

Research Article

Det-WiFi: A Multihop TDMA MAC Implementation for Industrial Deterministic Applications Based on Commodity 802.11 Hardware

Yujun Cheng, Dong Yang, and Huachun Zhou

School of Electronic and Information Engineering, Beijing Jiaotong University, Beijing, China

Correspondence should be addressed to Dong Yang; dyang@bjtu.edu.cn

Received 13 December 2016; Revised 4 March 2017; Accepted 28 March 2017; Published 16 April 2017

Academic Editor: Feng Wang

Copyright © 2017 Yujun Cheng et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Wireless control system for industrial automation has been gaining increasing popularity in recent years thanks to their ease of deployment and the low cost of their components. However, traditional low sample rate industrial wireless sensor networks cannot support high-speed application, while high-speed IEEE 802.11 networks are not designed for real-time application and not able to provide deterministic feature. Thus, in this paper, we propose Det-WiFi, a real-time TDMA MAC implementation for high-speed multihop industrial application. It is able to support high-speed applications and provide deterministic network features since it combines the advantages of high-speed IEEE802.11 physical layer and a software Time Division Multiple Access (TDMA) based MAC layer. We implement Det-WiFi on commercial off-the-shelf hardware and compare the deterministic performance between 802.11s and Det-WiFi under the real industrial environment, which is full of field devices and industrial equipment. We changed the hop number and the packet payload size in each experiment, and all of the results show that Det-WiFi has better deterministic performance.

1. Introduction

In recent years, wireless communication has gained significant importance in industrial automation. A great amount of industrial applications adopts wireless network control systems [1–5] thanks to their ease of deployment and the low cost of their components compared with the wired control systems. Most of industrial applications require assured maximum end-to-end delivery latency and reliable transmission, namely, the deterministic feature of the system. Besides, the sampling rate of different applications is varied and some applications require the network to provide extremely high transfer speed. Thus, both of the transmission rate and deterministic feature of the network should be considered when designing a control system.

Several standards are dedicated for manufacturing automation and process automation, such as WirelessHART [6], ISA100.11a [7], and WIA-PA [8]. They are based on the IEEE 802.15.4-2006 [9] standard and have already been

applied. Their main characteristic is the use of the TDMA based MAC protocol to enable more reliable and real-time communication. However, the transfer rate defined in IEEE 802.15.4 is up to 250 kbps, which cannot provide high enough sampling rate for high-speed application. On the other hand, IEEE 802.11 [10] is able to support high-speed communication (up to 150 Mbps in 802.11n). Besides, the IEEE 802.11s [11] amendment, which aims to provide support for multihop networks including both static topologies and ad hoc networks, seems hopeful to be applied in multihop industrial application. However, it still adopts carrier sense multiple access with collision avoidance (CSMA/CA) based MAC layer, which does not provide any time delay guarantee on data delivery and is not feasible for most of industrial automation application.

To address this problem, a huge amount of literature has been proposed. The proposed schemes can be classified into two groups. The first group aims to optimize the conventional 802.11 MAC layer [12–15]. Some default

parameters in 802.11 MAC, such as contention window and backoff mechanism, are optimized to meet the industrial application demand. However, the uncertain delay brought by CSMA/CA still cannot be avoided. The second group substitutes the TDMA MAC layer for the conventional 802.11 MAC. Without the CSMA/CA mechanism, a better deterministic performance can be achieved. This paper belongs to this group, and, thus, this group of solutions will be reviewed below.

Many TDMA based MAC protocols have been previously proposed based on commodity 802.11 hardware. SoftMAC [16] initiates an implementation to create a precise control over the wireless transmission and reception. Based on Atheros MadWifi driver, the author proposes several properties of commodity 802.11 hardware that should be disabled to make it a flexible platform. Soft-TDMAC [17] is a multihop TDMA protocol which is said to realize microsecond synchronization. The real-time performance of Soft-TDMAC is great in testbed result due to the centralized network structure and TDMA based MAC layer. However, it is not designed for control applications. RT-WiFi [18] aims to provide high sampling rate and deterministic timing guarantee on packet delivery in wireless control systems. It is also based on a TDMA data link layer combined with IEEE 802.11 physical layer. But it can only deploy in completely connected networks; that is, it cannot support any kind of multihop application.

In this paper, we provide a detailed presentation of the Det-WiFi, which is able to provide support for high-speed multihop industrial deterministic application. It adopts a centralized network architecture and uses time division multiplexing access strategy to provide end-to-end delay guarantee of data. The main contributions of this paper can be summarized as follows:

- (1) We designed Det-WiFi, a new MAC protocol for high-speed multihop industrial deterministic application. The detailed MAC features are considered to make sure the system can be used in practice. Compared with the existed protocols, Det-WiFi is capable of enabling real-time and high-speed communication simultaneously.
- (2) We implemented Det-WiFi based on commodity 802.11 hardware, which is ubiquitous and relatively cheap. With several driver modifications, Det-WiFi is both easy and affordable to apply to an industrial control system.
- (3) We set up a testbed under the real industrial environment to validate the performance of Det-WiFi. We tested and analyzed both Det-WiFi and 802.11s protocol in detail. All of the results show that Det-WiFi has better deterministic performance compared with 802.11s.

The rest of the paper is organized as follows. Section 2 presents the system design of Det-WiFi. Section 3 describes the Det-WiFi implementation details. Section 4 explains the testbed configuration and experiment results. Section 5 concludes the paper.

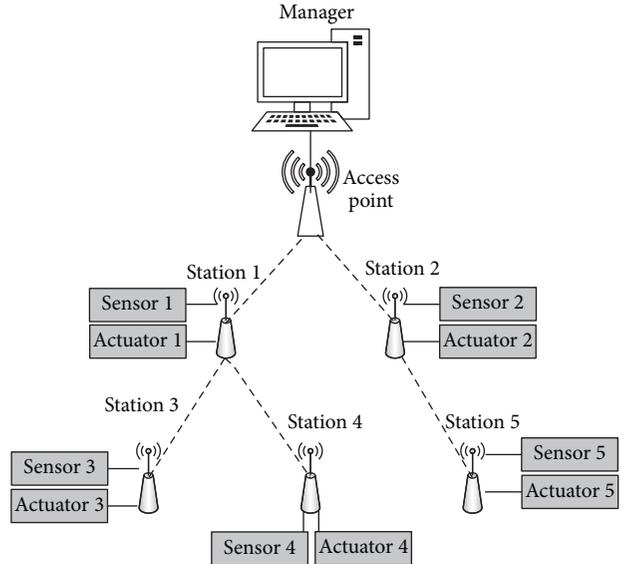


FIGURE 1: Det-WiFi network structure.

2. Det-WiFi Protocol Design

Det-WiFi is a wireless multihop protocol, which aims to provide high-speed, real-time, and reliable communication for a wide variety of industrial real-time applications. In this section, we present the design details of Det-WiFi protocol.

Figure 1 shows a typical Det-WiFi network structure for industrial real-time control system. The network structure is composed of four parts: manager, access point (AP), stations, and actuators and sensors. The manager is the brain of the network, and it is located at the top of the structure. When a station joins the network, manager receives information from the station, including station address, parent station address, and receive signal strength indicator (RSSI). Thus, the manager holds a lot of information of the stations and has the capability to control the entire network using the information. The AP is a bridge between the manager and stations. It is responsible for delivering all of the uplink or downlink messages to manager or stations. The AP and the stations form a multihop network; each station is attached with one actuator and one sensor. Stations control the actuators according to the management information from AP and monitor the data from sensors. Since the sensors generate a big amount of data and almost all of the traffic in the network requires deterministic guarantee, it is challenging to design a scheme to fulfill such stringent requirements, especially for a multihop network solution.

To address this problem, Det-WiFi adopts TDMA scheme instead of distributed coordination function (DCF) in regular Wi-Fi to ensure the reliability and real-time capability of the network. DCF is a mechanism based on carrier sense multiple access with collision avoidance (CSMA/CA). It is a general media access method, but it does not seem practicable for stringent real-time communication due to the undetermined time delay introduced by channel contention and backoff mechanism. Hence, DCF is unable to provide end-to-end

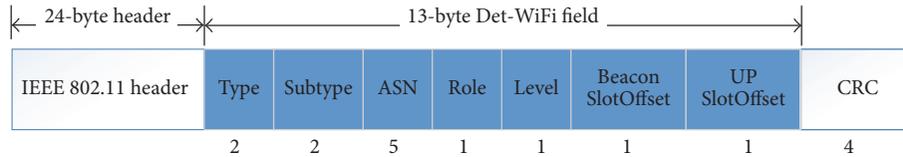


FIGURE 2: The beacon frame format.

delay guarantee and is not appropriate for deterministic communication. In TDMA scheme, in contrast, communication resource is divided into time-slots and assigned to stations. Due to its contention-free and predictable time delay, it is feasible for deterministic communication. Moreover, Det-WiFi employs centralized network architecture and manager is responsible for controlling the entire network, such as time-slot allocation and scheduling, sending routing information to stations. With the control of the manager and the TDMA scheme, Det-WiFi is feasible for the industrial real-time applications.

2.1. Network Joining Process. The network joining process of Det-WiFi consists of two steps: AP joining process and stationing joining process.

Det-WiFi begins to work when the manager boots up. At the start, if AP detects the manager online, it will prepare for the network joining process. It sends the *JOIN_REQUEST* frame to the manager, which contains join information such as its MAC address and hop number. These pieces of information is stored locally by the manager after it receives the *JOIN_REQUEST* frame from AP. After the join information is completely stored, a frame called *JOIN_REPLY* is sent back to AP from manager. *JOIN_REPLY* frame contains control information and resource allocation information. The most important information is time-slot allocation information, including beacon time-slot offset and uplink time-slot offset. Manager utilizes the values of time-slot offsets to allocate beacon and uplink time-slot to AP, respectively. According to these pieces of information, AP knows which time-slot it should occupy and adjusts itself to the appropriate transmission state. At last, AP sends a frame back called *JOIN_ACK* to make the manager confirm its joining status. If the manager receives *JOIN_ACK* frame correctly, AP is considered in joining state, and the three-way handshake joining process is completed.

After the AP joins the network, the stations should prepare for joining process. At this time, AP is going to broadcast the beacon frames (Figure 2) periodically, which contains ASN (absolute slot number), beacon slot offset, role, level, and up slot offset. ASN is used to record the global slot number; level is the maximum child stations that AP can control; role shows the role of the beacon sender (AP or station); beacon slot offset shows which beacon slot the AP occupies; and up slot offset shows which up slot is allocated to AP. If any station hears the beacon, it is going to choose AP as its neighbor and parent according to the beacon frame. Actually, one station may hear several beacons from different neighbors, so it will select one as its parent. After parent selection is finished, it could send *JOIN_REQUEST* frame

to the manager through the AP for joining the network. The joining procedure of station is similar to the procedure of AP; the only difference is that it sends frames to the manager through the AP rather than by itself directly. It is worth mentioning that AP plays the role of bridge between manager and stations in station joining process. It just forwards the joining frames as well as the time-slot allocation information. AP does not generate packets or do any scheduling stuff. Once this station joins the network, it broadcasts its own beacon frames periodically. Other stations which hear the beacons follow similar steps to join the network until all the stations join the network. Since then, the deployment of Det-WiFi is completed and it is ready to collect sensor data and control the actuators.

2.2. Frame and Time-Slot Design. The basic frame structure is comprised of an 802.11 header and the Det-WiFi field. The IEEE 802.11 header is used to control basic 802.11 network features, and the function of Det-WiFi network is controlled by Det-WiFi field actually. There are three main frame types in Det-WiFi, including beacon frame, data frame, and management frame. The frames are distinguished by type and subtype field.

To avoid collision, data transmission procedure in TDMA is divided into time-slots. Only one station can access the channel at the same time, and each station only accesses the channel in the time-slot which is allocated for it. In Det-WiFi, time-slot allocation for one station is completed when the station receives the *JOIN_REPLY* from manager (forwarded by AP) in joining procedure (Section 2.1), and the time-slot information is recorded locally in the time-slot table of the station. Because our goal is to design and implement a basic architecture of a deterministic wireless network, a detailed discussion of the various methods of allocating time-slot and scheduling time-slot are beyond the scope of this paper.

In Det-WiFi, A superframe is formed by an infinite cycling sequence of all the allocated time-slots. Time-slots are circulated as superframes go by. Generally, there are three main types of time-slots in a superframe: beacon slot, transmitting (Tx) slot, and receiving (Rx) slot. AP and stations broadcast beacons in beacon slot, and data is transmitted and received in transmitting and receiving slot, respectively. The structure of the three main types of time-slots are shown in Figure 3.

Beacon slot is used to broadcast beacon frames; thus no acknowledgement (ACK) is required; other frames are sent in the Tx slot and need ACK to confirm whether the frame transmits to the target successfully; after a station has received a data frame in Rx slot, it should send an ACK back to confirm its receiving state. Because of the difference among the three

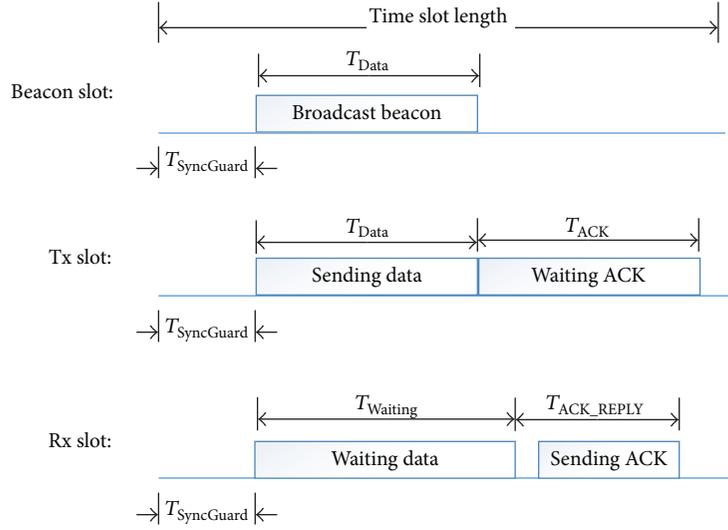


FIGURE 3: Structure of three main slot types.

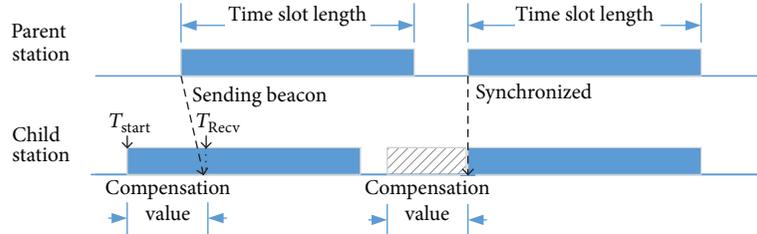


FIGURE 4: Time-slot synchronization procedure.

types of slots, the designs of these time-slots are not the same. The slot sizes of three types of slots can be simply expressed (corresponding to Figure 3) as follows:

$$T_{\text{BeaconSize}} \geq T_{\text{SyncGuard}} + T_{\text{Data}}, \quad (1)$$

$$T_{\text{TxSize}} \geq T_{\text{SyncGuard}} + T_{\text{Data}} + T_{\text{ACK}}, \quad (2)$$

$$T_{\text{RxSize}} \geq T_{\text{SyncGuard}} + T_{\text{Waiting}} + T_{\text{ACK_REP}}, \quad (3)$$

where $T_{\text{SyncGuard}}$ is an interval to make the station be able to tolerate slightly synchronization inaccuracy. T_{Data} is the delay in data transmission. It is caused by both Linux kernel delay and radio transmission delay. T_{ACK} is the ACK transmission time, and it is similar to T_{Data} . T_{Waiting} is the time in which receiver side waits for the frame incoming. The measured values of these parameters are shown in Section 3.3.

2.3. Synchronization. Accurate synchronization is the foundation of TDMA mechanism. In Det-WiFi network time system, the local clock of AP is used as the network reference clock. Every station should keep synchronization directly or indirectly with the AP. The process of synchronization is comprised of two parts: ASN synchronization and time-slot synchronization.

As we mentioned in Section 2.1, ASN is used to record the global slot number. Every time-slot passes by, ASN adds

one, and it keeps increasing since the network boots up. It is contained in every beacon frame. When a parent station broadcasts its beacons, the children stations read the ASN information and synchronize its local ASN with the parent station. It is important because synchronized ASN ensures the AP and all the stations are in the same time-slot state. Besides ASN synchronization, time-slot synchronization is also finished when the children stations receive beacons from their parent (Figure 4). When children stations receive beacons, receiving time T_{recv} is recorded. Besides, every start time of time-slot T_{start} is recorded by stations. If transmission delay is ignored, calibration value for children stations is

$$T_{\text{calib}} = T_{\text{recv}} - T_{\text{start}} + T_c, \quad (4)$$

where T_c is the compensation for the Linux kernel delay, and, luckily, the delay is quite stable. In our test, this delay is $10 \mu\text{s} \pm 2$. Thus, when next slot is coming, children stations can synchronize with the parent station using a changed slot size:

$$T_{\text{next}} = T_{\text{timeslot}} + T_{\text{calib}}. \quad (5)$$

After that, children stations should repeat the time-slot procedure synchronization constantly to maintain the synchronization state. Instead of beacon frames, any frame exchanged with parent station, no matter data frame or

beacon frame, can be used for synchronization, based on the same synchronization mechanism.

In order to reduce the impact of synchronization error, in Det-WiFi, synchronization guard time (Section 2.2) is set at the start of one time-slot. The guard time can improve the error tolerance in the synchronization procedure. However, there is another issue called variation accumulation, which means the synchronization variation can accumulate across the multihop network. To address this problem, Det-WiFi limits the hop number. The maximum hop number which Det-WiFi is able to support is:

$$N_{\max} = \frac{T_{\text{SyncGuard}}}{2 \times T_{\text{SyncVar}}}, \quad (6)$$

where N_{\max} is the maximum hop number, $T_{\text{SyncGuard}}$ is the guard time which is set to $100 \mu\text{s}$ in Det-WiFi, and T_{SyncVar} is the synchronization variation which is $2 \mu\text{s}$. Thus, we can figure out that the supported maximum hop number is 25 hops. In practice, the hop number can be set to 10 to avoid any unknown synchronization variation, which is still enough for many industrial applications.

3. Implementation

In this section, we describe the implementation details of Det-WiFi from three parts: system architecture, driver modification, and timers.

3.1. System Architecture. As we mentioned in Section 3, the network structure of Det-WiFi is comprised of manager, AP, stations, and attached sensors and actuators. Sensors and actuators are varied in different industrial applications, and, therefore, we only reserve an interface for them and use a data generator to emulate their work process in testbed experiments. Because we focus on the deterministic feature of Det-WiFi, we only simply implement basic function of the manager. It just stores the information from stations and distributes a fixed time-slot table to stations after stations join. AP and stations adopt the same hardware and network stack, and they just run in different work mode. In the following paragraph, we will focus on the implementation of stations and AP.

The system architecture of station (or AP) is shown in Figure 5. To achieve deterministic performance, some default features of hardware need to be modified. Thus, in our system, we use the AR9287 chipset-based commodity 802.11 b/g/n hardware. It uses ath9k hardware driver module on Linux system, which is open source and easy to modify. On the top of ath9k, mac80211 is a framework that provides standard IEEE802.11 MAC related functionality. We use Ubuntu 14.04 as the operating system, and the kernel version is 3.14.57. Considering the compatibility issues, we just modified the two modules slightly. We will talk about the detailed modification steps in Section 3.2.

Based on the two kernel modules, Det-WiFi is comprised of three components: packet queues, task scheduler, and system state container (SSC). There are two packet queues in Det-WiFi: sending queue (Tx queue) and receiving queue

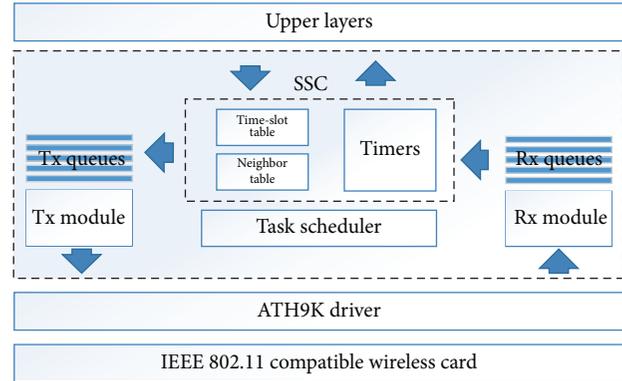


FIGURE 5: Det-WiFi system architecture.

(Rx queue). When packets have been prepared to send, they are put in the sending queue, waiting for the appropriate time-slot, and then are sent to the driver to transmit. Similarly, when packets are received from lower layer, they are stored in the receiving queue. Task scheduler is used to schedule the tasks and control the behavior of the Det-WiFi, including sending beacons, slots cycling, and network joining. These tasks are distinguished by priority level according to the urgency of the tasks, and the task scheduler will execute high-priority task first rather than the low-priority task. SSC consists of time-slot table, neighbor table, and timers: time-slot table records the time-slot cycling sequence, which is acquired from the manager when it joins the network; neighbor table is used to store the neighbors information, which is advertised in the neighbors beacons; timers are responsible for maintaining the time information of Det-WiFi; and most of the tasks are triggered by several timers. The detailed information of timers will be shown in Section 3.3.

Due to the compatible MAC design, the original network upper layers are well-supported. Many applications based on standard UDP/TCP can work without any problem. Moreover, some of real-time multihop applications which cannot be deployed before are able to deploy now because of the deterministic network features.

3.2. Driver Modification. In order to realize multihop deterministic characteristics, some default features of driver should be modified. Luckily, ath9k is an open source driver, and many MAC features of network interface card are able to be changed. We mainly focus on two parts: frame format modification and disabling CSMA mechanism.

It is hard to reuse 802.11 frame format directly due to the topology and structure difference between Det-WiFi and 802.11. Thus, most of the default control and management frames of IEEE 802.11 are abandoned, and only data frame is remained. The control and management information is contained in the payload (Det-WiFi field) of the default 802.11 data frame. However, to send the modified frame properly and successfully, several places of the IEEE 802.11 header need to be changed: the frame should be claimed broadcast, and then the sending process will not be bothered by the default MAC address; the fragment bit is set to zero to tell the receiver

it is not a fragment frame; every source address field is filled with self-defined address instead of the MAC address of NIC. These modifications of frame format bring much convenience to the multihop communication and ensure the frame still can be recognized by the driver.

Several CSMA mechanisms have negative influence on deterministic feature, including virtual carrier sense (NAV) PHY Clear Channel Assessment (CCA) and transmission backoff. Thus, these mechanisms should be disabled. The AR9287 chipset provides a diagnostic register which has several special functions. We set *AR_DIAG_IGNORE_VIRT_CS* and *AR_DIAG_FORCE_RX_CLEAR* flags to disable the NAV and force a CCA, respectively. In order to disable the transmission backoff, the contention windows CW_{min} and CW_{max} are set to zero in the initializing process of driver queues. It is worth mentioning that there are four queues in the driver which are known as *AC_BE*, *AC_BK*, *AC_VI*, and *AC_VO*. These correspond to the four priorities of traffic in enhanced distributed channel access (EDCA) which is proposed in IEEE 802.11e [19] amendment. To ensure all frames transmit in order, only *AC_VO* queue is retained, and we map all frames to that queue. As for the transmission rate, the default minstrel rate control algorithm is disabled, and we adopt a fixed transmission rate 54 Mbps. However, when a fixed transmission rate is used, the driver will send the frame at the lowest rate (1 Mbps in 802.11g). The reason is that the driver will modify the transmission rate automatically when the frame is claimed broadcast or has no ACK policy. After modified, the NIC is able to send frame at the fixed rate of 54 Mbps.

3.3. Timers. In order to realize microsecond precision timing function, timers based on Linux kernel jiffies cannot be employed, as they only provide millisecond granularity. Instead, we adopt high resolution timer (hrtimer) [20, 21] for timing tasks. Hrtimer is able to provide submicrosecond timing precision, and it can fully satisfy the timing demand.

There are several timers based on hrtimers in Det-WiFi in charge of triggering several tasks. The most important timer called mtimer is to keep time-slot generating. When the mtimer is triggered, a new time-slot starts. One Tx slot is comprised of transmission procedure and receiving ACK procedure (discussed in Section 2.2). Thus the transmission delay

$$T_{Data} = T_{RadioDelay} + T_{KernelDelay}, \quad (7)$$

where $T_{RadioDelay}$ is the radio transmission delay and $T_{KernelDelay}$ is the Linux kernel delay. If the data is sent at the rate of 54 Mbps and the frame length is 524 bytes (500 bytes for the payload and 24 bytes for the 802.11 header), the radio transmission delay is

$$T_{RadioDelay} = \frac{(500 + 24) \times 8}{54 \times 10^6} = 77.6 \mu s. \quad (8)$$

However, we measured the total transmission delay from the time when a frame is just handed to driver to the time when the frame is sent out successfully; the average delay is 236 μs . From (7), we can figure out that the average Linux



FIGURE 6: Testbed in real industrial environment.

kernel delay is 158.4 μs . Then we changed the frame length to 224 bytes and 34 bytes, and the average kernel delay is almost unchanged. We measured that the ACK delay is 167 μs in the same way, which is consistent with expected values. If we set the synchronization guard time to 150 μs , we can calculate the time-slot size from (2):

$$T_{TxSize} \geq 150 \mu s + 236 \mu s + 167 \mu s = 553 \mu s. \quad (9)$$

The size of beacon slot and Rx slot is smaller; thus (9) can be used to determine the system slot size and the mtimer timing value.

4. Performance Evaluation

To evaluate the deterministic multihop features of Det-WiFi, we set up a testbed under the real industrial environment (Figure 6), which is full of field devices and industrial equipment. We compare the performance between Det-WiFi and 802.11s in this scenario. Open80211s [22] is an open-source implementation of the ratified IEEE 802.11s wireless mesh standard. Moreover, it is well-supported in the ath9k driver. Therefore, we intend to test the performance of 802.11s based on Open80211s. There are three PCs in our experiment, all of them are equipped Atheros AR9287 IEEE 802.11 compatible NIC. Every device installed a UDP packets generator program, and it generates packets at fixed intervals (5 ms) to emulate the sampling process. The frequency of NIC is set to 2.462 Ghz corresponding channel 11 of IEEE 802.11 b/g/n. We set the time-slot length 600 μs , which is longer than 553 μs to reserve enough time for time-slot internal guard, and the time-slot table is fixed for the experiment.

In our experiment, we focus on the transmission delay of MAC layer and packets loss ratio, which show the deterministic performance of the network. Among the delay metrics, the maximum delay and delay standard deviation are relatively appropriate to reflect the deterministic feature of the network. Besides, the mean delay is also a key metric. To measure the packet loss ratio, we set two counters to record the sent out packets and received packets, respectively. Packet loss is calculated as the subtraction of the number of sent-out packets from the number of received packets. To measure the latency of the network, we start a timer when a frame is

TABLE 1: Comparison of latency and packet loss ratio between Det-WiFi and 802.11s in real industrial environment.

Payload	Scenarios	Max latency (μs)		Mean latency (μs)		Latency standard deviation (μs)		Packet loss ratio	
		Det-WiFi	802.11s	Det-WiFi	802.11s	Det-WiFi	802.11s	Det-WiFi	802.11s
50 bytes	Two-hop	1321	31626	1084	1303	25.08	388.3	0.22%	0%
	Four-hop	2718	16076	2147	2371	55.25	631.1	0.56%	0%
200 bytes	Two-hop	1303	13906	1119	1285	29.87	301.6	0.20%	0%
	Four-hop	2702	53695	2284	2517	40.73	894.5	0.59%	0%
500 bytes	Two-hop	1394	21200	1142	1225	29.13	363.7	0.22%	0%
	Four-hop	2861	27300	2361	2505	50.06	768.9	0.58%	0%

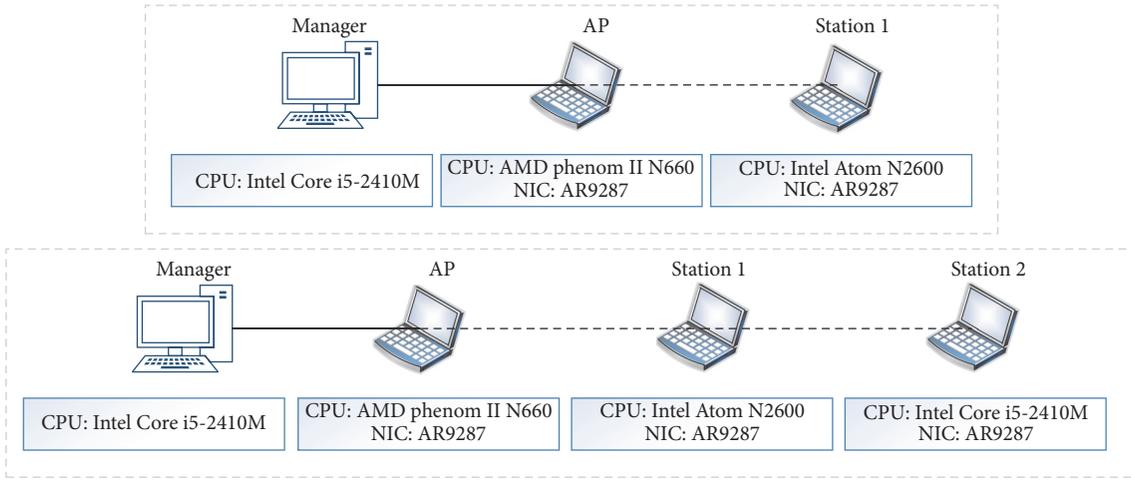


FIGURE 7: Two test topologies' setting.

ready to transmit at the MAC layer. As the frame arrives at the destination station, it sends back the same length frame at once. At last, we measure the time between the frame start time and the response frame arrival time, both at the MAC layer. Based on this measuring method, we compare the latency and packet loss ratio between Det-WiFi and 802.11s in the scenario of two-hop and four-hop networks, respectively. The two test topologies are shown in Figure 7. In each scenario, we transmit three kinds of data packets in different size (50 B, 200 B, and 500 B payload) to emulate different sampling data. Each experiment runs for 10 minutes. Besides, each experiment is conducted for five times to ensure the reliability of test data, and the result is the average of the five.

The test results are shown in Table 1. We compare the mean, max, standard deviation of MAC latency, and packet loss ratio between 802.11s and Det-WiFi. When adopting different packet size, other metrics in both two-hop and four-hop scenarios are almost unchanged except the mean latency. For the Det-WiFi, the mean latency is slightly increasing with the increase of packet size, because the larger packet needs more time to transmit when the transmission rate is fixed. Since the other metrics are not influenced by the payload size, we take the 500 B payload size condition as an example to validate the deterministic performance of Det-WiFi. The latency standard deviation in 802.11s network

is up to 12.5 times to that of Det-WiFi in both scenarios, and the max latency in 802.11s is 15 times and 9.5 times to that of Det-WiFi in two-hop and four-hop experiments, respectively. The high latency standard deviation and max latency in 802.11s network should be attributed to the random backoff mechanism, which makes the time latency uncertain. However, we observe there are 0.22% and 0.58% packet loss in Det-WiFi, respectively. This is due to the slight interference traffic introduced by other 802.11 based devices in the industrial field, and we confirmed it by monitoring the channel using Wireshark. The packet loss is totally acceptable compared with the high latency standard deviation in the 802.11s network. The same conclusion can be drawn from the 50 B and 200 B payload size condition.

We also plot histograms (Figures 8–10) to illustrate the deterministic performance of the 802.11s and Det-WiFi more intuitively. Considering the packet payload size only has small influence on the mean latency, we still take the 500 B payload size condition (Figure 10) for instance. In two-hop scenario (Figure 10(a)), over 95% of packets are delivered in the interval of 1100–1200 μs and almost 100% of packets arrive in 1200 μs in Det-WiFi. On the contrary, the latency in 802.11s distributes separately. Only 86.2% of packets arrive in 1200 μs and 4.7% of packets are delivered over 1400 μs . In the four-hop scenario (Figure 10(b)), the deterministic performance is

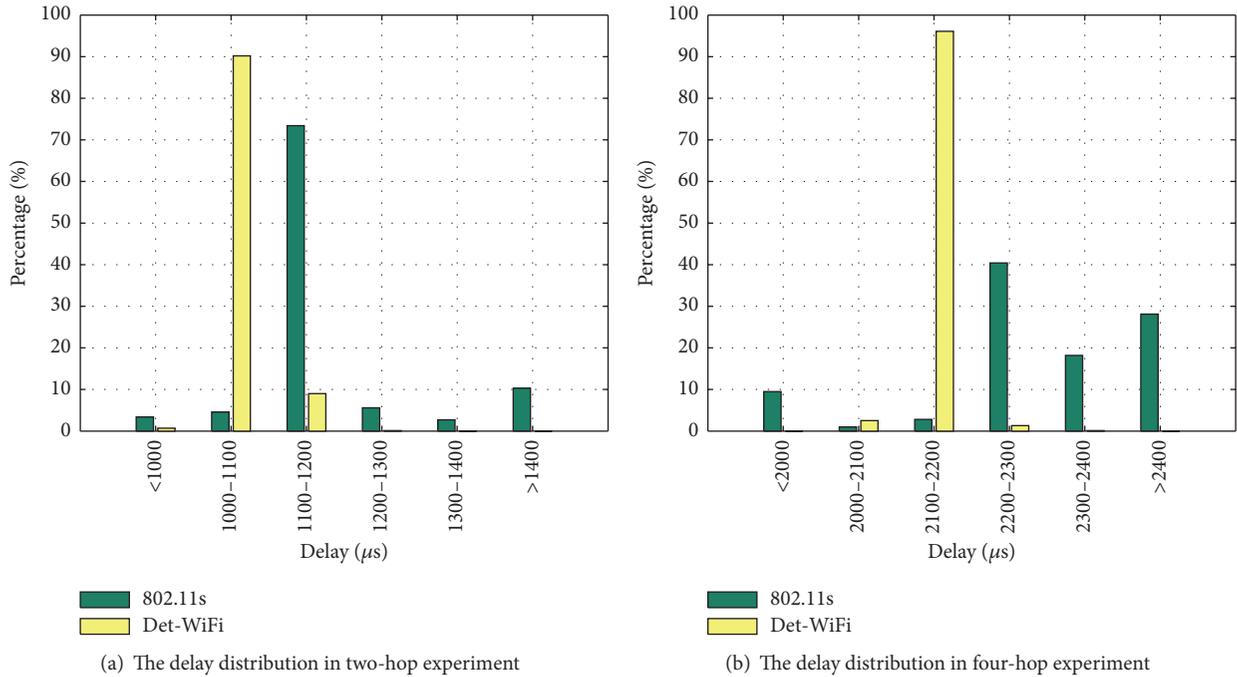


FIGURE 8: The delay distribution in 50 B payload size condition.

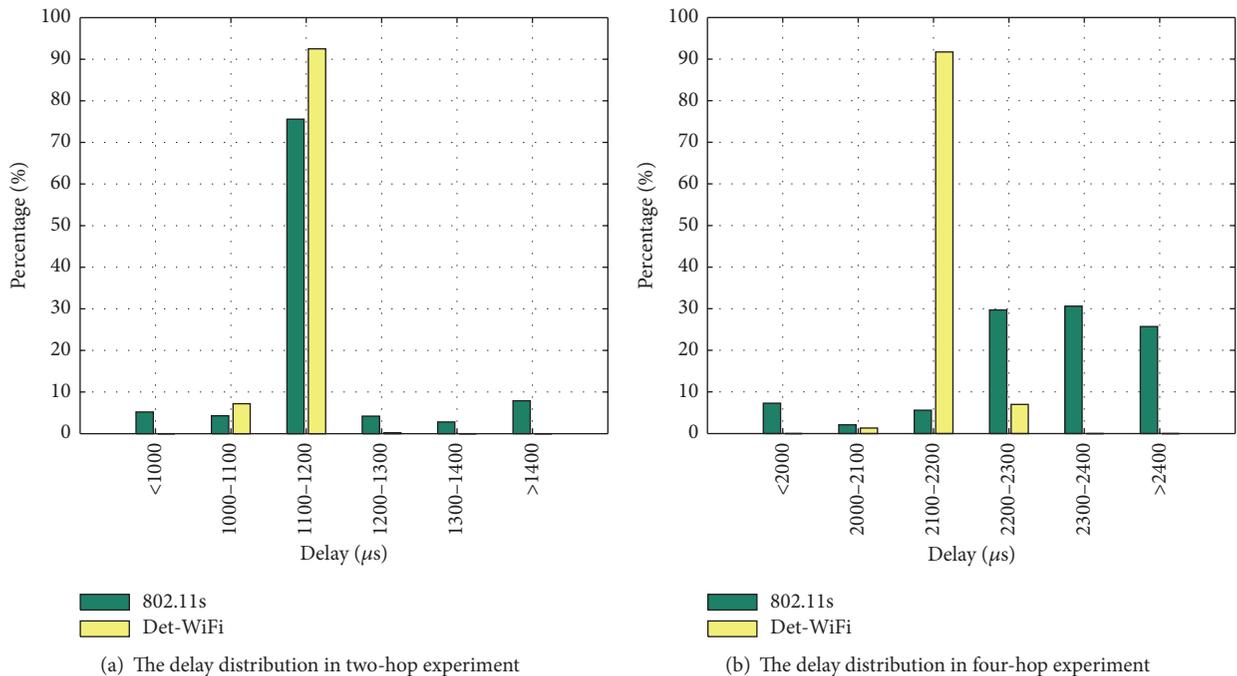


FIGURE 9: The delay distribution in 200 B payload size condition.

even worse in 802.11s network because more hops bring more channel contention and backoff, while the latency of Det-WiFi is still quite stable. In 802.11s network, more than 20% of packets are delivered over 2400 μ s and the latency variance is even greater than the two-hop scenario. In the 50 B and 200 B payload size condition (Figures 8 and 9), the histograms show similar results.

5. Conclusion

In this paper, we propose the Det-WiFi, which aims to provide support for high-speed multihop industrial deterministic demand application. It is based on the physical layer of IEEE 802.11, adopts a centralized network architecture, and uses TDMA strategy instead of CSMA/CA mechanism. Besides,

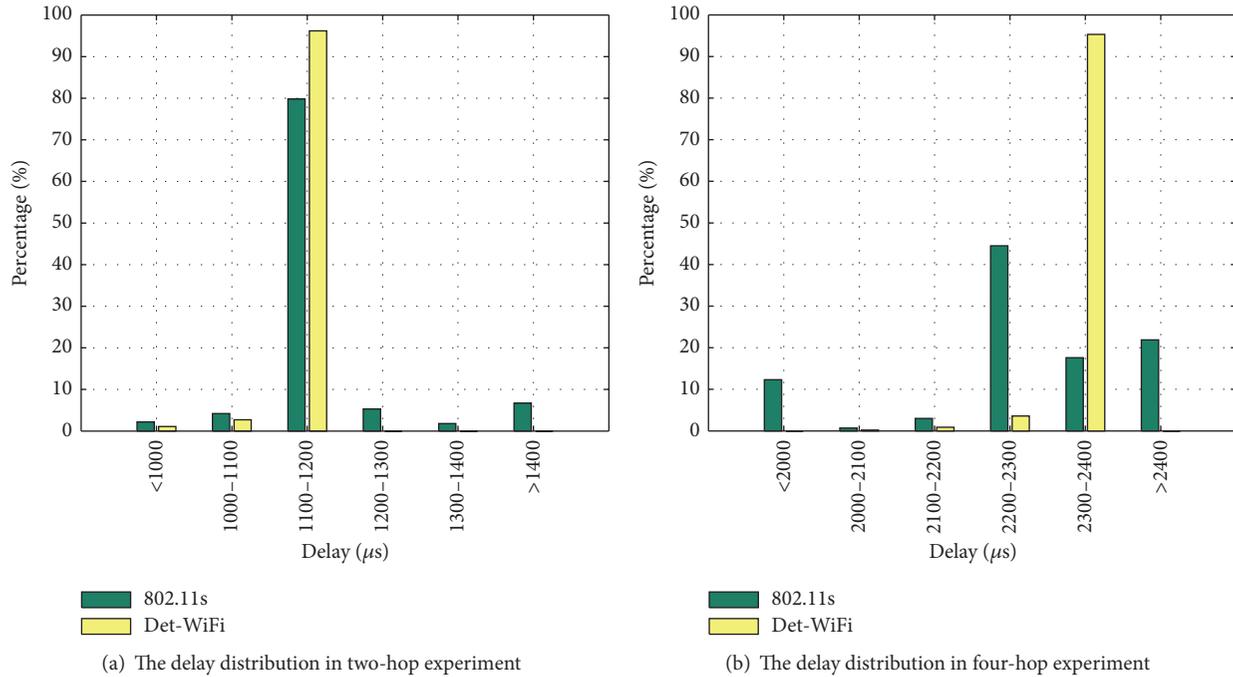


FIGURE 10: The delay distribution in 500 B payload size condition.

Det-WiFi is implemented based on the commercial IEEE 802.11 hardware with a few modifications, which makes it have good compatibility. To validate the performance of Det-WiFi, we set up testbed under the real industrial environment and compared the deterministic performance between 802.11s network and Det-WiFi. The time latency and packet loss ratio are chosen to reflect the deterministic feature of the two. In the 500 B payload size condition, the latency standard deviation in 802.11s network is up to 12.5 times that of Det-WiFi, and the max latency in 802.11s is 15 times and 9.5 times that of Det-WiFi in two-hop and four-hop experiments, respectively. We also repeat the experiment in 50 B and 200 B payload size condition; the test results show that Det-WiFi has better deterministic performance compared with the 802.11s network.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by Electronic Information Technology Innovation and Cultivation (Grant no. Z171100001217004), Fundamental Research Funds for the Central Universities (Grants nos. 2015JBM006 and 2015YJS011), and National 863 Program of China (Grant no. 2015AA016103).

References

- [1] M. Magno, D. Boyle, D. Brunelli, B. O'Flynn, E. Popovici, and L. Benini, "Extended wireless monitoring through intelligent

hybrid energy supply," *IEEE Transactions on Industrial Electronics*, vol. 61, no. 4, pp. 1871–1881, 2014.

- [2] O. Kreibich, J. Neuzil, and R. Smid, "Quality-based multiple-sensor fusion in an industrial wireless sensor network for MCM," *IEEE Transactions on Industrial Electronics*, vol. 61, no. 9, pp. 4903–4911, 2014.
- [3] S. X. Ding, P. Zhang, S. Yin, and E. L. Ding, "An integrated design framework of fault-tolerant wireless networked control systems for industrial automatic control applications," *IEEE Transactions on Industrial Informatics*, vol. 9, no. 1, pp. 462–471, 2013.
- [4] M. Chen, "Reconfiguration of sustainable thermoelectric generation using wireless sensor network," *IEEE Transactions on Industrial Electronics*, vol. 61, no. 6, pp. 2776–2783, 2014.
- [5] X. Chen, R. Q. Hu, G. Wu, and Q. C. Li, "Tradeoff between energy efficiency and spectral efficiency in a delay constrained wireless system," *Wireless Communications and Mobile Computing*, vol. 15, no. 15, pp. 1945–1956, 2015.
- [6] J. Song, S. Han, A. K. Mok et al., "WirelessHART: applying wireless technology in real-time industrial process control," in *Proceedings of the 14th IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS '08)*, pp. 377–386, April 2008.
- [7] ISA100, <http://www.isa.org/isa100>.
- [8] IEC 62601:2015, <https://webstore.iec.ch/publication/23902/>.
- [9] IEEE 802.15 WPAN Task Group 4, <http://www.ieee802.org/15/pub/TG4.html>.
- [10] "IEEE 802.11 Wireless Local Area Networks," <http://www.ieee802.org/11/>.
- [11] IEEE 802.11-TASK GROUPS, http://www.ieee802.org/11/Reports/tgs_update.htm.
- [12] F. Tramarin, S. Vitturi, M. Luvisotto, and A. Zanella, "On the use of ieee 802.11n for industrial communications," *IEEE*

- Transactions on Industrial Informatics*, vol. 12, no. 5, pp. 1877–1886, 2016.
- [13] G. Tian, S. Camtepe, and Y.-C. Tian, “A deadline-constrained 802.11 MAC protocol with QoS differentiation for soft real-time control,” *IEEE Transactions on Industrial Informatics*, vol. 12, no. 2, pp. 544–554, 2016.
- [14] F. Babich, M. Comisso, M. D’Orlando, and A. Dorni, “Deployment of a reliable 802.11e experimental setup for throughput measurements,” *Wireless Communications and Mobile Computing*, vol. 12, no. 10, pp. 910–923, 2012.
- [15] D. K. Lam, K. Yamaguchi, Y. Shinozaki et al., “A fast industrial WLAN protocol and its MAC implementation for factory communication systems,” in *Proceedings of the 20th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA ’15)*, September 2015.
- [16] M. Neufeld, J. Fifield, C. Doerr, A. Sheth, and D. Grunwald, “Softmac—flexible wireless research platform,” in *Proceedings of the 4th Workshop on Hot Topics in Networks HotNets-IV*, 2005.
- [17] P. Djukic and P. Mohapatra, “Soft-TDMAC: a software-based 802.11 overlay TDMA MAC with microsecond synchronization,” *IEEE Transactions on Mobile Computing*, vol. 11, no. 3, pp. 478–491, 2012.
- [18] Y.-H. Wei, Q. Leng, S. Han, A. K. Mok, W. Zhang, and M. Tomizuka, “RT-WiFi: real-time high-speed communication protocol for wireless cyber-physical control applications,” in *Proceedings of the IEEE 34th Real-Time Systems Symposium (RTSS ’13)*, pp. 140–149, December 2013.
- [19] 802.11-1997 - IEEE Standard for Information Technology-Telecommunications and Information Exchange Between Systems-Local and Metropolitan Area Networks-Specific Requirements-Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications.
- [20] T. Gleixner and D. Niehaus, “Hrtimers and beyond: transforming the linux time subsystems,” in *Proceedings of the Ottawa Linux Symposium (OLS ’06)*, Ottawa, Canada, 2006.
- [21] W. Torfs and C. Blondia, “TDMA on commercial off-the-shelf hardware: fact and fiction revealed,” *AEU—International Journal of Electronics and Communications*, vol. 69, no. 5, pp. 800–813, 2015.
- [22] Open80211s project, <https://github.com/o11s/open80211s>.



Hindawi

Submit your manuscripts at
<https://www.hindawi.com>

