Computer Science and Information Systems 13(2):475–492 DOI: 10.2298/CSIS150401005G

Adaptive TDMA Scheduling for Real-Time Flows in Cluster-Based Wireless Sensor Networks

Gohar Ali[1], Kyong Hoon Kim[2], and Ki-Il Kim*[2]

[1] Sur University College,Oman
[2] Department of Informatics
Research Center for Aerospace Parts Technology
Gyeongsang National University
Jinju-Dearo 501, Korea, 52828
*kikim@gnu.ac.kr

Abstract. To prevent scalability problem caused by increasing traffic of real-time applications, cluster architecture with Time Division Multiple Access (TDMA) scheduling scheme is mostly deployed in wireless sensor networks. However, even though it have proven good scalability and suitability for real-time communications, but static scheduling lacks of adaptability in several situations by not admitting some real-time flows where slots remain available. *To solve mentioned problem, in this paper, we propose an adaptive cluster-based scheduling scheme for real-time flows by utilizing the time slots for flows according to type of flows such as inside or outside cluster. So, the proposed scheme can achieve better utilization of channel by minimizing the number of unused channels for real-time flows than static scheme through new adaptive TDMA scheduling scheme* . Simulation results show that more flows are admitted and delivered within the deadline in the proposed scheme by utilizing unused time slots accordingly.

Keywords: real-time, , cluster, TDMA, scheduling, wireless sensor networks.

1. Introduction

Real-time communication in Wireless Sensor Networks (WSNs) is increasing day by day due to their significant importance in a wide range of applications. Example applications include health monitoring, disaster management, seismic monitoring, fire monitoring system, industrial process monitoring and control. In all these systems, sensed data have to reach the monitoring station within specific time period or before the expiration of deadline [19, 3, 5, 14, 10].

According to this demand, scheduling of real-time data at Medium Access Control (MAC) layer becomes important and challenging where efficient utilization of the channel is also required to deliver real-time data within deadline by considering the constraints such as interference relations and priority of flows [8]. The existing techniques to handle the channel access in WSNs are mainly divided into contention-based and contention-free schemes. In contention-based scheme, collision occurs when two or more nodes transmit on the channel at the same time. Consequently, it results in unpredictable additional delay that is inappropriate for real-time application. Compared to contention-based approach, contention-free MAC schemes are mostly predictable in term of transmission delay so they are suitable for real-time communications [15, 1]. Among diverse contention-free MAC approaches, many research works have focus on TDMA method.

Until now, TDMA scheduling for real-time communication have been well studied in [16, 9, 2]. In WirelessHART network [16], real-time packets are scheduled using TDMA. Since this scheduling algorithm allows only one transmission in a time slot by not providing spatial reuse, it limits the scalability of such approach. On the other hand, TDMA scheduling algorithm in [9] is proposed to allocate time slots based on requests and deadline of nodes. This scheduling works where the nodes are one hop away from the destination. However, in some situations, all nodes cannot transmit directly to base station. Furthermore, since the mentioned schemes not only have scalability problem but also support one-hop scheduling, this weakness leads to our new cluster-based TDMA scheduling algorithm in [2]. More detailed, our proposed scheduling algorithm divides intra-cluster time slots into two separate slots: IntraSend and IntraRecv. Real-time flows generated from a cluster are guaranteed to use IntraSend time slots, while those aiming at destination at the cluster are scheduled in IntraRecv time slots. But, due to the static slot allocation, the intra-cluster scheduling algorithms cannot utilized available empty slots. So, some flows cannot be delivered within the deadline due to static allocation.

In order to solve mentioned problem by extending the our previous work, in this paper, we propose adaptive TDMA scheduling algorithm under a cluster architecture. In the proposed scheme, if the clusters that have either IntraSend or IntraRecv flows, they do not distinguish IntraSend and IntraRecv scheduling. It implies that flows in a cluster can make use of two time slots together. But, in the previous scheme, these time slots cannot be occupied by flows due to static allocation. That is, an adaptive scheduling scheme can utilize the intra-cluster time slots according to the states of the flows dynamically. Finally, simulation results show that proposed schemes have higher number of flows admitted and delivered within the deadline than the previous work.

The rest of the paper is organized as follows. Related work is described in the section 2. the system model and notation is introduced in the section 3. In the section 4, we explain the proposed scheduling framework with example. Section 5 provides performance evaluation and section 6 concludes the paper.

2. Related Work

In this section, we briefly present the TDMA scheduling algorithms under cluster architecture. Cluster-based TDMA scheduling with different objectives have been proposed for WSNs in [13, 11, 6, 17, 7, 2, 12]. In [13], cluster-based data collection scheme was proposed where cluster header(CH) can directly access the sink. The intra-cluster communication are TDMA-based and CH in one-hop receives the data from their members. However, even though they separately considered the inter-cluster and intra-cluster delay in this scheme, the end-to-end delay was not investigated. In [11], coloring based TDMA scheduling algorithm was proposed to avoid the interference and achieve the real-time performance. The intra-cluster delay decreases by scheduling maximum independent sets of nodes in the same slots. In [6], cluster-based efficient data reporting control scheme was proposed. For intra-cluster data gathering, only few member nodes were selected as reporting node to meet the required throughput and save energy in each round. Moreover, for inter-cluster data reporting, delay bounded data were routed by applying the shortest path algorithm with the help of combination between CHs and member nodes. In the point of CH selection, the authors [12] addressed a novel selection algorithm to maximize the lifetime for motion detection. But, the real-time communication issues were not mentioned in this paper. Another cluster-based TDMA scheme to accomplish optimized energy efficiency and minimum delay was proposed in [17]. This scheme reduced the end-to-end latency by using the slot reuse concept. In [7], cluster-based unique convergecast scheduling problem of information gathering was investigated. All nodes in a cluster has 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 our previous work [2], real-time flows were scheduled by using TDMA time slots under cluster architecture. The time slots were divided into three different slots namely IntraSend, InterComm and IntraRecv slots. In IntraSend slots, real-time flows were scheduled from source nodes to its CHs. In InterComm time slots, real-time flows from source CHs were scheduled to the destination CHs. Finally, in IntraRecv time slots, flows from destination CHs were transmitted to the destination nodes. Compared to other approaches, our scheme performs flow scheduling with the priority of flows where the inter-cluster communication consists of CHs that are less-resource constraints. However, due to static scheduling, intra-cluster scheduling cannot utilize empty slots adaptively.

3. System Model and Notations

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 $\overrightarrow{uv}$ indicates that node u can send data to node v.

When nodes communicate each other, two types interference such as primary and/or secondary interference, may occur [18]. 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.

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 as shown in Figure 1. By using IntraSend slots, member nodes send real-time packets to their CHs. While a CH transmits flows to the destination CH in InterComm slots, the flows from destination CH are transmitted to destination node in IntraRecv slots .

3.2. Assumptions

The following assumptions have been made for the proposed network model.

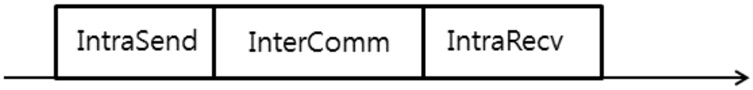


Fig. 1. Minor frame

(1) The nodes are randomly deployed and belong to one CH that is already known.
(2) All nodes should send neighboring as well as flow information (flow id, source-destination pair, deadline) to its CH at the initial phase and CHs sends results of IntraSend scheduling as well as information of neighboring CHs to one centralized point. The centralized point is already known to CHs at the deployment phase.
(3) A centralized point runs proposed scheduling algorithm and distributes time slots to all CHs.
(4) The routing path of nodes denoted by φ which is computed by centralized point through the shortest path algorithm. Any change in the flow information should be notified to CH as well as centralized point in order to trigger new allocation.
(5) Signal of CH is strong as much as possible to communicate with other neighbor CHs.
(6) There is no link between member nodes located in different clusters.
(7) There is no interference among different clusters.
(8) If two nodes are within predetermined distance, then they can communicate and interference with each other.
(9) The flows are established in case that both source and destination belong to different clusters.

3.3. Notations

The TDMA frame number is represented by $frame_{minor}$ while the number of slots in a frame is denoted by N_{minor}. In each frame, N_{IS}, N_{IC}, N_{IR} denotes the number of time slots for IntraSend(IS, InterComm(IC) and IntraRecv(IR) scheduling, respectively. In IntraSend scheduling for flow F_i, the arrival time to the source CH is denoted by a_i^{SH} while the arrival time to the destination CH is represented by a_i^{DH} in InterComm scheduling. Finally, arrival time to destination nodes of the flow F_i is defined by a_i in IntraRecv scheduling. The notation and equation number to include each variable are shown in Table 1.

4. The Proposed Real-Time Flow Scheduling

4.1. Procedure

Our proposed TDMA scheduling algorithm consists of IntraSend, InterComm and IntraRecv scheduling algorithms. These algorithms are divided into initialization and scheduling parts. In the initialization part of IntraSend and IntraRecv algorithms, each flow shares their information with their CHs. These information include the flow number and their interference information.

Table 1. Notations

Notation	Meaning	Equation
$eframe_i^{IS}$	The local frame number of F_i in IntraSend scheduling	(1)
a_i^{SH}	The global scheduling frame number of $eframe_i^{SH}$	(1)
$rframe_i^{IC}$	The local frame number of F_i' release time in InterComm scheduling	(2)
$eframe_i^{IC}$	The local frame number of F_i in InterComm scheduling	(3)
a_i^{DH}	The global scheduling frame number of $rframe_i^{IC}$	(3)
$rframe_i^{IR}$	The local frame number of F_i' release time in IntraRecv scheduling	(4)
$eframe_i^{IR}$	The local frame number of F_i in IntraRecv scheduling	(5)
a_i^{dest}	The global scheduling frame number of $eframe_i^{IR}$	(5)

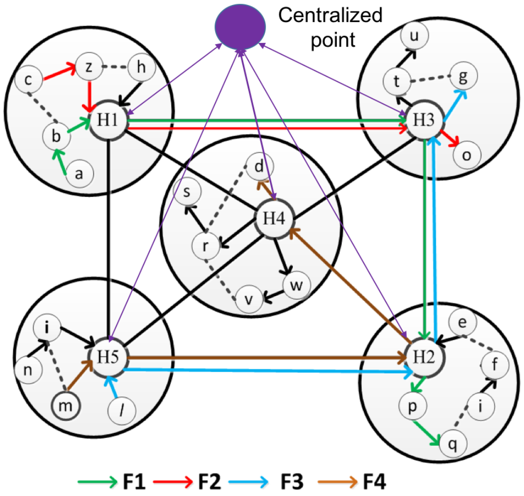


Fig. 2. An example of cluster-based real time flows

For example, in Figure 2 for initialization of IntraSend algorithm, F_1 and F_2 share flow number and their interference with CH H_1. Similarly, for IntraRecv algorithm, F_2 and F_3 share their flow number as well as interference with CH H_3. Through the initialization of InterComm algorithm, each CH shares their results with centralized point. Next, in the scheduling part, IntraSend algorithm is used to schedule the flows from source nodes to their CHs while flows from one cluster to other clusters are scheduled by InterComm algorithm. Finally, the flows from the CH to destination node is scheduled by the IntraRecv algorithm.

Since some clusters are likely to have only IntraSend or IntraRecv flows, therefore, these clusters do not need to differentiate intra-cluster scheduling but can use both time slots for their intra-flows. Based on this analysis, a new adaptive scheduling scheme is proposed. For a given set of real-time flows, each cluster counts the number of IntraSend and IntraRecv flows. If a cluster has only IntraSend flow (e.g. H1 and H5 in Figure 2),

Flow#	IntraSend		InterComm			IntraRecv		
	ri	1	a_i^{SH}	2	3	a_i^{DH}	4	a_i
F1	0	a→b	1				b→H1	4
F2	0						c→z	4
F3	0	*l*→H5	1	H5→H2	H2→H3	3	H3→g	4
F4	0							

(a)

Flow#	IntraSend		InterComm			IntraRecv		
	ri	5	a_i^{SH}	6	7	a_i^{DH}	8	a_i
F1	4			H1→H3	H3→H2	7	H2→p	8
F2	4	z→H1	5					
F3								
F4	4	m→H5	5	H5→H2		6		

(b)

Flow#	IntraSend		InterComm			IntraRecv		
	ri	9	a_i^{SH}	10	11	a_i^{DH}	12	a_i
F1	8	p→q	9					
F2				H1→H3		10	H3→O	12
F3								
F4					H2→H4	11	H4→d	12

(c)

Fig. 3. Example of Proposed Adaptive scheduling

the IntraSend scheduling of the cluster uses both N_{IS} and N_{IR} time slots. Similarly, if a cluster has only IntraRecv flows, the IntraRecv scheduling of the cluster uses both N_{IS} and N_{IR} time slots (e.g. H3 in Figure 2). The flowchart of adaptive scheduling is shown in Figure 4. In addition, We also propose another adaptive scheduling scheme where clusters that have both IntraSend and IntraRecv flows can utilized each other slots. The details of adaptive scheduling are discussed in the following subsections.

4.2. Proposed Adaptive Scheduling

In this scheduling, the intra-cluster slots are utilized adaptively according to the state of the intra-cluster flows. Because the number of flows scheduled in the frame is known at the beginning of each IntraSend or IntraRecv frame, if there is no flow in IntraSend frame, intraRecv flows are scheduled in this frame instead of IntraSend. Similarly, intraSend flows can be scheduled in IntraRecv frame in case of no IntraRecv flows. In order to support this scheduling, three scheduling algorithms are demanded to perform sequently. At the end of each scheduling for the frame size, the number of intra-cluster flows are updated so that the next intra-cluster scheduling algorithm can make use of corresponding information.

The sequence of adaptive scheduling algorithm is shown in Figure 4. As shown in for intra-cluster scheduling result, IntraSend and IntraRecv scheduling are performed if their flows are ready. On the other hand, if there is no ready flow in IntraSend frame, then IntraRecv flows are scheduled instead. In the same approach, IntraSend flows are scheduled if IntraRecv flows are not ready in corresponding frame.

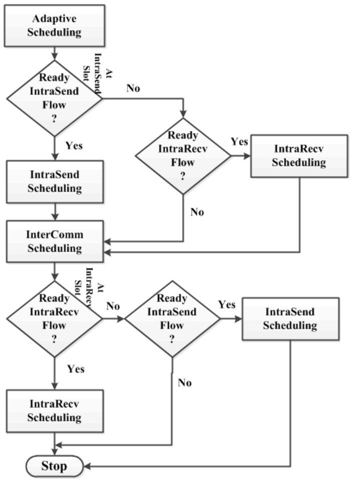


Fig. 4. Proposed Scheduling Flowchart

4.3. Example and Algorithm of Proposed Adaptive Scheduling

We explain the proposed scheduling with an example for four real-time flows in Figure 2. These flows are scheduled in a sequence of IntraSend, InterComm and IntraRecv scheduling at each frame through proposed adaptive scheduler as shown in Figure 4. The N_{IS}, N_{IC}, N_{IR} is set to 1, 2, and 1, respectively.

IntraSend Scheduling The pseudo-code of IntraSend scheduling is described in Algorithm 1. The flows are scheduled in the cluster for N_{IS} slots $(line\ 5)$. Each CH manages the IntraSend scheduling result by calling the function IA-FP-Scheduling 4 $(line\ 10)$. After scheduling, then expected arrival time of flows that is required to reach their CHs are computed from the result of IntraSend scheduling $(line\ 13)$.

As an example in Figure 3(a), according to intraSend scheduling of C_1, F_1 are scheduled first then F_2 is next turn. As transmission $\overrightarrow{ab}$ of F_1 is interferencing with transmission $\overrightarrow{cz}$ of F_2, so transmission $\overrightarrow{cz}$ is blocked because the priority of F_2 is lower than F_1. Similarly, for cluster C_5, transmission $\overrightarrow{lH5}$ of F_3 is conflicting with $\overrightarrow{mH_5}$ of F_4, therefore

Algorithm 1 IntraSend Scheduling $(G_C, \mathcal{F}, N_{IS})$

```
/* $G_C = (V_C, E_C)$: the graph of cluster $C$ */
for each cluster $C_i$ do
    for each flow $F_i \in \mathcal{F}$ do
        $F_k \leftarrow F_i$ by using the CH for $dst_k$
        $\varphi_k \leftarrow$ a path from $src_i$ to the CH in $\varphi_i$
        for $k$ from 1 to $|\varphi_k|$ and $k \leq N_{IS}$ do
            $\overrightarrow{uv} \leftarrow$ the $k$-th edge in $\varphi_i$
        endfor
        $\mathcal{F}_C \leftarrow \mathcal{F}_C \cup (r_i, F_k, \overrightarrow{uv})$
    endfor
    $\mathcal{T}_{IS_C} \leftarrow$ IA-FP-Scheduling $(G_C, \mathcal{F}_C)$
endfor
for each flow $F_i$ do
    Calculate $r_i^{SH}$ from the result of $\mathcal{T}_{IS_C}$
endfor
return $\mathcal{T}_{IS_C}$
```

$\overrightarrow{mH5}$ is blocked according to priority. So, only F_3 can be reached its CH. And, its arrival time to source CH is calculated from result of IntraSend scheduling by equation (1) where $eframe_i^{IS}$ is the frame number of last transmission of F_i in the IntraSend scheduling. The CH schedules each flows into the local scheduling table starting from zero. Thus, $eframe_i^{IS}$ is the local scheduling frame number of F_i resulting from IntraSend scheduling so that the arrival time to cluster head in the global frame should be calculated. Equation (1) converts the local scheduling frame number into the arrival frame number in global scheduling table.

$$a_i^{SH} = \left\lfloor \frac{\mathit{eframe}_i^{IS}}{N_{IS}} \right\rfloor \times N_{minor} + \mathit{eframe}_i^{IS} \bmod N_{IS} + 1 \tag{1}$$

InterComm Scheduling In this scheduling, the flows that arrive to the source CH are transmitted to the destination CH. For this purpose, we build a sub-graph G_{header} consisting of only CHs. The InterComm scheduling is used to allocate N_{IC} slots and its scheduling result is shared among CHs. As shown in Algorithm 2, the routing path contains only the sub-path from source CH to destination CH $(line\ 5)$. Each flow is scheduled for N_{IC} slots $(line\ 6)$. The function IA-FP-scheduling 4 is used to adjust flows for InterComm scheduling $(line\ 11)$. After scheduling, the expected arrival time of flows is obtained from the result of InterComm scheduling $(line\ 13)$. In InterComm scheduling, this arrival frame number, a_i^{SH}, is again converted into the local scheduling frame of the InterComm scheduling node in order to schedule the flow after arrival at the CH, which is derived by Equation (2).

$$\mathit{rframe}_i^{IC} = \begin{cases} \left\lfloor \frac{a_i^{SH}}{N_{minor}} \right\rfloor \times N_{IC} & \text{if } a_i^{SH} \neq \emptyset. \\ N_{IC} \times frame_{minor} & \text{if } a_i^{SH} = \emptyset. \end{cases} \tag{2}$$

By this approach, only F_3 is arrived at CH at time 1. So, both $\overrightarrow{H5H2}$ and $\overrightarrow{H2H3}$ can be scheduled at frame 1 in InterComm scheduling as shown in Figure 3(a). The arrival time to the destination CH of flow is determined by Equation (3). Similar to Equation (2), a_i^{DH} and a_i^{dest} in Equation (3) and (5) correspond to the global frame number at which flow F_i is delivered in the destination cluster and the destination node, respectively.

Algorithm 2 InterComm Scheduling $(G_{header}, \mathcal{F}, N_{IC})$

```
/* $G_{header}$: the graph consisting of only CHs */
for each flow $F_i \in \mathcal{F}$ do
    $F_k \leftarrow F_i$
    $src_k \leftarrow$ the source CH of $F_i$
    $dst_k \leftarrow$ the destination CH of $F_i$
    $\varphi_k \leftarrow$ a path from $src_k$ to $dst_k$ in $\varphi_i$
        for $k$ from 1 to $|\varphi_k|$ and $k \leq N_{IC}$ do
            $\overrightarrow{uv} \leftarrow$ the $k$-th edge in $\varphi_i$
        endfor
    $\mathcal{F}_C \leftarrow \mathcal{F}_C \cup \{(\mathit{rframe}_i^{IC}, F_k, \overrightarrow{uv})\}$
endfor
$\mathcal{T}_{IC} \leftarrow$ IA-FP-Scheduling $(G_{header}, \mathcal{F}_C)$
for each flow $F_i$ do
    Calculate $r_i^{DH}$ from the result of $\mathcal{T}_{IC}$
endfor
return $\mathcal{T}_{IC}$
```

$$a_i^{DH} = \left\lfloor \frac{\mathit{eframe}_i^{IC}}{N_{IC}} \right\rfloor \times N_{minor} + \mathit{eframe}_i^{IC} \bmod N_{IC} + N_{IS} + 1 \tag{3}$$

IntraRecv Scheduling In IntraRecv scheduling, the flows arrived at the destination CH are transmitted to destination node. Like IntraSend scheduling, each cluster manages the IntraRecv scheduling result through the pseudo-code shown in Algorithm 3. The scheduling is done by calling the function IA-FP-scheduling 4 $(line\ 11)$. Similar to previous example, the release time slots of IntraRecv scheduling flows can be derived from the arrival time of F_i at destination CH by Equation (4).

$$\mathit{rframe}_i^{IR} = \begin{cases} \left\lfloor \frac{a_i^{DH}}{N_{minor}} \right\rfloor \times N_{IR} & \text{if } a_i^{DH} \neq \emptyset. \\ N_{IR} \times frame_{minor} & \text{if } a_i^{DH} = \emptyset. \end{cases} \tag{4}$$

Interestingly, in Figure 3(a) for cluster C_1, no IntraRecv flows are ready to be scheduled. So, IntraSend transmission $\overrightarrow{bH_1}$ of F1 and $\overrightarrow{cz}$ of F_2 are scheduled in IntraRecv slots. Since only F_3 is ready for IntraRecv scheduling in cluster C_3, its transmission $\overrightarrow{H_3g}$ is scheduled in corresponding slot. The flows that arrive to destination node are determined by Equation (5).

```
**Algorithm 3** IntraRecv Scheduling $(G_C, \mathcal{F}, N_{IR})$
/* $G_C = (V_C, E_C)$: the graph of cluster $C$ */
for each cluster $C_i$ do
    $\mathcal{F}_C \leftarrow \varnothing$
    for each flow $F_i \in \mathcal{F}$ do
        $F_k \leftarrow F_i$ by using the CH for $src_k$
        $\varphi_k \leftarrow$ a path from the CH to $dst_i$ in $\varphi_i$
        for $k$ from 1 to $|\varphi_k|$ and $k \leq N_{IR}$ do
            $\overrightarrow{uv} \leftarrow$ the $k$-th edge in $\varphi_i$
        endfor
        $\mathcal{F}_C \leftarrow \mathcal{F}_C \cup \{(rframe_i^{IR}, F_k, \overrightarrow{uv})\}$
    endfor
    $\mathcal{T}_{IR_C} \leftarrow$ IA-FP-Scheduling $(G_C, \mathcal{F}_C)$
    for each flow $F_i$ do
        Calculate ${r_i}^{a_i}$ from the result of $\mathcal{T}_{IR_C}$
    endfor
    return $\mathcal{T}_{IR_C}$
```

$$a_i^{dest} = \left\lfloor \frac{\mathit{eframe}_i^{IR}}{N_{IR}} \right\rfloor \times N_{minor} \quad + \mathit{eframe}_i^{IR} \bmod N_{IR} + N_{IS} + N_{IC} + 1 \tag{5}$$

In the next $frame_{minor}$, the flows are scheduled as shown in the following Figure 3(b). Finally, in Figure 3(c), in C_2, since there is no ready flow for IntraSend scheduling, IntraRecv transmission $\overrightarrow{pq}$ of F_1 can be scheduled in the IntraSend slots.

After IntraRecv scheduling, the release frame for IntraSend flows that are not delivered to the destination node can be found by the Equation (6) while their release IntraSend slots are calculated by Equation (7).

$$\mathit{rframe}_i^{IS} = \begin{cases} \left\lfloor \frac{a_i}{N_{minor}} \right\rfloor \times N_{IS} & \text{if } a_i \neq \emptyset. \\ \\ N_{IS} \times frame_{minor} & \text{if } a_i = \emptyset. \end{cases} \tag{6}$$

$$r_i = \left\lfloor \frac{\mathit{eframe}_i^{IS}}{N_{IS}} \right\rfloor \times N_{minor} \quad + \mathit{eframe}_i^{IS} \bmod N_{IS} \tag{7}$$

The Algorithm 4 presents how to schedule real-time flows with consideration of interference and priority of flows. The IntraSend, InterComm and IntraRecv algorithms perform scheduling of their flows by calling Algorithm 4. The scheduling algorithm finds the first possible frame for a transmission $\overrightarrow{uv}$ $(line\ 5)$. When the flow interferes with one with higher priority, it is blocked and delayed by one frame $(line\ 8 - 11)$.

Algorithm 4 IA-FP-Scheduling (G, $\mathcal{F}$)

```
$/ * G = (V, E), \mathcal{F} = \{f_i(r_i, F_i, \varphi_i)\}$
Initialize TDMA scheduling table $\mathcal{T}$ with $\varnothing$.
for $i$ from $1$ to $N_F$ do
  $frame \leftarrow r_i$
  for $j$ from $1$ to $|\varphi_i|$ do
    $\overrightarrow{uv} \leftarrow$ the $j$-th edge in $\varphi_i$
    do
      $schedulable \leftarrow true$;
      for $k$ from $1$ to $i - 1$ do
        if $interfere$ ($\overrightarrow{uv}$, $\mathcal{T}[j][frame]$) $= true$ then
          $schedulable \leftarrow false$;
      if $schedulable = false$ then $frame \leftarrow frame + 1$;
    while ($schedulable = false$);
    $\mathcal{T}[i][frame] \leftarrow \overrightarrow{uv}$;
    $frame \leftarrow frame + 1$;
  endfor
endfor
return $\mathcal{T}$
```

5. Performance Evaluation

5.1. Simulation Environment

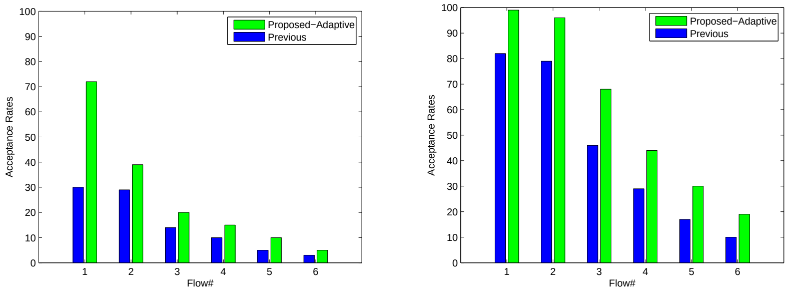
We have developed simulation program to evaluate the proposed scheduling algorithm. We used the GENSEN tool [4] to generate the network topology where about 50 nodes are generated randomly in $100m \times 100m$ area. The number of clusters is varied from 4 to 8 while each CH is selected as a node located near the center of the cluster. The simulation result provides the ratio of 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 5 hops and then $M_{deadline} = 2$, deadline is set to $10(= 5 \times 2)$.

To analyze the performance of our proposed scheduling algorithm, we compare the proposed-adaptive and previous scheme in [2] based on four metrics i.e. impact of deadline, impact of flows, impact of clusters and impact of intra-cluster slots. For simplicity, we refer proposed-adaptive as adaptive in the following sections.

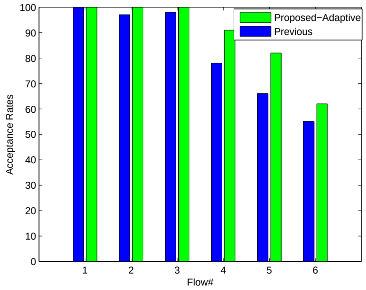
5.2. Impact of Deadline

In Figure 5, we analyzed the impact of the deadline on each flow acceptance rate in all comparative schemes. We take a short deadline from 1 to 3 while fix the numbers of flows as 6 in 6 clusters.

As shown in Figure 5(a)-(c), the acceptance rate of adaptive scheme is higher than the previous one because the IntraSend and IntraRecv scheduling in previous scheme cannot utilize empty slots. Due to this wastage of slots, more flows with shorter deadline miss their deadlines. Compared to the previous scheme, the adaptive scheme reveals higher acceptance rate. This is because intra-cluster scheduling in previous scheme cannot use each other empty slots for clusters that have both IntraSend and IntraRecv flows. However, in adaptive scheme, although clusters have both IntraSend and IntraRecv flows, but if only one type of the flow is not ready at that time, then the other type of flows can make use of both IntraSend and IntraRecv slots.



(a) deadline = 1 (b) deadline = 2



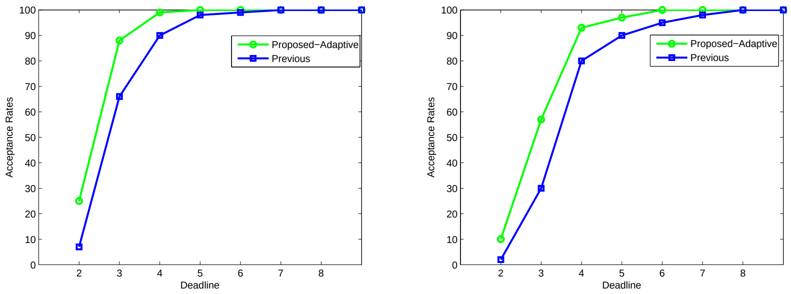
(c) deadline = 3

Fig. 5. Simulation results w.r.t deadline

5.3. Impact of number of flows

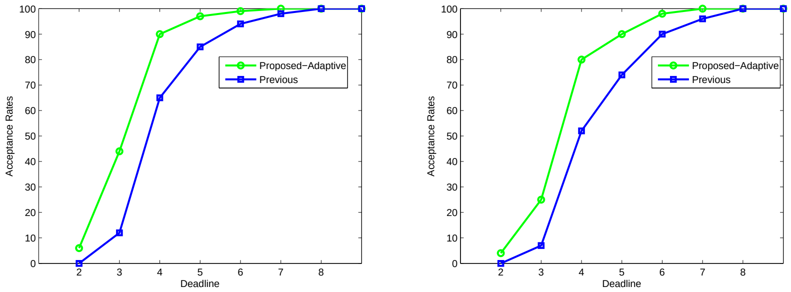
In this section, we have analyzed the number of flows meeting deadline in comparative schemes by varying the flow numbers from 4 to 8. For each flow case, we run 100 random cases and count the number of cases where all flows are scheduled in their deadlines. The number of clusters are fixed to 6 while deadlines are vary from 2 to 8.

As shown in Figure 6, as the number of flows increases the acceptance rate decreases due to conflict of flow scheduling in all schemes. By analyzing the result, we can figure out that the previous scheme accommodates less number of flows as compared to adaptive schemes. The reason is that flows are not completely scheduled in the allocated slots as the number of flows increases in the previous scheme due to insufficient slots or high priority flow interference. Therefore, these flows are to be scheduled in the next frame corresponding slots, hence less flows are scheduled at short deadline. Compare to previous scheme, intra-cluster scheduling in adaptive schemes can make use of each other slots. So in adaptive scheme, if IntraSend flows in a cluster do not complete its scheduling in the allotted IntraSend slots, then IntraRecv slots can be utilized by IntraSend flows if its IntraRecv flows are not ready.



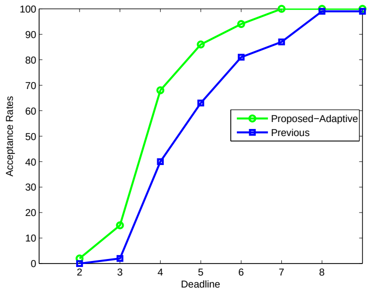
(a) # of flows = 4

(b) # of flows = 5



(c) # of flows = 6

(d) # of flows = 7

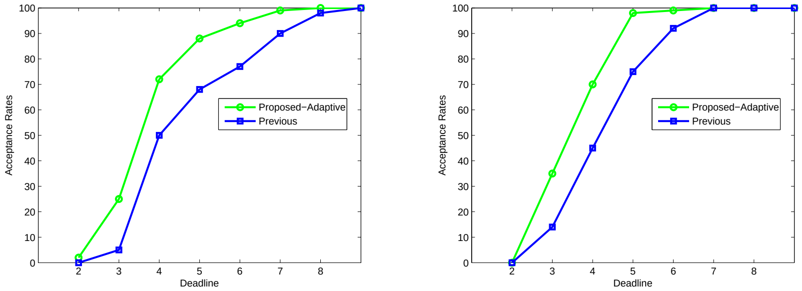


(e) # of flows = 8

Fig. 6. Simulation results w.r.t flow

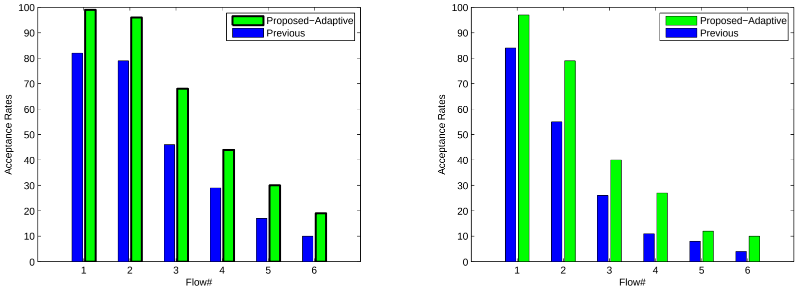
5.4. Impact of number Clusters

In Figure 7, we analyze flows delivered within the deadline as a function of number of clusters. For this purpose, the clusters are varied from 4 to 8 while number of flows is

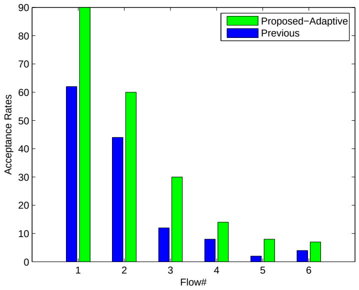


(a) # of clusters = 4 (b) # of clusters = 8

Fig. 7. Simulation results w.r.t cluster



(a) sequence = 1:2:1 (b) sequence = 2:2:1



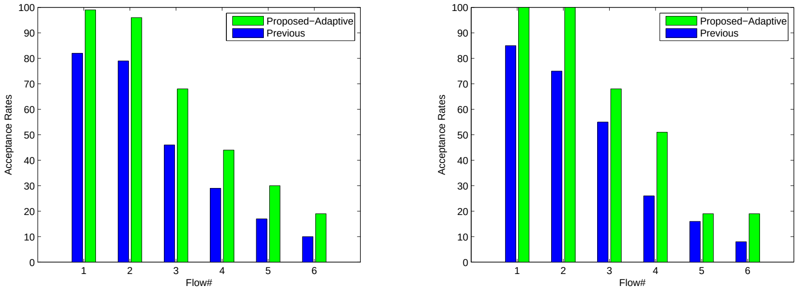
(c) sequence = 3:2:1

Fig. 8. Simulation results w.r.t IntraSend

fixed to 6. We run 100 random cases for each cluster number and count the number of cases where all flows are scheduled in their deadlines.

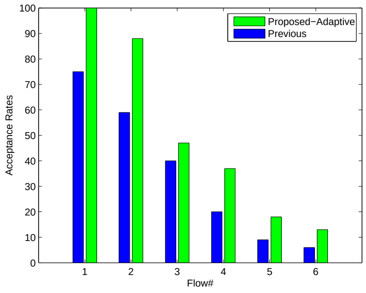
As show in Figure 7, if lower number of clusters that have high intra-cluster interference is assumed, less numbers of flows are scheduled in the available intra-cluster slots. But, as the number of cluster increases, the intra-cluster flows interference is reduced. Therefore, more flows are scheduled in the intra-cluster slots. For eight clusters, the acceptance rate decreases due to high interference between the CHs.

Compared to the previous scheme, the adaptive scheme shows higher acceptance rate. This is because the allocated intra-cluster slots are insufficient for clusters that have high intra-cluster interference. Therefore, in this case, the adaptive scheme takes the advantage of intra-cluster schedule. For instance, in adaptive scheme for clusters that have high IntraSend flows interference, then these flows cannot be completely scheduled in allocated IntraSend slots so it is possible to use IntraRecv slots. On the other hand, the intra-cluster scheduling can only utilize their allocated slots and do not utilize slots of one another for clusters that have high intra-cluster interference in previous scheme. Due to this problem in previous scheme, intra-cluster flows waste each other empty slots and more flows miss their deadlines.



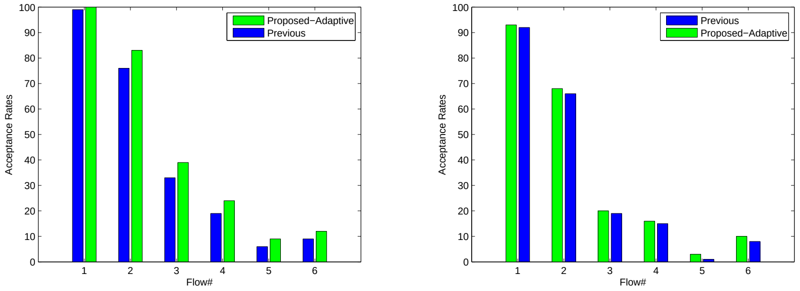
(a) sequence = 1:2:1

(b) sequence = 1:2:2



(c) sequence = 1:2:3

Fig. 9. Simulation results w.r.t IntraRecv



(a) # of slots = 2:2:2 (b) # of slots = 3:2:3

Fig. 10. Simulation w.r.t intra-cluster results

5.5. Impact of IntraSend/IntraRecv time slots

To analyze the performance of intra-cluster scheduling of adaptive and previous scheme, the time slots for both IntraSend and IntraRecv scheduling are varied while number of flow is fixed to 6 for six clusters. In Figure 8, we show the analysis of the IntraSend scheduling of above comparative scheme when IntraSend time slots are varied from 1 to 3 while the IntraRecv time slots are fixed to 1. As shown in Figure 8(a)-(c), the acceptance rate of all schemes decreases as the number of IntraSend slots increases. This is because most of the IntraSend slots are unused when the flows already reach their source CHs. However, the adaptive scheme has higher acceptance rate than previous scheme. This comes to insufficient slot for IntraRecv scheduling which allocates one slot.

Since the IntraRecv scheduling cannot utilized the IntraSend slots in previous scheme due to static scheduling, more flows are likely to miss their deadlines. Similarly in Figure 9, for the analysis of IntraRecv scheduling of comparative scheme, the IntraRecv time slots are varied from 1 to 3 while IntraSend time slots are fixed to 1. As shown in acceptance rate of the adaptive IntraRecv scheduling, they reveal higher rate than the previous IntraRecv scheduling. Because the allocated IntraSend slot is not sufficient for IntraSend scheduling, so the adaptive IntraSend scheduling utilizes slots from IntraRecv slots while previous scheme wastes these IntraSend slots. As a result, more flows miss their deadlines in the previous scheme.

In Figure 10, IntraSend and IntraRecv have the same ratio of time slots. As the number of intra-cluster slots increases, performance of both schemes decreases because of the unused slots. Moreover, performance of proposed scheme becomes similar to previous scheme as the number of time slots increases. When allocated slots are sufficient for intra-cluster scheduling, proposed scheme and previous scheme do not utilize Intra-cluster slots.

6. Conclusions

In this paper, we proposed new adaptive TDMA scheduling algorithm under cluster based architecture in order to extend our previous work which is likely to waste some time slots due to static scheduling algorithm. Depending on the state of flow, some slots are allocated to other flows. Simulation results also were presented to prove the suitability of the proposed schemes in many different aspects. Related to this work, further research work will be focused on how to extend proposed scheme in more realistic model. Also, a new routing protocol considering sensor node's constraints will be concerned for proposed scheme.

Acknowledgments. This work was supported by Basic Science Research Program (2015R1D1A1 A01056979) through the National Research Foundation of Korea (NRF) funded by the Ministry of Education and Sur University College,Sur, Oman.

References

1. Akyildiz, I.F., Su, W., Sankarasubramaniam, Y., Cayirci, E.: A survey on sensor networks. IEEE Communications Magazine, 40(8), 102–114 (2002)
2. Ali, G., Kang, S.K., Kim, K.H., Kim, K.I.: Towards cluster-based real-time flow scheduling in interference-aware wireless sensor networks. In: Proceedings of the IEEE 16th International Conference on Computational Science and Engineering (CSE),. pp. 523–530. IEEE, Sydney, Australia (2013)
3. Bellavista, P., Cardone, G., Corradi, A., Foschini, L.: Convergence of manet and wsn in iot urban scenarios. IEEE Sensors Journal, 13(10), 3558–3567 (2013)
4. Camilo, T., Silva, J., Rodrigues, A., Boavida, F.: Gensen a topology generator for real wireless sensor networks deployment. Santorini Island, Greece (2007), [Online]. Available:http://link.springer.com/chapter/10.1007
5. Cao, X., Chen, J., Xiao, Y., Sun, Y.: Building-environment control with wireless sensor and actuator networks: Centralized versus distributed. IEEE Transactions on Industrial Electronics, 57(11), 3596–3605 (2010)
6. Choe, H.J., Ghosh, P., Das, S.K.: Qos-aware data reporting control in cluster-based wireless sensor networks. Computer Communications, 33(11), 1244–1254 (2010)
7. Choi, H., Wang, J., Hughes, E.A.: Scheduling for information gathering on sensor network. Wireless Networks, 15(1), 127–140 (2009)
8. Ergen, S.C., Varaiya, P.: Pedamacs: power efficient and delay aware medium access protocol for sensor networks. IEEE Transactions on Mobile Computing, 5(7), 920–930 (2006)
9. Fattah, H., Leung, C.: An overview of scheduling algorithms in wireless multimedia networks. IEEE Wireless Communications, 9(5), 76–83 (2002)
10. Hill, J., Szewczyk, R., Woo, A., Hollar, S., Culler, D., Pister, K.: System architecture directions for networked sensors. ACM SIGPLAN, 28(5), 93–104 (2000)
11. Kang, H., Zhao, Y., Mei, F.: A graph coloring based tdma scheduling algorithm for wireless sensor networks. Wireless Personal Communications, 72(2), 1005–1022 (2013)
12. Lan, K., Chou, C., Wang, T., Li, M.: On the efficiency of cluster-based approaches for motion detection using body sensor networks. Computer Science and Information Systems, 8(4), 1051–1071 (2011)
13. Lotfinezhad, M., Liang, B., Sousa, E.S.: Adaptive cluster-based data collection in sensor networks with direct sink access. IEEE Transactions on Mobile Computing, 7(7), 884–897 (2008)
14. Mottola, L., Picco, G.P., Ceriotti, M., Guna, S., Murphy, A.L.: Not all wireless sensor networks are created equal: A comparative study on tunnels. ACM Transactions on Sensor Networks, 7(2), 15:1–15:33 (2010)
15. Rhee, I., Warrier, A., Aia, M., Min, J., Sichitiu, M.L.: Z-mac: A hybrid mac for wireless sensor networks. IEEE/ACM TRANSACTIONS ON NETWORKING, 16(3), 511–524 (2008)

16. Saifullah, A., Xu, Y., Lu, C., Chen, Y.: Real-time scheduling for wirelesshart networks. In: IEEE 31st Real-Time Systems Symposium (RTSS),. pp. 150–159. IEEE, San Diego, California (2010)
17. Shi, L., Fapojuwo, A.: Tdma scheduling with optimized energy efficiency and minimum delay in clustered wireless sensor networks. IEEE Transactions on Mobile Computing, 9(7), 927–940 (2010)
18. Subramanian, R., Lloyd, E.L.: Scheduling algorithms for multihop radio networks. IEEE/ACM TRANSACTIONS ON NETWORKING, 1(2), 166–177 (1993)
19. Torfs, T., Sterken, T., Brebels, S., Santana, J., van den Hoven, R., Spiering, V., Bertsch, N., Trapani, D., Zonta, D.: Low power wireless sensor network for building monitoring. IEEE Sensors Journal, 13(3), 909–915 (2013)

Gohar Ali received Ph.D degree from Gyeongsang National University, Jinju, Korea. He is currently with Sur University College,Oman. His research interests include sensor networks and real-time systems.

Kyong Hoon Kim received his B.S., M.S., and Ph.D. degrees in Computer Science and Engineering from POSTECH, Korea, in 1998, 2000, 2005, respectively. Since 2007, he has been an associate professor at the Department of Informatics, Gyeongsang National University, Jinju, Korea. His research interests include real-time systems, Grid and Cloud computing, and security.

Ki-Il Kim received M.S. and Ph.D. degrees in Computer Science from ChungNam National University, Daejeon, Korea. He is currently with the Department of Informatics at Gyeongsang National University. His research interests include ad hoc and sensor networks.

Received: April 1, 2015; Accepted: January 8, 2016.