Interference Aware Real-Time Flows Scheduling in Cluster Based Wireless Sensor Networks

Gohar Ali[1], KyongHoon Kim[2], Ki-Il Kim[2,*] Monther Aldwairi[3,4]

[1] Department of Information Systems and Technology, Sur University College, Oman.

[2]Department of Informatics, Research Center for Aerospace Parts Technology Gyeongsang National University, Jinju, Korea.

[3]Department of Network Engineering and Security, Jordan University of Science and Technology, Irbid 22110, Jordan

[4]College of Technological Innovation, Zayed University, Abu Dhabi, P.O. Box 144534, U.A.E

Received 22 December 2015; received in revised form 11 March 2016; accepted 17 March 2016

Abstract

For the real-time scheduling in wireless sensor networks, Time Division Multiple Access (TDMA) under cluster architecture is usually considered as reasonable as well as scalable approach. Under this architecture, several type of inferences should be taken into account to allocate time slot. For this goal, we have already proposed a new real-time scheduling algorithm. But, it has several assumptions and constraints so their deployment is limited to some specific real scenarios. In this paper, we additionally concern the interference of a node which belongs to two clusters concurrently. Token based scheduling algorithm between two cluster heads is proposed to solve this problem. Token is given to each cluster head consequently so order of slot allocation is achieved automatically. Finally, simulation results are given to prove that more flows are delivered within the deadline than previous work due to avoidance of inter-cluster interference in efficient way.

Keywords: real-time, Interference, TDMA, scheduling, wireless sensor networks

1. Introduction

Recently, Wireless Sensor Networks (WSN) are widely employed for many real-time applications in the real world due to low cost. The representative example includes fire monitoring systems, health monitoring, disaster management, industrial process monitoring and traffic monitoring systems[1]. In these systems, the events detected by sensors must be reported within certain time limit, called deadline. But, it is harder to provide real-time communications in WSN than typical networks due to constraints such as computational power on node and communication bandwidths in networks. But, when it comes to time sensitive property of sensing data, real-time communication is one of great research challenges in WSN. To address this research challenges, like typical networks, diverse schemes have been proposed in each layer approach. Among them, as real-time approach for medium access control layer, TDMA scheduling is promising approach to meet real-time requirement since we can expect the required time in deterministic way. However, in order to apply TDMA in WSN, it is more difficult to achieve due to natural properties of wireless communication based on broadcast. Because communication between pair of nodes is affected by other nodes that are within the interference range of these nodes and transmit data at the same time, interference-aware scheme becomes important in WSN.

*Corresponding author. E-mail address: kikim@gnu.ac.kr

Tel.: +82-55-7721373; Fax: +82-55-7721379

For this purpose, interference-aware real-time communication for WSN have been addressed previously in [2-4]. However, since mentioned schemes have scalability problem as well as require additional time caused by coloring scheme, we have proposed a new cluster-based scheduling algorithm in [5]. In our scheme, TDMA was introduced and real-time flows were scheduled by considering the interference of flows within cluster and between cluster heads. But, due to assumptions on simple network model, the proposed scheme ignored interference problem at the overlapped node which belongs to two cluster areas concurrently. But, since these overlapped nodes are natural and easily detected in the real deployment scenarios, this problem is designated as one of main research challenge to be resolved in our previous work.

In order to solve mentioned problem, in this paper, we extend our previous works by taking interference between node and neighboring node within the other cluster into account. For this extension, token based slot assignment algorithm is newly proposed to avoid interference between clusters. Token carried by stoken field in a message is given to each cluster head which has chance to allocate the time slot at this time. If a cluster head possesses stoken value larger than zero, this node assigns time slot for the flow. And, the stoken value is decreased by 1. And then, stoken value is passed to next node according to flow priority. This procedure will continue until stoken value becomes zero. As a result, time slot allocation is solved in deterministic way to meet requirement. Moreover, we provide simulations results in order to show the effectiveness of the proposed scheme in the aspects of delivery ratio within the deadline. The simulation scenarios are made differently according to the ratio of slots for intra-cluster and inter-cluster communication.

The rest of the paper is organized as follows. We describe the overview of our previous work and their limitation in the section 2. In the section 3, we present the proposed scheduling algorithm. And then, we provide performance evaluation results and their analysis in the section 4. We conclude the paper in the section 5.

2. Overview of Existing Schemes

In this section, we briefly present the existing interference aware TDMA scheduling as well as TDMA scheduling algorithms under cluster architecture. Also, we explain our research motivation caused by our previous work.

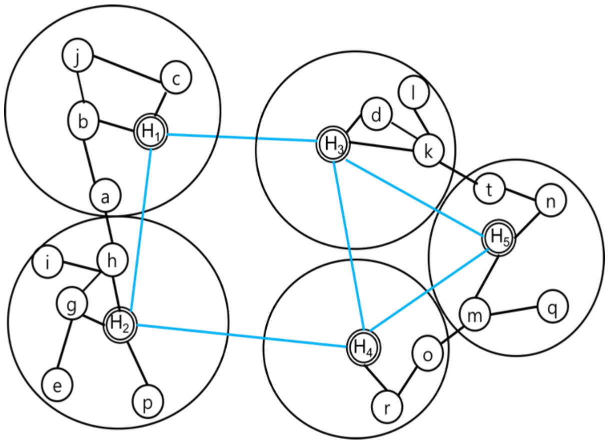
At first, interference aware TDMA scheduling is categorized as centralized and distributed approach. Two centralized, node-based and level-based TDMA scheduling algorithm is proposed in [6]. Both algorithms are based on station assign slots to nodes by considering interference relation. These algorithms attempts to find TDMA schedule that minimizes the number of required time slots. Also, conflict free TDMA scheduling algorithm for multi-hop intra and inter-cluster is presented in [7,8]. The algorithm schedules every node across 3-hop neighbor for the purpose of avowing the interface to improve the delay, throughput and energy efficiency. In some algorithms, the knowledge of 2-hop neighborhoods is assumed to avoid interference [9]. Also, a distributed 2-hop coloring algorithm is proposed to find conflict-free slots across 2-hop neighbors. For the real-time TDMA scheduling, WirelessHART standard for sensor-actuator network has shown the interest in real-time communications [1]. These protocols use centralized approach for the transmission schedule and avoid concurrent transmissions with in the same channel. The approach uses different priority scheme in order to achieve the real-time communications through TDMA scheduling. This standard increases the delay and limits the scalability in an efficient way. In addition, implicit prioritization access protocol [10] and real-time query scheduling algorithm was studied in [11], respectively. They assume a pre-given routing tree for different scheduling strategies by considering the trade-off between prioritization and throughput. Also, real-time capacity for wireless networks by assuming ideal TDMA is proposed to deliver the data by their deadline in [12].

Secondly, adaptive distributed randomized was implemented in cluster-based TDMA scheme for WSN. The slots are assigned to node based on the one-hop and two-hops neighbors' information. The cluster head is likely to allocate more slots and improve channel utilization and throughput. However, in order to acheive the energy balance, the cluster head is alternated by other cluster members which increase the overheads due to the reassignment of slots. In [13], cluster-based data collection scheme was proposed for the intra and inter cluster delay. The intra-cluster communication is TDMA-based and cluster head in one-hop receives the data from their members. The cluster heads then transmit the gathered information to the sink. However, in this scheme, the end-to-end delay was not investigated any more. In [14], cluster-based TDMA scheduling algorithm was proposed to achieve the real-time performance. The interference is avoided by distributed distance-2 vertex coloring algorithm. The intra-cluster delay decreases by scheduling maximum independent sets of nodes in the same slots. In [15], QoS-aware cluster-based data reporting control scheme was proposed. In this scheme, each node transmits the sensing data to its cluster head. And then, aggregated sensing data is sent to the base station. The whole scheme consists of Intra-cluster data reporting(IntraDRC) and inter-cluster control(InterDRC) method. For IntraDRC, only few member nodes were selected as reporting node to meet the required throughput and save energy in other rounds. Delay bounded data was routed by taking minimum number of hops which consists of combination of cluster heads and member nodes in InterDRC.

In addition, Cluster-based TDMA scheme to accomplish optimized energy efficiency and minimum delay was proposed in [16]. This scheme reduces the end-to-end delay by using the slot reuse concept. For the saving of battery power and fast and efficient query response to sink, TDMA algorithm is proved to be more effective than contention based protocol. In [17], cluster-based unique convergecast scheduling problem of information gathering is investigated. They assume that all nodes in a cluster have exactly one packet of information to be sent to the sink. They proposed heuristic algorithm based on spanning tree where the degree of transmission parallelism is increased by scheduling independent sets in the same slot. In [18], an adaptive clustering scheme is proposed for event-driven applications. A mechanism called an event-driven cluster head selection adapts the formation of the cluster according to the change in event sending. Thus, the clusters are adapted according to the change in event sensing so that cluster-based structure guarantee real-time requirement. However, the information accuracy is reduced by combining the packets. In cluster-tree model based on single channel and time division cluster scheduling for real-time flows has been derived [19]. The authors proposed a receiver-oriented scheduling algorithm which offers bounded latency for data gathering applications. They claimed that their algorithms can effectively reduce the interference among clusters when cluster heads send data to the root. As a survey paper, the authors in [20] introduce TDMA-based MAC in the point of how they avoid collisions, overhearing and idle listening and therefore energy efficient. And then, well referred several TDMA-based MAC protocols both centralized and distributed are presented. They categorize the existing scheme and compare them in the point of delay, topology, and scheduling way. Also, each protocol is well analysis with advantage and disadvantage together. Also, they address that TDMA based approaches have some drawbacks too like limited scalability as a function of number of flows, long time for adaptability to network changes, strict time synchronization and suggest finding a distributed energy efficient standard MAC protocol suitable for wireless sensor network.

In addition to mentioned schemes, our previous scheduling algorithm consists of IntraSend, InterComm, and IntraRecv TDMA scheduling algorithms as in [5]. In IntraSend slots, normal nodes send real-time flows into their cluster heads. Cluster heads transmit flows to the destination clusters in InterComm time slots. In IntraRecv slots, real-time flows from other clusters are delivered to destination nodes. One minor TDMA frame consists of consecutive slots of IntraSend, InterComm, and IntraRecv slots. This minor frame is repeated during the run-time. Cluster coloring scheme is one possible solution to avoid the above problem in intra-cluster real-time flows scheduling. The intra-cluster time slots are further divided into number of colors used. Thus each intra-cluster scheduling algorithm will schedule real-time flows in the corresponding coloring slots. If a cluster graph is planar, four-coloring is possible, which implies that the number time slots for intra-clustering is increased by four times in order to solve the inter-cluster interference problem due to Signal to Noise plus Interference Ratio (SNIR). In this scheme, we develop IntraSend, InterComm and IntraRecv algorithm, respectively. Fig. 1 shows example network model and routing path for the flow. Also, Table 1 shows the example of flow scheduling when our algorithm is applied for example in

Fig. 1. By the proposed scheme, we verify that our proposed scheme accepts more real-time flows than existing scheme due to flexibility.



(a) Network architecture

Flow	Routing Path
F_1	$a \rightarrow b \rightarrow H_1 \rightarrow H_3 \rightarrow H_5 \rightarrow m \rightarrow q$
F_2	$i \rightarrow h \rightarrow H_2 \rightarrow H_3 \rightarrow H_5 \rightarrow k \rightarrow l$
F_3	$p \rightarrow H_2 \rightarrow H_4 \rightarrow H_5 \rightarrow n \rightarrow t$
F_4	$e \rightarrow g \rightarrow H_2 \rightarrow H_4 \rightarrow H_5 \rightarrow d$
F_5	$j \rightarrow c \rightarrow H_1 \rightarrow H_2 \rightarrow H_4 \rightarrow r \rightarrow o$

(b) Routing path

Fig. 1 Example of flow scheduling

Table 1 Example of IntraSend scheduling results

Cluster	**Flow**	**r_i**	**1**	**5**	**9**	**a_i^{SH}**
H1	F1	0	a->b	b->H1		5
	F5	0		j->c	c->H1	9
H2	F2	0	i->h	h->H2		5
	F3	0	p->H2			1
	F4	0	e->g		g->H2	9

However, there are several assumptions for this model such as routing path and topology information. Among them, we assume that there is no interference among different clusters as well as homogenous cluster heads with same transmission range. However, this assumption arises deployment issue because overlapped node between cluster is too natural and easily detected and homogenous cluster heads are ideal models. For example, in the Fig. 1(a), node a and h belong to different cluster. But, this situation rarely happens in real world where random deployment is general strategy. In this paper, we consider this interference during scheduling. So, the new network model is depicted with overlapped sensor nodes in the Fig. 2.

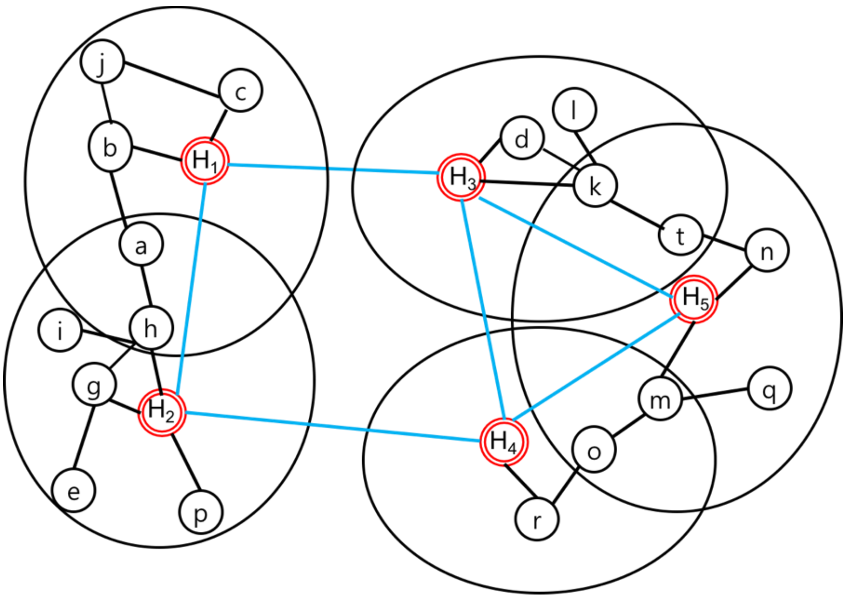


Fig. 2 New network model

3. Proposed Scheme

3.1 Network and Flow Model

The network is represented by a graph G= (V, E) where V is the set of sensor nodes and E denotes the edges between nodes. These edges are represented by e = (u, v). We assume that if there is edge between two nodes then they can communicate with each other. Thus, communication edge of $\overrightarrow{uv}$ indicates that node u can send data to node v. When nodes communicate each other, two types interference such as primary and secondary interference, may occur. A primary interference occurs when a node receives and transmits at the same time or receives more than one transmission destined to it. On the other hand, secondary interference occurs when an intended receiver of a particular transmission is also within the transmission of another transmission intended for other nodes.

Under this network model, as for real-time communication, a real-time flow denoted by F_i, is a set of message streams from a source node src_i to a destination node dst_i. Each flow F_i has unique priority i so that flow F_i has higher priority than F_j for $i < j$. For the frame structure, global TDMA time slots are divided into three different time slots namely, IntraSend, InterComm and IntraRecv slots like our previous work. By using IntraSend slots, member nodes send real-time packets to their cluster heads. While a cluster heads transmits flows to the destination CH in InterComm slots, the flows from destination CH are transmitted to destination node in IntraRecv slots. So, end-to-end communication is accomplished by assigning appropriate time slot in InterComm, InterComm, and IntraRecv. Some flow needs all type of slots while other do some of them according to the current of node location.

3.2 Procedure of algorithm

Identical to the previous work, the main procedures of proposed scheme consist of initialization and scheduling step. In initialization step, each cluster decides IntraSend and IntraRecv scheduling for corresponding flows and manages the time tables in the cluster head. At the same time, cluster heads decide the scheduling of flows for InterComm time slots and share the table. In scheduling step, each node transmits flows according to three TDMA time tables for given flows.

In the initialization step, clusters share their inter-cluster interference among other clusters. For example, in Fig. 2, both H_1 and H_2 share the interference of a and h. Accordingly, H_3, H_4, and H_5 share their interference information. In IntraSend scheduling, each cluster head schedules their out-going flows in the priority order. In order to avoid inter-cluster interference problem, we use a token-based flow scheduling approach where a flow can be scheduled after receiving all required tokens from other clusters. For example, the scheduling of F_2, F_3, and F_4 in cluster head C_2 can be done after the header H_2 receives one token which is generated after scheduling of F_1 in C_1. Similarly, the flow F_5 is scheduled when the header receives three tokens from H_2. The scheduling information regarding to inter-cluster interference is also shared when the scheduling token is generated.

The second scheduling step is InterComm scheduling which schedules intra-cluster flows among cluster heads. The last step is IntraRecv scheduling which schedules all flows arriving at the destination clusters. We also use the same token-based approach as in IntraSend scheduling step so that possible inter-cluster interference is avoided. Thus, in this paper, we describe IntraSend scheduling algorithm in detail in the next subsection.

3.3 IntraSend Scheduling

Each node sends its cluster information to all neighbor nodes so that inter-cluster interference information is shared. For example, both nodes of a and h in Fig. 2 are aware that they may cause inter-cluster interference. Cluster heads gather all the interference information from nodes in their clusters. We denote V_i^I as the set of nodes of cluster i which interfere with other cluster nodes, and C_i^I as the set of such clusters.

Next, we define a token for each flow to be scheduled in IntraSend scheduling. The token of flow F_i, denoted as $stoken_i$, indicates the number of tokens remaining to start scheduling of flow F_i. The token value is initialized by the number of higher-priority flows in neighbor clusters with inter-cluster interference. $stoken_i$ shows how to decide the token value of each flow. For example, flows departing from C_2 need tokens from higher-priority flows in neighbor clusters with interference relationship, which is C_1. Since there is flow F_1 in C_1, the token values of F_2, F_3, and F_4 are initialized by one. Similarly, flow F_5 in C_1 should receive tokens from F_2, F_3, F_4 so that its $stoken_i$ is three, as shown in Table 1.

Table 1 The initial values of $stoken_i$ in Fig. 2

Flow	Src. Cluster	Interf. Cluster (IC)	Higher-priority Flows in IC	stoken
F1	C1	C2	-	0
F2	C2	C1	F1	1
F3	C2	C1	F1	1
F4	C2	C1	F1	1
F5	C1	C2	F2, F3, F4	3

3.4 Description of algorithm

Fig. 3 shows two algorithms which run concurrently in each cluster head. The cluster head manages the set of flows to schedule in the cluster, denoted as F_C (line 1 -- 8). If the value of $stoken_i$ is greater than zero, the scheduling of flow *i* is delayed until the value becomes zero (line 10). This condition is released by receiving the remaining tokens from neighbor cluster heads (line 20 -- 23).

If $stoken_i$ is zero, flow *i* is scheduled by the algorithm IA-FP-Scheduling[5] which is an interference-aware scheduling inside a cluster. In IA-FP-Scheduling, the release time of F_i is computed. Also, if the lower priority flow interferes with higher priority, then it is delayed by one frame. So, the algorithm finds the first schedulable frame for a transmission $\overrightarrow{uv}$. After the scheduling of flow i, the cluster head notifies it with the scheduling information. The scheduling information with regard to interference nodes in V_C^I is stored into I and sent to all interfered neighbor clusters, C_C^I (line 12 -- 18). When a cluster head receives this information, the stoken values of flows to schedule are decreased by one. And the scheduling information of other clusters is updated as shown in line 22 -- 23. For each variable, the definition and meaning of them can be found in [5].

Algorithm IntraSend-Scheduling(C, $\mathscr{F}$)

/* $G_c = (V_c\ , E_c)$: the graph of cluster C */

1: $\mathscr{F}_c \leftarrow \varnothing$

2: **for** each flow $F_i \in \mathscr{F}$ **do**

3: **If** $src_i \in V_c$ **then**

4: $F_k \leftarrow F_i$ by updating the cluster head as dst_k

5: $\rho_k \leftarrow$ a path from src_i to the cluster head in ρ_i

6: $\mathscr{F}_c \leftarrow \mathscr{F}_c \cup \{ (0, F_k, \rho_k) \}$

7: **Endif**

8: **endfor**

9: **for** $f_i \in F_C$ by the priority order **do**

10: **while** $stoken_i > 0$ **wait** ;

11: $a_i^{SH} \leftarrow$ ***IA-FP-Scheduling*** $(G_C, f_i,$ *T*, *IntraSend*);

$T \leftarrow \emptyset$
for t **from** 0 **to** a_i^{SH} **do**
Let $T[i][t]$ be $\overrightarrow{uv}$.
if $u \in V_C^I$ **or** $v \in V_C^I$ **then**
$T \leftarrow T \cup \{ (i, t, \overrightarrow{uv}) \}$
endfor
for each cluster j in C_C^I **do**
send T to the cluster head of j
endfor
endfor

Algorithm RecvTokenSch (T)
for $f_i \in F_C$ **do**
$stoken_i \leftarrow stoken_i$ **-1;**
endfor
for each $(i, t, \overrightarrow{uv}) \in T$ **do**
$T[i][t] \leftarrow \overrightarrow{uv}$;
endfor

Fig. 3 Scheduling algorithm with token

4. Performance Evaluation

4.1 Simulation Environment

We have developed simulation programs to simulate the proposed framework. We used the GENSEN tool [21] to generate the network topology where about 50 nodes are generated randomly in 100m * 100m area. The number of clusters is varied from 4 to 8. For each simulation, we generate 100 random cases and count the number of cases where all flows are scheduled in their deadlines. A real-time flow in generated randomly in the network topology so that source and destination node belong to different clusters. The routing path of a flow follows the shortest path algorithm form the source to the destination and assumed to be determined in predetermined way. The simulation results provide the number of flows delivered within the deadline. The deadline of flow is the multiple of the number of hops from source to destination denoted by $M_{deadline}$. So, if a flow has 6 hops and $M_{deadline} = 2$ then the deadline is defined by 12 (= 6 * 2). The proposed scheme performance is analyzed based on two metrics i.e impact of flow numbers and impact of Intra-cluster and Inter-cluster slots.

4.2 Impact of flows

In Fig. 4, the acceptance rate for number of flows=8 shows lower than the others due to high contention in intra-cluster scheduling. Furthermore, the acceptance rate of flows increases at different rate from others flows as the deadline increases. For instance, the case that the deadline is set to 3 and number of flows is 4 has 55% acceptance rate while eight number of flows shows 5% acceptance rate. However, when the deadline becomes 5, the acceptance rate of number of four flows is 40% higher than the case that the number of flows is 8. The reason for reduction of the difference between the acceptance rate is, as the deadline increases each flows has much time to be scheduled so it increases the acceptance rate of flows. This also implies that all real-time flows can be delivered within the deadline if the acceptable deadline is given. On the other hand, how to assign acceptable deadline remains as another research challenge.

4.3 Impact of intra/inter time slots

In addition to simulation for number of flows, in case of the high number of intra-cluster time slots, the acceptance rate is low as you can see in Fig. 5. This is because most of the time slots are unused when the flows leave from their source cluster. Similarly, another opposite case that lower number of intra-cluster time slots the performance with N_{IC}=1 also shows low performance than one with N_{IC} =2. This is mainly because it is not enough to schedule inter-cluster flows due to high interference of cluster head when the number of N_{IC} becomes 1. While N_{IS} represents the number of IntraSend time slots per frame, N_{IC} does the number of InterComm time slots for frame. Finally, the number of IntraRecv slots for frame is given by N_{IR}. Each case is depicted in an order of N_{IS}- N_{IC}- N_{IR} in Fig. 5 and 6.

Similarly, in another scenario to analyze the performance of Inter slot number in Fig. 6, the number of inter slots number are increased by 2 times of intra-cluster time slots. As shown in Fig. 6, poor performance is observed for larger number of intra/inter slot number. Because too many time slots are allocated to each scheduling scheme, it increases the schedule length so more flows miss their deadline.

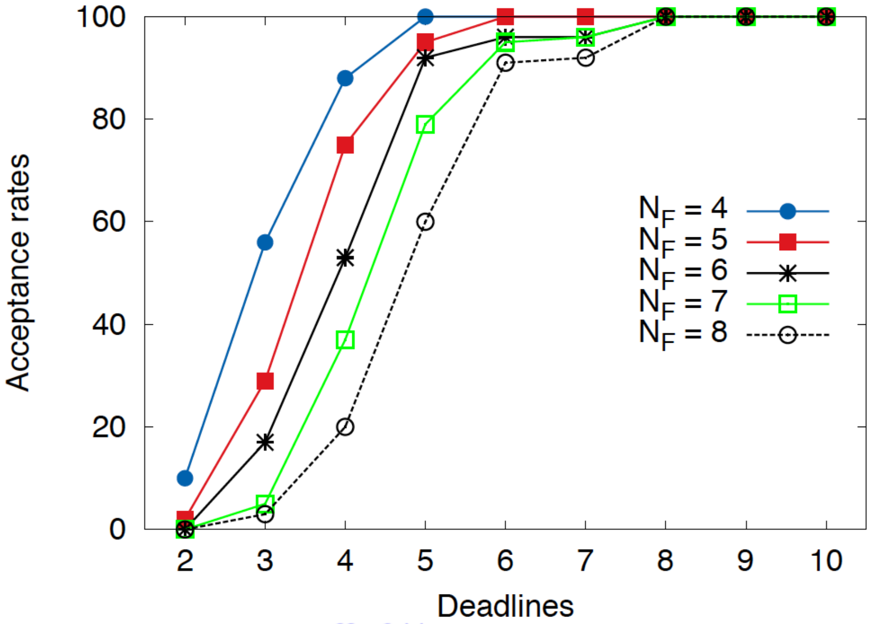


Fig. 4 Acceptance rates as a function of deadline

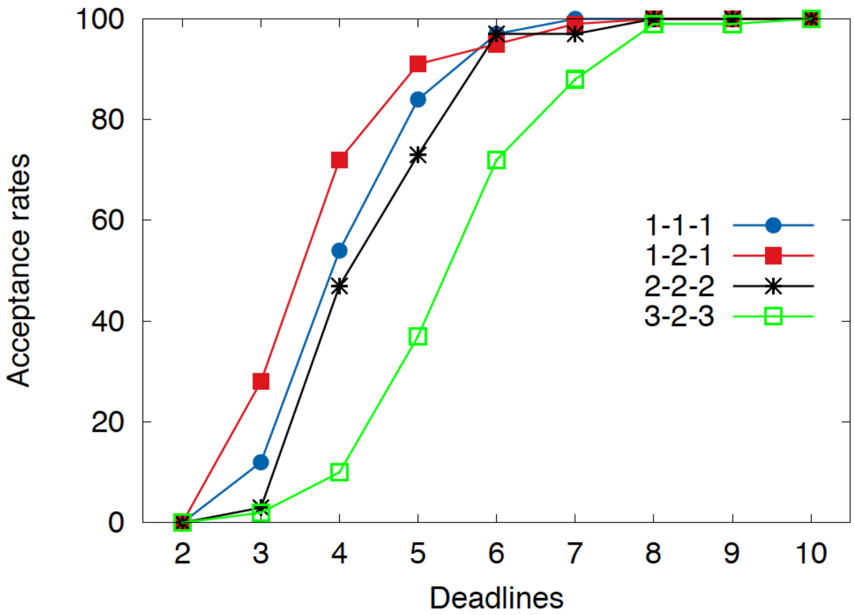


Fig. 5 Acceptance rates as a function of slots

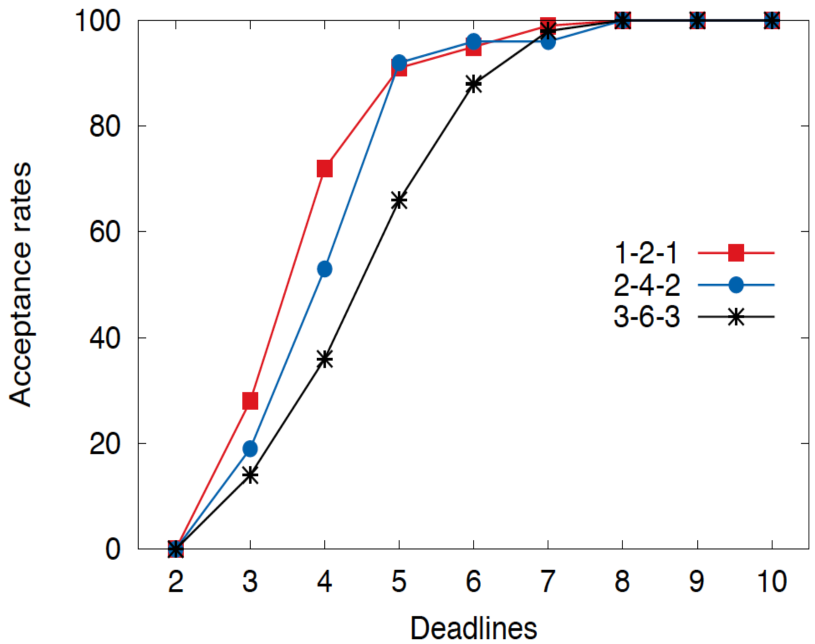


Fig. 6 Acceptance rates as a function of slots

5. Conclusion

Real-time communications attract research interests in community and become important in many fields of wireless sensor networks. In this paper, we proposed a new scheduling approach for real-time communication under cluster architecture in WSN by extending our previous work. To avoid the essential interference between cluster which was ignored in our previous work, a newly token based slot assignment algorithm is proposed. Simulation results demonstrated that more flows are admitted than the previous work. Also, we analyzed the acceptance rate of flows which meets its deadline on different parameters.

Related to the proposed algorithm, we plan to develop and design algorithm by applying a compositional real-time scheduling algorithm into the proposed algorithm. Also, other assumptions including routing path will be removed in the further research work sequentially in order for real deployment.

Acknowledgments

This work was funded by Sur University College, Sur, Oman and supported in part by Zayed University Research Office, Research Incentive Grant R15121. Also, it was supported by Basic Science Research Program (2015R1D1A1A01056979) through the National Research Foundation of Korea (NRF) funded by the Ministry of Education.

References

[1] A. Saifullah, Y. Xu, C. Lu, and Y. Chen, “Real-time scheduling for WirelessHART networks,” Proc. of Real-Time Systems Symposium, 2010.

[2] S. C. Ergen and P. Varaiya, “TDMA scheduling algorithms for wireless sensor networks,” Wireless Networks, vol. 16, no. 4, pp. 985-997, 2010.

[3] M. D. Francesco, C. M. Pinotti, and S. K. Das, “Interference-free scheduling with bounded delay in cluster-tree wireless sensor networks,” Proc. of Modeling Analysis and Simulation of Wireless and Mobile Systems, Paphos, Cyprus, 2012.

[4] O. Chipara, C. Wu, C. Lu, and W. Griswold, “Interference-aware real-time of scheduling for wireless sensor networks,” Proc. of Euromicro Conference on Real-Time Systems, 2011.

[5] G. Ali, S. Y. Kang, K. H. Kim, and K. I. Kim, “Towards cluster-based real-time scheduling in interference-aware wireless sensor networks,” Proc. of IEEE International Conference on Computational Science and Engineering, 2013, pp. 523-530.

[6] S. C. Ergen and P. Varaiya, "TDMA scheduling algorithm for wireless sensor networks," Wireless Networks, pp. 985-997, 2010.
[7] P. M. Pawaer, R. H. Nielsen, N. R. Pardad, S. Ohmori, R. Prasad, "GCF: green conflict free TDMA scheduling for wireless sensor networks," Proc. of IEEE International Conference on Communications, pp. 5726-5730, 2012.
[8] P. M. Pawaer, R. H. Nielsen, N. R. Pardad, S. Ohmori, and R. Prasad, "M-GCF: multicolor-green conflict free TDMA scheduling for wireless sensor networks," Proc. of Wireless Personal Multimedia Communications, pp. 143-147, 2012.
[9] I. Rhee, A. Warrie, J. Min, and L. Xu, "DRAND: distributed randomized TDMA scheduling for wireless ad hoc networks," IEEE Transactions on Mobile Computing, vol. 8, no. 10, pp. 1384-1396, 2009.
[10] M. Caccamo, L. Zhang, L. Sha, and G. Buttazzo, "An implicit prioritized access protocol for wireless sensor networks," Proc. of IEEE Real-Time Systems Symposium, 2002.
[11] T. He, B, M. Blum, Q. Cao, J. A. Stankovic, S. H. Son, and T. F. Abdelzaher, "Robust and timely communication over highly dynamic sensor networks," Real Time System, vol. 30, no. 3, 2007.
[12] O. chipara, L. Chenyang, and G. Roman, "Real-time query scheduling for wireless sensor networks," Proc. of Real-Time Systems Symposium, pp. 389-399, 2007.
[13] M. Lotfinezhad, B. Liang, and E. S. Sousa, "Adaptive cluster-based data collection in sensor networks with directed sink access," IEEE Transactions on Mobile computing, vol. 7, no. 7, pp.884-897, 2008.
[14] H. Kang, Y. Zhao, and F. Mei, "A graph coloring based data tdma scheduling algorithm for wireless sensor networks," Wireless Personal Communications, vol. 72, no. 2, pp. 1005-1022, 2013.
[15] H. J. Choe, P. Chosh, and S. K. Das, "QoS-aware data reporting control in cluster-based wireless sensor networks," Computer Communications, vol. 33, no. 11, pp. 1244-1254, 2010.
[16] L. Shi and A. Fapojuwo, "TDMA scheduling with optimized energy efficiency and minimum delay in clustered wireless sensor networks," IEEE Transactions on Mobile Computing, vol. 9, no. 7, pp. 927-940, 2010.
[17] H. Choi, J. Wang and E. A. Hughes, "Scheduling for information gathering on sensor networks," Wireless Networks, vol. 15, no. 1, pp.127-140, 2009.
[18] B. Abid, "An event-driven clustering scheme for data aggregation in real-time wireless sensor networks," Proc. of Advanced Information Networking and Application, pp. 48-55, 2013.
[19] M. D. Francesco, C. M. Pinotti, and S. K. Das, "Interference-free scheduling with bounded delay in cluster-tree wireless sensor networks," Proc. of ACM international conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems, pp. 96-106, 2012.
[20] P. Pal and P. Chatterjee, "A survey on TDMA-based mac protocols for wireless sensor network," International Journal of Emerging Technology and Advanced, vol. 4, no. 6, pp. 219-230, June 2014.
[21] T. Camilo, J. S. Silva, A. Rodrigues, and F. Boavida, "Gensen: A topology generator for real wireless sensor networks deployment," Proc. of IFIP Workshop on Software Technologies for Future Embedded Ubiquitous Systems, Santorini, Greece, 2007.