2012

ZigBee-assisted ad-hoc networking of multi-interface mobile devices

Yanfei Wang
Iowa State University

ZigBee-assisted ad-hoc networking of multi-interface mobile devices

by

Yanfei Wang

A thesis submitted to the graduate faculty

in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

Major: Computer Science

Program of Study Committee:

Wensheng Zhang, Major Professor

Lu Ruan

Ting Zhang

Iowa State University

Ames, Iowa

2012

DEDICATION

I would like to dedicate this thesis to my parents, my husband Zhen Li, and to my daughter Carolyn Li without whose encouragement, love, understanding, and support I would not have been able to complete this work. I would also like to thank my friends for their loving guidance and support during the writing of this work.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ACKNOWLEDGEMENTS

I would like to take this opportunity to express my thanks to those who helped me with various aspects of conducting research and the writing of this thesis.

First and foremost, I would like to thank my major professor Dr. Wensheng Zhang for providing me an opportunity for working on this research topic, and his guidance, patience and support throughout this research and the writing of this thesis. His deep insights and valuable directions have often inspired me and renewed my hopes for completing my graduate education. Without his continuous encouragement, guidance and help, I would not have completed this work.

I would also like to thank my committee members Dr. Lu Ruan and Dr. Ting Zhang for their efforts, valuable advices and contributions to this work.

Deep thanks to my friends and co-workers Hua Qin, Zi Li and Xuejia Lu at Computer Science Department. Without the discussion, warm help, guidance and encouragement from them, I would not have completed the research and writing of this thesis. Lastly, I would like to thank my friends Ka Yang, Xi Chen, and Yue Zhao at Department of Electrical and Computer Engineering for their warm help and guidance so that I could resolve many installation and implementation issues about TinyOS, Ubuntu Operating System, and MadWiFi.

ABSTRACT

Wireless ad hoc network is decentralized wireless network, which does not rely on a preexisting infrastructure, such as routers in wired networks or access points in managed (infrastructure) wireless networks. Instead, each node participates in routing by forwarding data for other nodes. The determination of which nodes forward data is made dynamically based on the network connectivity. Node density has a great impact on the performance and efficiency of wireless ad hoc networks by influencing some factors such as capacity, network contention, routing efficiency, delay, and connectivity.

On one hand, maintaining stable connectivity is a big challenge for sparsely deployed and highly dynamic ad hoc wireless network. Vehicle ad hoc network (VANET) which consists of highly mobile vehicles with wireless interfaces is one type of such network, especially in rural areas where vehicles traffic are very sparse. One of the most important applications built on top of VANET is the safety application. In VANET safety applications, source vehicles that observe accidents or some other unsafe conditions of the roads generate warning messages about the conditions, and propagate the warning messages to the following vehicles. In this way, the following drivers have the opportunity to do some necessary action before they reach the potential danger zone to avoid accident. The safety application requires timely and accurate warning message detection and delivery. However, recent researches have shown that sparse and highly dynamic vehicle traffic leads network fragmentation, which poses a crucial research challenge for VANET safety application.

On the other hand, reducing contention and thus maximizing the network throughput is also a big challenge for densely deployed ad hoc wireless network, especially when many devices are located in a small area and each device has heavy duty message to transmit. The WiFi interface perhaps is the most common interface found in mobile devices for data transfer as it provides good combination of throughout, range and power efficiency. However, the WiFi interface may have to consume a large amount of bandwidth and energy for contention and combating collision, especially when mobile devices located in a small area all have heavy traffic to transmit.

Meanwhile, ZigBee is an emerging wireless communication technology which supports low-cost, low-power and short-range wireless communication. Nowadays, it has been common for a mobile device, such as smart phone, PDA and laptop, to have both WiFi and Bluetooth interfaces. As the ZigBee technology becomes more and more mature, it will not be surprising to see the ZigBee interface commonly embedded in mobile devices together with WiFi and Bluetooth interfaces in the near future. The co-existence of the ZigBee and the WiFi interfaces in the same mobile device inspires us to develop new techniques to address the above two issues.

Specifically, this thesis presents two systems built based on ZigBee-assisted ad-hoc networking of multi-interface mobile devices. In order to achieve stable connectivity in a sparse and dynamic VANET, the first system integrates a network of static roadside sensors and highly mobile vehicles to improve driving safety. In order to reduce contention in a densely deployed ad hoc wireless network, the second system assists WiFi transmission with ZigBee interface for multi-interface mobile devices. Extensive implementations and experiments have been conducted to demonstrate the effectiveness of our proposed systems.

CHAPTER 1. Introduction

For a sparsely-deployed wireless ad hoc network, maintaining stable connectivity is crucial to the success of many applications built on top of it. Take the vehicle ad-hoc network (VANET) for example, it is an emerging mobile ad hoc network paradigm formed by highly mobile vehicles and static roadside infrastructure stations. Due to its ability to support both vehicle-to-vehicle and vehicle-to-infrastructure wireless communication, researchers have proposed many applications to be deployed on top of it. One of the most important applications of VANET is to improve driving safety by the collaboration of mobile vehicles and roadside stations. In VANET safety applications, source vehicles that observe accidents or some other unsafe conditions of the roads generate warning messages about the conditions, and propagate the warning messages to the following vehicles. When receive the warning message from preceding vehicles, drivers can do some necessary actions before they reach potential danger zones to avoid accidents. The main purpose of the VANETs safety application is to detect and propagate dangerous road conditions promptly and accurately to following drivers. However, since a VANET consists of highly mobile vehicles and the speed and density of the vehicle network vary from time to time, it may not guarantee stable wireless connectivity when the network density is low (e.g., in rural highways, midnight time). Therefore, the VANET itself may not provides timely detection or notification of dangerous road condition.

On the other hand, reducing contention is important for a densely deployed ad-hoc wireless network, especially when each device's transmission duty is heavy. The WiFi interface perhaps is the most common interface found in mobile devices for data transfer as it provides good combination of throughout, range and power efficiency. It is commonly observed that the WiFi communication become slow when many mobile devices having heavy traffic to transmit are located in a small area (e.g., conference room, library, stadium, etc.). The reason is that WiFi interface may have to consume a large amount of bandwidth and thus energy for contention and combating collision in such situation.

ZigBee is an emerging wireless communication technology defined by the IEEE 802.15.4 standard. The major feature of ZigBee which distinguishes itself from other wireless technologies is its provision of low-cost, low-power and short-range wireless communication. It is targeted for missions of low data-rate, and supports low-power consumption, security, and reliability. It supports multiple network topologies such as point-to-point, point-to-multipoint and mesh networks. It has been common for a mobile device, such as smart phone, PDA and laptop, to have both WiFi and Bluetooth interfaces. As the ZigBee technology becomes more and more mature, it will not be surprising to see the ZigBee interface commonly embedded in mobile devices together with WiFi and Bluetooth interfaces in the near future. The co-existence of the ZigBee and the WiFi interfaces in the same mobile device inspires us to develop new techniques to address the above two issues.

Towards addressing the connectivity problem in a VANET, instead of relying on expensive roadside infrastructure stations, this thesis proposes to integrate the VANET with an inexpensive wireless sensor network (WSN). That is, sensor nodes with ZigBee interface are deployed along the roadside to sense road conditions, and to buffer and deliver information about dangerous conditions to vehicles regardless of the density or connectivity of the VANET. Each vehicle is equipped with a device which has both WiFi and ZigBee interfaces. Each vehicle can communicate with other vehicles using the WiFi interface, and each vehicle can communicate with the roadside sensors using the ZigBee interface. With the help of roadside sensors, the VANET-WSN can provide guaranteed timely detection and delivery of road conditions to drivers, consuming low energy and supporting long life time. Along with the concept of VANET-WSN integration, new challenges arise and should be addressed. In this thesis, we investigate these challenges and propose schemes for effective and efficient vehicle-sensor and sensor-sensor interactions. Prototype of the designed system has been implemented and tested in the field. Extensive simulations have also been conducted to evaluate the designed schemes. The results demonstrate various design tradeoffs, and indicate that satisfactory safety and energy efficiency can be achieved simultaneously when system parameters are appropriately chosen.

To leverage the ZigBee interface for improving the communication performance of mobile devices in a dense ad hoc network, we propose a ZigBee-assisted WiFi transmission system where the ZigBee interface is used to coordinate the communication activities of WiFi interfaces to reduce contention and collision. In our proposed system, each device has two wireless communication interfaces, WiFi interface and ZigBee interface. The ZigBee interface is in charge of contention management, while the WiFi focuses on transmitting the real data. A prototype of the proposed system and a detailed simulator of it have been implemented; extensive experiments and simulations have been conducted. The results show that, the proposed system can achieve significantly higher throughput and energy efficiency than a system running the standard IEEE 802.11 protocol; moreover, nodes running the proposed system can co-exist with nodes running the IEEE 802.11 protocol, and both types of nodes can achieve better performance than when they all run the IEEE 802.11 protocol.

1.1 Integration of VANET and WSN

1.1.1 Background and Motivation

Driving is an indispensable part of the life of many people; meanwhile, driving accidents have been a big threat to the lives, health and wealth of the people. The past years have witnessed substantial efforts on improving driving safety. Among them, the most prominent technological one might be the emerging vehicular ad hoc network (VANET) and the safe driving-targeted applications built atop the VANET. The VANET is composed of highly-mobile vehicles and sparsely-deployed roadside stations, each equipped with wireless communication devices and optionally with sensing devices. Wireless communication can be conducted between vehicles and/or between vehicles and roadside stations. On top of the VANET, applications have been developed to collect, process, share and deliver real-time information about road conditions. In vehicle safety applications that communicate by some wireless interface, vehicles disseminate traffic-related information to all reachable nodes based on broadcast transmission. When a source vehicle detects a hazard or an accident, it can generate a safety message to the succeeding vehicles. This safety message contains real-time traffic information, which is used to warn travelers before they drive to potential danger zones on the road.

These systems sometimes help in accident prevention, but they are not always effective since the underlying VANET does not provide guaranteed real-time detection of road conditions or stable communication connectivity. Firstly, the VANET only opportunistically monitors road conditions. That is, only when a vehicle or a roadside station detects or is notified of some conditions, can the information be shared within the VANET. For example, a driver driving at night may not know about a deer roaming on the road ahead because no vehicle or roadside station detects or is notified of the condition. Secondly, the VANET can be disconnected due to high mobility and unpredictable movements of vehicles and the sparse deployment of roadside stations. If the VANET is disconnected, critical information about road conditions known by one partition of the VANET cannot be shared timely with vehicles that need to know it but are in other partitions.

To provide real time detection of the road condition, deploying more roadside stations appears to be a solution. This, however, may significantly increase the investment cost; also, lack of power supply is a big obstacle to do so in rural areas. To guarantee the timely delivery of the dangerous road condition, wide area wireless networks (such as cellular networks) could be used to connect disconnected segments of the VANET. This approach may achieve communication connectivity, but it does not solve the problem of lacking guaranteed real-time sensing of road conditions.

1.1.2 Related Work

The interest in vehicular networks research has been increasing exponentially over the last few years. Farnoud et al. in [12] used a positive orthogonal code to distribute a transmission pattern for broadcast messages. In this paper, the performance in terms of the success probability and the average delay in message delivery was reported. In [10], a model for deriving the packet delivery delay between disconnected vehicles, the re-healing time, was proposed. It was shown that this time can increase to values in excess of 100 seconds in multihop disconnected communications, which is unacceptable for vehicular networks. A pure vehicle-to-vehicle network may not be sufficient to ensure good performance and efficiency when the network is sparse. Therefore, many researchers proposed the vehicle-to-infrastructure (V2I) communication where roadside units (RSUs) with better equipment are deployed to increase network connectivity. DV-CAST protocol was proposed in [39], which is a distributed vehicular multi-hop broadcast protocol that relies only on local topology information for handling broadcast messages in VANETs. DV-CAST can operate in all traffic regimes, including extreme scenarios such as dense and sparse traffic regimes. In [31], mathematical models (based on [10]) was developed to determine the average delay of a packet between disconnected source and destination in the presence of RSUs as relays. The results obtained for a specific number of RSUs (connected and disconnected) are compared with the ones where no RSUs are in place. The results show that significant improvements can be achieved with RSUs. In order to improve VANET connectivity, [36] proposed to deploy a limited number of RSUs. In that paper, they proposed a new safety message routing flow mechanism between the vehicles and the RSUs and derive the analytical performance of the proposed mechanism. The results show that the proposed scheme is feasible and can substantially enhance vehicle safety on highways. Due to small number of RSUs, neihter of above proposals can achieve real time detection and delivery of the road condition messages.

With the advance of semiconductor technology, the sensor becomes a good candidate to facilitate the communication in VANET. A static-sensor assisted adaptive routing protocol [9] has been proposed, where the static sensors are deployed at the intersections to facilitate the routing by temporarily storing the messages. A prototype of hybrid sensor-vehicular networks has been proposed in [41]. Besides, many related applications have been investigated. Bohil et al. [6] propose a secure WSN-based roadside architecture to support accident prevention and post-accident investigation. But very little detail is given on the underlying network design and implementation. MobEyes (Smart Mobs for Urban Monitoring) has been developed in [22]. It exploits vehicle mobility to opportunistically diffuse summaries of sensed data. An in-vehicle sensor network has been proposed for remote vehicle diagnosis and management in [18], where sensors inside and around the vehicle form a hierarchical network.

Most of existing work directly applies the technologies in the conventional WSN to the

VANET but do not consider much the unique characteristics of the VANET (e.g., high mobility, pre-defined layout, potential interference, dynamic traffic, etc.). The collaboration between sensors and vehicles has not been exploited adequately. Therefore, a more seamless and profitable integration of the WSN and the VANET is still demanded. Our design targeted at fully leveraging the complementary capabilities and features of sensors and vehicles and providing the basic communication framework for the WSN-VANET applications. Demonstrated by the results of extensive simulation and implementation, our system can meet the efficiency requirement of the WSN and the timeliness requirement of driving-safety applications of the VANET. It maintains constant performance against dynamic traffic patterns. More importantly, our design provides scalability and flexibility for large-scale and realistic application scenarios.

1.1.3 Overview of VANET-WSN System

Since the inexpensive ZigBee can provide low-power, short-range wireless communication, we propose to integrate the VANET with the WSN to provide timely detection of road conditions and to help connect partitioned segments of the VANET. Wireless sensor nodes, for example, MicaZ motes [15], are much *cheaper* than roadside stations of current VANETs. Besides, some inexpensive, low-power and small-size sensing modules, for example, the WiEye passive infrared sensors [25], have been commercialized and can be installed on the motes to sense road conditions with low cost.

These sensor nodes can be deployed along roadside with higher density than current roadside stations to form a connected network together with the VANET. The sensor nodes can sense the road conditions, collect and process the sensing data to find out information useful for safe driving, and deliver the information to vehicles that need it. The sensor nodes also can buffer the safety-related information generated by vehicles, and forward the information to vehicles in different partitions of the VANET.

1.1.4 Two Examples VANET-WSN System Can Be Applied to

Following are two examples showing that deploying WSNs can greatly help in preventing road accidents:

- Example I. Deploying WSN along rural roads can help prevent vehicle-animal collision accidents. As shown in Fig. 1.1, the WSN nodes deployed along roadside can detect a deer roaming on the road and propagates the information within the nearby area. Approaching vehicles will get the warning beforehand, and have enough reaction time to do some necessary work to avoid accident. The advantage brought by the deployment of WSN is significant. It may help to avoid 1.5 million vehicle-deer collisions happening every year (according to auto insurer State Farm) which result in about 150 deaths and $1.1 billion losses [11].

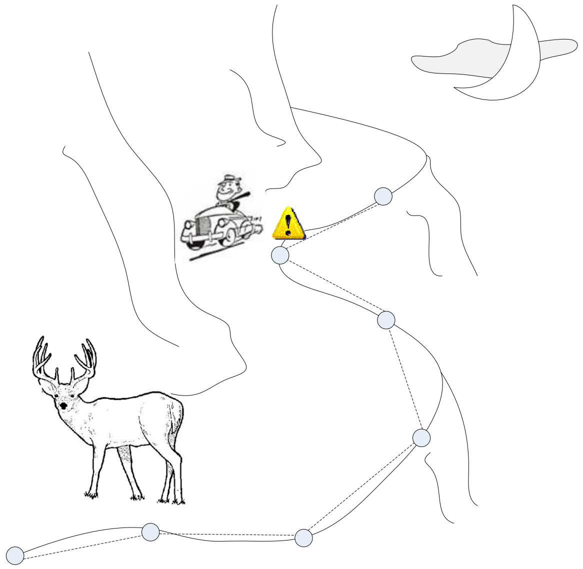


Figure 1.1: Example 1 for VANET-WSN Integration

- Example II. Fig. 1.2 shows that, bad road conditions (e.g., slippery surface) detected by an isolated vehicle can be told to nearby roadside WSN nodes, and the WSN nodes can then collaborate with each other to propagate the information to other vehicles approaching this dangerous area. Note that, this cannot be accomplished if only VANET can be used since the VANET is not connected.

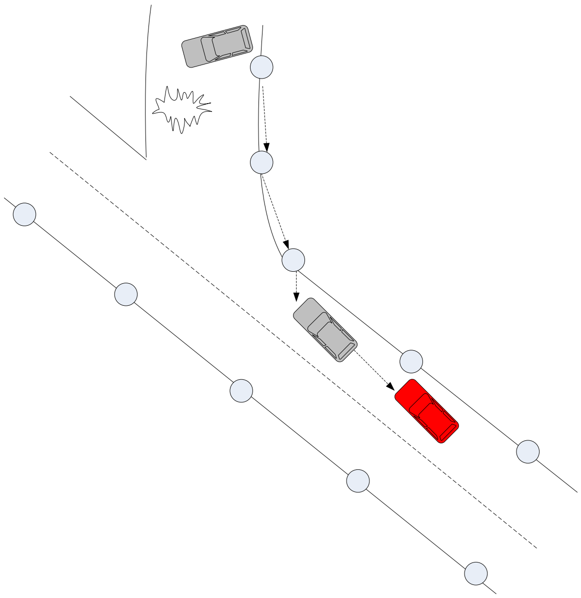


Figure 1.2: Example 2 for VANET-WSN Integration

1.1.5 Objectives and Chanlleges

To realize the proposed VANET-WSN system, several important issues should be investigated. Firstly, the system should be really *viable* in the real scenarios. The impact of interference, noises and other environmental factors on the performance of the system should be investigated. Secondly, the system should be *highly scalable*, considering the large scale of highway system in the world. As the scale of deployment increases, the difficulty in deploying and maintaining the system should not increase much, and the quality of service and the energy efficiency of the system should remain stable. Thirdly, the system should be *flexible* to changes in the real world. WSN nodes may fail or lose time synchronization, the highways may be extended or reshaped, and traffic pattern may change from time to time. It is desired that the deployment and the working parameters of VANET-WSN system can be adjusted with low overhead as the above changes happen. Fourthly, *energy efficiency* should be maximized for the roadside WSN. Although WSN nodes can be deployed and redeployed by humans and their batteries can be replaced manually when necessary, it is still important to minimize the energy consumption and maximize the network life time to reduce energy and maintenance costs. Finally, satisfactory *quality of service* should be attained. Dangerous road conditions should be detected and the information about the dangers should be delivered to related vehicles, both in a timely fashion, to ensure driving safety.

1.1.6 Major Contribution

Towards tacking the above issues, this thesis makes the following major contributions:

- We adopt the idea of group-based modular design to achieve *scalability* and *flexibility*. In our design, the roadside WSN is made up of sensor groups. Each group works autonomously and asynchronously, and neighboring groups interact with each other through a gateway node shared by them. Deployment or redeployment of a group does not affect others; topology and working parameter adjustments conducted within each group do not affect others, either. The modularity nature of the network and the autonomy nature of each module enable easy deployment, extension and reconfiguration. Moreover, our design takes the dynamics of traffic flows and the particularity of sensor distribution into full consideration, making our system highly flexible and scalable in various application scenarios.

- The objectives of *energy efficiency* and *quality of service* are achieved by (i) an event-driven duty cycle scheduling strategy which also leverages the VANET to minimize energy consumption in the WSN, and (ii) low-contention and low-delay communication protocols which ensure contention-less communication within a group and can reduce inter-group contentions with certain coordination costs.

- A prototype of our designed system has been implemented and tested in the field to study the *viability* of the system. Based on realistic vehicle traffic traces and roadside sensor-to-sensor communication traces, extensive simulations have also been conducted to study the impact of various factors on the system performance. The results demonstrate various design tradeoffs, and indicate that desired quality of service and energy efficiency can be achieved simultaneously when system parameters are appropriately chosen.

To the best of our knowledge, this is the first work that proposes, implements and evaluates an integrated VANET-WSN system for driving safety.

1.2 ZigBee-Assisted WiFi Transmission

1.2.1 Background and Motivation

The WiFi interface perhaps is the most common interface found in mobile devices for data transfer as it provides good combination of throughout, range and power efficiency. The WiFi uses CSMA/CA protocol to mediate access to the shared communication medium, which is space. In CSMA/CA protocol, a node wishing to transmit data first sense the channel for a predetermined amount of time to determine whether or not another node is transmitting on the channel within the wireless range. If the channel is idle, then the node is permitted to begin the transmission process. If the channel is busy, the node defers its transmission for a random period of time. There exists two problems which CSMA/CA protocol can not handle. One is the *hidden node problem*, another is the *exposed node problem* [20], which are described as follows.

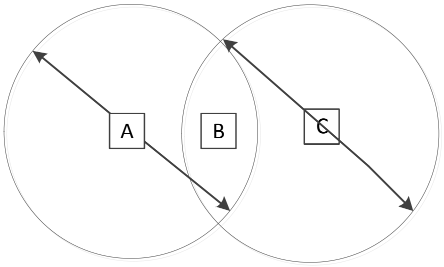


Figure 1.3: The hidden node problem. Although A and C are hidden from each other, their signal can collide at B.

- *hidden node problem*: Consider the situation depicted in Fig. 1.3, where A and C are both within range of B but not each other. Suppose both A and C want to communicate with B and so they each send a frame. A and C are unware of each other since their signals do not carry that far. These two frames collide with each other at B, but neither A nor

C is aware of this situation. A and C are said to be hidden nodes with respect to each other.

- *exposed node problem*: As shown in Fig. 1.4, each of the four nodes is able to send and receive signals that reach just the nodes to its immediate left or right. Suppose B is sending to A. Node C is aware of this communication because it hears B's transmission. However, C can not conclude that it cannot transmit to anyone just because it can hear B's transmission. It is fine for C to transmit to D since C's transmission to D will not interfere with A's ability to receive from B.

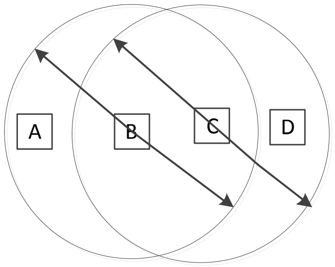


Figure 1.4: The exposed node problem. Although B and C are exposed to each other's signals, their is no interference if B transmits to A while C transmits to D.

In order to address the *hidden node problem* and *exposed node problem*, WiFi uses RTS/CTS mechanism by default to access the channel. In RTS/CTS mechanism, the sender transmits a Request to Send(RTS) frame to the receiver; the RTS frame includes a field that indicates how long the sender wants to hold the channel. The receiver then replies with a Clear to Send(CTS) frame; this frame echoes this length field back to the sender. If the sender hears the CTS, it starts to transmit data frame. If the sender does not receive the CTS after a period of time, it concludes that the RTS must collide with some other RTS. In this case, the sender waits a random amount of time before trying again. Any node that hears the CTS frame knows that it is close to the receiver, and therefore cannot transmit for the period of time reserved by the sender. Any node that sees the RTS frame but not the CTS frame is not close enough to the receiver to interfere with it, and therefore, is free to transmit. The Simplified Algorithm of

WiFi MAC protocol is depicted as Fig. 1.5.

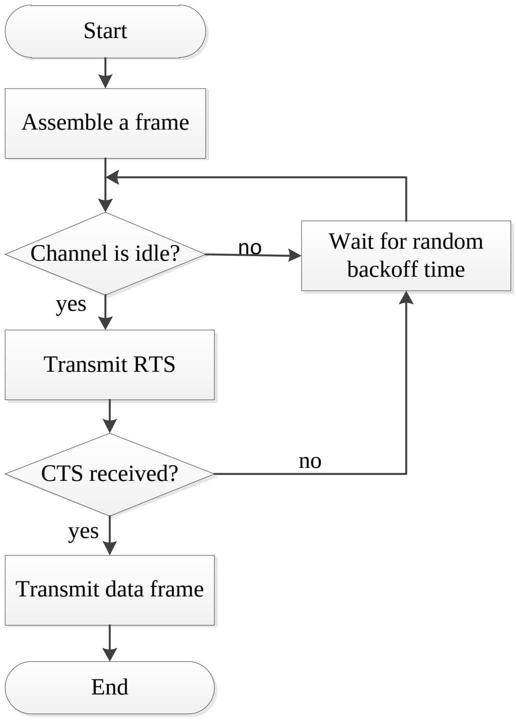


Figure 1.5: Simplified Algorithm of WiFi MAC protocol

As described above, the WiFi interface has to consume a large amount of bandwidth and energy for contention and combating collision, especially when mobile devices located in a small area (e.g., conference room, library, stadium, etc.) all have heavy traffic to transmit. Besides, the RTS/CTS mechanism is by default used to access the channel. Unlike data packets, these control packets are transmitted with a lower rate. According to our simulation results, the maximum throughput(i.e., saturation throughput) decreases and the energy consumption increases(illustrated in Fig. 1.6) rapidly, as the number of concurrent transmitters(running the default IEEE 802.11g protocol) rises. To reduce contention, many protocols have been proposed. However, most of them (e.g., Overlay MAC [30], TDM MAC [19], token-passing MAC [24], etc.) require to either modify the underlying MAC protocol or introduce extra control overhead.

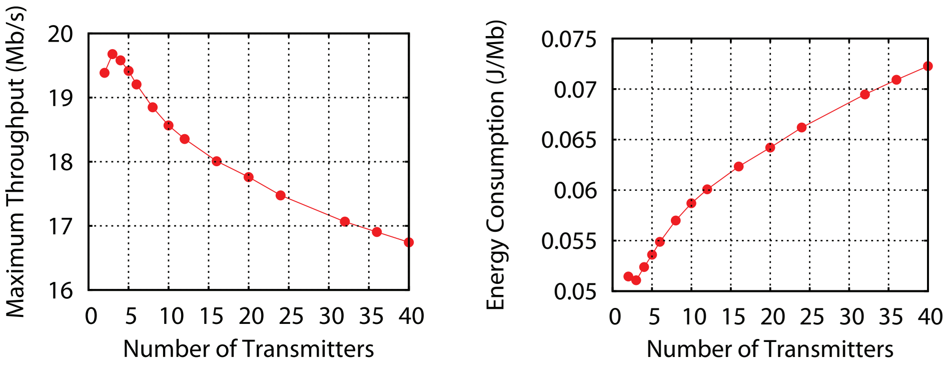


Figure 1.6: Impact of contention

Recently, mobile devices are increasingly equipped with multiple network interfaces [5, 47, 16]. It has been common for a mobile device, such as smart phone, PDA and laptop, to have both WiFi and Bluetooth interfaces. As the ZigBee technology becomes more and more mature, embedded ZigBee interfaces have emerged and the size is becoming smaller and smaller [37, 32]. It will not be surprising to see the ZigBee interface commonly embedded in mobile devices together with WiFi and Bluetooth interfaces in the near future. With ZigBee interfaces, mobile devices can communicate with various electrical and electronic appliances to realize the smart home entertainment and control, home awareness, mobile services, commercial building and smart industrial plants [44].

1.2.2 Proposed ZigBee-Assisted WiFi Transmission system

The co-existence of the ZigBee and the WiFi interfaces in the same mobile device inspires us to develop new techniques to address the above issue. The key idea is that nearby mobile devices use their ZigBee interfaces to coordinate their communication activities to reduce contention and collision. The rationales behind the idea are as follows. The ZigBee interface and the WiFi interface can use different channels, and hence the coordination using ZigBee interfaces will not consume the WiFi bandwidth. As the WiFi transmission has higher rate and energy consumption than ZigBee transmission, the utilization of WiFi for large-size data transmission and ZigBee for small-size control message transmission presents an ideal, efficient resource allocation pattern. Such collaboration is possible because ZigBee may not be used frequently in the places, such as conference room, library and stadium, where WiFi traffic could be very heavy.

In this thesis, we propose a simple yet effective ZigBee-assisted WiFi transmission system for the *high traffic density* scenario. In this system, mobile devices leverage ZigBee communication to form clusters where each cluster has a cluster head and multiple cluster members that can directly communicate with the head via the ZigBee interface. According to the communication demands of individual mobile devices, members in the same cluster collaboratively run a TDMA-like protocol with the ideal goal that, at any moment only one of them attempts to use the WiFi channel so as to eliminate or greatly reduce the contention within a cluster and thus mitigate the contention in the whole network. This system runs on top of the underlying WiFi MAC protocol, therefore, it does not need to modify the underlying WiFi MAC protocol.

To evaluate the feasibility and performance of our proposed system, a prototype of it has been implemented on a testbed containing 10 laptops, each equipped with WiFi and ZigBee interfaces. The implementation does not need any change in the underlying MAC protocol, and thus is compatible with existing standards. The experiment results show that, out proposed system has up to 49% higher throughput than the IEEE 802.11g protocol.

To further evaluate the performance of our proposed system in a large-scale network and in a hybrid network with both nodes running our system and nodes running the IEEE 802.11 protocol, a detailed ns2-based simulator is built and extensive simulations have been conducted. The results show that our proposed system can increase network throughput by 18%, reduce power consumption by 32%, and achieve much better fairness, compared to the IEEE 802.11 protocol. The results also show that the performance of every individual node in the hybrid network can be improved, and the performance of the overall network increases as the fraction of nodes running our system increases.

1.2.3 Related Work

In the recent years, numerous efforts have made to improve the performance of IEEE 802.11 network. To improve throughput, various TDMA-typed protocols have been proposed and implemented. Overlay MAC [30] proposes a multi-hop TDMA-typed MAC protocol for IEEE

802.11 network by employing a distributed algorithm to allocate time slots among nodes and also implements a precise control over time slots. Koutsonikolas et al. [19] design and implement a TDM MAC protocol for multi-hop wireless mesh networks using a programmable wireless platform. Inspired by the IEEE 802.4 Token Ring protocol, the token-passing protocols, such as [27] and [24], have been proposed to the performance of IEEE 802.11 network. Different from their work, our proposed protocol works transparently to the underlying MAC layer and introduces zero control overhead.

Fairness is also a popular research issue in 802.11 networks. In [13], a combination of an Inter Frame Space (IFS) based Distributed Fair Scheduling (IDFS) with the backbone of IEEE 802.11 DCF is designed to achieve better performance in terms of throughput and fairness. A cross-layer approach, IFS based Distributed Fair Queuing (IDFQ) algorithm, has been proposed in [21]. It is adaptive to the collision state in the system by considering physical characteristics of wireless channel and designs a mapping function at the MAC layer to achieve proportional fairness and improve network throughput. Compared to these methods, our approach of achieving the fairness is purely built atop the 802.11 MAC layer, which is untouched, to guarantee the compatibility.

Moreover, some research has been conducted recently to investigate co-located interfaces to assist WiFi transmission. One of the first work is Blue-Fi [5], which brings forth the idea of using other co-located interface to assist WiFi transmission. It uses the co-located Bluetooth to predict the availability of the WiFi connectivity by using user's trend of repeatedly encountering the same set of bluetooth devices and cell-towers. Different from Blue-Fi, our system uses ZigBee interface, which has a much longer communication range. Thus, it can provide a better communication capability under the mobile environment. Our proposed system is motivated by this feature. Because of using different hardware and methodologies, the accomplishment of Blue-Fi and Z-WiFi are also different. Besides, ZiFi [47] utilizes ZigBee radios to identify the existence of WiFi networks through WiFi beacons, while WiZi-Cloud protocols [16] have been proposed to use WiFi-ZigBee radios on mobile phones and Access Points to achieve ubiquitous connectivity, high energy efficiency, real time intra-device/inter-AP handover. Unlike those work, our work focuses on improving the performance WiFi transmission under the DCF through reducing contention. In general, the previous work targets on saving energy, but our work aims to improve the throughput, power efficiency and fairness.

1.3 Organization of Thesis

The rest of the thesis is organized as follows. Chapter 2 presents the system overview of our proposed VANET-WSN system. Chapter 3 illustrates the detailed design of VANET-WSN system, and Chapter 4 reports the implementation and simulation results of VANET-WSN system. Chapter 5 first presents the preliminaries of our proposed ZigBee-assisted WiFi transmission system, then elaborates our proposed design, and finally analyze co-existence of S-WiFi and Z-WiFi systems. Chapter 6 reports the results of comprehensive simulation and prototype implementation of the ZigBee-assisted WiFi transmission system. Finally, chapter 7 concludes the work and discusses future work.

CHAPTER 2. System Overview of VANET-WSN System

2.1 Network Deployment

The proposed system consists of highly mobile vehicle nodes and static roadside sensor nodes. Each vehicle node has two communication interfaces: a WiFi (IEEE 802.11) interface for communication with other vehicle nodes; and a ZigBee (IEEE 802.15.4) interface for communication with roadside sensor nodes. In our prototype, each vehicle node is an on-car laptop with an embedded WiFi card and an attached Telosb mote [15].

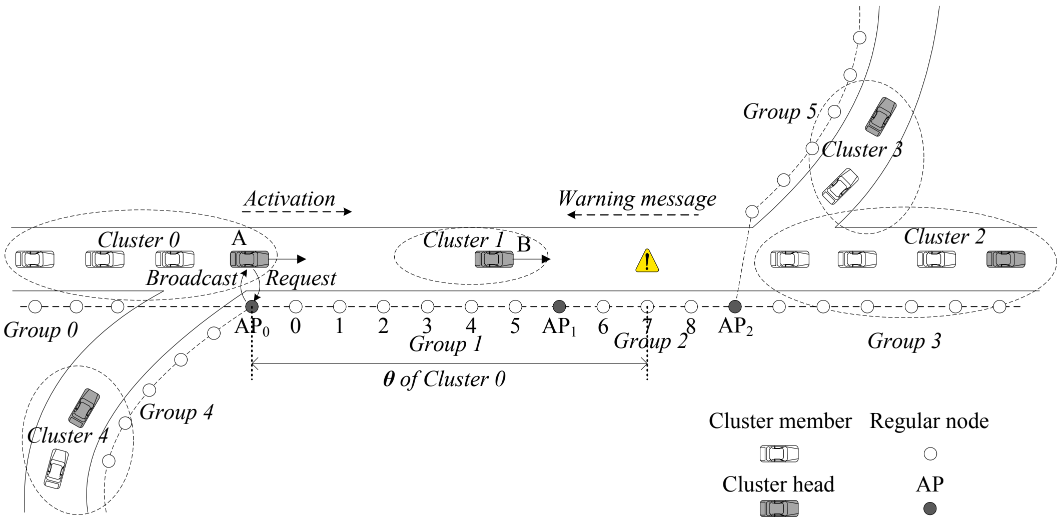


Figure 2.1: Network Deployment

Each sensor node has a ZigBee interface used to communicate with other sensor nodes and with vehicle nodes, and in our prototype, each sensor node is a Telosb mote. Sensor nodes are also mounted with sensors which are used to sense road conditions.

As illustrated in Fig. 2.1, sensor nodes are deployed along one side of the highway. We consider only one-way highways, though the system can be extended to two-way roads. The sensor nodes form a connected network. According to their roles, sensor nodes have two different classifications: the *regular sensor node* and the *access point sensor node* (called AP thereafter), which can sense and relay messages, while APs have extra responsibilities of discovering and communicating with vehicles, and managing the network. APs are much fewer than regular nodes. Regular nodes that are deployed between two adjacent APs along the roadside form a *group*. As shown in Fig. 2.1, one highway may merge into another one, two highways may be connected with a ramp, and one highway may branch into two or more highways; hence, the roadside sensor network is not linear. In our design, the node connected with three or more linear segments must be an AP.

In practice, some roads (e.g., in mountain areas) may be more prone to safety-related events than others; hence, sensor nodes may only be deployed along the roads with high risks. This way, deployed sensor nodes do not form a single connected network, but multiple disconnected networks. Our design is flexible and is applicable to such deployment due to the modularity approach adopted.

2.2 Duty Cycle Scheduling and Warning Message Forwarding

A connected partition of vehicular nodes on a highway forms a cluster. Cluster formation has been widely studied and is beyond the scope of this thesis. Each cluster maintains a *cluster head*, a node which is running at the front of the cluster. It is responsible for communicating with roadside sensors on behalf of the whole cluster. As shown in Fig. 2.1, there are five clusters, where cluster 2 and cluster 3 are connected but they are on different highways.

2.2.1 On-Demand Duty Cycle Scheduling

As illustrated in Fig. 2.2, each AP periodically broadcasts a beacon message. If the AP has buffered some safety-related information that its nearby vehicles should be aware of, it will piggyback these messages in its beacon message. When a passing cluster head hears the message, it sends its registration request to the AP.

In response to the request, the AP generates and broadcasts an activation message. The activation message is then propagated by the roadside sensors hop by hop along the moving

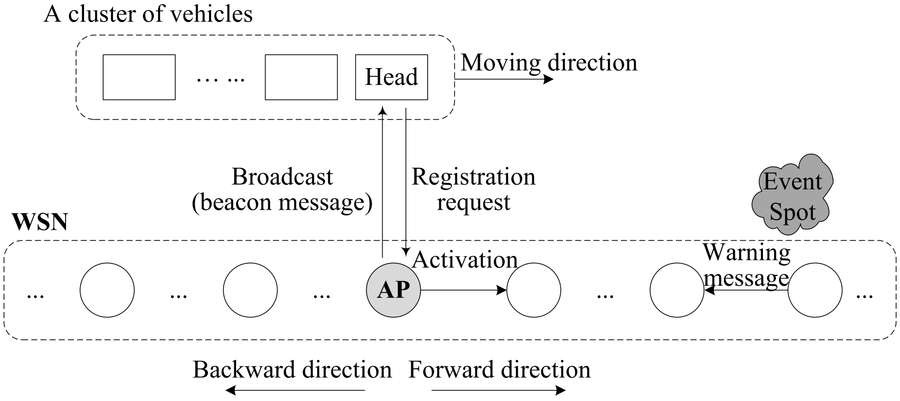


Figure 2.2: Big Picture of the Integrated VANET-WSN System

direction of this cluster (called *forward direction* till it reach a certain number (denoted as θ, a system parameter) of hops away. When each roadside sensor receives the activation message, it become active. In this way, The AP activates sensor nodes that are within θ hops along the forward direction, if these nodes have not been activated yet. After be activated, these activated sensor nodes will be able to proactively monitor the conditions of the roads. To save energy, a roadside sensor is active only when there is a vehicle cluster approaching to its sensing range within θ hops. In order to save energy, if there is no vehicle approaching this sensor, the sensor does not need to be active.

2.2.2 Warning Message Propagation

If a dangerous condition is detected (e.g., a deer is roaming on the road) by the roadside sensor, the detecting sensor node will generate a warning message and propagate it along the direction opposite to the moving direction of the vehicles (called *backward direction* hereafter). The warning message will be propagated along the backward direction until the message reaches the heads of all incoming clusters that requested the activation of the sensor nodes. Then, the warning message can be propagated within the clusters of vehicles by using a certain data dissemination protocol such as [23, 7, 38]. This way, VANET nodes are leveraged whenever possible to reduce the workload of the roadside WSN to save its energy. In this way, the drivers can receive the dangerous road condition before they reach the potential area, such that they have enough reaction time to do some necessary work to avoid accident.

CHAPTER 3. Detailed Design of VANET-WSN System

The above description on duty cycle scheduling and warning message propagation remains high-level. To realize these functionalities, practical and efficient protocols should be designed for scheduling duty cycles of sensors and for propagating activation messages in forward direction or warning messages in the backward direction. Since the duty cycle scheduling and message propagation are not independent of each other, we will study them together.

One big challenge in designing these protocols is that, the forward and the backward propagations take place on the same communication channel, and hence they should be scheduled appropriately to avoid or reduce collisions to minimize both propagation delay and energy consumption. Although CSMA/CA-based protocols are commonly used in wireless ad hoc and sensor networks, TDMA-based protocols are preferred in our system for following reasons:

- As opposed to CSMA-based MAC protocols commonly used in wireless networks, sensor nodes in the proposed system often have very little data to transmit (packets are generated only when cluster heads pass APs or events are detected by sensors). Meanwhile, once there is data to transmit, the data should be transmitted in a timely fashion to guarantee quality of service. If CSMA/CA is adopted, time and energy may be wasted for long idle listening, medium contention, etc.

- To improve network throughput and support real-time data delivery in WSN, TDMA-based MAC protocols [35, 4, 33] have been proposed recently. Although they can achieve real-time transmission, their different application scenarios (e.g., high data rate, special network structure, etc.) make them unsuitable for our proposed system. Moreover, these protocols are for unidirectional communication, while our system requires bidirectional. Thus, designing a new TDMA-based protocol becomes necessary.

- Further, the special network topology in the proposed system can facilitate the application of TDMA-based protocol. Each sensor has a limited number of neighboring nodes, which are pre-determined, making the assignment of the time slots for transmission easier. By carefully assigning the transmission slot, we can avoid or greatly mitigate the hidden terminal problem that is hard to be solved by using CSMA/CA-based protocols.

However, TDMA-based protocols require time synchronization among nodes, which is hard to accomplish in large-scale systems. To accommodate bidirectional communication in a single channel and meanwhile achieve scalability, we adopt the idea of modularity: sensor nodes are divided into groups; within each group, duty cycles of nodes and bidirectional propagations are scheduled to achieve both contention-less communication and energy efficiency; inter-group communication is handled by APs shared by different groups. In our proposed system, we only require nodes within the same group to be time synchronized.

In this section, we first present our proposed intra-group and inter-group scheduling schemes. Then, we discuss the choice of system parameters and bootstrapping of the system.

3.1 Intra-group Scheduling

Sensors in the same group are time synchronized, and the approach to maintain the synchronization is to be presented in Chapter 4. The time is divided into slots of fixed length. During each slot, a packet can be sent from a sensor to its neighbors successfully if there is no interference; hence, we call the length of a slot a *packet time (denoted as τ).* A certain number of slots form a period, and the length of a period is denoted as p. Protocols for duty cycle scheduling and medium access control (MAC) are presented in the following such that, every c_f period(s), a packet can be propagated hop by hop from the most back sensor of the group to the most front sensor along the forward direction, and every c_b period(s), a packet can be propagated hop by hop from the most front sensor to the most back sensor along the backward direction. Here, we call c_f the *forward interval* and call c_b the *backward interval.*

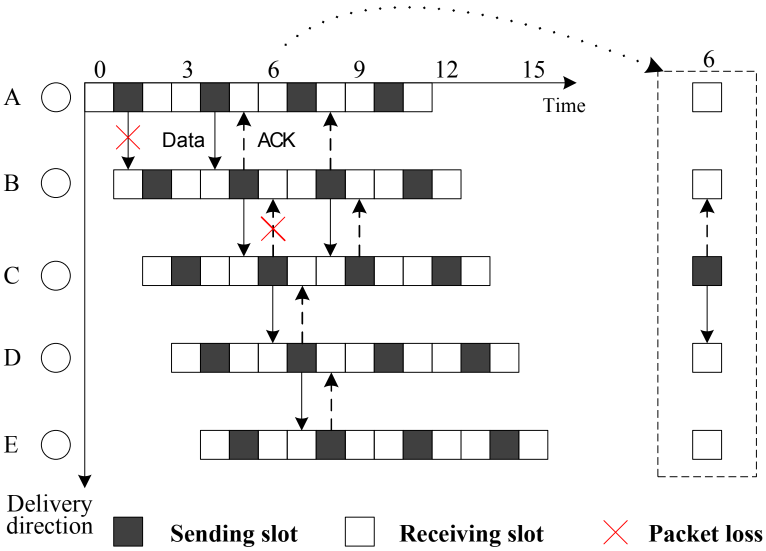


Figure 3.1: Example of Intra-group Scheduling for Forward Activation Msg Propagation

3.1.1 Overview of scheduling mechanism for Forward Propagation

Without loss of generality, let us consider the example shown in Fig. 3.1, where circles $A, B, \cdots, E$ represent sensors in the same group, A is the most back sensor and E is the most front sensor. We want to schedule the duty cycles of these nodes and their communication behaviors such that a packet can be forwarded from A to E hop by hop.

Taking into account the unique characteristics of the network topology, we adopt the following methods to design forwarding propagation protocol which has no contention, high energy efficiency and low propagation delay:

Firstly, TDMA-based access control is adopted to eliminate contention. For each sensor, a certain number of slots are reserved for it for sending or receiving. The reservation of slots follows the following rule: During the slot reserved for node X for sending, none of its one-hop and two-hop neighbors is allowed to send packets. For the example in Fig. 3.1, during the slots for sensor C to send packets, sensors A, B, D and E are not allowed to send packets. This way, contention (even the hidden terminal problem) can be eliminated.

Secondly, the broadcast nature of transmission is leveraged to speed up packet propagation and reduce acknowledgement overhead. Specifically, after a node has received a data packet from its previous hop, it forwards the packet to the next hop immediately in the next slot. Due to the broadcast nature of transmission, the data packet can also reach the previous hop, serving as the acknowledgement. If the packet cannot reach the previous hop due to errors in the channel, the packet that has arrived at the next hop can be propagated further without waiting for the acknowledgement packet being successfully sent to the previous hop.

Thirdly, reserved retransmission slots can be dynamically shared among sensors in the same group. For reliability, retransmission slots are reserved for sensors. However, the quality of different links may not be the same and may change dynamically. For example, sometimes the link between sensors A and B may be better than the link between D and E, and vice versa in other time. Considering this, our design can enable sensors to dynamically share a certain total number of retransmission slots.

3.1.2 Scheduling Detail for Forward Propagation

The forward scheduling protocol is detailed as follows.

- *Reservation of slots*: The most back sensor is assigned with $3(r+1)$ sequential slots, where r is the system parameter specifying the maximum times to retransmit a packet by all nodes in the group, which we call *retransmission quota* hereafter. Without loss of generality, we call the first slot assigned to the node as slot 0, and the remaining slots are called slot $1, 2, \cdots, 3(r+1)-1$, respectively. Slots $3i+1$ ($i = 0, \cdots, r$) are reserved for sending while others are reserved for receiving. If we use R to represent a slot for receiving and S to represent a slot for sending, all these slots can be represented as a sequence of $r+1$ RSR's. For each of the remaining sensors in the group, it is also assigned with $3(r+1)$ sequential slots of the same sensing/receiving pattern, except that its first slot is one slot later than that of its previous hop. In the middle of Fig. 3.1, the scheme for slot reservation is shown for a group composed of 5 nodes and parameter $r = 3$.

- *Sending of a packet*: If a sensor has a packet to propagate, it will send it out at the first sending slot. If it overhears the forwarding of this packet or receives an acknowledgement in the next slot, which is reserved for receiving, from the next hop, the transmission is successful. Otherwise, it will retransmit the packet in the next sending slot. The procedure continues until the transmission is successful or the slots reserved for sending have been used up. In the case that the reserved slots have been used up, the packet can be transmitted in the next reserved propagation time (i.e., nearly $c_f \cdot p$ time later).

- *Receiving/forward of a packet*: If a sensor does not have any packet to send, it will listen in the first slot. If it does not hear anything from its previous hop, it can go to sleep in the following two slots since it can be predicted that it will not have any sending or receiving in the next two slots. If it receives a packet from its previous hop, it will transmit the packet to next hop immediately in the next slot, which is a slot reserved for sending. Then the follow-up procedure for checking if the packet has been successfully sent and retransmitting the packet is the same as in the part of *Sending of a packet*. Note that, if its forwarding is not overheard by its previous hop, the previous hop node may resend the packet. In this case, this forwarding node should be able to identify the duplication; then, it will send a dedicated acknowledgement packet to its previous hop in the next sending slot.

Note that, if multiple sensors in the group have packets to send, these packets can all be propagated except that, some sensor in the middle may have multiple packets to send/forward. In this case, it can merge these packets into one if possible and send it, or send these packets one by one.

An example for packet sending and forwarding is also shown in Fig. 3.1, which is explained as follows. Sensor A wants to send a packet to sensor E. It starts the transmission at its first available sending slot, slot 1. However, this packet gets lost. Hence, A will not receive the acknowledgement from B during the following receiving slot, so it retransmits the packet in the next available sending slot, i.e., slot 4, and it succeeds. Upon receiving the new packet, B immediately forwards the packet, which serves as both data packet to downstream node (C) and acknowledgement to upstream node (A). This sending packet has been received by C but is not acknowledged successfully. Thus, B assumes that C has not received the packet and retransmits that packet. This retransmission packet can be overheard by A and C. A simply ignores it while C attempts to resend the acknowledgement. Due to the good link quality in following propagation, the packet can eventually reach E even before every packet has been successfully acknowledged.

3.1.3 Scheduling Detail for Backward Propagation

With the same idea, the protocol for backward warning message propagation can also be designed similarly. The most front sensor is assigned with $3(r+1)$ sequential slots following the RSR pattern. For each of the remaining sensors in the group, it is also assigned with $3(r+1)$ sequential slots of the same sensing/receiving pattern, except that its first slot is one slot later than that of its following hop. Fig. 3.2 has shown an example of backward propagations next to forward propagations, which is explained as follows.

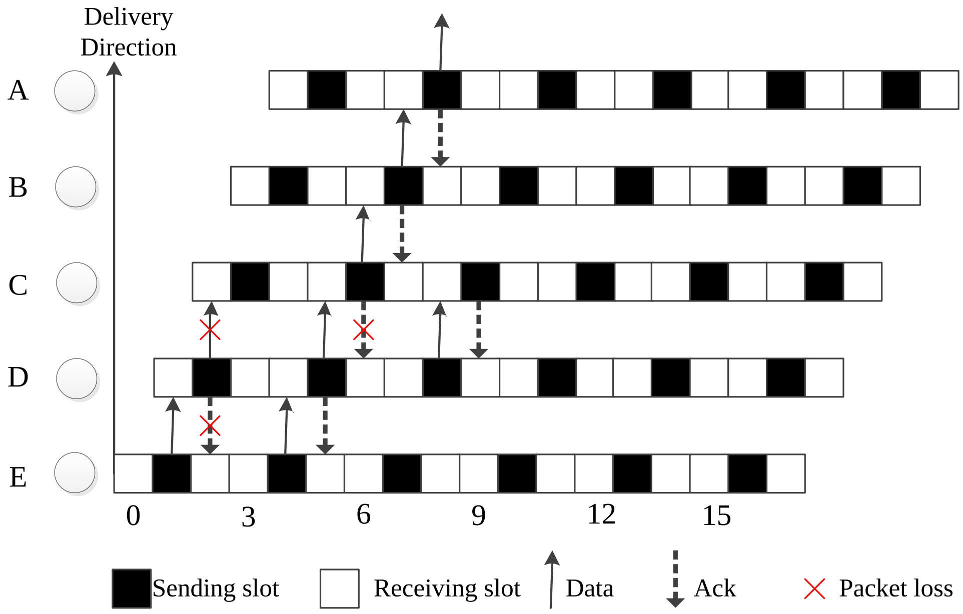


Figure 3.2: Example of Intra-group Scheduling for Backward Warning Msg Propagation

Sensor E wants to send a warning message to sensor A. It starts the transmission at its first available sending slot, slot 1. Sensor D receives the message at slot 1 and immediately forwards the message, which serves as both data message to upstream node (C) and acknowledgement to downstream node (E). Unfortunately, neither the transmission to upstream node nor the transmission to downstream node succeeds. At slot 4, node E observes that it hasn't received the acknowledgement from node D. Node E retransmits the warning message to node D at slot 4. Since node D does not receive acknowledgement from node C, it retransmits the message at slot 5. Both node C and node E hears the packet retransmitted by node D. Upon receiving the new message, C immediately forwards the packet, which serves as both data packet and acknowledgement. This sending packet has been received by B but is not acknowledged successfully. After node B receives the message, it transmits it to node A. Finally, node A receives the message at slot 8. Node D retransmits the message since it does not receive the acknowledgement from node C, and finally get acknowledged. Due to the good link quality in previous propagation, the message eventually reaches A even before every packet has been successfully acknowledged.

In order to save energy, a sensor can turn off its radio during the slots that are not reserved for sending or receiving.

3.2 Inter-Group Scheduling

The scheduling of each group is made independently. When two groups are connected together at an AP, an issue arises: how can the AP successfully pass packets from one group to another with low delay? To address this issue, the AP needs to cooperate with its neighboring regular nodes (called *boundary nodes*, for example, nodes 5 and 6 are boundary nodes of AP_1 in Fig. 2.1) as follows.

The AP needs to know the schedules of its boundary nodes. For this sake, the boundary nodes periodically tell the AP their schedules, by either explicitly sending the schedule or implicitly piggybacking it in the data packet. Knowing the schedules of its boundary nodes, the AP should be active when any of its boundary nodes is active. This way, packets sending to the AP will not be missed if no collision occurs.

The AP also follows the protocol below to ferry packets between groups:

(i) Initially, the AP is in the *idle* state. Suppose the AP is connected with multiple groups, we call a group connects to it on the backward direction as its *upstream* group and a group connects to it on forward direction as its *downstream* group. For example, in Fig. 2.1, Group 1 is a upstream group of AP_1 while Group 2 is a downstream group of AP_1. When the AP receives a forward (backward) packet from its upstream (downstream) group, AP is bound to delivering the forward (backward) packet and hence sets itself to the *forward (backward) state*.

(ii) The AP in *forward (backward) state* is dedicated to delivering the forward (backward) packet. Any incoming backward (forward) will be just buffered and not acknowledged.

(iii) The AP will make an attempt to send the forward (backward) packet to the downstream (upstream) group if (a) any boundary node is in its forward (backward) receiving slots and (b) the last attempt was two time slots away from the current attempt to ensure AP to have enough time to get the possible acknowledgement.

(iv) Step (iii) is repeated until the AP has got an acknowledgement from its downstream (upstream) group. Then, AP will check its buffers to see if there is any packet ready to be delivered. If so, step (iii) and (iv) will be repeated; otherwise, the AP goes back to step (i).

Fig. 3.3 shows an example. B_u and B_d are two boundary nodes of the AP, whose schedules are shown in the figure. At time t_0, B_u sends a forward packet (pkt_0) to the AP. Since the AP is in the idle state, it switches to the forward state and sends out the packet acting as acknowledgement. This packet can be also received by B_d as well. However, since B_d is now in backward phase, it will just buffer this packet without acknowledging it. Suppose at time t_1 B_d wants to send a backward packet (pkt_1) to AP. Since AP is now in the forward state, it will also just buffer this packet without acknowledging it. After some time, the forward phase of B_d becomes available. Then, the AP sends pkt_0 at B_d's first available receiving slot. Note that, even if this packet can not be successfully transmitted to B_d, B_d can still send out the acknowledgement for the previously buffered packet. Upon receiving the acknowledgement from downstream group, the inter-group delivery of that packet is accomplished. At that time, the AP checks its buffer and finds the backward pkt_1 is there to be delivered. Then, it changes to backward state and starts another inter-group delivery.

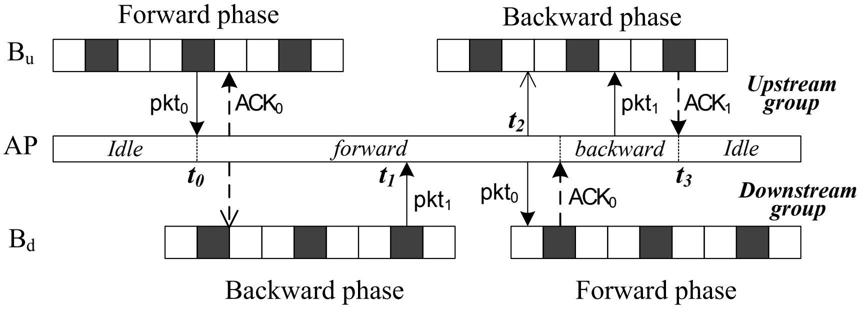


Figure 3.3: Example of Scheduling for Inter-group Communication

Resolution for Extreme Collisions: In some extreme case, data packets from two boundary nodes may arrive at the AP simultaneously and the schedules of these two boundary nodes match exactly. Then, the AP may never receive the data packet from either node because collision always exists. In this case, the above scheme fails. Thus, we propose the *yield* mechanism to deal with this situation. The basic idea is to let one boundary node yields to the other when they do not receive the acknowledgement from the AP for a certain number of times (which indicates the possible occurrence of collisions). At this time, one boundary node will start resending the packet once every two sending slots, while the other remains the same. Note that the working of this mechanism can be coordinated by the AP.

Broadcast of AP Beacon Messages: When none of the boundary nodes of the AP is in their reserved slots for backward or forward propagation slots, shown as "BC" blocks in Fig. 3.4, the AP can pick some time point to broadcast beacon messages such that passing cluster heads can discover and contact with the AP. The interval between two consecutive beacon messages should be short enough to ensure that a passing cluster head cannot miss it during its stay within the transmission range of the AP. During the time that the AP does not broadcast beacon messages and none of its boundary nodes is active in their forward/backward propagation, the AP can go to sleep to save energy.

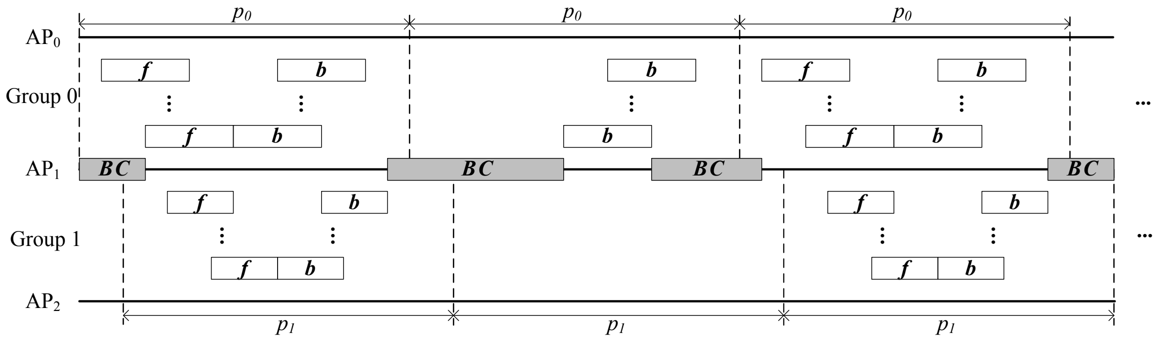


Figure 3.4: Forward and Backward Propagation Scheduling in a System with Multiple Groups (Group 0: $c_b = 1$ and $c_f = 2$; Group 1: $c_b = c_f = 2$; two groups have the same period $p_0 = p_1$; BC: slots that the AP_1 can use to broadcast beacon messages)

3.3 Discussion on System Parameters

In this section, we show the relation between various system parameters by presenting our derived equations.

3.3.1 System Parameters n and r

Based on the probability model for estimating the packet loss between two nodes in [8], we can derive the following inequality by requiring the expected number of *retransmissions* that a packet needs to cross a group should be no greater than the retransmission quota r. n is the number of sensors in a group and p_i is the packet loss ratio of node i in the group.

$$\sum_{i=1}^{n} \frac{p_i}{1-p_i} \leq r \tag{3.1}$$

If we assume that each sensor node has the uniform packet loss ratio, say $\bar{p}$, then from the Equation (3.1) we can get the lower bound of r.

$$r = \frac{n\bar{p}}{1-\bar{p}} \tag{3.2}$$

3.3.2 System Parameter c_f and c_b

Here, we only show the impact of c_f on the delay of forward propagation, and the impact of c_b on backward propagation is similar. The propagation delay within a group includes two parts: the intra-group delay (among regular nodes) and inter-group delay (at AP). Considering the *worst* case that the forward phase of the downstream boundary node is just over when the packet reaches the AP, the propagation speed within a group (including one AP), denoted by v_p, is

$$v_p = \frac{(n+1)I}{(3r+n+2)\tau + c_f p}, \tag{3.3}$$

where τ is the length of a slot, p is the length of a period (defined before) and I is the distance between two neighboring sensors. From this equation, we can see that by changing c_f we can dynamically control the propagation speed and further satisfy the delay requirement.

3.3.3 System Bootstrapping and Maintenance

System bootstrapping is conducted group by group. Initially, there is only one starting AP_0 on the road. The administrator of the roadside WSN determines the system parameters of the newly deployed group, which include n, r, c_b and c_f. Then the sensors in this group are preloaded with these parameters and are deployed one by one. Next, another AP (denoted as AP_1) is deployed. After the deployment, all nodes in the group synchronize their time clock through exchanging a message between each pair of neighboring nodes. After deployment, the clock of each sensor will differ after some amount of time due to clock drift, caused by clocks counting time at slightly different rates. All nodes except the boundary node follows the previous AP within the same group synchronize itself with its previous node when it receives activation message from its previous node.

Due to the modularity nature of our system, system maintenance (e.g., sensor addition, replacement, system parameter resetting, rotation of APs etc.) can be performed within groups autonomously. The basic standpoint is that any adjustment to the network can be bounded within two APs (or the group), making the impacts local. Since their underlying principles are similar to the above system bootstrapping, we will not elaborate in this thesis.

CHAPTER 4. Implementation and Simulation of VANET-WSN System

4.1 Implementation and Field Tests

We implement the proposed system using network embedded system C (*nesC*) programming language and test it in the field. The system is run on TinyOS, which is a free and open source component-based operating system and platform targeting wireless sensor networks (WSNs). In the implementation, three different type sensor nodes, namely, AP, regular node and vehicle node, have been prototyped. The vehicle node is implemented atop a laptop injected with a Telosb mote. AP and regular nodes are implemented atop Telosb motes with WiEye passive infrared sensors mounted. The Telosb motes run TinyOS-2.1.0 in Ubuntu Operating System. Table 4.1 shows the image sizes of the software modules developed for these components:

Table 4.1: Image size of VANET-WSN implementation software

Component	ROM	RAM
AP	33274bytes	1604bytes
Regular node	32194bytes	1600bytes
Vehicle node	17558bytes	1086bytes

4.1.1 Implementation Detail

- *TinyOS and nesC programming language*: TinyOS[14] is a free and open source component-based operating system and platform designed specifically for wireless sensor network nodes. Sensor networks consist of many tiny, low-cost, low-power, short wireless communication range sensor nodes. Each node of the wireless sensor network execute concurrent, reactive programs that must operate with severe memory and power constraints. TinyOS is an event-driven operating system designed for sensor network nodes that have very limited resources (e.g., 8K bytes of program memory, 512 bytes of RAM). TinyOS is an embedded operating system written in the nesC programming language as a set of cooperating tasks and processes[43]. NesC is an extension to C programming language designed to embody the structuring concepts and execution model of TinyOS.

- *Time syncrhonization of roadside sensor nodes*: For TDMA-based protocols to work, time synchronization is a prerequisite. To realize time synchronization, we use two interfaces provided by TinyOS-2.1.0 library: *TimeSyncAMSend* and *TimeSyncPacket*, which provide the primitives to synchronize a group of nodes through exchanging packets. We use the send command provided by TimeSyncAMSend interface and eventTime command provided by TimeSyncPacket interface keep the roadside sensors synchronized. The send command has four parameters, which are receiver address, message content, message length, and event time. The send command sends a regular message, and it also performs sender-receiver time synchronization. The eventTime parameter holds the time of some event as expressed in the local clock of the sender. The receiver can obtain the time of this event (expressed in its own local time) via the TimeSyncPacket interface. The eventTime command of interface TimeSyncPacket can be called by the receiver of the transmitted message. The time of the synchronization event is returned as expressed in the local clock of the caller. This command must be called only on the receiver side and only for messages transmitted via the TimeSyncSend interface. Each node sends the activation/warning message using the TimeSyncAMSend in the forward/backward direction, and each receiver synchronized its local clock with the sender node when it receives the message. In this way, we keep each group of the roadside sensor to be synchronized.

- *Business Logic of vehicles*: The vehicle sensors keep listening the message sent out by roadside sensors. If vehicle node receives warning messages from roadside sensors, it displays the warning messages on the laptop screen. If vehicle node receives beacon messages periodically broadcast by roadside sensors, it broadcasts a request message to roadside sensors, and informs roadside sensors that it will pass along and asks sensors activate themselves if they are inactive. If drivers observe some dangerous road condition, it broadcasts a warning message about the road condition to roadside sensors using ZigBee interface.

- *Business Logic of Regular Nodes*: Each regular node keeps two timers, which are Forward Phase Timer(FPT) and Backward Phase Timer(BPT). FPT (BPT) is responsible to broadcast and receive forward activation (backward warning) messages. When a sensor node except the most back (front) one receives an forward activation (backward warning) message from previous (following) node, it synchronizes itself with the sender node, at the same time, it save a local copy of the received message. The sensor will broadcast the local copy message at sending slots under the following situations: a).The local copy message has not been broadcast. b).The local copy message has been broadcast, but the sender has never received the acknowledgement. We also implement the *yield* mechanism to deal with the heavy condition at boundary nodes. We let previous boundary node yields to the other when they do not receive the acknowledgement from the AP for a certain number of times (which indicates the possible occurrence of collisions). At this time, the previous node will resend the packet once every two sending slots, while the other remains the same.

- *Business Logic of APs*: In order to be synchronized with both boundary node from previous group and boundary node from following group, APs need to keep four timers. PreForwdTimer (PostForwdTimer) is responsible to broadcast and receive activation (acknowledge of activation) message from previous (following) boundary node. PostBackTimer (PreBackTimer) is responsible to broadcast and receive a warning (acknowledge of a warning) message from following (previous) boundary node. APs broadcast beacon message periodically to vehicles and keep listening the request message and warning message from vehicles. APs generate new activation message and propagate the message in the forward direction to activate other roadside sensors if they receives some request message from vehicles. Another responsibility of APs is propagating activation message in the forward direction and warning message in the backward direction.

In order to collect data, we implement another two type nodes, which is start node and stop node. At the beginning of each experiment, the start node sends a message to the first AP, and wake up the whole system. At the end of each experiment, the stop node sends a stopping message to the last AP to terminate the whole system.

4.1.2 Feild Tests

The major purposes of field tests are two folds. The first is to test if the proposed system works in field, and the second is to find out the impact of real environmental factors on the proposed system, especially on the communication of the system. Hence, we conduct two sets of field tests in a large open parking lot: One set of experiments are to test how the whole system works. The test is conducted in two scenarios: there exist intensive WiFi traffics nearby and there does not. Another set of experiments are conducted to measure the impacts of environmental conditions on communication between two Telosb motes when the interference level varies. Here, we elaborate the findings and results from the first set of experiments, while the results from the second set of experiments are used as inputs to our simulation which is discussed in Chapter 4.2.

Two groups of Telosb motes (including totally 9 motes) are deployed along the roadside in a large open parking lot. The motes cover the length of 480 meters, the distance between two adjacent motes is 60 meters. A vehicle repeatedly runs along the motes. Whenever the vehicle enters the road from one end and is discovered by an AP, the AP will wake up all the rest motes to start sensing. Warning messages are generated by the AP located at the other end of the road at a constant frequency, and the messages are propagated to the AP who discovers the vehicle and then is delivered to the vehicle.

Other experimental parameters are as follows. Transmission range is 100 meters. AP broadcasts beacon message every 10 seconds and an event is generated every 20 seconds. Retransmission quota (r) is fixed at 3 while the number of hops to activate (θ) is set to 8. Vehicle speed is about 20miles/hour. Each test lasts for 20 minutes.

4.1.3 System Performance with Interference

As WiFi communication is expected to co-exist with the proposed system, we first test the working and performance of the system when WiFi communication exists. For this sake, two laptops equipped with WiFi cards are put near each of the APs, respectively, to serve as interferers. To make the interference strong, about 10Mbps traffic is exchanged between them, and the traffic is generated by using LAN Traffic V2 [26]. In the experiment, WiFi communication uses channel 6 (the default channel) and ZigBee uses channel 26. For comparison, we also conduct experiment for the situations of no WiFi traffic.

In these experiments, we set the forward/backward interval (c_f/c_b) and group size (n) to 3. We measured the average per-hop delay for the forward/backward message propagation, which are denoted as $D_{Forward}$ and $D_{Backward}$ respectively. The results are shown in the table 4.2. From the results, we can see that the average delay measured with interference is slightly (i.e., between 5% and 9%) higher than that without interference for both forward and backward message propagation. Note that, the simulated interference traffic is intensive. This indicates that the impact of interference on propagation delay is not significant.

Table 4.2: Propagation Delay

$n = c_f = c_b = 3$	With Interference	No interference
$D_{Forward}$(ms)	54.07	**51.06**
$D_{Backward}$(ms)	95.99	**88.31**

4.1.4 System Performance with Varying Parameters

Since the impact of interference from WiFi traffic is insignificant, we conduct more extensive experiments without the interference. In the experiments, we vary the system parameters (i.e., c_f, c_b and n) and measure the propagation delay. The results are as table 4.3 and table 4.4.

As we can see the largest forward propagation delay is about 205ms per hop, which means the speed for propagating activation messages from an AP which detects an incoming cluster of vehicles to other sensors that should be activated is about 293m/s, i.e., 659miles/h, which is

Table 4.3: Forward Activation Message Propagation Delay

$D_{Forward}$(ms)	$c_f = c_b = 2$	$c_f = c_b = 3$	$c_f = c_b = 4$
$n = 3$	63.04	**51.06**	59.57
$n = 4$	205.19	75.73	157.25
$n = 5$	85.76	122.73	102.65

Table 4.4: Backward Warning Message Propagation Delay

$D_{Backward}$(ms)	$c_f = c_b = 2$	$c_f = c_b = 3$	$c_f = c_b = 4$
$n = 3$	111.69	**88.31**	97.30
$n = 4$	540.59	105.39	189.08
$n = 5$	293.40	327.29	424.17

much faster than the speed of a vehicle. The largest backward delay is about 540 ms per hop, which means the speed to propagate a warning message to related vehicles is about 111m/s, i.e., 250miles/h, which is also much faster than the speed of a vehicle.

We can also see that the backward delay is higher than the forward delay. The reason is found to be that, the forward phases of boundary nodes happen to have a better match than their backward phases in our experiments. Consequently, each forward packet arriving at APs can be relayed to the downstream group immediately, while some backward packets have to wait for the next available backward phase of the downstream group. Besides, we can see that the propagation delay goes up as the forward/backward interval increases most of the time. Occasionally, it varies. By analyzing the collected data at each node, we find that the variations are caused by the random packet loss, which affects the average delays.

4.2 Simulation

NS2-based simulation has been conducted to evaluate our design. We evaluate the impacts of system parameters and environmental factors on the system performance. The system parameters include *group size* and *forward/backward interval.* The performance metrics include *energy consumption* (the average energy consumption per hour of all APs and regular nodes) and *propagation delay* (the time from when the event occurs to when the cluster head receives the warning message, which is normalized as delay per hop).

We conduct both *theoretical evaluation* and *empirical evaluation*. For the theoretical evaluation, we vary the system parameters within theoretically-possible ranges to evaluate our system performance. For the empirical evaluation, we follow the empirical traffic data to generate traffic and use the packet transmission traces collected from field experiments.

4.2.1 Setup

Table 4.5 shows the parameters fixed in the simulation. We simulate a highway with more than 200 sensor nodes deployed along one side. Based on field experimental result, we set the packet time (i.e., length of a slot in the proposed protocols) to 25ms. We assume $c_f = c_b$. In addition, since retransmission quota is decided by group size and packet loss ratio as shown in Equation 3.2, we do not explicitly consider it in our simulation.

Table 4.5: Parameters setup for VANET-WSN simulation

Road length	18900m × 20m
Number of hops to activate (θ)	50
AP Beacon interval	600ms
Sensor transmission range	100m
Inter-node distance (I)	90m
Vehicle transmission range	250m
Slot length (τ)	25ms
Average packet loss ratio at sensor and vehicle	15%
Interval between two events	6 minutes
Simulated time	1 hour

4.2.2 Theoretical Evaluation

Since the system performances are associated with three different parameters, we evaluate each performance metric by varying one parameter while fixing the other two. Besides, the arrival rate of the clusters is set to be 2 cluster per minute and the average speed of the vehicles is 30m/s (67.5miles/hour).

4.2.2.1 Group Size n

Fix $c_f = c_b = 5$. In Fig. 4.1, we can see that the impact of group size on delay becomes insignificant as it goes up. This is because, by combining Equation (3.2) with (3.3), we know that v_p converges to a constant value as n approaches infinity and c_f (or c_b) and t are fixed. For the energy consumption on sensor side, we can see that a larger group consumes less energy per node. This is because forming a large group can reduce the number of boundary nodes and APs, which consume more energy than regular nodes, in a given area.

However, we can not conclude from the above results that, the larger the group size is the better performance we can achieve. The major problem of forming large group is that the message propagation delay at APs becomes larger as the group size increases. This means that the activation process becomes slow; hence, a vehicle may move ahead of the activation message, which is not desired for safety. Thus, an appropriate group size should be around 25.

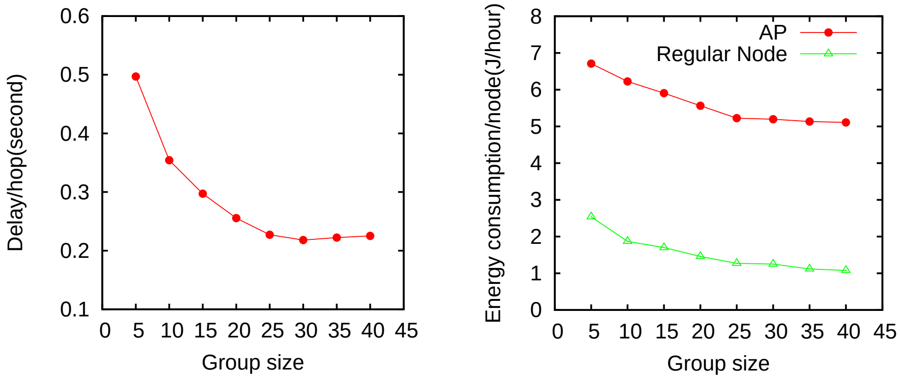


Figure 4.1: Impacts of n on System Performance

4.2.2.2 Forward/Backward Interval c_f and c_b

Fix $n = 20$. It is obvious that by using larger forward/backward interval the sensor nodes can get a larger fraction of time to sleep. Therefore, the delay increases and energy decreases accordingly, which is approximately linear, as shown in Fig. 4.2. Since group size is often pre-defined according to the road topology and packet loss ratio is related to environmental conditions, the forward/backward interval should be the key factor affecting our system performance.

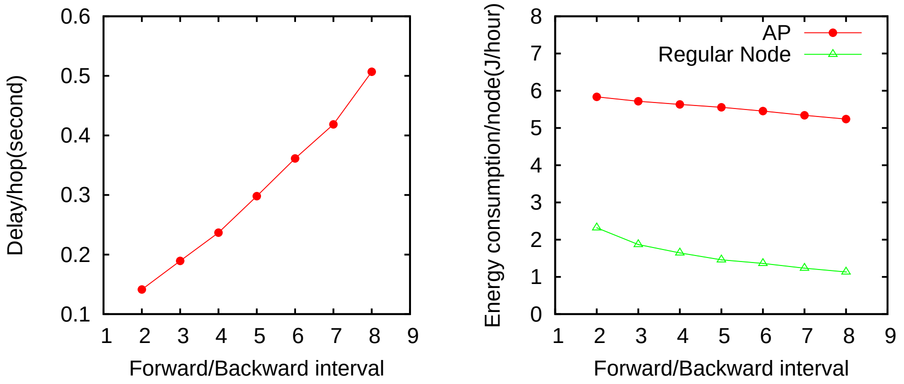


Figure 4.2: Impacts of c_f and c_b on System Performance

4.2.3 Empirical Evaluation

In order for the simulation to better reflect the real-world traffic, we use the empirical vehicle traffic data, measured on I-80 highway in California in [45], to generate traffic for our simulation. Also in field experiments, we log the packet transmission of the sensor nodes under different traffic scenarios. These logs are transformed into the packet loss traces and then fed into our simulator to determine the reception and dropping of the incoming packets, serving as a realistic emulation of *packet loss ratio* for our system.

4.2.3.1 Traffic Generation

As shown in Table 4.6. In our simulation, the vehicle clusters are generated following the three traffic categories proposed in [45]: *night traffic* with very low traffic volume and high speed (1 am - 3 am), *free-flow traffic* with moderate traffic volume and high speed (10 am - 12 pm) and *rush-hour traffic* with low speed and very high traffic volume (3 pm - 5 pm).

4.2.3.2 Packet Transmission Traces

To emulate the road-side interference, we deploy some sensor nodes in the middle of two *regular nodes* to act as the interferers by randomly broadcasting messages (20 packets/sec on

Table 4.6: Traffic Generation

Traffic category	Average speed(m/s)	Density(veh/m)	# of Clusters/ hour
Night traffic	30.93	0.0019	133
Free-flow traffic	29.15	0.0125	57
Rush-hour traffic	10.73	0.0364	<1

average). The reason that we choose sensors, rather than WiFi devices, as interferers is to make the interference more intensive since they share the same channel with roadside sensors in our system. The distance between two nodes is nearly 100m. According to different traffic categories, different numbers of interferers are employed: 2 for night traffic, 4 for free-flow and 6 for rush-hour. For comparison, we also tested no interferer scenario. The number of transmissions for a serial of packets (called *packet trace*) are logged, as shown in Fig. 4.3.

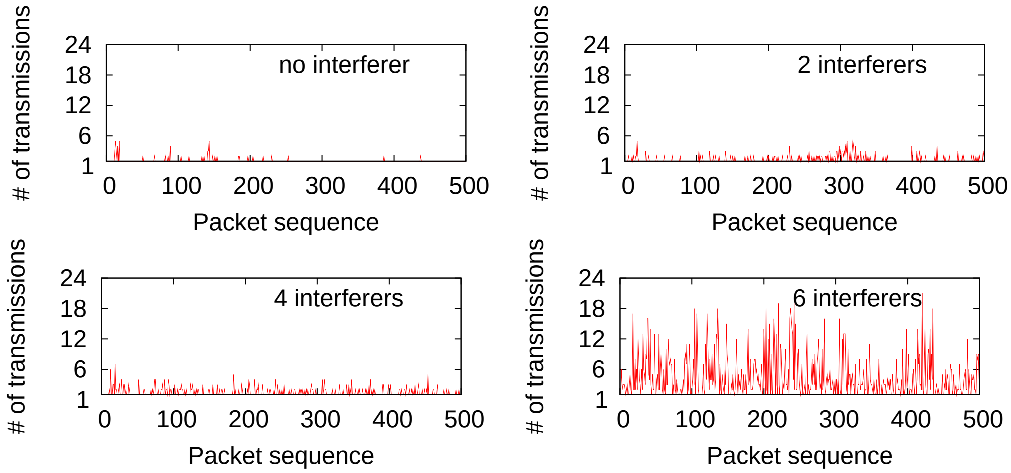


Figure 4.3: Outdoor Experiment: Packet Transmission Traces

4.2.3.3 Results

Based on the empirical traffic generation and collected packet transmission traces, simulations are conducted. The first figure in Fig. 4.4 shows the propagation delay under three traffic categories. We see that forward/backward interval is the dominant factor that affects the delay, especially when group size is large. The delay in free-flow traffic is slightly higher than that in night traffic. However, the delay in rush-hour is much higher than the other two cases due to severe interference. Actually, this does *not* lead to a low system performance, since according

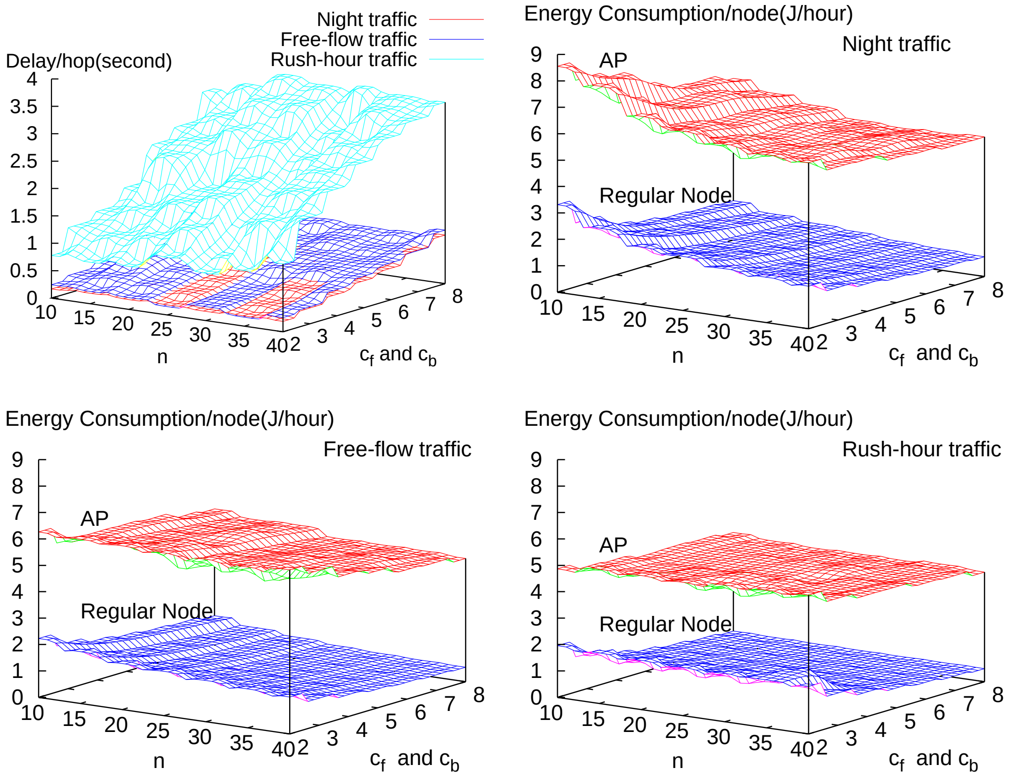


Figure 4.4: Impacts of n, c_f and c_b on System Performance in Three Scenarios

to our proposed system model only cluster head interacts with the WSN. As shown before, cluster arrival rate in rush-hour traffic is less than 1, which means the VANET in rush-hour traffic is almost always connected. The warning propagation can be always conducted via the VANET rather than the WSN.

Therefore, *the utilization of sensor nodes is inversely proportional to the traffic (or cluster) density. The energy consumption of both APs and regular nodes in the rush-hour traffic scenario is the lowest while that in the night traffic scenario is the highest.*

4.2.3.4 Average Propagation Speed and Estimated Node Lifetime

From the simulation, we can also obtain the average message propagation speed in different traffic categories and estimate the lifetime of sensor nodes. The results are shown in the table 4.7 and table 4.8 below. To estimate the lifetime, we take both sensing and communication energy consumption into account under the traffic dynamics of a day. For sensing, experiment has been conducted to measure the sensing energy consumed by passive infrared sensor [25]. Each sampling consumes about 8.7mJ. Each sensor node samples the road condition every 10 seconds when it is activated. The communication energy consumption is obtained from the simulation by following the specification of Telosb mote [15]. Specifically, we consider the power for receiving(62.1mW), transmitting (52.2mW), sleeping (3μW), idling (1.41mW) and transition (426μW), when transmission range is about 75 $\sim$ 100m by default. Each sensor is equipped with 2×AA batteries with 20,000J in total. For system parameters, $n = 20$ and $c_f = c_b = 5$.

Table 4.7: Average Propagation Speed

Traffic category	$\hat{v_p}$(miles/hour)
Night traffic	425
Free-flow traffic	356
Rush-hour traffic	115

Table 4.8: Estimated Node Lifetime

Node type	Lifetime(days)
AP	145
Regular node	545

CHAPTER 5. Design of ZigBee-Assisted WiFi Transmission System

5.1 Preliminaries

5.1.1 System Model and Design Objectives

To run our proposed system, each network node (e.g., laptop) has two wireless interfaces: ZigBee (IEEE 802.15.4) and WiFi (IEEE 802.11). We call such nodes **Z-WiFi** nodes. The WiFi interface is for data transmission while the ZigBee interface is for coordinating node transmission activities. Due to current popularity of the IEEE 802.11 protocol, Z-WiFi nodes may co-exist with the nodes that do not have or use ZigBee but use the **S**tandard IEEE 802.11 protocol. We call such nodes **S-WiFi** nodes.

Our design targets mainly at the scenarios where data traffic is heavy due to high node density and/or high packet transmission rate per node. Network nodes can be either static or mobile. If the nodes are mobile, we assume that mobility is relatively low. For example, nodes are carried by people who stay in conference rooms, libraries, cafe shops, stadiums, etc., where it is typical that a node is static or moves for a while and then pauses for a while and so on and so forth following the well-known random waypoint model. Our design objectives are as follows.

- *High Throughput*: The network nodes gather information such as network deployment, transmission rate of each node by their ZigBee interfaces. Then network nodes carefully schedule the data transmission of WiFi interfaces based on the information collected by the ZigBee interfaces. In this way, our design should reduce the contention among nodes and thereby increase the throughput.

- *Energy Efficiency*: Through reducing the contention experienced by the WiFi interfaces, our design should also decrease the power consumption of nodes.

- *Compatibility*: On one hand, our system should not demand changes in the existing WiFi and ZigBee standards. On the other hand, Z-WiFi and S-WiFi nodes should not harm each other, but should be in the win-win status when co-exist.
- *Fairness*: Our design should organize data transmission of WiFi interfaces in a way that the shared channel is shared relatively fairly among all nodes.

5.1.2 Practical Concerns

To apply in practice such a design where heterogeneous interfaces co-exist and collaborate, two major concerns have to be investigated.

5.1.2.1 Interference

Both ZigBee and WiFi interfaces work on 2.4GHz frequency band. Experiments have been done [17, 34], which show that WiFi can have severe interference on ZigBee communication if their working channels overlap. However, if their channels are separated, the interference becomes insignificant. Although lots of research has been done to evaluate the performance of co-located ZigBee and WiFi, they rarely investigated the scenario under our investigation, where both ZigBee and WiFi interfaces co-exist in the same station. Hence, we conducted experiments to measure the impacts of interference on packet delivery ratio (PDR) of ZigBee interfaces in this scenario.

Both indoor (e.g, library) and outdoor (e.g., parking lot) experiments have been conducted, using two laptops. Each laptop has a ZigBee interface (i.e., attached Crossbow telosB mote) and a WiFi interface. Two WiFi interfaces uniformly generate data traffic in both directions and run the IEEE 802.11g protocol to keep network throughput *saturated*, while one ZigBee interface transmits packets at a constant rate to the other. The channel of ZigBee interfaces is fixed to Channel 26, while WiFi is tuned to Channel 6 (default channel) or Channel 11 (closest to Channel 26).

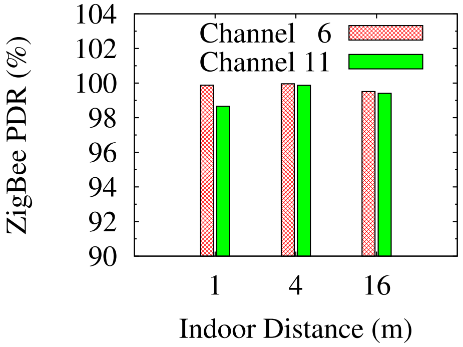


Figure 5.1: Impact of WiFi transmission on ZigBee PDR

The results of indoor experiments is shown in Fig. 5.1, and the results of outdoor experiments are very similar. From Fig. 5.1, by using the non-overlapping channels for ZigBee and WiFi, the packet loss ratio of ZigBee communication is small ($< 2\%$), which not only motivates but also justifies our leveraging of the co-existence of ZigBee and WiFi interfaces to benefit each other.

5.1.2.2 Transmission Range

The other concern is the range difference of ZigBee and WiFi. As the transmission range of ZigBee (i.e., $10 \sim 75$ meters [44]) is shorter than that of WiFi (i.e., $38 \sim 140$ meters [42]), it is impossible to completely eliminate contention by forming clusters using ZigBee. Contention may still exist among different clusters. However, this does not necessarily result in degradation of system performance. Particularly, when the intensity of contention is low (no more than 4 contenders, as demonstrated in Fig. 1.6), the throughput does not decrease, but even slightly goes up, as the number of contenders increases. This is because the default minimum contention window is relatively too large when few nodes contend, leading to insufficient use of the channel. This fact allows the presence of a certain degree of contention among different clusters without degrading system performance.

5.2 Proposed Design

Fig. 5.2 depicts the architecture of our proposed Z-WiFi system, which is built atop WiFi and ZigBee. Thus, it is transparent to and independent of these standards. The *cluster maintenance* component works through communication over the ZigBee interface. A *packet buffering queue* is used to temporarily buffer packets from the upper layer. Through monitoring the status of the queue, packet arrival rate can be inferred, based on which the *transmission scheduler* dynamically computes the TDMA-like schedule for WiFi transmission within a cluster. The schedule is executed by the *packet controller* component which controls the timing and speed for passing packets in the *packet buffering queue* down to the underlying IEEE 802.11 MAC layer. In addition, the *duty-cycle scheduling* component can power off the ZigBee interface when possible to reduce its energy consumption.

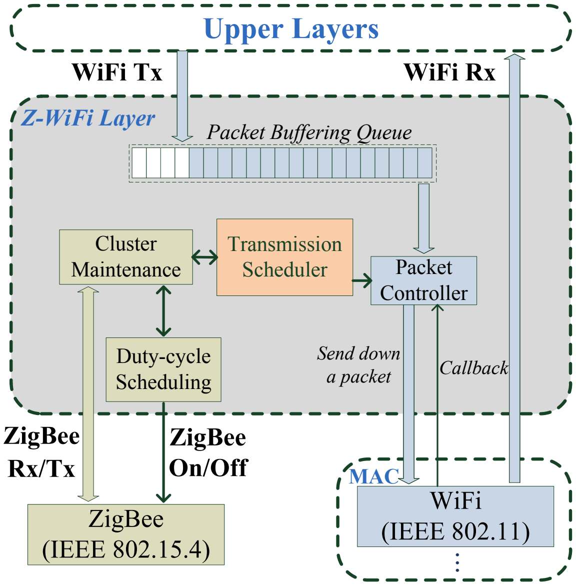


Figure 5.2: System architecture

In this section, we present the deign details of our proposed Z-WiFi network. Briefly, we first present the cluster formation scheme. Then, the intra-cluster and the inter-cluster coordination are elaborated, respectively. After that, a heuristics is designed to determine when our system should be turned on to maximize the performance. For convenience, the notations used in this thesis are listed in the Table. 5.1 below.

Table 5.1: Notation of ZigBee-Assisted WiFi Transmission System

τ_w	Slot length for WiFi transmission
τ_z	Slot length for ZigBee transmission
f_w	Number of slots within a WiFi transmission frame
f_z	Number of slots within a ZigBee duty-cycle frame
i	The unique index of a node in a cluster ($i = 0, 1, ..., M$)
N	The maximum index used in a cluster
r_i	Packet arrive rate of the node with index i
n_i	Number of slots assigned to the node with index i for WiFi transmission
λ	Penetration rate of Z-WiFi nodes
W	Minimum contention window
k	Number of nodes within a cluster
δ	Parameter for assigning n_i
γ	Parameter for turning on/off the ZigBee interface
ω	Parameter for adjusting W

5.2.1 Cluster Formation

To facilitate the coordination of their transmissions for reduced contention, we propose to organize nodes that have potential need for contention into a single cluster through ZigBee communication. Based on existing cluster formation protocols for wireless networks [40], we propose in the following a cluster formation scheme efficient for the scheduling of WiFi transmission.

Initially, each node marks itself as a free node (denoted as **FN**). To obtain information about neighboring nodes, each node periodically broadcasts a *beacon message*, defined as $\langle Node\_id, CH\_id, i, r_i \rangle$, via its ZigBee interface. Here, $Node\_id$ is the network-wise unique id of the sender, CH_id is the node id of its cluster head (denoted as **CH**) if the sender has joined a cluster (otherwise it is empty), and i is a cluster-wide unique *index* of the sender, assigned by the corresponding CH, when it joins the cluster. Besides, r_i is its current packet arrival rate (in the unit of *bits/second*) of the node with index i, estimated through monitoring the status of its packet buffering queue. Note that, if the sender is a cluster member (denoted as **CM**) or a FN, r_i is the packet arrival rate of its own; if it is a CH, r_i is the sum of packet rates of all nodes in its cluster. The usage of r_i and i is to be detailed later.

Based on beacon exchange, each node can maintain a neighbor information list to record the most recent information about its neighbors. If a FN has heard a beacon from one or multiple CHs, it chooses the one whose cluster has the *smallest* packet arrival rate to join. Otherwise, if a FN does not find any CH after a certain rounds of beacon exchange, it announces itself as a CH candidate by broadcasting a *formation* packet piggybacking the number of FNs in its neighborhood. When a node that is not a CH candidate first receives the formation packet, it waits for a certain period of time to overhear other possible formation packets; when the backoff expires, the candidate CH having the largest number of FNs is chosen as its CH and a *registration* packet is sent back to the candidate to join. Upon receiving a registration packet, the candidate node becomes a new CH. In response to each registration from a new CM, the CH sends back an *index* packet, in which a cluster-wide unique index i (i is a positive integer) is assigned to the CM. Note that, the index of a CH is 0.

5.2.2 Intra-cluster Coordination for WiFi Transmission

5.2.2.1 Time Synchronization

Based on the cluster structure, WiFi transmissions of nodes within the same cluster between CH and CMs for reduced contention are coordinated. Each CM is time-synchronized with its CH. This can be achieved by requiring the CH to attach its local time to each message that it sends through its ZigBee interface, and each receiving CM adjusts its local time to align with that of the CH.

5.2.2.2 Packet Arrival Rate Estimation

Each node measures the packet arrival rate (i.e., r_i) at its packet buffering queue, rather than at application layer. When packet buffering queue is full, any incoming packet from upper layer is dropped, which imposes a limit on the value of r_i. Hence, r_i cannot be infinitely large.

5.2.2.3 Slot Assignment and Schedule Representation

With the synchronized time reference, time is divided into frames and each frame is further sliced into slots of equal length. The length of a slot, denoted as τ_w, is the empirical time needed to send a packet through WiFi interface. The CH assigns the slots in each frame to the nodes in its cluster, according to their packet arrival rates. In the following, we show how the CH computes the WiFi transmission schedule (i.e., the slots to transmit), how it is represented and how the CH updates the schedule to its CMs by using the ZigBee interfaces.

A WiFi transmission schedule is represented and sent as a sequence of binary bits, which can be contained in the payload of a single ZigBee packet. A sequence consists of many subsequences of 0(s) separated by a 1. For example, sequence

$$\underline{00000}11\underline{0000}1\underline{000}1\underline{00}1\underline{000}100\cdots 0$$

represents that a WiFi transmission schedule, where each frame has 17 slots, nodes with indices 0, 1, 2, 3, 4 and 5 are assigned with 5, 0, 4, 3, 2 and 3 slots, respectively. Node 0 (i.e., the CH) can perform WiFi transmission during the first 5 slots of each frame, node 1 may not exist or has no packet to send, node 2 can perform WiFi transmission during the 6th to the 9th slot of each frame, and so on and so forth. WiFi transmission schedule periodically is updated and broadcasted by the CH via its ZigBee interface as the packet arrival rate may change in each node.

Particularly, in our experiments, we set the payload size to 28 bytes, which is the default payload size used by TinyOS. Once the payload size is determined, the maximum number of slots in a frame is also determined. We denote the maximum number of slots in a frame as f_w^{max}. Also, we use r_i to denote the packet rate of node with index i ($i = 0, \cdots, N-1$) in the cluster, recalling that each node is assigned a unique index. Let δ ($0 < \delta \leq 1$) be a predetermined system parameter. The number of slots allocated to each node i (denoted as n_i) and the actual number of slots composing a frame (denoted as f_w) is computed as follows:

$$n_i = \left\lfloor \min \left\{ \delta \cdot \frac{r_i}{B \cdot \tau_w}, f_w^{max} \cdot \frac{r_i}{\sum_{j=0}^{N-1} r_j} \right\} \right\rfloor > 0, \tag{5.1}$$

$$f_w = \sum_{i=0}^{N-1} n_i \leq f_w^{max}, \tag{5.2}$$

where B is the WiFi bandwidth. Thus, $r_i/B\tau_w$ represents the expected number of packets sent by node i. The rationale behind the slot computation is of three folds:

- For the sake of fairness, the number of slots allocated to a node is proportional to the packet arrival rate of the node while the total number of slots composing a frame should not exceed f_w^{max}.

- The ratio between the number of slots and the packet arrival rate is determined by system parameter δ. The larger is δ, the longer is a frame and the larger number of consecutive slots a node can use for WiFi transmission, and vice versa. Through our experiments, increasing δ leads to decrease in energy consumption and increase in packet delay, and vice versa. To balance energy consumption, δ is set to 0.2.

- Based on Eq. (5.1), the *clustering condition* can be defined as follows: *a FN node can join or form a cluster only if for any node i (including itself) in the resulted cluster* $n_i > 0$ *can be satisfied.* On one hand, a node with very few packets to send do not need to join or form a cluster and it can just use the IEEE 802.11 protocol as a FN. On the other hand, a node with a high packet rate should not be allowed to join a cluster if its joining makes any existing node in the cluster have *zero* slot to transmit. Thus, after a certain period of time, it will attempt to form a new cluster.

5.2.2.4 Schedule Execution

Ideally, each node transmits data through its WiFi interface only during the slots assigned to it, and one packet uses one slot time (i.e., τ_w) to be transmitted. It follows that n_i packets should be sent down to the underlying 802.11 MAC layer in each frame. However, in practice, this is hardly true, considering the following two facts.

- *Overutilizing scheduled slots*: Due to the inter-cluster contention, it is very likely that the n_{i-1} packets scheduled for transmission at node $i-1$ cannot be transmitted completely when $n_{i-1}\tau_w$ time is used up. Therefore, the buffered packets in the underlying MAC layer may contend or collide with the packets sent by the next node i.

- *Underutilizing scheduled slots*: As the data from upper layer is unpredictable, the packet size could be very flexible. It is different to accurately predict the number of packets that can fit in with the scheduled transmission slots. Thus, it is possible that the transmission of a packet is finished before the corresponding slot expires.

To make full use of each slot, we propose to use the *callback* (i.e., notification of the completion of a packet transmission) from the underlying MAC layer to control the timing for passing packets downwards, as illustrated in Fig. 5.2. Specifically, when the scheduled transmission time (i.e.,$n_i\tau_w$) begins, the packet buffering queue delivers a packet to the MAC layer. As long as the scheduled time does not run out and there is an available packet for transmission, a packet will be pushed down to the MAC layer once the callback of previous packet is received.

5.2.2.5 Duty-cycle Scheduling on ZigBee Interface

To effectively and efficiently communicate among the ZigBee interfaces of the CH and the CMs in a cluster, we propose a duty-cycle scheduling scheme for ZigBee communication.

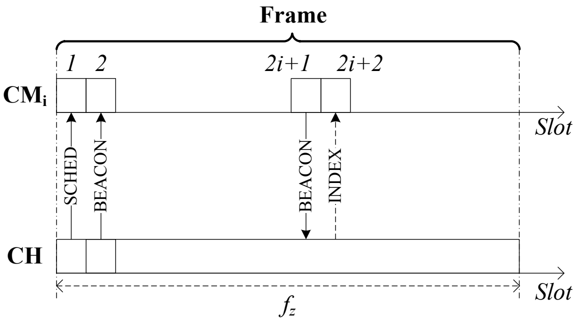


Figure 5.3: Duty-cycle of ZigBee interface

As illustrated in Fig. 5.3, the synchronized time is divided into slots each of length denoted as τ_z, which is the typical length of time needed to process and transmit a maximum ZigBee packet (e.g, 10 $\sim$ 20ms). A certain number of slots form a frame. The number of slots in a frame, denoted as f_z, is a system parameter. During each frame, the communication between the CH and each CM is scheduled based on the index of each CM. Specifically, the CH always wakes up and transmits its own beacon message at the first slot and the WiFi transmission schedule at the second slots. Correspondingly, CMs also wake up at the first two slots to hear these information from its CH. Then, each CM with index i goes to sleep until the beginning of the $(2i + 1)$th slot, when it sends out its beacon to the CH. In the $(2i + 2)$th slot, it waits for possible notification of index change from the CH. As to be discussed in Section 5.2.3, the index of a node may need to be changed when clusters are merged.

In addition, multiple clusters may co-exist and their ZigBee communication could interfere with each other. To deal with this, a node does not start transmission immediately at the beginning of the ZigBee transmission slot; instead, a random backoff is made to reduce collisions.

5.2.3 Inter-cluster Dynamics for Dealing With Mobility

Due to mobility, a CM may move out the range of its CH and join another cluster; a FN may discover a CH and join the cluster headed by that CH; a CH may move into the range of another CH and their clusters may be merged to reduce the number of co-existing clusters and hence inter-cluster contention. In this subsection, we briefly describes cluster switching, joining and merging.

5.2.3.1 Cluster Switching

When a CM with index i finds it has moved out of the ZigBee communication range of its CH, i.e., failing to receive beacon from its CH for a certain time (e.g., $3f_z$ time), it attempts to discover nearby CHs by overhearing beacons. If it finds some CHs, it joins the cluster that has the lowest overall packet arrival rate. If no CH is found in vicinity, it becomes a FN, which can either join another cluster, or form its own cluster. Note that, if a CH fails or is turned off, its CMs will not be able to receive beacon messages from it, in which case they will react as if they have moved out of the communication range of the CH and perform cluster switching as depicted above.

5.2.3.2 Cluster Joining

When a CM or CH becomes a FN, it first tries to join other cluster by turning on its ZigBee and listening for a certain time (e.g., $3f_z$ time). If it finds some CHs in the vicinity, a registration packet is sent. Upon receiving the registration packet, the CH acknowledges that node by replying an *index packet* containing a unique index (typically the *smallest unused* index in the cluster) assigned to that node, if the clustering condition (See Eq. (5.1)) can be satisfied. Once the index packet is successfully received by the FN, it becomes a CM of that cluster. If no CH is found, it starts the cluster formation process as described in Section 5.2.1, if the clustering condition can be satisfied.

5.2.3.3 Cluster Merging

To dynamically minimize the cluster density and hence reduce inter-cluster contention, cluster merging is proposed as follows. As CHs are always awake, they may overhear WiFi transmission schedule packets from nearby clusters. When a CH (CH1) overhears a schedule packet from another CH (CH2), it checks if it can cover more than half of CMs of CH2. If so, merging process will be conducted through the negotiation between these two CHs. As a results, the nodes that are in the cluster of CH2 and covered by CH1 are merged into the cluster of CH1, while the rest of CMs become FNs, which with either join other clusters or form a new cluster later.

5.2.4 Turning on/off ZigBee

Our system is designed mainly to improve WiFi performance in high-contention scenarios, and the IEEE 802.11 protocol can already achieve the optimal throughput when the contention is low. To avoid unnecessary control overhead, we propose a simple heuristic parameter γ for turning off ZigBee interfaces of Z-WiFi nodes when the contention is low and turning on them when the contention is high. The nodes without using ZigBee interface run the IEEE 802.11 protocol.

Specifically, each node records transmission time (i.e., duration from the arrival of a packet to the reception of corresponding ACK) of the most recent outgoing packets. Let T_{pkt} be average transmission time, then

- ZigBee is turned off, if $T_{pkt} < 0.5 \times \gamma\tau_w$;
- ZigBee is turned on, if $T_{pkt} > 1.5 \times \gamma\tau_w$.

$\gamma\tau_w$ represents the expected packet delivery delay when system throughput is saturated. The above conditions attempt to find a range, beyond whose upper bound the ZigBee interface should be turned on to improve performance and below whose lower bound ZigBee interface can be turned off to save energy. Besides, the difference between the upper bound and the lower bound is $\gamma\tau_w$, which is designed to prevent frequent switching caused by random vibration of the network traffic.The selection of γ is to be studied in Section 6.1.

5.3 Co-existence of Z-WiFi and S-WiFi

In this section, we first present an analytical model for our proposed system by taking account of *penetration rate* (i.e., the proportion of Z-WiFi nodes in a system), denoted as λ. Then, based on the model, we enhance our proposed system to be able to operate with the co-existence of S-WiFi nodes.

5.3.1 Model

As outlined in [46, 28], our model is also built upon the Markov chain model for analyzing saturation throughput. Similarly, we assume that a fixed number (n) of nodes (including λn Z-WiFi nodes and $(1-\lambda)n$ S-WiFi nodes) contend with each others and each node always has a packet available for transmission. Let W denote the minimum contention window, m represent maximum backoff stage (i.e., $2^m W$ is the maximum contention window) and M be the maximum retransmission count (e.g., $m = 4$ for data frame and $m = 7$ for RTS frame in the IEEE 802.11b/g protocol).

Suppose the probability that a packet sent by a node collides with others is p. According to [46], the probability (η) that each node transmits a packet in a randomly chosen slot time can be computed as a function of p and W,

$$\eta = \phi(p, W) = \begin{cases} \frac{2(1-2p)(1-p)}{W \cdot B(m+1)+A(m+1)} & \text{if } m \leq M \\ \frac{2(1-2p)(1-p)}{W \cdot B(M+1)+A(m+1)+W \cdot 2^M p^{M+1} A(m-M)} & \text{if } m > M \end{cases}, \tag{5.3}$$

where $A(x) = (1-2p)(1-p^x), B(x) = (1-p)(1-(2p)^x)$ and $p = 1-(1-\eta)^{n-1}$.

The above model is used to analyze the saturation throughput of homogeneous WiFi systems, where all network nodes have exact the same configurations. In our system, we propose to allow different settings for the WiFi interfaces on Z-WiFi and S-WiFi nodes. Thus, the model can be extended as follows.

Virtually, each Z-WiFi cluster is represented as a *super* Z-WiFi node that can transmit all the time. Let W_z and W_s be the minimum contention window used by Z-WiFi and S-WiFi node, respectively. Let C denote the expected number of clusters. Then, we can have

- the probability that a super Z-WiFi node (i.e., a Z-WiFi cluster) transmits in a randomly chosen slot time is

$$\eta_c = \phi(p_c, W_z), \tag{5.4}$$

where

$$p_c = 1-(1-\eta_c)^{C-1}(1-\eta_s)^{(1-\lambda)N}; \tag{5.5}$$

- the probability that a S-WiFi node transmits in a randomly chosen slot time can be computed as

$$\eta_s = \phi(p_s, W_s), \tag{5.6}$$

where

$$p_s = 1-(1-\eta_c)^{C}(1-\eta_s)^{(1-\lambda)n-1}. \tag{5.7}$$

5.3.2 Achieving Fairness Through Adjusting W

5.3.2.1 Formulation of Fairness

From the above-presented model, it is easy to observe that a super Z-WiFi node has the same chance to transmit as a S-WiFi node (i.e., $\eta_c = \eta_s$) if $W_z = W_s$. Suppose there are k nodes in a cluster. Then, the average transmission probability for a Z-WiFi node is η_s/k, which means S-WiFi nodes have an advantage over Z-WiFi nodes in accessing the channel. To deal with this problem and make Z-WiFi nodes have equal chance to access the channel, we propose to dynamically adjust the minimum contention window W_z of Z-WiFi nodes to achieve fair sharing of the channel with S-WiFi nodes.

Assume that all S-WiFi nodes use default the contention window (i.e., 31 for 802.11b and 15 for 802.11g), denoted as W_{def}. That is, $W_s = W_{def}$. We define that Z-WiFi and S-WiFi nodes fairly access the channel if

$$P_c = k \cdot P_s, \tag{5.8}$$

where P_c and P_s are respectively the probability that a super Z-WiFi node and a S-WiFi node can *successfully* transmit in a randomly chosen slot time. Specifically, we have

$$\eta_c(1-\eta_c)^{C-1}(1-\eta_s)^{(1-\lambda)n} = k \cdot \eta_s(1-\eta_c)^{C}(1-\eta_s)^{(1-\lambda)n-1} \tag{5.9}$$

Combining Eq. (5.4), (5.5), (5.6), (5.7) and (5.9), the optimal value of W_z can be only found *numerically* given a particular set of system parameters, similar to [46, 28].

5.3.2.2 Heuristics

Due to the complexity of solving the above equations, we design a heuristics to choose W_z based on a simplified model, where we assume $m = 0$ (i.e., only the first transmission of a packet is considered). Hence, from Eq. (5.4) and (5.6), we have

$$\eta_c = \frac{2}{W_z+1} \qquad \text{and} \qquad \eta_s = \frac{2}{W_{def}+1}. \tag{5.10}$$

Combining these two equations with Eq. (5.9), we can get

$$W_z = \frac{1}{k}(W_{def}-1)+1, \tag{5.11}$$

where $0 < 1/k \leq 1$. Since $W_{def} - 1 > 0$, W_z monotonously increases as $1/k$ becomes larger. Thus, when $1/k = 1$ (i.e., only one node in the cluster), $W_z = W_{def}$ which is the same as S-WiFi nodes; when $0 < 1/k < 1$ (i.e., more than one node in the cluster), $1 < W_z < W_{def}$ which enable Z-WiFi nodes to compete with S-WiFi nodes. The value of W_z for a Z-WiFi node is dynamically chosen as follows.

- According to the neighbor information, each Z-WiFi node computes the expected value of minimum contention window $E[W_z]$ by

  $$E[W_z] = (1/k)^{\omega} (W_{def} - 1) + 1, \tag{5.12}$$

  where ω is a *positive* system parameter to adjust the changing rate of W_z as cluster structure alters. Note that $0 < (1/k)^{\omega} \leq 1$ as well.

- If $\lfloor E[W_z] \rfloor = \lceil E[W_z] \rceil$, then $W_z = \lfloor E[W_z] \rfloor = \lceil E[W_z] \rceil$

- If $\lfloor E[W_z] \rfloor \neq \lceil E[W_z] \rceil$, then

  - $W_z = \lfloor E[W_z] \rfloor$ with the probability $\lceil E[W_z] \rceil - E[W_z]$;
  - $W_z = \lceil E[W_z] \rceil$ with the probability $E[W_z] - \lfloor E[W_z] \rfloor$.

By appropriately choosing the value of ω, Z-WiFi nodes can achieve no worse performance than S-WiFi nodes as to be shown in Section 6.1.

Remarks: The above heuristics is purely based on the *local* information (i.e., k), and thus is efficient to be implemented. Moreover, it enables fair access of the channel not only between Z-WiFi nodes and S-WiFi nodes but also among Z-WiFi clusters of different sizes. Particularly, the heuristics will force a Z-WiFi cluster of more nodes to use a smaller W to compete with clusters of fewer nodes, so as to achieve fairness for the Z-WiFi nodes in different clusters.

CHAPTER 6. Implementation and Simulation of ZigBee-Assisted WiFi Transmission System

6.1 Simulation

To evaluate our proposed system in a large-scale network, we simulate the system with ns2 simulator. In the simulation, the following major metrics are studied:

- *Network throughput* (Mb/s) is the total amount of data successfully transmitted (i.e., ACKed at sender side) in the network. To measure the throughput, each node runs an application which keeps sending UDP packets and by default totally all these nodes generate the data input with an average rate of 20.4Mb/s (i.e., 22 packets/s at each node on average). All the packets have maximum payload size.

- *Energy consumption* (J/Mb) is computed as the total amount of energy consumed by all network interfaces of all nodes divided by the number of Mbs of data that has been successfully transmitted. The energy consumed by the WiFi interface is measured according to the specified power consumption rate of SX-SDWAG 802.11g wireless module [3] (i.e., 1047mW for transmission, 513mW for reception and 420mW for being idle) and the power consumed by the ZigBee interface is measured according to the specified power consumption of CC2420 RF transceiver [2] (i.e., 52.2mW for transmission, 56.4mW for reception, 1.28mW for being idle, 0.06μW for sleeping and 0.06mW for transition).

- *Throughput fairness* is measured with respect to the fairness index (FI) [29], which is defined as $FI_{tp} = \frac{\mu(\chi)}{\mu(\chi)+\sigma(\chi)}$, where $\mu(\chi)$ and $\sigma(\chi)$ are the mean and the standard deviation of χ at all network nodes. χ is the ratio of throughput to input. Obvious, FI_{tp} is between 0 and 1. The more closer FI_{tp} approaches 1, the better is the fairness.

Unless otherwise specified, our simulation use the settings shown in the Table. 6.1. Also, we adopt the random waypoint mobility model, where the pause time is fixed to 20s. Besides collision-caused drops, each node intentionally drops 2% (based on the experiment results shown in Fig. 1.6) incoming packets on ZigBee communication to simulate the packet loss due to interference from WiFi, which use the default IEEE 802.11g protocol.

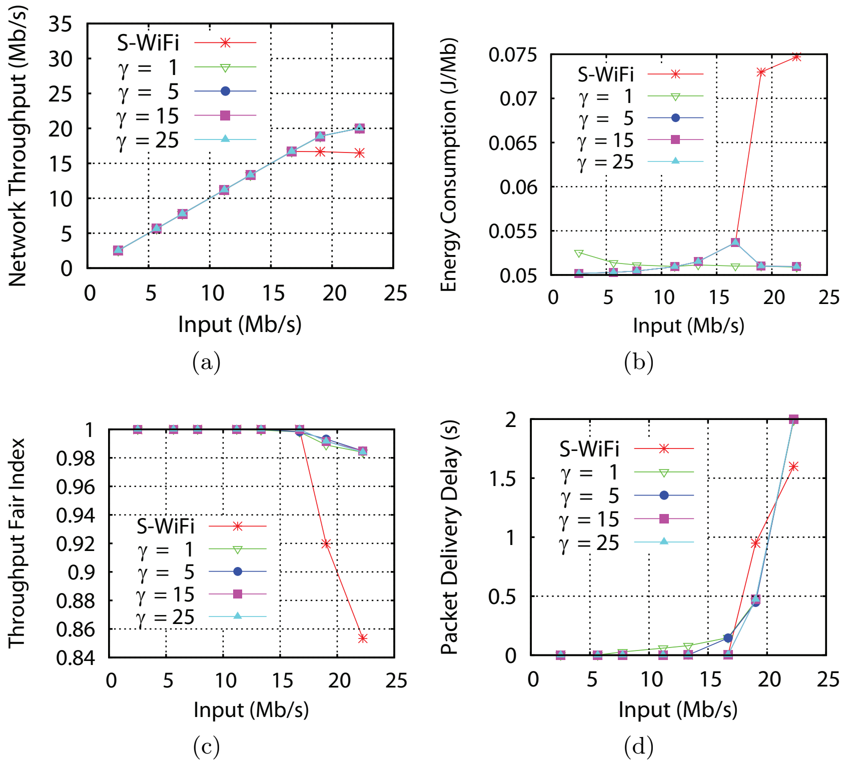
Table 6.1: Parameter setup

Simulation time	1 hour
Number of nodes	50
Network scale	100m × 100m
Average moving speed	2m/s
Range of WiFi (R_w)	120m
Range of ZigBee (R_z)	60m
ZigBee slot length (τ_z)	0.02s
WiFi slot length (τ_w)	0.001s
Length of frame (f_z)	150 slots
ZigBee on/off parameter (γ)	15
Contention window parameter (ω)	2.0
Packet buffer size	50 packets

6.1.1 Comparing with S-WiFi system and studying parameter γ

Recall that parameter γ affects when to turn on/off ZigBee interface of a Z-WiFi node to choose using either our proposed protocol or the IEEE 802.11 protocol. To find the best time to turn on ZigBee so as to maximize the performance, we compare Z-WiFi systems (with $\lambda = 100\%$), configured with four different values of γ (i.e., 1, 5, 15 and 25), with the S-WiFi system (with $\lambda = 0\%$).

From Fig. 6.1a, we can see that when network input is below 17Mb/s, S-WiFi system can almost deliver all incoming packets. When input is beyond 17Mb/s, S-WiFi nodes reach the maximum throughput. At this time, ZigBee interface of Z-WiFi nodes should be turned on to assist WiFi transmission. As shown in Fig. 6.1a and 6.1c, $\gamma = 5, 15$ or 25 can precisely render ZigBee turned on at the right time. This is because, due to accumulated waiting delay in the packet buffer queue, packet transmission delay rises up drastically (from less than one



(a) (b)

(c) (d)

Figure 6.1: Choosing parameter γ by comparing with S-WiFi

millisecond to more than hundreds of milliseconds) once S-WiFi system gets saturated. Thus, large values of γ (e.g., $\gamma > 5$) can work appropriately. Particularly, when ZigBee interface is turned on (i.e., input exceeds 17Mb/s), energy consumption drops rapidly, as shown in Fig. 6.1b, which shows that our proposed system can save energy.

When $\gamma = 1$, ZigBee interface is turned on when network input (i.e., contention) is low. At this time, our protocol cannot help, as the S-WiFi system has already achieve the optimal throughput. Hence, the overhead introduced for ZigBee communication makes Z-WiFi systems consume more energy.

In addition, we also measure average packet delivery delay from application layer, as illustrated in Fig. 6.1d. From the results, setting γ to 15 or 25 can guarantee that Z-WiFi system can achieve no longer packet delivery delay than S-WiFi system when input is below 21Mb/s. When input is above 21Mb/s, our system also becomes saturated and thereby packet delivery delay increases. Note that the packet delivery delay of Z-WiFi system is longer than that of

S-WiFi system only when the throughput of Z-WiFi is higher than S-WiFi.

To summarize from the above results, *our proposed system can improve the network throughput by 18%, reduce the energy consumption by 32% and provide much better fairness, when the network traffic density is high.*

6.1.2 Co-existence of S-WiFi and Z-WiFi Systems

6.1.2.1 Choosing Contention Window Parameter ω

Recall that, we propose parameter ω to dynamically adjust W_z of Z-WiFi nodes to compete with the co-existing S-WiFi nodes. Fig. 6.2a illustrates the impact of ω on overall throughput (including both Z-WiFi and S-WiFi nodes) with different penetration rate. According to the results, the throughput increases as ω becomes larger. This is because using small W can make full use of the channel when contention is very low. When ω is beyond 2.0, the throughput stays approximately constant as the average value W approaches its lower bound.

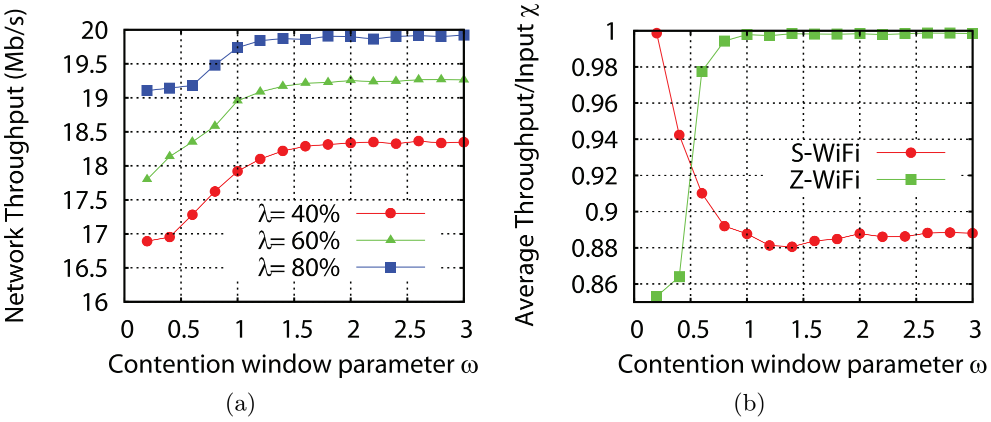
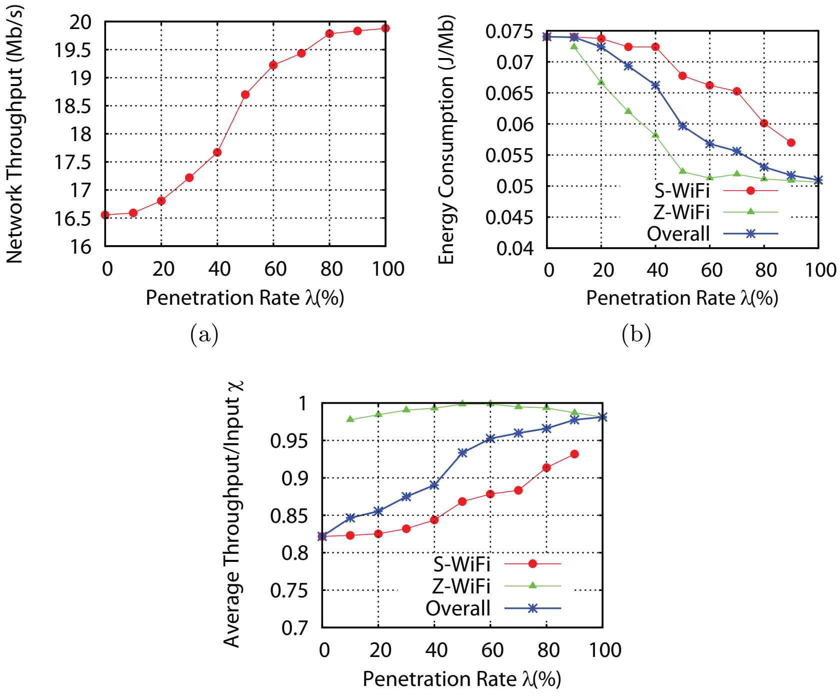


Figure 6.2: Impact of parameter ω on throughput

By fixing $\lambda = 60\%$, we further measured the average throughput/input ratio (χ) of Z-WiFi and S-WiFi nodes, respectively, as ω varies. The result is shown in Fig. 6.2b. Generally, as ω increases, Z-WiFi node gets more chance to transmit while S-WiFi node gets less chance. When $\omega = 0.5$, Z-WiFi nodes have the similar χ to S-WiFi, which means Z-WiFi and S-WiFi nodes fairly share the channel; when ω is beyond 0.5, Z-WiFi nodes outperform S-WiFi nodes. This also holds for the cases of other penetration rates. To obtain the maximum overall throughput, we choose $\omega = 2$ in our simulation.

6.1.2.2 Impact of Penetration Rate on Performance



(a)

(b)

(c)

Figure 6.3: Impact of penetration rate λ on performance

Fig. 6.3a gives the overall network throughput, which increases as λ approaches 100%. Fig. 6.3c and Fig. 6.3b show the average throughput/input ratio (χ) and energy consumption of Z-WiFi nodes and those of S-WiFi nodes, respectively, as λ changes. Besides, the corresponding overall averages are also shown. From these figures, we can see that both node throughput ratio and energy consumption of Z-WiFi and S-WiFi nodes are improved, as more and more nodes use our proposed system. This is because by using our system the network contention is reduced, which can also benefit the co-existing S-WiFi nodes simultaneously. Similar trends can be seen from the overall performance.

In Fig. 6.3c, as λ increases, χ of Z-WiFi nodes first reaches 1. Then, it slightly drops below 1. This is because when λ becomes larger, the size of each cluster also increases, leading to more schedule inconsistencies (due to emulated packet loss) within the same cluster. As a result, more transmission slots of the nodes in the same cluster may overlap, resulting in more collisions. Thus, χ slightly drops. Moreover, in Fig. 6.3b, the energy consumption of Z-WiFi nodes decreases very slowly when λ becomes larger. This is because the energy has to be consumed by the regular RTS/CTS of the underlying IEEE 802.11 protocol, which sets an upper bound for the energy saving that can be accomplished.

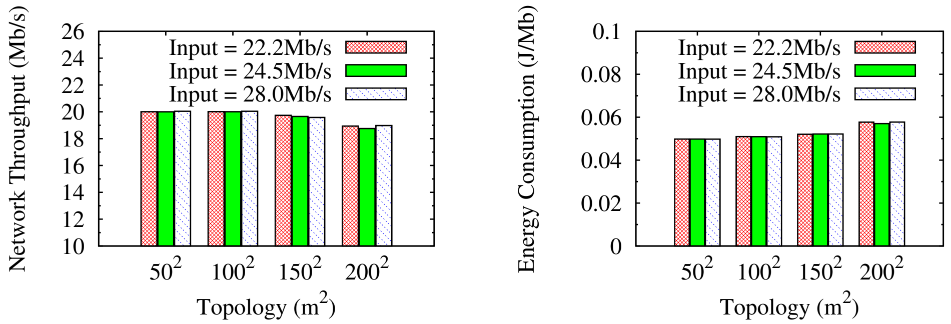
As a conclusion of the above analysis, *using our proposed system can benefit all network nodes (both Z-WiFi and S-WiFi nodes) in terms of both throughput and energy. As penetration rate grows, the performance of both individual nodes and the overall networks is improved.*

6.1.3 Performance with different network scale

Fig. 6.4 shows how our system works with different network scale. From Fig. 6.4a, the throughput slightly decreases as the scale of the network becomes larger, due to the number of clusters increasing. When the number of clusters within WiFi transmission range increases, contention gets more severe, which degrades the performance. However, the number of clusters will not become too large, since cluster merging mechanism is applied, which can ensure the number of interfering clusters close to $\lceil R_w^2/R_z^2 \rceil$ (e.g. 4 under our simulation). For energy consumption illustrated in 6.4b, more clusters consume more energy in transmission coordination and cluster maintenance.

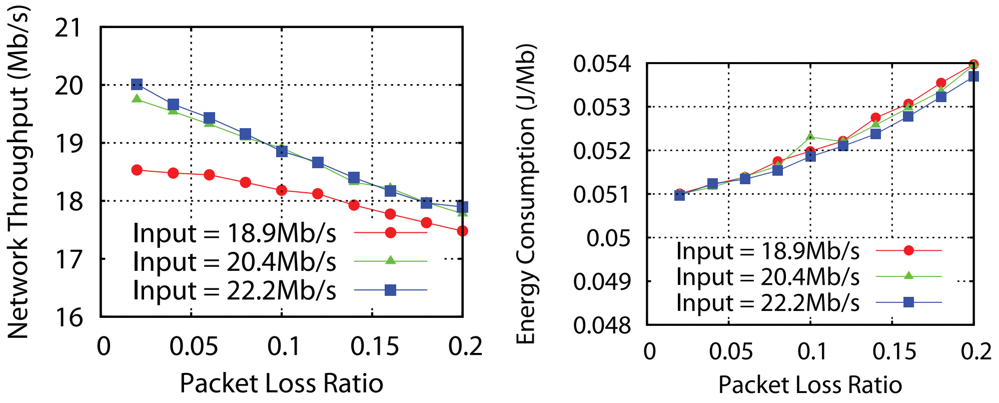
6.1.4 Impact of ZigBee Packet Loss on Performance

Apart from random collision-caused packet loss, we also study the packet loss due to other environmental phenomena (e.g., interference, obstacle, multipath, etc.). Thus, we conduct a simulation by varying packet loss ratio from 2% to 20%. As shown in Fig. 6.5, our performance degrades slightly as loss ratio gets larger. For throughput, it is because of the insufficient utilization of channel caused by increasing delay or error in updating WiFi transmission schedule. The energy consumption increases mainly because of the increased energy consumption for contention caused by schedule inconsistencies.



(a) (b)

Figure 6.4: Impact of network scale on performance



(a) (b)

Figure 6.5: Impact of ZigBee packet loss on performance

6.2 Implementation

6.2.1 Prototyping

As a proof of concept, we implement a prototype of our proposed system. We build a testbed with 10 DELL D-Series laptops (called *nodes* hereafter), each running the Ubuntu Linux 8.10 (kernel 2.6.27-17-generic). Each node is also equipped with a D-Link WNA-2330 Wireless G Notebook Adapter (108Mbps, 802.11g, Atheros chipset, PCMCIA) and a Crossbow telosB mote (i.e., ZigBee interface). Note that *the wireless adapter is built with the state-of-the-art technology, which can deliver higher throughput than standard 802.11g devices.* The scheduling of WiFi transmission is implemented upon MadWiFi [1], an open-source driver for Atheros chipset-based 802.11 Wireless LAN devices. The prototyped ZigBee communication is implemented upon TinyOS 2.1.1 platform, where 10 nodes form a cluster shown in Fig. 6.6. The WiFi interfaces of all nodes run in the ad hoc model and are tuned to Channel 3, and the ZigBee interfaces are tuned to Channel 26; thus, the interference between them is minimal. Besides, the implementation of transmission scheduling is based on *software timer* provided by Linux kernel, which can allow a minimum granularity of 1μs. At the beginning of each experiment, we ask a dummy ZigBee node broadcasting a dummy message as start signal, when other ZigBee nodes receive this dummy signal, they start their ZigBee timer and trigger WiFi interface. In this way, We keep both the ZigBee interface and WiFi interface time synchronized.

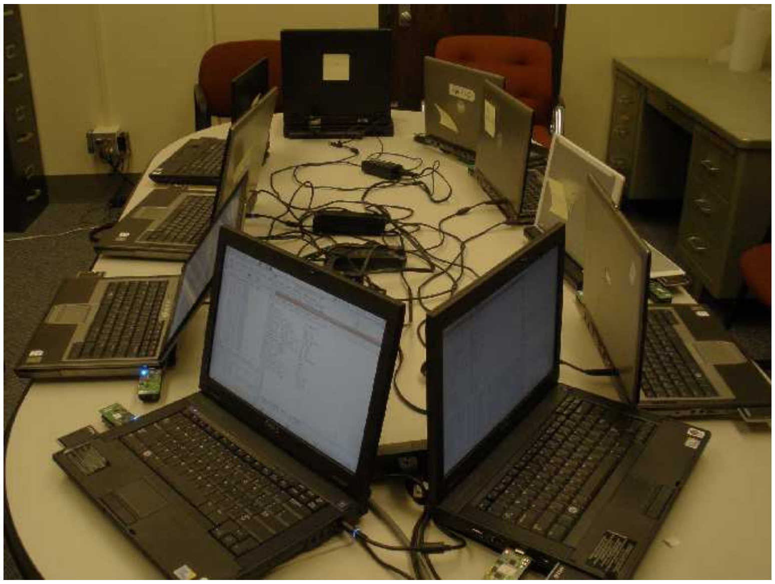


Figure 6.6: Experiment Testbed

Experiments have been conducted on the prototyped system to evaluate the feasibility and the performance of our designed system. For comparison, two sets of experiments are conducted by running the IEEE 802.11 protocol and our proposed system, respectively. Through the experiments, we measure the maximum network throughput as the number of nodes increases from 2 to 10. To measure the maximum throughput, each node generates UDP traffic of 34.8 Mb/s to its neighbor node. Each packet has a payload of 1450 bytes, which makes the overall packet to exactly fit into a single MAC-layer frame. The duration of each experiment run is 5 minutes. The experiment is conducted three times. Besides, $n_i = 10$ and $\tau_w = 0.001$s.

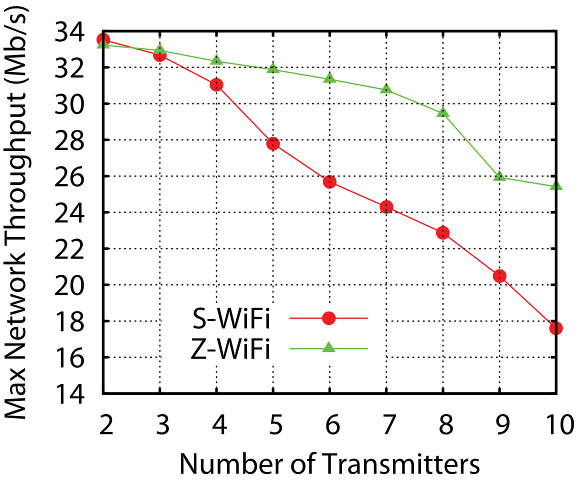


Figure 6.7: Maximum Network Throughput

6.2.2 Experiment Results

The experiment results are shown in Fig. 6.7. In general, compared with the IEEE 802.11 protocol, our proposed system can improve the network throughput significantly. Particularly, when the number of involved nodes reaches 10, the improvement of throughput can be as high as 49.1%. As expected, our proposed system outperforms the IEEE 802.11 protocol when the number of transmitters is large (e.g., more than 4 nodes in our experiment). As that number keeps increasing, the difference becomes more significant because the IEEE 802.11 protocol suffers from severe contention and the throughput drops fast.

Moreover, the standard deviation (STDV) of throughput among different nodes is also measured, as shown in the table below. From the results, we can see that using our proposed system introduces much lower throughput STDV, which indicates better throughput fairness.

# of transmitters	Throughput STDV of S-WiFi	Throughput STDV of Z-WiFi
4	1.1016	0.1780
6	0.8016	0.1281
8	0.7698	0.1775

Through the experiments, our proposed system has been shown to be able to improve throughput significantly and provide fair sharing of bandwidth.

CHAPTER 7. Conclusion and Future Work

7.1 Summary of Thesis

In this thesis, we first studied the big challenges for both sparsely deployed and densely deployed wireless ad hoc network. We then introduced the emerging wireless communication technology, namely ZigBee, which supports low-power, low-cost, short-range communication. Maintaining stable connectivity is a big challenge for sparsely deployed and highly dynamic ad hoc wireless network. Reducing contention and maximizing network throughput is also a big challenge for densely deployed ad hoc wireless network, especially when many devices locate in a small area and each device has heavy duty message to transmit. Inspired by the fact that more and more devices support multiple different wireless communication interfaces, we propose two systems to address above challenges by assisting existing wireless ad hoc network by ZigBee interfaces.

An integrated VANET-WSN system was proposed to address the connectivity issue in sparsely deployed VANET. Protocols were designed for efficient vehicle-sensor and sensor-sensor interactions. Prototype of the system has been implemented and tested in the field to verify its feasibility. The simulation results indicate that, with appropriately chosen system parameters, satisfactory safety and energy efficiency can be achieved simultaneously.

In order to reduce contention of pure WiFi network, we have proposed a simple yet effective system for ZigBee-assisted WiFi transmission to improve system throughput. Mobile devices form clusters based on the information gathered by their ZigBee interfaces. Coordinated through ZigBee interfaces, members in each cluster take turns to transmit using their WiFi interface, resulting in reduced contention and collision. Results of experiment and simulation have verified our design by showing that, the throughput, power consumption and fairness can be improved.

7.2 Future Work

There are several future research works we can do about our proposed VANET-WSN system. The main future works of VANET-WSN system are summarized as follows.

1. Even though the node failure within one group has no effect on other groups because of the modularity of our design, it is important for VANET-WSN to detect node failure and replace the failed node with good one timely. Otherwise, it is possible that the dangerous road condition happens within the range covered by the group which contains the failed node will not be seen by incoming drivers.

2. Detecting nodes with low battery and replace the batteries timely is another important and challenge work. Sensors may not work well when their battery is low. The communication range may become short and the sensing accuracy may become low.

3. Since the conditions of highways may be very complicated. From drivers's prospective, false warning message is no better than no warning message. Thus Detecting the real road condition accurately is very important for drivers. In our proposed system, we believe in the sensors. Whatever warning message sent out by the detecting sensors, we believe the message is true. However, it is possible that the sensor may send out some false warning message. It is another challenge to make sure that only the true warning message will be propagated through our proposed network.

4. In this thesis, we only consider one-way road. Even though it is easy to extend our proposed system to two-way roads, new challenges may appear when we deploy the system on two way roads, such as ZigBee interference from both road sides.

There are several future research works we can do about our proposed ZigBee-Assisted WiFi Transmission System too. The main future works of ZigBee-Assisted WiFi Transmission System are summarized as follows.

1. In this thesis, we assume wireless devices are static. If they are mobile, we assume the mobility is relatively low. We need to address more challenges when we extend our proposed system to highly mobile devices and highly dynamic network.
2. As a proof of concept, we implement a prototype of our proposed system. And the result shows that our proposed system can improve the network throughput significantly. However, the implementation is a simplified version of our designed system and the test bed scale is small. We only use 10 laptops to do the experiment, and the system only has one cluster. In order to further demonstrate the effectiveness of our proposed systems, we need to do more complicated implementation and more experiments.

7.3 Acknowledgements

This work was partially supported by the National Science Foundation under Grands CNS-0834593, CNS-0834585, CNS-0831874, and by the ONR under Grant N00014-09-1-0748.

BIBLIOGRAPHY

[1] http://madwifi-project.org.

[2] Cc2420 rf transceiver. In *http://www.chipcon.com*.

[3] Silex wireless modules. In *http://www.silexamerica.com*.

[4] G. S. Ahn, E. Miluzzo, A. T. Campbell, S. G. Hong, and F. Cuomo. Funneling-mac: A localized, sink-oriented mac for boosting fidelity in sensor networks. *SenSys '06*.

[5] G. Ananthanarayanan and I. Stoica. Blue-fi: Enhancing wi-fi performance using bluetooth signals. In *MobiSys*, 2009.

[6] J. Bohli, A. Hessler, O. Ugus, and D. Westhoff. A secure and resilient wsn roadside architecture for intelligent transport systems. *WiSec'08*.

[7] P. Costa, D. Frey, M. Migliavacca, , and L. Mottola. Towards Lightweight Information Dissemination in Inter-Vehicular Networks. *VANET '06*.

[8] D. De Couto, D. Aguayo, J. Bicket, and R. Morris. High-throughput path metric for multi-hop wireless routing. *MOBICOM*, 2003.

[9] Y. Ding, C. Wang, and L. Xiao. A static-node assisted adaptive routing protocol in vehicular networks. *VANET'07*.

[10] N. Wisitpongphan et al. Routing in sparse vehicular ad hoc wireless networks. *IEEE Journal on Selected Areas in Communications*, 25(8):1538–1556, 2007.

[11] State Farm. State Farm Statistics. *http://www.philadelphia-accident-lawyers.com/auto-car-accidents/deer-road-safety.html*, 2005.

[12] F. Farnoud and S. Valaee. Reliable broadcast of safety messages in vehicular ad hoc networks. In *IEEE INFOCOM*, 2009.

[13] G.Indumathi and K.Murugesan. Distributed Fair Scheduling with Distributed Coordination Function in WLAN. *ICON '08.*

[14] J. Hill, R. Szewczyk, A. Woo, S. Hollar, D. E. Culler, and K. S. J. Pister. System architecture directions for networked sensors. in architectural support for programming languages and operating systems. *IEEE Wireless Commun.*, pages 93–04, 2000.

[15] CROSSBOW TECHNOLOGY INC. Wsn. *http://www.xbow.com/.*

[16] T. Jin, G. Noubir, and B. Sheng. Wizi-cloud: Application-transparent dual zigbee-wifi radios for low power internet access. In *InfoCom*, 2011.

[17] S. M. Kim, J. Chong, C. Jung, and et al. Experiments on interference and coexistence between zigbee and wlan devices operating in the 2.4ghz ism band. In *NGPC*, 2005.

[18] T. Kim, S., Y. Lee, and W. Hong. Design and Evaluation of In-vehicle Sensor Network for Web based Control. *ECBS' 06.*

[19] D. Koutsonikolas, T. Salonidis, and et al. Tdm mac protocol design and implementation for wireless mesh networks. In *CoNEXT*, 2008.

[20] Bruce S. Davie Larry L. Peterson. *Computer networks A systems approach.* Morgan Kaufmann, San Francisco, 4th edition, 2007.

[21] J.F. Lee, W. Liao, and M.C. Chen. Inter frame space (IFS) based Distributed fair queuing for proportional fairness in IEEE 802.11 WLAN. *TVT '07.*

[22] U. Lee, B. Zhou, M. Gerla, E. Magistretti, P. Bellavista, and A. Corradi. Mobeyes: smart mobs for urban monitoring with a vehicular sensor network. *Wireless Communications, IEEE*, Oct. 2006.

[23] I. Leontiadis, P. Costa, and C. Mascolo. Persistent Content-based Information Dissemination in Hybrid Vehicular Networks. *PerCom '09.*

[24] I. Liu, F. Takawira, and H. Xu. A hybrid token-cdma mac protocol for wireless ad hoc networks. In *TMC*, 2008.

[25] EasySen LLC. Wieye - sensor board for wireless surveillance and security applications. *http://www.easysen.com/WiEye.htm.*

[26] Packet Data Systems Ltd. www.pds-test.co.uk. *Web Link.*

[27] X. Lu, G. Fan, and R. Hao. A Dynamic Token Passing MAC Protocol for Mobile Ad Hoc Networks. *IWCMC*, 2006.

[28] Q. Ni, T. Li, T. Turletti, and Y. Xiao. Saturation throughput analysis of error-prone 802.11 wireless. In *JWCMC*, 2005.

[29] D. Qiao and K. G. Shin. Achieving efficient channel utilization and weighted fairness for data communications in ieee 802.11 wlan under the dcf. In *IWQoS*, 2002.

[30] A. Rao and I. Stoica. An overlay mac layer for 802.11 networks. In *MobiSys*, 2005.

[31] A. B. Reis, S. Sargento, and O. K. Tonguz. On the performance of sparse vehicular networks with roadside units. *IEEE VTC*, pages 1–5, 2011.

[32] LS Research and LLC. Zigbee proflex module. In *http://www.lsr.com.*

[33] I. Rhee, A. Warrier, M. Aia, and J. Min. Z-MAC: a hybrid mac for wireless sensor networks. *SenSys '05.*

[34] K. Shuaib, M. Boulmalf, F. Sallabi, and A. Lakas. Co-existence of zigbee and wlan: A performance study. In *Proc. IEEE/IFIP Int. Conf. Wireless & Optical Communications Networks*, 2006.

[35] W. Z. Song, R. Huang, B. Shirazi, and R. LaHusen. TreeMAC: Localized TDMA MAC Protocol for Real-time High-data-rate Sensor Networks. *PerCom '09.*

[36] Sok-Ian Sou and Ozan K. Tonguz. Enhancing vanet connectivity through roadside units on highways. *IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY*, 60(8), 2011.

[37] One RF Technology. Short range rf. In *http://www.one-rf.com.*

[38] O. Tonguz, N. Wisitpongphan, F. Bai, P. Mudalige, and V. Sadekar. Broadcasting in VANET. *INFOCOM '08.*

[39] O. K. Tonguz, N. Wisitpongphan, and F. Bai. Dv-cast: A distributed vehicular broadcast protocol for vehicular ad hoc networks. *IEEE Wireless Commun.*, 17(2):47–57, 2010.

[40] J. Wan, D. Yuan, and X. Xu. A review of cluster formation mechanism for clustering routing protocols. In *ICCT*, 2008.

[41] E. Weingatner and F. Kargl. A Prototype Study on Hybrid Sensor-Vehicular Networks. *GI/ITG KuVS Fachgesprach "Wireless Sensor Networks"*, 2007.

[42] Wikimedia. Ieee 802.11. In *http://en.wikipedia.org/wiki/802.11.*

[43] Wikimedia. Tinyos. In *http://en.wikipedia.org/wiki/TinyOS.*

[44] Wikimedia. Zigbee. In *http://www.wikipedia.org/wiki/Zigbee.*

[45] N. Wisitpongphan, F. Bai, P. Mudalige, V. Sadekar, and O. Tonguz. Routing in sparse vehicular ad hoc wireless networks. *JSAC*, 2007.

[46] H. Wu, Y. Peng, K. Long, S. Cheng, and J. Ma. Performance of reliable transport protocol over ieee 802.11 wireless lan: Analysis and enhancement. In *InfoCom*, 2002.

[47] R. Zhou, Y. Xiong, G. Xing, L. Sun, and J. Ma. Zifi: Wireless lan discovery via zigbee interference signatures. In *MobiCom*, 2010.