

Packet Aggregation Real-Time Scheduling for Large-Scale WIA-PA Industrial Wireless Sensor Networks

XI JIN, Shenyang Institute of Automation, Chinese Academy of Sciences and Key Laboratory of Networked Control Systems, Chinese Academy of Sciences, China
NAN GUAN, Hong Kong Polytechnic University, China
CHANGQING XIA, JINTAO WANG, and PENG ZENG, Shenyang Institute of Automation, Chinese Academy of Sciences and Key Laboratory of Networked Control Systems, Chinese Academy of Sciences, China

The IEC standard WIA-PA is a communication protocol for industrial wireless sensor networks. Its special features, including a hierarchical topology, hybrid centralized-distributed management and packet aggregation make it suitable for large-scale industrial wireless sensor networks. Industrial systems place large real-time requirements on wireless sensor networks. However, the WIA-PA standard does not specify the transmission methods, which are vital to the real-time performance of wireless networks, and little work has been done to address this problem.

In this article, we propose a real-time aggregation scheduling method for WIA-PA networks. First, to satisfy the real-time constraints on dataflows, we propose a method that combines the real-time theory with the classical bin-packing method to aggregate original packets into the minimum number of aggregated packets. The simulation results indicate that our method outperforms the traditional bin-packing method, aggregating up to 35% fewer packets, and improves the real-time performance by up to 10%. Second, to make it possible to solve the scheduling problem of WIA-PA networks using the classical scheduling algorithms, we transform the ragged time slots of WIA-PA networks to a universal model. In the simulation, a large number of WIA-PA networks are randomly generated to evaluate the performances of several real-time scheduling algorithms. By comparing the results, we obtain that the earliest deadline first real-time scheduling algorithm is the preferred method for WIA-PA networks.

CCS Concepts: • **Networks** → **Network protocol design**; *Network management*; Network reliability; Network performance analysis;

Additional Key Words and Phrases: Industrial wireless sensor networks, packet aggregation, real-time scheduling, multi-channel, TDMA

This work was supported in part by the National Natural Science Foundation of China under Grants 61502474 and 61533015, in part by the Youth Innovation Promotion Association of the Chinese Academy of Sciences.

Authors' addresses: X. Jin, Shenyang Institute of Automation, Chinese Academy of Sciences and Key Laboratory of Networked Control Systems, Chinese Academy of Sciences, 114 Nanta Street, Shenyang, 110016, China; email: xijin@ieec.org; N. Guan, Hong Kong Polytechnic University, Hong Kong, China; email: nan.guan@polyu.edu.hk; C. Xia, J. Wang, and P. Zeng, * Shenyang Institute of Automation, Chinese Academy of Sciences and Key Laboratory of Networked Control Systems, Chinese Academy of Sciences, 114 Nanta Street, Shenyang, 110016, China; emails: {xiachangqing, wangjintao, zp}@sia.cn.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

© 2018 Copyright held by the owner/author(s). Publication rights licensed to ACM.

1539-9087/2018/09-ART88 \$15.00

<https://doi.org/10.1145/3266228>

ACM Reference format:

Xi Jin, Nan Guan, Changqing Xia, Jintao Wang, and Peng Zeng. 2018. Packet Aggregation Real-Time Scheduling for Large-Scale WIA-PA Industrial Wireless Sensor Networks. *ACM Trans. Embed. Comput. Syst.* 17, 5, Article 88 (September 2018), 19 pages.
<https://doi.org/10.1145/3266228>

1 INTRODUCTION

Wireless sensor networks have been one of the most important parts of industrial systems. With the rise of smart factories, the scale of wireless sensor networks has been increasing. Recently, the research on large-scale industrial wireless sensor networks has attracted rapidly increasing attention [23].

WIA-PA [15], WirelessHART [14], and ISA100.11a [16] are IEC standards for industrial wireless sensor networks. WirelessHART and ISA100.11a adopt centralized resource management, which makes it easy to implement global optimization but hard to manage large-scale networks. WIA-PA is based on a hierarchical framework and supports a hybrid centralized-distributed management. Therefore, it has higher scalability than WirelessHART and ISA100.11a and is the only IEC standard that is suitable for large-scale industrial wireless sensor networks. In steel mills and oil fields, WIA-PA has been used [32, 33].

For industrial systems, real-time performance is essential. For example, in the cement manufacturing industry, the temperature information of rotary kilns is important. If a packet with a high-temperature warning is delayed or lost, the kilns can explode. To satisfy real-time requirements, the WIA-PA standard supports many techniques that can improve real-time performance, such as using dedicated time slots, aggregating packets, and transmitting on multiple channels. However, it does not specify the corresponding aggregation and scheduling methods. This makes networks more flexible, and users can design different methods to satisfy different requirements. In this article, we focus on how to aggregate and schedule packets to improve real-time performance. There is much related work on wireless sensor networks, e.g., References [3, 7, 12]. However, these are not suitable for WIA-PA networks because they cannot coordinately solve the aggregation and scheduling problems for the flows that have harmonic periods. Currently, related work on WIA-PA networks, e.g., References [20, 31], does not consider the scheduling problem, and the common practice in WIA-PA networks is to adopt best-effort methods that do not guarantee the real-time performance. Therefore, to deploy *real-time* applications on WIA-PA wireless networks, this article studies the real-time aggregation scheduling method for WIA-PA. More precisely, our contributions in this article are as follows:

First, to reduce resource requirements, original packets are aggregated into the minimum number of aggregated packets. Although there are plenty of aggregation techniques in previous works, none of them consider the real-time requirements of heterogeneous-period packets that are adopted in WIA-PA networks. We transform our aggregation problem into a bin-packing problem and analyze the traditional bin packing method based on WIA-PA networks. The traditional bin-packing method does not consider the temporal feature of packets. This leads to inappropriate aggregation timings and a decrease in the method performance. To make the traditional bin-packing method suitable for our problem, we propose a novel aggregation method that combines the traditional method with a real-time scheduling method. The simulation results indicate that our method outperforms the traditional bin-packing method, aggregating up to 35% fewer packets.

Second, we focus on how to schedule these aggregated packets under real-time constraints. There are some related works on real-time scheduling algorithms in wireless sensor networks. However, the available resources in WIA-PA networks are different from other wireless sensor

networks. Although the hierarchical framework supported by WIA-PA networks can improve the scalability, it increases the difficulty of coordinating inter-level resources. Previous works cannot be used without any modification. Therefore, we propose a method for transforming the WIA-PA model into a universal model and then adopt existing real-time scheduling algorithms to solve our problem. The performance comparison among these real-time scheduling algorithms indicates that EDF is the preferred method for WIA-PA networks. In addition, our aggregation method can help in scheduling algorithms such that the real-time performance is improved by up to 10%. The reason is that to satisfy the real-time constraints of all original packets, the real-time requirement of an aggregated packet is determined by the original packet with the highest real-time requirement, and our aggregation method aggregates high real-time packets together and prevents them from improving the real-time requirements of other aggregated packets. Thus, under the same real-time constraints of original packets, our aggregation method makes scheduling algorithms easy-to-find feasible solutions for aggregated packets.

2 RELATED WORKS

Wireless sensor networks are resource-constrained. Aggregating packets is a good way to improve their performance when resources are limited. Aggregation methods have been extensively studied in previous works. LEACH [12] is an outstanding protocol that considers data aggregation in cluster-based wireless sensor networks at a very early stage. There are several studies on how to improve the performance of LEACH, such as using routing protocols [3] and optimizing the choice of cluster heads [1]. On the other hand, some advanced technologies have been introduced to further optimize results, such as multiple input multiple output [10], network coding [25], opportunistic routing [27], and aerial access points [22]. However, these works did not consider the temporality and cannot be used in real-time networks.

The decision version of time-optimal data gathering on a single channel is NP-complete in a weak sense [7]. Although some works tried hard to improve the network performance [6, 11], the limited number of resources of a single channel restricts the performance improvement. Then, multiple channels were introduced. The work in Reference [28] presented a time-optimal multi-channel scheduling algorithm for binary-tree networks, and the work in Reference [30] proposed a time- and channel-optimal scheduling algorithm for linear networks. Based on the above two publications, the same authors took the impact of packet copying into account to enhance the channel utilization [29]. For mesh-topology networks, the work in Reference [18] proposed a method based on spanning trees to reduce end-to-end delay. In these works, all flows had the same period. The above mentioned works cannot be applied to heterogeneous periods supported by WIA-PA networks. The work in Reference [26] focused on heterogeneous periods and proposed scheduling algorithms to improve the real-time performance of wireless sensor networks. For the same network model, the work in Reference [17] not only optimized the real-time performance but also improved the channel utilization, and the work in Reference [8] solved the mobility-aware real-time scheduling problem. However, these works did not consider heterogeneous-period packet aggregation and real-time scheduling for WIA-PA networks.

3 WIA-PA OVERVIEW AND PROBLEM STATEMENT

In this section, we briefly introduce the WIA-PA networks. Details of the WIA-PA standard not shown in this article are irrelevant or orthogonal to the studied problem. A WIA-PA network is shown in Figure 1. The upper level is a mesh network and has a centralized network manager. The lower level includes many clusters with a star topology. Each cluster head is a node of the upper level. The centralized manager is only used for the upper network, and each cluster network is managed by its cluster head.

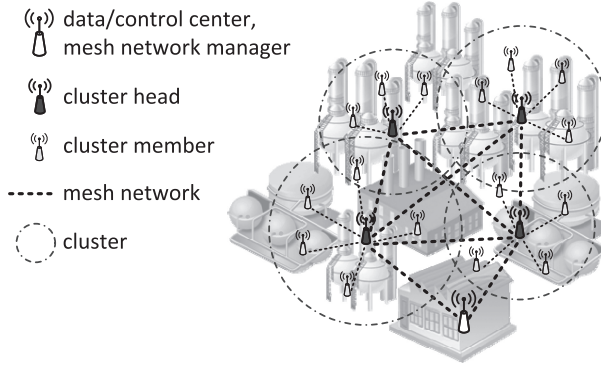


Fig. 1. A WIA-PA network.

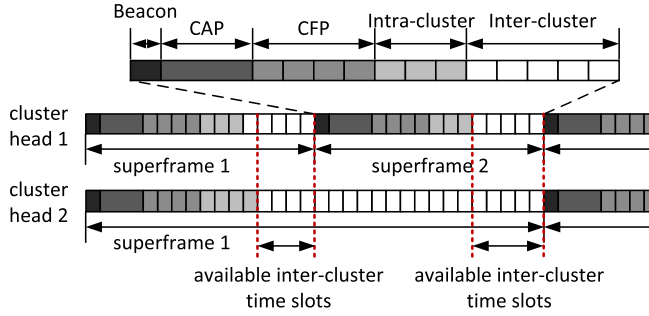


Fig. 2. Superframe structure.

The timing behavior of each cluster head for transmitting data packets follows a superframe structure (as shown in Figure 2), which begins with a synchronized beacon and then includes four parts: contention access period (CAP), contention-free period (CFP), intra-cluster communication period, and inter-cluster communication period. The CAP is based on the CSMA mechanism and is used for device joining. Before a node joins the network, it has no assigned resources and accesses the wireless channel only via the CSMA mechanism. To obtain network controllability, other parts are based on the TDMA mechanism. The CFP is reserved for mobile devices. The intra-cluster period and the inter-cluster period are used for intra-cluster communication and inter-cluster communication, respectively. The lengths of these periods can be configured in the protocol stack and cannot be changed online and remotely. In this article, the lengths of the CAP, CFP, and intra-cluster period in all clusters are given the same constant, and all other time slots belong to inter-cluster periods. Each cluster head repeatedly executes its superframe at a specific period, and different heads are allowed to have different periods, such as cluster heads 1 and 2 in Figure 2.

WIA-PA networks support multiple channels, thus, multiple superframes can run on different channels in parallel. However, a node cannot serve more than one transmission simultaneously. In this article, we do not consider dynamic device joining and mobile devices but focus only on how to use the intra- and inter-cluster periods to transmit sensing data from cluster members to the data center under real-time constraints.

Cluster members periodically sense data from the environment, and then use the intra-cluster period to send sensing data to their cluster head. Depending on the requirements of industrial systems, the periods of dataflows can be different. WIA-PA networks adopt harmonic periods (i.e.,

for any two dataflows, one's period is a multiple of the other's), which are also used in other industrial wireless sensor network standards. The sensing data are usually very small (e.g., 2 bytes of temperature data, 4 bytes of vibration data, and 32 bytes of diagnostic data). The transmission of a packet through a link must occupy a dedicated time slot and a dedicated channel no matter how small the packet is. Therefore, when a cluster head receives several small packets, it first aggregates them into a big packet and then sends the big packet instead of those small packets. In this way, more original small packets can be transmitted in resource-constrained wireless sensor networks. Note that in this article, packet aggregation involves putting several raw data into a packet but not to process them since the processing of data may discard some information required by industrial controls. Aggregated packets are sent to the data center via the mesh network using time slots of inter-cluster periods. In industrial systems, we consider only hard real-time networks, i.e., the time required for the sensing data to travel from the cluster members to the data center must not exceed the relative deadlines.

In the transmission process, two problems need to be solved: the aggregation problem in cluster networks and the scheduling problem in the mesh network. For the first problem, to transmit more raw data in the case of limited resources, as much data as possible should be aggregated into a packet. However, there are two restrictions: (1) the size of an aggregated packet must not exceed its upper bound; (2) the time spent to perform the aggregation process should not affect the real-time performance of the mesh network. Therefore, the first problem is how to aggregate harmonic-period packets into the minimum number of aggregated packets while not affecting the real-time performance of end-to-end transmissions (Section 4). For the second problem, the superframes of different clusters have different periods, and not all of the time slots of cluster heads can be used for inter-cluster transmissions. Only when two cluster heads are both in their inter-cluster periods can they communicate with each other. The available inter-cluster time slots in Figure 2 describe that scenario. Therefore, the second problem is how to schedule aggregated packets using the available inter-cluster time slots such that the required real-time performance can be satisfied (Section 5).

4 INTRA-CLUSTER AGGREGATION METHOD

In this section, we describe how to aggregate packets in a cluster. We first introduce the cluster model, and then introduce our aggregation method.

4.1 Cluster Model

A cluster includes a cluster head and several cluster members. The set of cluster members is denoted as $N = \{n_1, n_2, \dots\}$. Each cluster member n_i has a dataflow that is characterized by $f_i = \langle p_i, s_i \rangle$, and the flow set is $F = \{f_1, f_2, \dots\}$. The first packets of all dataflows are released at time 1, and then flow f_i periodically generates packets with a period p_i . The j th packet of flow f_i is denoted as τ_{ij} . τ_{ij} is released at time $p_i \times (j - 1) + 1$ and has to be sent to the data center before its deadline $p_i \times j$, i.e., in our network the period p_i is also the relative deadline. The payload size of original packets of flow f_i is s_i , and S is the upper bound of the payload size of an aggregated packet.

The cluster head maintains its superframe structure to execute periodically. All original packets are submitted during the intra-cluster period. Therefore, the superframe period must be the minimal period among all flows, i.e., the superframe period $q = \min_{f_i \in F} \{p_i\}$. If $q > \min_{f_i \in F} \{p_i\}$, some of the packets with deadlines smaller than q are not schedulable. An intra-cluster period includes C time slots. In this article, C is a given parameter. We will evaluate its impact on our algorithm in Section 7 and optimize it in our future work. When several original packets are aggregated into a new packet, the deadline of the aggregated packet is the earliest deadline among them. We

consider only the aggregation in a hyperperiod $H = LCM(p_1, p_2, \dots)$. After that, all aggregations are cyclically repeated. Since flows adopt harmonic periods, $H = \max_{f_i \in F} \{p_i\}$.

4.2 Problem Formulation

To explain our problem more clearly, we formulate it as a 0-1 ILP problem. We use the symbol δ_{ab} to denote the b th aggregated packet in the a th superframe. If $\rho_{ab} = 1$, then some packets are aggregated into the new packet δ_{ab} ; otherwise, there is no δ_{ab} . For the original packet τ_{ij} , if $c_{ija} = 1$, then the original packet is sent to the cluster head in the a th superframe; if $u_{ijab} = 1$, then the original packet is aggregated into the new packet δ_{ab} . Wireless sensor networks are resource-constrained, and one packet has to occupy a dedicated resource. Thus, the number of packets should be reduced such that the available resources can guarantee the requirements of all packets. Therefore, the objective of the aggregation problem is to find c_{ija} and u_{ijab} for each original packet such that the number of aggregated packets is minimum.

The formulation is as follows:

$$\text{obj. min} \quad \sum_{\forall a \in [1, \frac{H}{q}], \forall b \in [1, |F|]} \rho_{ab}.$$

The minimization problem should be subject to the following constraints:

- (1) *Uniqueness constraint*: Each packet is submitted and aggregated only once:

$$\begin{aligned} \forall i \in [1, |F|], \forall j \in \left[1, \frac{H}{p_i}\right], \quad & \sum_{\forall a \in [1, \frac{H}{q}]} c_{ija} = 1, \\ \forall i \in [1, |F|], \forall j \in \left[1, \frac{H}{p_i}\right], \quad & \sum_{\forall a \in [1, \frac{H}{q}], \forall b \in [1, |F|]} u_{ijab} = 1. \end{aligned}$$

- (2) *Usage constraint*: If $\exists u_{ijab} = 1$, then the aggregated packet δ_{ab} is used, i.e., $\rho_{ab} = 1$; otherwise, $\rho_{ab} = 0$:

$$\begin{aligned} \forall a \in \left[1, \frac{H}{q}\right], \forall b \in [1, |F|], \forall i \in [1, |F|], \forall j \in \left[1, \frac{H}{p_i}\right], \\ \rho_{ab} \geq u_{ijab}. \end{aligned}$$

- (3) *Timing constraint*: τ_{ij} is released at the $(\frac{p_i}{q} \cdot (j-1))$ th superframe and delivered to the data center before the $(\frac{p_i}{q} \cdot j)$ th superframe. Between them, τ_{ij} is submitted first, and then aggregated:

$$\forall \tau_{ij}, \frac{p_i}{q} \cdot (j-1) \leq \sum_{\forall a \in [1, \frac{H}{q}]} (c_{ija} \cdot a) \leq \sum_{\forall a \in [1, \frac{H}{q}]} \left(\left(\sum_{\forall b \in [1, |F|]} u_{ijab} \right) \cdot a \right) < \frac{p_i}{q} \cdot j.$$

- (4) *Size constraint*: The size of the aggregated packet cannot exceed the upper bound of the payload size:

$$\forall \delta_{ab}, \quad \sum_{\forall i \in [1, |F|], \forall j \in [1, \frac{H}{q}]} (u_{ijab} \cdot s_i) \leq S.$$

- (5) *Time slot constraint*: In each superframe, the number of submitted original packets cannot exceed the number of time slots in an intra-cluster period:

$$\forall a \in \left[1, \frac{H}{q}\right], \quad \sum_{\forall i \in [1, |F|], \forall j \in [1, \frac{H}{q}]} c_{ija} \leq C.$$

4.3 Aggregation Method

Although WIA-PA supports packet aggregation, it does not specify how to aggregate packets. The aggregation problem is similar to the bin-packing problem. The original packets sent by cluster members correspond to objects, and aggregated packets are bins. Their objectives are both to minimize the number of aggregated packets (bins). Therefore, the classical algorithms of the bin-packing problem can be used to solve our aggregation problem. This article focuses on an effective and efficient algorithm—First Fit (FF) puts each object into the first-fit bin and opens a new bin if the object does not fit into any open bins.

However, there are some differences between the aggregation problem and the traditional bin-packing problem. First, each packet has a lifetime, measured from its release time to its deadline. Second, the number of original packets that are submitted in a superframe cannot exceed C . These lead to the temporal features of the aggregation problem, i.e., before an aggregated packet is generated, the original packets that should be added to this aggregated packet must be generated and assigned dedicated time slots in the intra-cluster period. In addition, when the FF algorithm is used in our network, there is no strategy for specifying the order of submissions. To improve the real-time performance, we specify that short-deadline packets be submitted first. Therefore, the aggregation problem is different from the traditional offline bin-packing problem. For the traditional bin-packing problem, the approximation ratio of the online FF is 1.7 [9]. Meanwhile, because of the temporal feature, the approximation ratio of our problem approaches 2. Theorem 4.1 shows the approximation ratio of our problem.

THEOREM 4.1. *In our problem, the FF algorithm has a tight approximation ratio of 2.*

PROOF. We first prove 2 is an upper bound of the approximation ratio, and then use an example to show the approximation ratio can approach to 2 (i.e., it is tight).

Suppose that an optimal solution aggregates all original packets into μ aggregated packets. For the FF algorithm, we classify superframes into two sets V and W . The set V includes the superframes that generate only one aggregated packet, while other superframes are in the set W . There is no superframe that does not generate any aggregated packet. If this kind of superframe exists, then the original packets that have the smallest relative deadlines and are generated at these superframes cannot be sent to the upper mesh network before their deadlines. Therefore, $\mu \geq |W| + |V|$. For the w th superframe in the set W , we use $k_w + 1$ ($k_w \in \mathbb{Z}^+$) to denote the number of aggregated packets in this superframe. Thus, the approximation ratio \mathcal{R} is as follows:

$$\mathcal{R} = \frac{\sum_{\forall w \in W} (k_w + 1) + |V|}{\mu}.$$

The size of the aggregated packet in the v th superframe in the set V is denoted as A_v . A_w and B_w are the sizes of the second-smallest and the smallest aggregated packets in the w th superframe in the set W , respectively.

Then, we get that

$$\sum_{\forall v \in V} A_v + \sum_{\forall w \in W} (A_w \cdot k_w + B_w) \leq \mu \cdot S. \quad (1)$$

We know that $A_w + B_w > S$. If not, they can be aggregated together. Thus, $B_w > S - A_w$. Then, the second term of the left side of Equation (1) is as follows:

$$\begin{aligned} \sum_{\forall w \in W} (A_w \cdot k_w + B_w) &> \sum_{\forall w \in W} (A_w \cdot k_w + S - A_w) \\ &= \sum_{\forall w \in W} (A_w \cdot (k_w - 1) + S) = \sum_{\forall w \in W} (A_w \cdot (k_w - 1)) + S \cdot |W|. \end{aligned}$$

Therefore, Equation (1) is rewritten as

$$\sum_{v \in V} A_v + \sum_{w \in W} (A_w \cdot (k_w - 1)) + S \cdot |W| < \mu \cdot S. \quad (2)$$

Then, Equation (2) is divided by S on both sides. We obtain

$$\sum_{v \in V} \frac{A_v}{S} + \sum_{w \in W} \frac{A_w}{S} \cdot (k_w - 1) + |W| < \mu. \quad (3)$$

We set $\alpha = \min_{v \in V} \{\frac{A_v}{S}\}$ and $\beta = \min_{w \in W} \{\frac{A_w}{S}\}$. A_v and S can be any positive integer, and $A_v \leq S$. Thus, the lower bound of α approaches to 0. Therefore, $\alpha \in [\varepsilon, 1]$, where $\varepsilon \rightarrow 0$, and $\beta \in [\frac{1}{2} + \varepsilon, 1]$ because if $\beta \leq \frac{1}{2}$, then $\exists B_w \leq \frac{S}{2}$, and an A_w packet and a B_w packet can be aggregated together. Thus, from Equation (3), we obtain

$$\begin{aligned} \alpha \cdot |V| + \beta \cdot \sum_{w \in W} (k_w - 1) + |W| &< \mu \\ \Rightarrow \alpha \cdot |V| + \beta \cdot \sum_{w \in W} (k_w + 1) - 2\beta \cdot |W| + |W| &< \mu \\ \Rightarrow \mathcal{R} < \frac{1}{\mu\beta}(\mu - |W| - \alpha|V|) + \frac{|V|}{\mu} + \frac{2|W|}{\mu}. \end{aligned}$$

Therefore, if the upper bound of \mathcal{R} appears, then α should be given the minimum value ε , and β also is the minimum value $\frac{1}{2} + \varepsilon$ since $\mu \geq |W| + |V|$.

Based on the derivation, we know that when \mathcal{R} is the upper bound, the result of the FF algorithm is as follows. In each superframe in V , the size of the aggregated packet is ε . In each superframe in W , there is one aggregated packet with the size $\frac{1}{2} \cdot S$ and k_w aggregated packets with the size $(\frac{1}{2} + \varepsilon) \cdot S$. In the k_w aggregated packets, only the first packet includes multiple original packets, while others include only one original packet. If others include multiple original packets, there must exist an original packet whose size is less than $\frac{1}{2} - \varepsilon$; the original packet can then be aggregated into the previous aggregated packets.

When \mathcal{R} is the upper bound, the optimal solution is as follows. Each superframe has at least one aggregated packet that cannot be moved to other periods. Thus, the optimal solution is to move $\sum_{w \in W} k_w$ packets into the $|V|$ aggregated packets in V such that more aggregated packets are eliminated. In the ideal case, the $|W|$ aggregated packets, which include multiple original packets, can be separated into the $|V|$ aggregated packets. If $|V|$ is big enough, the impact of $|W|$ aggregated packets on the payload of $|V|$ aggregated packets is also ε . Therefore, the $|W|$ aggregated packets can be ignored. Then, the $|W|$ aggregated packets with the size $\frac{1}{2} \cdot S$ can be aggregated into $\frac{|W|}{2}$ packets. Thus, the remaining $(\sum_{w \in W} k_w - 2 \cdot |W|)$ packets are discussed as follows.

—If $\sum_{w \in W} k_w - 2 \cdot |W| \geq |V|$, then $|V|$ aggregated packets in W are eliminated. Thus, the number of aggregated packets in the optimal solution is $\sum_{w \in W} k_w - 2 \cdot |W| - |V| + \frac{|W|}{2} + |V| + |W| = \sum_{w \in W} k_w - \frac{|W|}{2}$. So,

$$\mathcal{R} = \frac{\sum_{w \in W} k_w + |W| + |V|}{\sum_{w \in W} k_w - \frac{|W|}{2}} \leq \frac{2 \cdot \sum_{w \in W} k_w - |W|}{\sum_{w \in W} k_w - \frac{|W|}{2}} = 2.$$

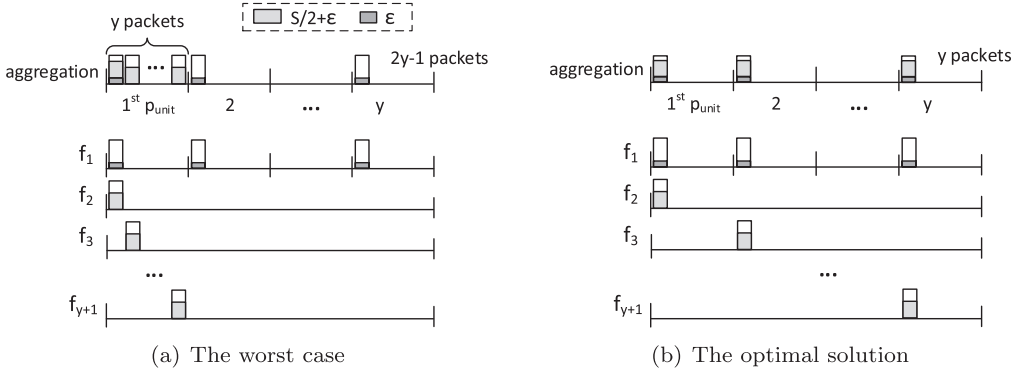


Fig. 3. Illustration of Theorem 4.1.

—If $\sum_{w \in W} k_w - 2 \cdot |W| < |V|$, all $(\sum_{w \in W} k_w - 2 \cdot |W|)$ packets in W are eliminated. Thus,

$$\mathcal{R} = \frac{\sum_{w \in W} k_w + |W| + |V|}{\frac{3}{2} \cdot |W| + |V|} < \frac{(2 \cdot |W| + |V|) + |W| + |V|}{\frac{3}{2} \cdot |W| + |V|} = 2.$$

Therefore, we determine that the approximation ratio is 2.

Then, we give a special example to show that the approximation ratio can approach to 2. The example considers only the intra-cluster period; other periods are neglected. There are $y + 1$ flows in a cluster. f_1 is characterized by $\langle p_{unit}, \epsilon \rangle$, and others have the same tuple $\langle y \cdot p_{unit}, \frac{S}{2} + \epsilon \rangle$. The number of time slots in an intra-cluster period is $y + 1$. For the FF algorithm, the worst case is that in which the first packet of f_1 and all packets of the other flows are delivered to the cluster head and aggregated in the first superframe (as shown in Figure 3(a)). Any two packets with the size $\frac{S}{2} + \epsilon$ cannot be aggregated together due to the size constraint S . In this case, there are y aggregated packets in the first superframe, and $y - 1$ aggregated packets in the following superframes. Therefore, the total number of aggregated packets is $2y - 1$. For the optimal solution, in each superframe, a packet with the size $\frac{S}{2} + \epsilon$ and a packet with the size ϵ are delivered and aggregated (as shown in Figure 3(b)). In this case, the total number of aggregated packets is y . Thus, the approximation ratio $\frac{2y-1}{y}$ approaches to 2 as $y \rightarrow +\infty$. Therefore, 2 is a tight approximation ratio. \square

The approximation ratio of our problem is larger than that of the traditional problem because the complete online method and the temporal feature restrict the choice of solutions. We propose two strategies to extend the classical FF algorithm such that its flexibility can be improved. Our proposed RM-FF algorithm is shown in Algorithm 1. The symbol c_a denotes how many time slots of the intra-cluster period have been used in the a th superframe (line 1).

Our first strategy is to specify the process order according to the temporal features. The original packet that has a shorter lifetime should be submitted and processed first because fewer time slots are available for it. This can prevent the situation where an original packet with a long lifetime, which has many available time slots, occupies the sole available time slot of an original packet with a short lifetime. The process order is similar to that of the RM (Rate-Monotonic) scheduling algorithm [21], which schedules the task with the shorter period first. We assign RM priorities to the flows, and the flow with the highest priority is submitted and processed first (lines 2 to 6). For each flow, we consider only how to schedule its first packet because use of the RM priority method in harmonic systems causes subsequent schedules to be cyclically repeated (line 16). First, we find

the earliest unused time slot that the current original packet can be submitted to (lines 7 to 9). Then, the original packet is aggregated using the FF algorithm in its available time interval (line 11). Finally, the number of unused time slots is updated (line 17).

The aggregation problem is not completely online, as we know the lifetimes and sizes of all packets, and we can further adjust the order of submissions. Therefore, the second strategy is to use some offline strategies to optimize the solution. The most famous offline FF algorithm is the first-fit decreasing algorithm, which first sorts objects in decreasing order based on their sizes and then uses the FF algorithm to pack them. Therefore, in our problem, when two flows have the same RM priority, the flow that has the bigger size is assigned the higher priority (lines 2 to 4). When a new aggregated packet δ_z is generated, its release time r_z and deadline d_z are set as the start time of the current inter-cluster period and the deadline of the original packet, respectively (lines 12 to 13). The symbol Λ in line 13 denotes the sum of the given lengths of the beacon, CAP, and CFP. If an aggregated packet has the same deadline as that of the new aggregated packet δ_z , and the sum of their sizes is less than the upper bound of the payload size (line 14), then they are aggregated together (line 15). At most one aggregated packet δ_b satisfies the condition of line 14. If there are two packets, they should have been aggregated together in previous iterations. This step is carried out to avoid the situation where because of the restriction of time slots, an original packet cannot be aggregated into an existing aggregated packet that is generated before its submission time, even though they have the same real-time requirement.

In this algorithm, an original packet can be aggregated into an aggregated packet that has a deadline that is earlier than its own. It is impossible for an original packet to be aggregated into an aggregated packet with a deadline that is later than its own because our harmonic system adopts the RM priority, i.e., the lifetimes of previously processed packets do not exceed the deadline of the current original packet. The number of iterations of the **for** loop in line 6 is $O(|F|)$, and the time complexity of line 11 is $O(|F|\frac{H}{q})$. Thus, the time complexity of Algorithm 1 is $O(|F|^2\frac{H}{q})$.

5 INTER-CLUSTER SCHEDULING METHOD

Based on the results of our proposed aggregation method, in this section, we present our inter-cluster scheduling method.

5.1 Mesh Network Model

In our model, superscripts are used to distinguish between different clusters, and subscripts indicate different members in a cluster. A WIA-PA network \mathcal{N} contains multiple clusters $\mathcal{N} = \{N^1, N^2, \dots\}$. The upper mesh network is constructed by cluster heads $\{n^1, n^2, \dots\}$. If clusters N^i and N^j are able to interfere with each other when they transmit packets simultaneously on the same channel, then $\eta^{ij} = 1$; otherwise, $\eta^{ij} = 0$. In addition, if nodes n^i and n^j are able to communicate reliably with each other, then $e^{ij} = 1$; otherwise, $e^{ij} = 0$. There are M available channels ($1 \leq M \leq 15$). If two one-hop transmissions that belong to different packets do not use the same node, they can be transmitted in parallel on different channels. A packet δ_z is routed to the data center along path $\pi_z = \{e_{ab}, e_{bc}, \dots\}$. All routing paths are fixed before schedule generation, and the routing techniques used in wireless sensor networks (e.g., References [24] and [2]) can be used to obtain paths.

5.2 Our Scheduling Method

Our inter-cluster scheduling method has two parts. First, because they have the same starting time and harmonic periods, the intra-cluster periods of any two nodes must overlap each other. Thus, if $\eta^{ij} = 1$, clusters N^i and N^j must be isolated by different channels. Second, aggregated packets

ALGORITHM 1: RM-FF

Input: F, C
Output: all aggregated packets

```

1  $\forall a \in [1, \frac{H}{q}], c_a = 0;$ 
2 for  $\forall i, j \in [1, |F|]$  do
3   if  $(p_i > p_j)$  or  $(p_i = p_j \text{ and } s_i \leq s_j)$  then
4      $f_j$  is assigned higher priority;
5 sort flows according to priorities, where  $f_1$  has the highest priority;
6 for  $i = 1$  to  $|F|$  do
7   find  $g$  in the range  $[1, \frac{p_i}{q}]$  such that  $\forall a \in (0, g - 1], c_a = C$  and  $c_g < C;$ 
8   if  $\nexists g$  then
9     return unschedulable;
10  else
11    invoke FF to process  $\tau_{i1}$  in the time interval  $[(g - 1) \cdot q + 1, p_i];$ 
12    if  $\tau_{i1}$  opens a new bin  $\delta_z$  then
13       $r_z = (g - 1) \cdot q + \Lambda + C; d_z = p_i;$ 
14      if  $\exists \delta_b, d_b = d_z, \hat{s}_z + \hat{s}_b \leq S$  then
15        move the packets in  $\delta_b$  to  $\delta_z$ , and delete  $\delta_b;$ 
16      repeat the above process for the intervals  $[g \cdot q + h \cdot p_i, p_i + h \cdot p_i], \forall h \in [1, \frac{H}{p_i});$ 
17       $\forall h \in [0, \frac{H}{p_i}), c_{g+h \cdot (\frac{p_i}{q})} ++;$ 
18 return all aggregated packets;

```

are scheduled using the ragged inter-cluster time slots. To improve the real-time performance, a novel method is needed to address the ragged time slots. We introduce the two parts in detail in the following.

5.2.1 Channel Assignment Among Clusters. In a cluster, packet submissions use at most one channel. Since all these submissions communicate with the cluster head, even if there are multiple channels, the sole cluster head cannot service multiple communications. To isolate the overlapping clusters, they should be assigned to different channels. This problem is the traditional vertex coloring problem. All clusters form the vertex set. $\eta^{ij} = 1$ means that there exists an edge between vertices N^i and N^j . The channels correspond to colors. Then, the problem is to find a color (channel) assignment such that any two vertices (clusters) sharing the same edge have different colors. Some classical algorithms, such as RLF [19] and DSATUR [4], can be used to solve this problem.

5.2.2 Inter-Cluster Scheduling Method. The release times and deadlines of all aggregated packets are assigned by our aggregation method. The inter-cluster scheduling problem is similar to the work in Reference [26] which is as follows: given the release times and deadlines of all packets, schedule packets under real-time constraints in a pure TDMA mesh network. However, our upper mesh network is not a pure TDMA network. In our mesh network, only when receiving and sending nodes are in their inter-cluster periods do the time slots between them become available. However, we can transform our problem to make it solvable via the CLLF method proposed in Reference [26]. The four parts (beacon, CAP, CFP, and intra-cluster period) result in the unavailable time slots. For each node n^i , these parts can be regarded as a virtual packet whose release time corresponds to the beginning of the beacon interval, whose deadline is the last time slot of the intra-cluster period, whose path includes only the node n^i (denoted as $\pi_i = \{e_{ii}, e_{ii}, \dots\}$), and whose path length is the sum of these parts. Virtual packets do not use channel resources. The main idea of the work in Reference [26] is to first schedule the packet that uses the node with

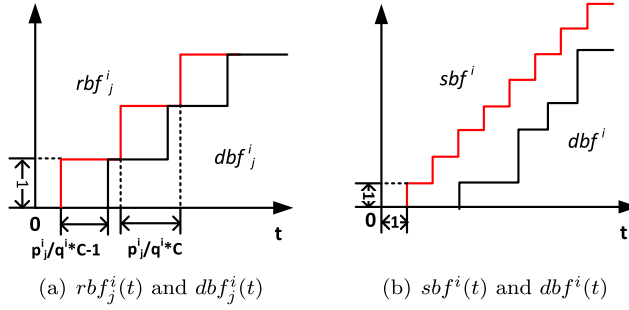


Fig. 4. Bound functions.

the least amount of slack time. Virtual packets have no slack time. Once they are released, they are scheduled on the corresponding nodes until their deadlines are reached. Thus, the time slots in the lifetime of virtual packets are unavailable. This problem is the same as our original problem. Therefore, the difference between our problem and Reference [26] can be transformed into a uniform packet model, and CLLF can be used to solve our problem. The transformed scheduling problem can also be solved using the classical real-time scheduling algorithms, such as EDF, LLF, and PD [21]. In our evaluation, we will compare these algorithms.

6 ANALYSIS OF SCHEDULABILITY

In this section, we analyze the schedulability of the upper- and lower-level networks.

6.1 Lower-Level Network

Our analysis applies only to the work-conserving submission method, which is explained in Definition 6.1.

Definition 6.1. Work-conserving submission method: A submission method that is never idle if there are packets waiting to be submitted is called a work-conserving submission method.

Our submission method is work-conserving, as we find the earliest time slot to submit packets (line 5 in Algorithm 1). In the analysis of the lower-level network, we consider only the intra-cluster period. Other parts can be eliminated from the time dimension, as they do not affect the schedulability of the lower-level network. The analysis of the lower-level schedulability is shown in Theorem 6.2. First, we introduce some functions related to network capacity and transmission requirements. The submission capacity of intra-cluster periods is characterized by a supply bound function $sbf^i(t)$, which denotes the cumulative submission capacity supplied by the cluster N^i within the time interval t (as shown in Figure 4(b)). Thus, $sbf^i(t) = t$.

The submission requirement of a flow f_j^i in the cluster N^i is characterized by a request bound function $rbf_j^i(t)$, which denotes the maximum cumulative submission requirement within the time interval t (as shown in Figure 4(a)). Then, we know that $rbf_j^i(t) = \lceil \frac{t}{\frac{p_j^i}{q^i} \cdot C} \rceil$, where $\frac{p_j^i}{q^i} \cdot C$ changes

the flow period to a form that includes only intra-cluster periods. Submission requirements must be satisfied before their deadlines. A demand bound function $dbf_j^i(t)$ is used to characterize the minimal cumulative submission capacity demanded by the flow f_j^i such that its deadline is not missed, which is obtained by horizontally shifting rbf_j^i to the right for $\frac{p_j^i}{q^i} \cdot C - 1$ (as shown in

Figure 4(a)). Thus,

$$dbf_j^i(t) = \begin{cases} 0 & \text{if } t < \frac{p_j^i}{q^i} \cdot C \\ rbf_j^i\left(t - \frac{p_j^i}{q^i} \cdot C + 1\right) & \text{otherwise} \end{cases}.$$

The total demand bound function of cluster N^i is $dbf^i(t) = \sum_{\forall f_j^i \in F^i} dbf_j^i(t)$.

THEOREM 6.2. *When a work-conserving submission method is used, all packets of cluster N^i can be transmitted under real-time constraints if and only if $\forall t \in [1, H], sbf^i(t) \geq dbf^i(t)$.*

PROOF. First, for a work-conserving submission method, if $\forall t \in [1, H], sbf^i(t) \geq dbf^i(t)$, then all demands can be satisfied. Thus, all packets can be submitted before their deadlines. Second, if all packets can be submitted in time, then the supply submission capacity is at least larger than or equal to the minimal submission demand, i.e., $\forall t \in [1, H], sbf^i(t) \geq dbf^i(t)$. Thus, the theorem holds. \square

6.2 Upper-Level Network

The inter-cluster period of node n^i is $(q^i - \Lambda - C)$. Similarly, we use a function $\overline{sbf}^i(t)$ to characterize the maximal cumulative transmission capacity of node n^i within the time interval t . Thus,

$$\overline{sbf}^i(t) = (q^i - \Lambda - C) \cdot \left\lfloor \frac{t}{q^i} \right\rfloor + \min\{t \bmod q^i, q^i - \Lambda - C\}.$$

A demand bound function $\overline{dbf}_z^i(t)$ indicates that an aggregated packet δ_z demands the minimal cumulative transmission capacity supplied by node n^i in the time interval t . This is expressed as

$$\overline{dbf}_z^i(t) = \max\{\min\{t - (d_z - \gamma_z^i) + 1, x_z^i\}, 0\},$$

where the first one-hop transmission that uses the node n^i is γ_z^i hops away from the destination, i.e., before the time $(d_z - \gamma_z^i)$, δ_z must be supplied at least one unit of transmission capacity. The symbol x_z^i represents the number of times that δ_z uses n^i . Thus, x_z^i and 0 are the upper and lower bounds of the capacity demands. If δ_z passes through n^i , then $x_z^i = 2$; if n^i is the source or destination of δ_z , then $x_z^i = 1$; and if δ_z does not use n^i , then $x_z^i = 0$. The total demand bound function of the node n^i is

$$\overline{dbf}^i(t) = \sum_{\forall \delta_z} \overline{dbf}_z^i(t).$$

Thus, the analysis of the upper-level schedulability is as shown in Theorem 6.3.

THEOREM 6.3. *If $\exists n^i, \exists t \in [1, H], \overline{sbf}^i(t) < \overline{dbf}^i(t)$, then the network is not schedulable.*

PROOF. $\overline{sbf}^i(t)$ is the maximal transmission capacity. If the maximal capacity cannot satisfy the minimal demand $\overline{dbf}^i(t)$, then there is no feasible solution no matter which method is used. Therefore, the network must not be schedulable. \square

7 EVALUATION

In this section, we conduct simulations to illustrate the performance of our proposed methods. First, we evaluate the aggregation methods, based on which we then compare the performance of the different scheduling algorithms in WIA-PA networks. All simulations are written in C and run on a Windows machine with 3.4GHz CPU and 16GB memory.

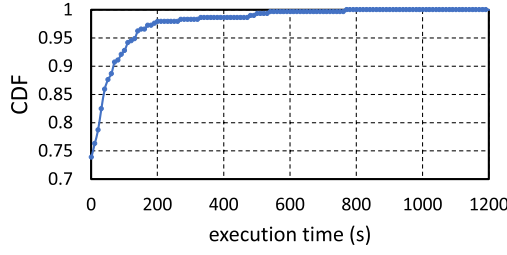


Fig. 5. Execution time of CPLEX.

7.1 Aggregation Method

Our test cases are generated randomly based on the following parameters: the number of flows $|F|$, the length of the intra-cluster period C , the range of the flow periods $[p^l, p^u]$, and the maximum size of an original packet s^u . The number of cluster members is the same as the number of flows. Even if the number of cluster members is greater than the number of flows, the members that do not generate flows have no impact on the algorithm performance and can be omitted. Aggregation methods run inside clusters. Therefore, in this subsection, all test cases adopt star topology, and there is no routing algorithm. We compare our RM-FF aggregation method to two methods: (1) the classical FF method and (2) CPLEX, in which the single-threaded CPLEX solver [13] is exploited to obtain the optimized solution from the formulation (see Section 4.2). However, due to the unacceptable execution time of CPLEX, we set a time limit allowed for a call to the optimizer. If CPLEX cannot find the optimal solution under this time limit, then the most feasible solution found is used in the following figures. To obtain an appropriate time limit, we run the following simulations. We randomly generate test cases based on the parameters: $|F| \in [10, 35]$, $C \in \{\frac{|F|}{2}, \frac{2|F|}{3}\}$, $p^l, p^u \in [2^6, 2^{12}]$ and $s^u \in \{16, 32, 64, 128\}$. These parameters cover all of the following simulations. We use CPLEX to solve them and record execution times. Figure 5 shows the cumulative distribution function (CDF) of these execution times. The time limit of these test cases is 1 hour. We find that 99% of solvable test cases can find optimal solutions within 600 seconds. Therefore, in the following simulations, we set the time limit as 600 seconds.

Figure 6 shows the number of aggregated packets under different packet sizes. Flow periods and packet sizes are randomly selected in the range from 2^6 to 2^{10} and the range from 1 to the coordinating x-coordinate, respectively. The payload size of an aggregated packet does not exceed 128, which is defined in the standard WIA-PA. Each point in the figures is the average of 200 test cases. All these test cases can be solved within 1 ms using FF and RM-FF, and all results are normalized using CPLEX. Our RM-FF method is better than FF and yields solutions close to the optimized solutions. As s^u increases, our method slightly deviates from CPLEX. This deviation occurs because the aggregating of big packets is less feasible. When big packets and small packets are accurately matched, the results are optimal. However, a match is hard to find. Our method only introduces a slight fluctuation. FF first increases and then decreases. In Figure 6(a) and (b), when $s^u = 16$, FF is nearly equal to CPLEX because in each superframe, almost all original packets can be aggregated into one aggregated packet. Meanwhile, in Figure 6(c) and (d), as $|F|$ and C increase, the number of aggregated packets increases, and the solution space becomes more complex. Thus, the naive FF method cannot be used to find optimized solutions. In Figure 6(a), when $s^u = 32$, FF has 35% more packets than RM-FF. Then, as s^u increases, the diversity of the aggregating packets decreases. Thus, all methods are similar to each other.

Figure 7 focuses on the impact of the number of flows on aggregating packets. The packet size is randomly selected in the range $[1, 64]$, and $p^u = 2^{10}$. Other parameters are as shown in the figures.

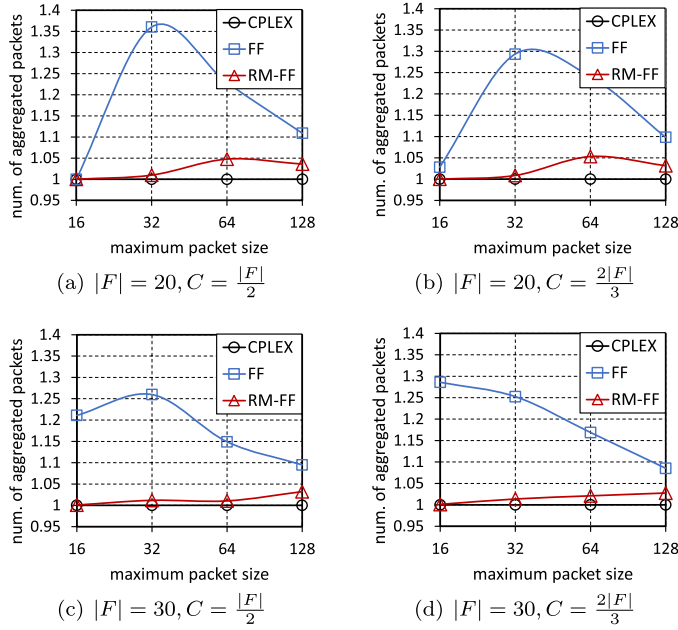


Fig. 6. Comparison of aggregated packets for different packet sizes.

Our RM-FF method first increases and then decreases as $|F|$ increases. This occurs because when $|F| = 10$, the solution space is small, and it is easy to find the optimal solution. Then, extension of the space leads to a reduction in performance. Compared with CPLEX, our method introduces only a small number of extra aggregated packets, i.e., up to 5%. When $|F| > 20$, the execution time required by CPLEX to find the optimal solution is unacceptable. Within the time limit, CPLEX finds some feasible but not optimal solutions. When $|F| = 35$, these feasible solutions are less effective than those found by our method within 1 ms. The flow period range of Figure 7(c) and (d) is larger than that of Figure 7(a) and (b). As a result, CPLEX can find more feasible but not optimal solutions. Therefore, the effectiveness of our method is slightly improved.

7.2 Scheduling Algorithm

In the upper level, mesh networks are randomly generated based on the equation $Area \times \xi = \frac{|N|d^2\sqrt{27}}{2\pi}$ [5], where $Area$ is the square area, ξ denotes the network density and d denotes the transmission range, which is 40 meters. The data center is at the center of the $Area$, and cluster heads are randomly deployed around it in the $Area$. Routing paths are obtained using the Dijkstra algorithm. The intra-cluster parameters are randomly selected in the ranges $F \in [10, 35]$, $p^l \in [2^6, 2^9]$, $p^u \in [2^{10}, 2^{12}]$, $C \in \{\frac{|F|}{2}, \frac{2|F|}{3}\}$ and $s^u \in [1, 64]$. The three aggregation methods are used for the lower-level networks, and five real-time scheduling methods are used for the upper-level network [21]: (1) EDF, in which the aggregated packet with the smallest relative deadline is scheduled first; (2) least laxity first (LLF), in which the aggregated packet with the least laxity is scheduled first; (3) proportional deadline monotonic (PD), in which the aggregated packet with the smallest relative subdeadline, which is defined as the relative deadline divided by the number of hops, is scheduled first; (4) earliest proportional deadline first (EPD), in which the aggregated packet with the smallest absolute subdeadline, which is defined as the remaining time divided by the number of remaining hops, is scheduled first; and (5) CLLF [26], in which the one-hop transmission

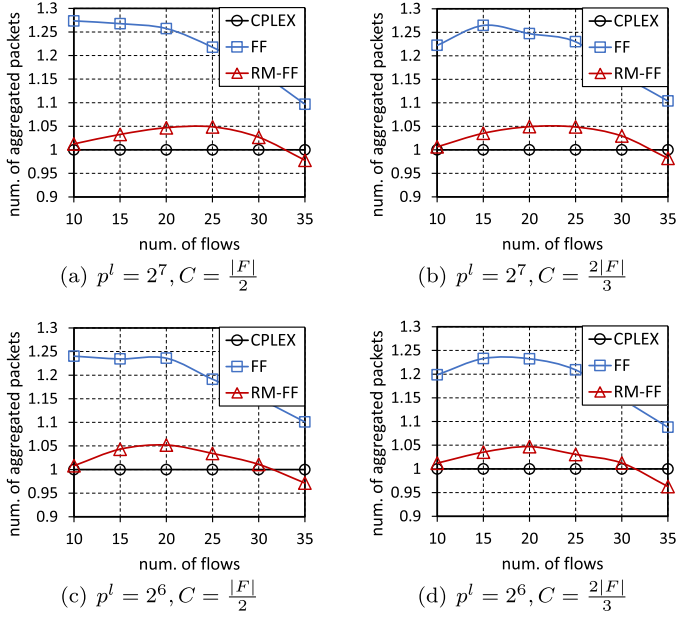


Fig. 7. Comparison of aggregated packets for different numbers of flows.

with more conflict is scheduled first. The classical RM and deadline-monotonic (DM) real-time scheduling methods are not considered because in the upper network, aggregated packets are not periodic.

Figure 8 shows a comparison of schedulable ratios, which are defined as the percentage of test cases for which an algorithm is able to find a feasible schedule. For each point, 2,000 test cases are randomly generated. Comparing the three subfigures, schedulable ratios increase as the number of channels M and the network density ξ increase because more channels can transmit more packets in parallel, and a denser network has shorter path lengths. Thus, when M and ξ increase, networks are more easily scheduled.

Although some related works have compared these real-time scheduling algorithms, their results are different from ours. For example, in Reference [26], CLLF has the best performance, and LLF and EDF are the worst. However, in WIA-PA networks, the results are opposite. There are three reasons for this difference: (1) the schedules of virtual flows are fixed and cannot be changed, (2) a source sends several aggregated packets with the same deadline along the same path at the same time, and (3) WIA-PA networks only focus on uplink transmissions. The CLLF method assesses only the conflict because of which the current one-hop transmission suffers. Whether the subsequent hops encounter more conflict is not considered. Virtual flows must always be scheduled once they are released. The conflict introduced by virtual flows is very serious. On the other hand, WIA-PA networks schedule only uplink transmissions and multiple packets are transmitted at the same time along the same path. These processes cause more conflict near the data center. Therefore, in WIA-PA networks, when flows are assigned priorities in terms of conflict, the method that does not consider the subsequent conflict inevitably induces large errors.

The LLF, PD, and EPD methods consider the impact of routing paths on priority assignment. They all schedule the furthest packets first, i.e., they block more packets near the center. This exacerbates the transmission conflict. In the LLF method, schedules are determined in terms of the remaining time minus the remaining hops, while in PD and EPD, they are determined in terms of

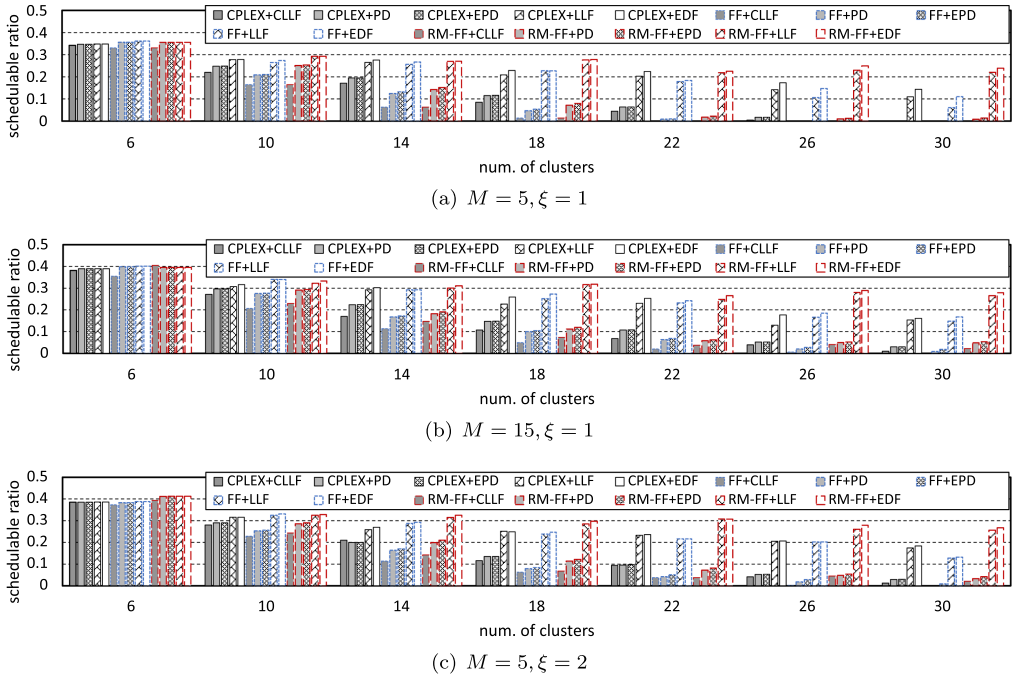


Fig. 8. Comparison of schedulable ratios.

the deadline divided by the number of hops. Thus, paths have a greater effect on PD and EPD. They quickly schedule packets to the region near the center, and then more conflict occurs. Thus, LLF has higher schedulable ratios than PD and EPD. Comparing to the three methods, the method EDF does not consider routing paths and decreases the occurrence of conflicts near the center. Therefore, EDF has the highest schedulable ratio and the most stable performance. It is the preferred method for WIA-PA networks.

From Figure 8, we find that our RM-FF aggregation method makes the process of scheduling algorithms more effective even though CPLEX has fewer aggregated packets than RM-FF. The objective of CPLEX is to minimize the number of aggregated packets. If we do not consider the unacceptable execution time, the solutions obtained by CPLEX are optimal. However, the number of aggregated packets and the real-time requirements of aggregated packets both have impacts on the process of scheduling algorithms. CPLEX only optimizes one factor, while RM-FF considers these two factors together. RM-FF aggregates short-deadline packets first. In other words, it eliminates more high real-time packets than CPLEX. Therefore, the packets aggregated by RM-FF have looser real-time constraints and can be scheduled easily. In Figure 8(b), when the number of clusters is 30 and 26, RM-FF+EDF has 10% higher real-time performance than CPLEX+EDF and FF+EDF.

8 CONCLUSIONS

This article focuses on the aggregation scheduling problem of WIA-PA networks. First, we design an aggregated method for a lower-level network to aggregate original packets into the minimum number of aggregated packets such that in the upper-level network these aggregated packets can be scheduled in real time with a limited amount of resources. Second, in the upper-level network, we transform the WIA-PA model into the universal model and then use existing algorithms to

schedule packets under real-time constraints. The simulation results indicate that our methods can effectively reduce the number of aggregated packets, and that the real-time EDF scheduling algorithm is the most suitable for WIA-PA networks. In the future, we will study the impact of the length of an intra-cluster period on network performance.

REFERENCES

- [1] T. Abirami and S. Anandamurugan. 2016. Data aggregation in wireless sensor network using shuffled frog algorithm. *Wirel. Pers. Commun.* 90, 2 (2016), 537–549.
- [2] Jamal N. Al-Karaki and Ahmed E. Kamal. 2004. Routing techniques in wireless sensor networks: A survey. *IEEE Wirel. Commun.* 11, 6 (2004), 6–28.
- [3] M. Aslam, Nadeem Javaid, A. Rahim, U. Nazir, Ayesha Bibi, and Z. A. Khan. 2012. Survey of extended LEACH-based clustering routing protocols for wireless sensor networks. In *High. Perform. Comput. Commun. & Int. Confer. Embed. Softw. Syst.* IEEE, 1232–1238.
- [4] Daniel Brélaz. 1979. New methods to color the vertices of a graph. *Commun. ACM* 22, 4 (1979), 251–256.
- [5] Tiago Camilo, Jorge Sá Silva, André Rodrigues, and Fernando Boavida. 2007. Gensen: A topology generator for real wireless sensor networks deployment. In *SEUC*. Springer, 436–445.
- [6] Octav Chipara, Chenyang Lu, and John Stankovic. 2006. Dynamic conflict-free query scheduling for wireless sensor networks. In *ICNP*. IEEE, 321–331.
- [7] Hongsik Choi, Ju Wang, and Esther A. Hughes. 2009. Scheduling for information gathering on sensor network. *Wirel. Netw.* 15, 1 (2009), 127–140.
- [8] Behnam Dezfouli, Marjan Radi, and Octav Chipara. 2016. Mobility-aware real-time scheduling for low-power wireless networks. In *INFOCOM*. IEEE, 1–9.
- [9] György Dósa and Jiri Sgall. 2013. First fit bin packing: A tight analysis. In *LIPICs-Leibniz International Proceedings in Informatics*, Vol. 20. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik.
- [10] Qiang Gao, Yi Zuo, Jun Zhang, and Xiao-Hong Peng. 2010. Improving energy efficiency in a wireless sensor network by combining cooperative MIMO with data aggregation. *IEEE Trans. Veh. Technol.* 59, 8 (2010), 3956–3965.
- [11] Longjiang Guo, Yingshu Li, and Zhipeng Cai. 2016. Minimum-latency aggregation scheduling in wireless sensor network. *J. Comb. Optim.* 31, 1 (2016), 279–310.
- [12] Wendi Rabiner Heinzelman, Anantha Chandrakasan, and Hari Balakrishnan. 2000. Energy-efficient communication protocol for wireless microsensor networks. In *HICSS*. IEEE, 1–10.
- [13] Kenneth Holmström, Anders O. Göran, and Marcus M. Edvall. 2009. User’s guide for Tomlab/CPLEX v12. 1. *Tomlab Optimization*. Retrieved July 1, 2017 from https://tomopt.com/docs/TOMLAB_CPLEX.pdf.
- [14] IEC62591. 2016. IEC 62591: Industrial networks–Wireless communication network and communication profiles–WirelessHART. Retrieved December 1, 2016 from <https://webstore.iec.ch/publication/24433>.
- [15] IEC62601. 2015. IEC 62601: Industrial networks – Wireless communication network and communication profiles – WIA-PA. Retrieved December 1, 2015 from <https://webstore.iec.ch/publication/23902>.
- [16] IEC62734. 2014. IEC 62734: Industrial networks – Wireless communication network and communication profiles – ISA 100.11a. Retrieved November 1, 2014 from <https://webstore.iec.ch/publication/7409>.
- [17] Xi Jin, Fanxin Kong, Linghe Kong, Wei Liu, and Peng Zeng. 2017. Reliability and temporality optimization for multiple coexisting WirelessHART networks in industrial environments. *IEEE Trans. Ind. Electron.* 64, 8 (2017), 6591–6602.
- [18] Xi Jin, Huiting Xu, Changqing Xia, Jintao Wang, and Peng Zeng. 2017. Convergecast scheduling and cost optimization for industrial wireless sensor networks with multiple radio interfaces. *Wirel. Netw.* (2017), 1–15.
- [19] Frank Thomson Leighton. 1979. A graph coloring algorithm for large scheduling problems. *J. Res. Nat. Bur. Stand.* 84, 6 (1979), 489–506.
- [20] Wei Liang, Xiaoling Zhang, Yang Xiao, Fuqiang Wang, Peng Zeng, and Haibin Yu. 2011. Survey and experiments of WIA-PA specification of industrial wireless network. *Wirel. Commun. Mob. Comput.* 11, 8 (2011), 1197–1212.
- [21] Jane W. S. Liu. 2000. *Real-Time Systems*. Prentice Hall.
- [22] Divya Lohani and Shirshu Varma. 2016. Energy efficient data aggregation in mobile agent based wireless sensor network. *Wirel. Pers. Commun.* 89, 4 (2016), 1165–1176.
- [23] Chenyang Lu, Abusayeed Saifullah, Bo Li, Mo Sha, Humberto Gonzalez, Dolvara Gunatilaka, Chengjie Wu, Lanshun Nie, and Yixin Chen. 2016. Real-time wireless sensor-actuator networks for industrial cyber-physical systems. *Proc. IEEE* 104, 5 (2016), 1013–1024.
- [24] Nikolaos A. Pantazis, Stefanos A. Nikolidakis, and Dimitrios D. Vergados. 2013. Energy-efficient routing protocols in wireless sensor networks: A survey. *IEEE Commun. Surv. Tutor.* 15, 2 (2013), 551–591.
- [25] Rashmi Ranjan Rout and Soumya K. Ghosh. 2014. Adaptive data aggregation and energy efficiency using network coding in a clustered wireless sensor network: An analytical approach. *Comput. Commun.* 40 (2014), 65–75.

- [26] Abusayeed Saifullah, You Xu, Chenyang Lu, and Yixin Chen. 2010. Real-time scheduling for wirelessHART networks. In *RTSS*. IEEE, 150–159.
- [27] Jungmin So and Heejung Byun. 2014. Opportunistic routing with in-network aggregation for asynchronous duty-cycled wireless sensor networks. *Wirel. Netw.* 20, 5 (2014), 833–846.
- [28] Pablo Soldati, Haibo Zhang, and Mikael Johansson. 2009. Deadline-constrained transmission scheduling and data evacuation in WirelessHART networks. In *ECC*. IEEE, 4320–4325.
- [29] Haibo Zhang, Fredrik Österlind, Pablo Soldati, Thiemo Voigt, and Mikael Johansson. 2015. Time-optimal convergecast with separated packet copying: Scheduling policies and performance. *IEEE Trans. Veh. Technol.* 64, 2 (2015), 793–803.
- [30] Haibo Zhang, Pablo Soldati, and Mikael Johansson. 2009. Optimal link scheduling and channel assignment for convergecast in linear WirelessHART networks. In *WiOPT*. IEEE, 1–8.
- [31] Meng Zheng, Wei Liang, Haibin Yu, and Yang Xiao. 2017. Performance analysis of the industrial wireless networks standard: WIA-PA. *Mob. Netw. Appl.* 22, 1 (2017), 139–150.
- [32] ZKAW. 2017. A monitoring and control system in oil fields. Retrieved December 5, 2017 from http://www.zkaw.com.cn/columns_yyal/FrontColumns_navigation01-navFirstColumnId=87&&columnsId=106&FrontColumns_navigation01-yyalFirstColumnId=106&comp_stats=comp-FrontColumns_navigation01-yyal.html.
- [33] ZKAW. 2017. A temperature monitoring system in steel mills. Retrieved December 5, 2017 from <http://wia.sia.cn/03.asp?pd=cp&id=69&anclassid=15&nclassid=609>.

Received December 2017; revised June 2018; accepted August 2018