Suppressing Interference Between Periodic and Non-Periodic Traffic in Wireless Coexistence Scenarios

Ryota Ikeuchi and Hiroyuki Yomo
Graduate School of Engineering Science, Kansai University
3-3-35 Yamate-cho, Suita, Osaka, 564-8680 Japan
e-mail: {k097114, yomo}@kansai-u.ac.jp

***Abstract*—In this paper, we consider a wireless coexistence scenario where multi–radio platforms are employed to simultaneously support periodic and non–periodic traffic. Considering a scenario where wireless terminals generating periodic traffic over one frequency band change their operating band to the other band after detecting long–term communication failures, we consider how to suppress mutual interference between periodic and non–periodic traffic over the shared channel. In this paper, we propose a transmission control alleviating negative impact of mutual interference by exploiting interface heterogeneity, traffic periodicity, and queue management. As a means to suspend packet transmissions, we propose two types of queue management: transmission control in application layer and medium access control layer. The proposed schemes realize high packet delivery ratio of periodic traffic by suppressing transmissions of terminals with non–periodic traffic at the timing when periodic traffic is expected to be transmitted by their hidden terminals. We also investigate the feasibility of the proposed schemes with experiments. With computer simulations and experiments, we investigate the practicality and effectiveness of the proposed schemes.**

Keywords—Wireless Coexistence; Factory Automation; IEEE 802.11; IEEE 802.15.4; Internet of Things; Experimental Studies

I. Introduction

This paper is an extended version of [1], which investigates transmission control to suppress interference between heterogeneous traffic generated by wireless devices employing different access technologies. The proliferation of diverse wireless access technologies, such as LTE, WiFi, ZigBee, Bluetooth, etc., has been accelerated during the last decade to support heterogenous traffic with different requirements. Today, we have an option to simultaneously exploit these technologies with multi–radio platforms [2][3]: for instance, small, low–price IoT devices, which are equipped with multiple interfaces operating over different frequency bands, such as 2.4/5GHz and 920MHz, are commercially available [4].

In this paper, we exploit multi–radio platforms to enhance robustness of wireless networks in a highly noisy environment. A typical use–case is factory [5], where there are many metal objects blocking communication links between transmitters and receivers [6]. Furthermore, there can be noise emitted from industrial machines, as well as interference from many radio equipment around a factory. The resulting instability of communication channels causes temporal communication failure, which can last for a long period of time. If we employ wireless devices with a single interface operating over a specific frequency band in such an unstable environment, we cannot offer reliable transmissions of data: once blocking, noise or interference is generated over an operating frequency band, each device cannot avoid them. The lack of reliability for data transmissions in a factory can result in serious incidents that could even cause human life to be in danger. Therefore, in our work, we focus on the usage of wireless devices equipped with multiple radio interfaces operating at different frequency bands, called Flexible Terminal (FT). With FT, even if noise or interference is generated over one frequency band, its operating band can be changed to the other frequency band, which enables us to avoid communication failures due to noise and interference. More specifically, we employ radio standards operating at unlicensed frequency bands: IEEE 802.11 at 2.4 GHz and IEEE 802.15.4g at 920MHz since these standards are widely employed in many industrial fields [7].

Besides the heterogeneity of radio interface, the heterogeneity of communication traffic has become a common trend in current wireless networks. In addition to non–periodic (bursty) traffic generated by classical applications, such as Internet access and video/image transfer, more deterministic and periodic traffic has become a dominant pattern especially in a scenario with sensor devices deployed for monitoring purpose [8][9]. In general, small amount of data is generated by sensor devices, for which 920MHz radio supporting low data rate with large coverage is a favorable option. On the other hand, 2.4GHz commonly used by WiFi offers higher data rate with smaller coverage than 920MHz, which makes it suited for supporting Internet access and transfer of large–size image/video files. In this work, we employ FTs to simultaneously support periodic and non–periodic traffic. In a normal operation mode without any noise or interference, FTs with non–periodic traffic employ an interface operating at 2.4 GHz while FTs with periodic traffic use an interface operating at 920MHz. Then, we consider a scenario where noise or interference is generated by surrounding devices/machines over 920MHz, and each FT with periodic traffic changes its operating interface to that at 2.4GHz. In this case, there is mutual interference between FTs with periodic traffic and FTs with non–periodic traffic. In this work, we propose a transmission control, which suppresses mutual interference by exploiting interface heterogeneity, traffic periodicity, and queue management. In the proposed scheme, FTs with non–periodic traffic detect possible hidden FTs with periodic traffic by using difference of propagation characteristics of different frequency bands. Then, FTs with non–periodic traffic predict the transmission timing of FTs with periodic traffic, and suppress their packet transmissions at the predicted timing with adaptive queue management. With computer simulations and experiments, we investigate the practicality and possible gain of the proposed scheme. The new contributions of this paper in comparison to [1] are as follows.

- Besides the application–level queue management proposed in [1], in this paper, we propose a medium access control (MAC)–level queue management for our transmission control. With computer simulations, we show that the proposed MAC–level queue management achieves the best performance in terms of packet delivery ratio and throughput.
- Although the proposed MAC–level queue management achieves the best performance, it requires the queue management at physical layer (PHY)/MAC layer. In this paper, we investigate the feasibility of queue management at PHY/MAC layer by using a WiFi/Bluetooth coexistence function prepared in a off–the–shelf WiFi module. With the additional experiments, we show that the queue management at PHY/MAC layer is feasible, which confirms that the MAC–level queue management with WiFi/Bluetooth coexistence function is a promising means of interference management.
- More detailed explanations on the proposed schemes and experimental setting with additional figures are presented.

The rest of the paper is organized as follows. We review related work in Section II. After describing the system model in Section III, we present our proposed transmission control in Section IV. After showing and discussing some simulation results in Section V, Section VI presents the feasibility study of the proposed scheme with experiments. Finally, Section VII concludes the paper with several future work.

II. State of the Art

The periodicity of traffic has been exploited to avoid packet collisions among wireless devices in several existing studies. Most of them propose to schedule/shift the timing of packet generations of periodic flow so that they are not overlapped over time. These approaches can be categorized into the application–level and MAC–level. The MAC–level approaches are difficult to implement into the current wireless standards as it requires the modification of MAC protocols. On the other hand, the application–level approaches are easy to implement since it can be implemented over the existing MAC protocols. For instance, a Time Division Multiple Access (TDMA)–based MAC protocol with scheduling of periodic flows to overcome packet contentions has been proposed in [10]. A self–organizing TDMA protocol supporting periodic message exchange in vehicular networks is analyzed in [11]. As an application–level approach, a scheduling method is introduced in [12], where packet creation timing of periodic flow is adjusted in order to reduce contentions and packet collisions. However, these works only consider avoiding collisions among periodic flow and the coexistence and packet collisions with non–periodic traffic are not investigated.

Another related work to our study is the investigation on hidden terminal problem in carrier sense multiple access (CSMA) networks. The most well–known solution to hidden terminal problem is RTS/CTS handshake defined in IEEE 802.11 [13]. However, it has been reported that the efficiency of RTS/CTS handshake is low when short packets, such as small amount of data generated by sensor nodes in our scenario, are involved in data transmissions [14]. Furthermore, RTS/CTS mechanism does not fundamentally solve problems on collisions among hidden terminals: RTS frames transmitted by hidden terminals can collide with high probability [15]. Another requirement specific to industrial applications is more strict and deterministic protection for sensing data in comparison to Internet access/file transfer [16], which is difficult to achieve with RTS/CTS handshake even with quality of service (QoS) differentiation defined in IEEE 802.11e [17]. In contrast, our work proposes a mechanism to deterministically avoid interference between hidden terminals without resorting to RTS/CTS mechanisms.

III. System Model

In this section, we describe the system model considered in this paper.

A. System Model

In this work, we employ FTs with interfaces operating at 2.4GHz and 920MHz. In general, 920MHz signals have larger propagation distance than 2.4 GHz while the former achieves lower data rate than the latter. We consider a factory–like indoor area where FTs and a single Flexible Gateway (FG), which is in charge of aggregating data generated by FTs, are deployed as shown in Figure 1. The FG is also equipped with 2.4GHz and 920MHz interfaces to receive data from FTs. Some FTs are supposed to generate non–periodic, bursty, and heavy–load traffic, which are called NP–FTs. Since this type of traffic is in general supported by higher PHY rate at 2.4GHz that has limited communication range, we assume that NP–FTs are deployed near FG. On the other hand, FTs except for NP–FTs are assumed to generate periodic, light–load traffic, which are called P–FTs. A typical example of P–FT is a sensor device generating monitoring data of industrial machines and/or a given environment, which are deployed at various places within an area. This requires P–FTs to employ an interface and/or parameters realizing a larger communication range, for which 920MHz is more favorable option. We assume that the information on period of P–FT's traffic is known and shared by all FTs/FG. This is a reasonable assumption since these terminals and gateway are considered to be deployed by a single administrator of a factory. Furthermore, the timing of packet–generations of P–FTs are controlled to be equally separated over time so that they are not overlapped. This enables us to avoid contention among P–FTs. In a normal operation mode, NP–FTs employ 2.4GHz interface while P–FTs utilize 920MHz interface. Here, 2.4 GHz interface is supposed to follow IEEE 802.11 PHY/MAC protocol while 920MHz interface is in accordance with IEEE 802.15.4g/e PHY/MAC protocol. Note that both of these standards employ CSMA with collision avoidance (CSMA/CA) protocol. The FG receives data from both NP–FTs and P–FTs by using its

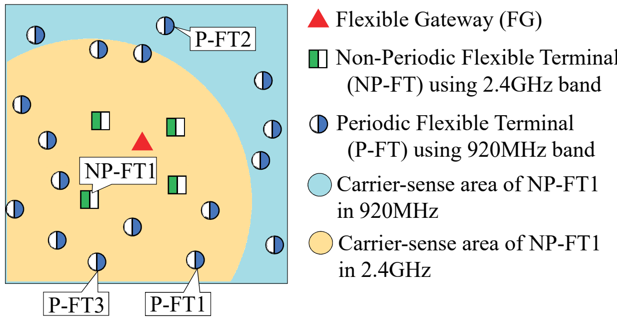

Figure 1. The considered, factory–like system model.

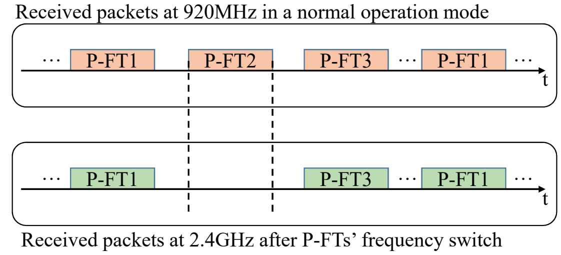

Figure 2. An example of hidden terminal detection.

two interfaces. It is assumed that the carrier–sense range of 2.4GHz interface is smaller than that of 920MHz as shown in Figure 1: for example, carrier–sense range of NP–FT1 in Figure 1 over 920MHz is sufficiently large to detect signals transmitted by all terminals while it can only sense signals transmitted by a part of terminals over 2.4GHz.

In this work, we consider a scenario where severe noise/interference is caused over 920MHz, which can be emitted from industrial machines and/or radio devices deployed inside/outside a factory area, and 920MHz interface suffers from continuous communication failures for a long period of time. As mentioned in Section I, FTs are able to switch their operating interface. Therefore, P–FTs, which operate with 920MHz interface in a normal operation mode, can switch their operating interface to 2.4GHz, e.g., after detecting continuous packet errors or after receiving some instruction if there is a central entity to monitor the radio environment. Here, each P–FT is assumed to employ low PHY rate (e.g., 1Mbps) at 2.4GHz, which enables each P–FT to achieve sufficiently large communication range to transmit data to FG. However, when P–FTs and NP–FTs share the same 2.4GHz frequency band, another problem can occur, which is a hidden terminal problem. For example, as shown in Figure 1, NP–FT1 and P-FT2 cannot sense their signals with each other at 2.4GHz. Therefore, CSMA/CA mechanisms do not work properly among these nodes after P–FT2 changes its operating band to 2.4GHz, which can cause packet losses at FG, thereby degrading packet delivery ratio and throughput. In this work, we propose a mechanism to suppress transmissions of NP–FTs to avoid interference with hidden P–FTs by exploiting interface heterogeneity, traffic periodicity, and adaptive queue management.

IV. Proposed Transmission Control

The proposed scheme controls packet transmissions of NP–FTs in order to suppress interference with their hidden P–FTs.

A. Mechanism to Detect Hidden Terminals

The NP–FTs first need to identify possible hidden terminals in order to suppress their mutual interference. This is achieved by exploiting the heterogeneity of interface. Each NP–FT observes traffic over 920MHz and 2.4GHz while they are not transmitting their own data. In the normal operation mode, P–FTs transmit data at 920MHz. In this case, each NP–FT finds packets of all P–FTs over 920MHz since they can easily reach each NP–FT thanks to a large communication range of 920MHz. An example of packet receptions at NP–FT1 shown in Figure 1 in a normal operation mode is depicted in the upper part of Figure 2. Here, P–FT1, P–FT2, and P–FT3 are the terminals whose locations are specified in Figure 1. As shown in Figure 2, NP–FT1 observes periodic receptions of all P–FTs at 920MHz interface. After P–FTs detect noise/interference at 920MHz, they switch their interfaces to 2.4GHz, where NP–FT1 receives packets only from P–FTs located within its communication range at 2.4GHz. The lower part of Figure 2 shows an example of packet receptions at NP–FT1 over 2.4GHz. As shown in the figure, NP–FT1 cannot receive packets transmitted by P–FT2 since P–FT2 is out of communication range of NP–FT1. Then, NP–FT1 finds that it has a hidden terminal of P–FT2 over 2.4GHz. At this timing, NP–FTs can also find that P–FTs have changed their operating band to 2.4GHz. Thus, by comparing packet receptions at 920MHz and 2.4GHz, each NP–FT can identify its hidden terminals over 2.4GHz, whose packets can cause collisions against itself.

B. Basic Idea of Proposed Transmission Control

While receiving packets from P–FTs in the normal operation mode, each NP–FT records the reception timing of each P–FT. Based on this information and pre–knowledge of the period of packet transmissions of each P–FT, each NP–FT predicts the timing of periodic packet transmissions. Then, each NP–FT suppresses its packet transmissions when the transmissions of its hidden P–FTs are expected. This is achieved by our proposed Transmission Control (TC), which executes queue management to control timing of packet transmissions at different layers.

The basic idea of the proposed TC is shown in Figure 3. Here, the blue solid arrow shows the predicted transmission timing of a hidden P–FT. With the proposed TC, a duration called Suspending Duration (SD), which consists of Pre–SD (before the predicted timing) and Post–SD (after the predicted timing) is prepared. A NP–FT attempts to suspend its packet transmission over SD with queue management described later. In Figure 3, the dashed green arrow represents the timing when packets are generated at upper layer of NP–FT. Once SD is over, NP–FT starts transmitting packets. Note that packets generated at non–SD duration can be immediately transmitted as in the packet P4 in Figure 3. The flowchart of these operations of the proposed transmission control is

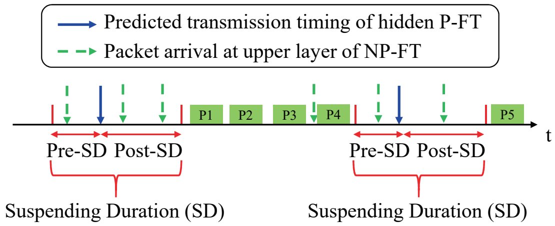

Figure 3. Basic idea of the proposed transmission control.

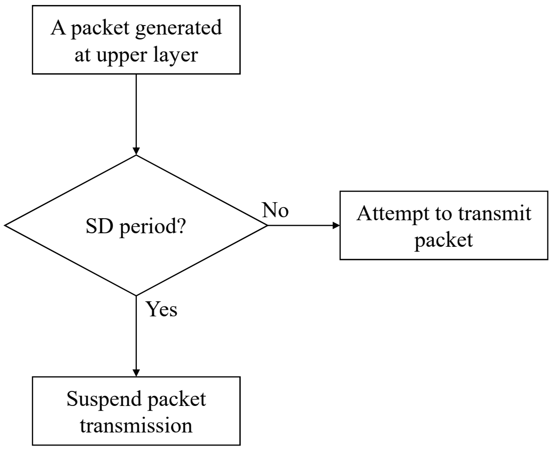

Figure 4. A flowchart of basic operations of the proposed transmission control.

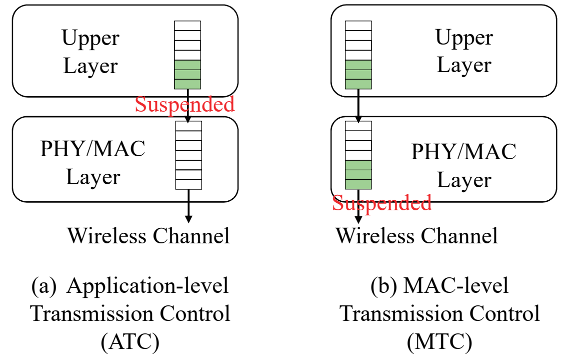

Figure 5. Proposed queue management for transmission control: (a) application–level transmission control (ATC) and (b) MAC–level transmission control (MTC).

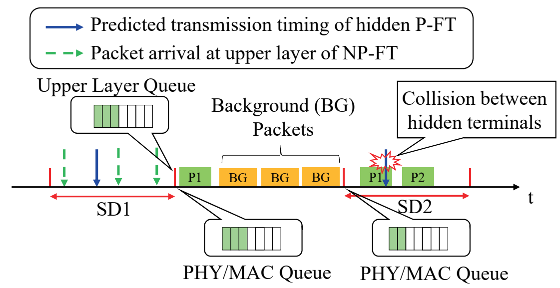

Figure 6. An example of problem on controlling packet transmissions with ATC.

shown in Figure 4. The duration of Pre–SD and Post–SD are decided considering trade–off between achievable Packet Delivery Ratio (PDR) of P–FTs and throughput of NP–FTs as discussed in Section V-B in more detail.

As a means to suspend packet transmissions during SD, we propose two types of queue management: application–level TC (ATC) and MAC–level TC (MTC), which are respectively depicted in Figure 5 (a) and (b). With ATC, even if packets are generated at upper layer, they are stored into upper layer queue without passing them into PHY/MAC layer. On the other hand, with MTC, packets are passed from upper layer to PHY/MAC layer even during SD. However, each NP–FT suspends packet transmissions at PHY/MAC level, i.e., it does not transmit any packet over the air (i.e., wireless channel) during SD. Note that MTC requires us to modify firmware installed into WiFi module/chip so that we can arbitrarily control transmission timing of packets at PHY/MAC level. Thus, MTC has higher complexity of implementation than ATC.

C. Drawback of ATC

Although ATC has lower complexity than MTC as described above, it has difficulty to precisely control the timing when signals are actually transmitted at PHY/MAC level. This problem is explained in an example shown in Figure 6. Here, a NP–FT suspends passing packets to PHY/MAC layer during the first SD, and three packets are stored in the upper–layer queue. These packets are passed to PHY/MAC layer after the first SD is over, which are then stored in PHY/MAC queue. The transmissions of packets in PHY/MAC queue are managed by PHY/MAC module, therefore, these packets are transmitted if they win contentions against the other terminals. In the example of Figure 6, it is supposed that NP–FT succeeds in transmitting a packet P1 by winning the contention. However, it fails to transmit packets P2 and P3 due to the lost contentions with BackGround (BG) traffic. Then, these 2 packets remain in PHY/MAC queue in the beginning of the next SD. As mentioned above, packet transmissions of these lower–layer packets are controlled by PHY/MAC module, therefore, they can be transmitted even during SD, which can cause a collision with packets transmitted by hidden P–FTs.

A possible solution to the above–mentioned problem is to control the number of packets to be passed to PHY/MAC layer module based on the congestion level over the channel, i.e., each NP–FT controls the number of packets passed to PHY/MAC layer module in the end of SD in such a way that these packets can be transmitted in the following non–SD period at the PHY/MAC level. This requires each NP–FT to continuously monitor the congestion level over the operating channel. Note that background traffic at 2.4GHz are not necessarily generated by WiFi terminals, whose packets can be decoded by NP–FT, but generated by the other radio equipment, e.g., Bluetooth or Microwave oven. In this case, each NP–FT needs to monitor the congestion level without decoding each background signal. Therefore, in the following subsection, we first investigate whether it is practically possible for a WiFi terminal to conduct real–time monitoring of busy rate (i.e., fraction of time, during which the channel is occupied by radio signals) of a channel.

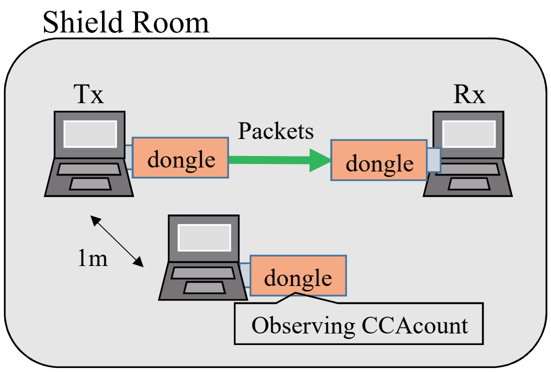

Figure 7. Experimental setting to investigate the feasibility to use CCAcount as a measure of busy rate.

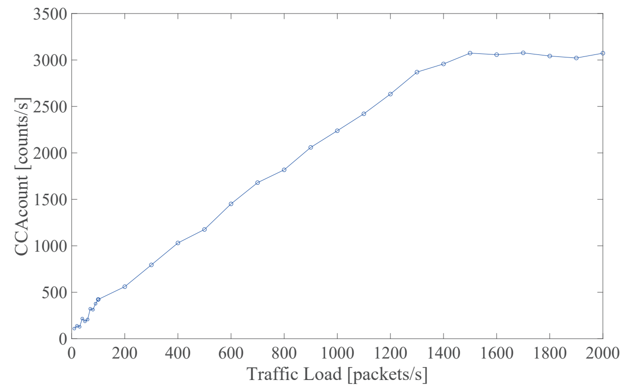

Figure 8. Experimental results on CCAcount against traffic load.

D. Feasibility to Monitor Congestion Level

We found a parameter called CCAcount in a device driver of an off–the–shelf WiFi module (Buffalo WL–U3–866DS [18]). The parameter seems to be related to busy rate of a channel, however, there was no evidence that this parameter represents our desired information on busy rate. Therefore, we conducted experiments to check the relationship between CCAcount and busy rate of a channel. The experimental setting is shown in Figure 7. In the experiments, we prepared 3 laptop PCs with USB dongles of WL–US–866DS. A laptop PC (Tx PC) was configured to be a transmitter of packets, which are directed to Rx laptop PC. A laptop PC to observe CCAcount was located at a sufficiently close position to Tx PC. The busy rate was varied by changing the number of packets transmitted per a unit time, for which the output of CCAcount was monitored at the observing PC. The PHY rate, packet size, and ACK size of packet transmissions were respectively set to be 54Mbps, 1496Bytes, and 46Bytes. The busy rate for each traffic load can be calculated based on these parameters. The measurements were conducted inside a shield room.

Figure 8 shows the output of CCAcount against traffic load (packets/s). From this figure, we can see that CCAcount increases as traffic load increases, which saturates over the range of high traffic load. There is a maximum traffic load that can be generated by a single WiFi terminal, which depends on back–off parameters and Inter–Frame Space (IFS) of IEEE 802.11, where the saturation is observed. From this figure, we can confirm that there is a direct relationship between CCAcount and traffic load, i.e., busy rate of the channel, which enables us to employ CCAcount as a measure of busy rate of the channel.

E. ATC with traffic adaptation

In this work, we introduce traffic adaptation into ATC, which controls the number of packets to be passed to PHY/MAC layer based on the observed CCAcount. In ATC, each NP–FT observes CCAcount during each non–SD period. This can be realized only by obtaining the corresponding information from WiFI device driver. The output of CCAcount is converted to the traffic load by using a linear equation approximating the relationship between CCAcount and traffic load over the load range of [0:1500] packets/s in Figure 8, which is used to calculate the busy rate. Based on the derived busy rate, the maximum number of packets permitted to be passed to PHY/MAC layer at the next non–SD period, N_{max}, is decided. N_{max} is calculated as follows:

$$N_{max} = \frac{(1 - B_{ave})T_{NSD}}{T_D \cdot \alpha}. \tag{1}$$

Here, T_{NSD} is the duration of next non–SD period, T_D is the duration required to transmit a single data frame including SIFS and ACK duration, and α is a parameter to vary effective number of N_{max}, and B_{ave} is average busy rate calculated as

$$B_{ave} = \frac{\sum_{i=1}^{W} B_i}{W}, \tag{2}$$

where B_i is busy rate calculated for the i-th last non–SD period, and W is the window size (number of non–SDs) used for calculating average busy rate. N_{max} calculated with (1) represents the estimated (effective) number of packets that can be transmitted by a single NP–FT during free period in the following non–SD period. Note that α is introduced in order to take the impact of back–off duration and number of contending FTs into account. With smaller (larger) α, the estimation of N_{max} becomes more optimistic (pessimistic). The range of α considered in this paper is set to [0.4, 6.0].

The proposed ATC is executed in the end of every SD period. For instance, in the end of SD1 in Figure 6, N_{max} is calculated by using busy rate over the last W non–SD periods. Then, if the number of packets stored in the upper–layer queue is equal to or more than N_{max}, only N_{max} packets out of stored packets are passed to PHY/MAC layer, and no more packets are passed to PHY/MAC layer during the following non–SD period. Otherwise if the number of packets stored in the upper–layer queue is less than N_{max}, all stored packets are passed to PHY/MAC layer. Then, newly arriving packets in the following non–SD period can be passed to PHY/MAC layer as long as the total number of packets passed to PHY/MAC layer does not exceed N_{max}. With these operations, we can reduce the probability that packets remain in PHY/MAC queue in the end of each non–SD period.

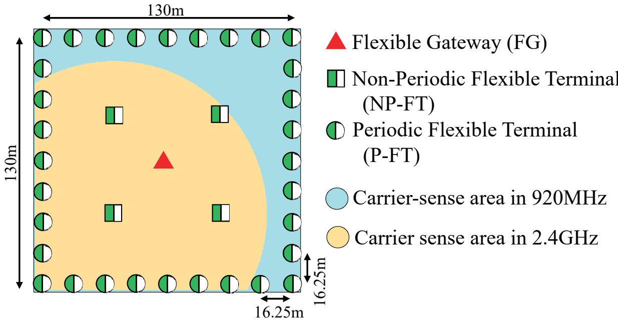

Figure 9. Simulation Model.

TABLE I: Simulation Parameters

	NP–FT	P–FT
PHY rate	54Mbps	1Mbps
Communication range	75m	100m
Carrier-Sense Range	100m	100m
Packet generation	Poisson (mean λ)	period = 1s
Data size	2000Bytes	200Bytes
ACK size	30Bytes	
DIFS	28μs	
SIFS	10μs	
Slot time	20μs	
Max. Num. of Retransmissions	3	
Min. Contention Window	31	
Simulation Duration	20s	

V. Simulation Model and Results

In this section, we provide numerical results obtained by our computer simulations, and discuss the benefit brought by the proposed transmission control in detail.

A. Simulation Model

The simulation model is shown in Figure 9. The layout given in Figure 9 is selected since it can increase the number of hidden terminals, which allows us to consider a worst–case scenario. In the simulations, communication performance after P–FTs change their operating frequency band to 2.4GHz is evaluated. The main parameters used in simulations are shown in Table I. Most of the parameters are taken from the IEEE 802.11g standard [19]. The P–FTs generate packets with period of 1s, and their generation timing are scheduled so that they do not overlap with each other. In the evaluation, since there are 32 P–FTs, a period of 1s is divided into 32 sections, and the beginning of each section is randomly assigned to each P–FT as its generation timing. Each NP–FT applies the proposed ATC/MTC to its hidden P–FTs. We use the application–level PDR of P–FT and throughput of NP–FT as performance measures. A packet is decided to be lost and discarded once the number of retransmissions reaches the maximum value. For simplicity, packet errors are assumed to occur only due to collisions. The throughput is defined as the amount of data successfully delivered by NP–FTs to FG. The simulation is conducted by a custom–made simulator developed with Matlab software.

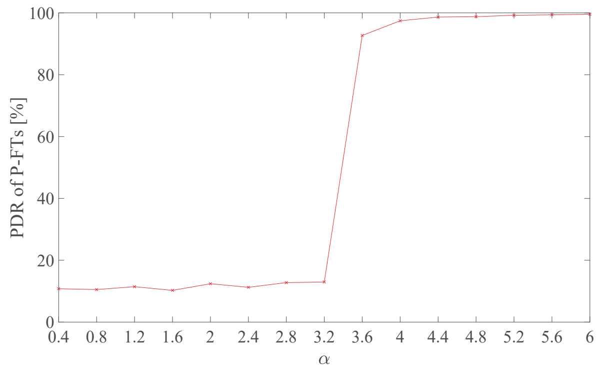

Figure 10. PDR of NP-FTs against α for ATC.

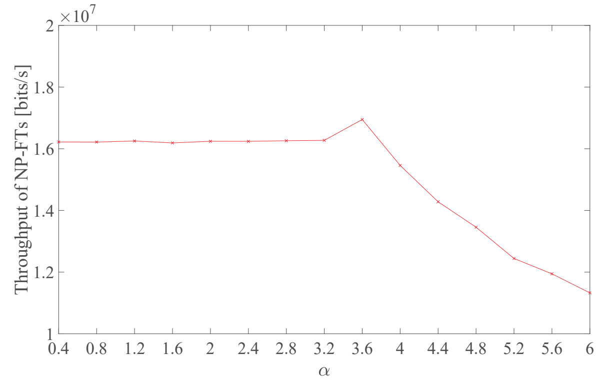

Figure 11. Throughput of NP-FTs against α for ATC.

B. Simulation Results

Below, we show simulation results averaged over 5 simulation trials. Figure 10 shows PDR of P–FTs against the parameter of α in (1) when the proposed ATC is employed with Pre–SD = 2ms, Post–SD = 6ms, W = 10, and $\lambda = 400$ [packets/s]. From Figure 10, we can see that PDR of P–FT is degraded with smaller α. With smaller α, each NP–FT passes a larger number of packets to PHY/MAC layer in the end of SD as calculated by (1), which exceeds the number of packets that can be transmitted at PHY/MAC layer during the next non–SD period. In this case, packets remained in PHY/MAC queue can be transmitted simultaneously with hidden P–FTs, which causes collisions with high probability. This problem is alleviated by increasing the value of α where the number of packets passed to PHY/MAC layer is reduced. Therefore, PDR of P–FT is improved with larger value of α. However, larger values of α force each NP–FT to keep more packets in its upper–layer queue, and degrade its throughput performance. This is confirmed in Figure 11, where throughput of NP–FTs against α is shown. The throughput of NP–FTs is largely degraded with too large α, i.e., the range of α exceeding 3.6. From these results, we can see that there is an appropriate value of α to be employed to achieve both high PDR of P–FTs and high throughput of NP–FTs. In the following evaluations, we employ $\alpha = 3.6$ based on the above results.

Next, we investigate the impact of SD length on the achievable performance of the proposed ATC. We have observed the same tendency for ATC and MTC, therefore, we only

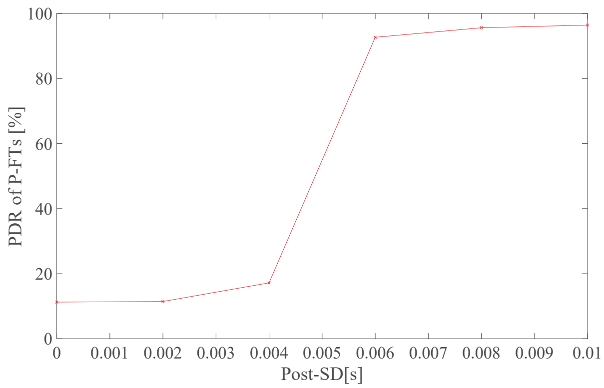

Figure 12. PDR of P-FTs against Post-SD for ATC.

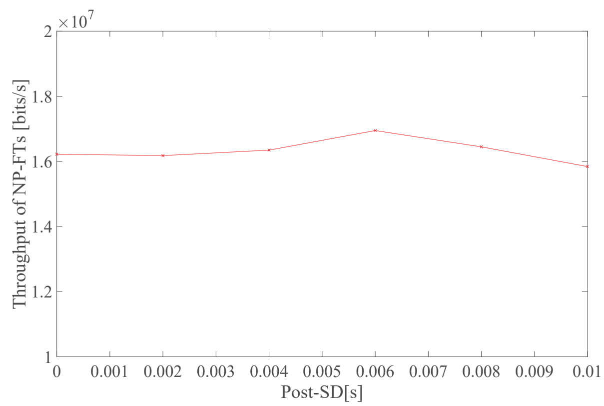

Figure 13. Throughput of NP-FTs against Post-SD for ATC.

show results of ATC here. Figures 12 and 13 respectively show PDR of P–FTs and throughput of NP–FTs against the length of Post–SD, where Pre–SD is fixed to be 2ms, $W = 10$, $\alpha = 3.6$, and $\lambda = 400$ [packets/s]. First, from Figure 12, we can see that a sufficiently large value of Post–SD is required to achieve high PDR of P–FTs. Each packet generated at P–FT is transmitted with CSMA/CA protocol, where its actual transmission timing at PHY/MAC level can be delayed due to contentions with the other NP–FTs and P–FTs within its carrier–sense range. Therefore, if NP–FT employs too small Post–SD, it can transmit packets with hidden P–FTs whose transmissions are delayed due to CSMA/CA operations. The increase of Post–SD also offers the improvement on throughput as shown in Figure 13 thanks to higher probability to avoid mutual collisions, however, too large Post–SD leads to the reduction of throughput of NP–FTs since it can reduce the duration for NP–FTs to be able to transmit their packets. From these figures, we can see that Post–SD of 6ms is an appropriate choice in our considered settings.

Finally, we respectively show PDR of P–FTs and throughput of NP–FTs against traffic load of NP–FTs for different schemes in Figures 14 and 15. Here, we set Pre–SD = 2ms, Post–SD = 6ms, $W = 10$, and $\alpha = 3.6$. The results of W/O TC in these figures represent achievable performance of an existing scheme, which follows conventional IEEE 802.11 MAC protocol without employing our proposed TC. We also show results of ATC without traffic adaptation in these figures.

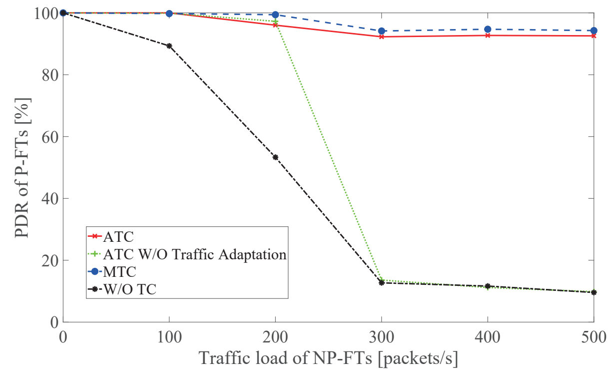

Figure 14. PDR of P-FTs against traffic load of NP-FTs.

From Figure 14, we can first see that PDR of P–FTs is largely degraded if we do not employ TC. This is due to packet collisions between NP–FTs and their hidden P–FTs. By introducing ATC, PDR of P–FTs can be improved, however, we can obtain gain only over the range of small traffic load of NP–FTs if we do not introduce traffic adaptation into ATC. As the traffic load of NP–FTs increases, more packets are stored in the upper–layer queue in the end of each SD, which can exceed the number of packets that can be handled at PHY/MAC level during the following non–SD period. Therefore, more collisions occur for larger traffic of NP–FTs, which degrades PDR of P–FTs. On the other hand, it can be seen that the proposed ATC with traffic adaptation achieves high PDR of P–FTs even for larger traffic load of NP–FTs thanks to the adjustment of number of packets passed to PHY/MAC queue, which is adapted to the observed traffic load. The proposed MTC achieves the highest PDR since it can stop/start the transmissions of packets at PHY/MAC level according to the schedule of SD and non–SD. However, we can see that the proposed ATC also achieves PDR close to MTC. Next, from Figure 15, we can see that the proposed ATC does not degrade throughput of NP–FTs even with the introduction of SD. The avoidance of collisions eventually leads to throughput improvement. With the proposed ATC, packets are stored in the upper–layer queue according to the estimated traffic load. If the actual traffic load is smaller than the estimated value, all packets passed to PHY/MAC queue can be transmitted at early timing within a non–SD period, after which no packet is transmitted since there is no packet in PHY/MAC queue. This problem does not occur with the proposed MTC, therefore, throughput of the proposed ATC does not reach close to MTC. From these results, we can confirm that the proposed ATC, which has lower complexity than MTC, can significantly improve PDR of P–FTs while achieving slightly better throughput of NP–FTs in comparison to the the case without TC. Furthermore, the proposed MTC has the highest PDR and throughput at the cost of complexity of implementation.

VI. Feasibility study of MTC with experiments

In the previous section, we have shown that the proposed MTC achieves the best PDR and throughput. However, the

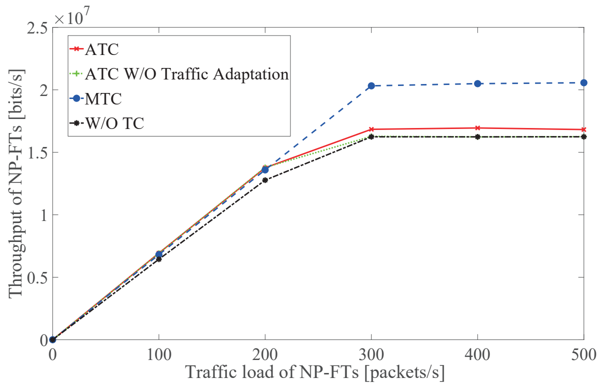

Figure 15. Throughput of NP-FTs against traffic load of NP-FTs.

proposed MTC requires us to control the transmissions of packets at the lowest level of protocol stack, i.e., to conduct queue management at PHY/MAC level. In this section, we investigate the feasibility of queue management at PHY/MAC layer with experiments.

A. Experimental Setting

In order to realize the queue management at PHY/MAC layer, we attempt to utilize a function of WiFi/Bluetooth coexistence implemented in some WiFi/Bluetooth combined modules. This function is prepared for a module to stop the transmissions of WiFi packets while transmitting Bluetooth signals, thereby avoiding interference between them. The basic operation to realize queue management at PHY/MAC layer by exploiting WiFi/Bluetooth coexistence function is shown in Figure 16. In general, WiFi PHY/MAC module has layered structure of queue management. The lowest queue is called as CSMA queue, which stores a packet to be transmitted if the corresponding node wins the contention through CSMA operations. The other packets are stored in PHY/MAC queue, waiting for the process of CSMA. The WiFi/Bluetooth coexistence function is able to suspend passing packets from PHY/MAC queue to CSMA queue while it transmits Bluetooth signal. In this work, we configure the WiFi/Bluetooth coexistence function so that it can output the suspending command to WiFi module at arbitrary timing even if there is no actual transmission of Bluetooth signal. This way, we can control the timing for PHY/MAC queue to pass a packet to CSMA queue. Although we cannot control the transmission of CSMA queue once a packet is inserted into it, we can minimize the number of packets out of control by using this function. The function is implemented by using JWX6051 module with AR9380 chip onboard [20].

The experimental setting is shown in Figure 17. In the experiment, we use three JWX6051 WiFi modules. WiFi Module 1 is supposed to be a transmitter of periodic traffic (i.e., P–FT) while WiFi Module 2 is assumed to be a transmitter of non–periodic traffic (i.e., NP–FT). As described in Section III, we consider a scenario where these two modules are hidden with each other. In order to construct a situation where these two modules are hidden with each other, they are respectively put

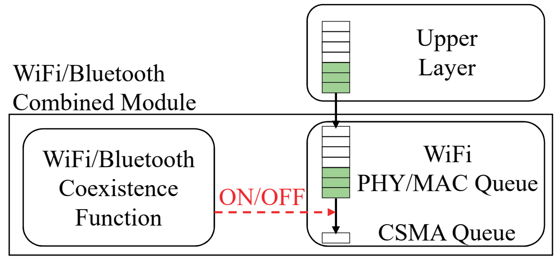

Figure 16. The queue management at PHY/MAC layer exploiting WiFi/Bluetooth coexistence function.

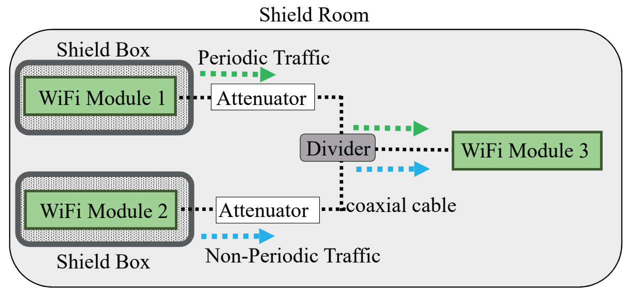

Figure 17. The experimental setting of feasibility study of MTC.

into shield boxes. This way, one module cannot sense radio signals transmitted by the other module. In order to extract radio signals transmitted by each module, they are connected to coaxial cable. These cables are combined by a divider after attenuation. Finally, the combined signals are received by WiFi Module 3 through a coaxial cable. With this setting, we can imitate a situation where P–FT and NP–FT are hidden with each other. All the experiments are conducted inside a shield room. The parameters employed in the experiments are shown in Table II.

B. Experimental Results

In the experiments, the periodic traffic is generated by WiFi Module 1 with the interval of 1.024s. After 100 packets are transmitted by WiFi Module 1, its transmission timing is shifted by 0.1s. This is repeated 10 times. This realizes 10 different gaps between the generation timing of periodic traffic and SD set by WiFi Module 2 for the transmissions of non–periodic traffic.

Figure 18 shows PDR of periodic traffic over time when MTC exploiting WiFi/Bluetooth coexistence function is em-

TABLE II: Parameters for experiments

Frequency band	2.4 GHz
Data Size	1554 Bytes
Data Rate of Non–periodic Traffic	18 Mbps
Data Rate of Periodic Traffic	1 Mbps
ACK Size	46 Bytes
ACK Rate of Non–Periodic Traffic	12 Mbps
ACK Rate of Periodic Traffic	1 Mbps
Generation Rate of Non-Periodic Traffic	200 packets/s
Period of SD	1 s
SD	0.2 s
Non–SD	0.8 s

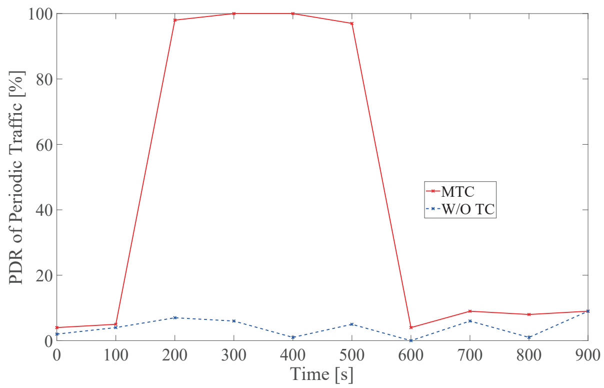

Figure 18. PDR of periodic traffic against time shift.

ployed. We also plot experimental results without employing TC in the same figure. Each value in the horizontal axis corresponds to the gap between the transmission timing of periodic traffic and SD set for MTC. From this figure, we can first see that PDR of periodic traffic is largely degraded when TC is not employed. This is because WiFi Module 1 and Module 2 are hidden with each other, and their transmitted packets collide at WiFi Module 3 with high probability if we do not employ any transmission control. Some periodic packets can be received successfully when they are transmitted during back–off periods of non–periodic traffic, therefore, PDR is larger than 0%. On the other hand, Figure 18 shows that PDR of MTC is 100% for the time of 300s and 400s. Over this range of time, periodic traffic is transmitted during SD period of MTC. This means that queue management exploiting WiFi/Bluetooth coexistence function properly works for avoiding collisions with periodic traffic. For 200s and 500s, some periodic packets are lost due to collisions. This is probably caused by the difficulty to control the transmission of packet stored into CSMA queue. For the other region of horizontal axis, PDR of MTC is degraded because the transmission timing of periodic traffic is outside of SD of MTC. We also confirmed that throughput of non–periodic traffic is not degraded even if we employ MTC using WiFi/Bluetooth coexistence function. From these results, we can conclude that WiFi/Bluetooth coexistence function is a promising means to realize MTC.

VII. Conclusions

In this paper, focusing on a wireless coexistence scenario where multi–radio platforms are employed to support heterogenous traffic, we proposed transmission control, which suppresses mutual interference between hidden terminals generating periodic and non–periodic traffic. The proposed transmission control exploits interface heterogeneity, traffic periodicity, and queue management in order to suppress interference. As a means to suspend packet transmissions, we proposed two types of queue management: application–level transmission control and MAC–level transmission control. Furthermore, we proposed traffic adaptation for application–level transmission control, which adapts the amount of packets passed from the upper layer according to the observed congestion level. We first confirmed with experiments the practicality for WiFi device to monitor congestion level in a real–time manner, which is required for traffic adaptation. Then, we evaluated the gain of the proposed application–level and MAC–level transmission control in terms of packet delivery ratio and throughput by computer simulations. Our numerical results showed that the proposed application–level and MAC–level transmission control significantly improve packet delivery ratio of periodic traffic while slightly improving throughput of non–periodic traffic in comparison to reference schemes. Finally, we investigated the feasibility of MAC–level transmission control with experiments. We realized the queue management at PHY/MAC level by exploiting WiFi/Bluetooth coexistence function. Our experimental results showed that WiFi/Bluetooth coexistence function is a promising means to realize MAC–level transmission control.

Our future work includes experimental evaluations of the proposed application–level and MAC–level transmission control with actual multi–radio platforms in a practical environment. More extensive verifications of simulation results, e.g., with a larger number of simulation trials and comparison with theoretical results, are also our future work. Furthermore, in this paper, it is assumed that the transmission timing of periodic traffic can be ideally estimated by the other terminals. However, in practice, this estimation can be incomplete, which can shift suspending duration from the desired duration. This causes degradation of packet delivery ratio of periodic traffic and throughput of non–periodic traffic. Therefore, we need to design a practical mechanism of synchronization, and to evaluate the impact of estimation error on the achievable performance of the proposed transmission control.

Acknowledgement

This work includes results of the project entitled “R&D on Technologies to Densely and Efficiently Utilize Radio Resources of Unlicensed Bands in Dedicated Areas,” which is supported by the Ministry of Internal Affairs and Communications as part of the research program “R&D for Expansion of Radio Wave Resources (JPJ000254)”.

References

[1] R. Ikeuchi and H. Yomo, “Transmission Control to Suppress Interference Between Periodic and Non-Periodic Traffic in Wireless Coexistence Scenarios,” *Proc. of The Fifteenth International Conference on Digital Telecommunications (ICDT 2020)*, Feb. 2020.

[2] J. Gummeson, D. Ganesan, M. D. Corner, and P. Shenoy, “An Adaptive Link Layer for Heterogenous Multi-Radio Mobile Sensor Networks,” *IEEE Journal on Selected Areas in Communications*, vol. 28, no. 7, pp. 1094–1104, Sept. 2010.

[3] D. K. Tosh and S. Sengupta, “Heterogenous Access Network(s) Selection in Multi-Interface Radio Devices,” *IEEE International Workshop on Managing Ubiquitous COmmunications and Services*, pp. 117–122, 2015.

[4] “Smart Energy Gateway CUBE, NextDrive Co.” accessed: 2020-012-04. [Online]. Available: https://www.nextdrive.io/en/cube-smart-energy-gateway/

[5] S. Dietrich, G. May, O. Wetter, H. Heeren, and G. Fohler, “Performance Indicators and Use Case Analysis for Wireless Networks in Factory Automation,” in *2017 22nd IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, Sep. 2017, pp. 1–8.

[6] M. Dungen *et al.*, “Channel Measurement Campaigns for Wireless Industrial Automation,” *Automatisierungstechnik*, vol. 67, no. 1, pp. 7–28, 2019.

[7] A. Varghese and D. Tandur, “Wireless Requirements and Challenges in Industry 4.0,” *Proc. of 2014 International Conference on Contemporary Computing and Informatics (IC3I)*, pp. 634–638, 2014.

[8] M. Sansoni *et al.*, “Comparison of M2M Traffic Models Against Real World Data Sets,” *Proc. of IEEE 23rd International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD)*, pp. 1–6, 2018.

[9] X. Cao, J. Chen, Y. Cheng, X. X. Shen, and Y. Sun, “An Analytical MAC Model for IEEE 802.15.4 Enabled Wireless Networks With Periodic Traffic,” *IEEE Transactions on Wireless Communications*, vol. 14, no. 10, pp. 5261–5273, Oct. 2015.

[10] T. W. Carley, M. A. Ba, R. Barua, and D. B. Stewart, “Contention-free Periodic Message Scheduler Medium Access Control in Wireless Sensor/Actuator Networks,” in *RTSS 2003. 24th IEEE Real-Time Systems Symposium, 2003*, 2003, pp. 298–307.

[11] T. Gaugel, J. Mittag, H. Hartenstein, S. Papanastasiou, and E. G. Strom, “In-depth Analysis and Evaluation of Self-organizing TDMA,” in *2013 IEEE Vehicular Networking Conference*, 2013, pp. 79–86.

[12] A. H. Nguyen, Y. Tanigawa, and H. Tode, “Scheduling Method for Solving Successive Contentions of Heterogeneous Periodic Flows Based on Mathematical Formulation in Multi-Hop WSNs,” *IEEE Sensors Journal*, vol. 18, no. 21, pp. 9021–9033, 2018.

[13] Kaixin Xu, M. Gerla, and Sang Bae, “How Effective is the IEEE 802.11 RTS/CTS Handshake in Ad Hoc Networks,” in *Global Telecommunications Conference, 2002. GLOBECOM ’02. IEEE*, vol. 1, 2002, pp. 72–76 vol.1.

[14] L. Zhang *et al.*, “Signal Strength Assistant Grouping for Lower Hidden Node Collision Probability in 802.11ah,” *Proc. of 9th International Conference on Wireless Communications and Signal Processing (WCSP)*, pp. 1–6, Oct. 2017.

[15] K. Shih, W. Liao, H. Chen, and C. Chou, “On Avoiding RTS Collisions for IEEE 802.11-based Wireless Ad Hoc Networks,” *Elsevier Computer Communications*, vol. 32, no. 1, pp. 69–77, Jan. 2009.

[16] A. A. K. S., K. Ovsthus, and L. M. Kristensen, “An Industrial Perspective on Wireless Sensor Networks: A Survey of Requirements, Protocols, and Challenges,” *IEEE Communications Surveys & Tutorials*, vol. 16, no. 3, pp. 1391–1412, Third Quarter 2014.

[17] “IEEE Standard for Information technology–Local and metropolitan area networks–Specific requirements–Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications - Amendment 8: Medium Access Control (MAC) Quality of Service Enhancements,” *IEEE Std 802.11e-2005 (Amendment to IEEE Std 802.11, 1999 Edition (Reaff 2003)*, pp. 1–212, 2005.

[18] “BUFFALO, WI-U3-866DS,” accessed: 2020-12-04. [Online]. Available: https://www.buffalo.jp/product/detail/wi-u3-866ds.html

[19] D. Vassis, G. Kormentzas, A. Rouskas, and I. Maglogiannis, “The IEEE 802.11g Standard for High Data Rate WLANs,” *IEEE Network*, vol. 19, no. 3, pp. 21–26, May 2005.

[20] “JWX6051, Qualcom Atheros Inc.” accessed: 2020-12-04. [Online]. Available: minipci.biz/data/datasheets/jjplus/SpecSheet_JWX6051-0_20161006.pdf