IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received July 25, 2018, accepted August 24, 2018, date of publication August 31, 2018, date of current version September 28, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2868099

An Efficient Retransmission Scheme for Reliable End-to-End Wireless Communication Over WSANs

JIAN MA[1], DONG YANG[1], (Member, IEEE), HONGCHAO WANG[1], AND MIKAEL GIDLUND[2], (Member, IEEE)

[1]Beijing Jiaotong University, Beijing 100044, China
[2] Department of Information and Communication Systems, Mid Sweden University, SE-851 70 Sundsvall, Sweden

Corresponding author: Dong Yang (dyang@bjtu.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 61771040 and in part by the Electronic Information Technology Innovation and Cultivation under Grant Z171100001217004.

ABSTRACT A wireless sensor and actuator network is increasingly recognized as an important technology in the realization of the future Internet of Things. The ability to cater to the demands of real-time, reliable, and resource-constraint communication in industrial areas is highly dependent on efficient scheduling of communication links. Harsh industrial environments make packet retransmission inevitable, which causes the waste of link resource and deteriorates the reliability. Most of the studies that focus on real-time and reliable performance underestimate the complexity of packet retransmission. Therefore, this paper proposes an efficient retransmission scheme to guarantee deterministic communication and decrease the resource utilization. We combine deterministic communication with a novel reliable method by proposing a flow-based slot scheduling with a concession timeslot assistant. The proposed concession slot works by using the CCA and the pre-signal to avoid the contention of one channel during the communication on shared links. Furthermore, we use the theory of discrete-time Markov chain to analyze another two randomly backoff-based and flow-based shared link schedules. This paper demonstrates that the proposed retransmission scheme can significantly improve the reliability of end-to-end packet delivery and the efficiency of slot utilization, as well as decrease energy consumption. The proposed scheduling is applied in a real factory, where the efficiency of the slot is significantly improved, and a flow reliability of 95.3% under a 12% packet error rate is guaranteed.

INDEX TERMS Wireless sensor and actuator networks (WSANs), time slotted channel hopping (TSCH), reliable wireless communication.

I. INTRODUCTION

Wireless communication is increasingly recognized as an important technology in the realization of the future Internet of Things (IoT) [1]. Wireless Sensor Networks (WSNs) is a branch of wireless network technology, and it is gradually becoming more advanced with low-power consumption and easy deployment as key features. It is used in a variety of fields, such as medical systems, transportation and military. In industrial manufacturing, Wireless Sensor and Actuator Networks (WSANs) are emerging as a part of WSNs focusing on monitoring and control of the field devices. As a key technology in the recent surge in research on Industry 4.0, Factories of the Future (FoF) and Industrial Internet of Things (IIoT), WSANs can branch out to include application in industry [2], [3]. However, the stringent reliability requirements and time efficiency as well as harsh conditions such as complicated environment and metal interference, result in more packet loss in WSANs transmissions compared with traditional WSNs [4]. Today, there are several dominant standards, such as WirelessHART, ISA100.11a, WIA-PA and IEEE 802.15.4e, all of which employ the Timeslotted Channel Hopping (TSCH) Medium Access Control (MAC) protocol instead of pure contention-based MAC protocol carrier-sense multiple access with collision avoidance (CSMA/CA). The IEEE802.15.4e-2012 standard defines a number of MAC protocols for IEEE802.15.4. In this paper, we focus on TSCH, which inherits from WirelessHART and ISA100.11a. However, these issued standards do not give the

specific link scheduling for efficient data retransmission. An obstacle faced in guaranteeing the deterministic communication in industrial environments, however, is that resources such as slot, channel and energy are restricted [5], [6].

Focus has been on meeting these restrictions: there has been much research on resource scheduling design by introducing priority [7]–[9], analyzing end-to-end delay [10] and reducing the amount of transmitted data [11]–[13]. The key problem is that they did not attach importance to the effect of packet retransmission on each hop over a multi-hop network. Therefore, the Shared links and the TSCH-CA algorithm are proposed by IEEE 802.15.4e to remedy the packet loss in Guaranteed Time Slot (GTS) [14]. Shared links are intentionally assigned to more than one devices for transmission and TSCH-CA algorithm adopts the slotted backoff scheme to decrease the probability of collision.

The scheduling proposed in [15] involves the shared links but without considering the collision. Yan *et al.* [16] proposed dedicated scheduling and shared scheduling by forming a hypergraph to improve the reliability in harsh control environments. However, the algorithm costs the solver software for hours to identify optimal results. Reference [17] proposes multiple retransmission schemes, but these optimized retransmission schemes are used in single-hop wireless networks. In [18] and [19], the contention-based retransmission schemes are proposed and these authors also analyze the probability of collision. Also in [20], the reliability is improved by adding and sharing retransmission slots. However, these methods still have not dealt well with collision and the abundant idle listening. Reference [21] proposes a novel link scheduling scheme using the virtual token to identify the utilization of the shared links. Though this can allow a more flexible use of available links at each point in time, the increased energy consumption cannot be ignored since all the nodes in a path remain in reception mode before correctly receiving the message. Based on IEEE 802.15.4e, the schedules in [22]–[24] allocate more time slots to guarantee the low latency in critical applications. However, the schedules cannot support enough data flows, since many abundant slots are allocated to nodes even if not necessarily. As for using the approaches of channel preemption, [25] and [26] respectively propose the PriorityMAC and the RushNet, which are able to ensure that the transmission of critical data is prioritized. However, in both of the above approaches, the problems of multi-hop networks, such as hidden terminal problems, exist. Moreover, neither of these articles integrate their approaches into a slot scheduling to highlight their advances. Furthermore, an exhaustive search needs to be adapted in practical deployments [27], however, there is lack of practical evaluations in above researches.

Addressing the above problems, this paper proposes an efficient retransmission scheme by designing a flow-based shared links schedule and a Concession Time Slot (CTS) to improve the reliability of end-to-end transmission and slot utilization. The main contributions of this paper are listed in the following:

1) We use the approaches Clear Channel Assessment (CCA) and pre-signal to make nodes realize the utilization of the shared links for a flow, where it can avoid the contention of shared links and idle listening.

2) We discuss the inadequacies of contention-based shared links and use Discrete-Time Markov Chain (DTMC) to analyze the average end-to-end transmission reliability of two common shared links schedules. We also give a clear comparison between the proposed scheme and other two common schedules.

3) We test our proposed scheduling design in a real factory to prove the improvement of the performance. The WSANs system with the proposed scheduling is in use at a cooperative company.

The remainder of this article is organized as follows. Section II presents the problem statement of the slot scheduling design. Section III introduces the prototype and model of our system. Section IV explains the proposed slot scheduling algorithm and the function of concession slot. Section V analyses the transmission reliability of the scheduling, as well as the scheduling of two other algorithms. The analytical and experimental results are illustrated in Section VI and the paper is concluded in Section VII.

II. CHALLENGES OF THE SCHEDULING DESIGN

On the one hand, an unavoidable event in wireless communication is retransmission, especially under the harsh conditions of industrial environments. The packet loss is so unpredictable that it cannot be remedied by allocating more GTSs to all the nodes for the retransmission. Thus, the relative mechanisms [15] and [28] based on the TSCH-CA are proposed to improve the performance of retransmission. However, the competition of the Shared Time Slot (STS) makes the slot utilization inefficient. Since when a node loses the competition or fails retransmission, it needs to exponentially back off for several slots. The limited number of STS is usually insufficient to guarantee reliable arrival within deadline, especially in a multi-hop network.

On the other hand, an important consideration in time slots scheduling, the slot allocation always depends on the order of routes to a great extent. For example in Fig. 1(a), showing a simple topology with a monitoring application, node 4 is in duty bound to send the collected information to the manager through the flow 4→3→2→1→ sink. An essential slot assignment within a superframe is shown in Fig. 1(c). Four consecutive slots are allocated to each node following the order of routes. However, when node 4 fails to transmit the packet to node 3, the remaining slots allocated to 3→2→1→ sink are wasted. There is a receiver ready to receive the packet with idle listening radio in each remaining slot. Although there is a guard-time listening window, the effects do not only include unnecessary energy consumption, but also the deprivation of reliable and real-time performance. Therefore, the general idea of the existing algorithm is to add the STSs for retransmission. Two kinds of slot assignments are prevalent, where [15] puts all the STSs behind the GTSs as shown

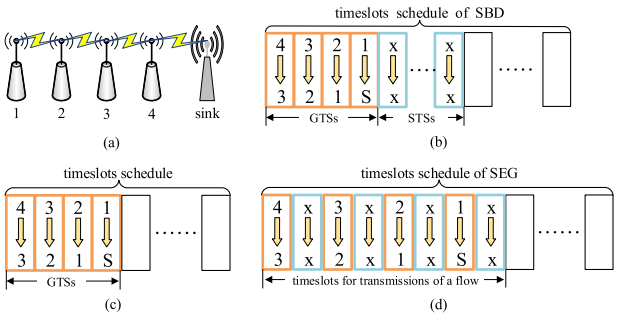

FIGURE 1. The cases of slots assignment.

in Fig. 1(b) and [28] separates a superframe into several parts according to the hop or level of each node as shown in Fig. 1(d). In this paper, the above two types of assignment are referred to as SBD (Shared slots Back of Dedicated slots) and SEG (Segment) slots scheduling respectively. The problem with the former assignment is that one of the failed transmissions results in wasting GTSs later in the flow. While another allocation can solve the above problem, it needs more STSs to guarantee reliability, which results in increasing the length of the superframe and extra energy consumption.

Finally, the resource restrictions make the above challenges more difficult. Most applications have to finish the end-to-end transmission within a certain time interval, because the data of some devices and the command to actuators possess strict timeliness in industrial production. Therefore, there is an end-to-end deadline in each scheduled flow. The minimum deadline not only limits the size of the network but also the length of the superframe, since slot is the basic time unit in TDMA. A slot is set to over 10 ms in WirelessHART and IEEE 802.15.4e low latency deterministic network (LLDN). For example, some devices need to send collected information to the manager with several different sampling periods. One of the shortest periods with minimal deadline is 500 ms. Each device carries out a half-duplex communication in a single channel, thus the total number of slots is restricted to 50 for one channel. Moreover, some high-level devices have the responsibility to help their child nodes relay the packets, which additionally occupy more slots. Moreover, it is necessary to allocate a sufficient number of STSs and management slots to improve reliability and maintain the network. The topology with 10 monitoring devices needs at least 20 GTSs. The remaining 30 slots are enough to set to be shared and management slots. However, there is not a sufficient number of slots with a 500-ms deadline when up to 15 devices are added. With these items in mind, scheduling a suitable slot allocation should be carefully considered when designing a monitoring and control system.

III. SYSTEM MODEL

A. SYSTEM DESCRIPTION

In the practical application, we develop a monitoring and control system as shown in Fig. 2. All welder machines (WM) are

Algorithm 1 Flow-Based Data Slots Schedule

Initiate L_Γ, $uP \longleftarrow 0$ and $dP \longleftarrow L\Gamma$
Input: A new v_i with F_i, p_i, h_i, t_i which has less h_i.
Output: $\mathcal{S}$ for each $v_i(j) \in V_{Fi}$
for each F_i **do**
  $so \longleftarrow \lfloor p_i/p_{min} \rfloor$, $(pi \geq p_{min})$;
  if t_i is upstream flow **then**
    for uP **from** 0 **to** L_Γ **do**
      if $s_{uP,\ldots,s_{uP}+hi+NSi} = \emptyset$ **then**
        for each $v_i(j) \in V_{Fi}$ **from** $v_i(0)$ **to** $v_i(h_i\text{-}1)$ **do**
          Allocate CTS s_{uP} to $\mathcal{S}$ $v_i(j)$ and $\mathcal{S}$ $v_i(j+1)$;
          Allocate CTS s_{uP} to Γ;
          $uP \longleftarrow uP + 1$;
        Allocate STSs $s_{uP,\ldots,s_{uP}+NSi}$ to all $\mathcal{S}$ $v_i(j)$;
        Allocate STSs $s_{uP,\ldots,s_{uP}+NSi}$ to Γ;
        $uP \longleftarrow uP + NS_i$;
        break;
  else if t_i is downstream flow **then**
    for dP **from** L_Γ **to** 0 **do**
      if $s_{dP} - hi - NSi, \ldots, s_{dP} = \emptyset$ **then**
        Allocate STSs $s_{dP}-NSi, \ldots, s_{dP}$ to each $\mathcal{S}v_i(j)$;
        Allocate STSs $s_{dP} - NSi, \ldots, s_{dP}$ to Γ;
        $dP \longleftarrow dP\text{–}NS_i$;
        for each $v_i(j) \in V_{Fi}$ **from** v_i (0) **to** $v_i(h_i\text{-}1)$ **do**
          Allocate CTS s_{dP} to $\mathcal{S}$ $v_i(j+1)$ and $\mathcal{S}$ $v_i(j)$;
          Allocate CTS s_{dP} to Γ;
        $dP \longleftarrow dP\text{–}1$;
        break;
  if $uP \geq dP$ **then**
    return FAIL;

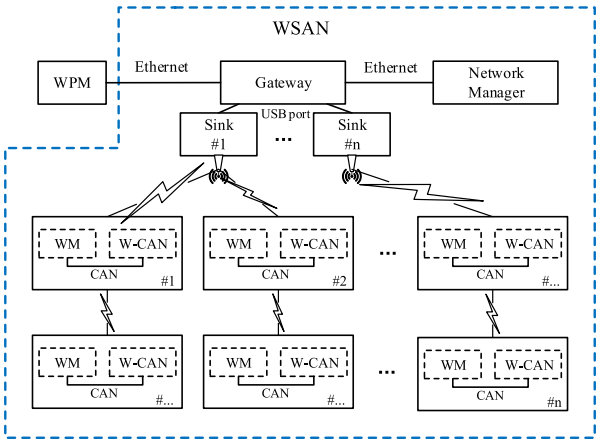

FIGURE 2. The structure of the system.

monitored and controlled by the welding parameter manager (WPM). The WSAN system consists of four types of devices, i.e. a W-CAN, sinks, NetworkManager and a gateway. W-CAN is a wireless communication and processing device which is responsible for the data conversion between a WM and wireless protocol. It also plays the role of router to form a multi-hop mesh network. The sink constructs the connection

between the gateway and the W-CAN node, where it communicates with the gateway via the USB port. Multiple sinks are employed in the system to make full use of the multiple channels, in addition, a sink is responsible for a subgraph with a channel. NetworkManager is a centralized controller responsible for managing the whole network, such as routing and scheduling. The W-CAN devices periodically collect information from the WMs to report to NetworkManager, while occasionally receiving commands from NetworkManager.

B. NETWORK MODEL

In our real-time slots schedule, we consider a directed graph $G = (V, E)$ with N fixed nodes denoted by field devices $v_i \in V$ $(i = 0, 1, \ldots, N)$ and gateway g. The presence of a link (v_i, v_j) $(i \neq j)$ denoted by $l_{i,j}$ in the graph means that node v_i transmits packets to node v_j. For common industrial monitoring and control applications, the tree topology is in the majority. The periodic end-to-end communication between a source and destination is divided into upstream delivery and downstream delivery, where upstream delivery is from the sensor or actuator to the gateway while the downstream delivery is from the gateway to the actuators.

Each of these end-to-end deliveries is considered a flow $F_i \in F$, where $F = \{F_0, F_1, \ldots, FN_F\}$. The nodes in flow F_i are denoted by $VF_i = \{v_i\ (0), v_i(1), \ldots, v_i(h_i)\}$, where h_i refers to the number of hops. For upstream, $v_i\ (0)$ represents the sensor or actuator and $v_i(h_i) = g$, while $v_i(0) = g$ and $v_i(h_i)$ represents the actuator in the downstream flow. Every flow has a period p_i and a deadline d_i, where $d_i \leq p_i$. A collection of consecutive slots constitutes a superframe denoted by Γ, which repeats in a fixed time interval. The length of the superframe is denoted by L_Γ (i.e. the number of slots) and depends on the size of the network and sampling rate of the field devices. We adopt a superframe offset (so) to schedule the device, the sampling rate of which is multiple L_Γ. For example, when the period of the device is twice as long as the minimum period, we set $so = 2$. In this paper, the W-CAN node should send the information to the manager and then receive the command from the manager within a superframe.

All field devices are time synchronized with a limited time error of ± 100 μs in a network. Each device has a source clock which is possibly the gateway or the high-level node. The approach of synchronization is according to the timing information carried by the data packet or acknowledgement which belongs to the clock-source node. Imperfect time synchronization requires receivers to listen for a while before the slot starts, and then stop listening (after the guard time). In this paper, due to the modification of the slot, each node needs to be synchronized in time. The frequency offset of a crystal is λ with a fairly wide temperature range. As the function of the time t, the representation of the local clock $C_v(t)$ in a sensor node v is $C_v(t) = at + b$, where a is the clock drift, which characterizes the skew rate or frequency of the clock, and b is the offset of node v's clock, which represents the difference to time t. For the sake of simplicity, only b is adjusted when synchronizing. If the local clock of source node $C_s(t)$ is given, the time deviation is $\Delta t = C_s(t) - C_v(t)$. The period of synchronization is T and the maximum tolerable time deviation is δ, so we have $T < \delta/\lambda$.

IV. THE PROPOSED SCHEME

In this paper, we combine the flow-based slot allocation algorithm with the proposed concession slot to guarantee reliability and improve the efficiency of slot utilization. The continuous slots for a flow and the concession slot not only avoid the waste of slot utilization, but also avoid the contention at the STSs. Therefore, the proposed schedule is named flow-based slot yielding schedule (FSYS) in this paper. Meanwhile, the proposed slot allocation is according to the order of node hop, where it also can decrease the complexity of network formation. On the one hand, all nodes demonstrate an unpredictable behavior in practical applications, such as shutting down to save energy or restarting when not in use. The slot allocation of new nodes may frequently occurred. On the other hand, the slot allocation algorithm is executed on the NetworkManager in a centralized network, the results of allocation need to be propagated to corresponding nodes. However, if the slot allocation is according a certain attribute, the position of all previous slots may be changed in the superframe after scheduling. Thus, it is better that the slot allocation is not scheduled according to the order of certain attributes found among all the nodes, but rather according to the order of hop. On the basis of nodes hop, the information of slot allocation only needs to be propagated to the nodes which are related to the route of node. We assume that all the flows have the same priority in this paper.

A. SLOT SCHEDULING ALGORITHM

For better integration with the concession slot, the proposed slot scheduling is based on the flow, including a group of consecutive CTSs for each flow. For flexibility, when all nodes joined the network, the flow-based scheduling algorithm is executed by NetworkManager, aiming at each subgraph individually. After generating the slot allocation table, the NetworkManager sends a reply to the node along the flow routing path. Thus, all related nodes (from $v_i\ (0)$ to $v_i(h_i - 1)$) in the routing path need to parse the reply packet and update their own slots tables. The process of information propagation adapts to our flow-based slot allocation and only need a flow transmission, which can decrease the complexity of network updates. When a node leaves the network for some unpredictable reasons, the NetworkManager also needs to send a disconnecting response packet along the routing path and the related nodes release the unused slots.

We assume that each subgraph occupies a channel offset, which employs the channel hopping approach. The actual channel is calculated by (absolute slot number + channel offset) *mod* the number of available channels. As shown in Algorithm 1, the node which has less hop is scheduled first and its data flows are assigned to corresponding slots for transmission and reception. Like the source node, it has its

own attributes e.g. flow F_i, period p_i, the number of hop h_i and the type of flow t_i. The additional schedule table S will be generated to each node for each flow after executing the algorithm, including the superframe offset so, slot offset and slot type, with the exception of when there is not enough slots to be allocated. The upstream flow consists of the source node v_i (0), a set of relaying nodes and the destination of the sink, while the downstream flow consists of the exchange between source and destination. We denote uP and dP as the pointers to mark the current assigning slot offset of the upstream part and downstream part respectively, since the slot allocation of the upstream flow is in sequence and the downstream flow is the opposite. Thus, s_{uP} represents the first empty slot, which can be allocated to the transmission of upstream flow, and s_{dP} is the last empty slot, where its previous slots can be allocated to the transmissions of downstream flow. Before allocating slots, the superframe offset of such a flow must be determined. The superframe offset is the active frequency of a slot among the superframes. First we need to find the minimum period p_{min} among the flows and let p_{min} be the length of the superframe. All the periods of flows depend on the requirement of application, thus we calculate the p_{min} before the start of network formation. Therefore, other flows with various sampling periods calculate their superframe offsets by dividing p_{min}. Next, behind the consecutive CTSs of flow F_i, NS_i STSs are set to be used for retransmission. Each number of STSs is related to the length of corresponding flow and can be calculated as $NS_i = \delta \rangle h_i$, where δ is recommended on the range of (0, 1].

B. FUNCTION OF CONCESSION SLOT

One of the vital contributions in this proposal is that the STSs are not necessary to be contended by several competitors. In this paper, major improvements are the timer settings and the operation in CTS compared with the acknowledged transmission slot in IEEE 802.15.4e. As shown in Fig. 3, the standard defines timeslot diagram and corresponding operations to schedule the device to complete half-duplex communication. At the beginning of a sender's slot, the node prepares the frame for transmission and performs CCA in *CCA* during *TsTxOffset*. It transmits the frame if the channel is free, if not, it returns to sleep. Correspondingly, the receiver enters listening mode after *TsRxOffset* for the duration of *TxRxWait*. If it detects the frame during the guardtime, it receives and validates the frame. The acknowledgement (ACK) should be replied if the frame is valid. The ACK delay time (*TsAckWait*) includes the switching time from the receiving mode to transmitting mode. Meanwhile, the sender switches to receiving mode from transmission mode and waits for the arrival of an ACK frame during the guardtime. Remaining unscheduled devices are in sleep mode to avoid collision and save energy. The proposed CTS does not change the length of the slot, which is set to be 10 ms. The operation from the beginning of the frame transmission or reception remains unchanged. Instead, we adjust the timers and the operations before transmitting and receiving the frame respectively.

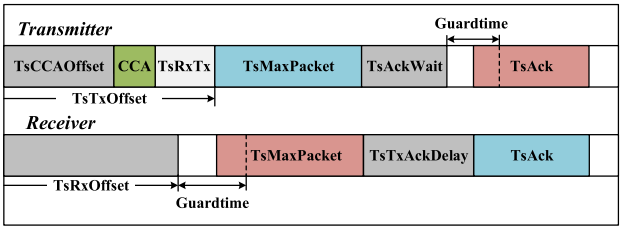

FIGURE 3. The general slot structure.

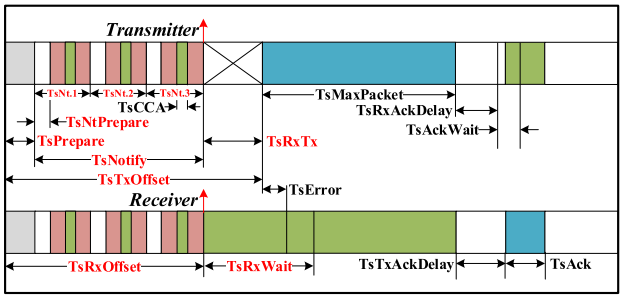

FIGURE 4. The CTS structure with modification in our system.

TABLE 1. Timers and specifications of CTS.

Symbol	Timer(μs)	Description
TsPrepare	200	Time for preparation of CCA or pre-signal
TsTxOffset	2620±100	Time between beginning of slot and start of frame transmission
TsRxOffset	2120±100	Start of the slot to when transceiver must be listening
TsRxWait	1560±100	Time to wait for start of frame
guardtime	1000	Time for guarding the idle listening
TsMaxPacket	4256	Maximum time of data frame
TsTxAckDelay	1000 ±100	End of frame to start of ACK
TsRxAckDelay	900±100	End of message to listen ACK
TsAckWait	400 ±100	Minimum time to wait for ACK
TsAck	832	Time for ACK frame
TsCCA	128	Minimum CCA detection time
TsRxTx	500	Switching time from receive to transmit
TsNtPrepare	100	Time for preparation of pre-signal
TsNotify	1920±100	Operation time for deciding transceiver state
TsNt.x(*x=1, 2, 3*)	640	Time for one operation of CCA or pre-signal

The main purpose of the modification is to notify the node in the currently active slot, to compromise with the previous-hop node which failed to transmit or receive in its active slot.

The designed concession slot is shown in Fig. 4 and the corresponding descriptions and timer values are listed in Table 1, where the main additional timers and operations are in the part of *TsTxOffset* and *TsRxOffset*. Considering that the length of a slot is limited within 10 ms, we add three notifying units (*TsNt.x*) to send the pre-signal or perform CCA with a 100-μs time interval for preparation, where x

can be selected as 1, 2, or 3. The three notifying units can well support four-hop networks and it is enough to satisfy most of industrial applications. In the meantime, *TsCCA* is set in the middle of each *TsNt.x* to detect the corresponding signal of the previous-hop node, since it reserves a 206-μs time interval to avoid interlaced detection. To decide the operations in *TsNotify*, we reserve 200 μs on *TsPrepare* for necessary assessment. The reason for lengthening the time interval of *TsRxTx* to 500 μs is that nodes need to determine their final radio statement in the current slot according to the previous operations and assessments. Therefore, preparations such as loading packet and setting channel are operated in *TsRxTx*, which is executed on *TsCCAOffset*. According to the link schedule algorithm, each node has its link scheduling. However, the link schedules may be inflexible and unable to adapt to failures caused by the harsh and complicated environment. Therefore, the proposed CTS temporarily changes the node state according to the actual running circumstances.

The active slot is the slot pre-allocated to the node, the type of which is transmitting (*Tx*) or receiving (*Rx*). Therefore, s_l represents the slot offset of next active slot and the node will update it after the active slots, which are *Rx* type originally. However, the update of s_l suspends when the node encounters failed transmitting in a *Tx* type slot and it will be updated at the last slot of current flow. Finally, only one of the three slot types, including sleep (*Slp*), can be determined to each node in one CTS after a series of operations and determinations, which correspond to the mode of the radio transceiver. There are three types of operations in each *TsNt.x*, which are Idle, CCA and Signal. The operation of Idle means the node does nothing in corresponding *TsNt.x*; CCA means the node performs CCA in corresponding *TsNt.x* and Signal means the node sends pre-signal in corresponding *TsNt.x*.

The node, scheduled in the current flow, remains in the original state *Slp* when $s_c < s_l$, where s_c represents the current slot. The node is needed to check the statement of F_c (the mark of the current flow). The *SUCCEED* statement refers to the successful transmission of the packet of the current flow, and the radio of the node has remained in *Slp*. Otherwise, the presence of the packet belonging to the current flow (p_{Fi}) needs to be assessed by the flags *EXIST* or *NO_EXIST*. If the packet is in the node's buffer, the node should send the packet without any operations in the whole *TsNotify*. Thus no matter what the original state is, the radio state will be changed into *Tx*. Then, if the type of current slot is STS, the node will change the radio into *Rx*. If it is not the STS, the node continues to compare s_c with s_l and $s_l + 1$. If $s_c = s_l$, the original radio state s_t is *Tx* or *Rx*. When $s_t = Tx$, it means that the node has not received the packet at previous slot and has no packet to relay. Thus, the node signals in *TsNt.1* to notify its receiving node to sleep and performs CCA in *TsNt.2* to check the state of channel. It will change its state to *Rx* if the channel is idle, otherwise it will sleep. When $s_t = Rx$, the node wants to know whether the previous-hop node has a packet to send. It performs CCA in *TsNt.1*, since the previous-hop node signals in *TsNt.1* if it has no packet

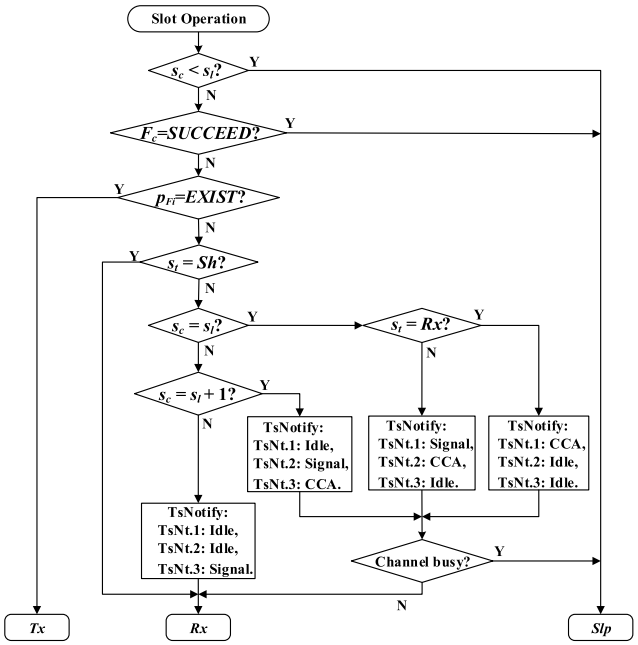

FIGURE 5. Node operation within a concession slot period.

to send. Thus, the node changes to *Slp* if the channel is busy. Among *TsNotify*, less than one CCA operation and one Signal operation acts for avoiding fault detection. Thus the results of the channel assessment through the operation which contains CCA can decide the state of the radio transceiver. When the slots of the flow are exhausted in the current superframe, it resets F_c to *READY* regardless of whether or not the flow succeeds.

The STSs need to be allocated for retransmission behind the CTSs. When the STSs are coming, the node will sleep if the current flow has been finished. Otherwise, the state of node is changed to *Tx* if it still has packet in its buffer, or the state is changed to *Rx* if it has not finished the current flow and has no packet either. The STSs in the proposed slot scheduling are not competed by other nodes because of the flow-based sequence of the slot allocation. The integration of scheduling and operation can not only improve the reliability but also the efficiency of slot utilization.

C. EXAMPLE

To illustrate the scheduling and operation, we consider the example depicted in Fig. 1. As the four nodes joined the H_N^k network, the flow 4→3→2→1→ sink is formatted and the corresponding GTSs are allocated by the NetworkManager. As shown in Fig. 6, each node has already confirmed its original radio state at each slot. For instance, from slot 1 to slot 6, the original states of Node 4 are *Tx*, *Slp*, *Slp*, *Slp*, *Sh* and *Sh* respectively. Here, we assume that each of links 4→3 and 3→2 will occur a failed transmission once and the remaining transmissions are successful. After a failed transmission at slot 1, Node 4 stops updating $s_l = 1$ and needs to retransmit its packet. When the slot 2 is coming,

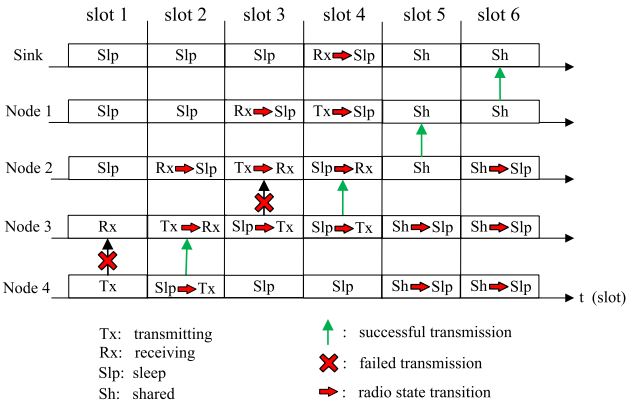

FIGURE 6. Example of node operation during a flow transmission.

Node 4 has $F_c = READY$, $p_{Fi} = EXIST$; Node 3 has $s_l = 2$, $F_c = READY$, $p_{Fi} = NO\_EXIST$ and $s_t = Tx$; Node 2 has $s_l = 2$, $F_c = READY$, $p_{Fi} = NO\_EXIST$ and $s_t = Rx$. Thus, Node 4 does nothing at *TsNotify*; Node 3 signals at *TsNt.1* and performs CCA at *TsNt.2*; Node 2 only performs CCA at *TsNt.1*. According to the proposed node operation in Fig. 5, Node 4, 3, 2 change their original radio states into *Tx*, *Rx* and *Slp* at slot 2, respectively. Similarly, after the successful transmission at slot 2, the Node 1, 2, 3 still needs to change their original states to corresponding states at slots 3 and 4, though a failed transmission occurs at slot 3. In this example, two STSs are used, where $NS_i = 2$. At the two STSs, we can see that Node 3 and 4 both change the *Sh* state into *Slp* to decrease the energy consumption and Node 1 and 2 continue to finish the flow transmissions without contention.

V. PERFORMANCE ANALYSIS

This section presents an analysis of successful transmission probability based on the flow of the two aforementioned schedules, i.e. SBD and SEG, and the proposed method FSYS. To integrate the real-time performance and reliability, we define $Pa(F)$ as a metric, which is the average successful transmission probability of all the flows by direct transmission or retransmission in the current superframe. In other words, the overdue packets are considered as lost packets. Table 2 lists the given notations for the evaluation. The corresponding analytical models are built to describe the schedules. To guarantee a fair comparison, the length of the superframe and the total number of data timeslots are the same in all three schedules. The main difference is the sequence of slots allocation and the function of new type of slot.

First, we use the Discrete-Time Markov Chain (DTMC) to analyze the delivery performance of SBD and SEG based on the methods of [19] and [29]. We denote $\mathbf{S}$ as the collection of all possible states that one of the nodes encounters while suffering consecutive STSs with other competitors. The S is indicated as $\mathbf{S} = \{T_0, T_1, \ldots, T_M, Su, Fa, B_{1,1}, \ldots, B_{M,BW-1}\}$. We assume that there are N nodes competing for a group of consecutive STSs. The mechanism of competition is slotted

TABLE 2. Symbols used in the analysis.

Symbol	Description
T_α	the states of transmission
$B_{\alpha,\beta}$	the states of backoff
Su	the state of successful transmission
Fa	the state of failed transmission
BW	the window size of random backoff
M	the maximum number of transmission retries
L_{worst}	the number of longest steps that the state moves from T_0 to Fa
L_{max}	the quantity of all the possible state within L_{worst} STSs
N	the number of nodes in participation
$\mathbf{D}_N$	the transition matrix generated by N nodes
H_N^k	the probability of successfully contending a slot with fixed N devices and within k STSs
σ	the average probability that the node successfully transmits its packet within NS STSs and the number of the competitor is less than N
p	average packet error rate in a slot transmission
NS	the number of STSs
N_F	the number of flows
h_i	the number of hops in F_i
$Pa(F)$	the average probability that all the flows successfully transmit or retransmit within the current superframe

random backoff following the TSCH-CA. During one of the STSs s_k, the number of nodes in state x is indicated as $Xs_k(x)$, where $x \in \mathbf{S}$. Specifically, $Xs_0 = \{Xs_0(T_0) = N, Xs_0(x) = 0, x \in \mathbf{S}, x \neq T_0\}$ represents that there are N nodes in the T_0 state during the s_0. Suffering every STS, several possible Xs_{k+1} could be generated according to state Xs_k. Let n indicate the number of nodes attempting transmissions in s_k simultaneously, where $n = \sum_{\alpha=0}^{M} X_{s_k}(T_\alpha)$. If $n < 2$, collision is impossible, and the possible states in s_{k+1} are

$$\begin{cases} X_{s_{k+1}}(Su) = X_{s_k}(Su) + n \\ X_{s_{k+1}}(Fa) = X_{s_k}(Fa) \\ X_{s_{k+1}}(T_\alpha) = X_{s_k}(B_{\alpha,1}) 1 \le \alpha \le M \\ X_{s_{k+1}}(B_{\alpha,\beta}) = X_{s_k}(B_{\alpha,\beta+1}) 1 \le \alpha \le M, \quad 1 \le \beta \le BW - 2 \\ X_{s_{k+1}}(B_{\alpha,BW-1}) = 0 1 \le \alpha \le M \end{cases}$$

$n \geq 2$ represents that all nodes suffer the collision. We use R_α to indicate the number of nodes that have failed the attempt of T_α transmission in s_k, where the R_α can be obtained by

$$R_\alpha = \begin{cases} 0 & n < 2 \\ X_{s_k}(T_\alpha) & n \geq 2 \end{cases} \quad (1)$$

Therefore, in the states of R_α, failed nodes are changed into $T_{\alpha+1}$ or $B_{\alpha+1,\beta}$, $\alpha \in [0, M\text{-}1]$, $\beta \in [1, BW\text{-}1]$ because of the random backoff mechanism as shown in Fig. 7. However, the nodes lose their opportunity for retransmission and change their state into Fa in s_{k+1} when having failed on T_M during s_k. Each of the $\alpha + 1$ states could be changed by a random quantity of nodes among these R_α nodes. We use q_α to indicate one of the quantity compositions, which is defined as $q_\alpha = \{q_\alpha(T_{\alpha+1}), q_\alpha(B_{\alpha+1,1}), \ldots, q_\alpha(B_{\alpha+1,BW-1})\}$, where

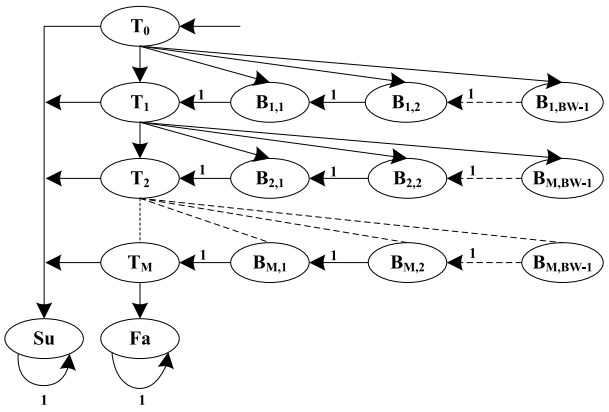

FIGURE 7. The transition states of node transmission.

$q_\alpha(x)$ refers to the number of nodes that are randomly changed into the x state. Obviously, q is one of the possible weak compositions of integer R_α into BW parts and there is $q_\alpha(T_{\alpha+1}) + q_\alpha(B_{\alpha+1,1}) + \ldots + q_\alpha(B_{\alpha+1,BW-1}) = R_\alpha$. From this, we can deduce the following possible states in s_{k+1} if $n > 2$:

$$\begin{cases} X_{s_{k+1}}(Su) = X_{s_k}(Su) + n \\ X_{s_{k+1}}(Fa) = X_{s_k}(Fa) + X_{s_k}(T_M) \\ X_{s_{k+1}}(T_\alpha) = X_{s_k}(B_{\alpha,1}) + q_{\alpha-1}(T_\alpha) \quad 1 \le \alpha \le M \\ X_{s_{k+1}}(B_{\alpha,\beta}) = X_{s_k}(B_{\alpha,\beta+1}) + q_{\alpha-1}(B_{\alpha,\beta}) \quad 1 \le \alpha \le M, \\ \qquad\qquad 1 \le \beta \le BW - 2 \\ X_{s_{k+1}}(B_{\alpha,BW-1}) = q_{\alpha-1}(B_{\alpha,BW-1}) \quad 1 \le \alpha \le M \end{cases}$$

Each of the R_α failed attempts can generate $\binom{R_\alpha + BW - 1}{BW - 1}$ weak compositions of q_α into BW parts, where $\alpha \in [0, M\text{-}1]$. Thus, the probability that one of possible weak compositions is generated by

$$P(q_\alpha) = \left(\frac{R_\alpha!}{q(T_{\alpha+1})! \cdot q(B_{\alpha+1,1})! \cdot \ldots \cdot q(B_{\alpha+1,BW-1})!} \right) \cdot \left(\frac{1}{BW} \right)^{R_\alpha} \quad (2)$$

The collection of all possible weak compositions generated by R_α failed attempts is indicated as $Q_\alpha(R_\alpha)$, where $q_\alpha \in Q_\alpha(R_\alpha)$. Each of the possible compositions q_α can constitute a stochastic system transition $Q_g = \{q_0, q_1, \ldots, q_{M-1}\}$. Thus we can obtain all possible transitions by

$$\mathbf{\Lambda}_{Q_g} = \bar{\times}_{\alpha=0}^{M-1} Q_\alpha(R_\alpha) \quad (3)$$

where $\bar{\times}$ is the Cartesian product. For each possible $Q_g x$ in Λ_{Qg} at s_k STS, we can deduce all the possible system states Xs_{k+1} and the probability of generating each possible Xs_{k+1} is

$$P_{Q_g} = \prod_{\alpha=0}^{M-1} P(q_\alpha) \quad (4)$$

As shown in the figure, the step length of the worst case, where the state moves from T_0 to Fa, is

$$L_{worst} = M \cdot BW + 1 \quad (5)$$

Therefore, we can traverse all possible states $\{D_0, D_1, \ldots, D_\tau, \ldots, D_{Lmax}\}$ generated by N nodes according to each possible system state Xs_k and obtain the final transition matrix $\mathbf{D}_N$ according to (2), (3) and (4), where L_{max} is the quantity of all possible states within L_{worst} STSs. For instance, Xs_0 corresponds to D_0, X_1 corresponds to possible states $\{D_1, D_2, \ldots, D_\tau\}$ and so on. With no loss of generality, we assume that $\boldsymbol{\pi} s_0 = [1, 0, \ldots, 0]$ is the vector of the initial state probability. Thus, after passing k STSs, the probability can be calculated as $\boldsymbol{\pi} s_k = \boldsymbol{\pi} s_0 \cdot \mathbf{D}_N^k$. Finally, the average probability of successfully contending a slot, where N devices use the TSCH-CA algorithm to transmit a packet within k STSs is calculated by

$$H_N^k = \frac{1}{N} \sum_{\tau=0}^{L_{\max}} D_\tau(Su) \cdot \pi_{s_k}[\tau] \quad (6)$$

Furthermore, we assume that there are N GTSs allocated to N nodes for a flow. The node failing transmission in its GTS will attempt retransmission during NS STSs. The average probability that the node successfully transmits its packet within NS STSs is calculated by

$$\sigma = \sum_{n=1}^{N} \binom{N}{n} \cdot p^n \cdot (1-p)^{N-n} \cdot H_n^{NS} \quad (7)$$

where the number of competitors is less than N. The binomial coefficient $\binom{N}{n}$ represents that there are n nodes failed to successfully transmit the packet on their own GTSs.

A. SBD

For SBD scheduling, all the failed transmissions at GTSs are retried at STSs together by using the slotted random backoff mechanism. Its $Pa(F)$ can be calculated as

$$Pa(F) = \frac{1}{N_F} \sum_{i=0}^{N_F} \left((1-p)^{h_i} + \sum_{j=1}^{h_i} p \cdot (1-p)^{h_i - j} \cdot \sigma^j \right) \quad (8)$$

whereN_F is the number of flows.

B. SEG

As for SEG scheduling, the slot allocation to each flow is divided into the GTS and STS segments by hop or level. The number of slots in each segment is different. This means that there is an average successful transmission probability σ_j during jth slots segment, where the σ_j is calculated as

$$\sigma_j = \sum_{n=1}^{N_j} \binom{N_j}{n} \cdot p^n \cdot (1-p)^{N_j - n} \cdot H_n^{NS_j} \quad (9)$$

where N_j is the number of GTSs in the jth slots segment and NS_j is the number of STSs in the jth slots segment. The binomial coefficient $\binom{N_j}{n}$ represents that there are n nodes failed to successfully transmit the packet on their own GTSs in the jth slots segment. Focusing on the successful transmission of the flow, we need to integrate the successful transmission probability of each hop node in each slot segment. The $Pa(F)$ of SEG scheduling is calculated as

$$Pa(F) = \frac{1}{N_F} \sum_{i=0}^{N_F} \prod_{j=1}^{h_i} ((1-p) + p \cdot \sigma_j) \tag{10}$$

C. FSYS

The STSs are not contended by several nodes in FSYS, since the scheduling is based on the flow and the STSs are assigned after a series of CTSs, which belong to a single flow. In addition, the individual operation within a concession slot makes the slot utilization more flexible. The successful end-to-end transmission only needs h_i slots, no matter if they are CTSs or STSs. It means that h_i successful transmissions can lead ith flow to be successfully transmitted within $h_i + NS_i$ slots. Therefore, we can obtain the $Pa(F)$ of the proposed FSYS by

$$Pa(F) = \frac{1}{N_F} \sum_{i=0}^{N_F} \left((1-p)^{h_i} \cdot \sum_{m=0}^{NS_i} \binom{h_i + m - 1}{m} \cdot p^m \right) \tag{11}$$

where NS_i is the number of STSs allocated to F_i.

VI. RESULTS

To verify the performance improvement of the proposed scheduling, we simulated the analytical results of (8), (10) and (11) by Matlab and carried out practical experiments in a factory to compare the schedules of SBD, SEG and the proposed FSYS. The hardware of the W-CAN node consists of an LPC1769 mote and CTM1050T CAN chip as shown in Fig. 8. A 779 ∼ 787MHz band, configured in the radio transceiver AT86RF212B is chosen over the more common 2.4 GHz band because it offers a greater propagation range and a stronger diffraction ability for the wireless communication. The radio chip AT86RF212B also supports basic wireless operations according to the IEEE802.15.4-2011 standard [30]. We have implemented network stack on the LPC1769 mote, e.g. TDMA-based MAC protocol and source-routing network protocol. In this paper, we implemented our individual node operation by adjusting the timers of TSCH timeslot according to the Table 1 and performing corresponding operations, such as CCA and pre-signal. We have tested that the distance of the transmission is about 160-180 m in the line of sight in a factory. Four available channels can be used in the channel hopping approach. The proposed schedule also can be effective on the 2.4 GHz band when applying on the smaller areas. The power of each node is supplied by two batteries. Fig. 8 also shows that each

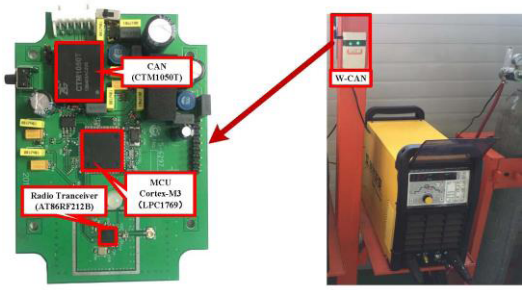

FIGURE 8. WM and the hardware of the W-CAN node.

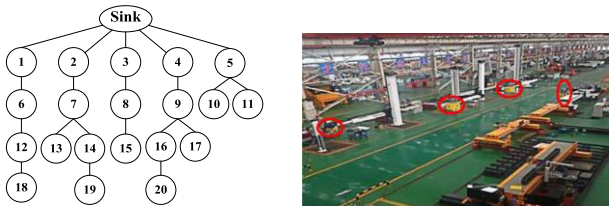

FIGURE 9. The topology of the experiments and the site of implementation in the factory.

TABLE 3. Parameters of experiments.

Parameters	Value
The number of nodes (N)	20
BW (in SBD and SEG)	4
M	3
Maximum number of hop	4
Length of Superframe (L_f)	1 s
Transmission power	11 dBm
Transmission rate	250 kb/s
Size of packet	80 Bytes

welder machine comes equipped with a W-CAN node to collect the information periodically.

There are 20 welder machines with W-CAN nodes randomly deployed in a 200 × 100 m^2 factory as shown in Fig. 9(b). For a fair comparison, the nodes are set to be static and form a multi-hop tree topology, as seen in Fig. 9(a). The W-CAN nodes sample the data from welder machines every second and send it to NetworkManager within a superframe. Considering the sample period and transmission deadline, we set the length of the superframe to 100 slots according to Algorithm 1. Moreover, our proposed slot operation has a limited time error of synchronization which is ±100 μs. The crystals we used are capable of maintaining a frequency of ±10 ppm over a fairly wide temperature range. The period of synchronization is set to be 1 s in our proposed scheduling less than 10 s in the limited period of synchronization. Each round of the experiment lasts for an hour and the remaining experimental parameters are listed in Table 3.

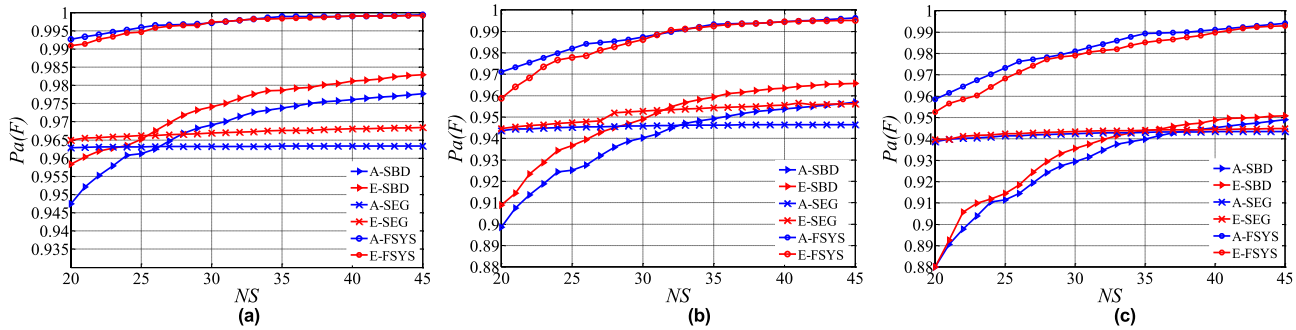

FIGURE 10. The analytical and experimental results of transmission reliability using three slot scheduling with different interference. (a) $p = 0.05$. (b) $p = 0.1$. (c) $p = 0.12$.

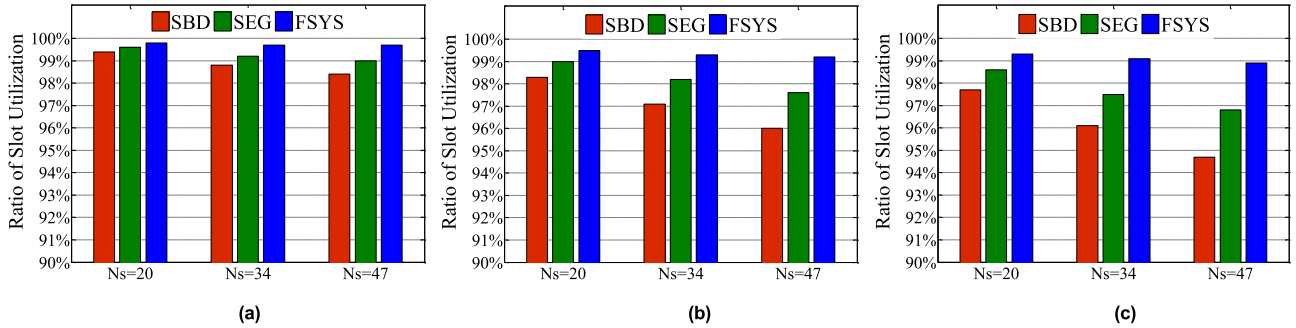

FIGURE 11. Comparison of the slot utilization with different interference in the factory experiments. (a) $p = 0.05$. (b) $p = 0.1$. (c) $p = 0.12$.

The experiments are conducted to compare transmission reliability, slot utilization and energy consumption, all of which are under different levels of interference including a 5%, 10% and 12% packet error rate respectively. They are corresponding to $p = 0.05$, $p = 0.1$, $p = 0.12$ in the analytical results. The level of interference is adapted by adding extra interference nodes around each device. The packet error rate is adjusted by changing the frequency and power of the radio transceiver on the interference nodes. The analytical and experimental results of transmission reliability are shown in Fig. 10, where A and E represent the results of analysis and experiment respectively. The three figures roughly validate the reliability improvement of the proposed scheduling compared with two other scheduling designs. In general, the curve tendency of the experimental results coincides with the analytical results. These figures enable us not only to demonstrate the performance of the research but also enable us to validate the research in our analytical method. As can be seen, the $Pa(F)$ is affected by the number of STSs, where NS remains the same in the each round of comparison between the three scheduling designs. The analytical $Pa(F)$ is a bit lower than the experimental results in the SBD and SEG scheduling designs. This is because one of the transmissions may be successful if the contention occurred in practice. However, the results of the analysis are in general in accordance with the results of real tests using FSYS, because the stochastic factor of contention is avoided by using the approach of slot operation. Figs. 10(a), 10(b) and 10(c), show that the $Pa(F)$ is 0.991, 0.959 and 0.953 using the least STSs respectively, under the 5%, 10% and 12% packet error rate. However, the $Pa(F)$ of SBD results only reach 0.958, 0.909 and 0.88 using 20 STSs with a 5%, 10% and 12% packet error rate respectively. A slight improvement is performed in the SEG scheduling compared with SBD, where the $Pa(F)$ is 0.965, 0.945 and 0.94 under the same condition. As the number of STSs increases, the reliability of all the flows using FSYS is close to 1 under different interferences, which is much higher than the other two scheduling designs.

Although the transmission reliability can be improved by adding STSs for retransmission, the efficiency of slot utilization is reduced consequentially. We selected node 1 as the observation object to count the slot utilization using debugging mode in real work settings. The effective slots allocated to a node represent that the node successfully transmits or receives a packet, or that it sleeps in the slot, with the exception of failed transmission and idle listening in STSs. The purpose of comparing slot working utilization with different interference was to show the efficiency of active slots. The results of node 1 using three scheduling designs are shown in Fig. 11. Using 20 STSs, the slot utilization gap between the three scheduling designs is not big; they are all over 97.5%, also under a 12% packet error rate. However, when the total number of STSs increases to 45, FSYS improves the slot working efficiency by 4.2% and 2.1% more

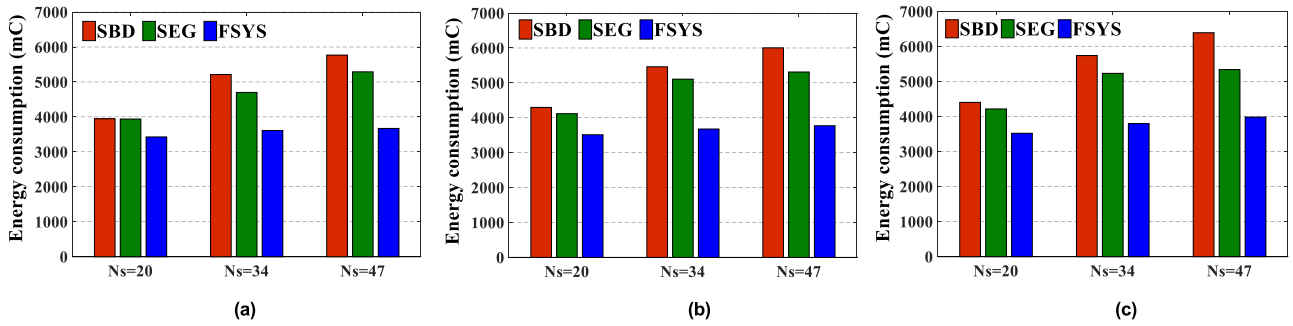

FIGURE 12. Comparison of the energy consumption with different interference in the factory experiments. (a) $p = 0.05$. (b) $p = 0.1$. (c) $p = 0.12$.

than the SBD and SEG scheduling under a 12% packet error rate. This is due to the fact that the STSs in SBD and SEG are not fully employed by the random backoff mechanism.

Moreover, the improvement of slot utilization also results in a reduction of energy consumption. Thus we also measured the energy consumption of node 2 in the factory experiments as shown in Fig. 12. We use a 2.7V and 120F supercapacitor to be the power supply of node. The supercapacitor is fully charged before experiments and we measure the energy consumption by measuring the voltage of supercapacitor after experiments. The results of slot utilization and energy consumption are only counted and measured on one node for our limited capability. The results roughly indicate that the proposed scheduling saves a significant amount of energy, especially in cases where there is more interference. The total number of data timeslots is same among the three scheduling, however, the utilization of STS for each node is different. The STSs in FSYS are only used for the nodes which needs to retransmission in a flow. In SBD and SEG, the nodes, which has not successfully transmitted the current packet, need to use a group of STSs for random back-off mechanism. Thus, these nodes cost the extra energy consumption. We can deduce that FSYS can significantly contribute to slot efficiency and reduction of energy consumption for the whole network.

VII. CONCLUSION

In this paper, we have proposed an efficient retransmission scheme over WSANs, including a slot scheduling algorithm and the concession timeslot. The slot scheduling is based on allocating consecutive GTSs and CTSs to a flow. The function of CTS can be integration with the slot scheduling well and make the retransmission more reliable and flexible. We have also analyzed the average end-to-end transmission reliability of the scheduling approaches of SBD and SEG by DTMC to be able to compare it with the proposed scheduling FSYS. The experiments have been carried out in a real factory. The obtained analysis and experimental results show that the proposed methods significantly improve the transmission reliability and the efficiency of slot utilization. The results also prove that the scheduling method is more flexible and able to reduce the energy consumption.

REFERENCES

[1] M. Raza, N. Aslam, H. Le-Minh, S. Hussain, Y. Cao, and N. M. Khan, "A critical analysis of research potential, challenges, and future directives in industrial wireless sensor networks," *IEEE Commun. Surveys Tuts.*, vol. 20, no. 1, pp. 39–95, 1st Quart., 2018.

[2] X. Li, D. Li, J. Wan, A. V. Vasilakos, C.-F. Lai, and S. Wang, "A review of industrial wireless networks in the context of industry 4.0," *Wireless Netw.*, vol. 23, no. 1, pp. 23–41, Jan. 2017.

[3] *Factories of the Future. Multi-Annual Roadmap for the Contractual PPP Under Horizon 2020*, Eur. Commission, Brussels, Belgium, 2015, p. 54.

[4] A. A. Kumar, K. Ovsthus, and L. M. Kristensen, "An industrial perspective on wireless sensor networks—A survey of requirements, protocols, and challenges," *IEEE Commun. Surveys Tuts.*, vol. 16, no. 3, pp. 1391–1412, 3rd Quart., 2014.

[5] C. Lu *et al.*, "Real-time wireless sensor-actuator networks for industrial cyber-physical systems," *Proc. IEEE*, vol. 104, no. 5, pp. 1013–1024, May 2016.

[6] F. Dobslaw, T. Zhang, and M. Gidlund, "QoS-aware cross-layer configuration for industrial wireless sensor networks," *IEEE Trans. Ind. Informat.*, vol. 12, no. 5, pp. 1679–1691, Oct. 2015.

[7] C. Xia, J. Xi, L. Kong, and Z. Peng, "Bounding the demand of mixed-criticality industrial wireless sensor networks," *IEEE Access*, vol. 5, pp. 7505–7516, 2017.

[8] N. Nasser, L. Karim, and T. Taleb, "Dynamic multilevel priority packet scheduling scheme for wireless sensor network," *IEEE Trans. Wireless Commun.*, vol. 12, no. 4, pp. 1448–1459, Apr. 2013.

[9] T. Rahman, H. Ning, H. Ping, and Z. Mahmood, "DPCA: Data prioritization and capacity assignment in wireless sensor networks," *IEEE Access*, vol. 5, pp. 14991–15000, 2016.

[10] A. Saifullah, Y. Xu, C. Lu, and Y. Chen, "End-to-end communication delay analysis in industrial wireless networks," *IEEE Trans. Comput.*, vol. 64, no. 5, pp. 1361–1374, May 2015.

[11] J. Long, M. Dong, K. Ota, A. Liu, and S. Hai, "Reliability guaranteed efficient data gathering in wireless sensor networks," *IEEE Access*, vol. 3, pp. 430–444, May 2015.

[12] F. Dobslaw, M. Gidlund, and T. Zhang, "Challenges for the use of data aggregation in industrial wireless sensor networks," in *Proc. IEEE Int. Conf. Autom. Sci. Eng.*, Aug. 2015, pp. 138–144.

[13] M. Dong, K. Ota, and A. Liu, "RMER: Reliable and energy-efficient data collection for large-scale wireless sensor networks," *IEEE Internet Things J.*, vol. 3, no. 4, pp. 511–519, 2016.

[14] *IEEE Standard for Local and Metropolitan Area Networks—Part 15.4: Low-Rate Wireless Personal Area Networks (LR-WPANs) Amendment 1: MAC Sublayer*, IEEE Standard 802.15.4e-2012, 2012, pp. 1–225.

[15] S. Han, X. Zhu, A. K. Mok, D. Chen, and M. Nixon, "Reliable and real-time communication in industrial wireless mesh networks," in *Proc. IEEE Real-Time Embedded Technol. Appl. Symp.*, Apr. 2011, pp. 3–12.

[16] M. Yan *et al.*, "Hypergraph-based data link layer scheduling for reliable packet delivery in wireless sensing and control networks with end-to-end delay constraints," *Inf. Sci.*, vol. 278, no. 4, pp. 34–55, 2014.

[17] O. T. Valle, A. V. Milack, C. Montez, P. Portugal, and F. Vasques, "Experimental evaluation of multiple retransmission schemes in IEEE 802.15.4 wireless sensor networks," in *Proc. 9th IEEE WFCS*, vol. 12, May 2012, pp. 201–210.

[18] Z. Tao, M. Gidlund, and J. Åkerberg, "WirArb: A new MAC protocol for time critical industrial wireless sensor network applications," *IEEE Sensors J.*, vol. 16, no. 7, pp. 2127–2139, Apr. 2016.
[19] C. Ouanteur, D. Aissani, L. Bouallouche-Medjkoune, M. Yazid, and H. Castel-Talev, "Modeling and performance evaluation of the IEEE 802.15.4e LLDN mechanism designed for industrial applications in WSNs," *Wireless Netw.*, vol. 23, no. 5, pp. 1343–1358, 2016.
[20] M. Hashimoto, N. Wakamiya, M. Murata, Y. Kawamoto, and K. Fukui, "End-to-end reliability- and delay-aware scheduling with slot sharing for wireless sensor networks," in *Proc. Int. Conf. Commun. Syst. Netw. IEEE*, Jan. 2016, pp. 1–8.
[21] S. Montero, J. Gozalvez, and M. Sepulcre, "Link scheduling scheme with shared links and virtual tokens for industrial wireless sensor networks," *Mobile Netw. Appl.*, vol. 22, no. 6, pp. 1083–1099, Dec. 2017.
[22] S. Duquennoy, B. Al Nahas, O. Landsiedel, and T. Watteyne, "Orchestra: Robust mesh networks through autonomously scheduled TSCH," in *Proc. ACM Conf. Embedded Netw. Sensor Syst. ACM*, 2015, pp. 337–350.
[23] Y. Jin, P. Kulkarni, J. Wilcox, and M. Sooriyabandara, "A centralized scheduling algorithm for IEEE 802.15.4e TSCH based industrial low power wireless networks," in *Proc. IEEE Wireless Commun. Netw. Conf.*, Apr. 2016, pp. 1–6.
[24] M. Ojo and S. Giordano, "An efficient centralized scheduling algorithm in IEEE 802.15.4e TSCH networks," in *Proc. IEEE Conf. Standards Communi. Netw. (CSCN)*, Oct./Nov. 2016, pp. 1–6.
[25] W. Shen, T. Zhang, F. Barac, and M. Gidlund, "PriorityMAC: A priority-enhanced MAC protocol for critical traffic in industrial wireless sensor and actuator networks," *IEEE Trans. Ind. Informat.*, vol. 10, no. 1, pp. 824–835, Feb. 2014.
[26] C. J. M. Liang, K. Chen, N. B. Priyantha, J. Liu, and F. Zhao, "RushNet: Practical traffic prioritization for saturated wireless sensor networks," in *Proc. ACM Conf. Embedded Netw. Sensor Syst.*, 2014, pp. 105–118.
[27] M. Sha, D. Gunatilaka, C. Wu, and C. Lu, "Implementation and experimentation of industrial wireless sensor-actuator network protocols," in *Proc. Eur. Conf. Wireless Sensor Netw.*, Porto, Portugal, 2015, pp. 234–241.
[28] D. Yang *et al.*, "Assignment of segmented slots enabling reliable real-time transmission in industrial wireless sensor networks," *IEEE Trans. Ind. Electron.*, vol. 62, no. 6, pp. 3966–3977, Jun. 2015.
[29] S. Chen *et al.*, "Performance analysis of IEEE 802.15.4e time slotted channel hopping for low-rate wireless networks," *KSII Trans. Internet Inf. Syst.*, vol. 7, no. 1, pp. 1–21, 2013.
[30] *Low-Rate Wireless Personal Area Networks (WPANs)*, IEEE Standard 802.15.4–2011, 2011.

JIAN MA received the B.S. degree in communication engineering from Beijing Jiaotong University, Beijing, China, in 2013, where he is currently pursuing the Ph.D. degree in communications and information science. His specific areas of research interest mainly focus on low-power network technologies and industrial wireless sensor networks.

DONG YANG (M'11) received the B.S. degree from Central South University, Hunan, China, in 2003, and the Ph.D. degree in communications and information science from Beijing Jiaotong University, Beijing, China, in 2009. From 2009 to 2010, he was a Post-Doctoral Research Associate with Jonkoping University, Jonkoping, Sweden. In 2010, he joined the School of Electronic and Information Engineering, Beijing Jiaotong University, where he is currently a Professor. His research interests are network technologies, including Internet architecture, industrial Internet, and wireless sensor networks.

HONGCHAO WANG (M'14) received the B.S. degree in communication engineering and the Ph.D. degree in communication and information systems from Beijing Jiaotong University, Beijing, China, in 2005 and 2012, respectively. He is currently with the School of Electronic and Information Engineering, Beijing Jiaotong University. His research interests include Internet architecture, network security, and wireless sensor networks.

MIKAEL GIDLUND (M'98–SM'16) received the M.Sc. and Ph.D. degrees in electrical engineering from Mid Sweden University, Sundsvall, Sweden, in 2000 and 2005, respectively. Since 2014, he has been a Full Professor of computer engineering with Mid Sweden University. In 2005, he was a Visiting Researcher with the Department of Informatics, University of Bergen, Bergen, Norway. From 2006 to 2007, he was a Research Engineer and a Project Manager, responsible for wireless broadband communication at Acreo AB, Kista, Sweden. From 2007 to 2008, he was a Senior Specialist and a Project Manager with responsibility for next-generation IP-based radio solutions at Nera Networks AS, Bergen. From 2008 to 2013, he was a Senior Principal Scientist and a Global Research Area Coordinator of wireless technologies in ABB Corporate Research, with the main responsibility to drive technology and strategy plans, standardization, and innovation in the wireless automation area. He has pioneered the area of industrial wireless sensor network. He holds over 20 patents (granted and pending applications) in the area of wireless communications. He has authored or co-authored over 100 scientific publications in refereed journals. His research interests include wireless communication and networks, wireless sensor networks, access protocols, and security.

He was a recipient of the Best Paper Award at the IEEE International Conference on Industrial IT in 2014. He is currently an Associate Editor of the IEEE Transactions on Industrial Informatics and the Vice-Chair of the IEEE IES Technical Committee on Cloud and Wireless Systems for Industrial Applications.