Improving ZigBee Networks Robustness with Multi-channel Capability

Xiaolei Cui

Assignment: Master of Science Thesis
Instructor: Ir. Wei Yuan
Dr. Martin Jacobsson
Supervisor: Prof.dr.ir. Ignas Niemegeers
Date: 10-08-2009

Wireless and Mobile Communication group
Faculty of Electrical Engineering,
Mathematics and Computer Sciences
Delft University of Technology

Distributed Sensor System Group,
Philips Research,
Eindhoven

Acknowledgement

By the time of closing my master thesis project, I would like to thank my daily mentor Wei Yuan who gives me the opportunity to finish this project at Philips Research, Eindhoven. Without his profound knowledge, suggestions and support, I cannot come to this final outcome. I've learned a lot from him during the time at Philips, not only academic knowledge, but also the academic way of thinking.

I am extremely grateful to my mentor at TU Delft, Dr. Martin Jacobsson, with whose help I've worked out this thesis project. Also I express my gratitude to my coordinator and student assistant supervisor Dr. Gerard Janssen. His support, patience and kindness encourage me a lot during my study and work in Delft. This acknowledgement goes to Prof. Ignas Niemegeers, telecommunication department and my colleagues at Philips Research as well, their kind support and help have been of great value in this thesis project.

Thanks for my dear families and friends in Delft, Eindhoven and of course in China. Your encouragement is the most important thing for bringing this thesis project to fruition.

Abstract

Based on IEEE 802.15.4, ZigBee is developed for low-power and low-data-rate wireless communication and it is building up remarkable position for wireless sensor network (WSN). As ZigBee is using the 2.4GHz Industrial, Scientific, and Medical (ISM) unlicensed frequency band, coexistence issues arise as there are also other wireless technologies sharing the same band, such as 802.11b/g WiFi, Bluetooth, cordless phones and even microwave ovens. Due to the low transmission power, ZigBee is potentially vulnerable to the interference introduced by these technologies rather than vice versa. Therefore, it is desirable to improve the robustness of ZigBee networks. As WiFi is widely deployed and often collocated with ZigBee networks in applications, such as hospitals and home buildings, we take WiFi as the main interference source and work on finding solutions to enhance the robustness of ZigBee networks under WiFi as well as other interferences.

To improve the robustness of ZigBee networks, a feature called frequency agility is specified in the ZigBee standard. We found, however, some inadequacies in the standard that needs to be improved before the frequency agility can function well in practice as it is supposed to do. A better periodical window method is proposed to improve the detection time to interference. Besides, in case that there is only a part of the whole network suffering from some local interference, it is neither necessary for the whole network to move to a new idle channel because this movement is costly and risky, nor possible to find an idle channel for the whole network to move to. Therefore, we extend the frequency agility function by enabling a single ZigBee network to work on multiple channels. As some local interference appears, the part of the network which is under the interference can move to a new idle channel while maintaining the communication links with the other part of the network which stays on the original channel and the moved part can move back to the original channel when the interference disappears.

OPNET simulations shows that our multi-channel solution can significantly improve the robustness of ZigBee networks in a cost-efficient way,

Table of Content

Chapter 1 Introduction

Wireless sensor networks (WSNs) are becoming popular telecommunication technologies recently. It can be treated as a collection of nodes organized into a cooperative network and these individual nodes are able to interact with their environment by sensing or controlling [1]. Many wireless sensor networks are using the ZigBee protocol, which is based on the IEEE 802.15.4 standard. Due to this, they are distinguished from other wireless technologies by lower power consumption, lower data rate, and lower device cost. At the moment, wireless sensor networks are widely used for environmental monitoring, intelligent control, medical and health care, logistics and various other applications.

As the foundation of wireless sensor networks, IEEE 802.15.4 standard is finalized by the Institute of Electrical and Electronics Engineers (IEEE) and it covers both physical layer and MAC sublayer of a low-rate Wireless Personal Area Network (WPAN). Based on IEEE 802.15.4, the ZigBee Alliance, an association of companies working together to enable reliable, cost-effective, low-power, wirelessly networked, monitoring and control products based on an open global standard [2], builds up the entire ZigBee protocol stack architecture by adding network layer construction, security, application services, layer interfaces and etc. The ZigBee protocol stack operates as a very low-cost, very low-power-consumption, two-way, wireless communications standard and it is now establishing its place as an enabler for the wireless sensor networks because of these features [3].

As ZigBee is using the 2.4GHz Industrial, Scientific and Medical (ISM) unlicensed frequency band, coexistence issues arise as there are also other wireless technologies sharing the same band, such as 802.11b/g WiFi, Bluetooth, cordless phones and even microwave ovens. Due to the low transmission power, ZigBee is potentially vulnerable to the interference introduced by these wireless technologies rather than *vice versa*. It will undergo poor robustness and performance in the presence of heavy interference, and therefore it is desirable to improve ZigBee networks robustness. As WiFi is widely deployed and often collocated with ZigBee networks in applications, such as hospitals and home buildings, we take WiFi as the main interference source in this thesis work and try to find solutions to enhance the robustness of ZigBee networks under interference.

A feature called frequency agility is specified in the ZigBee standard [3] to improve the robustness of ZigBee networks. According to this feature, if interference is detected and reported in the current channel, a ZigBee network may move to a clear channel based on some mechanisms. We find out, however, some inadequacies in the standard that needs to be improved before the frequency agility can function well in practice as it is supposed to do. These inadequacies include the long response time

to interference, the undetermined channel scan duration, the potential ACK packet lost, etc. In our thesis, we focus on solving the long response time problem and propose a periodical window method which can dramatically improve the response time to interference.

Besides, in case that there is only a part of the whole network suffering from some local interference, it may neither be necessary for the whole network to move to a new idle channel because this movement is costly and risky, nor possible to find an idle channel for the whole network. Therefore, we extend the frequency agility function by enabling a single ZigBee network to work on multiple channels, which is called a multi-channel solution in this thesis. As the interference appears, the part of the network which is under the interference can move to a new idle channel while maintaining the communication links with the other part of the network which stays on the original channel. The moved part can move back to the original channel as the interference disappears. OPNET simulations show that our multi-channel solution can significantly improve the robustness of ZigBee networks in a cost-efficient way.

The goal of this thesis project is to present coexistence issues between ZigBee and WiFi and to find solutions. Furthermore, we will propose and investigate new solutions for improving ZigBee robustness and therefore performance in terms of interference.

The structure of this thesis will go on as follows. Chapter 2 will give an overview introduction to IEEE 802.15.4 and ZigBee standards, as well as a discussion about wireless sensor networks. In Chapter 3, an introduction to IEEE 802.11b/g WiFi will be given and the coexistence issues between ZigBee and WiFi will be clearly presented. Furthermore, an existing solution called fake CTS packets solution will be described. Advantages and disadvantages about this solution will be discussed. Then in Chapter 4, an introduction to the frequency agility feature will be given. Meanwhile inadequacies in the standard as well as our improvement will be presented. After that in Chapter 5, a multi-channel solution for improving ZigBee robustness and performance will be presented and discussed. Finally in Chapter 6, the thesis will be concluded.

Chapter 2 ZigBee and wireless sensor networks

In the previous chapter, an overall introduction to the thesis background and layout were given. In this chapter, section 2.1 will introduce the IEEE 802.15.4 and ZigBee standards. In section 2.2, wireless sensor networks and their applications will be described.

2.1 IEEE 802.15.4 and ZigBee

2.1.1 IEEE 802.15.4 overview

The IEEE 802.15.4 standard is finalized by the Institute of Electrical and Electronics Engineers (IEEE). It defines the protocol and compatible interconnection for data communication devices using low-data-rate, low-power, and low-complexity short-range radio frequency transmissions in a wireless personal area network (WPAN) [5]. The IEEE 802.15.4 standard builds up its architecture based on open systems interconnection (OSI) seven-layer model and defines the physical layer (PHY), the Medium Access Control (MAC) sublayer and also layer interfaces.

The IEEE 802.15.4 standard adopts a wideband physical layer using a Direct Sequence Spread Spectrum technique (DSSS). The standard provides specifications for operating in three different frequency bands [6]:

- 868MHz in Europe
- 915 MHz in North America
- 2.4GHz all over the world

Moreover, a frequency Division Multiplexing (FDM) approach is adopted in order to allow the coexistence of several networks in the same location. The channel allocation scheme is the following [6]:

- 1 channel (Channel 0) in the 868 MHz band
- 10 channels (Channel 1 - 10) in the 915MHz band
- 16 channels (Channel 11 - 26) in the 2.4GHz band

As the 2.4GHz band is the only one available worldwide, it is adopted by the IEEE 802.15.4 PHY radio frequency in most applications. In all the 16 channels of the 2.4GHz band, IEEE 802.15.4 defines that each channel has a 2 MHz bandwidth and a 5MHz channel spacing, which is illustrated in Figure 2.1. A parameter table of IEEE 802.15.4 in 2.4GHz band is also summed up in Table 2-1:

Table 2-1 ZigBee parameters in 2.4GHz frequency band

PHY (MHz)	Frequency (MHz)	Modulation	Bit rate (kb/s)	Symbols
2450	2400–2483.5	O-QPSK	250	16-ary Orthogonal

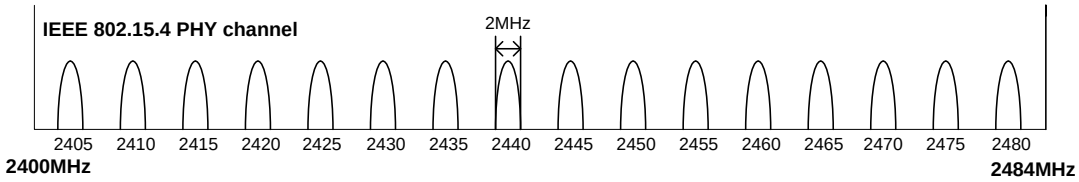


Figure 2.1 ZigBee channels in 2.4GHz band

The IEEE 802.15.4 standard defines that the PHY layer includes the following fundamental abilities and responsibilities [5]:

- Activation and deactivation of the radio transceiver
- Energy detection (ED) within the current channel
- Link quality indicator (LQI) for received packets
- Clear channel assessment (CCA) for carrier sense multiple access with collision avoidance (CSMA/CA)
- Channel frequency selection
- Data transmission and reception

In order to perform CCA, the PHY provides three methods which are listed below [5]:

- *Mode1:* Energy detection only (ED). CCA will report a busy medium status to MAC sublayer upon detecting energy exceeding a given ED threshold.
- *Mode2:* Carrier sense only (CS). CCA will report a busy medium upon detecting of a signal compliant with this standard with the same modulation and spreading characteristics of PHY.
- *Mode3:* Combination of *mode1* and *mode2*.

As mode1 with energy detection is simple to realize without any prior knowledge, it is becoming a popular method in clear channel assessment and adopted in this thesis project.

Besides the PHY layer, the IEEE 802.15.4 standard also defines the following responsibilities and tasks for the MAC sublayer [5]:

- Handle access to PHY
- Generating network beacons if the device is a coordinator
- Synchronizing to network beacons
- Supporting device security
- Employing the CSMA/CA mechanism for channel access
- Handling and maintaining the GTS mechanism
- Providing a reliable link between two peer MAC entities

2.1.2 ZigBee overview

The ZigBee protocol suit is standardized by ZigBee Alliance, an association of companies working together to enable reliable, cost-effective, low-power, wirelessly networked, monitoring and control products based on an open global standard [2] and it is based on IEEE 802.15.4 standard. In the protocol, ZigBee Alliance provides the network layer and the framework for the application layer to build up the ZigBee protocol. The application layer framework consists of the application support sublayer (APS) and the ZigBee device objects (ZDO) [3]. End manufacturers will define their own application objects on-demand, which will use the application layer framework and share APS and securities services with the ZDO. The general ZigBee stack architecture is illustrated in Figure 2.2.

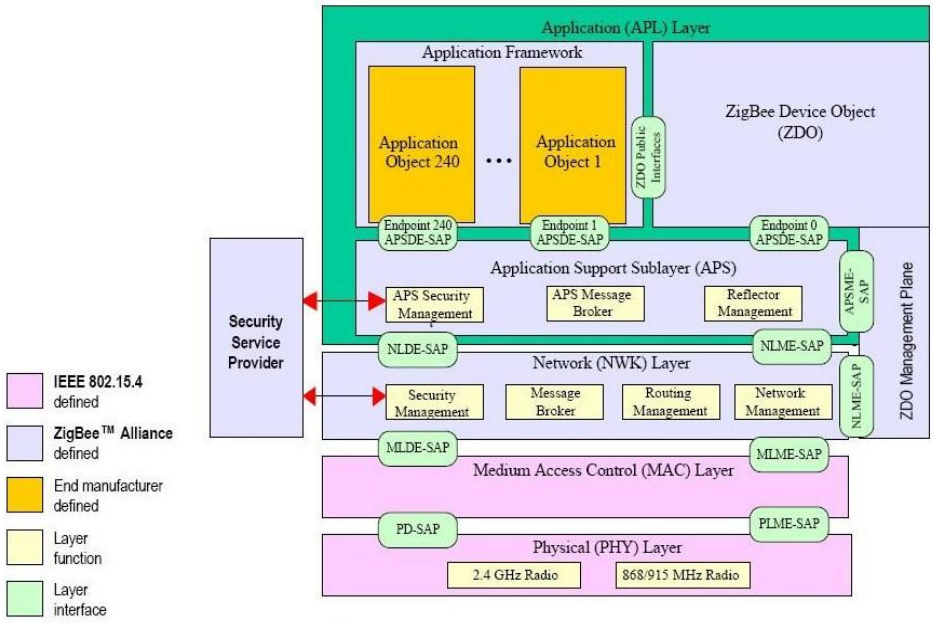


Figure 2.2 Outline of the ZigBee Stack Architecture
(Source: ZigBee Alliance**)**

Working on the above structures and architectures, three types of devices are defined in ZigBee:

- Personal Area Network (PAN) Coordinator: can only be a Full function device (FFD), which supports all the functions and features specified by ZigBee standard. It is responsible for managing the overall network.
- Router: can only be a FFD. A router can act as an intermediate router, passing on data from other devices.
- End node: can be a FFD or a Reduced Function device (RFD), which can only transmit or receive data without other functions, such as routing. RFD nodes have reduced functionality in order to minimize the complexity and the cost [7].

Medium access in a ZigBee network is based on a combination of random access and scheduled access [6]. Two modalities are available for medium access:

beacon-enabled and nonbeacon-enabled. In beacon-enabled modality, at least one device in the PAN will transmit beacon frames at a regular interval for synchronization. Normally the PAN coordinator takes this responsibility. While in nonbeacon-enabled modality, PAN does not contain any devices that transmit beacon frames at a regular interval. So there is no explicit synchronization provided in this modality. In our thesis, we focus on the popular nonbeacon-enabled modality which is more flexible and particularly suited for mesh topology introduced below.

In ZigBee networks, the protocol supports three types of topologies:

- *Star*: There are only two types of devices available in star topology, coordinator and end node. All end nodes are communicating with the coordinator directly. Star topology is simple, cost-efficient but vulnerable, it is highly dependent on the coordinator. If the coordinator fails, the whole network will collapse.
- *Mesh*: It supports full peer to peer communication and can only operate in non-beacon mode. Mesh topology is reliable and robust. It allows the network to self-heal after router failure. However, the disadvantage of mesh topology is a larger power consumption.
- *Tree*: It is the combination of Star and Mesh topology. Communication can be deployed in beacon mode. In tree topology, routers can go to sleep which save power consumption, but it does not support self-healing any more.

The topology illustrations are shown in Figure 2.3, 2.4 and 2.5.

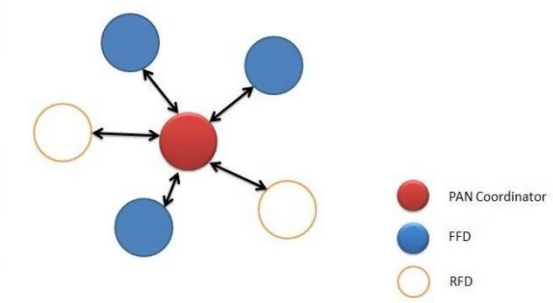


Figure 2.3 Example of Star topology

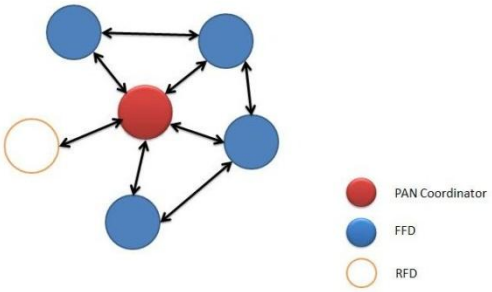


Figure 2.4 Example of Mesh topology

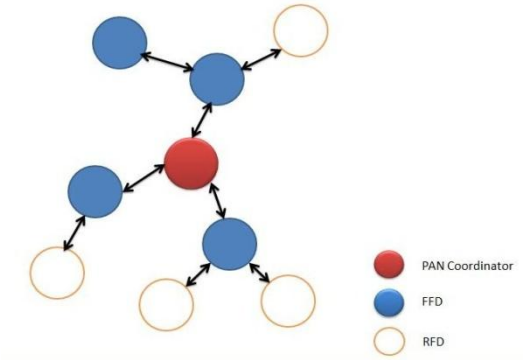


Figure 2.5 Example of Tree topology

2.2 Wireless sensor networks

As introduced in Chapter 1, wireless sensor networks are a type of wireless systems in which a large number of sensor nodes are deployed. These sensor nodes are becoming tiny, cheap and low power consumption due to recent technologies, which accelerate the development of wireless sensor networks as well. At present, wireless sensor networks are mainly using the ZigBee protocol to initialize and maintain the networks and accomplish data transmissions.

In wireless sensor networks, nodes usually collaborate to fulfill their tasks as a single node is incapable of doing so. Therefore, wireless sensor networks are powerful and amenable to support many different practical applications [8], such as intelligent buildings, facility management, precision agriculture, logistics, environment control, medical and health care, etc.

In order to handle these applications above, wireless sensor networks have to face many requirements, such as Quality of Service, fault tolerance, extended battery lifetime, scalability, wide range of densities and maintainability [8]. To realize these requirements and improve WSN robustness, new methods and mechanisms are being worked out.

Chapter 3 Coexistence issues

In Chapter 2, ZigBee protocol and wireless sensor networks are introduced. In this chapter, we will first present coexistence issues in the 2.4GHz band, and then discuss the coexistence issues between ZigBee and WiFi. After that, we will introduce an existing solution, the fake CTS solution, to the coexistence issues.

3.1 Investigations of coexistence issues

3.1.1 Coexistence issues background

The 2.4GHz Industrial, Scientific, and Medical (ISM) unlicensed frequency band is shared by many wireless technologies, such as IEEE 802.11b/g WiFi, ZigBee, Bluetooth, cordless phone, etc. Due to the low transmit power, ZigBee is potentially vulnerable to the interference introduced by these wireless technologies rather than *vice versa*. In practical applications, such as hospitals or home buildings, 802.11b/g WiFi (WiFi for short below in our thesis project, excluding IEEE 802.11a which is in 5GHz band) is widely deployed and often collocated with ZigBee networks. Meanwhile, comparing to Bluetooth and cordless phone, WiFi is usually working in long-duration once it starts, e.g. online video or large file transfer. Therefore, in this thesis work, we focus on the coexistence issues between ZigBee and WiFi.

WiFi standard defines 14 channels with 5MHz distance between two adjacent channels in the 2.4GHz band. The bandwidth of each channel is 22MHz. In Europe, only three non-overlapping channels 1, 7 and 13 can used concurrently [12]. Figure 3.1 shows the available WiFi channels in Europe and the overlapping channels between WiFi and ZigBee. As we can see, the channels of ZigBee and WiFi are mostly overlapped.

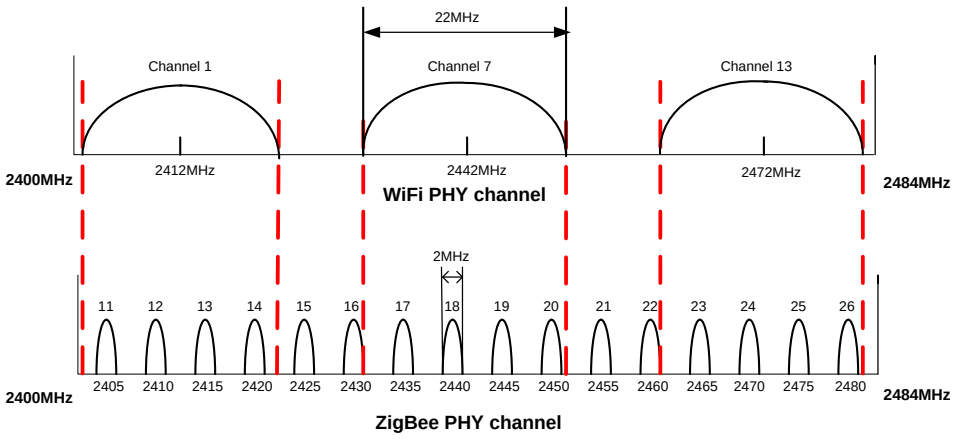


Figure 3.1 Channels of WiFi and ZigBee

In previous work [4] [9], a clear ZigBee and WiFi coexistence model is already presented. Taking into account the significant transmission power and the timing differences between ZigBee and WiFi, the coexistence model can be interpreted in two aspects, namely *power* and *timing*.

3.1.2 Power aspect in coexistence issues

The transmission powers of WiFi node and ZigBee node are significantly different. The typical power value for WiFi and ZigBee node are 100mw [10] and 1mw [5] respectively. Therefore, in the case of comparable CCA thresholds, the power difference will lead to three distinct regions, R_1, R_2 and R_3. In each of these regions, WiFi and ZigBee exhibit different interactive behavior and hence different performance.

- R_1: a region in which ZigBee node and WiFi node can sense each other
- R_2: a region in which ZigBee node can sense WiFi node rather than vice versa
- R_3: a region in which neither can sense the other, but ZigBee node could still suffer WiFi interference

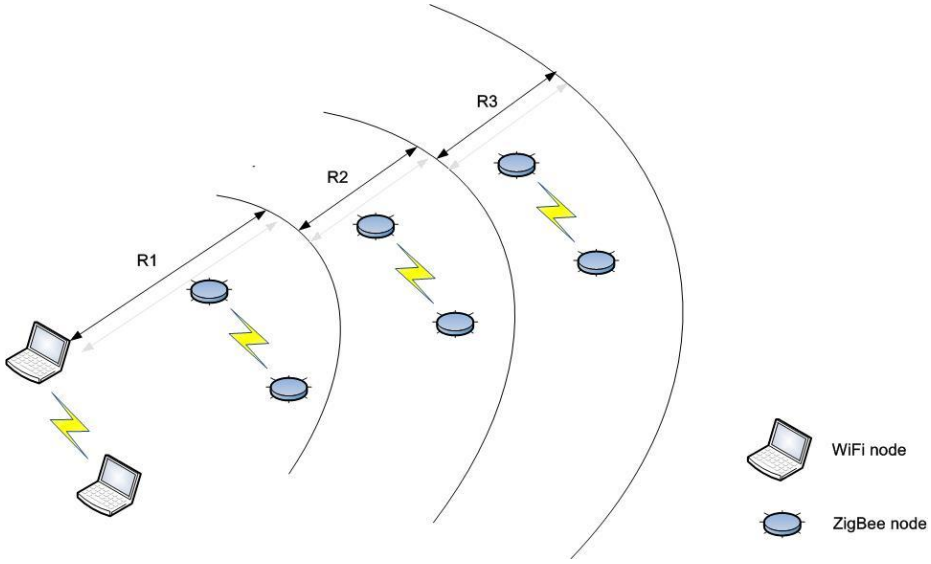


Figure 3.2 Coexistence model of ZigBee and WiFi

To quantify these regions, an indoor propagation model [11] [13] recommended by IEEE 802.11.2 specification is used. In this indoor propagation model, path loss follows Line Of Sight (LOS) free-space propagation up to 8m and then attenuates more rapidly with a coefficient of 3.3. According to a reliable transmission distance of IEEE 802.15.4 nodes reported in [14], the attenuating coefficient is adjusted to 4. Therefore, the path loss is expressed as below:

$$L(x)_{path\_loss} = \begin{cases} 20\log_{10}(\frac{4\pi x}{\lambda}), x \le x_0 \\ 20\log_{10}(\frac{4\pi x_0}{\lambda}) + 40\log_{10}(\frac{x}{x_0}), x > x_0 \end{cases} \quad (3\text{-}1)$$

Where x is the distance between a transmitter and a receiver and x_0 is LOS distance 8m, λ is the wavelength. By taking 6dB SIR at receiver and other parameters listed in Table 3-1, R_1, R_2 and R_3 are obtained and listed in Table 3-2.

Table 3-1 System parameters of ZigBee and WiFi

	IEEE 802.15.4	IEEE 802.11b	IEEE 802.11g
Transmit Power	0 dBm	20 dBm	20 dBm
Receiver Sensitivity	-85 dBm	-76 dBm	-82 dBm
Bandwidth	2 MHz	22 MHz	22 MHz
Transmit Rate	250 kbps	11 Mbps	6 Mbps
Backoff Unit T_{bs}	320 μs	20 μs	9 μs
SIFS	192 μs	10 μs	10 μs
DIFS	N/A	50 μs	28 μs
CCA Duration	128 μs	N/A	N/A
CW_{min}	7	31	15
Center Frequency	2410 MHz	2412 MHz	2412 MHz
Payload size	1 byte	1024 bytes	1024 bytes

Table 3-2 Coexistence regions of ZigBee and WiFi

Region	IEEE 802.11b	IEEE 802.11g
R_1	32 m	32 m
R_2	67 m	67 m
R_3	95 m	95 m

3.1.3 Timing aspect in coexistence issues

In the timing aspect, WiFi nodes have a priority over ZigBee nodes to access the channel, because:

(1) From the above Table 3-1, WiFi nodes have a much shorter timing than ZigBee nodes, e.g. the backoff unit is 320μs, 20μs and 9μs for IEEE 802.15.4, IEEE 802.11b and IEEE 802.11g respectively. The shorter timing gives WiFi nodes shorter waiting and backoff time and therefore a priority over ZigBee nodes.

(2) CSMA/CA mechanisms in ZigBee and WiFi are significantly different:
WiFi nodes will first sense the channel for a DIFS interval. If busy, the node will defer its transmission. When the channel is sensed idle, the node will uniformly generate a random backoff value from contention window (CW). This backoff counter will decrease by one as long as the channel is sensed idle for a backoff time unit. If a transmission is detected during this backoff, the backoff counter will become frozen, and it will resume after the channel is sensed one DIFS idle again. When the backoff counter becomes zero, the data packet is transmitted. After receiving a packet correctly, the receiver will immediately send back an ACK to the source after one SIFS. CW will remain the same when the channel is busy, it will

only double its size when ACK is not received.

ZigBee nodes, however, will not sense the channel during a backoff period. Instead, sensing is only done in a Clear Channel Assessment (CCA) period. Furthermore, whenever the channel is determined busy, the CW size will be doubled till the maximum size 32.

The above two reasons explain the channel sharing situation of ZigBee and WiFi. If heavy WiFi traffic is assumed, in the region R_1 where ZigBee and WiFi nodes can sense each other, WiFi nodes have a channel access priority over ZigBee nodes. This is shown in Figure 3.3. In the region R_2 where WiFi nodes cannot sense ZigBee nodes, the blind transmission of WiFi packets will lead to overlapping in packet transmission and destroy the ZigBee packet, which is illustrated in Figure 3.4.

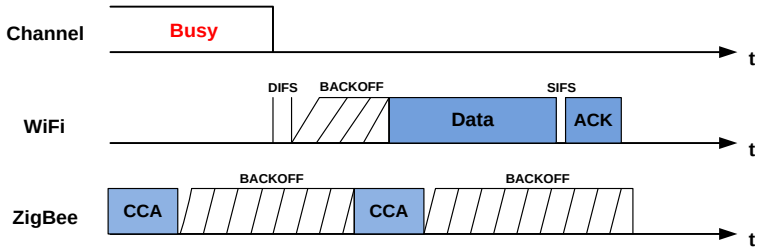


Figure 3.3 WiFi have priority over ZigBee in R_1

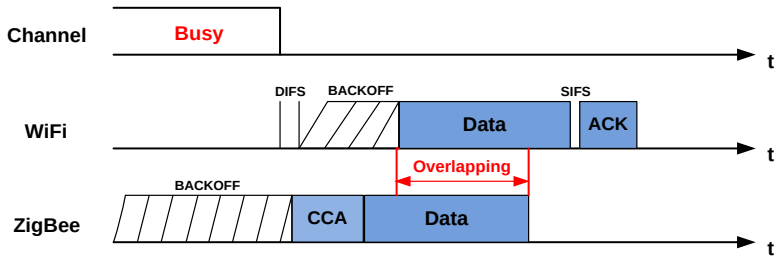


Figure 3.4 Collision when WiFi cannot sense ZigBee in R_2

3.2 Fake CTS solution to coexistence issues

As mentioned above, the ZigBee and WiFi coexistence issue is mainly caused by the different characteristics of IEEE 802.15.4 and IEEE 802.11b/g. If WiFi interference is heavy, the performance of ZigBee networks will dramatically degrade. Therefore, a coexistence method [14] was proposed at Philips Research to improve the ZigBee robustness. In this method, WiFi traffic can be stopped for a certain period of time rather than continuously occupying the channel in order to allow ZigBee nodes to transmit. The WiFi traffic is paused when they receive a "*fake CTS*" frame from a controlling node. The function of generating fake CTS frames is integrated into a ZigBee coordinator.

Every WiFi node has a Virtual Carrier Sense indicator called Network Allocation Vector (NAV) which can reserve the channel for pending packet. A station willing to transmit a packet first transmits a short control packet called Request To Send (RTS). The destination node will respond a control packet called Clear To Send (CTS) packet if the channel is idle. All other stations receiving either RTS or CTS will set their NAV for certain duration which is indicated in RTS or CTS. This mechanism guarantees that all the stations in the area reserve the channel for the pending packet. A simple illustration is shown below in Figure 3.5.

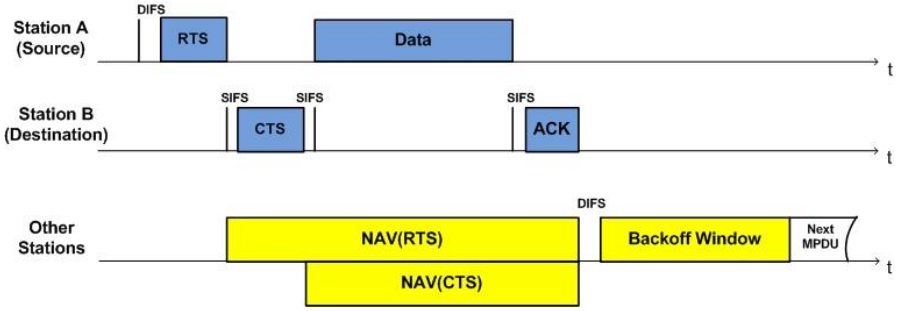


Figure 3.5 RTS/CTS mechanism

Comparing with RTS frame, a CTS frame does not need response after transmission and it also has a shorter length than that of an RTS frame. Consequently, the proposed method is using fake CTS frames to pause WiFi traffic periodically. The key point is that after generating a fake CTS frame, the source node will not transmit the data packet. Thus, other WiFi nodes who receive the fake CTS frame will reserve the channel for the presumptive data packet. This reserved idle time interval on the channel can therefore be seized by ZigBee nodes for transmission. In this way, the ZigBee performance can be improved.

In implementation, the fake CTS function is integrated into a ZigBee coordinator, which is designed as a hub to realize the coexistence method between WiFi and ZigBee. The hub has both WiFi and ZigBee sides in itself. The WiFi side has the function of generating fake CTS frames to silence other WiFi traffic for designated duration. The ZigBee side is a normal IEEE 802.15.4 coordinator which is capable of transmitting and receiving ZigBee packets.

Measurements after implementation prove that this method can dramatically improve the ZigBee performance. Figure 3.6 shows the format of fake CTS period (*P*) and reserved idle window (*W*). And illustrations of the *P*, *W* and ZigBee packet loss ratio are shown in Figure 3.7 and 3.8. It is clear that the ZigBee performance is getting better with a high window-to-period ratio (*WPR*).

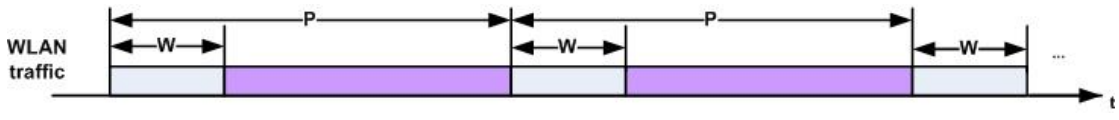


Figure 3.6 Format of fake CTS period (*P*) and reserved idle window (*W*)

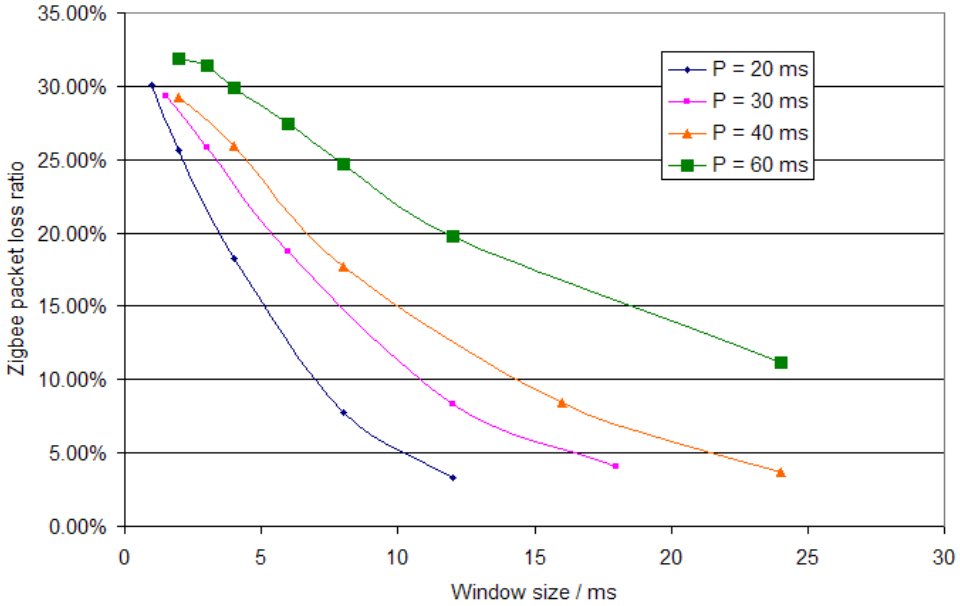


Figure 3.7 The relations among *P*, *W* and ZigBee packet loss ratio

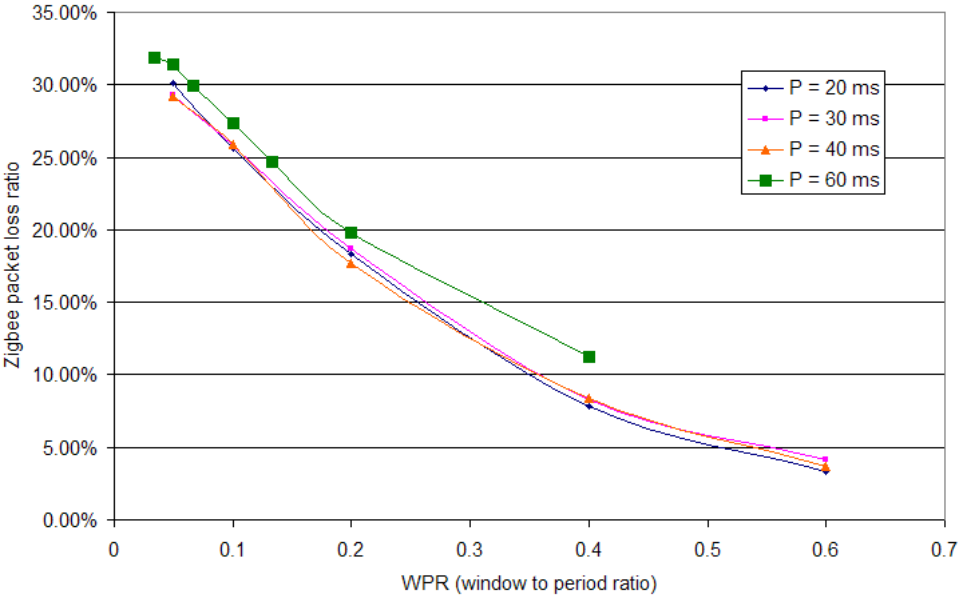


Figure 3.8 ZigBee packet loss ratio and WPR (window-to-period ratio)

3.3 Discussions

The fake CTS method can dramatically improve the ZigBee performance. However, it also has obvious disadvantages. In the first aspect, the channel will be periodically reserved by fake CTS frames in this method, which will degrade the WiFi traffic and performance. So the improvement of ZigBee performance can be seen as the sacrifice of WiFi performance. In the second aspect, the practical ZigBee applications are usually in very low traffic. Therefore, a large idle-to-period ratio value and is wasting channel resource. Furthermore in the third aspect, the coordinator will

consume much energy for transmitting fake CTS frames periodically. When the coordinator is battery support as usual, potential risk is lead to the ZigBee network as the coordinator is playing the key role in this network. Based on these three considerations, we want to work out a better solution which can improve ZigBee performance and robustness properly, as well as guarantee WiFi performance. Meanwhile, this method should also avoid the resource wasting problem in the fake CTS method.

The fake CTS method is illustrated in Figure 3.9 which focuses on solving coexistence issue in the time domain. In the next chapter, a new solution based on frequency agility (which is introduced in ZigBee Specification 2007 [3]) will be introduced. This new solution will solve the coexistence issue in the frequency domain which is illustrated in Figure 3.10.

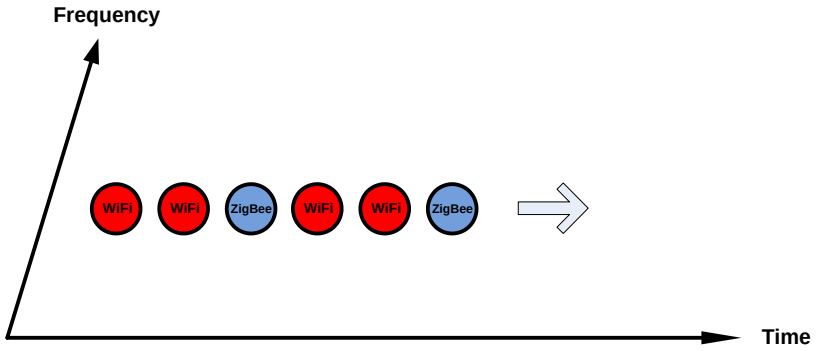


Figure 3.9 Coexistence solution in time domain

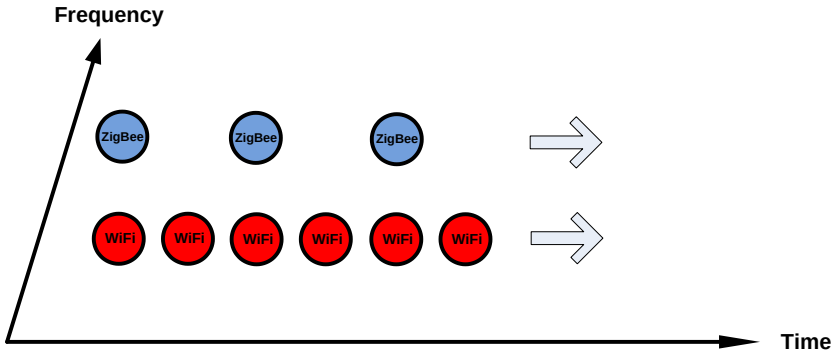


Figure 3.10 Coexistence solution in frequency domain

Chapter 4 Frequency agility solution

In the previous chapter, we introduce the coexistence issues, including the background and the existing solution for improve ZigBee robustness. However, due to the drawbacks of existing solution which have been discussed in section 3.3, a better solution is desirable.

To improve the robustness of ZigBee networks, a feature called frequency agility is specified in the ZigBee standard [3]. It focuses on solving the coexistence issues by moving the ZigBee network to a new clear channel when the current channel is under severe interference. However, we find out some inadequacies in the standard which are needed to be improved before the frequency agility can function well in practice as it is supposed to do. Therefore, in section 4.1, we will first introduce frequency agility, which is specified in ZigBee specification 2007. After that in section 4.2, we will list and discuss the inadequacies of frequency agility. Finally in section 4.3, we propose a new periodical window method to improve the long response time to interference. Comparisons between this method and the one in the standard are given.

4.1 Frequency agility overview

Operating in the 2.4GHz band, ZigBee is aided by the choice of 16 available channels. Therefore, it is reasonable to accept the fact that the ultimate feature to mitigate interference is the ability to move a ZigBee network to an idle channel while the current channel is under interference [15]. This method is called frequency agility and is specified in the ZigBee specification 2007 [3]. It is worth noting that frequency agility is totally different from the frequency hopping. In case where the interference detected, ZigBee devices can scan for a better channel and move the whole network to the new channel which allows the network to adapt over time to changing RF environments. This operation is done under the direction of network manager.

According to the ZigBee specification [3], in frequency agility, a device will become the Network Channel Manager. Once interference is detected, it will act as the key role for receiving the interference reports and changing the network channel if necessary. The default network channel manager is the PAN coordinator. However it can also be another node by sending a *Mgmt_NWK_Update_req* update. In the frequency agility operation, the network coordinator and each router are responsible for tracking transmit failures and total transmissions. The failure number will be recorded in the *TransmitFailure* field in the neighbor table, while the total transmission attempts are kept in the NIB (Network layer information base) counter. The flowchart of tracking

behavior can be illustrated as following in Figure 4.1.

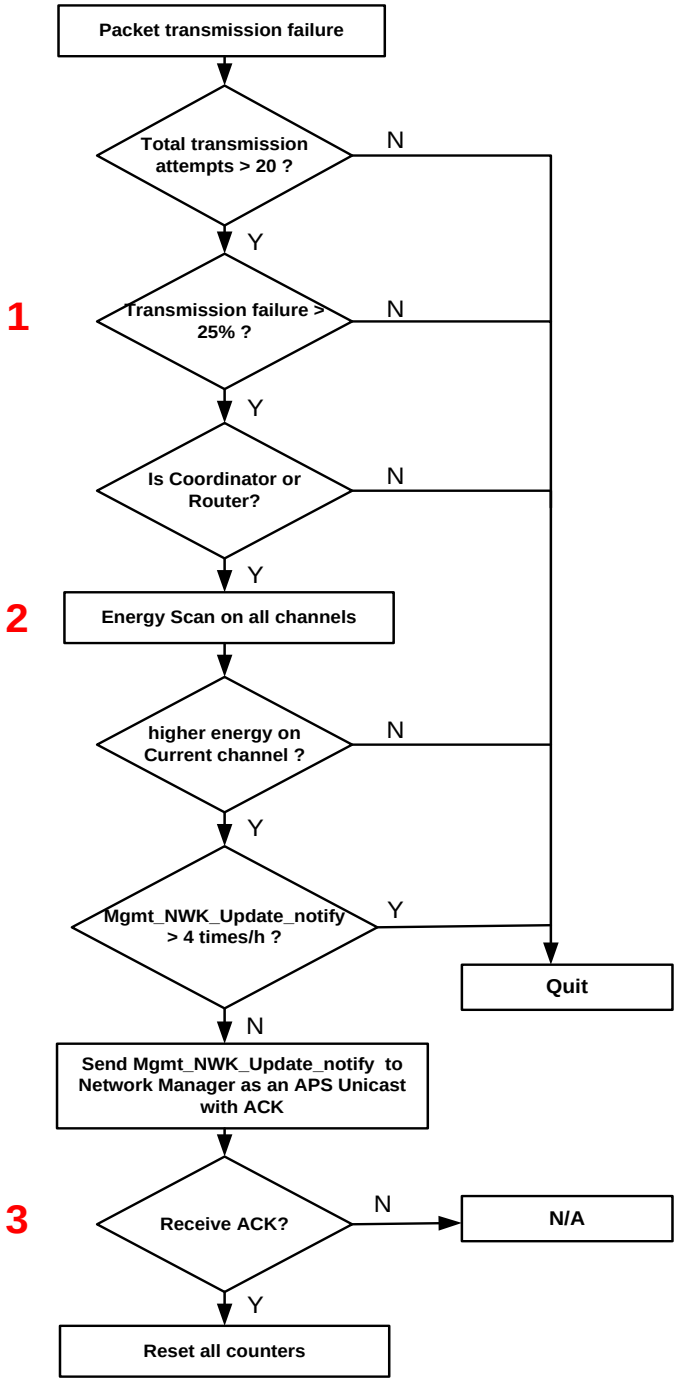


Figure 4.1 Tracking and Reporting Behaviors in ZigBee frequency agility

As described in the above figure, coordinator or router will notify the interference to the Network Manager with an *Mgmt_NWK_Update_notify* packet if all the requirements above are fulfilled. Upon receipt of an unsolicited notification, the network manager will evaluate if a channel change is required in the network. In specification, the recommended procedures are given as below in Figure 4.2.

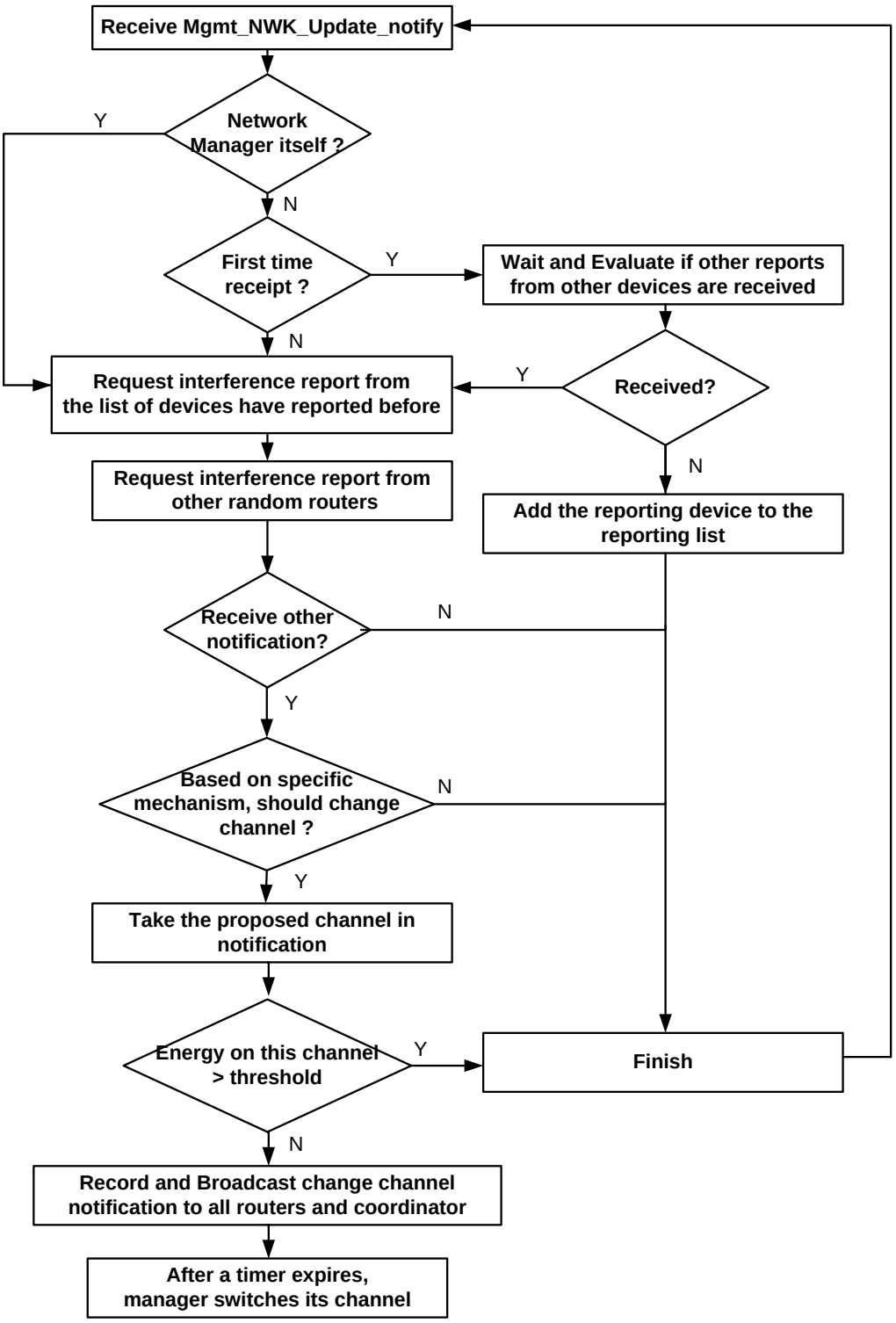


Figure 4.2 Network Channel Manager Behavior in frequency agility

Based on above procedures, when the network channel manager receives interference reports, it will determine whether to move the whole network to a new channel based on some mechanisms. If it decides to move, the manager will direct the whole network to leave the current operating channel and move to a new one

based on the procedures in the flowchart.

4.2 Inadequacies in frequency agility

Frequency agility is supposed to mitigate the interference issues. However, we find that some inadequacies are needed to be improved before the frequency agility can function well in practice. If not, these inadequacies will result in low efficiency and severe robustness issues. Moreover, these inadequacies are mainly in the tracking and reporting behaviors of the frequency agility. We find out and list three inadequacies below, which are marked in Figure 4.1 as well.

4.2.1 Response time to interference

As marked number one in Figure 4.1, the device will report interference to network manager when transmit failures exceed 25% after 20 packets have been sent. The transmit failure ratio is calculated in formula (4-1) as below and compared with the threshold 25%.

$$r_{failure} = \frac{\text{Transmission failure number}}{\text{Total transmission attempts}} \tag{4-1}$$

According to the specification, *TransmitFailure* field in the neighbor table is used to record the transmission failure number. The maximal available value of the transmission failure number is 0xff, which is 255 in decimal system. Once the number rolls over past 255, it will be reset back to 0. On the other hand, the total transmission attempts are counted by “nwkTxTotal”, as a NIB attribute. Once a packet is generated in NWK layer, the nwkTxTotal counter will increment by one. The maximum available value of the total transmission attempt is 0xffff, which is 65535 in decimal system. When the value rolls over past 65535, the NWK layer will reset the nwkTxTotal counter back to 0, as well as the failure number in the neighbor table. These two parameters are summed up in Table (4-1).

Table 4-1 Parameters in tracking interference

Parameter	Counter	Maximal value
Transmission failure number	*TransmitFailure* Field	255
Total transmission attempt	nwkTxTotal	65535

However, the method of tracking interference does not work well in practice. The maximal available values of total transmission attempt and transmission failure number have big difference, which result in the 25% threshold can hardly be reached in local counting period. For instance, if the interference starts after 30000 packets have been sent and we assume as much as ten neighbors and all the packets are lost

after interference starts, when all these ten nodes reach to maximum transmission failure number 255, the failure ratio is only 7.83% as calculated in formula (4-2), which is far less than 25%. Therefore, the interference cannot be detected and report before the counters are reset to 0.

$$r_{transmit\_failure} = \frac{255*10}{30000+255*10} = 7.83\% \quad (4\text{-}2)$$

In practical ZigBee applications, the traffic intensity is low. Assume the traffic intensity is 5pkt/s, the rest time before counters reset will be around 2 hours based on (4-3). This response time is definitely too large and not acceptable.

$$t_{rest} = \frac{(65535-30000)}{5\,pkt/s} = 7107s \quad (4\text{-}3)$$

In order to fix this problem, we propose a periodical window method, which help the node to detect and report interference timely and surely. This periodical window method will be explained in detail in section 4.3.

4.2.2 Channel scan duration

As marked number two in the Figure 4.1, the second inadequacy is about the energy scan duration spending on all channels. According to the ZigBee specification, the scan duration of conducting energy scan on each channel is calculated as following in formula (4-4), where the valid range of *n* is from 0 to 14.

$$Scan\_Duration = aBaseSuperframeDuration*(2^n+1)\ \ symbols \quad (4\text{-}4)$$

Suppose that *aBaseSuperframeDuration* is the default 960 symbols, the total scan duration is therefore ranging from 0.492 to 4026.788 seconds depending on different *n* value [16]. Apparently, the scan durations have large differences. Parameter Choosing is a trade-off, large scan duration is not acceptable for interference detection and report, while a small duration cannot guarantee the assessment accuracy.

The standard does not supply an effective choosing way, decisions should be made upon different practical implementations. Or some methods that help to subtract the total scan duration can be considered [16].

4.2.3 ACK packet lost

After transmit the unicast *Mgmt_NWK_Update_notify* packet, an ACK response from the network channel manager is expected as marked number three in Figure 4.1. Due to the existing interference, both the *Mgmt_NWK_Update_notify* and ACK packets could fail to get access to the channel and finally dropped.

From previous calculation and simulation [4], it is clear that the ZigBee nodes always have the chance to access channel in the region R_1, even WiFi traffic is saturated. This situation is described in Figure 4.3. Based on studies in [4] and the system parameters in table 3-1, the ZigBee nodes have the different probability to access the channel under different WiFi payload length in saturated WiFi mode, which is illustrated in Figure 4.4. Even in the worst case that WiFi has the 1500 bytes packet length, the ZigBee packet channel access probability is still around 38.5%. Therefore, it shows that the ACK packet is expected to be received finally after repeating several times.

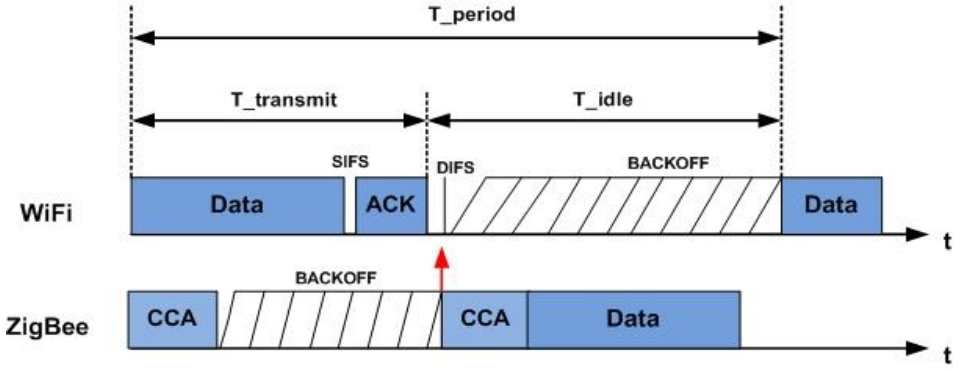


Figure 4.3 ZigBee channel access in Time aspect

Figure 4.4 ZigBee packet loss ratio under saturated WiFi traffic (from [4])

However in the region R_2, both the *Mgmt_NWK_Update_notify* and ACK packet could be corrupted after sending out, because WiFi cannot sense ZigBee and is doing blind transmission. This is also desired to be improved by repeating packet transmission for several times as this will improve the possibility of receiving packets correctly in some

cases, e.g. the packet is partly overlapped, but the transmitted packet may be received successfully due to a sufficient SIR at receivers.

4.3 Better periodical window method

4.3.1 General procedure

As mentioned in section 4.2.1, the existing tracking method in the frequency agility will probably result in very long response time, during which the node will suffer severe interference. Therefore, a periodical window method is proposed to help the ZigBee node to detect and report interference timely and surely. The parameters in this method are listed in Table (4-2).

Table 4-2 Parameters in periodical window method

Parameter	Explanations
d	Periodical window size (packet)
N_{TX}	Total TX number in d (packet)
N_{fail}	Total failure number in d (packet)
α	Transmit failure ratio to trigger frequency agility (%)
N_0	Transmit failure number to trigger frequency agility (packet)

In this method, each periodical window is built up by *d* packets and followed by another periodical window. Therefore, all ZigBee packets are divided into different periodical windows and the tracking and calculation are based on each periodical window period.

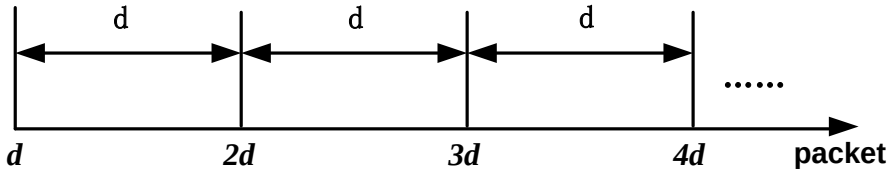


Figure 4.5 Periodical window period

Within each periodical window period, N_{TX} is the total transmit number and N_{fail} is used for counting the total failure number. Once generating a packet, N_{TX} will increment by one. If one generated packet fails to transmit, N_{fail} will add by one.

N_0 is the packet failure number to trigger frequency agility in each periodical window period. The value of N_0 is based on the threshold α and window size d. The calculation is presented below in formula (4-5):

$$N_0 = \lceil d * \alpha \rceil \quad (4\text{-}5)$$

In each periodical window, once N_{fail} equals to N_0, the frequency agility will be triggered. If frequency agility is not triggered and N_{TX} rolls over past d, the next periodical window starts and both two counters N_{TX} and N_{fail} will be reset back to 0. This whole procedure can be described as below in Figure 4.6.

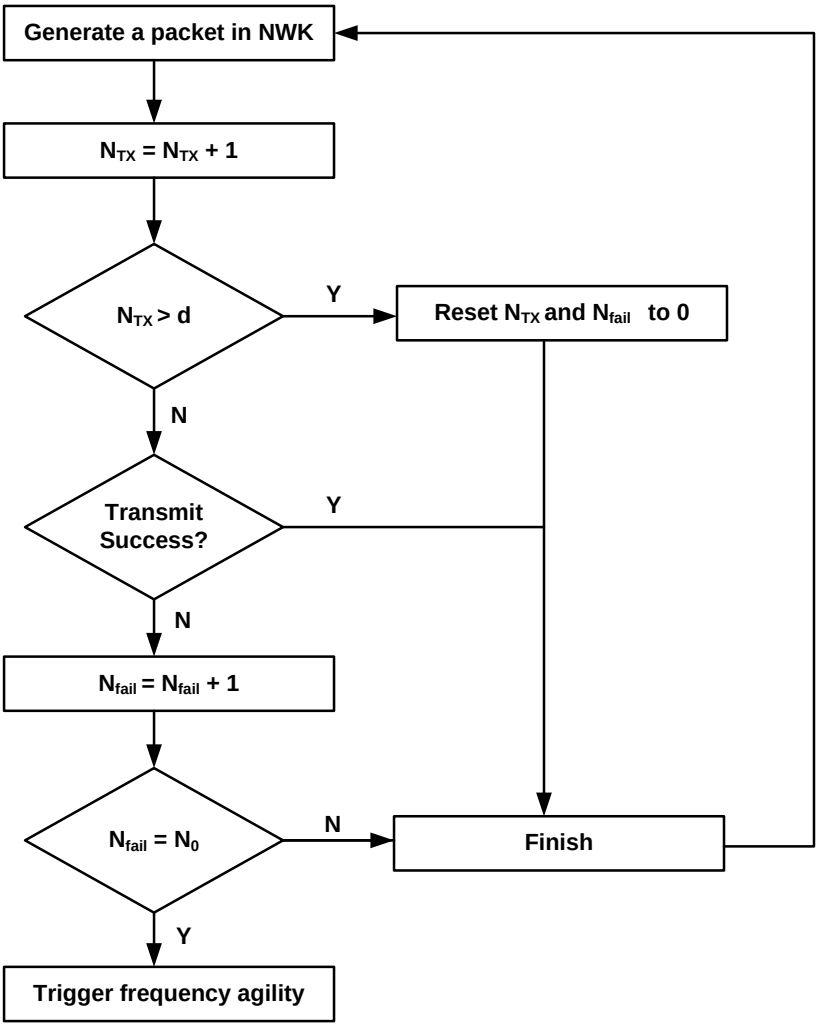


Figure 4.6 Procedure of periodical window method

The remarkable contrast between this periodical window method and the original one is that the periodical window method dramatically shortens the interference response time. In the method without periodical window, inadequacy is mainly caused by the large difference in maximal available value between nwkTxTotal and *TransmitFailure* field, which are 65535 and 255 respectively. However in periodical window method, a proper window size and is used to achieve an improvement in response time.

In the following sections, the response time of methods without and with periodical window will be calculated and compared. Due to the typical ZigBee network applications and their low duty cycle properties, we assume that the ZigBee traffic rate is constant. Meanwhile, we assume only one pair of ZigBee nodes exists in the calculation for simplicity and WiFi interference is heavy enough to trigger frequency

agility.

4.3.2 Method 1: without periodical window

In the method without periodical window, parameters are listed and illustrated in Table (4-3) and Figure 4.7 respectively.

Table 4-3 parameters in periodical window calculation

Parameter	Explanations
N_{fail}	Failure number in Neighbor Table (packet)
N_{TX}	nwkTxTotal (packet)
N_0	Transmit failure number to trigger frequency agility (packet)
t_0	ZigBee traffic start time (s)
t_i	N_{fail} and N_{TX} reset time (s)
t_s	WiFi start time (s)
R	ZigBee traffic rate (packet/s)
η	ZigBee packet failure ratio due to WiFi interference (%)
α	Transmit failure ratio to trigger frequency agility (default 25%)
$T(t_s)$	Response time after WiFi starts (s)

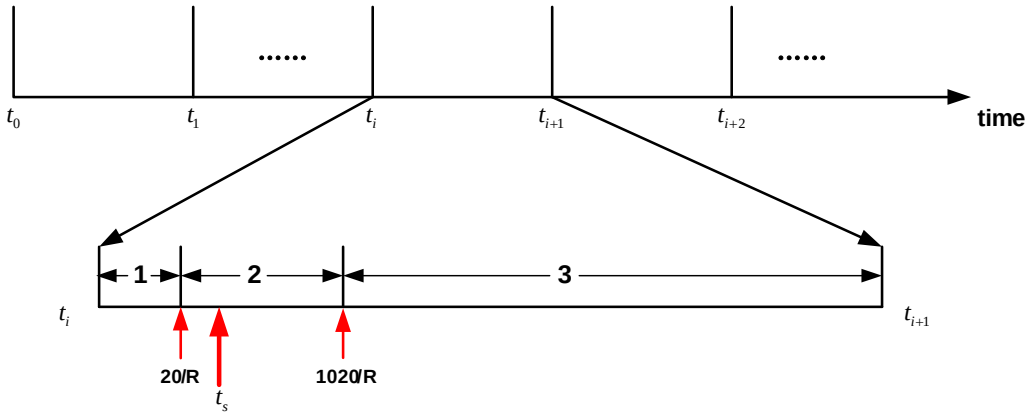


Figure 4.7 Method without periodical window in time aspect

As the maximum available value of N_{TX} is 65535, the failure packet tracking is done periodically. We define the time for sending every 65535 packets as a period, which therefore depends on the ZigBee traffic rate R. In each period from t_i to t_{i+1}, N_{TX} is increased by one when a packet is generated in NWK layer, and N_{fail} is increment by one when a packet dropped due to channel busy.

After interference is introduced at t_s, N_0 is the needed packet failure number to trigger frequency agility *and* $T(t_s)$ is the response time from interference starts till frequency

agility is triggered. In this method, N_0 is not a constant value and is calculated by α and N_{TX}. The calculation is presented below in formula (4-6):

$$N_0 = \lceil \mathrm{N_{TX}} * \alpha \rceil \tag{4-6}$$

Once N_{fail} equals to N_0 in local period, the frequency agility will be triggered. If frequency agility is not triggered in this local period and N_{TX} rolls over past 65535, the next window period starts and both two counters N_{TX} and N_{fail} will be reset back to 0.

According to the standard, as shown in Figure 4.1, the failure number N_{fail} will be counted only after 20 ZigBee packets have been sent, even if the interference appears earlier in local period. On the other hand, due to the maximal available values 255 and 65535 in *TransmitFailure* filed and total transmission attempts respectively, the frequency agility will not be triggered if N_{TX} is already larger than 765 in the local period when the interference starts. Because even all the 255 packets are lost after interference starts, packet failure ratio is still smaller than threshold α according to formula (4-7).

$$\frac{255}{765+255} = \frac{255}{1020} \leq \alpha \tag{4-7}$$

Consequently, there are three different scenarios when calculating the response time. They are explained below and presented in Figure 4.8.

(i) Interference appears before 20 ZigBee packets have been sent (Sub-period 1). The frequency agility will be triggered in local period.

(ii) Interference appears after 20 ZigBee packets, but before 765 packets (Sub-period 2). The frequency agility may be triggered in either the current or the next period.

(iii) Interference appears after 765 ZigBee packets have been sent (Sub-period 3). The frequency agility can only be triggered in the next period.

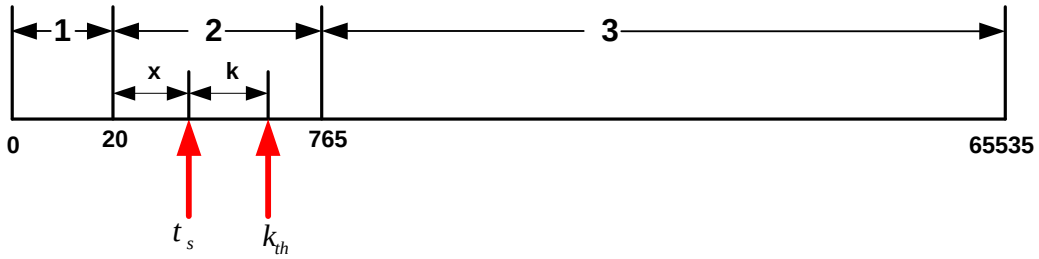


Figure 4.8 Packet relationships in one period

In all three scenarios, scenario (ii) has two possibilities and it is more complicated than others. So we first analyze and calculate the response time in scenario (ii), and in scenario (i) and (iii) after that.

In scenario (ii):

(1) frequency agility is trigged in the current period:

In Figure 4.8, when interference appears in the Sub-period 2, assume (20+*x*) packets have been transmitted and the frequency agility will be triggered at the k_{th} packet after interference appears.

The valid range of *k* depends on two aspects:

(a) If all the packets are lost after interference starts, N_1 packets are needed to trigger frequency agility, where

$$\frac{N_1}{20+x+N_1} > 25\% \tag{4-8}$$

(b) According to (4-9), if N_{TX} equals to 1020 before α is fulfilled, the frequency agility cannot be triggered in local window even with maximal failure number 255. Therefore, there are only N_2 packets available for *k* after interference starts if interference is triggered in local period, which is showed in (4-12).

$$\frac{255}{1020} = \alpha = 25\% \tag{4-9}$$

Based on (a) and (b), the valid range of *k* falls in [*N1*, *N2]* if the frequency agility is triggered in local period, where

$$N_1 = \lceil \frac{20+x}{3} \rceil \tag{4-10}$$

$$N_2 = 1020 - (20+x) \tag{4-11}$$

The failure number N_0 to trigger frequency agility can be calculated below:

$$N_0 = \lceil (20+x+k) * \alpha \rceil \tag{4-12}$$

Denote *Y* as N_{fail} before the k_{th} packet (namely in former *(k-1)* packets), we have

$$Y = N_0 - 1 \tag{4-13}$$

The response time that frequency agility is triggered at the k_{th} packet is:

$$T(\mathrm{k}_{th}) = T(\text{response time in local period})$$

$$= \frac{k}{R} \tag{4-14}$$

We denote the probability that frequency agility is triggered at the k_{th} packet as :

$$g(\eta, Y, k) =$$

$$\Pr[\text{ Y packets fail in (k-1) packets }] \cap \Pr[\text{ other packets succeed in (k-1) packets }] \cap \Pr(\text{ } k_{th} \text{ packet fails })$$

$$= C_{k-1}^{Y} * \eta^{Y} * (1-\eta)^{(k-1)-Y} * \eta \tag{4-15}$$

In this scenario, the probility that frequency agility is triggered at the k_{th} packet is:

$$\mathrm{Pr}_k = g(\eta, Y, k) \tag{4-16}$$

Therefore, the expected response time is:

$$E[\mathrm{T}]_{ii\_local} = \sum_{k=N_1}^{N_2} [\mathrm{Pr}_k * T(\mathrm{k}_{th})] \tag{4-17}$$

Given interference starts in sub-period 2, the probability that frequency agility is triggered in local period will be:

$$\mathrm{Pr}_{ii\_local} = \sum_{k=N_1}^{N_2} (\mathrm{Pr}_k) \tag{4-18}$$

(2) frequency agility is trigged in the next period:

In this situation, the frequency agility will be triggered after 20 packets have been sent in next period. Similar to the calculation in (1), the valid range parameters are listed below with *x* equals to 0.

$$N_1 = \lceil \frac{20}{3} \rceil = 7 \tag{4-19}$$

$$N_2 = 1020 - 20 = 1000 \tag{4-20}$$

The response time that frequency agility is triggered at the k_{th} packet in the next period is: $T(\mathrm{k}_{th}) = T(\text{waiting time in local period}) + T(\text{response time in next period})$

$$= (t_i + \frac{65535}{\mathrm{R}} - \mathrm{t_s}) + (\frac{20k}{\mathrm{R}}) \tag{4-21}$$

The probability that frequency agility is triggered at the k_{th} packet is:

$$\mathrm{Pr}_k = g(\eta, Y, k) \tag{4-22}$$

Therefore, the expected response time when frequency agility is triggered in the next period will be:

$$E[\mathrm{T}]_{ii\_next} = \sum_{k=N_1}^{N_2} [\mathrm{Pr}_k * T(\mathrm{k}_{th})] \tag{4-23}$$

Given the WiFi interference starts in sub-period 2, the probability that frequency agility is triggered in the next period will be:

$$\mathrm{Pr}_{ii\_next} = 1 - \mathrm{Pr}_{ii\_local} \tag{4-24}$$

Therefore, considering possibilities (1) and (2) together, if the WiFi interference starts

in sub-period 2, the total expected response time is:

$$E[\mathrm{T}]_2 = \mathrm{Pr}_{ii\_next} * E[\mathrm{T}]_{ii\_next} + \mathrm{Pr}_{ii\_local} * E[\mathrm{T}]_{ii\_local} \quad (4\text{-}25)$$

In scenario (i):

Similar to the possibility (2) in scenario (ii), the valid range of *k* falls in [*N1*, *N2]* where:

$$N_1 = \lceil \frac{20}{3} \rceil = 7 \quad (4\text{-}26)$$

$$N_2 = 1020 - 20 = 1000 \quad (4\text{-}27)$$

The response time that frequency agility is triggered at the k_{th} packet is:

$$T(\mathrm{k}_{th}) = T(\text{waiting time till 20 packets}) + T(\text{response time in local period})$$

$$= (t_i + \frac{20}{R} - \mathrm{t_s}) + (\frac{k}{\mathrm{R}}) \quad (4\text{-}28)$$

The probability that frequency agility is triggered at the k_{th} packet is:

$$\mathrm{Pr}_k = g(\eta, Y, k) \quad (4\text{-}29)$$

Therefore, the expected response time is:

$$E[\mathrm{T}]_1 = \sum_{k=N_1}^{N_2} [\mathrm{Pr}_k * T(\mathrm{k}_{th})] \quad (4\text{-}30)$$

In scenario (iii):

In scenario (iii), the valid range of *k* falls in [*N1*, *N2]* where:

$$N_1 = \lceil \frac{20}{3} \rceil = 7 \quad (4\text{-}31)$$

$$N_2 = 1020 - 20 = 1000 \quad (4\text{-}32)$$

The response time that frequency agility is triggered at the k_{th} packet is:

$$T(\mathrm{k}_{th}) = T(\text{waiting time in local period}) + T(\text{response time in next period})$$

$$= (t_i + \frac{65535}{R} - \mathrm{t_s}) + (\frac{20}{R} + \frac{k}{\mathrm{R}}) \quad (4\text{-}33)$$

The probability that frequency agility is triggered at the k_{th} packet is:

$$\mathrm{Pr}_k = g(\eta, Y, k) \quad (4\text{-}34)$$

Therefore, the expected response time is:

$$E[\mathrm{T}]_3 = \sum_{k=N_1}^{N_2} [\mathrm{Pr}_k * T(\mathrm{k}_{th})] \tag{4-35}$$

To sum up three scenarios, the expected response time *E[T]* in the method without periodical window is:

$$\begin{cases} E[\mathrm{T}] = E[\mathrm{T}]_1 & \text{where interference starts in sub-period 1} \\ E[\mathrm{T}] = E[\mathrm{T}]_2 & \text{where interference starts in sub-period 2} \\ E[\mathrm{T}] = E[\mathrm{T}]_3 & \text{where interference starts in sub-period 3} \end{cases} \tag{4-36}$$

4.3.3 Method 2: with periodical window

In the method with periodical window, parameters are listed and illustrated in Table 4-4 and Figure 4.9 respectively.

Table 4-4 parameters in method with periodical window

Parameter	Explanations
$\mathbf{t_0}$	ZigBee traffic start time (s)
$\mathbf{t_i}$	N_{TX} and N_{fail} reset time (s)
$\mathbf{t_s}$	WiFi start time (s)
R	ZigBee traffic rate (packet/s)
ŋ	ZigBee packet failure ratio due to WiFi interference (%)
α	ZigBee packet failure ratio threshold to trigger frequency agility (%)
d	Periodical window size (packet)
$\mathbf{N_{TX}}$	Total TX number in periodical window (packet)
$\mathbf{N_{fail}}$	Failure number in window (packet)
$\mathbf{N_0}$	Transmit failure number threshold to trigger frequency agility (packet)
M	Remaining packet number in local period after WiFi starts (packet)
$\mathbf{T(t_s)}$	Response time after WiFi interference starts (s)

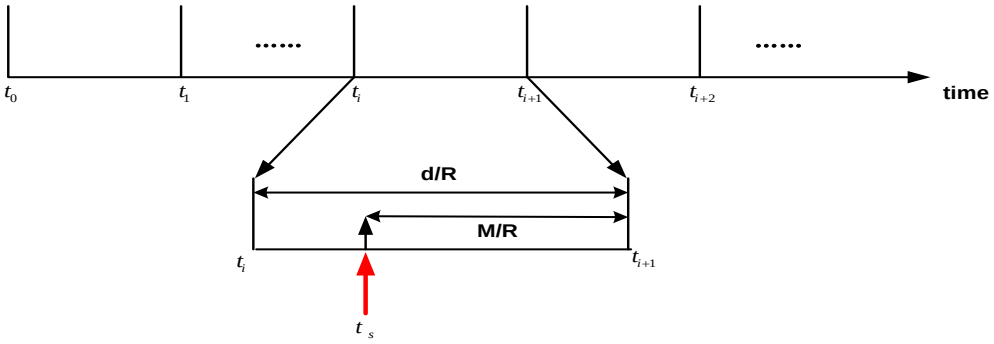


Figure 4.9 Method with periodical window in time aspect

In the periodical window method, failure packets tracking are done periodically in the time aspect. Every *d* packets builds up one period and the period duration in the time aspect depends on the traffic rate *R*. In each period from t_i to t_{i+1}, N_{TX} is increased by one when a packet is generated in NWK layer, and N_{fail} is increment by one when a packet is dropped due to a busy channel.

After interference is introduced at t_s, N_0 is the packet failure number to trigger frequency agility *and* $T(t_s)$ is the response time from when the interference starts till frequency agility is triggered. Different from the method without periodical window, N_0 is a constant value in this method and is calculated by α and *d*. The calculation is presented below in formula (4-37):

$$N_0 = \lceil d * \alpha \rceil \tag{4-37}$$

Once N_{fail} equals to N_0 in the local period, the frequency agility will be triggered. If frequency agility is not triggered in this local period and N_{TX} rolls over past *d*, the next periodical window starts and both two counters N_{TX} and N_{fail} will be reset back to 0.

Based on the parameters above, there are two scenarios in the periodical window method which are explained below and presented in Figure 4.10.

(i) $M \geq N_0$, the frequency agility is triggered either in local periodical window or next one (Sub-period 1).

(ii) $M < N_0$, the frequency agility can only be triggered in the next periodical window (Sub-period 2).

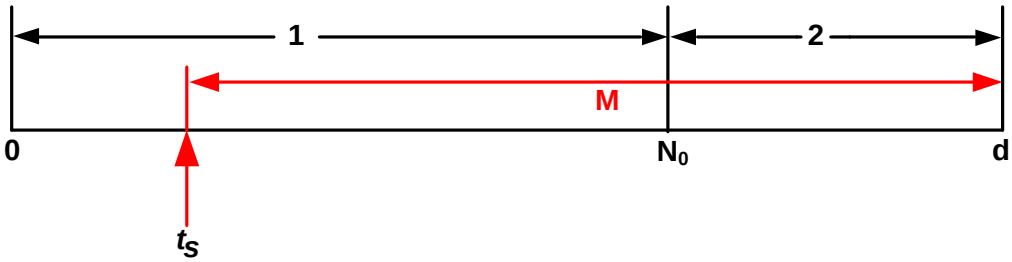


Figure 4.10 Packet relationships in one period

Like in the method without periodical window, scenario (i) has two possibilities and it is

more complicated. So we first analyze and calculate the response time in scenario (i), and in scenario (ii) after that.

In scenario (i):

(1) Frequency agility is triggered in the local window

Assume the frequency agility is triggered at the i_{th} packet in the local window after interference starts, the valid range of *i* falls in [*N1*, *N2]*. It means that at least *N1* packets are needed to trigger frequency agility and *N2* is the maximal available packet number in this local window for triggering frequency agility, where

$$N_1 = N_0 \tag{4-38}$$

$$N_2 = M \tag{4-39}$$

Denote *Y* as N_{fail} before the i_{th} packet (namely in former *(i-1)* packets), where

$$Y = N_0 - 1 \tag{4-40}$$

The response time that frequency agility is triggered at the i_{th} packet in local window is:

$$T(i_{th}) = T(\text{response time in local window})$$

$$= \frac{i}{\mathrm{R}} \tag{4-41}$$

The probability that frequency agility is triggered at the i_{th} packet in local window is:

$$\mathrm{Pr}_k = g(\eta, Y, k) \tag{4-42}$$

Therefore, the expected response time if frequency agility is triggered in local window will be:

$$E[\mathrm{T}]_{1\_local} = \sum_{i=N_1}^{N_2} [\mathrm{Pr}_i * T(i_{th})] \tag{4-43}$$

Given interference starts in sub-period 1, the probability that frequency agility is triggered in the local window will be:

$$\mathrm{Pr}_{1\_local} = \sum_{i=N_1}^{N_2} (\mathrm{Pr}_i) \tag{4-44}$$

(2) Frequency agility is triggered in next window

Assume the frequency agility is triggered at the i_{th} packet in next window after

interference starts, the valid range of *i* falls in [*N1*, *N2]*, where

$$N_1 = N_0 \tag{4-45}$$

$$N_2 = d \tag{4-46}$$

The response time that frequency agility is triggered at the i_{th} packet in the next window is:

$$T(i_{th}) = T(\text{waiting time in local window}) + T(\text{response time in next window})$$

$$= (t_i + \frac{d}{R} - t_s) + (\frac{i}{\mathrm{R}}) \tag{4-47}$$

The probability that frequency agility is triggered at the i_{th} packet in the next window is:

$$\mathrm{Pr}_i = g(\eta, Y, i) \tag{4-48}$$

Therefore, the expected response time if frequency agility is triggered in the next window will be:

$$E[\mathrm{T}]_{1\_next} = \sum_{i=N_1}^{N_2} [\mathrm{Pr}_i * T(i_{th})] \tag{4-49}$$

Given that the interference starts in sub-period 1, the probability that frequency agility is triggered in next window will be:

$$\mathrm{Pr}_{1\_next} = 1 - \mathrm{Pr}_{1\_local} \tag{4-50}$$

To sum up possibilities (1) and (2), if interference starts in sub-period 1, the total expected response time is:

$$E[\mathrm{T}]_1 = \mathrm{Pr}_{1\_local} * E[\mathrm{T}]_{1\_local} + \mathrm{Pr}_{1\_next} * E[\mathrm{T}]_{1\_next} \tag{4-51}$$

In scenario (ii):

Assume the frequency agility is triggered at the i_{th} packet in next window after interference starts, the valid range of *i* falls in [*N1*, *N2]*, where

$$N_1 = N_0 \tag{4-52}$$

$$N_2 = d \tag{4-53}$$

The response time that frequency agility is triggered at the i_{th} packet in next window is:

$$T(i_{th}) = T(\text{waiting time in local window}) + T(\text{response time in next window})$$

$$= (t_i + \frac{d}{R} - t_s) + (\frac{i}{\mathrm{R}}) \tag{4-54}$$

The probability that frequency agility is triggered at the i_{th} packet in next window is:

$$\mathrm{Pr}_i = g(\eta, Y, i) \tag{4-55}$$

Therefore, the expected response time is:

$$E[\mathrm{T}]_2 = \sum_{i=N_1}^{N_2} [\mathrm{Pr}_i * T(i_{th})] \tag{4-56}$$

To sum up the two scenarios, the expected response time *E[T]* in the method with periodical window is:

$$\begin{cases} E[\mathrm{T}] = E[\mathrm{T}]_1 & \text{where interference starts in sub-period 1} \\ E[\mathrm{T}] = E[\mathrm{T}]_2 & \text{where interference starts in sub-period 2} \end{cases} \tag{4-57}$$

4.3.4 Comparisons and discussions

Based on above calculations and derivations in section 4.3.3, comparisons about response time in one period between the two methods are showed below in Figure 4.11 and 4.12 based on same parameters in Table 4-5. It is obvious that the proposed periodical window method will dramatically shorten the response time of frequency agility.

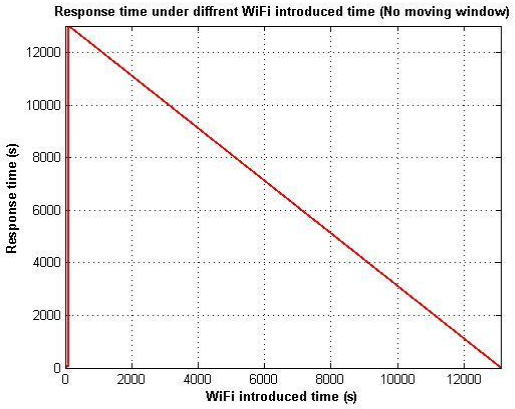


Figure 4.11 Expected response time in method without periodical window

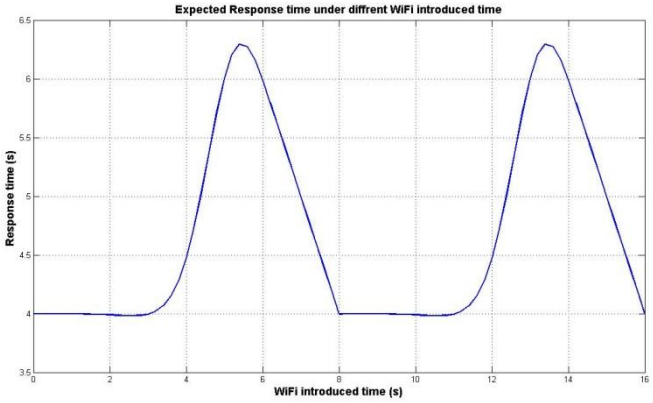


Figure 4.12 Expected response time in periodical window method

As we can see from section 4.3.3, the response time in the periodical window method is related to all the parameters in the calculation, including packet failure ratio *η,* periodical window size *d*, ZigBee traffic rate *R* and triggering threshold *α*. To understand and validate the relationships between theses parameters and their impacts on response time, comparisons are made below. The default parameters are listed in Table 4-5. During the calculation in each group, only one parameter is changed at one time.

Table 4-5 Parameters in periodical window calculation

Parameter	Default values
R	5 packets/second
η	50%
α	25%
d	40 packets

(1) Different packet failure ratio η:

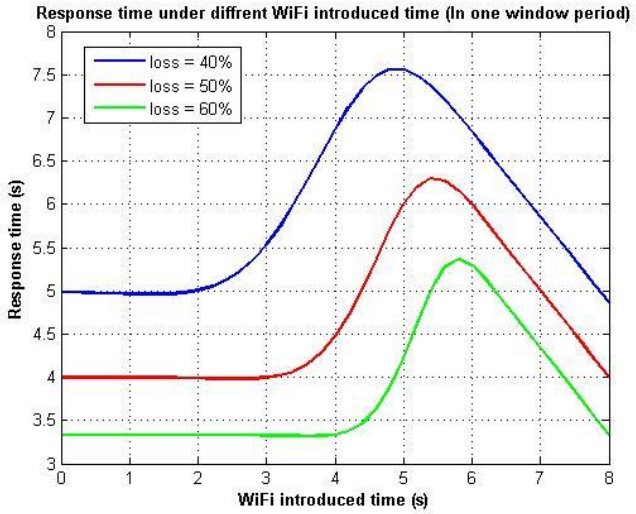


Figure 4.13 Expected response time with different packet loss ratio

With a higher packet failure ratio due to interference, N_{fail} is expected to grow quickly. Therefore, frequency agility is expected to be triggered faster and the average response time is smaller.

(2) Different periodical window size d:

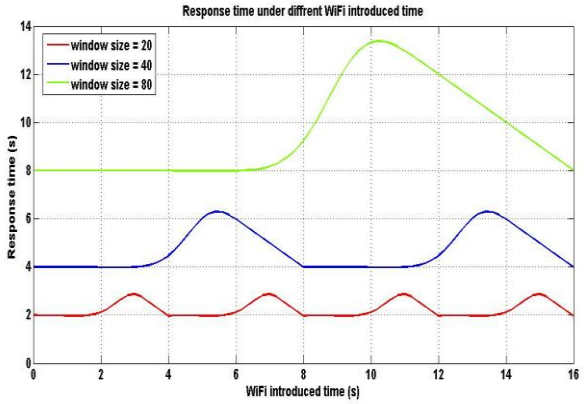


Figure 4.14 Expected response time with different periodical window size

As we can see in Figure 4.14, with different *d*, the window period will be different as well. Meanwhile, a larger *d* will need more time before the frequency agility is triggered, because N_0 is enlarged by *d*. On the other hand, although a smaller *d* can help the node to detect and report the interference quickly, it sometimes triggers false alarm. Because in a small periodical window, N_0 will also be small, fewer packets are needed to trigger the frequency agility. This may lead to false alarms due to random noise or short-time interference. The determination of *d* is trade-off between system sensitivity and false alarm ratio.

(3) Different ZigBee traffic rate R:

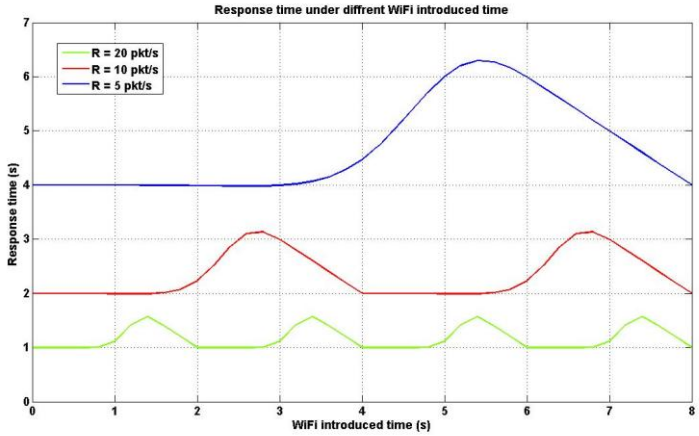


Figure 4.15 Expected response time with different ZigBee traffic rate

With different *R*, the window period is different, as well as the response time. The affects of *R* are similar to *d*, such as response time, false alarm, etc. To some extent, these two parameters can be treated as a couple. Depending on practical applications, different traffic rates *R* exist. Window size *d* should be adjusted according to the *R* in case of false alarm and large response time.

(4) Different triggering threshold α:

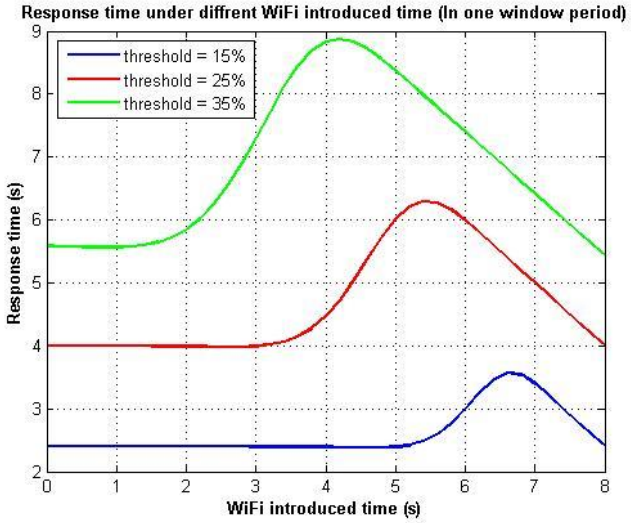


Figure 4.16 Response time with different ZigBee traffic rate

With a lower threshold, the interference is easier to be detected and reported. However, it could result in the false alarm problem again. A higher threshold will reduce the false alarm probability. However, it will also reduce the system sensitivity. In some cases, interference will not be detected and reported. The ZigBee network has to undergo the interference for transmission. In the ZigBee specification, 25% is the recommended threshold.

4.3.5 Simulations

In order to validate the analysis and compare the two methods, we do the simulations in OPNET.

First, we realize the channel moving ability in ZigBee network so that the whole network can move to a new channel under the direction of a PAN coordinator. After that, we realize the periodical window method in order to detect and report interference timely. Figure 4.17 shows the topology of the simulation. Based on the system parameters in Table 3-1 and other default parameters in Table 4-6, simulations

results are retrieved and presented.

Table 4-6 Parameters in periodical window simulations

Parameter	Default values
R	25 packets/second
ŋ	50%
α	25%
d	20

In Figure 4.18, the response time of the method with periodical window is calculated and simulated under different packet loss ratios. As we can see, the simulation results match our analysis in section 4.4.3 well.

Figure 4.17 Simulation Topology

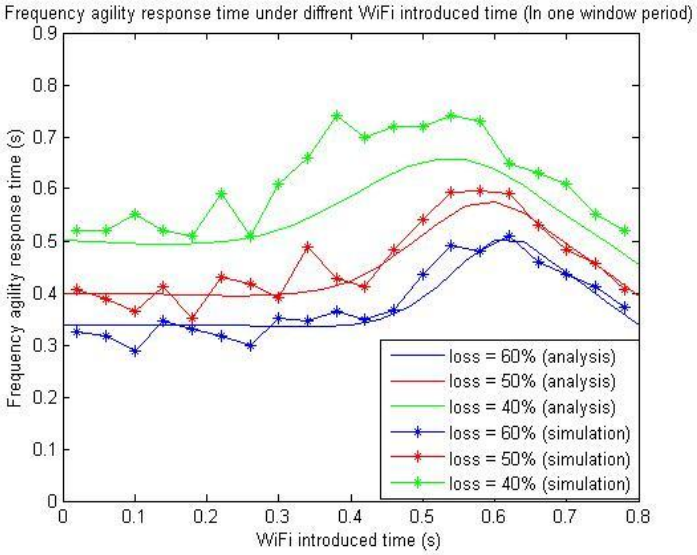


Figure 4.18 Comparison of response time

In Figure 4.19, 4.20 and 4.21, three random WiFi start times are used to compare the

response times between method with and without periodical window. The parameters and results are listed in Table 4-7. Clearly in the method without periodical window, ZigBee traffic is undergoing poor performance for a long time before the frequency agility is triggered. While the method with periodical window can dramatically shorten the response time in simulations and the ZigBee performance can recover much sooner.

Table 4-7 Simulation results in different WiFi start time

WiFi start time (s)	Response time without periodical window (s)	Response time with periodical window (s)
200	More than 2500	Less than 1
800	More than 1900	Less than 1
1400	More than 1100	Less than 1

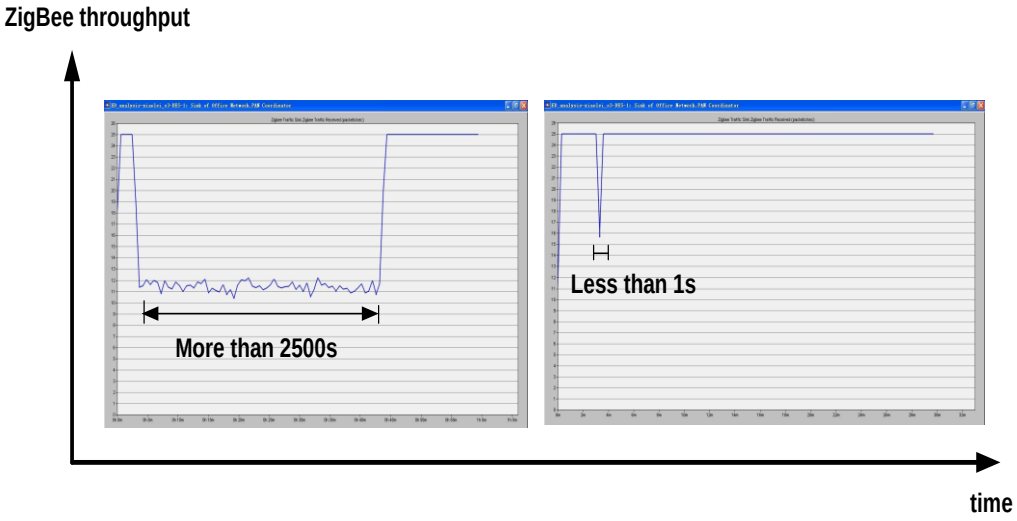


Figure 4.19 Response time comparisons (WiFi start at 200s)
(left: no periodical window right: with periodical window)

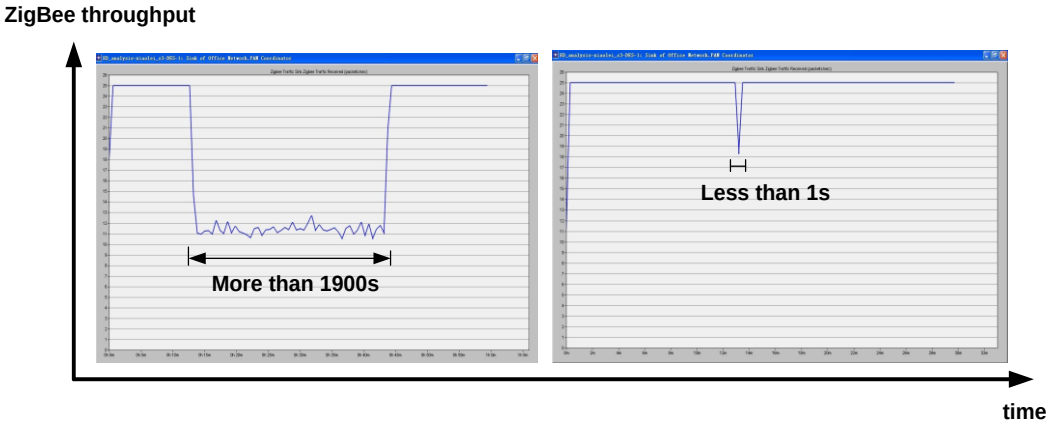


Figure 4.20 Response time comparisons (WiFi start at 800s)
(left: no periodical window right: with periodical window)

ZigBee throughput

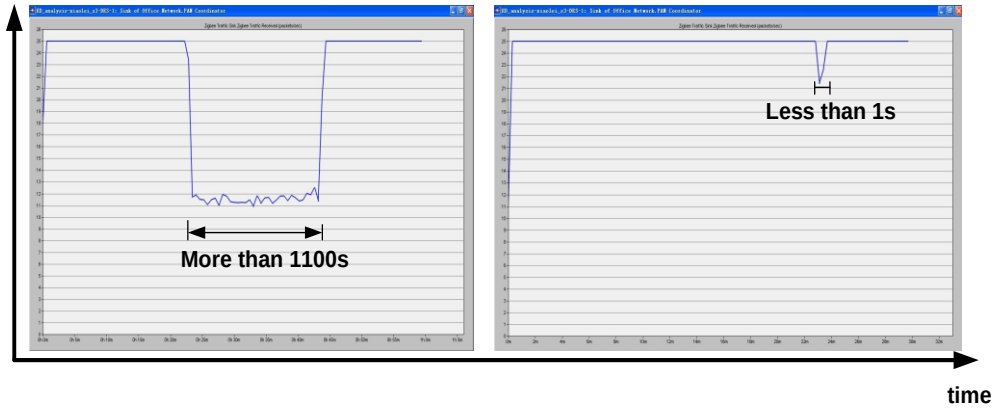


Figure 4.21 Response time comparisons (WiFi start at 1400s)
(left: no periodical window right: with periodical window)

Chapter 5 Multi-channel solution

Besides the periodical window method which is introduced in chapter 4, we propose a new multi-channel solution in this chapter. In section 5.1, we will explain why a multi-channel solution is needed. After that, the procedure and realization will be introduced in section 5.2. The simulation results and discussion will be followed in section 5.3 and 5.4 respectively.

5.1 Why multi-channel

With the proposed periodical window method, the interference can be detected and reported timely and surely. However, in some practical cases, even interference is detected or reported, the frequency agility still will not be triggered. Some examples are listed below:

(1) When interference is detected, the *Mgmt_NWK_Update_notify* message has been sent more than 4 times per hour to the network channel manager. In this case, the message will be ignored by the ZigBee node according to the tracking and reporting behaviors in Figure 4.1.

(2) Even when the ZigBee node has reported the *Mgmt_NWK_Update_notify* message, the network channel manager may decide to keep the original channel based on specific mechanism as described in Figure 4.2. For instance, the manager finds out that only a few nodes are undergoing interference in a large scale ZigBee network.

In the above cases, the frequency agility will not be triggered, and the performance of the nodes under interference will certainly be affected. Moreover, it is understandable that if there is only a part of the whole network suffering from some local interference, it is not necessary for the whole network to move to a new idle channel because this movement is costly and risky.

Therefore, a solution that can improve the performance of the nodes under interference is desirable when frequency agility is neither possible nor necessary to work. From this view, we extend the frequency agility function and propose a new solution by enabling a single ZigBee network to work on multiple channels. As some local interference appears, the part of the network which is under the interference can move to a new idle channel while maintaining the communication links with the other part of the network which stays on the original channel. The moved part can later move back to the original channel as the interference disappears.

5.2 Implementation of multi-channel solution

5.2.1 Model and parameters

In our solution model, when a ZigBee network is undergoing interference, four types of nodes can be classified based on two aspects, namely the *packet loss ratio* and *link relationship in the routing table*. Figure 5.1 illustrates these four types of nodes and their relationships. Due to different locations, each ZigBee node may have different packet loss ratio under the same local interference. Based on the failure threshold α, different packet loss ratio will determine whether a ZigBee node will trigger a channel change or not. The nodes that have a packet loss ratio larger than the threshold α will trigger a channel change. Link relationship in the routing table will determine whether a triggered node has direct communication link to a non-triggered node and vice versa.

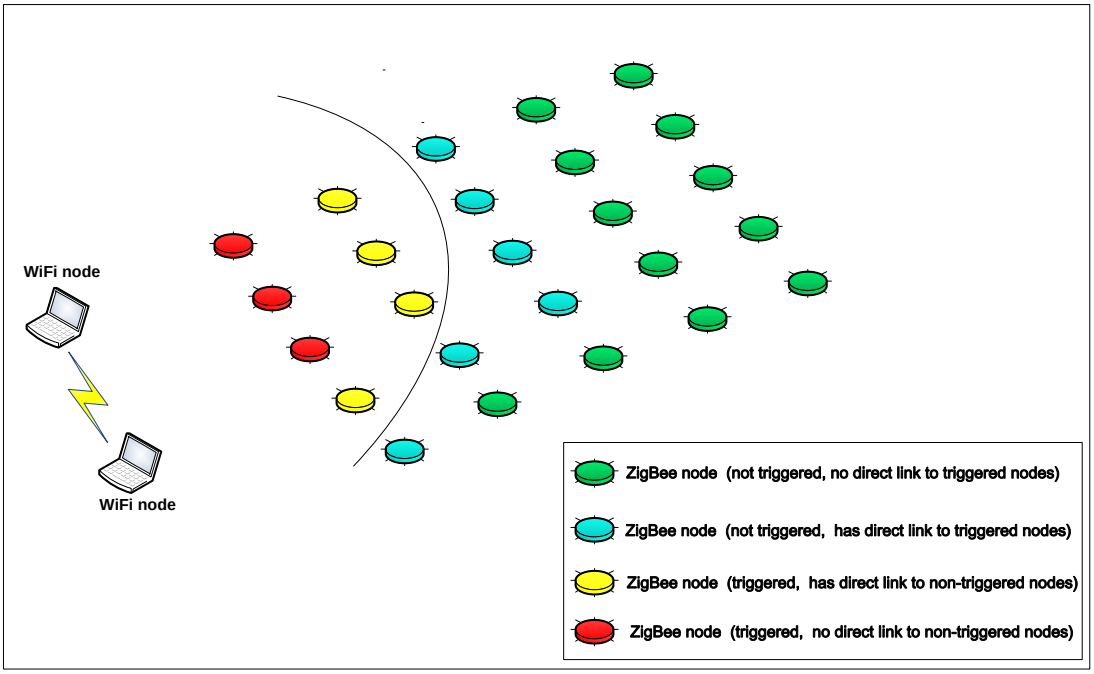


Figure 5.1 Model in multi-channel solution

As in the figure above, the red and yellow nodes have a packet loss ratio larger than the threshold α under WiFi interference and they will move to a clear channel. The cyan and green nodes don't have a large enough packet loss ratio to trigger a channel change, hence they will stay in the original channel. Moreover, the yellow and cyan nodes have direct communication link to each other and they are treated as the border nodes close to dividing line in the figure.

When serious local interference appears, the ZigBee network can work as two cooperative parts supported by the in multi-channel solution. The transmission

between triggered nodes will be operated on the new clear channel. The non-triggered nodes also communicate with each other in the original channel. The borders nodes like the yellow and cyan ones have the knowledge and ability to transmit packet on either the original or the new channel. According to the information in the routing table, they can decide which PHY channel to be used for different packets. Different type of nodes will transmit packets on specific channel for different destinations. The relationships are summarized in Table 5-1. After interference disappears, the ZigBee network is able to move back to one original channel as well. This procedure can repeat once the interference appears again and has no times limitation (like 4 times per hour in frequency agility standard).

Table (5-1) Communication channel relations (next hop channel)

Source \ Next hop	Red	Yellow	Cyan	Green
Red	new	new	/	/
Yellow	new	new	original	/
Cyan	/	new	original	original
Green	/	/	original	original

In order to realize the multi-channel solution, the periodical window method introduced in the previous chapter is used for interference detection and reporting. Meanwhile, some new parameters need to be defined and added to the system in order to fulfill all functions:

(1) In order to fulfill the multi-channel function on the node, we extend the routing table information in the NWK layer. The default information in the routing table includes destination address, next hop address, group ID flag, routing status, etc. We add “next hop channel” information into the routing table. When a node has a packet to transmit, it will check not only the destination address and the next hop address, but also the next hop channel to determine which PHY channel should be used for the next hop transmission.

(2) Besides the failure threshold α, a move back threshold β is added so that the triggered nodes can keep tracking the original channel condition. Once interference disappears, the nodes are capable of moving back to the original channel.

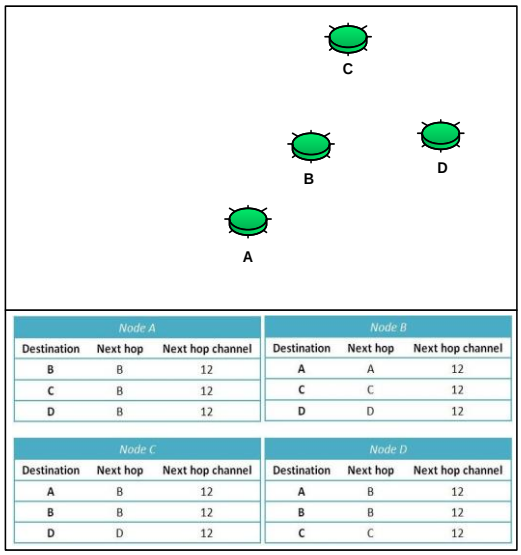
Above we list two important parameters added to the system, in the following section, the whole working procedure will be introduced.

5.2.2 Working Procedures

When interference appears, the nodes undergoing serious interference will be

triggered. The triggered node will broadcast a notification to its direct communication nodes, namely all the next hop nodes in its routing table. After that, the triggered node will move to a clear channel on the PHY layer as well as keep its NWK layer attributes. So the original communication links will remain the same as before, as well as the whole network. After receiving the notification, the direct communication node will refresh the next hop channel information for the triggered node. After that, according to the “next hop channel” attribute in the routing table, packets for the triggered nodes will be transmitted on the new channel, which will significantly improve the performance of these nodes.

An example is shown in Figure 5.2. In (a), the routing tables of the nodes before the interference appears are shown. When interference starts, node A and B are triggered and the procedure of broadcasting notification is illustrated in (b). In (c), the routing tables are refreshed and the nodes change their types.



Node A

Destination	Next hop	Next hop channel
B	B	12
C	B	12
D	B	12

Node B

Destination	Next hop	Next hop channel
A	A	12
C	C	12
D	D	12

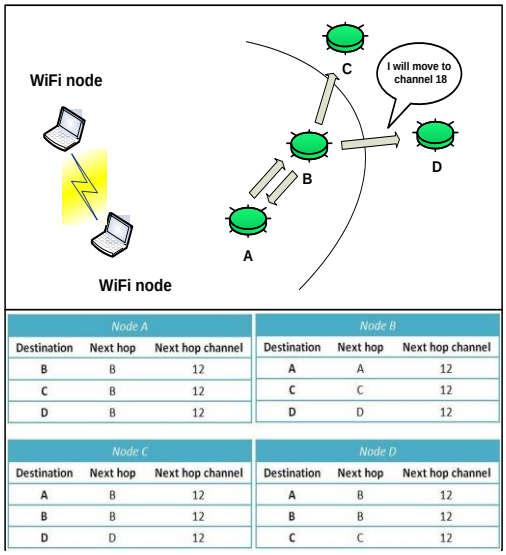
Node C

Destination	Next hop	Next hop channel
A	B	12
B	B	12
D	D	12

Node D

Destination	Next hop	Next hop channel
A	B	12
B	B	12
C	C	12

(a)



Node A

Destination	Next hop	Next hop channel
B	B	12
C	B	12
D	B	12

Node B

Destination	Next hop	Next hop channel
A	A	12
C	C	12
D	D	12

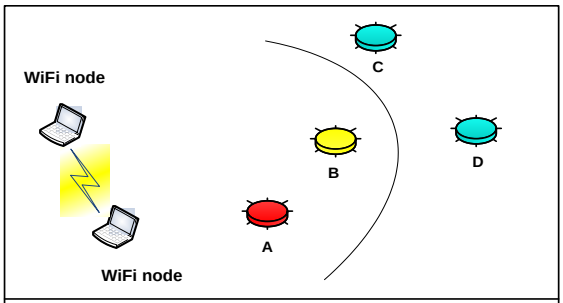
Node C

Destination	Next hop	Next hop channel
A	B	12
B	B	12
D	D	12

Node D

Destination	Next hop	Next hop channel
A	B	12
B	B	12
C	C	12

(b)



Node A

Destination	Next hop	Next hop channel
B	B	18
C	B	18
D	B	18

Node B

Destination	Next hop	Next hop channel
A	A	18
C	C	12
D	D	12

Node C

Destination	Next hop	Next hop channel
A	B	18
B	B	18
D	D	12

Node D

Destination	Next hop	Next hop channel
A	B	18
B	B	18
C	C	12

(c)

Figure 5.2 Example of broadcasting notification and refreshing routing table

Based on a similar procedure, these nodes can move back to the original channel when the interference disappears. The general working procedure of the multi-channel solution is described in Figure 5.3.

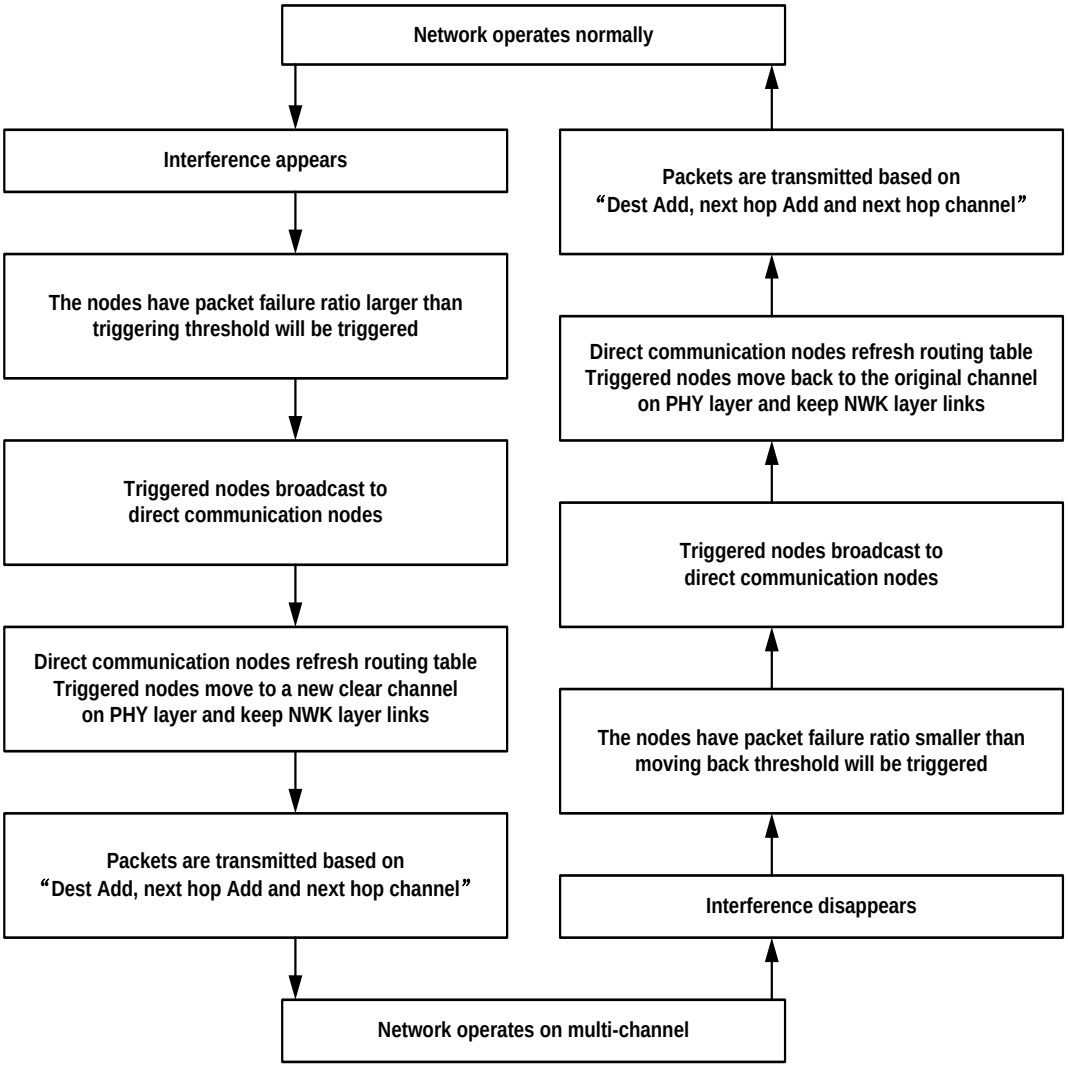


Figure 5.3 General working procedure of multi-channel solution

Figure 5.4 illustrates how a triggered node in the ZigBee network is working. The node will first follow the steps in the periodical window method until the interference is detected. After that, it will broadcast the interference notification to all the next hop addresses in the routing table, which are the direct communication nodes. The broadcast packet radius will be limited to one so that the packet will not be forwarded by the direct communication nodes any further. Because the broadcast procedure is finished under interference, as has been calculated in section 4.2.3, a multiple times broadcast can improve the possibility of receiving the notification packet. After the broadcast procedure, the triggered node will change its channel on the PHY layer and keep listening on this new channel. The NWK layer, however, is not aware of this. This

ensures that the ZigBee node is still operating in the existing network.

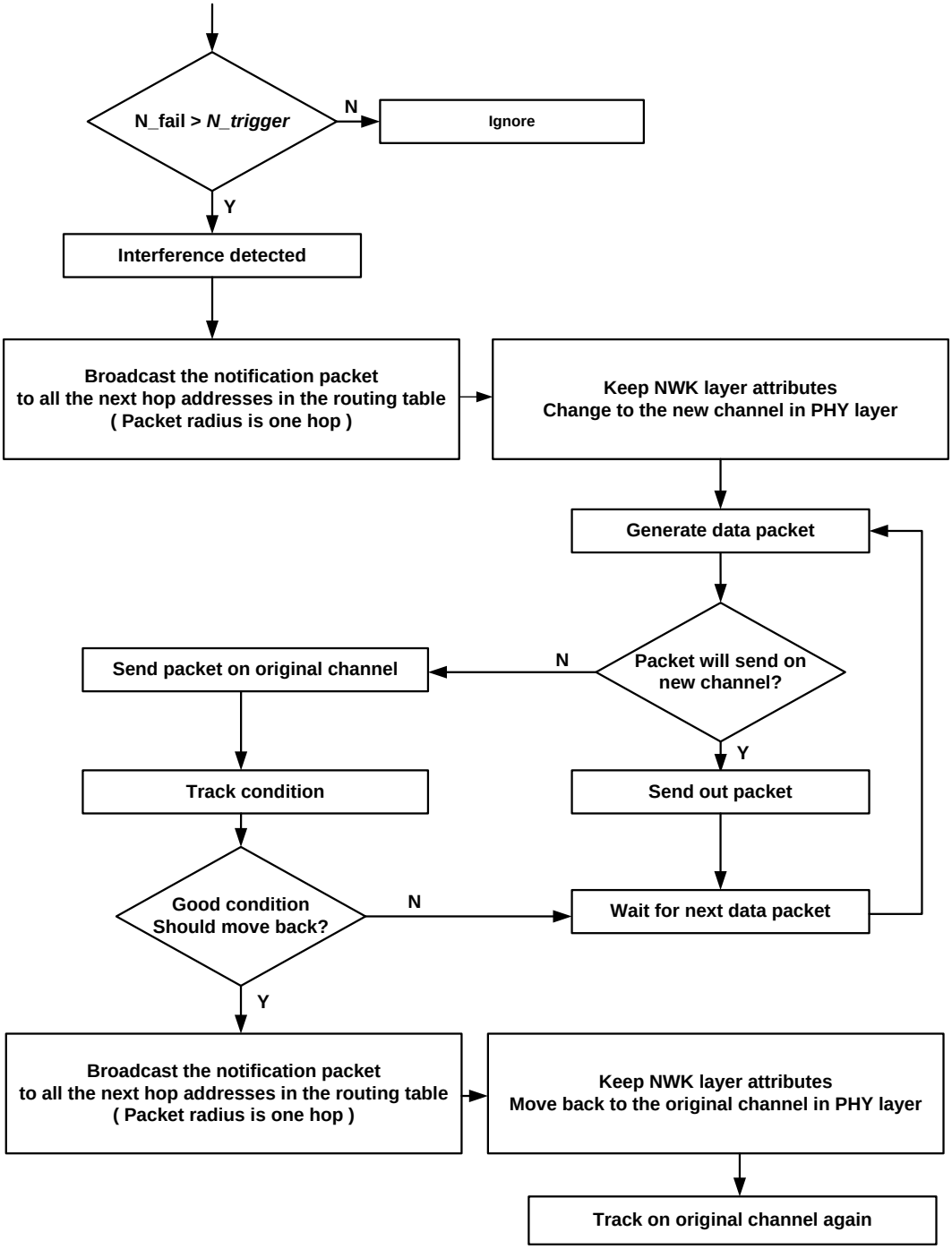


Figure 5.4 Working procedures of triggered nodes

As a triggered node, it will keep tracking and monitoring the original channel condition in order to move back to the original channel when interference disappears. As we can see in the Figure 5.1, the red nodes have only new channel information in their next hop channel attribute of the routing tables. But there are always some triggered nodes (the yellow ones) have both original and new channel information in their next hop channel information of the routing tables. When these nodes transmit a packet on

the original channel (which means transmit a packet to cyan or green node), they will track and calculate packet loss ratio on the original channel by using the periodical window method. When the packet loss ratio on the original channel drops below the move back threshold β, the triggered node will move back to the original channel. The procedure of moving back is similar as above.

Figure 5.5 explains the node response when it receives a notification report. When a node receives a broadcast report, it will refresh its next hop channel information in the routing table. However, we found that a delay before the refresh is necessary if the report is about moving to a new channel. Otherwise, if the node refreshes its routing table immediately after receiving a broadcast report, it may lose the chance to be triggered which it is supposed to be.

An example about refresh delay is illustrated in Figure 5.6. In this example, node A is only communicating directly with node B. Assume that once WiFi interference starts, node B is triggered earlier than node A and broadcasts a notification to its direct communication nodes. If node A refreshes its next hop channel information of the routing table immediately after receiving the notification from node B, all packets of node A are going to be transmitted on the new channel after that. Due to no interference on the new channel, the periodical window has no more failure packet anymore and frequency agility will not be triggered. Node A will therefore stay on the original channel for transmission. Every time node A wants to transmit a packet, it needs to temporarily change its PHY channel to the new one and move back after transmitting a packet to B. This wastes much resources and can be handled if A refresh its routing table after a delay when receive a notification from B. Since node A is expected to be triggered by WiFi interference during this delay.

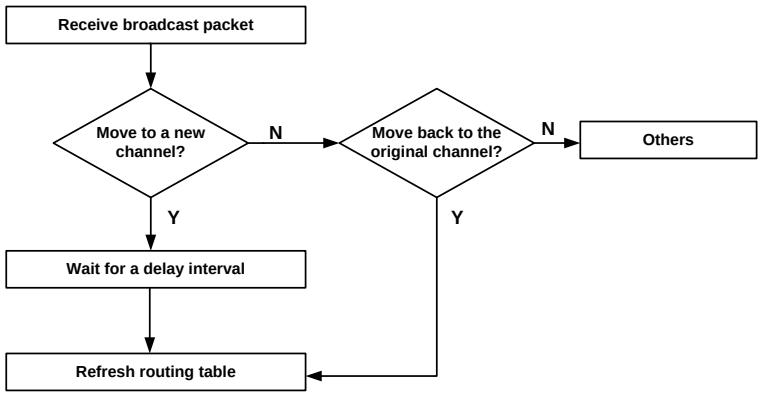


Figure 5.5 Working procedures of nodes receive notification

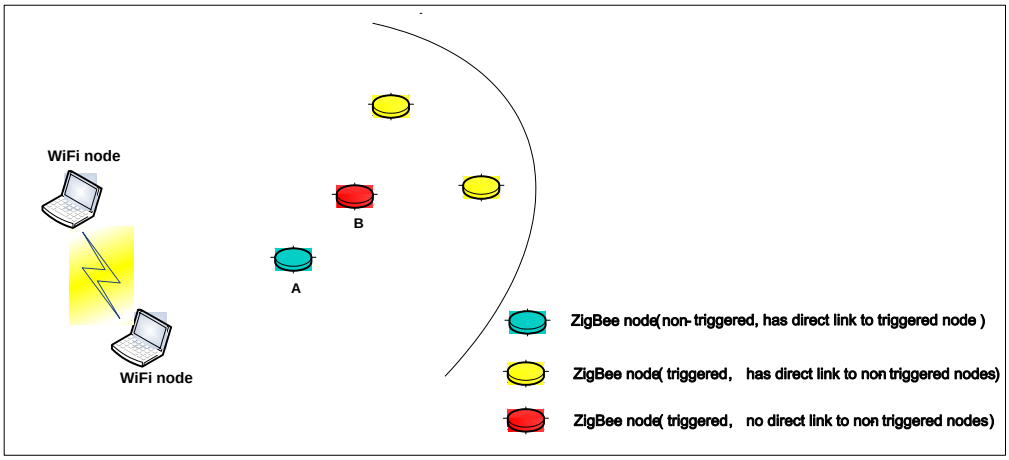


Figure 5.6 Example for needing refresh delay

5.2.3 Discussions

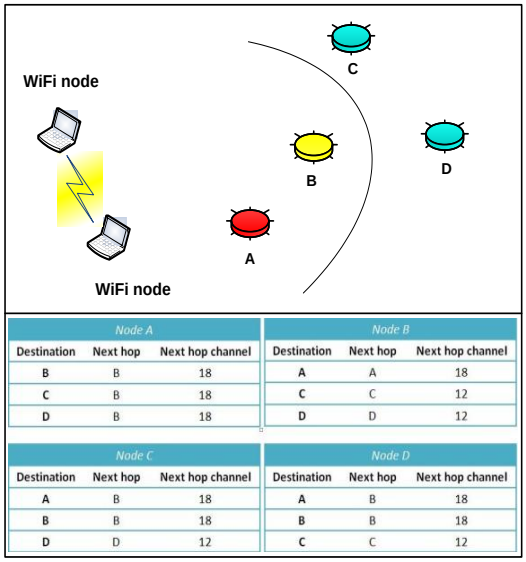
(1) Detect and move to new channel in short time duration

Once interference starts, the nodes will simultaneously suffer from the interference, no matter severe or not. Within a short time duration, all the nodes under severe interference will detect and report the interference. The refresh delay mentioned above is used for this short time duration. In this way, we assure that all the nodes under severe interference will be triggered as they suppose to be.

(2) *Move back to original channel step by step*

The triggered nodes that still have the original channel in the next hop channel information of routing table will keep on tracking and monitoring the channel condition of the original channel. It is also using the same method and parameters in periodical window method. Every time a packet is sent on the original channel, it is tracked and calculated in the periodical window. Once the failure ratio in one window decrease to β, the node will broadcast a notification and move back to the original channel.

Based on this procedure, the yellow nodes in Figure 5.1 will first move back to the original channel and notify their direct communication nodes, namely red and cyan nodes. Then the red nodes are becoming yellow nodes and the cyan nodes are becoming green nodes. The yellow nodes will become cyan nodes themselves. After that, the new yellows nodes will move back to original after their failure ratio on original channel decrease to β. Therefore, the whole network is moving back to original channel step by step, instead of together. Using the same example in Figure 5.2, the moving back procedure is shown below in Figure 5.7.



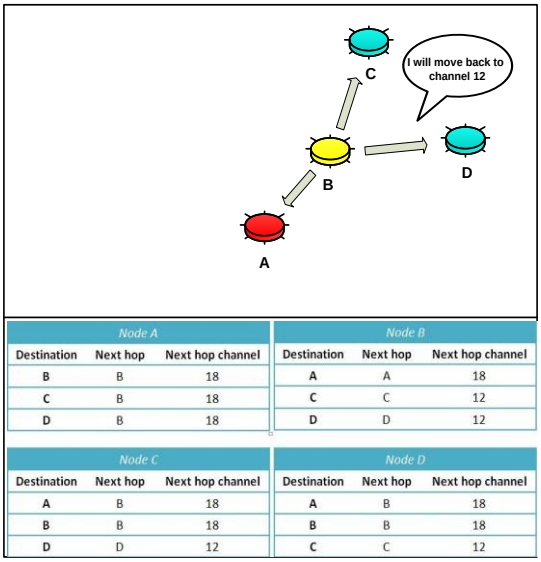
Node A		
Destination	Next hop	Next hop channel
B	B	18
C	B	18
D	B	18

Node B		
Destination	Next hop	Next hop channel
A	A	18
C	C	12
D	D	12

Node C		
Destination	Next hop	Next hop channel
A	B	18
B	B	18
D	D	12

Node D		
Destination	Next hop	Next hop channel
A	B	18
B	B	18
C	C	12

(a)



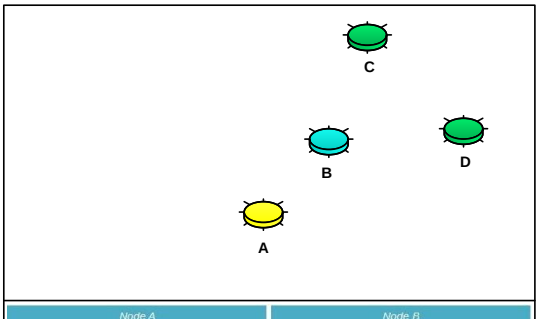
Node A		
Destination	Next hop	Next hop channel
B	B	18
C	B	18
D	B	18

Node B		
Destination	Next hop	Next hop channel
A	A	18
C	C	12
D	D	12

Node C		
Destination	Next hop	Next hop channel
A	B	18
B	B	18
D	D	12

Node D		
Destination	Next hop	Next hop channel
A	B	18
B	B	18
C	C	12

(b)



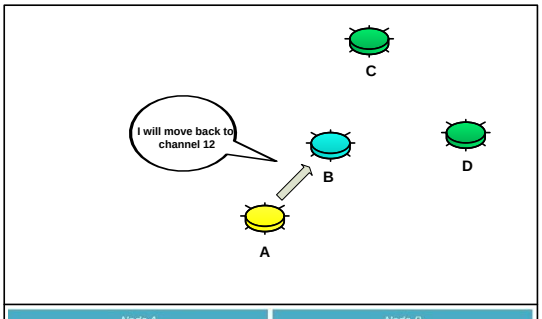
Node A		
Destination	Next hop	Next hop channel
B	B	12
C	B	12
D	B	12

Node B		
Destination	Next hop	Next hop channel
A	A	18
C	C	12
D	D	12

Node C		
Destination	Next hop	Next hop channel
A	B	12
B	B	12
D	D	12

Node D		
Destination	Next hop	Next hop channel
A	B	12
B	B	12
C	C	12

(c)



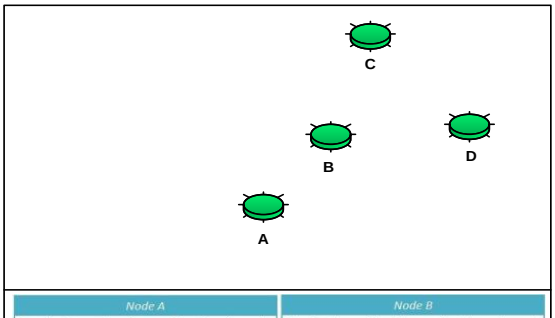
Node A		
Destination	Next hop	Next hop channel
B	B	12
C	B	12
D	B	12

Node B		
Destination	Next hop	Next hop channel
A	A	18
C	C	12
D	D	12

Node C		
Destination	Next hop	Next hop channel
A	B	12
B	B	12
D	D	12

Node D		
Destination	Next hop	Next hop channel
A	B	12
B	B	12
C	C	12

(d)



Node A		
Destination	Next hop	Next hop channel
B	B	12
C	B	12
D	B	12

Node B		
Destination	Next hop	Next hop channel
A	A	12
C	C	12
D	D	12

Node C		
Destination	Next hop	Next hop channel
A	B	12
B	B	12
D	D	12

Node D		
Destination	Next hop	Next hop channel
A	B	12
B	B	12
C	C	12

(e)

Figure 5.7 The procedure of moving back

(3) Compulsory transmission even CCA failure

In the multi-channel solution, when the triggered nodes try to transmit packets to non-triggered nodes with direct communication link, they still suffer interference and this may result in CCA failure. If the maximal backoff times have been reached, the nodes will send the packet after the last CCA attempt, no matter successful or not. This will improve the possibility of receiving packets correctly in some cases, e.g. when the channel is busy, but the transmitted packet may be received successfully due to a sufficient SIR at the receivers.

5.3 Simulations

We implement the multi-channel solution in OPNET for validation. In the simulation, three different scenarios are compared using the parameters in Table 5-1:

(i) The ZigBee network is far away from WiFi interference
(ii) The ZigBee network is located at medium distance
(iii) The ZigBee network is close to WiFi interference

In all the three scenarios, the following aspects of the ZigBee network are compared between methods with and without the multi-channel approach:

(1) global throughput
(2) global end-to-end delay
(3) global throughput drop due to CCA failure
(4) Performance of the node under severe interference

In the simulation, we assume a normal condition that the ZigBee packet failure ratio is less than 3%, which is caused by noise, short-time interference or other ZigBee nodes. So we set β as 3% and α as 25% which is recommended in the standard. The WiFi interference starts at 400s and ends at 700s. Meanwhile, the refresh delay is set to a two window size duration which assures that all the nodes under severe interference will be triggered during this delay. Finally we adopt low ZigBee traffic which is common in practice and designate a new clear channel to move to for simplicity.

Table 5-1 Parameters in our OPNET simulation

Parameter	Values
α	25%
β	3%
d	50 (packets)
R	5 (packets/second)
Refresh delay	20 (s)
Original channel	12
New channel	18
WiFi start time	400 (s)
WiFi stop time	700 (s)

Scenario (i): The ZigBee network is far away from the interference

In this scenario, the whole ZigBee network is far away from the WiFi interference such that after interference starts, only a few nodes will be under serious interference and move to a new channel. The topology and the triggered nodes are showed in Figure 5.8

Triggered nodes
2 nodes (N1, N9)

Figure 5.8 Topology in scenario (i)

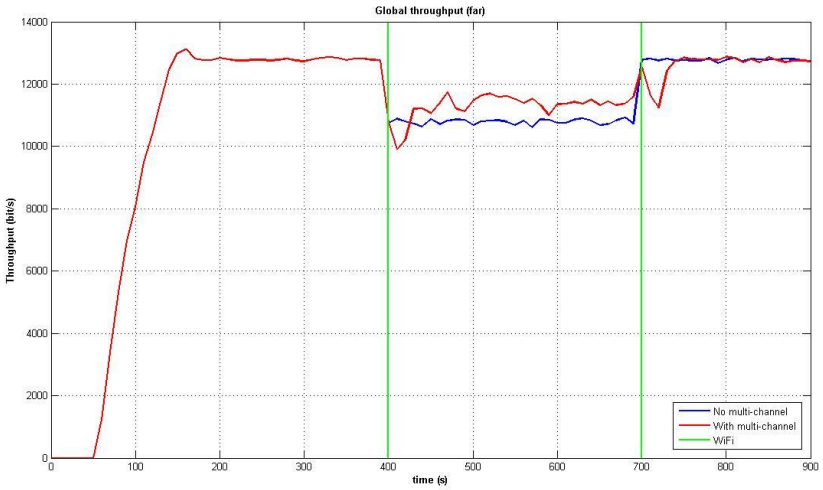


Figure 5.9 Global throughput in scenario (i)

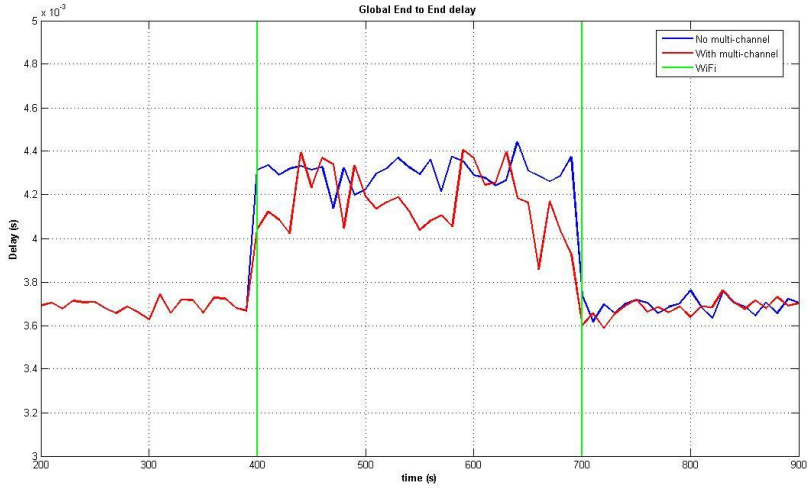


Figure 5.10 Global end-to-end delay in scenario (i)

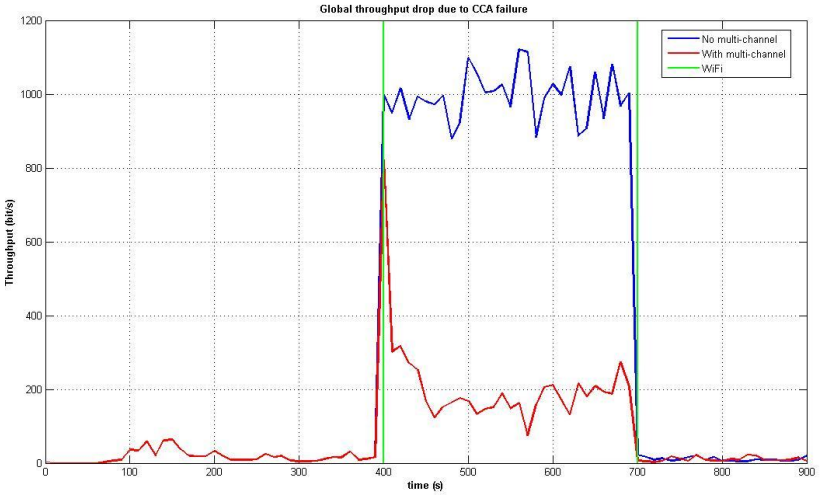


Figure 5.11 Global throughput drop due to CCA failure in scenario (i)

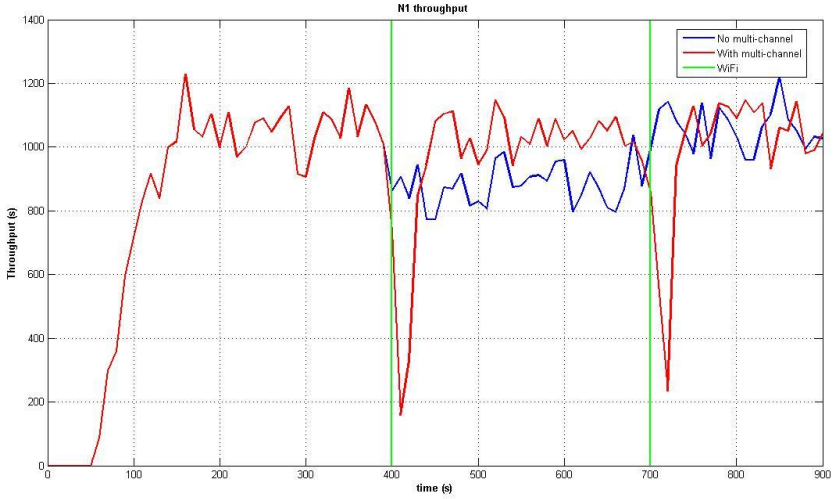


Figure 5.12 N1 throughput in scenario (i)

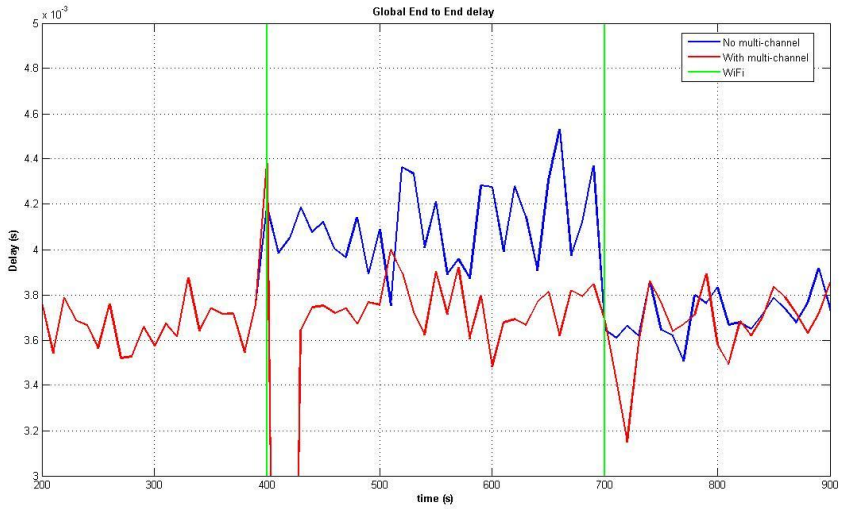
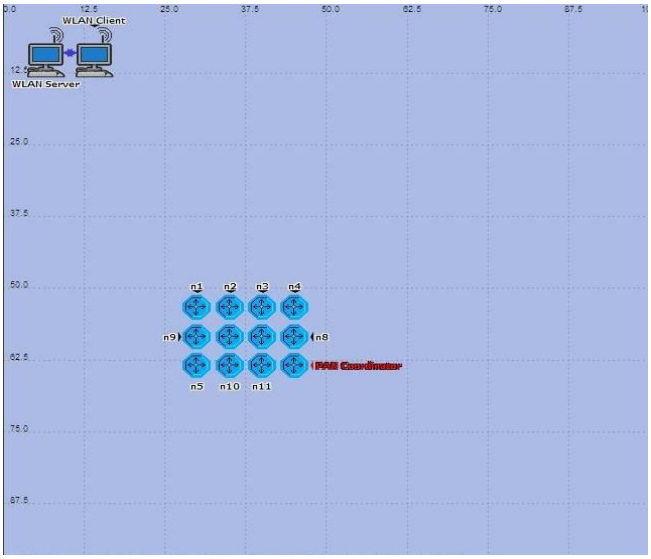


Figure 5.13 N1 end-to-end delay in scenario (i)

Because only a few nodes are triggered and move to a new channel, most of the non-triggered nodes are still under WiFi interference although not severe. In this scenario, the global throughput in Figure 5.9 and the global end-to-end delay in Figure 5.10 are partly improved. Meanwhile, the throughput drops due to CCA failure in Figure 5.11 is well improved. It benefits from the multi-channel nodes, as well as the compulsory transmission even with CCA failure which is mentioned in section 5.2.3. From Figure 5.12 and 5.13, it is clear that the performance of the triggered node in terms of throughput and end-to-end delay have been well improved which is desired in the multi-channel solution.

Scenario (ii): The ZigBee network is located at medium distance

In this scenario, the whole ZigBee network is located at a medium distance from the WiFi interference such that after interference starts, nearly half of the nodes will be under serious interference and move to a new channel. The topology and the triggered nodes are showed in Figure 5.14



Triggered nodes
5 nodes (N1-N4, N9)

Figure 5.14 Topology in scenario (ii)

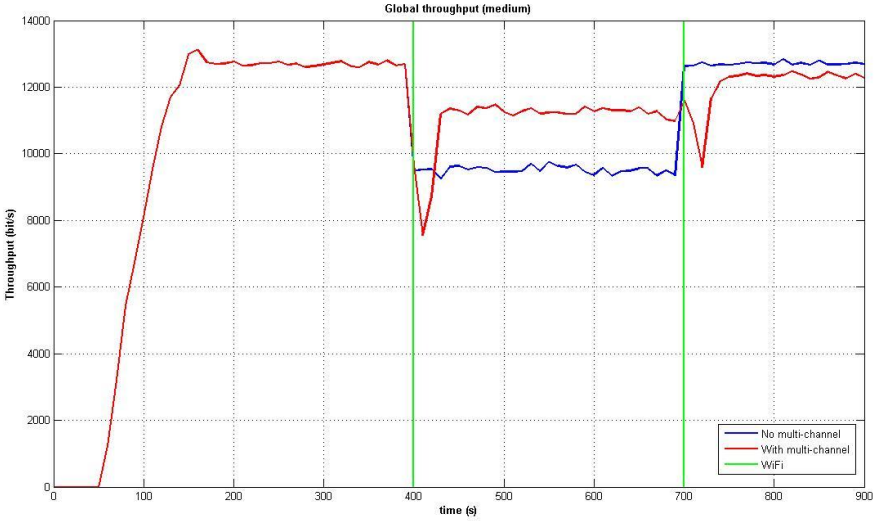


Figure 5.15 Global throughput in scenario (ii)

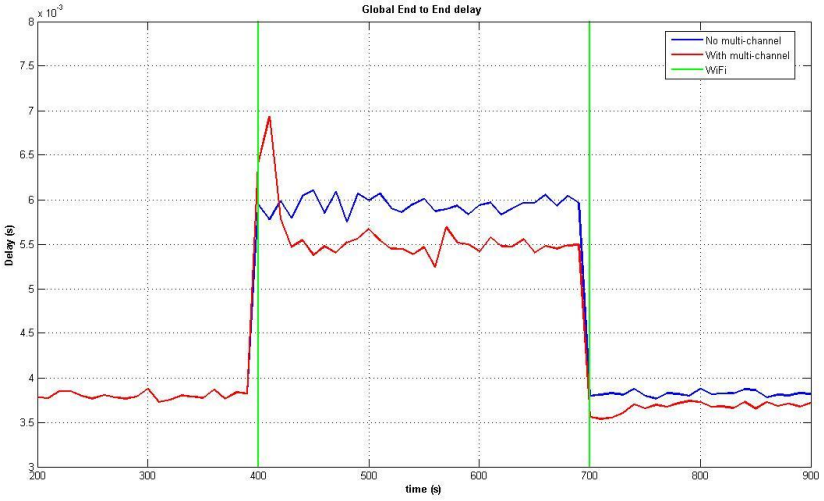


Figure 5.16 Global end-to-end delay in scenario (ii)

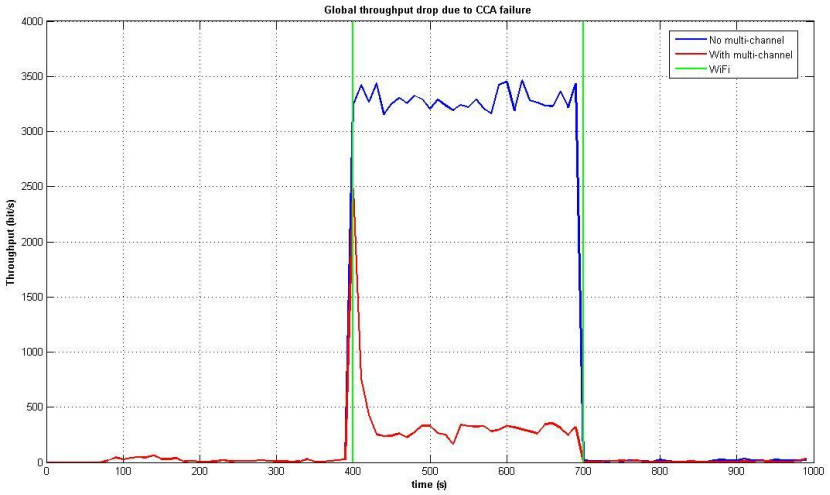


Figure 5.17 Global throughput drop due to CCA failure in scenario (ii)

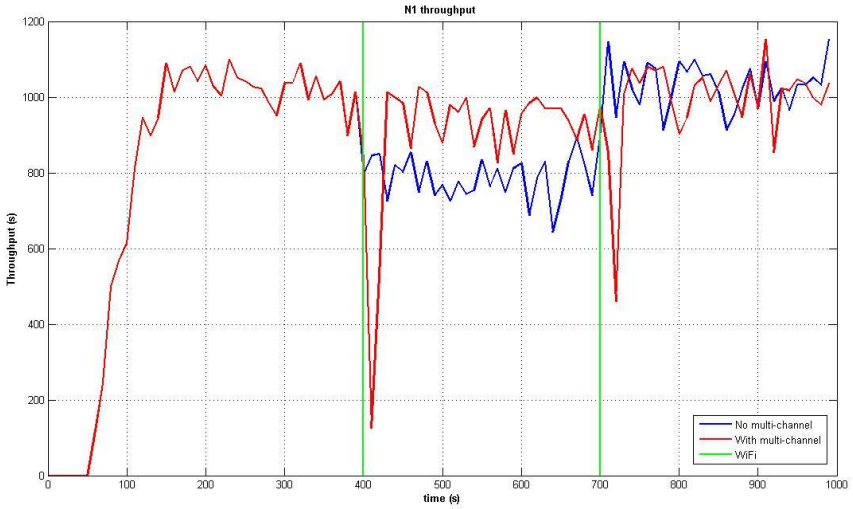


Figure 5.18 N1 throughput in scenario (ii)

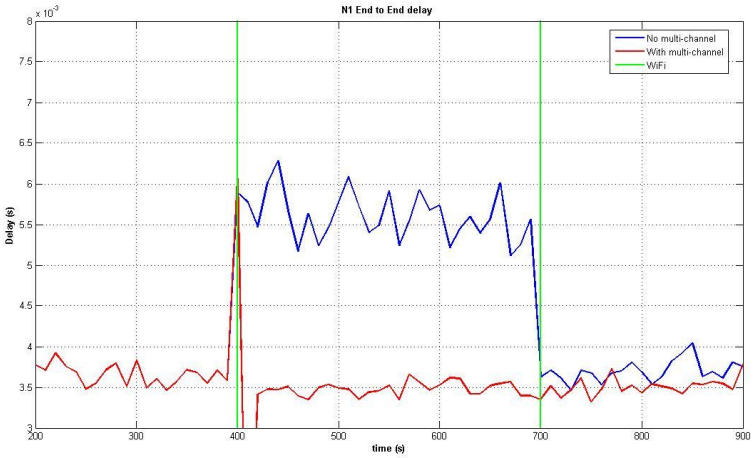
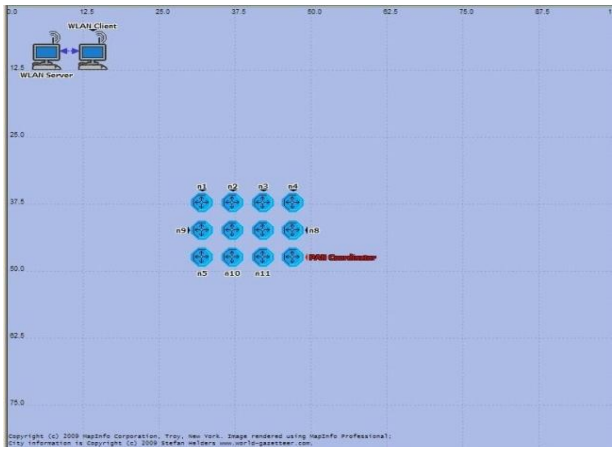


Figure 5.19 N1 end-to-end delay in scenario (ii)

Because almost half of the nodes are triggered and move to a new channel in this scenario, the global throughput in Figure 5.15 and the global end-to-end delay in Figure 5.16 are better improved than in scenario (i). Meanwhile, the throughput drops due to CCA failure in Figure 5.17 is also well improved as in scenario (i). From Figure 5.18 and 5.19, the performance of the triggered node in terms of throughput and end-to-end delay is also improved.

Scenario (iii): The ZigBee network is close to the interference

In this scenario, the whole ZigBee network is close to the WiFi interference such that after interference starts, most of the nodes will be under serious interference and move to a new channel. The topology and the triggered nodes are showed in Figure 5.20



Triggered nodes
10 nodes (N1-N10)

Figure 5.20 Topology in scenario (iii)

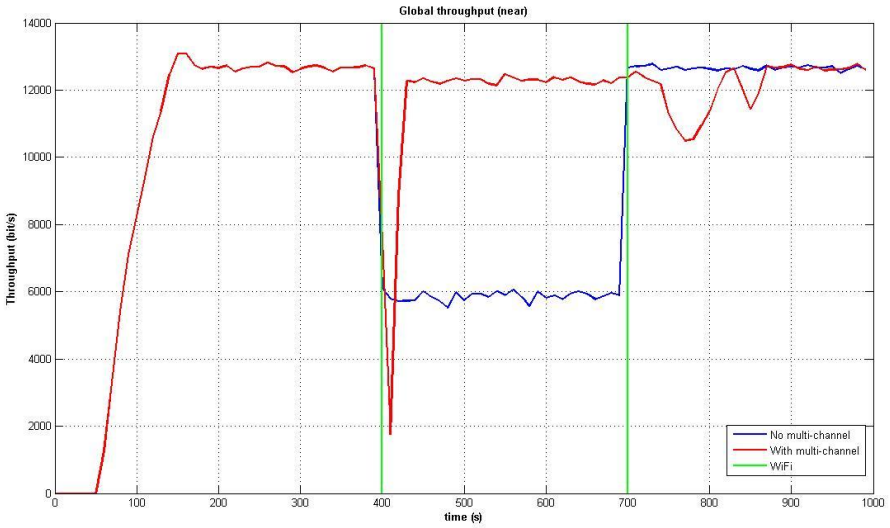


Figure 5.21 Global throughput in scenario (iii)

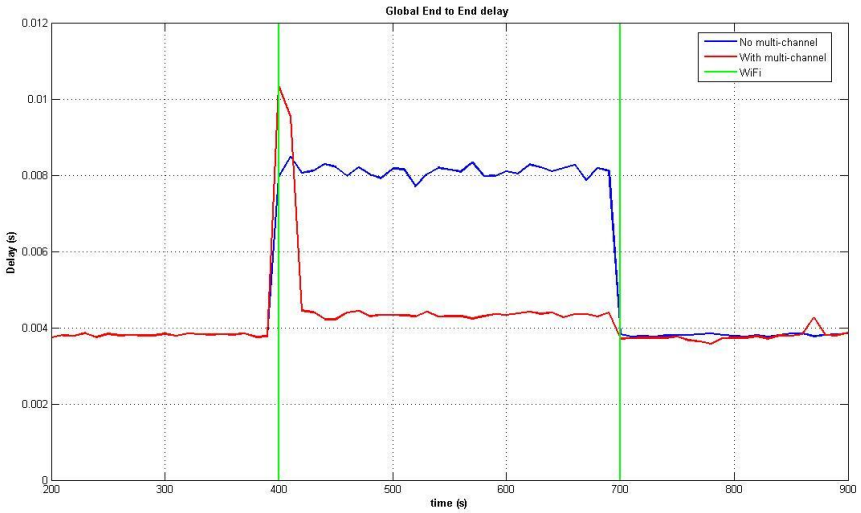


Figure 5.22 Global end-to-end delay in scenario (iii)

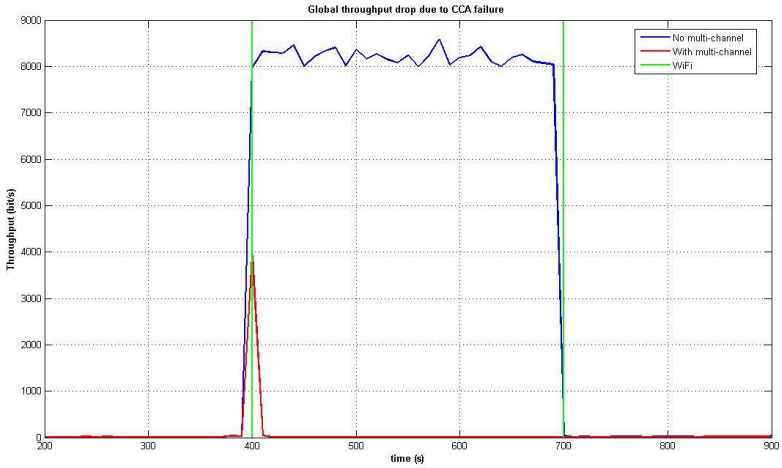


Figure 5.23 Global throughput drop due to CCA failure in scenario (iii)

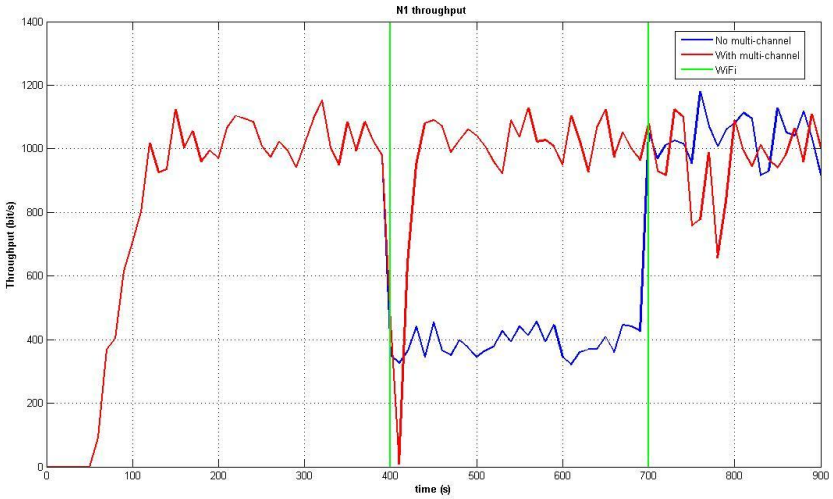


Figure 5.24 N1 throughput in scenario (iii)

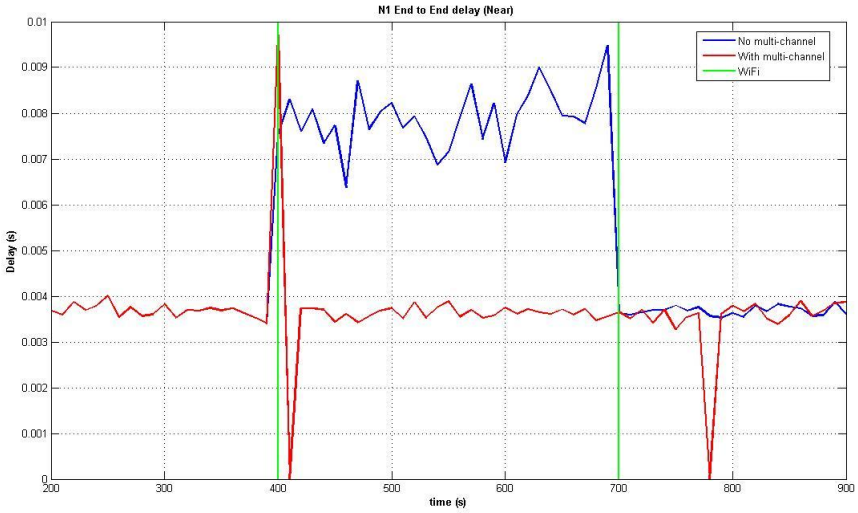


Figure 5.25 N1 end-to-end delay in scenario (iii)

Because most of the nodes are triggered and move to a new channel, only a few non-triggered nodes are still under WiFi interference. In this scenario, the global throughput in Figure 5.21 and the global end-to-end delay in Figure 5.22 are almost recovered back to normal before WiFi interference. At the same time, the throughput drops due to CCA failure in Figure 5.23 almost drops to 0. Meanwhile, from Figure 5.24 and 5.25, it is clear that the performance of the triggered node N1 in terms of throughput and end-to-end delay, have been dramatically improved as other two scenarios above.

Table (5-2) Improvement results in three scenarios

Parameter	Results
Global parameter	(iii) > (ii) > (i)
Triggered nodes parameter	Well improved in (i), (ii) and (iii)

In general, the improvement results can be summed up in Table 5-2. The simulation shows that our proposed multi-channel works well for improving the performance of triggered nodes in various scenarios, which matches our design motivation. If a ZigBee network has more triggered nodes, the global performance is expected to be better as the result of the multi-channel solution. However, the channel switching time in hardware is not included in the simulation, so the end-to-end delay in practice would be longer. The actual delay value depends on the hops and channel switching time in hardware.

Chapter 6 Conclusions

Based on the work in former chapters, we will first conclude the thesis in section 6.1. After that, future work is discussed in section 6.2.

6.1 Conclusions

In this master thesis, we first studied and investigated the coexistence issues between ZigBee and WiFi. Taking into account the significant transmission power and the timing differences between ZigBee and WiFi, a coexistence model based on power and timing aspects is presented. Moreover in the power aspect, three coexistence regions are identified. In each of these regions, WiFi and ZigBee exhibit different interactive behavior and hence different performance.

From the study and investigation above, we came to realize that ZigBee is potentially vulnerable to the interference introduced by WiFi rather than *vice versa*. Therefore, solutions are expected to improve the ZigBee network robustness. We studied an existing fake CTS solution to the coexistence issue in the time domain and found out some significant disadvantages in this solution.

In order to work out a better solution, we focus our research on the frequency agility, a feature specified in the ZigBee standard to improve the ZigBee networks robustness. However, some inadequacies in the standard need to be improved before the frequency agility can function well in practice as it is supposed to do, like long response time to interference, undetermined channel scan duration and ACK packet loss. In order to shorten the response time, we propose a periodical window method which can detect and report interference timely and surely. In this periodical window method, a small window is used for periodically tracking the failed packets. Both the mathematical analysis and the OPNET simulation results prove that this proposed periodical window method can dramatically shorten the response time of frequency agility.

We also find that, in some practical cases, even when interference is detected and reported, the frequency agility still cannot be triggered. The performance of the nodes under severe interference and the whole ZigBee network will then certainly be affected. Moreover, if only a part of the network suffers from local interference, it is not necessary for the whole network to move to a new idle channel because this movement is costly and risky. To overcome these problems, we propose a multi-channel solution. As interference appears, the part of the network which is under

severe interference can move to a new idle channel while maintaining the communication links with the other part of the network which stays on the original channel. After the interference disappears, the moved part can move back to the original channel. Simulation results show that this solution is flexible and effective. In various scenarios, the performance of nodes under severe interference can be well improved. If a ZigBee network has more nodes under severe interference, the global performance and robustness are significantly improved as the result of our multi-channel solution.

6.2 Future works

Certainly, we find that some disadvantages and inadequacies in both the periodical window method and the multi-channel solution are worth to be solved in further works which are explained below:

(i) Determination of the periodical window size is a trade-off. The window size will directly affect the response time and report accuracy. A large window size calls for longer response time, however, it can reduce the probability of false alarms. So the determination should be based on different applications. In future, we expect to find a mathematical model which can suggest an approximate value for the window size once α and β are given.

(ii) The interference notification in multi-channel solution may be lost under interference. As mentioned before, a multiple times broadcast can improve the possibility of receiving the notification packet, but not guaranteed. In order to improve the robustness, a better notification way is expected. Alternatively, an active ability to search for nodes that might have changed channel without notification is also desirable.

(iii) Some parameters in the multi-channel solution are expected to be optimized, such as the refresh delay, etc.

(iv) Always track the interference on the used channel and move away when it becomes occupied instead of only moving back and forth between an “original” and a “new” channel.

(v) Automatically detect an idle channel to move to instead of using a designated one.

Abbreviations and Acronyms

ACK	Acknowledgement
AODV	Ad hoc on-demand distance vector
APS	Application support sublayer
CCA	Clear Channel assessment
CS	Carrier sense
CSMA/CA	Carrier sense multiple access with collision avoidance
CTS	Clear to send
CW	Contention Window
DIFS	DCF interframe space
DSSS	Direct Sequence Spread Spectrum
ED	Energy Detection
FDM	Frequency Division Multiplexing
FFD	Full function device
IEEE	Institute of Electrical and Electronics
ISM	Industrial, Scientific and Medical
LOS	Line of sight
LQI	Link Quality Indicator
MAC	Medium Access Control
NAV	Network Allocation Vector
NIB	Network layer information base
NWK	Network
OSI	Open System Interconnection
PAN	Personal area network
PHY	Physical layer
RFD	Reduced function device
RTS	Request to send
SIFS	Short interframe space
SIR	Signal interference ratio
TX	Transmission
WPAN	Wireless Personal Area Network
WPR	Window-to-period ratio
WSN	Wireless sensor networks
ZDO	ZigBee device objects

List of major symbols

t_0	ZigBee traffic start time (s)
t_i	N_{TX} and N_{fail} reset time (s)
t_s	WiFi start time (s)
R	ZigBee traffic rate (packet/s)
η	ZigBee packet failure ratio due to WiFi interference (%)
α	Packet failure ratio threshold to trigger frequency agility (%)
β	Packet failure ratio threshold to trigger moving back (%)
d	Periodical window size (packet)
N_{TX}	Total TX number in periodical window (packet)
N_{fail}	Failure number in window (packet)
N_0	Transmit failure number threshold to trigger frequency agility (packet)
M	Remaining packet number in local period after WiFi starts (packet)
$T(t_s)$	Response time after WiFi interference starts (s)
x	Distance between a transmitter and a receiver
x_0	Line of sight distance

References

1) John A. Stankovic," Wireless Sensor networks", University of Virginia, 2006
2) ZigBee Alliance,"www.ZigBee.org"
3) ZigBee Standards Organization, "ZigBee specification", 2007
4) Wei Yuan, Xiangyu Wang, Jean-Paul M.G. Linnartz, "A Coexistence Model of IEEE 802.15.4 and IEEE 802.11b/g", 14th IEEE Symposium on Communications and Vehicular Technology in the Benelux, 2007
5) 802.15.4 IEEE standard for Information Technology-Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer Specifications for Low Rate Wireless Personal Area Networks (LR-WPANS), 2006
6) Luca De Nardis, Maria-Gabriella Di Benedetto, "Overview of the IEEE 802.15.4/4a standards for low data rate wireless personal data networks", 4th workshop on positioning, navigation and communication 2007,Germany
7) Bastin Tony Roy Savarimuthu, Morgan Bruce, Maryam Purvis, "A Software Framework for Application Development using ZigBee Protocol", The Information Science Discussion Paper Series, 2009/03
8) Holger Karl, Andreas Willig, "Protocols and Architectures for Wireless Sensor Networks", ISBN:0470095105, Wiley 2005
9) Wei Yuan, Xiangyu Wang, Jean-Paul M. G. linnartz, Ignas G. M. M. Niemegeers, "Experimental Validation of a coexistence model of IEEE 802.15.4 and IEEE802.11b/g Networks", accepted by International Journal of Distributed Sensor Networks, Taylor & Francis, 2009
10) IEEE standard 802.11 Wireless MAC and PHY layer Specification, 1999
11) K. Marquess, "Physical Model Sub-Group Discussion and Questions", IEEE 802.15/138R0, 1999
12) Guang Yang, Yu Yu, "ZigBee networks performance under WLAN 802.11b/g interference", Digital Object Identifier 10.1109/ISWPC.2009.4800615
13) A. Kamerman, "Coexistence between Bluetooth and 802.11 CCK Solution to Avoid Mutual Interference", Lucent Technologies, Jan 1999
14) Wenji Zhou, "Design and Evaluation of a Hub-Assisted WLAN/ZigBee Coexistence Method", master thesis report, Eindhoven University of Technology, 2008
15) Gilles Thonet, Patrick Allard-Jacquin, Pierre Colle, "ZigBee-WiFi Coexistence", White paper and test report, Schneider Electric, 2008
16) JY Jung and JW Lee, "Improved ZigBee Connection Method for Healthcare Device", International Conference on Hybrid Information Technology, 2006