest actor/actors directly by the sensors, but global tasks are assigned to the proper actors by the sink. It is assumed that the process of assigning tasks to actors follows a Poisson distribution. Moreover, we have assumed that all tasks are non-preemptive and the generation rate of local tasks ( $\lambda'$ ) and also the generation rate of global tasks ( $\lambda$ ) follows an exponential distribution.

#### 4. PROPOSED APPROACH

The main goals in the proposed approach is to reduce make-span, to improve the balances of workloads on actors and hence, increasing the network lifetime. We noticed the inherent unpredictability of WSANs in the proposed approach using the queuing theory. According to our assumptions, sensory information of a local task is directly sent to the closest actor and the best allocation rate of global tasks to the actors is determined. To minimize the make-span, a convex optimization model (TAMMs) based on queuing network has been proposed In order to improve load balancing we proposed a protocol (LBIP) that determines the allocation rates of tasks for each actor based on its capability in performing tasks. Finally, we proposed our time-sensitive and load balanced scheduling (TSLBS) that tries to makes an appropriate trade-off between minimization of make-span and load balancing enhancement in a WSAN.

#### 4.1 Task Assignment Model to Minimize Make-span (TAMMs)

The aim of TAMMs is to achieve the minimum make-span within a WSA. To attain this purpose, tasks arrival rates at each of the actors should be approximated properly. Since local tasks are directly assigned to closest actors by the sensor nodes, the allocation rates of global tasks should be tune appropriately. As mentioned before, it is assumed that all allocated tasks arrived at the actor,  $A_i$ , follows a Poisson distribution. To carry out steady state analysis on the actors, the sum of the local and global tasks arrival rates in each of the actors have to be less than the corresponding actor's service rate ( $\mu_i$ ). Hence, Eq. (2) must be satisfied for each actor  $i$ :

$$0 \leq \lambda'_i + \lambda_i < \mu_i \quad \forall i; 1 \leq i \leq n \quad (2)$$

Given these assumptions, we model each actor by a  $M/M/1$  queuing system model wherein tasks arrive at the actor  $A_i$  with  $\lambda'_i + \lambda_i$  and are performed with the rate  $\mu_i$ . Applying the steady state analysis to the continuous time Markov chain (CTMC) reached from  $M/M/1$  queues, average completion time of the actors could be obtained [22, 23]. Eq. (3) calculates the average completion time of the actor  $A_i$ . Proof of Eq. (3) can be found in [24].

$$W = \frac{1}{\mu_i - (\lambda'_i + \lambda_i)} \quad (3)$$

Since the goal of this phase is to minimize the make-span of the WSA, using Eqs. (1)- (3), this goal can be formulated as (4).

Goal:

$$\begin{aligned} & \text{Min} \left\{ \text{Max} \left( \frac{1}{\mu_1 - (\lambda'_1 + \lambda_1)}, \frac{1}{\mu_2 - (\lambda'_2 + \lambda_2)}, \right. \right. \\ & \left. \left. \frac{1}{\mu_3 - (\lambda'_3 + \lambda_3)}, \dots, \frac{1}{\mu_n - (\lambda'_n + \lambda_n)} \right) \right\} \\ & \lambda = \sum_{i=1}^n \lambda_i \\ & \lambda' = \sum_{i=1}^n \lambda'_i \\ & 0 \leq \lambda'_i + \lambda_i < \mu_i, \forall i; 1 \leq i \leq n \\ & 0 < \lambda, \lambda'_i, \mu_i : \text{Constant} \end{aligned} \quad (4)$$

To solve (4), the problem is modeled in GAMS software using DICOPT solver [24], which is based on the extensions of the external estimation algorithm for the equality relaxation policy. Applying the resultant tasks arrival rates to the actors, the make-span of the WSA could be minimized.

#### 4.2 Load Balancing Improvement Protocol (LBIP)

The only objective of minimizing make-span and ignoring load balancing in TAMMs, may result in overloading of some actors and idling of some others, resulting in the partitioning of the WSA. To avoid this problem, as shown in Figure 3, we proposed a load balanced protocol (LBIP) that tries to improve

balanced loads on the actors. In LBIP, Eq. (5) is used to compute the utilization of the actor  $A_i$  assuming that there is no restriction on the number of tasks in an actor queue; utilization of an actor indicates the actor's busy times.

$$\begin{aligned} \rho &= \frac{(\lambda'_i + \lambda_i)}{\mu_i}, \\ (\lambda'_i + \lambda_i) &< \mu_i \end{aligned} \quad (5)$$

To improve balanced loads on the actors, the arrival rates of tasks for each actor should be based on the capability of that actor. Therefore, in this phase of algorithm, all arrival rates and  $\rho_i$  pairs are sorted and based on (6), a proper pair of  $\lambda_i$  and  $\rho_i$  is chosen and thus,  $n - 1$  equations is made.

$$(\lambda'_i + \lambda_i) \times \rho_i = (\lambda'_j + \lambda_j) \times \rho_j \quad (6)$$

These equations can be solved using Eq. (7), resulting in proper arrival rates of tasks that result in more balanced loads on the actors. Because in this case further and bigger tasks are transmitted to the actors in which the capability of performing tasks is more than others.

$$\begin{aligned} \lambda &= \sum_{i=1}^n \lambda_i \\ \lambda' &= \sum_{i=1}^n \lambda'_i \end{aligned} \quad (7)$$

#### 4.3 Time-Sensitive and Load Balanced Scheduling Approach (TSLBS)

The main goal in the TAMMs is to minimize make-span while the main goal in the LBIP is the increase of load balancing. The proposed TSLBS is applying TAMMs and LBIP and tries to reduce make-span, while load balancing increased and the network lifetime enlarged. Figure 5 shows the steps followed by the proposed TSLBS scheduling.

However, the only objective of reducing make-span and ignoring load balancing in TAMMs, may lead to overloading of some actors and idling of some other actors, resulting in the partitioning of the WSA. On the other hand, ignoring make-span and the sole objective of load balancing in LBIP, may lead to unnecessary delays. To avoid these problems, both make-span and load balancing should be considered. To achieve this goal, TSLBS by using the average amounts of  $\lambda_i$  that was derived in the TAMMs and LBIP, makes a proper trade-off between make-span and load balancing and tries to optimize balanced loads on the actors and reducing the network make-span, too. Figure 4 shows the steps followed by the TSLBS.

### 5. EXPERIMENTAL RESULTS

The proposed TAMMs, LBIP, and TSLBS have been compared with STSA algorithm in a typical scenario. In addition, to study the effect of scale on the performance of the proposed approach, simulations have carried out in both large and small scales in two different settings. In the small scale, we assume a

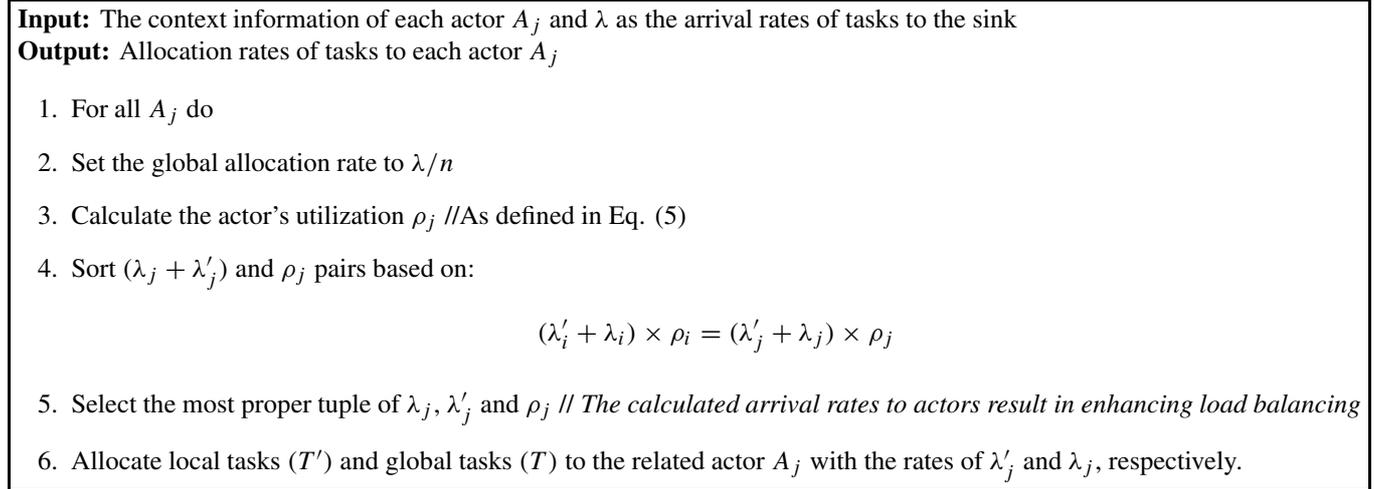


Figure 3 Proposed Load balancing Protocol.

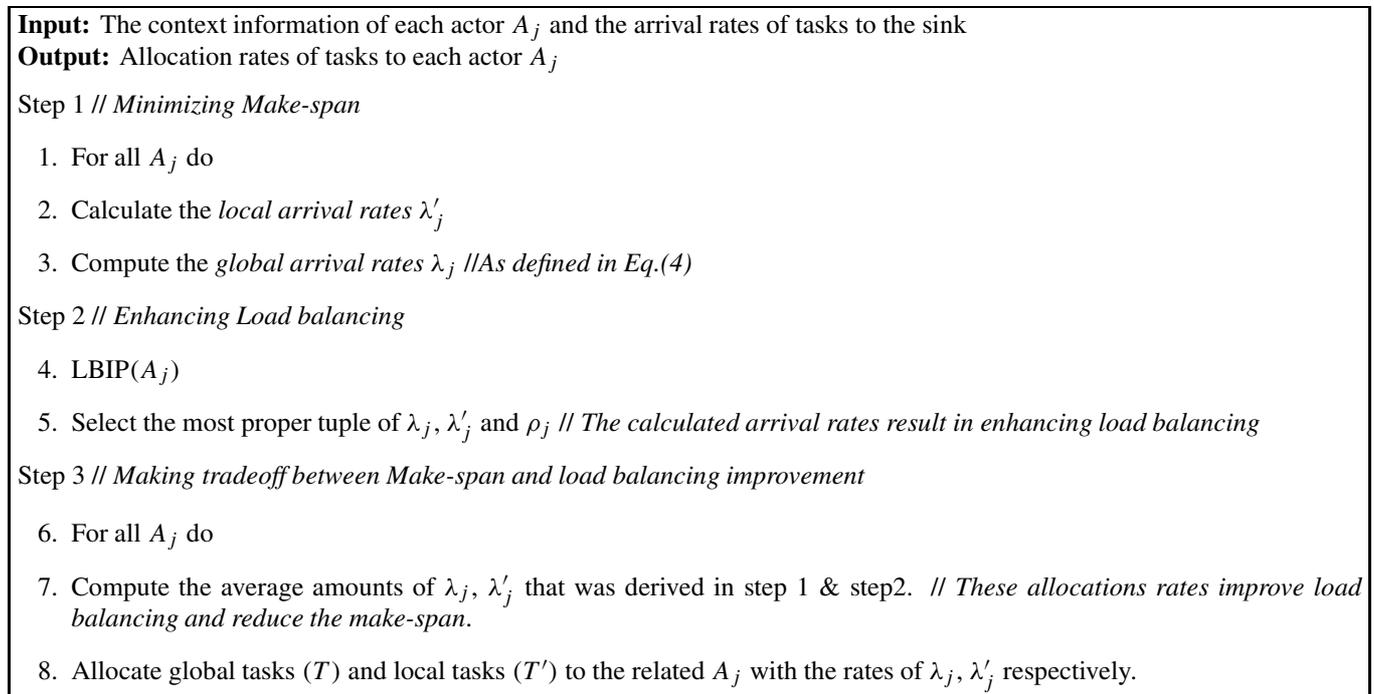


Figure 4 Steps followed by the proposed TSLBS scheduling.

two-dimensional square space,  $10m \times 10m$ , including 100 sensor nodes with 1 meter transmission range, and 7 actor nodes. We assume that the tasks to be carried out by actors were independent and that actors could browse the whole network with no restrictions on routing hops. The primary energy of each actor was assumed to be the same as others and equal to 20 Joules. In the large scale, we assume a two-dimensional square space,  $100m \times 100m$  including 1000 sensors with 10 meters transmission range, and 10 actor nodes. The primary energy of each actor is assumed to be the same as others and equal to 40 Joules. In our evaluations, we assume that each of the actors runs only a single task at any time. To have better evaluation, actors are selected from three different classes with slow, medium, and fast service rates. It is assumed that tasks are independent.

The workload of actors, the network make-span, the energy consumption of actors, and the network life time in small and

large scales are shown in Figures 5 to 12, respectively. In these figures, the efficiency of TAMMs, LBIP and TSLBS is evaluated in compare with each other and STSA in terms of make-span, load balancing, energy consumption and network lifetime. As figures 5 and 6 show, LBIP is the best in the balanced distribution of workloads on the actors in both small and large scales, respectively while STSA was the worse one.

As Figures 7 and 8 depict, although the make-span resulting from the TAMMs is much better than the make-span resulting from STAS and LIBP, it is a little better than the make-span resulting from TSLBS. However, the results are similar in both small and large scales, but the difference between the TAMMS and TSLBS is less in small scale.

Figure 9 and 10 show the energy consumptions of actors in Setting I and Setting II, respectively. Figure 10 shows, in small scale STSA and TAMMs have already the same operation, but

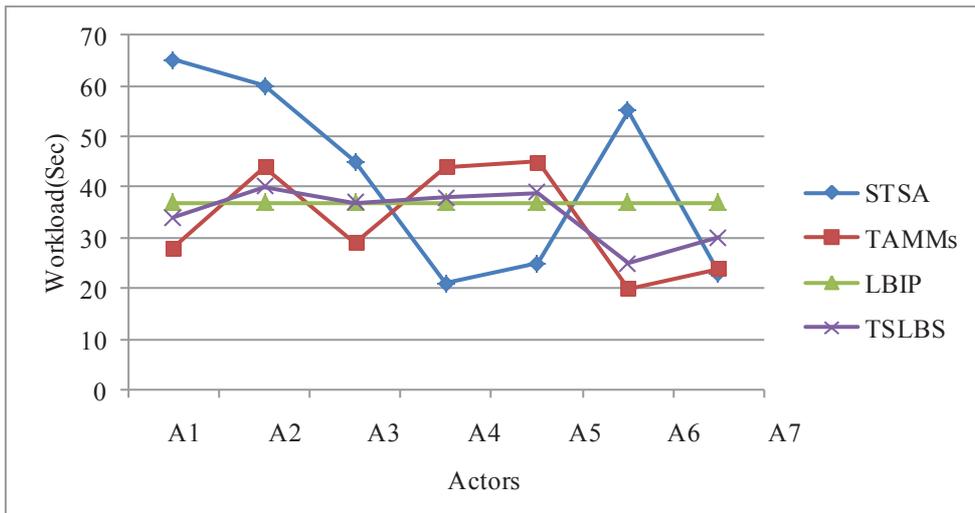


Figure 5 Workload of each actor in small scale.

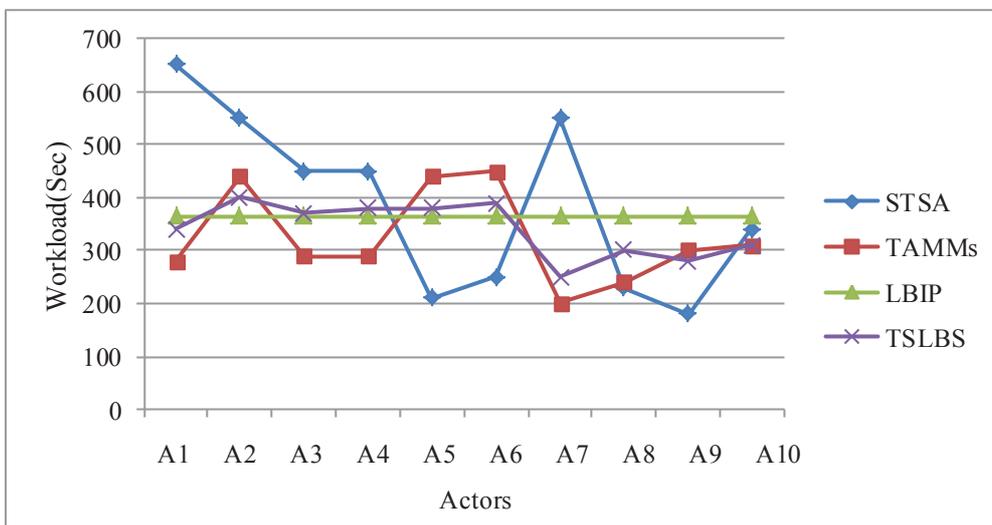


Figure 6 Workload of each actor in large scale.

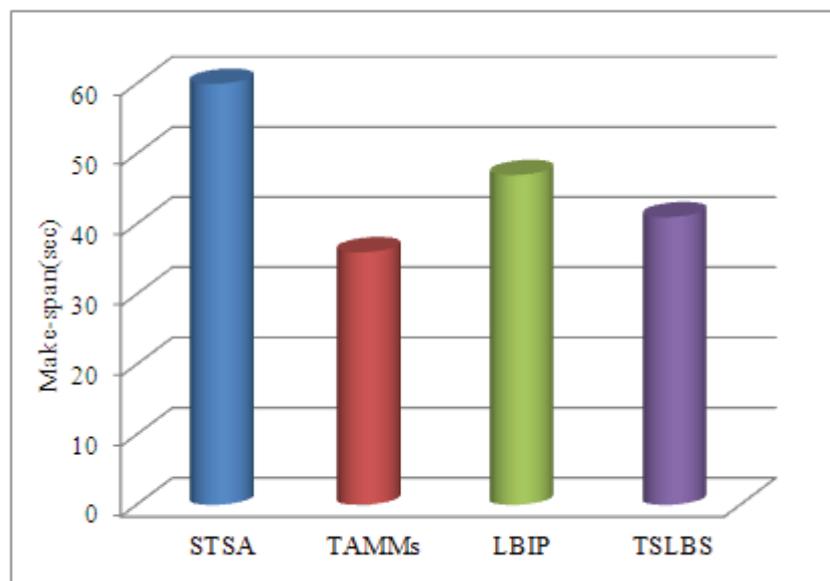


Figure 7 Make-span in small scale.

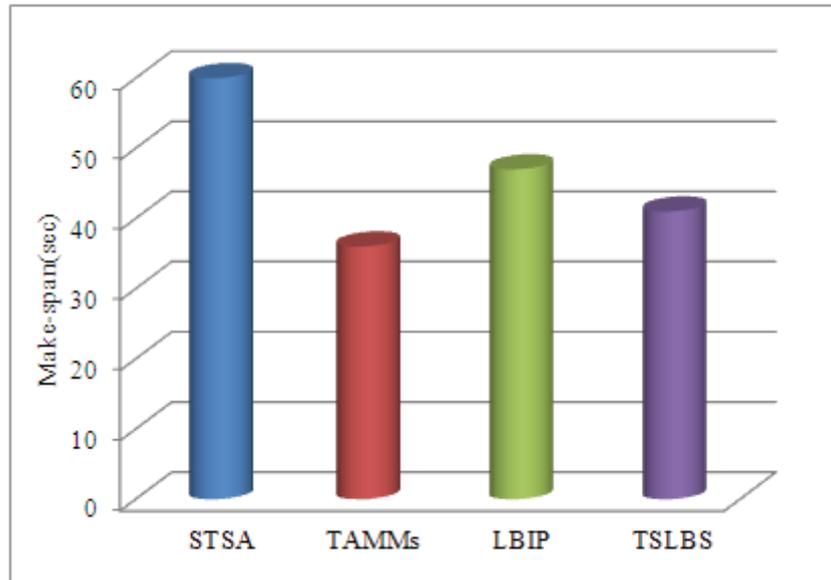


Figure 8 Make-span in large scale.

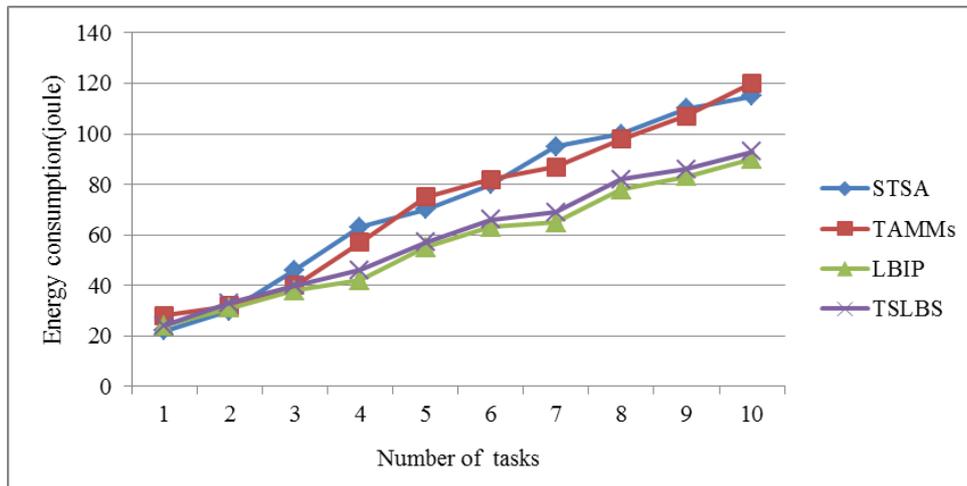


Figure 9 Energy consumptions of actors in large scale.

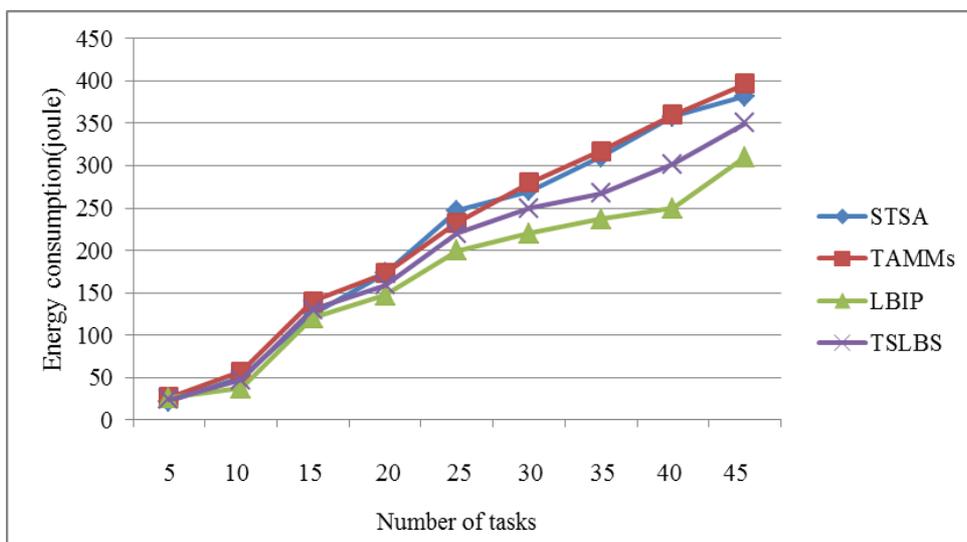


Figure 10 Energy consumptions of actors in large scale.

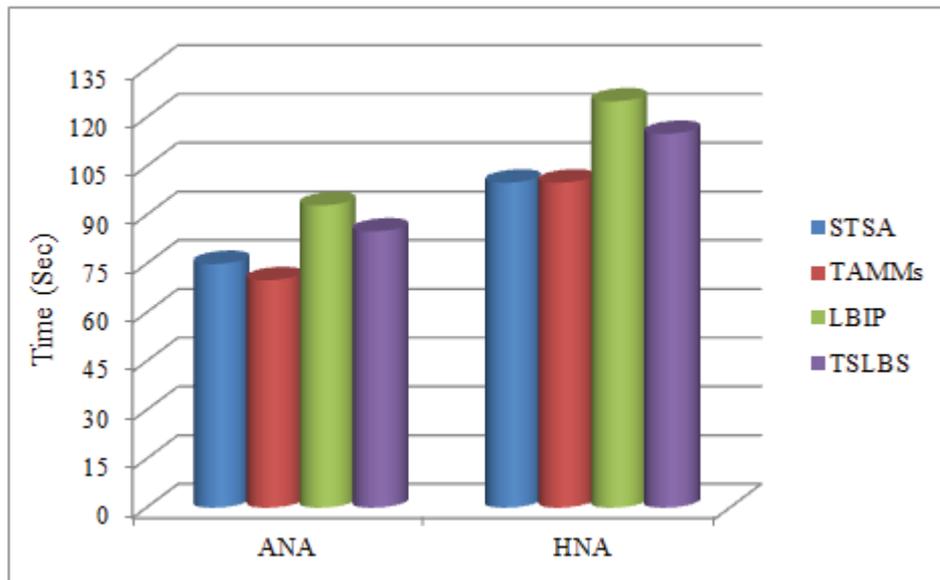


Figure 11 Network lifetime in small scale.

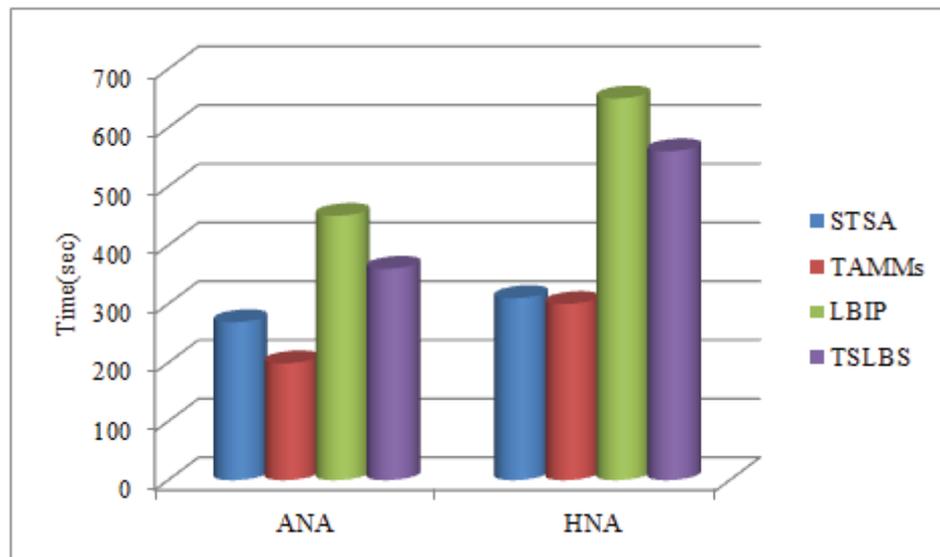


Figure 12 Network lifetime in large scale.

LBIP results in less energy consumption. Although the energy consumption resulting from LBIP is better than others, it is a little better than the energy consumption resulting from TSLBS. As shown in Figure 11, in large scale STSA and TAMMs have nearly the same operation in reducing energy consumption which is worse than others, but LBIP reduce the energy consumptions of actors better than others. It is worthy to note that STSA and TAMMs cannot perform any more tasks because of lacking of required energies, but LBIP and TSLBS can perform more tasks. However, as Figure 11 demonstrates, in large scale wherein the required energy to pass the distance between a selected actor  $A_j$  and the location of the task  $T_i$  allocated to actor  $A_j$  is significant in compare with  $E_{ij}$ ,  $C$  is operates considerably better than others in reducing energy consumptions of actors.

Figures 11 and 12 compare network life time of STSA, TAMMs, LBIP and TSLBS. The network lifetime is evaluated

in terms of all of the actor nodes alive (ANA) and half of the actor nodes alive (HNA) for the sake of clarity.

As shown in Figures 11 and 12, in terms of ANA, LBIP has done considerably better in both small and large scales while TAMMs done the worst. In terms of HNA, STSA and TAMMs have nearly the same operations and resulted in the worst HNA while LBIP performed the best and resulted in higher HNA compared with others.

All in all, STSA yields minimum make-span and LBIP yields the best load balancing and minimum energy consumption, but TSLBS yields reasonable results as a tradeoff between both load balancing and minimizing make-span objectives. It also performed well in enlarging network lifetime and reducing the energy consumptions of actor nodes.

## 6. CONCLUSION AND FUTURE WORKS

This paper proposed a time-sensitive and load balanced approach (TSLBS) comprising of three algorithms to schedule tasks in WSANs. TSLBS is based on queuing theory and models a WSAN with the queuing networks. The first algorithm tries to minimum make-span as its only objective. Enhancing the Balance of workloads on the actors is the objective of the second algorithm. Improving load balancing and hence, enlarging network life time along with reducing make-span are the objectives of the third proposed algorithm. Experimental results illustrated that the first algorithm did minimize make-span but was bad on load balancing. The second algorithm did best on load balancing but performed badly on make-span. The third proposed algorithm provided a suitable tradeoff between balance of workloads on the actors and minimizing make-span. Experimental results also showed lower make-span, higher load balancing, larger network lifetime, and lower energy dissipation of actors under the proposed algorithms compared to the STSA.

## REFERENCES

- Bai X, Yang S, Liu L, Cao M, Panneerselvam J, Sun Q, Wang H (2017) Collaborative Actuation of Wireless Sensor and Actuator Networks for the Agriculture Industry. *IEEE Access*. 5: 13286-13296
- Naderi H, Kangavari MR, Okhovvat M. ScEP: A Scalable and Energy Aware Protocol to Increase Network Lifetime in Wireless Sensor Networks. *Wireless Pers Commun*, 2015; 82: 611–623
- Kamespanah E, Sirjani M, Mechitov K, Agha G (2018) Modeling and analyzing real-time wireless sensor and actuator networks using actors and model checking. *International Journal on Software Tools for Technology Transfer*. 20(5): 547-561
- Okhovvat M, Kangavari MR. A mathematical task dispatching model in wireless sensor actor networks. *Comput Syst Sci & Eng*, 2019; 34 (1)
- Sharifi M, Okhovvat M (2012) Scate: A Scalable Time and Energy Aware Actor Task Allocation Algorithm in Wireless Sensor and Actor Networks. *ETRI Journal* 34(3): 330-340
- Nayak A, Stojmenovic I (2010) *Wireless Sensor and Actuator Networks: Algorithms and Protocols for Scalable Coordination and Data Communication*, Hoboken, New Jersey: Wiley Press
- Zonouz A. E, Xing L, Vokkarane V. M, Sun Y. L (2016) Hybrid wireless sensor networks: a reliability, cost and energy-aware approach. *IET Wireless Sensor Systems*. 6(2): 42-48
- Chen j, Cao X, Cheng P, Xiao Y, Sun Y (2010) Distributed Collaborative Control for Industrial Automation With Wireless Sensor and Actuator Networks. *IEEE Transaction on Industrial Electronics*. 57(12): 4219-4230
- Sharifi M, Najari Alamuti A (2007) A Hybrid Physical Architecture for Coordination in Wireless Sensor and Actor Networks. *International Review on Computers and Software (IRECOS)* 2(5): 555-560
- Stojmenovic I (2007) *Energy Conservation in Sensor and Sensor Actuator Networks, Wireless Ad-Hoc Networking: Personal-Area, Local-Area, and Sensory-Area Networks*, Wu S.-L, Tseng Y.-C, Eds., Auerbach Publications, Ch. 4, pp. 107-133
- Cao X, Chen J, Xiao Y, Sun Y (2010) Building Environment Control with Wireless Sensor and Actuator Networks: Centralized vs. Distributed. *IEEE Transaction on Industrial Electronic*. 57(11): 3596-3605
- Curiac D-L, Volosencu C (2014) Urban Traffic Control System Architecture Based on Wireless Sensor-Actuator Networks. In: *Proceeding of the 2nd International Conference on Manufacturing Engineering, Quality and Production Systems*, pp. 259- 263, March 30, 2014, pp. 259-263
- Shu Y, Shin K.G, Chen J, Sun Y (2017) Joint energy replenishment and operation scheduling in wireless rechargeable sensor networks. *IEEE Transaction on Industry Informatics* 13(1): 125-133
- Grzonka D, Koodziej J, Tao J, Khan S. U (2015) Artificial neural network support to monitoring of the evolutionary driven security aware scheduling in computational distributed environments. *Future Generation Computer Systems*. 51: 72-86
- Cooper RB. *Introduction to queueing theory*. 2nd Ed. Elsevier/North-Holland: Amsterdam; 1981
- Momeni H, Sharifi M, Okhovvat M (2012) A Mathematical Approach to Reduce the Mean Number of Waiting Tasks in Wireless Sensor Actor Networks. *Information Journal*. 15(10): 181-192
- de Farias C. M, Pirmez L, Delicato F. C, Li W, Zomaya A. Y, de Souza J. N (2013) A scheduling algorithm for shared sensor and actuator networks. In: *International Conference on Information Networking (ICOIN)*, Bangkok, Thailand, Januray 28-30, 2013, pp. 648-653
- Yang I, Lee D (2015) Networked Fault-Tolerant Control Allocation for Multiple Actuator Failures. *Mathematical Problems in Engineering* 2015(14): 1-14
- Okhovvat M, Kangavari M.R (2015) Modeling of Task Allocation in Wireless Sensor Actors Networks to Increase Actors' Utilization Using GSPN. In: *The Second International Conference on Knowledge-Based Engineering and Innovation (KBEL)*, Tehran, November 5-6, 2015, pp 1297-1302
- Brucker P. *Scheduling Algorithms*. 5th ed. Berlin, Germany: Springer, 2007
- Kim J, Dudin A, Dudin S, Kim C (2018) Analysis of a semi-open queuing network with Markovian arrival process. *Performance Evaluation* 120: 1-19
- He Y, Hsu W.J, Leiserson C.E (2008) Provably efficient online non-clairvoyant adaptive scheduling. *IEEE Transaction on Parallel and Distributed Systems*. 19(9): 1263–1279
- Bolch G, Greiner S, De Meer H, Trivedi KS. *Queuing Networks and Markov Chains*. 2nd ed. USA: Wiley, 2006.
- Generalized Algebraic Modeling Systems (GAMS). [Online]. Available: <http://www.gams.com>