Analysis of EDF Scheduling for Wireless Sensor-Actuator Networks

Chengjie Wu, Mo Sha, Dolvara Gunatilaka, Abusayeed Saifullah, Chenyang Lu, Yixin Chen
Cyber-Physical Systems Laboratory
Department of Computer Science & Engineering, Washington University in St. Louis

***Abstract*—Industry is adopting Wireless Sensor-Actuator Networks (WSANs) as the communication infrastructure for process control applications. To meet the stringent real-time performance requirements of control systems, there is a critical need for fast end-to-end delay analysis for real-time flows that can be used for online admission control. This paper presents a new end-to-end delay analysis for periodic flows whose transmissions are scheduled based on the Earliest Deadline First (EDF) policy. Our analysis comprises novel techniques to bound the communication delays caused by channel contention and transmission conflicts in a WSAN. Furthermore, we propose a technique to reduce the pessimism in admission control by iteratively tightening the delay bounds for flows with short deadlines. Experiments on a WSAN testbed and simulations demonstrate the effectiveness of our analysis for online admission control of real-time flows.**

I. Introduction

With the emergence of industrial standards such as WirelessHART [1] and ISA100 [2], process control industries are adopting *Wireless Sensor-Actuator Networks (WSANs)* in which sensors and actuators communicate through low-power multi-hop wireless mesh networks [3]. Since excessive communication delay may lead to severe degradation of control performance or even instability of the control system, it is critical to estimate worst-case end-to-end communication delays for real-time flows in WSANs [4]. Moreover, fast delay analysis is needed for online admission control and network reconfiguration in response to dynamic changes of channel conditions in industrial environments.

In this paper, we present a new delay analysis for periodic flows in WSANs in which transmissions are scheduled based on the *Earliest Deadline First (EDF)* policy, a common real-time scheduling policy that has been found to be an effective transmission scheduling policy for real-time WSANs in recent studies [5].

Our new delay analysis can be used to derive end-to-end delay bounds for real-time flows in WSANs. The key feature of our analysis lies in a novel approach to combine two types of delays in a WSAN: contention delays due to limited number of wireless channels, and conflict delays caused by conflicts among concurrent wireless transmissions involving a same device. Furthermore, we reduce the pessimism in admission control by iteratively tightening the delay bounds for flows with short deadlines.

We evaluate our delay analysis through experiments on a 63-node WSAN testbed and simulations. The experiment results demonstrate our delay analysis provides safe bounds of real end-to-end delays. The simulation results show that our delay analysis is effective in term of acceptance ratio when used for admission control. We also provide a comprehensive simulation study that compares a state-of-the-art fixed priority scheduling algorithm [6] and a dynamic priority scheduling algorithm [5]. Our simulations show EDF outperforms fixed priority scheduling [6] in term of real-time performance, while delivering competitive acceptance ratios to the existing dynamic priority scheduling policy at lower computational cost.

978-1-4799-0913-1/14/$31.00

II. WSAN Characteristics

We consider a network model that captures the key features of industrial WSAN standards (e.g., WirelessHART [1]). A WSAN consists of a set of field devices, a gateway and multiple access points. A field device could be a sensor, an actuator, or both. Each device (field device or access point) is equipped with a half-duplex omnidirectional radio transceiver compatible with the IEEE 802.15.4 standard.

The gateway communicates with field devices through access points. The access points and the field devices form a wireless mesh network. The WSAN has a centralized network management architecture that enhances the predictability and visibility of network operations at the cost of scalability. All devices are managed by a network manager, a software process running on the gateway. The network manager gets the real-time flow information from process control applications, collects the network topology information from the field devices, generates the routes, and performs admission control of the real-time flows. For the set of accepted flows, the network manager generates the transmission schedule for all field devices, and disseminates the superframe (including the routes and transmission schedule) to the field devices. When a field device detects a link failure, it notifies the network manager to update the routes and schedule.

The WSAN adopts a multi-channel Time Division Multiple Access (TDMA) protocol layer on top of the IEEE 802.15.4 physical layer. The clocks of all devices across the network are synchronized. Time is divided into 10 ms slots. Each time slot can accommodate one data packet transmission and its acknowledgment. To avoid potential collision between concurrent transmissions in the same channel, only one transmission is scheduled on each channel across the whole network. As a result, the total number of concurrent transmissions scheduled

in a time slot cannot exceed the number of channels. While this conservative design reduces network throughput and scalability, it avoids interference between transmissions within the network and therefore enhances reliability and predictability for industrial applications.

The WirelessHART standard supports two types of routing: source routing and graph routing. Source routing provides a single route for each flow, whereas graph routing provides multiple redundant routes in a routing graph and therefore enhances reliability through route diversity. Our analysis currently assumes source routing and can be easily extended to a model where each flow has multiple source routes and send redundant packets through every route to enhance reliability. Supporting graph routing is part of our future work.

III. EDF Scheduling in WSANs

We consider a set of periodic flows $F = \{F_1, F_2, \cdots, F_N\}$ to be scheduled on m channels. Each flow

$$F_k = (D_k, T_k, \alpha_k, \phi_k, C_k)$$

is characterized by a relative deadline D_k, a period T_k, a start time α_k, a route ϕ_k and an transmission count C_k. The route ϕ_k is composed of a sequence of links in the network from the source device s_k to the destination device a_k. To enhance reliability, at most κ (re)transmissions are scheduled for one link. Once the sender receives the acknowledgment, it will discard other retransmission retries. The transmission count C_k equals the total number of (re)transmissions scheduled for one packet of this flow along its route, i.e., $C_k = |\phi_k|\kappa$, where $|\phi_k|$ is the length of ϕ_k.

We follow the constrained deadline model where the deadline of each flow is within its period, i.e., $D_k \leq T_k$. Hence different packets of the same flow cannot co-exist in the network in the same time slot. For flow F_k, a new packet is released at source node s_k in the beginning of each period. We use $P_{k,j}$ to refer to the j^{th} packet of the flow F_k, whose release time is $r_{k,j} = \alpha_k + (j-1)\,T_k$. Packet $P_{k,j}$ needs to be delivered to the destination a_k through a sequence of transmissions along ϕ_k. If $P_{k,j}$ is delivered to the destination at slot $f_{k,j}$ through its route, its *end-to-end delay* $R_{k,j}$ is $f_{k,j} - r_{k,j} + 1$. A packet needs to complete all its transmissions before its absolute deadline $d_{k,j} = r_{k,j} + D_k$. We use R_k to denote the end-to-end delay of flow F_k, which is the maximum end-to-end delay of all its packets.

As we discussed in Section II, the network manager generates schedules for all field devices up to the hyper-period, i.e., the least common multiply of $\{T_k, k = 1, \ldots, n\}$. When generating schedules, the network manager follows the EDF scheduling policy. For all released packets, each packet is assigned a priority based on its absolute deadline. The packet with an earlier absolute deadline is assigned a higher priority. At any time slot, if there remains an available channel, among all released but not delivered packets which do not conflict with packets already scheduled in this time slot, the packet with highest priority is scheduled to this slot. This process repeats until all channels are occupied or all remaining packets conflict with at least one scheduled packet. Transmissions of the same packet can be scheduled on different channels at different time slots.

IV. Worst-Case End-to-End Delay Analysis

In this section, we present our worst-case end-to-end delay analysis for real-time flows under the EDF policy. A set of real-time flows is schedulable if every flow has a worst-case end-to-end delay that is no greater than its deadline. Given a set of real-time flows, our goal is to derive an upper bound on the worst-case end-to-end delay of every flow. The delay analysis can be used as a schedulability test of the flow set under EDF.

A. Terminology

Before analyzing the delays, we first introduce the terminology used in the analysis. We say a packet is *ready* if it is released and not delivered yet. We say a packet *executes* in a time slot if it has a transmission scheduled in this time slot. A packet can be delayed for two reasons.

- **Conflict delay:** Due to the half-duplex radio, two transmissions conflict with each other if they share a node (sender or receiver). Then only one of them can be scheduled at current time slot. Therefore, if a packet conflicts with another packet that has already been scheduled in the current time slot, it has to be postponed to a later time slot, resulting in conflict delay.
- **Contention delay:** As a WSAN does not allow concurrent transmissions in a same channel, each channel can only accommodate one transmission across the network in each time slot. If all channels are assigned to transmissions of other packets, a packet must be delayed to a later slot, resulting in contention delay.

To be more precise, we define the *conflict delay* of packet $P_{k,j}$ as the number of time slots when packet $P_{k,j}$ is delayed because it conflicts with higher priority packets. We denote conflict delay of packet $P_{k,j}$ as $Y^f_{k,j}$. We define the *contention delay* of packet $P_{k,j}$ as the number of time slots when $P_{k,j}$ is delayed because all the channels are occupied by higher priority packets and none of them conflict with $P_{k,j}$. We denote contention delay of packet $P_{k,j}$ as $Y^t_{k,j}$. Then the end-to-end delay of packet $P_{k,j}$ is

$$R_{k,j} = Y^f_{k,j} + Y^t_{k,j} + C_k, \tag{1}$$

where C_k is the transmission count of $P_{k,j}$ along its route.

We define the *interference* of a flow F_l on packet $P_{k,j}$ as the number of slots when $P_{k,j}$ waits for transmissions of packets belonging to F_l. We denote flow F_l's interference on packet $P_{k,j}$ as $I_{k,j}(l)$. Note the terminology *interference* refers to the time a packet is delayed by transmissions associated with another flow. It is not related to interference between concurrent wireless transmissions, which cannot occur in a WSAN because it does not allow concurrent transmissions in a same channel. We further categorize flow F_l's interference on packet $P_{k,j}$ into two: *conflict interference* $I^f_{k,j}(l)$ and

contention interference $I^t_{k,j}(l)$. Flow F_l's conflict interference on packet $P_{k,j}$ is the number of time slots when $P_{k,j}$ is delayed due to conflicting transmissions belonging to flow F_l. F_l's contention interference on $P_{k,j}$ is the number of time slots when $P_{k,j}$ waits while transmissions of flow F_l are executed and do not conflict with $P_{k,j}$. By definition, we have

$$I_{k,j}(l) = I^f_{k,j}(l) + I^t_{k,j}(l). \tag{2}$$

In the rest of this section, we present the worst-case delay analysis in following 4 steps.

1) We analyze the end-to-end delay bound of a packet given the interference of the other flows.
2) We derive an upper bound of a flow's conflict and contention interferences on a packet.
3) Combining 1) and 2), we give the upper bound of the end-to-end delay of a flow.
4) we reduce the pessimism in admission control by iteratively tightening the delay bounds of flows with short deadlines.

B. Conflict and Contention Delays

In this subsection we analyze the conflict delay and contention delay of a packet. Consider a packet $P_{k,j}$ of flow F_k released at time $r_{k,j}$ with absolute deadline $d_{k,j}$. We want to analyze the end-to-end delay of $P_{k,j}$ assuming both the conflict interferences and contention interferences of all the other flows on $P_{k,j}$ are given.

Lemma 1. *The conflict delay $Y^f_{k,j}$ of $P_{k,j}$ is upper bounded as follow:*

$$Y^f_{k,j} \leq \sum_{l \neq k} I^f_{k,j}(l). \tag{3}$$

Proof: For any time slot within $Y^f_{k,j}$, $P_{k,j}$ is delayed by conflict if and only if there is at least one scheduled higher priority packet conflicting with it. Let one of these higher priority packets belongs to flow F_l. Recall the definition of F_l's conflict interference on $P_{k,j}$ is the number of time slots when $P_{k,j}$ is delayed due to conflicting transmissions belonging to flow F_l. By definition, this time slot is a part of F_l's conflict interference $I^f_{k,j}(l)$.

Since our statement is not limited to a specific time slot within $Y^f_{k,j}$, we show any time slot in which $P_{k,j}$ suffers one conflict delay indeed belongs to at least one flow's conflict interference on $P_{k,j}$. Therefore, the total conflict delay of $P_{k,j}$ is bounded by the sum of conflict interferences of all other flows. ■

Lemma 2. *The contention delay $Y^t_{k,j}$ of $P_{k,j}$ is upper bounded as follow:*

$$Y^t_{k,j} \leq \lfloor \frac{\sum_{l \neq k} I^t_{k,j}(l)}{m} \rfloor. \tag{4}$$

Proof: We follow the same reasoning of the proof of the Lemma 1. For any time slot, $P_{k,j}$ is delayed by contention if and only if all channels are occupied by higher priority packets and none of them conflict with $P_{k,j}$. Then in this time slot, there must be m higher priority packets scheduled and none of them conflict with $P_{k,j}$.

Let $I^{t'}_{k,j}(l)$ denote the number of time slots when 1) $P_{k,j}$ is ready but not executing, 2) F_l is executing and 3) none of the executing packets conflict with $P_{k,j}$. Then the contention delay of $P_{k,j}$ is upper bounded by $\lfloor \frac{\sum_{l \neq k} I^{t'}_{k,j}(l)}{m} \rfloor$. Recall that $I^t_{k,j}(l)$ is the number of time slots when 1) $P_{k,j}$ is ready but not executing, 2)F_l is executing and 3) F_l does not conflict with $P_{k,j}$. Comparing the set of time slots within $I^t_{k,j}(l)$ and $I^{t'}_{k,j}(l)$, we see the latter one is a subset of the former one, so $I^{t'}_{k,j}(l)$ is no greater than $I^t_{k,j}(l)$. Therefore, $Y^t_{k,j}$ is upper bounded by $\lfloor \frac{\sum_{l \neq k} I^t_{k,j}(l)}{m} \rfloor$. ■

To meet $P_{k,j}$'s deadline, the end-to-end delay of $P_{k,j}$ should satisfy the following condition: $R_{k,j} = Y^f_{k,j} + Y^t_{k,j} + C_k \leq D_k$. To make flow F_k schedulable, this condition should hold for all its packets.

C. Upper Bound of Interferences

The conflict and contention delay bounds in Lemma 1 and Lemma 2 depend on the conflict and contention interferences. To give the worst-case end-to-end delay R_k of each flow F_k, the most straightforward approach is to compute every other flow F_l's conflict interference and contention interference on every packet of F_k up to the hyper-period. However, this is computationally expensive. We therefore derive upper bounds of the interferences that can be computed efficiently.

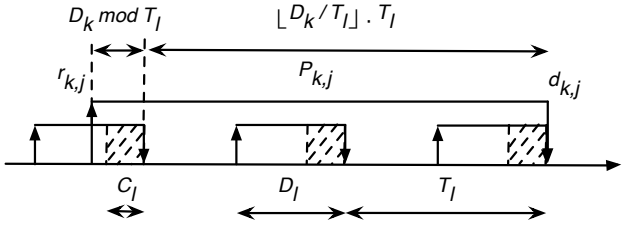


Fig. 1: Worst-case workload of flow F_l. Upper arrows and down arrows represent release times and absolute deadlines of packets. $P_{k,j}$ is the k^{th} packet of flow F_k. C_l, D_l and T_l are the transmission count, the relative deadline and the period of F_l respectively. $r_{k,j}$ and $d_{k,j}$ are the release time and the absolute deadline of packet $P_{k,j}$. Dashed areas are time slots when transmissions of packets are scheduled.

To start, we analyze the upper bound of the interference $I_{k,j}(l)$ which is the sum of conflict interference $I^f_{k,j}(l)$ and contention interference $I^t_{k,j}(l)$. It is obvious that the interference of flow F_l on any packet $P_{k,j}$ cannot exceed its workload within $P_{k,j}$'s lifetime $[r_{k,j}, d_{k,j}]$, where flow F_l's workload is the number of time slots when it executes. We denote the workload of F_l within $[r_{k,j}, d_{k,j}]$ as $W_{k,j}(l)$. The worst-case workload would be a upper bound of the interference. We show the worst-case workload in Figure 1 when the absolute deadline of one packet of F_l aligns with the absolute deadline

of $P_{k,j}$. We give the following lemma to upper bound F_l's workload.

Lemma 3. *The workload of F_l within $[r_{k,j}, d_{k,j}]$ is upper bounded as follow:*

$$W_{k,j}(l) \leq \lfloor D_k/T_l \rfloor C_l + \min(C_l, D_k \ mod \ T_l),$$

where $D_k \ mod \ T_l$ is the remainder of D_k divided by T_l.

Proof: We discuss the workload of F_l in three cases:

- $D_k < T_l$
- $D_k \geq T_l$ and $(D_k \ mod \ T_l) < D_l$
- $D_k > T_l$ and $(D_k \ mod \ T_l) \geq D_l$

In the first case, deadline of flow F_k is less than period of flow F_l. Within $[r_{k,j}, d_{k,j}]$, there is at most one packet of F_l active. Then the maximum workload of F_l is $\min(C_l, D_k)$, which follows this lemma.

In the second case, deadline of F_k is no less than period of F_l, and $D_k \ mod \ T_l$ is less than D_l. First, if $D_k \ mod \ T_l$ equals 0, then the number of flow F_l's packets within $[r_{k,j}, d_{k,j}]$ is D_k/T_l, and the total workload is $(D_k/T_l)C_l$, which follows this lemma. Then, we assume $D_k \ mod \ T_l > 0$, this is the exact case we show in Figure 1. there is one carry-in packet of flow F_l, which is released before $P_{k,j}$ and delivered after $P_{k,j}$'s release. The number of F_l's packets within $[r_{k,j}, d_{k,j}]$ is $\lfloor D_k/T_l \rfloor$. And the carry-in packet's workload is $\min(C_l, D_k \ mod \ T_l)$. Then the total workload is $\lfloor D_k/T_l \rfloor C_l + \min(C_l, D_k \ mod \ T_l)$, which also follows this lemma.

In the third case, the number of packets of F_l that are completely contained in $[r_{k,j}, d_{k,j}]$ is $\lfloor D_k/T_l \rfloor + 1$, given that $D_k \ mod \ T_l \geq D_l$. The workload of F_l is $(\lfloor D_k/T_l \rfloor + 1)C_l$. Because $D_k \ mod \ T_l \geq D_l \geq C_l$, the workload provided by this lemma is

$$\begin{aligned}\lfloor D_k/T_l \rfloor C_l + \min(C_l, D_k \ mod \ T_l) =& \lfloor D_k/T_l \rfloor C_l + C_l \\ =& (\lfloor D_k/T_l \rfloor + 1)C_l.\end{aligned}$$

Then this case follows the lemma as well. ∎

We have an upper bound of F_l's interference on packet $P_{k,j}$ as:

$$I_{k,j}(l) \leq W_{k,j}(l) \leq \lfloor D_k/T_l \rfloor C_l + \min(C_l, D_k \ mod \ T_l). \quad (5)$$

We use $\widehat{I_{k,j}(l)}$ to denote this upper bound. Then we have

$$\widehat{I_{k,j}(l)} = \lfloor D_k/T_l \rfloor C_l + \min(C_l, D_k \ mod \ T_l). \quad (6)$$

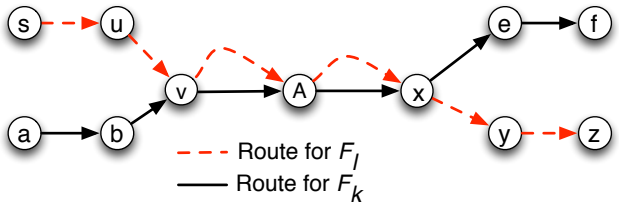


Fig. 2: An example to show conflict delay

Now we derive the upper bound of flow F_l's conflict interference $I^f_{k,j}(l)$ on packet $P_{k,j}$. Let $S_k(l)$ denote the maximum conflict interference that one packet of flow F_l can incur on one packet of flow F_k. Packets of flows F_k and F_l conflict with each other when their transmissions share at least one node. So $S_k(l)$ is the number of transmissions that share nodes with F_l's transmissions. It depends on the number of links in F_l's route that share nodes with F_k's route as well as the number of (re)transmissions scheduled on each link. We can count it by looking at routes of two flows.

As shown in Figure 2, F_k and F_l are two flows that share a part of their routes. The number of links in F_l's route that share nodes with F_k's route is 4, and they are $\{u \to v, v \to A, A \to x, x \to y\}$. For simplicity, assuming only one transmission is scheduled for each link, $S_k(l)$ in this example equals 4. After upper bounding the conflict interference that one packet of F_l can introduce, we can upper bound the total conflict interference that flow F_l can introduce on packet $P_{k,j}$.

Following the same reasoning of analyzing the maximum workload, we have the following corollary.

Corollary 1. *The conflict interference of flow F_l on packet $P_{k,j}$ is upper bounded as follow:*

$$I^f_{k,j}(l) \leq \lfloor D_k/T_l \rfloor S_k(l) + \min(S_k(l), D_k \ mod \ T_l). \quad (7)$$

Here we use $\widehat{I^f_{k,j}(l)}$ to denote this upper bound of $I^f_{k,j}(l)$, and

$$\widehat{I^f_{k,j}(l)} = \lfloor D_k/T_l \rfloor S_k(l) + \min(S_k(l), D_k \ mod \ T_l). \quad (8)$$

After upper bounding the interferences of flow F_l on packet $P_{k,j}$, we have the upper bound of end-to-end delay in the following theorem.

Theorem 1. *The end-to-end delay of flow F_k is upper bounded as follow:*

$$R_k \leq \sum_{l \neq k} \widehat{I^f_{k,j}(l)} + \lfloor \frac{\sum_{l \neq k}(\widehat{I_{k,j}(l)} - \widehat{I^f_{k,j}(l)})}{m} \rfloor + C_k = \hat{R}_k.$$

Proof: As we showed in Equation (1), $R_{k,j} = Y^f_{k,j} + Y^t_{k,j} + C_k$. Combining it with Lemma 1 and Lemma 2, we have

$$R_{k,j} \leq \sum_{l \neq k} I^f_{k,j}(l) + \lfloor \frac{1}{m} \sum_{l \neq k} I^t_{k,j}(l) \rfloor + C_k. \quad (9)$$

From Equations (6) and (8), we have upper bounds of $I_{k,j}(l)$ and $I^f_{k,j}(l)$. However, we don't have an upper bound of $I^t_{k,j}(l)$. Based on Equation (2), $I_{k,j}(l) = I^f_{k,j}(l) + I^t_{k,j}(l)$. In any case, $I^f_{k,j}(l) + I^t_{k,j}(l)$ cannot exceed $\widehat{I_{k,j}(l)}$. And $I^f_{k,j}(l)$ cannot exceed $\widehat{I^f_{k,j}(l)}$. As shown in Equation (9), $I^t_{k,j}(l)$ is divided by m and floored, which shows $I^f_{k,j}(l)$ has higher weight than $I^t_{k,j}(l)$ in the equation. Given the fixed upper bound of $\widehat{I_{k,j}(l)}$, any amount that we reduce from conflict interference and

add to contention interference will not increase the end-to-end delay. So we use $\widehat{I_{k,j}(l)} - I^f_{k,j}(l)$ to replace $I^t_{k,j}(l)$, use $I^f_{k,j}(l)$ to replace $I^f_{k,j}(l)$ and get the upper bound of R_k as the theorem shows. The intuition is we would rather overestimate conflict interference and underestimate contention interference, which will not violate the safety of our upper bound. ■

D. Improved Delay Analysis

We give an upper bound of end-to-end delay in Theorem 1. Bertogna et al. [7] proposed a technique to iteratively improve scheduability analysis. Inspired by their technique, In this subsection, we propose an *Improved Delay Analysis (IDA)*. From now on, we will call the end-to-end delay analysis in Theorem 1 as the *Basic Delay Analysis (BDA)* and use it as a foundation of our IDA.

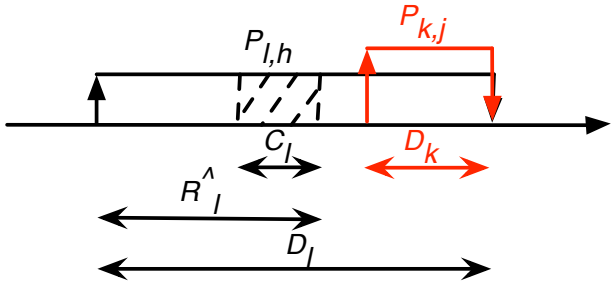


Fig. 3: An example for Observation 1. $\hat{R}_l$ is an upper bound of the end-to-end delay of flow F_l we obtained through BDA (Theorem 1).

We illustrate the intuition of IDA through an example shown in Figure 3. We consider the flow F_l's interference on packet $P_{k,j}$. In this example the deadline of packet $P_{l,h}$ is aligned with deadline of packet $P_{k,j}$. The upper bound of end-to-end delay of F_l is shown in the figure with $\hat{R}_l$. From the figure, $P_{l,h}$ is delivered to the destination before absolute deadline of $P_{l,h}$ as well as release time of $P_{k,j}$. Then all transmissions of $P_{l,h}$ are scheduled before the release of $P_{k,j}$. Clearly, the interference of F_l on $P_{k,j}$ is zero. However, based on BDA, the conflict interference of F_l on $P_{k,j}$ is $S_k(l)$ (maximum conflict delay that $P_{l,h}$ can incur on $P_{k,j}$), and the contention interference of F_l is not zero either. BDA hence overestimates the interference in this example, because it ignores the fact that $P_{l,h}$ is delivered well before its deadline.

This observation leads to a way to reduce the pessimism of our analysis. In BDA, the most difficult part is to assure the schedulability of flows with short deadlines, because an overestimation of interference would easily push the end-to-end delay bound of the packet over its short deadline. The intuition behind IDA is to tighten up the interference estimation by considering early completion of packets.

Observation 1. *Let $\hat{R}_l$ denote an upper bound of end-to-end delay of flow F_l, no transmissions of packet $P_{l,h}$ can be scheduled later than $r_{l,h} + \hat{R}_l$.*

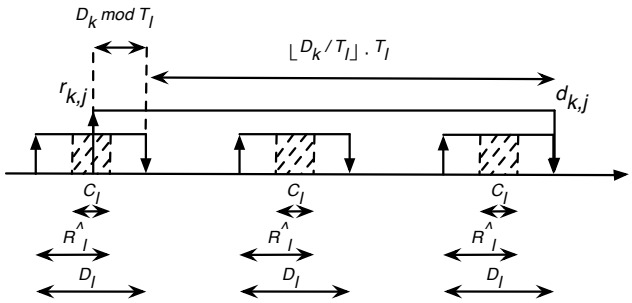


Fig. 4: Worst-case scenario under Observation 1

By incorporating this observation, we propose our IDA. In IDA, we use * to denote the new results of variables we already introduced in BDA.

We start by analyzing upper bounds of interferences. We show the worst-case interference of F_l on packet $P_{k,j}$ in Figure 4. Note that we also show the upper bound of end-to-end delay $\hat{R}_l$ of F_l in the figure.

Lemma 4. *Flow F_l's interference on packet $P_{k,j}$ is upper bounded as follow:*

$$\widehat{I_{k,j}(l)}^* = \min(C_l, \max(0, (D_k \bmod T_l) - (D_l - \hat{R}_l))) + \lfloor D_k/T_l \rfloor C_l. \quad (10)$$

Proof: We discuss in four cases.

- $D_k < T_l$ and $D_k \leq D_l - \hat{R}_l$
- $D_k < T_l$ and $D_k > D_l - \hat{R}_l$
- $D_k \geq T_l$ and $(D_k \bmod T_l) \leq D_l - \hat{R}_l$
- $D_k \geq T_l$ and $(D_k \bmod T_l) > D_l - \hat{R}_l$

In the first case, deadline of flow F_k is less than period of flow F_l. Within $[r_{k,j}, d_{k,j}]$, there is at most one packet of F_l active. Since deadline of F_k is no greater than gap between F_l's upper bound of end-to-end delay $\hat{R}_l$ and its deadline D_l. All transmissions of F_l are scheduled before $P_{k,j}$'s release time. Then F_l's interference equals zero in this case, which follows this lemma. This case is exactly what we show in Figure 3.

In the second case, there is also at most one packet of F_l active. However, the release time of $P_{k,j}$ is before F_l's upper bound of end-to-end delay. Packet $P_{k,j}$ is released $D_k - (D_l - \hat{R}_l)$ time slots before one packet of F_l complete its transmissions. Then the maximum possible interference is $\min\{C_l, D_k - (D_l - \hat{R}_l)\}$, which also follows this lemma.

In the third case, deadline of F_k is no less than period of F_l, and $D_k \bmod T_l$ is no greater than $D_l - \hat{R}_l$. First, if $D_k \bmod T_l$ equals 0, then the number packets of F_l within $[r_{k,j}, d_{k,j}]$ is D_k/T_l, and the total interference is $(D_k/T_l)C_l$, which follows this lemma. Then, we assume $0 < D_k \bmod T_l \leq D_l - \hat{R}_l$. There is one carry-in packet of flow F_l, which is delivered before $P_{k,j}$'s release. So its interference on packet $P_{k,j}$ is 0. Then the total interference is $\lfloor D_k/T_l \rfloor C_l$, which also follows this lemma.

In the last case, deadline of F_k is no less than period of F_l, and $D_k \ mod \ T_l$ is larger than $D_l - \hat{R}_l$. If $D_k \ mod \ T_l \geq D_l$, there are D_k/T_l+1 packets of F_l contained within $[r_{k,j}, d_{k,j}]$, the total interference is $(D_k/T_l + 1)C_l$, which follows this lemma. If $D_l - \hat{R}_l < D_k \ mod \ T_l < D_l$, the carry-in packet of flow F_l is partial within $[r_{k,j}, d_{k,j}]$, which is exactly what we show in Figure 4. The interference of this carry-in packet depends on $D_k \ mod \ T_l - (D_l - \hat{R}_l)$, and equals $\min(C_l, (D_k \ mod \ T_l) - (D_l - \hat{R}_l)$. This case follows the lemma as well. ■

Following the same reasoning of analyzing upper bound of interference, we have following corollary.

Corollary 2. *Flow F_l's conflict interference on packet $P_{k,j}$ is upper bounded as follow:*

$$\widehat{I^f_{k,j}(l)}^* = \min(S_k(l), \max(0, (D_k \ mod \ T_l) - (D_l - \hat{R}_l))) + \lfloor D_k/T_l \rfloor S_k(l). \quad (11)$$

Similar to Theorem 1, we give an upper bound of end-to-end delay of F_k here.

Corollary 3. *The worst-case end-to-end delay of flow F_k is upper bounded as follow:*

$$R_k \leq \sum_{l \neq k} \widehat{I^f_{k,j}(l)}^* + \lfloor \frac{\sum_{l \neq k} (\widehat{I_{k,j}(l)}^* - \widehat{I^f_{k,j}(l)}^*)}{m} \rfloor + C_k = \hat{R_k}^*. \quad (12)$$

The flow set $\{F_1, F_2, \cdots, F_n\}$ is schedulable if the following statement is true:

$$\hat{R_k}^* \leq D_k, \qquad k = 1, 2, \cdots, n. \quad (13)$$

We use an iterative algorithm to derive the upper bound of end-to-end delay $\hat{R_k}^*$. In the beginning, the initial upper bound $\hat{R_k}$ is set to D_k for all flows. In each iteration, $\hat{R_k}^*$ is calculated based on Equation (10)-(12). At the end of each iteration, for each flow F_k, $\hat{R_k}$ is set to $\hat{R_k}^*$. The algorithm enters a new iteration if the flow set is unschedulable and at least one flow has $\hat{R_k}^*$ updated, otherwise it terminates. We show the pseudo-code in Algorithm 1.

Algorithm 1: Iterative algorithm

```
R̂_k ← D_k, ∀k ≤ N;
repeat
    for k ≤ N do
        R̂_k^0 ← R̂_k;
        Calculate R̂_k^* based on (10)-(12);
        R̂_k ← R̂_k^*;
    end
until R̂_k ≤ D_k or R̂_k = R̂_k^0, ∀k ≤ N;
R̂_k^* ← R̂_k, ∀k ≤ N;
```

E. Complexity Analysis

BDA (Theorem 1) is polynomial. The calculation of upper bound of end-to-end delay of flow F_k is $O(n)$ since we have n flows. The complexity of BDA is $O(n^2)$ since we need to calculate the upper bound of end-to-end delay for every flow. The total time complexity is therefore $O(n^2)$.

IDA (Corollary 3) is pseudo-polynomial. The analysis in each iteration is $O(n^2)$ as discussed above. Since there are n flows, and each one's end-to-end delay can range from C_k to D_k, the number of iterations is upper bounded as $O(n \max(D_k - C_k, k \leq N))$. Thus, the overall complexity is $O(n^3 \max(D_k - C_k, k \leq N))$.

V. Evaluation

We evaluate our end-to-end delay analysis through both experiments on a physical WSAN testbed and simulations.

A. Experiments on a WSAN Testbed

We evaluate our delay analysis on an indoor WSAN testbed consisting of 63 TelosB motes, located on the fifth floors of Bryan Hall and Jolly Hall of Washington University in St. Louis. We implement a network protocol stack on the testbed, which comprises a multi-channel TDMA MAC protocol and a routing protocol. Time is divided into 10 ms slots and clocks are synchronized across the entire network using the Flooding Time Synchronization Protocol (FTSP) [8]. In the routing protocol, we want to find the maximum number of link-disjoint paths for any pair of nodes. We transform this problem into a maximum flow problem by assigning each link with unit capacity and use Edmonds−Karp algorithm [9] to generate link-disjoint routes.

Figure 5 shows the topology of the WSAN testbed. We use motes 129 and 155 (colored in green) as access points, which are physically connected to a root server (Gateway). The Network Manager runs on this root server. The rest of motes work as field devices. For each link in the testbed, we measured its *packet reception ratio (PRR)* by counting the number of received packets among 250 packets transmitted on the link. Following the practice of industrial deployment, we only add links with PRR higher than 90% to the topology of the testbed. To avoid channels occupied by the campus Wi-Fi, we use IEEE 15.4 channel 11 to 15 in our experiments.

We generate 8 flows in our experiment. The period of each flow is picked up from the range of $2^{0\sim7}$ seconds, which are typical periods used in process industry as defined in WirelessHART standard [1]. The length of the hyper-period is 128 seconds. The relative deadline of each flow equals to its period. All flows are schedulable based on our delay analyses. Each flow has two independent source routes. The maximum length of routes is 13 hops. Through this double-route approach, we enhance the network reliability under link failures. We run our experiments long enough such that each flow can deliver at least 100 packets.

Based on our experimental results, we evaluate our proposed approaches in terms of reliability and delay. We use *delivery ratio* to measure reliability. The *delivery ratio* of a flow is

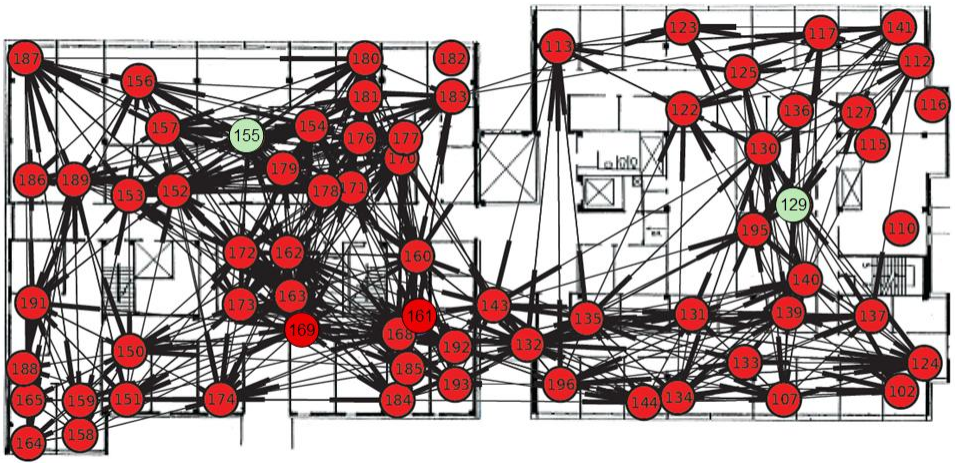


Fig. 5: Topology of the WSAN Testbed. Green circles represent access points. Red circles are field devices. Black arrows are wireless links.

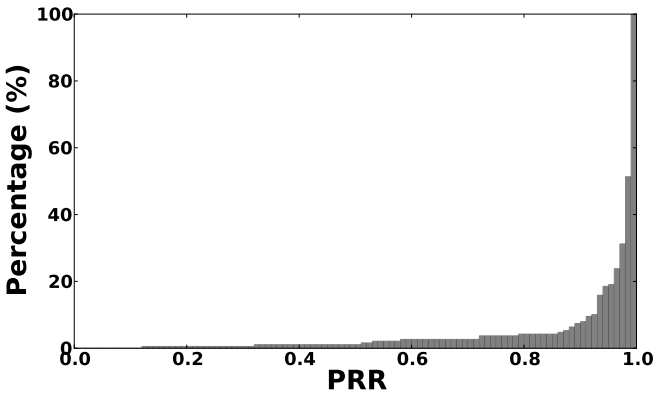


Fig. 6: Cumulative Histogram of Link Qualities

Flow Index	1	2	3	4	5	6	7	8
1^{st} Route	0.95	1.0	0.97	1.0	0.97	0.96	0.97	0.97
2^{nd} Route	1.0	1.0	0.99	0.99	1.0	0.97	1.0	0.42
Two Routes	1.0	1.0	1.0	1.0	1.0	1.0	1.0	0.99

TABLE I: Delivery Ratios of Flows

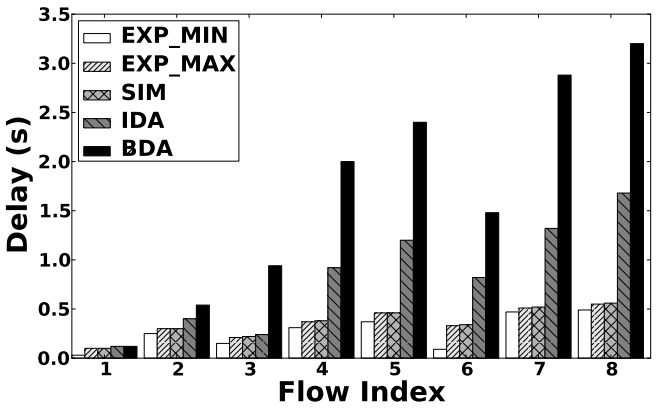


Fig. 7: End-to-End Delays

defined as percentage of packets that are successfully delivered to destination. Then, we compare the end-to-end delay we collected in experiments with our delay analyses, as well as the delay observed in simulations.

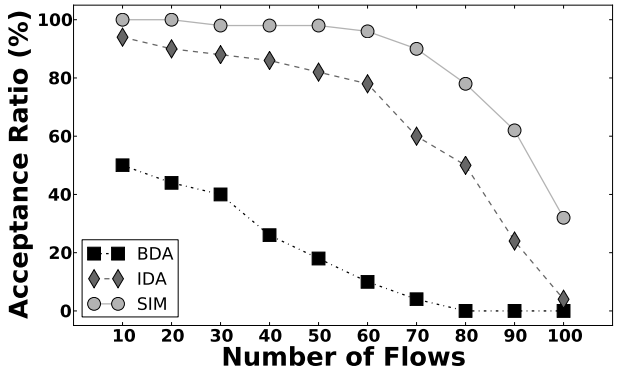
To study the reliability issue, we first measure the link qualities in our testbed . Figure 6 shows the cumulative histogram of link qualities (PRR) of 189 links we used in our experiments. Although we only picked up links that have PRR higher than 90% when we selected the links initially, we find some links have much lower PRR than the 90% threshold at run time. For example, link $112 \rightarrow 129$ has the lowest PRR of 12%. The dynamics of wireless links suggest it is necessary to have route redundancy.

Table I shows the delivery ratios of all 8 flows. We present the delivery ratio of each route as well as the aggregate delivery ratio of the two routes combined. Our results demonstrate the effectiveness of redundant routes in improving reliability. For example, the second route of flow 8 has a delivery ratio of 0.42, which is much lower than our expectation. However, by combing two routes together, flow 8 has a delivery ratio as 0.99.

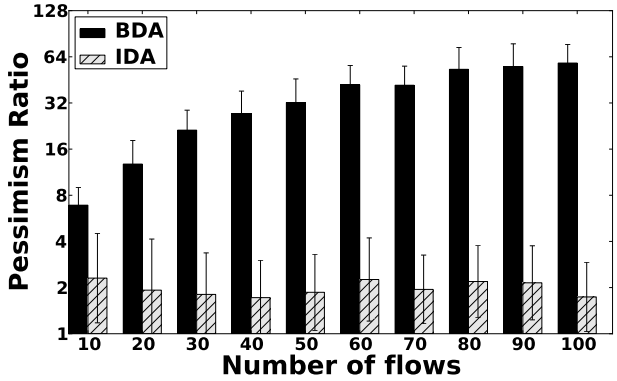
In Figure 7, we compare end-to-end delay from experiment results with delay analyses as well as simulation. We compare five delays for each flow: minimum delay in experiments (EXP_MIN), maximum delay in experiments (EXP_MAX), maximum delay in simulation (SIM), the improved delay analysis (IDA) in Corollary 3 and the basic delay analysis (BDA) in Theorem 1. The results show for every flow, the five delays follow the following order:

$\text{EXP\_MIN} \leq \text{EXP\_MAX} \leq \text{SIM} \leq \text{IDA} \leq \text{BDA}$.

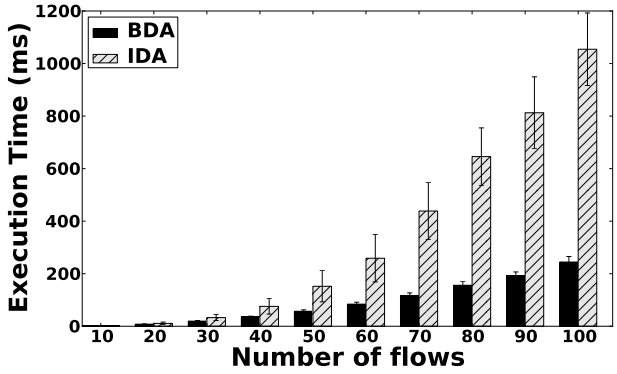
This shows that our delay analyses are safe upper bounds of the actual delays. In addition, SIM is consistently higher than EXP_MAX, which indicates our simulations can generate test cases with worse delays than those observed on the testbed. In following evaluation, we will provide a more comprehensive evaluation of the delay analyses based on simulations over different network topologies.



(a) Acceptance Ratio



(b) Pessimism Ratio (in log scale)



(c) Execution Time

Fig. 8: Schedulability Analysis on Random Topology

B. Simulations on Random Topologies

Besides the Testbed experiments, we also test our analyses on larger random topologies with simulations. The simulator shares the same routing and scheduling design with our testbed and is written in C++. All simulations are performed on a MacBook Pro laptop with 2.4 GHz Intel Core 2 Duo processor. We generate random networks with 400 nodes and 800 links. Links are chosen randomly and assigned PRR randomly in the range of $[0.90, 1.0]$. We test our delay analyses on different number of flows by increasing the number of source and destination pairs. The period T_k of the each flow F_k is randomly generated in the range of $2^{3\sim9}$ seconds. The relative deadline D_k of every flow F_k is randomly generated in the range of $(C_k, \beta * T_k)$ slots, here β is a randomly generated number in range of $(0, 1)$. C_k is the required time slots needed to deliver a packet from the source to the destination. For each flow set, we generate 100 test cases and simulate them on random topologies.

We compare our improved delay analysis (IDA) in Corollary 3 with the basic delay analysis (BDA) in Theorem 1 and the simulation (SIM). Our delay analyses are evaluated in terms of *pessimism ratio* and *acceptance ratio*. The former one is used to assess the tightness of the delay analyses, and the latter one is used to evaluate the effectiveness of our analyses for online admission control. For each flow, the *pessimism ratio* is defined as the ratio of its theoretical upper bound of end-to-end delay given by our analyses to its maximum end-to-end delay observed in simulation. The *acceptance ratio* is defined as the ratio of the number of test cases deemed schedulable by our analyses (or simulation) to the total number of test cases. A test case is schedulable in simulation if all flow instances released within the hyper-period meet their deadlines.

The acceptance ratios of IDA, BDA and simulation (SIM) are shown in Figure 8(a). The acceptance ratio of IDA remains close to simulations. The gap between IDA and SIM widens as the number of flows increases, but remains within 30%. This result indicates the effectiveness of IDA for admission control. The acceptance ratio of IDA is much higher than BDA, which shows the IDA highly outperforms BDA in terms of acceptance ratio.

Figure 8(b) shows pessimism ratios of IDA and BDA in log scale. Since if a test case is not schedulable under simulation, the simulator could not lay out the schedule of all flows, then we could not get the actual maximum end-to-end delay. So all pessimism ratios here are from test cases that are schedulable under simulation. This result confirms IDA greatly improves the tightness of the delay bound compared to BDA. The pessimism ratio of BDA increases as the number of flows increases. However, the pessimism ratio of IDA remain low despite the increase of number of flows. The median value of pessimism ratio for IDA is always around 2 in our simulations. This figure shows our IDA is scalable to large number of flows.

The time complexity of our algorithms are shown in Figure 8(c). The execution time of IDA grows faster than BDA as the number of flows grows while staying in an acceptable region. With 100 flows, the execution time of IDA is under 1.2 seconds, which is acceptable for admission control. Figures 8(a)-8(c) show the tradeoff between accuracy and time complexity. While IDA runs slower than BDA, it gives a much more precise estimation of end-to-end delay, which leads to a higher acceptance ratio.

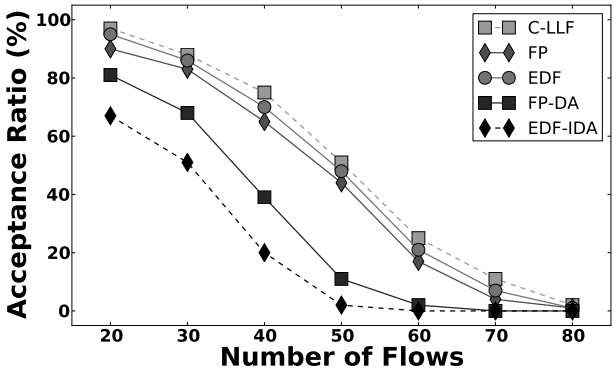
C. Comparative Study of Scheduling Policies

In this subsection, we compare EDF and our Improved Delay Analysis (EDF-IDA) with state-of-the-art dynamic and static priority scheduling algorithms and their delay analyses. For dynamic priority scheduling, we consider Conflict-aware

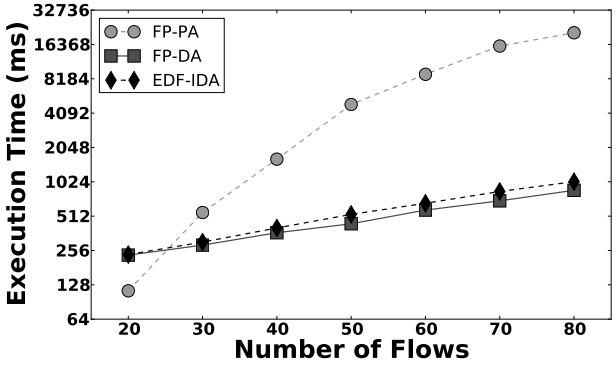
Least Laxity First (C-LLF) [5], which incorporates transmission conflicts into a Least Laxity First scheduling policy. However, there is no delay analysis for C-LLF in the literature. For fixed priority scheduling, we choose Fixed Priority with near optimal priority assignment based on heuristic search (FP) presented in [6], which was shown to significantly outperform traditional priority assignment policies. Delay Analysis for Fixed Priority scheduling policies (FP-DA) has been proposed in [10].

We compare scheduling policies through simulations on random topologies. To fully test the schedulability of different scheduling algorithms, we reduce the periods of flows from the range of $2^{6\sim11}$ to $2^{5\sim10}$. The rest of the simulation setups are same as the previous subsection. Figure 9(a) shows the acceptance ratios of different scheduling policies and their delay analyses. Since these is no delay analysis for C-LLF, we only include its acceptance ratio in simulations. Results show in simulations, EDF can schedule more flow sets than FP, which indicates that EDF is indeed an effective scheduling policy in practice. While C-LLF can schedule more flow sets than EDF, there is no schedulability analysis for C-LLF that can be used for fast online admission control. We also compare the acceptance ratios of delay analyses EDF-IDA and FP-DA here. Given the complexity that EDF brings to schedulability analysis, the acceptance ratio of EDF-IDA is slightly lower than FP-DA.

Figure 9(b) shows execution time (in log scale) of the delay analyses as well as the priority assignment algorithm needed by FP (denoted as FP-PA). The execution time of FP-PA is much higher than execution time of EDF-IDA and FP-DA. Note that priority assignment is an integral part of fixed priority scheduling and the near optimal priority assignment needs to be performed for admission control of flows. Given the high computational cost of priority assignment algorithm, EDF-IDA provides a more efficient admission test for real-time flows, which is particularly important for WSANs operating under dynamic wireless conditions in industrial environments.



(a) Acceptance Ratio



(b) Execution Time (in log scale)

Fig. 9: Comparison of Different Scheduling Policies

VI. Related Works

Real-time transmission scheduling in wireless sensor networks received considerable attention [11]. In contrast to previous works on traditional sensor networks, our research investigates real-time WSANs based on recent industrial standards such as WirelessHART with unique features. For example, our analysis is designed for wireless mesh networks running a multi-hop and multi-channel TDMA protocol. In contrast, the network model in previous works is based on single channel [12]–[17], or CSMA/CA MAC protocol [17], [18]. While some earlier works [12]–[16] analyze fixed priority scheduling in wireless networks, we focus on EDF scheduling, which is a dynamic priority scheduling policy. The probabilistic delay analyses for EDF proposed in [19] are not suitable for industrial WSANs that require safe bounds on network delays. Earlier efforts on real-time schedulability analysis for EDF [20], [21] adopt a cellular network structure and require wireless nodes with full-duplex transceivers.

In the area of industrial WSANs, earlier works study the transmission scheduling for WSANs with simple topologies such as linear [22], tree [23], [24] and cluster tree topologies [25]. Transmission scheduling of real-time flows for arbitrary WSAN topologies has been studied in [5]. It presents a real-time scheduling algorithm based on branch-and-bound and a dynamic priority scheduling algorithm called C-LLF. However, it does not present any delay analysis to derive its delay bound, therefore requires laying out the entire transmission schedule of the whole network, which incurs high computation delays in admission control. Near optimal rate selection for fixed priority scheduling has been studied in [26], [27]. End-to-end delay analysis for fixed priority scheduling in WSANs has been proposed in [10], [28]. The performance of fixed priority scheduling highly depends on the priority assignment, which is proven to be a difficult problem, and near-optimal priority assignment algorithms incur significant computational cost when used online.

While dynamic priority scheduling represents an attractive alternative to fixed priority scheduling, end-to-end delay analysis for dynamic priority scheduling has not been studied for WSANs. Our work provides an end-to-end delay analysis for EDF scheduling policy, which is a commonly used dynamic priority scheduling algorithm in real-time systems [7], [29], [30] and outperforms the state-of-the-art fixed priority scheduling algorithms in WSANs in our simulation study.

VII. Conclusions

With the emergence of industrial standards such as WirelessHART, wireless sensor-actuator networks (WSANs) are gaining rapid adoption in process industries. To meet the stringent real-time performance requirements of process control systems, there is a critical need for fast end-to-end delay analysis to support online admission control of periodic real-time flows in WSANs. This paper presents a new end-to-end delay analysis for WSANs under Earliest Deadline First (EDF) transmission scheduling, a widely used dynamic priority scheduling policy in real-time systems. Our analysis that can be used to derive end-to-end delay bounds for real-time flows in WSANs at moderate run time overhead. Experiments on a physical WSAN testbed and simulations demonstrate the effectiveness of our analysis for online admission control of real-time flows.

VIII. Acknowledgments

This work is supported by NSF through grants CNS-1035773 (CPS), CNS-1320921 (NeTS) and CNS-1144552 (NeTS).

References

[1] "WirelessHART specification," 2007, http://www.hartcomm2.org.
[2] "ISA 100," http://www.isa.org/isa100.
[3] J. Song, S. Han, A. Mok, D. Chen, M. Lucas, and M. Nixon, "WirelessHART: Applying Wireless Technology in Real-Time Industrial Process Control," in *RTAS'08*.
[4] J. Song, A. K. Mok, D. Chen, and M. Nixon, "Challenges of Wireless Control in Process Industry," in *Workshop on Research Directions for Security and Networking in Critical Real-Time and Embedded Systems,*, 2006.
[5] A. Saifullah, Y. Xu, C. Lu, and Y. Chen, "Real-Time Scheduling for WirelessHART Networks," in *RTSS'10*.
[6] ——, "Priority Assignment for Real-time Flows in WirelessHART networks," in *ECRTS'11*.
[7] M. Bertogna, M. Cirinei, and G. Lipari, "Schedulability Analysis of Global Scheduling Algorithms on Multiprocessor Platforms," *IEEE Transactions on Parallel and Distributed Systems*, 2009.
[8] M. Maróti, B. Kusy, G. Simon, and A. Lédeczi, "The Flooding Time Synchronization Protocol," in *SenSys'04*.
[9] J. Edmonds and R. M. Karp, "Theoretical Improvements in Algorithmic Efficiency for Network Flow Problems," *Journal of the ACM*, 1972.
[10] A. Saifullah, Y. Xu, C. Lu, and Y. Chen, "End-to-end Delay Analysis for Fixed Priority Scheduling in WirelessHART networks," in *RTAS'11*.
[11] J. Stankovic, T. Abdelzaher, C. Lu, L. Sha, and J. Hou, "Real-time communication and coordination in embedded sensor networks," *Proceedings of the IEEE*, 2003.
[12] O. Chipara, C. Lu, and G.-C. Roman, "Real-Time Query Scheduling for Wireless Sensor Networks," in *RTSS '07*.
[13] P. Jurcík, R. Severino, A. Koubâa, M. Alves, and E. Tovar, "Real-Time Communications Over Cluster-Tree Sensor Networks with Mobile Sink Behaviour," in *RTCSA '08*.
[14] O. Chipara, C. Wu, C. Lu, and W. Griswold, "Interference-Aware Real-Time Flow Scheduling for Wireless Sensor Networks," in *ECRTS'11*.
[15] N. Pereira, B. Andersson, E. Tovar, and A. Rowe, "Static-Priority Scheduling over Wireless Networks with Multiple Broadcast Domains," in *RTSS '07*.
[16] T. F. Abdelzaher, S. Prabh, and R. Kiran, "On Real-Time Capacity Limits of Multihop Wireless Sensor Networks," in *RTSS '04*.
[17] R. Oliver and G. Fohler, "Probabilistic Estimation of End-to-End Path Latency in Wireless Sensor Networks," in *MASS'09*.
[18] J. B. Schmitt and U. Roedig, "Sensor Network Calculus - A Framework for Worst Case Analysis," in *DCOSS '05*.
[19] P. Jayachandran and M. Andrews, "Minimizing End-to-End Delay in Wireless Networks Using a Coordinated EDF Schedule," in *INFOCOM'10*.
[20] M. Caccamo, L. Y. Zhang, L. Sha, and G. Buttazzo, "An Implicit Prioritized Access Protocol for Wireless Sensor Networks," in *RTSS'02*.
[21] M. Caccamo and L. Y. Zhang, "The Capacity of Implicit EDF in Wireless Sensor Networks," in *ECRTS'03*.
[22] H. Zhang, P. Soldati, and M. Johansson, "Optimal Link Scheduling and Channel Assignment for Convergecast in Linear WirelessHART Networks," in *WiOpt'09*.
[23] P. Soldati, H. Zhang, and M. Johansson, "Deadline-Constrained Transmission Scheduling and Data Evacuation in WirelessHART Networks," in *ECC '09*.
[24] H. Zhang, F. Osterlind, P. Soldati, T. Voigt, and M. Johansson, "Rapid Convergecast on Commodity Hardware: Performance Limits and Optimal Policies," in *SECON '10*.
[25] E. Toscano and L. Lo Bello, "Multichannel Superframe Scheduling for IEEE 802.15.4 Industrial Wireless Sensor Networks," *IEEE Transactions on Industrial Informatics*, 2012.
[26] A. Saifullah, C. Wu, P. Tiwari, Y. Xu, Y. Fu, C. Lu, and Y. Chen, "Near Optimal Rate Selection for Wireless Control Systems," in *RTAS'12*.
[27] ——, "Near Optimal Rate Selection for Wireless Control Systems," *ACM Transactions on Embedded Computing Systems*, 2014.
[28] A. Saifullah, Y. Xu, C. Lu, and Y. Chen, "End-to-End Communication Delay Analysis in Industrial Wireless Networks," *IEEE Transactions on Computers*, accepted.
[29] T. Baker, "Multiprocessor EDF and Deadline Monotonic Schedulability Analysis," in *RTSS'03*.
[30] M. Bertogna, M. Cirinei, and G. Lipari, "Improved Schedulability Analysis of EDF on Multiprocessor Platforms," in *ECRTS'05*.