Efficient Wireless Transmission Supporting Internet of Things

by

Mohammad Ghasemiahmadi
B.Sc., University of Terhan, 2015

A Dissertation Submitted in Partial Fulfillment of the
Requirements for the Degree of

MASTER OF APPLIED SCIENCE

in the Department of Electrical and Computer Engineering

Efficient Wireless Transmission Supporting Internet of Things

by

Mohammad Ghasemiahmadi
B.Sc., University of Terhan, 2015

Supervisory Committee

Dr. Lin Cai, Supervisor
(Department of Electrical and Computer Engineering)

Dr. Dr. Tao Lu, Departmental Member
(Department of Electrical and Computer Engineering)

Dr. Kui Wu, Outside Member
(Department of Computer Science)

Supervisory Committee

Dr. Lin Cai, Supervisor
(Department of Electrical and Computer Engineering)

Dr. Dr. Tao Lu, Departmental Member
(Department of Electrical and Computer Engineering)

Dr. Kui Wu, Outside Member
(Department of Computer Science)

ABSTRACT

The promise of Internet of Things (IoT) and mass connectivity has brought many applications and along with them many new challenges to be solved. Recognizing sensor networks as one of the main applications of IoT, this dissertation focuses on solutions for IoT challenges in both single hop and multi-hop communications. In single hop communications, the new IEEE 802.11ah and its Group Synchronized Distribution Coordination Function (GS-DCF) is studied. GS-DCF categorized nodes in multiple groups to solve the channel contention issue of dense networks. An RSS-Based grouping strategy is proposed for the hidden terminal problem that can arise in infrastructure-based single hop communications. For multi-hop communications, Physical Layer Network Coding (PNC) is studied as a robust solution for multi-hop packet exchange in linear networks. Focusing on practical and implementation issues of PNC systems, different challenges have been addressed and a Software Defined Radio (SDR) PNC system based on USRP devices is proposed and implemented. Finally, extensive simulation and experimental results are presented to verify the contribution of proposed algorithms in comparison with currently used methods.

Contents

List of Tables

List of Figures

ACKNOWLEDGEMENTS

I would like to thank:

Lin Cai, for mentoring, support, encouragement, and patience.

Yue Li, Hamed Mosavat, Haoyuan Zhang, Yuanzhi Ni and Zhe Wei for supporting me in the low moments.

Jianping Pan, Dawood Sajjadi and Maryam Tanha for offering me their full support throughout my program.

DEDICATION

I dedicate this work to my family. My parents, who I wouldn't be where I am if I didn't have their support. My better half, Ghazale, who's always been there for me.

Chapter 1

Introduction

The word *IoT*, which stands for Internet of Things, describes an environment where every *thing* is connected to the Internet. The use of the word *thing* suggests that no device is exempt. The number of opportunities that a world where every device, small or large is connected to a global network brings, is beyond anyone's imagination. As always, along with opportunities come challenges that need to be resolved for such a scenario to function properly. Almost all layers of communication protocol need to be redesigned to handle these challenges, together with an effort to come up with applications that can use this extra-connectivity in its full potential.

The *things* that IoT refers to can be everyday objects, from home appliances to cars and roads. But perhaps the importance of IoT can never be understood without considering sensors. Sensor networks used in monitoring and metering applications are one of the main and first applications of IoT. Sensors that can be attached to anything and can be anywhere with different requirements. For some emergency response related applications, latency is the key factor, whereas for some long term monitoring applications, power consumption is more important. Many IoT applications have a common feature that a large density of nodes is needed.

The deployment and maintenance cost of wired networks makes it almost inevitable to think that this super connected world can be only achieved using wireless communication. On the other hand, in wireless domain, there has been a massive increase in the number of nodes connected to an Access Point (AP) in a Wireless Local Area Network (WLAN) in recent years. Technologies such as IoT, sensor networks and Machine Type Communication (MTC) are anticipated to bring an even larger number of nodes all contending to access the license-free Industrial, Scientific, and Medical (ISM) bandwidth. This is a part of the reasons that the next generation

of IEEE 802.11 standards includes a new sub 1GHz bandwidth to ISM in order to cover the IoT traffic [13]. But the number of connections from both MTC and human devices increases so fast that using conventional IEEE 802.11 MAC protocol's Distributed Coordination Function (DCF) to handle such a large number of nodes results in a high collision probability and very low channel efficiency.

Research efforts have been put to make super dense IEEE 802.11 networks more efficient. In addition to physical layer solutions trying to make wireless communications more efficient and faster, a key issue is how to modify the DCF protocol in the MAC layer to achieve better resource utilization. For instance, [31] proposed a method to adjust the transmission opportunity according to collision probability. One promising solution is grouping users and limiting the competitions within the group, used in Group Synchronized Distributed Coordination Function in the IEEE 802.11ah standard [12]. It has been proved in [30] and [22] that grouping users and dedicating a Restricted Access Window (RAW) to each group increases the overall throughput in wireless networks. The work at [18] has also addressed the problem of super dense networks and proposed a grouping scheme based on transmission attempts to assign larger RAWs to crowder groups instead of having equally timed RAWs.

Although using GS-DCF function proposed in IEEE 802.11ah can improve final throughput compared to conventional IEEE 802.11 protocols, there are still some open issues resulting from the new sub 1GHz license-exempt bands in IEEE 802.11ah. Utilizing carrier frequencies smaller than 1 GHz can increase network coverage up to 1 km. This large distance makes the well-known hidden terminal problem more serious. As shown in [26, 25, 17], the hidden terminal problem can lead to a substantial performance degradation in infrastructure-based networks.

Although RTS/CTS and centralized grouping strategy have been proposed to address the hidden terminal problem [27, 24], they are not suitable for IoT scenarios with small packet size and large number of nodes.

Based on the grouping idea in the IEEE 802.11ah standard, this thesis aims to solve the problem in a distributed manner without using RTS/CTS. Our solution ensures that nodes in the same group are close enough to each other so that they can always sense each other's transmission in their own RAW.

Now, one can grasp an idea of how dominant the IoT can be by just looking at future standards in different wireless communication domains. In cellular communications, Device to Device (D2D) communication is obviously a major part of the next generation of LTE standard known as 5G [8]. There has also been dedicated effort to

support the dense network of nodes that an IoT environment brings and to support the very low latency that some time critical IoT applications like autonomous cars require. Just like cellular communications, wirelessLAN community have also taken big steps toward future IoT networks. IEEE 802.11ah standard has been designed with IoT in mind. It not only introduces new sub 1GHz bands for sensor networks, also addresses the intense channel contention that a dense network brings. For the first time among .11 standards, it can support up to 6000 nodes connected to one access point [16].

As new IoT applications emerge, the need for reliable multi-hop transmission between sensors is becoming more and more obvious. In many sensor network applications due to long distance between sensors and the sink data can not be gathered using a single sink node and it has to be relayed with the help of other nodes. Linear sensor networks are one of the main applications with this criteria. They can be used for monitoring long highways, railroads, skyscrapers and walls. In many of these applications using wire to connect sensors is either impossible or very expensive to deploy and maintain.

Multi-hop relaying has always been a challenge for wireless communications. As the number of hops grows, the end to end throughput suffers drastically. A significant research effort has been employed to improve the throughput and reliability of multi-hop transmissions. Recognizing interference as the main obstacle in multi-hop transmission, many new technologies have emerged. Technologies such as successive interference cancellation [3], multi-packet detection and collision resolution using zigzag [19], interference alignment [6, 7], and full-duplex (FD) radios [14] have all tried to use interference and broadcast nature of wireless channel in their favor.

Another promising technology is Physical Layer Network Coding (PNC). It can present a robust solution for multi-hop communications. Multi-hop communication is essential to IoT applications that have large number of sensors scattered in large geographical areas such as sensors monitoring highways, railroads, rivers and skyscrapers. The nature of these applications does not allow a direct connection to a sink node for every sensor. Thus, packets have to be relayed to the sink node with the help of other nodes.

While the theory of PNC technology has been around for a while, its implementation and real-life applications have been lacked behind. A practical PNC implementation has multiple challenges, most notably Carrier-Frequency Offsets (CFO) and timing synchronization. In the Multiple Access (MA) phase, the relay has to

receive a signal from two sources, tracking the CFO from two transmitters while their signals is superimposed is one of the main difficulties. Existing implementations require an interference free part in each signal which comes with a high overhead and is restrictive for multi-hop scenarios.

1.1 Research Objectives and Contributions

This thesis is trying to solve two issues for efficient wireless networking. For dense wireless access networks, the hidden terminal problem in infrastructure-based IoT scenarios under IEEE 802.11ah is considered. And for multi-hop transmissions, the implementation of a physical layer network coding system for relay transmissions is the focus.

1.1.1 RSS-Based Grouping Strategy

Different from the previous approaches, this thesis proposes an RSS-based grouping scheme. Nodes sense the received power from other nodes and join the group that has the largest sensed power. To the best of our knowledge, this is the first work to explicitly use signal strength for assigning nodes to their groups. The performance gain of the proposed method is then evaluated by analytical model and simulations. The main contributions of this work are three-fold. First, we propose an RSS-based grouping strategy to solve the hidden terminal problem in IEEE 802.11ah networks. Second, we derive the probability of two nodes in one group being able to sense each other. Third, extensive simulations using NS-2[5] have been conducted, and the results show that, by using the proposed method, the hidden terminal problem becomes negligible in IEEE 802.11ah networks and RTS/CTS is no longer needed.

1.1.2 Practical Physical Layer Network Coding Implementation

Despite being proposed a decade ago, physical layer network coding have not yet been fully used in practical applications. One of the reasons behind this delay is implementation challenges that a real-life PNC system has to face. Implementation challenges of PNC are addressed and studied in this dissertation. It proposes a novel idea to overcome the CFO challenge and have the relay correctly receive the sum signal

in MA phase. This thesis presents a design and implementation of a Software Defined Radio (SDR) PNC communication system on the GNURadio platform. The design is then implemented on a USRP testbed. Extensive simulations and experiments have been done to investigate the proposed implementation and its feasibility.

1.2 Dissertation Organization

The rest of this thesis is organized as following:

Chapter 2 gives background information about the IEEE 802.11ah standard, hidden terminal problem, physical layer network coding and software defined radio.

Chapter 3 explains the proposed RSS-based grouping strategy for GS-DCF protocol, then presents its analytical model and performance evaluations.

Chapter 4 explains the practical implementation of physical layer network coding technology on software defined radio platform and how the CFO and timing recovery challenges are solved. Then experimental and simulation results are presented.

Chapter 5 is the conclusion. It also presents areas for further research and improvement of this work.

Appendix A gives some advice for troubleshooting GNURadio and some tips on developing new blocks.

1.3 Bibliographic Notes

The work in chapter 3 was published in [10].

Chapter 2

Background

2.1 IEEE 802.11ah Standrad

The IEEE 802.11ah standard is one of the recent standards by IEEE standard organization. This standard introduces many new concepts for the first time, including new designs for both the physical and MAC layers. The 802.11ah standard has been designed with three use cases in mind, smart sensors, back-haul aggregation and extended range hot-spot. In smart sensor use case, 802.11ah is designed to handle a very large number of nodes (up to 6000) connected to one access point. In the back-haul use case, an 802.11ah access point is designed to be used as an aggregator for wireless personal area network (WPAN) devices that use the IEEE 802.15.4.g standard. WPAN (802.15.4g) devices have a small transmission range and lower data rate, so in this case, 802.15.4g routers gather data from 802.15.4g devices and send them to a single 802.11ah aggregator. In the extended hot-spot use case, 802.11ah access points can be used as candidates for cellular offloading, specially in outdoor environment where other 802.11 standards suffer from their short transmission ranges.

2.1.1 Physical Layer of IEEE 802.11ah

In the Physical Layer (PHY), 802.11ah makes use of sub 1GHz license-exempt bands. This is the main physical layer difference between this standard and the previous 802.11 standards. Also, 802.11ah mostly focuses on small bandwidths and does not allow any channel with >20 MHz bands. The use of lower carrier frequency gives 802.11ah a much larger transmission range compared to other 802.11 standards as the path loss increases with regard to carrier frequency. In outdoor scenarios, using

default transmission power (200mW), 1MHz channel and Modulation and Coding Scheme (MCS) 10, an 802.11ah device can achieve a range as high as 1 km [16].

2.1.2 Medium Access Control layer of IEEE 802.11ah

In the Medium Access Control (MAC) layer, 802.11ah introduces may new concepts from shortened frame formats to channel access and power management. In the main 802.11 standard, the number of nodes that can be connected to one access point is limited to 2007. IEEE 802.11ah extends this range to 8191 nodes to support the smart sensor use case. Also to better support dense network scenarios, 802.11ah shortens the length of frame headers, control frames such as ACKs and periodically transmitted frames such as beacon messages. The main change in 802.11ah which is also the focus of this thesis are the modifications in the channel access.

In order to handle the severe contentions between nodes in dense scenarios, 802.11ah divides all nodes connected the access point into different groups, and each group is given a specified time window called Restricted Access Window (RAW) to use the wireless channel while members of all other groups remain silent. This essentially divides the beacon period into multiple RAWs each dedicated to one group. It not only decreases the contention that one node experiences while using the wireless channel, but also lets nodes save power by going to sleep in the RAWs that do not belong to them. It is left to the access point to decide whether or not to allow nodes to cross the RAW boundary just to finish an ongoing transmission (Cross Slot Boundary). This channel access mechanism is called Group Synchronized Distributed Coordination Function (GS-DCF) which is to be used in 802.11ah instead of the traditional DCF.

2.2 The Hidden Terminal Problem

The hidden terminal problem is a well-known issue in wireless communication networks. It usually happens when a node falls outside of the sensing area of another node but they are both trying to transmit simultaneously. Fig. 2.1 illustrates a general case for the hidden terminal problem. For example, if node A, first starts the transmission of a packet to node B and in the middle of their communication, node C also decides to transmit a packet to B or another node. According to the Carrier-Sense Multiple Access with Collision Avoidance (CSMA/CA) procedure, used

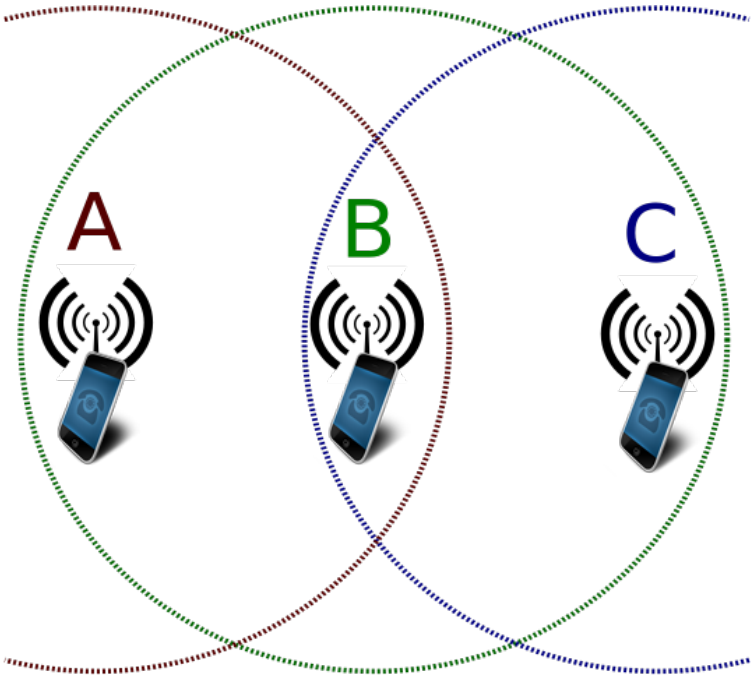


Figure 2.1: A simple illustration of hidden terminal problem

in many wireless MAC protocols, it will first listen to the channel, and since it cannot sense the signal coming from A, it will find the channel idle and initiate its transmission which will result in a collision at B and loss of both packets from A and C. The uplink transmissions in Wireless LANs at the infrastructure mode, where several wireless nodes want to transmit to the access point, are vulnerable to this problem.

The legacy solution for this problem is the use of Request To Send (RTS) and Clear To Send (CTS) messages. Using these messages, every node has to send an RTS packet to its receiver and wait until it gets a CTS packet before starting a transmission. Within the RTS and CTS packets, there is a Network Allocation Vector (NAV) which announce the anticipated channel occupation duration so any nodes receiving RTS or CTS should refrain from transmission during the time reserved according to NAV. In this way, although node C might not sense the RTS from node A, it will receive the CTS that B sends to A and know that the channel for B is going to be busy. The only chance of collision because of the hidden terminal then, is during the transmission of the RTS message. Since the RTS is a small control message, collision probability will be reduced significantly.

Given the extended range of communication in the 802.11ah standard, the hidden terminal problem becomes even more serious, and the RTS/CTS solution may not

be effective in dense networks when the number of nodes trying to transmit RTS simultaneously becomes large.

2.3 Physical Layer Network Coding

The concept of Network Coding is nothing new in the wireless domain. It relies on optimizing transmission of data by combining two or more messages in a network. Physical Layer Network Coding(PNC) is then a network coding happening at the physical layer.

We use a three node linear network example to illustrate this idea [29]. In Fig. 2.2, nodes A and B have packets to exchange and node R can act as a relay.

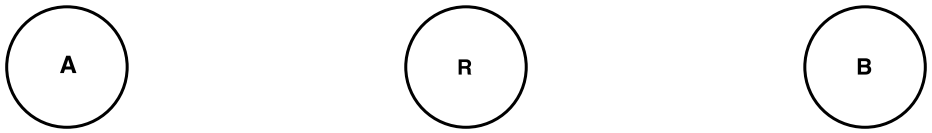


Figure 2.2: Simple three node linear network. Nodes A and B have packets to exchange and node R acts as relay

Assuming that A and B are too far for a reliable transmission, there are different ways for this exchange of packets to happen, traditional relaying, straightforward network coding and physical layer network coding.

An exchange of packets between A and B where relay only works as a traditional decode and forward node takes four time slots as illustrated in Fig. 2.3.

Using straightforward network coding, reduces the number of time slots required for that aforementioned exchanged to happen to three. Fig. 2.4 shows this scenario. This way, using the broadcast nature of the wireless channel, the relay having stored both A and B's message, broadcasts a combined version of X_1 and Y_1. Then both A and B can decode the other node's packet, having their own message and the message they received from the relay. The simplest coding scheme to be used by relay to generate the combined message is calculating XOR of X_1 and Y_1.

Physical layer network coding is similar to straightforward networking coding with the difference that in PNC, the two messages from A and B are coded (combined) in the physical layer using the additive nature of simultaneously arriving electromagnetic (EM) waves [29]. This reduces the number of time slots for the same exchange to happen to two. In a simple BPSK modulation where 0 and 1 are represented with -1

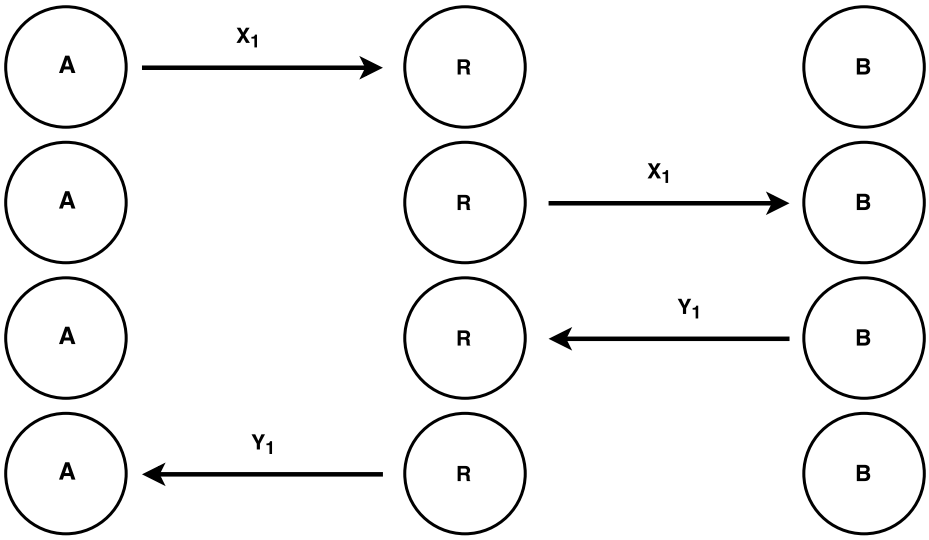


Figure 2.3: Traditional relaying. First, node A transmits its packet to the relay, then the relay forwards A's packet to B, then in time slot three, node B sends its packet to the relay, which is then forwarded to node A in time slot four.

and $+1$ respectively, if A and B transmit their packet to the relay at the same time, they relay will received symbols of -2, 0, and $+2$. At this step, every PNC capable relay requires a mapping function, to convert these new symbols into bits of data with a meaningful relationship to A and B's bits. In this simple BPSK example, mapping -2 and $+2$ symbols to 1 and 0 symbols to 0, means that the received message at the relay is again XOR of A and B's packets. Just like the straightforward network coding, the relay then broadcasts this information back to A and B and they will use it to extract the other node's packet.

The same technique can be extended to linear multi-hop networks with more than two hops. Although a multi-hop scenario suffers from interference from other nodes, PNC is among technologies that can still maintain a reliable error rate and much higher throughput.

2.4 Software Defined Radio

2.4.1 Overview

Software Defined Radio (SDR) is a general term for describing a communication system where the main components are implemented in software instead of a hardware. Depending on the application, the software can be running on either a host computer

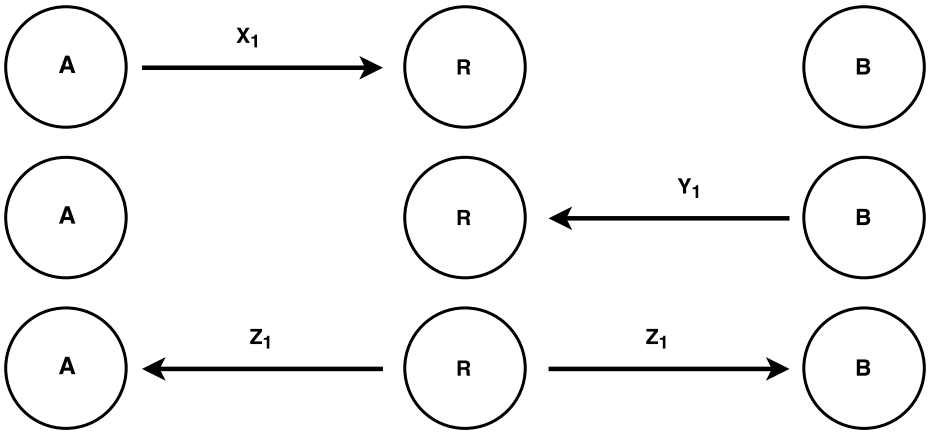


Figure 2.4: Straightforward network coding. First, A transmits its packet to relay, then B transmits its message to the relay, then relay calculates $Z_1 = X_1 \oplus Y_1$ and broadcasts it to A and B in time slot three, which can then decode the other nodes message by another XOR.

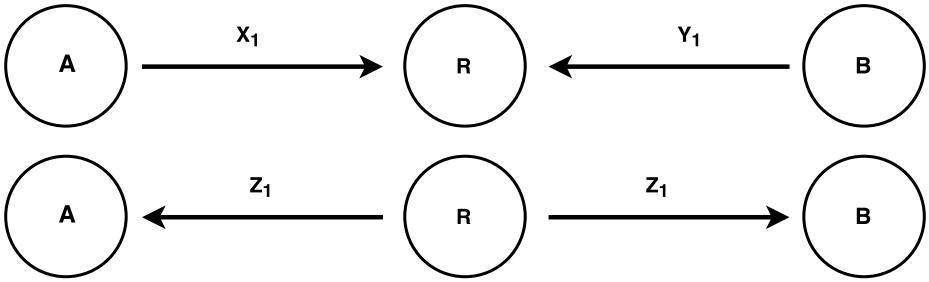


Figure 2.5: Physical layer network coding. In time slot one, A and B will transmit their packets to the relay at the same time (multiple access phase). The relay, using its mapping function extracts XOR of A and B's packets and broadcasts it to them in time slot two (broadcast phase).

or an embedded device. Developing radios in software has many advantages over hardware design. Software radio is a lot more rapid in development, it is easier to debug and it is customizable at a very low cost. Even in embedded system based SDR designs, the implementation can change with a new firmware, whereas a hardware system might need totally new device even for smallest modifications.

Fig. 2.6 shows main components of an SDR system. Analog to Digital Converter (ADC) and Digital to Analog Converter (DAC) are at the heart of every software defined radio. Since the carrier frequency for many protocols is much higher than hundreds of mega hertz, the signal received by the receiver chain antenna has a very high bandwidth which can not be directly converted to digital. Thus, a mixer would

have to bring the received signal to baseband domain before being sampled by ADC. Similarly in the transmitter chain, the signal after DAC is in baseband and is brought to the required carrier frequency by an analog mixer.

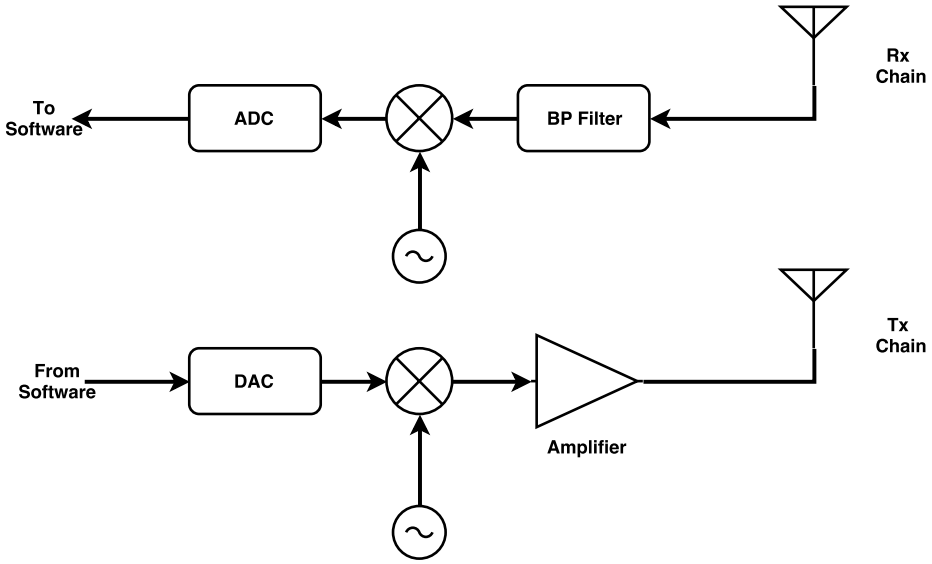


Figure 2.6: Simple illustration of software defined radio concept

2.4.2 Universal Software Radio Peripheral

Universal Software Radio Peripheral or USRP, is a famous SDR hardware by Ettus Research. USRPs have a motherboard with very accurate and high speed ADC and DAC, together with a Field-Programmable Gate Array (FPGA) for signal processing purposes. One of the key advantages of USRP is that it does the analog operations explained in the previous section on a different board than the motherboard. Daughterboards, as they are called by USRP community, have oscillators and mixers and analog amplifiers on them. Each daughterboard is designed for a specific range of frequency spectrum. This way, one can experiment on almost any frequency using the same motherboard by just using the appropriate daughterboard.

In normal applications, the FPGA on USRP adjusts the sampling rate of incoming and outgoing sample streams. Digital samples are transmitted to and received from a host computer using a LAN cable or USB. The FPGA is open to be programmed for high speed and low latency operations that can not tolerate the delay of communicating with a host computer.

USRP devices come with USRP Hardware Driver (UHD) to be installed on a normal computer. Using UHD, one can work with USRPs on MATLAB, Simulink, GNURadio or by writing softwares directly communicating with the UHD Application Programming Interface (API).

2.4.3 GNURadio

GNURadio is a free software by GNU project distributed under the terms of the GNU General Public License (GPL). It provides a software platform for signal processing and SDR systems. It can be used to either run pure simulations or interact with external hardware like USRP for SDR purposes. The user can implement arbitrary protocols by connecting different block and creating a flow graph for digital samples to be processed. GNURadio comes with many built-in blocks for modulation, coding, and many other signal processing functions. The user can also implement and add its own blocks to the platform. New blocks can be written in either C++ or Python language and blocks written in different languages can work together in the same system. This way, the user can have the advantages of Python and C++ together. Python can be use if rapid development is necessary and C++ language is used where optimization is concerned.

Chapter 3

Grouping For Dense Wireless Access Networks

3.1 Overview

The IEEE 802.11ah standard proposed for dense sensor networks, uses a sub 1GHz band that can result in transmission ranges as high as 1 km. In a network with such a large radius and density, the traditional hidden terminal problem becomes more serious and conventional solutions do not seem to address the problem in its full extent. Recognizing this challenge, an RSS-based grouping strategy is proposed to categorize nodes in a way that the hidden terminal problem can no longer be an issue. The performance gain of the proposed method is then evaluated by an analytical model and simulations.

3.2 Related Works

[25] has considered the hidden terminal with a new approach in a Constant Contention Window (CCW) scenario and studied the problem of two competing senders that cannot sense each other but are trying to transmit to a station that both can sense, a frequent situation in infrastructure mode networks.

One solution for the hidden terminal problem is using RTS/CTS mechanism. This mechanism solves the problem for the case that the transmission time of a data packet is much larger than that of RTS/CTS. However, it introduces additional overhead, and the collisions cannot be avoided due to the RTS packets from hidden terminals.

In [27], the hidden terminal problem in GS-DCF based networks was discussed. The proposed hidden matrix based regrouping (HMR) algorithm can reduce the number of hidden terminals but it is a centralized approach and requires the access point to discover hidden terminal relationship between two nodes first and then solve it by regrouping one of them to another group.

The research in [24] has also studied the hidden terminal problem but in IEEE 802.15.4-based networks. A grouping mechanism similar to IEEE 802.11ah is proposed but the strategy to deal with the hidden terminal problem is still based on discovering hidden terminal relationships and rescheduling nodes.

[2] proposed a Group-based MAC where a new station join a group if the estimated distance to the group leader is below half of the transmission distance to avoid hidden-terminal, and otherwise a new group will be formed. The AP cannot effectively control the number of groups.

When the AP can obtain the global information about the location of all nodes, existing clustering algorithms such as k-means can be applied and the group designation is completely controlled by the AP in a way similar to centralized grouping schemes described in [30].

However, when the number of terminals is large, collecting global information significantly increases the overhead of feedback. Assigning all of the nodes to their groups results in a high control overhead especially when the network is dynamic.

3.3 Rss-Based Grouping Strategy

3.3.1 System Model

In this work, we consider an infrastructure based IEEE 802.11ah network, where each wireless terminal transmits its packets directly to the access point. The network coverage area is a circle with radius R where the access point is at the center. Nodes are uniformly distributed in the circle and their locations follow a Poisson Point Process (PPP) with density λ. The latter two assumptions are used for analysis and they are not necessary for the algorithm to perform. Nodes are assumed to be static at their position which is a reasonable assumption for most IoT and sensor networks and an active node always has a packet to transmit.

The channel model used in the analysis is a pathloss model with parameters α

and β in a way that the received power at a distance d from the transmitter node is

$$P_r(d) = \frac{\beta}{d^\alpha} P_t, \tag{3.1}$$

where P_t is transmit power. In the simulation, we also consider the Rayleigh fast-fading and log-normal shadowing model where the received power is an exponential random variable and its mean $\bar{\gamma}$ is a log-normal random variable as

$$\bar{\gamma}_{dB} = \mathcal{N}(10 \log_{10}(P_r(d)), \sigma). \tag{3.2}$$

We refer to both centralized and distributed schemes used in [30] as random grouping schemes. We also do not distinguish between RAW slot crossing and not crossing cases known as CR-GS-DCF and NCR-GS-DCF respectively which has negligible impact on the grouping decision [12].

3.3.2 RSS-Based Grouping

In order to avoid disadvantages of a centralized solution mentioned before, one can utilize the idea of measuring the sensed power from other nodes to let users choose their own groups based on which group their neighbors are in. Here we propose a grouping scheme based on sensed power from other nodes. The grouping mechanism should be triggered by the access point (AP) using its Beacon Frame (BF) at each Grouping Update Period (GUP) and nodes should listen to the channel for the sensed power of pilot messages from group heads. Such a BF at the beginning of a GUP is called a Beacon* and its difference with conventional beacon frames is that there is a grouping procedure at its beginning.

We propose an RSS-based strategy that a node will listen to some pilot messages from randomly chosen nodes in a reserved time slot called Grouping Slot Time (GST) by an order specified by the AP.

```
for Each GUP do
    AP chooses M random nodes (group heads);
    AP informs group heads about their index in BF;
    AP includes M in BF;
    for Each node do
        if It is a group head then
            Transmits a pilot in its own GST;
        else
            Measure power of each GST;
        end
    end
    At the end of last GST for Each node do
        if It is a group head then
            Joins the group with its GST index;
        else
            Joins the Group with the largest power GST index;
        end
    end
end
```

Algorithm 1: RSS-based grouping

A time diagram of the algorithm is illustrated in Fig. 3.1 and an example of Voronoi cells is shown in Fig. 3.2.

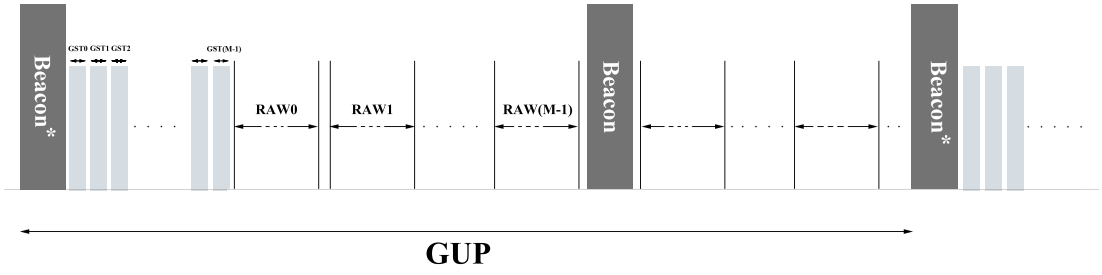


Figure 3.1: Time diagram of RSS-based grouping algorithm, Beacon* is a BF initiating the grouping scheme. As it is shown in the figure, a GUP can be as large as several beacon intervals. Essentially it can last as long as the AP is satisfied with the throughput performance.

Groups are formed in Voronoi cells around group heads. Nodes will then remain in their selected RAW until another trigger from the AP. Comparing to centralized

schemes studied in [30] which want the AP to assign each node to a group, this scheme has a much lower control overhead and it does not require any location information to be collected and exchanged in the network.

In the rare case that a node is very far away from all the group heads, which means that it can not have any measurement for any GST, it will randomly choose one of the groups or it can report to the AP to be assigned to a group.

3.3.3 Implementation Issues

The proposed scheme can be readily implemented in current IEEE 802.11 systems without any major modifications. The power measurement already exists in the MAC layer of all systems and they can provide RSSI at any time, so nodes can then find the sensed power of each GST transmission by measuring the average RSSI during the reception time of the pilot.

GST can be as small as an ACK time and there should be some guard distance time between pilot transmissions from group heads. For the best performance, the optimum value of GUP and GST should be determined which is beyond the scope of this thesis and they do not affect our following analysis. The grouping can also be triggered by the AP if the randomly selected headers result in unsatisfactory network performance.

Including M in BF is something that is already being done in the IEEE standard draft [12]. Informing group heads about their index can also be done using a mapping strategy similar to the one used for delivery traffic indication known as DTIM.

Our algorithm is also robust against interference from other APs. Since the GSTs are short and they could be transmitted consecutively with a gap of SIFS, any other CSMA/CA based station, e.g. other APs, cannot transmit anything during the grouping procedure.

3.3.4 Analytical Model

In order to compare the performance of grouping schemes on hidden-terminal probability, here we define two metrics and use them along with the probability of having a hidden terminal in a group to show the benefits and costs of each grouping scheme. The first metric is the average distance between nodes in a group, D_m, which is

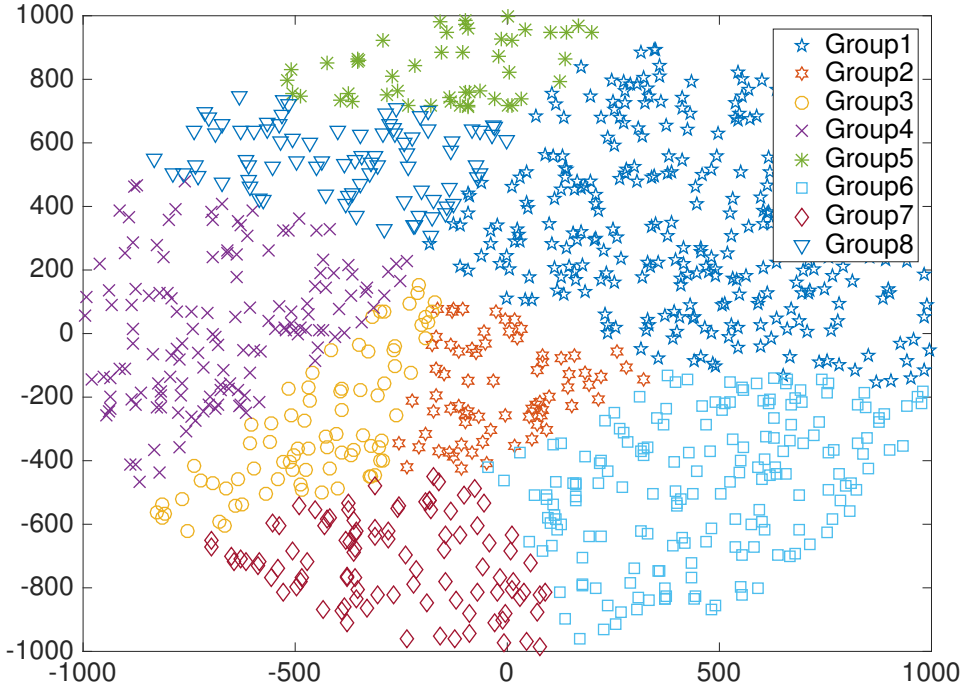


Figure 3.2: A sample grouping showing how nodes are categorized location-wise using RSS-based grouping

defined as

$$D_m = \frac{\sum_i \sum_{j \neq i} d_{ij}}{\binom{n_m}{2}}, \qquad \forall i \text{ and } j \in \text{group } m, \tag{3.3}$$

where n_m is the number of nodes in group m and d_{ij} is the distance between nodes i and j.

The second metric is the standard deviation of the number of nodes in different groups which specifies how evenly nodes are distributed between different groups.

$$\sigma = \sqrt{\frac{1}{M} \sum_{m=0}^{M-1} \left(n_m - \frac{N}{M} \right)^2}, \tag{3.4}$$

where N is the total number of nodes and M is the number of available groups.

The next step is finding the probability of encountering the hidden terminal problem for different schemes. The number of group head nodes in our proposed strategy is fixed so their locations follow Binomial Point Process (BPP) and there are M of

them in the entire coverage area. Thus, the probability that there are k group heads in an area of size B can be found as

$$Pr[N(B) = k|N(A) = M] = \binom{M}{k} \left(\frac{B}{A}\right)^k \left(1 - \frac{B}{A}\right)^{M-k}, \tag{3.5}$$

where $A = \pi R^2$ is total area of network. We use D to denote the distance between one node and its nearest group head. Every node will find its nearest group head, therefore for the CDF of D we have

$$F_D(x) = 1 - Pr[N(S(x)) = 0|N(A) = M] \quad x \geq 0, \tag{3.6}$$

where $S(x)$ is the overlapping area between a circle centered at the observed node with radius x and the coverage area. In general $S(x) \leq \pi x^2$ and the inequality happens for nodes close to the coverage area's border. We define $\hat{F_D}(x)$ by

$$\hat{F_D}(x) = 1 - Pr[N(\pi x^2) = 0|N(A) = M] \quad 0 \leq x \leq R. \tag{3.7}$$

It is always true that $\hat{F_D}(x) \geq F_D(x)$ for $0 \leq x \leq R$. However, when M is large enough, which is typically the case in practice [30] the group size is so small comparing with the whole network area that the difference between $F_D(x)$ and $\hat{F_D}(x)$ becomes negligible. We use the latter as an upper-bound of the real case.

$$\begin{aligned} F_D(x) \approx \hat{F_D}(x) &= 1 - Pr[N(\pi x^2) = 0|N(A) = M] \\ &= 1 - \binom{M}{0} \left(\frac{\pi x^2}{A}\right)^0 \left(1 - \frac{\pi x^2}{A}\right)^M \\ &= 1 - \left(1 - \frac{\pi x^2}{A}\right)^M \quad 0 \leq x \leq R. \end{aligned} \tag{3.8}$$

The validity of this approximation is then verified by simulation in Section 3.4.

We can find PDF of D by taking derivative of its CDF:

$$f_D(x) = \frac{2\pi M x}{A}(1 - \frac{\pi x^2}{A})^{M-1} \quad 0 \leq x \leq R. \tag{3.9}$$

Using this PDF we can find the probability of having a hidden terminal when our RSS-based grouping scheme is used.

Figure 3.3 shows a group head and a random node called node 1 in the same group

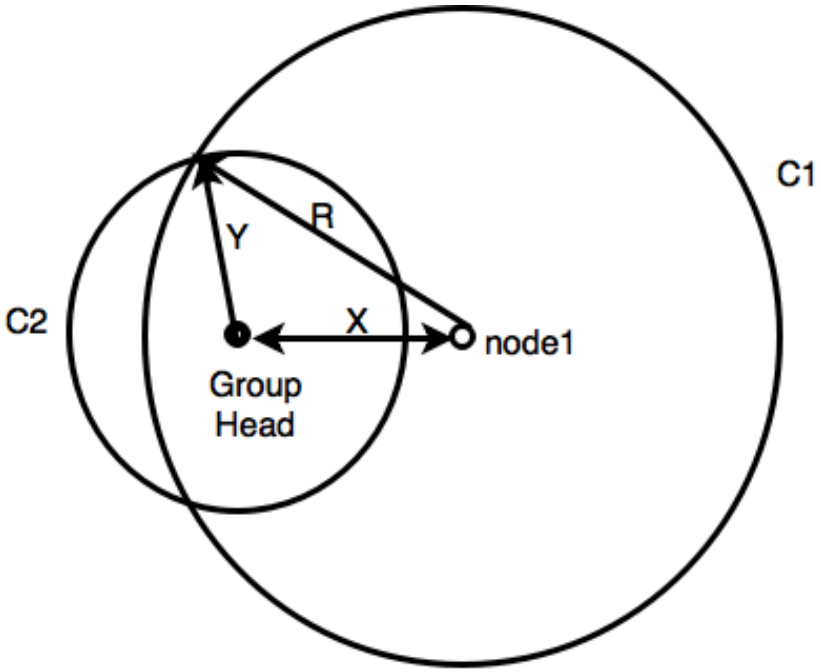


Figure 3.3: Two nodes and group head in one Voronoi cell

in distance X where X follows the PDF in (3.9). The circle C1 shows the sensing range of node 1 and C2 is a circle centered by the group head with the raduis Y which also follows the PDF in (3.9). The probability for node 2 which is Y away from the group head to be in the sensing range of node 1 is the ratio of the portion of C2 inside C1 to the perimeter of C2. Then, we can take the average over the PDF of X and Y to obtain the probability that two nodes are within each other's sensing range.

$$Pr_{\text{sense}} = \int_0^{\sqrt{\frac{A}{\pi}}} \int_0^{\sqrt{\frac{A}{\pi}}} \frac{g_1(x,y)}{2\pi y} f_X(x) f_Y(y) dx dy, \tag{3.10}$$

where Pr_{sense} is the probability that two nodes in the same group are in each other's sensing range, R_s is the sensing range of a node,

$$g_1(x,y) = \begin{cases} 2\pi y & R_s \geq x+y \\ 2y \arccos(\frac{x^2+y^2-R_s^2}{2xy}) & R_s < x+y \\ 0 & R_s < |x-y| \end{cases}, \tag{3.11}$$

and $f_X(x)$ and $f_Y(y)$ are the PDF of X and Y respectively.

We can also derive the same probability for random grouping schemes. The probability density function for the distance between two random nodes in a circle with the radius R is given in [21]

$$f(r) = \frac{2r}{R^2}\left(\frac{2}{\pi}\arccos(\frac{r}{2R}) - \frac{r}{\pi R}\sqrt{1-\frac{r^2}{4R^2}}\right), \quad 0 < r < 2R. \tag{3.12}$$

In this case, for random grouping, the probability of that two nodes in the same group are in each other's sensing region can be obtained as follows,

$$Pr_{\text{sense}} = \int_0^{2R} g_2(r)f(r)dr, \tag{3.13}$$

where

$$g_2(r) = \begin{cases} 1 & r \leq R_s \\ 0 & r > R_s \end{cases}. \tag{3.14}$$

These probabilities can then be calculated by numerical methods.

3.4 Performance Evaluation

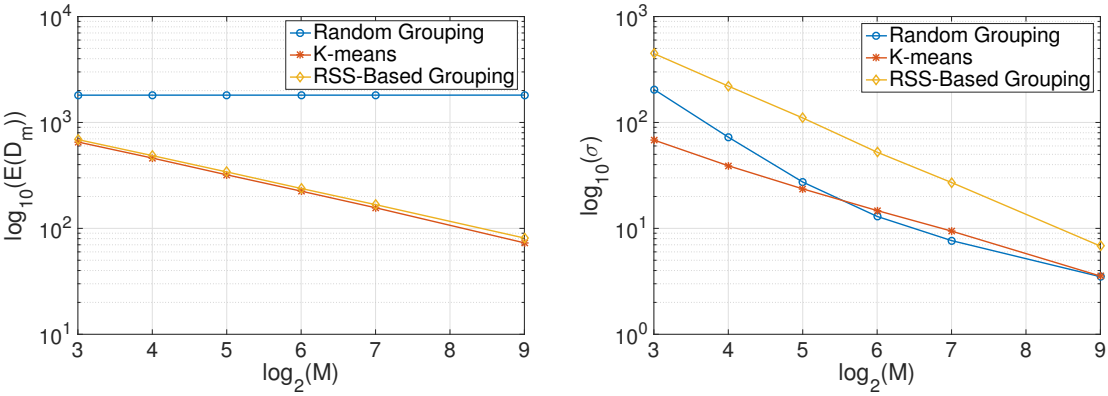
The performance of different grouping schemes is evaluated using MATLAB and NS-2[5]. MATLAB is used for numerical calculations and NS-2 is used for throughput simulations. The network coverage is assumed to be a circle with radius 1 km which is the case in sub 1GHz network having the access point at the center. As mentioned in Section 3.3.1, nodes are uniformly placed in the coverage area with their number following a Poisson distribution with density 0.0019 node per square meter which results in 6000 nodes in the network on average. The number of groups, M, was set in the range of 8 to 512. For each setting, the experiments were repeated 20 times, and the average was taken.

If the AP the knows location information of all the nodes, in a pure centralized solution where AP decides the group for each node, the k-means algorithm can be used to achieve near-optimal groupings. So we can use the performance with k-means grouping as a benchmark to see how well our RSS-based distributed grouping

algorithm is working compared with the centralized benchmark.

In Fig. 3.4a, the average distance between nodes in a group for the proposed scheme is always smaller than with the random grouping scheme and close to the k-means clustering algorithm. It is also obvious that a larger number of groups have a smaller average distance between nodes for all algorithms.

In Fig. 3.4b, we can see that the standard deviation of the number of nodes in different groups is the cost that the RSS-based grouping algorithm pays for the advantage of distributed solution. Groups with a large number of users will suffer from high collision probability and groups with a very small number of users may not fully utilize their RAW. It is shown later using NS-2 simulations, that despite having some dense groups, the RSS-based grouping algorithm has a better performance than random groupings. This is because the hidden node problem is a more severe problem than uneven groups. A few hidden terminals can dramatically decrease the performance of the whole network. Having more nodes that are all in the sensing range increases the competition but has a less destructive effect on the throughput than that due to hidden terminal.



(a) Average D_m over all groups

(b) Standard deviation of the number of nodes in different groups

Figure 3.4: Grouping performance metrics

Assuming that all nodes have the same sensing range, R_s, the probability of two random nodes in the same group be in their sensing range versus R_s for a constant number of groups is shown in Fig. 3.5a. The figure also shows that the approximation in (3.8) not affect the result very much. As we can see in the figure, best performance is for the clustering using the k-means algorithm but the RSS-based grouping scheme also performs very close to the k-means and much better than the random grouping

schemes. The same probability assuming Rayleigh fading channel between nodes is also shown in Fig. 3.5a. The performance with Rayleigh fading channels is degraded because considering fast fading the areas occupied by groups are no longer Voronoi and the group head with the highest sensed power is not necessarily the closest one. As shown in the figure the algorithm still performs well which indicates that, the RSS estimation accuracy required is low, which results in a small overhead.

With the simulation results following analytical results, our assumption in section 3.3.1 is verified. The simulation and analytical results are closer to each other as M grows. That is because, with a higher value of M, the edge effect becomes more negligible.

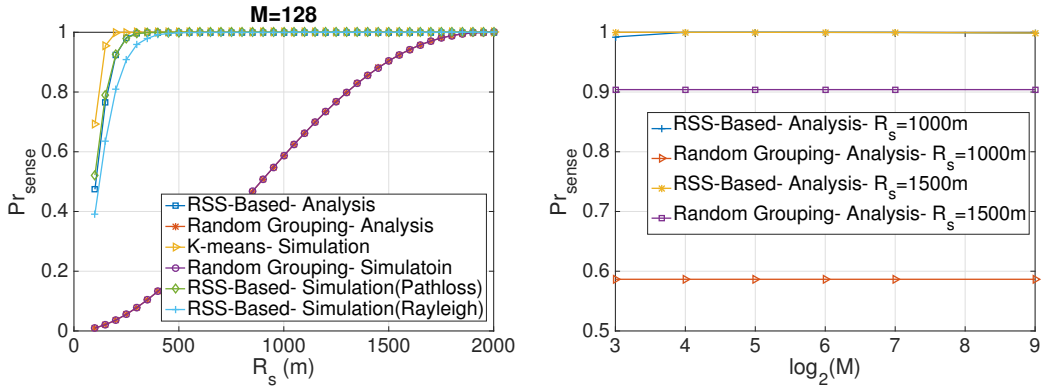


Figure 3.5: Probability of two nodes, within a group, being in the sensing range of each other

Fig. 3.5b shows the probability of two nodes in the same group being in each other's sensing range when all nodes' sensing range is fixed and the number of groups changes. This probability does not change with the number of groups for random grouping because nodes in the same group can still be anywhere in the coverage area and their distance still follows the same PDF. Though in case of the RSS-based grouping algorithm, a larger number of groups results in a higher probability for group members to be in the sensing range of each other because it makes the Voronoi cells smaller.

Fig. 3.6 shows the throughput simulation result for different grouping schemes along with k-means clustering algorithm using NS-2. Table 3.1 shows the parameter setting for simulations used in this section. Other common settings such as PHY and MAC header are set according to the IEEE 802.11 standard [1]. The payload size

is set to be low considering the IoT applications such as smart meter measurement reporting and all nodes are using RTS/CTS mechanism. Only the uplink traffic is considered and a non-collided transmission is assumed to be successful which that means the PHY layer BER is not considered.

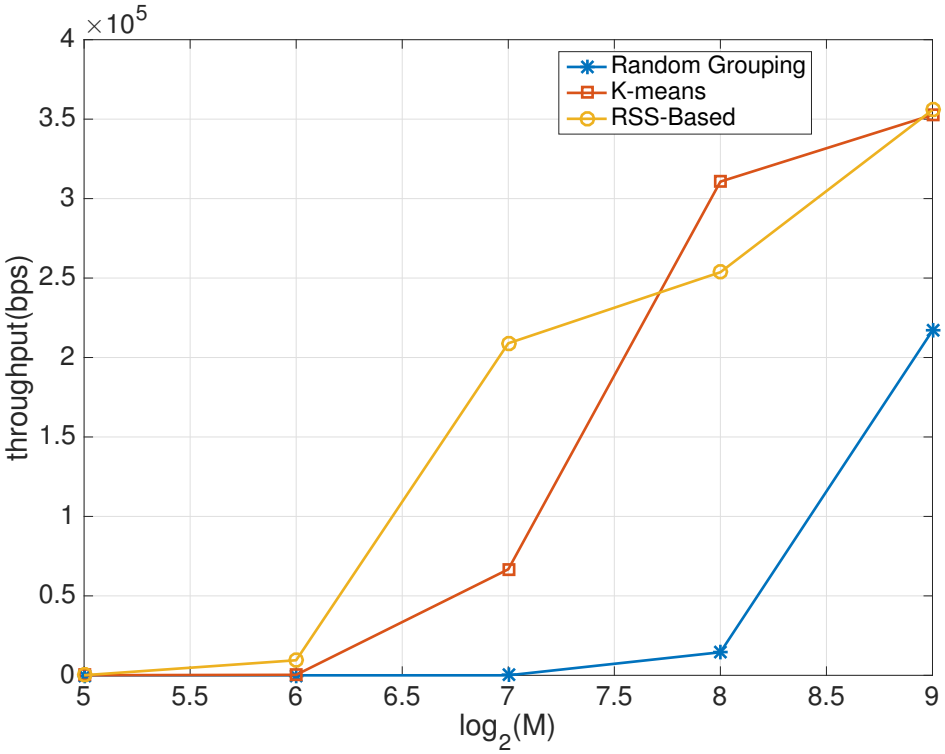


Figure 3.6: Throughput of Uplink traffic

In Fig. 3.6, the throughput for random scheme is very low because the time for a single packet transmission is more than the largest back off window (1024 slots) and almost any transmission would be interrupted when there is a few active nodes who are hidden terminals. In other words, a node with a hidden terminal may hardly success in transmitting a packet to the AP without collision.

This figure also shows the limited cost that RSS-based grouping pays due to the higher standard deviation of number of nodes in each group. The throughput using the proposed RSS-based grouping is close to that using the centralized K-mean solution. On the other hand, random grouping has much worse performance due to the hidden terminal problem even when the sensing range is as high as 1500m. Note that the theoretical sensing range without considering the shadowing effect may be much higher than the practical sensing range, so grouping considering the location is

Table 3.1: Throughput Simulation Parameter Setting

Parameter	Value
Slot time	20 μS
SIFS	10 μS
Payload Size	64 bytes
Data Rate	1Mbps
R_s	1500
CW min	31
CW max	1023

important to avoid hidden terminal and improve network performance.

Chapter 4

Multi-Hop Transmission

4.1 Overview

One of the technologies that is promising for multi-hop sensor network applications is physical layer network coding. It can increase the end to end throughput by decreasing the number of time slots that are required for a multi-hop transmission while maintaining an acceptable error rate.

Providing a solution for implementation challenges of PNC, feasibility of a real-life PNC system is proven by a software defined radio implementation on GNURadio platform. Different performance parameters are then measured in both simulation and experiment.

4.2 Implementation

We implemented physical layer network coding on USRP and GNURadio to build the multi-hop wireless testbed. For MPNC, the transmitter and receiver in the single-hop transmission can be made using regular transmission and reception chains. In this work, standard BPSK blocks of GNURadio are used for all steps of communication, modulation, timing recovery, phase tracking and demodulation.

For MA transmissions, signals from two transmitters are synchronized in the symbol level. The synchronization issue has been heavily investigated, e.g., SourceSync [23] can achieve low-overhead distributed synchronization with the accuracy of tens of ns, smaller than the symbol duration (e.g., 802.11 OFDM symbol time of 4 μs). As synchronization for a static TDMA network is relatively easy and it in fact makes

the estimation and compensation of CFOs and detection of preamble during the MA transmission more difficult, we use a MIMO cable to synchronize the transmitters in our USRP testbed, and focus on the most challenging part, the receiver of MA transmission with symbol level synchronization.

4.2.1 Signal formulation

Since the oscillator at the mixer of practical devices is never perfect, the real carrier frequency of each device may be slightly different from the nominal value. This causes a small rotating phase to remain in the signal after the mixer. In regular point-to-point communications, this frequency offset can be estimated and compensated using phase tracking algorithms. The same algorithms are no longer useful in MA transmissions since the signals are from two sources and there are two frequency offsets, so the resulting phase cannot be removed by a simple multiplication. Several algorithms have been proposed [20, 15, 11, 9] to compensate the effect of CFO when multiple signals are mixed at the receiver. These algorithms either require a small portion of interference-free part in two signals, which are not directly applicable for MPNC with symbol level synchronization. Another approach is to use the mean of two frequency offsets to partially compensate the frequency offsets for PNC MA transmission [20]. In multi-hop scenarios, error propagation is a severer problem than that in the two-hop PNC case, so a careful design that can accurately compensate the two CFOs is needed.

Assuming that nodes A and B are transmitting symbol a_k and b_k at the k_{th} symbol interval, respectively, the baseband signal can be written as

$$x(t) = \sum_{k=0}^{k=N} a_k g(t - kT), \qquad \text{and} \tag{4.1}$$

$$y(t) = \sum_{k=0}^{k=N} b_k g(t - kT), \tag{4.2}$$

respectively, where $g(t)$ is the pulse for the pulse shaping step (a root raised cosine (RRC) pulse is used here), N is the number of symbols in one packet, and T is the symbol interval. Considering the received signal as

$$r(t) = x(t - \tau)H_a e^{j2\pi(f_a)t} + y(t - \tau)H_b e^{j2\pi(f_b)t}, \tag{4.3}$$

where f_a (f_b) is the CFOs between the oscillator of the receiver and that of A (B), respectively, and τ is the time delay. In the MA transmission, f_a and f_b can be estimated at the beginning of each packet. Considering the wireless channels to be frequency flat and constant during the reception time of a packet, H_a (H_b) can be represented with one tap, essentially a complex number. The received signal after the Analog to Digital Converter (DAC) is given by

$$r_n = h_1 x e^{j\omega_1 n+\phi_1} + h_2 y e^{j\omega_2 n+\phi_2}, \tag{4.4}$$

where h_i, ϕ_i, and ω_i, $i = 1, 2$ are the gains, phases and digital frequency offsets imposed by the channel and the receiver circuit, respectively.

4.2.2 Timing recovery

Depending on the sampling rate of the Digital to Analog Converter (DAC), each symbol is represented with a constant number of samples. This parameter is called Samples per Symbol (SpS) and can be found as Tf_s, where f_s is the sample rate of the DAC. After passing the received signal from the matched filter, the next step is to down sample the signal to one sample per symbol before detection.

In the digital domain, depending on the offset between the transmitter and receiver's clocks, only one of the samples representing a symbol is used for detection and the distance between valid samples is equal to SpS. As shown in Fig. 4.1, to find the valid sample, the signal is divided into different branches and decimated with different time delays, from 0 to $SpS - 1$. The signal after downsampling can be expressed as

$$s_k = h_1 a_k e^{j(SpS)\omega_1 n+\phi_1} + h_2 b_k e^{j(SpS)\omega_2 n+\phi_2}. \tag{4.5}$$

Each branch is then used for preamble detection purposes and the result is connected to a switch. The same preamble detection algorithm is ran for all branches. If a preamble is identified, the signal would be annotated with a tag consisting of ϕ_i and ω_i ($i = 1, 2$). The switch then copies only the branch with a tag to output. It is possible to find a preamble in more than one branch. The preamble detector block also includes the mean square error of detection so the switch can choose the branch with the best signal to noise ratio.

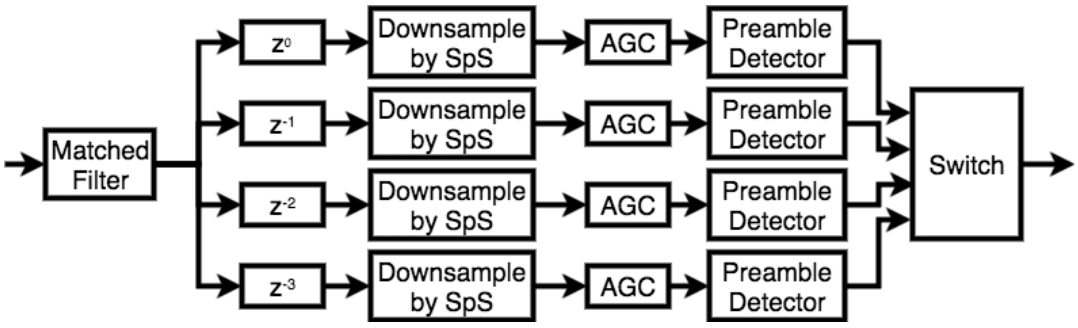


Figure 4.1: Receiver chain.

4.2.3 MA receiver, CFO estimation

To help the receiver estimate CFOs and detect MA transmissions, each frame begins with a Sync preamble and a start of frame delimiter (SFD) part. The sync preamble is a series of zeros and ones, and the SFD is used to verify the detection of a packet. In our work, the Sync preambles for node A and B are "1010... 10" and "0101...01", respectively, which is similar to the packet structure used in the IEEE 802.11 FH PHY.

Before using the signal to detect a preamble, the signal is passed from an Automatic Gain Control (AGC) unit, in order to normalize the channel gains, h_1 and h_2. In a window with a large enough number of symbols, there exists at least one instant that symbols from both sources has an equal or very close phase and have been added constructively. So the AGC adjusts the maximum absolute value of symbols to two. For BPSK, +1 and -1 represents 1 and 0, respectively. If we index all symbols of the sync part starting from 0, the k_{th} received symbols of Sync preamble can be written as

$$s_k = (-1)^{k+1} e^{p_{1,k}} - (-1)^k e^{p_{2,k}}. \tag{4.6}$$

For consistency, we replace all odd indexed symbols of Sync with their additive inverse so the k_{th} symbol can be represented as

$$s'_k = e^{p_{1,k}} - e^{p_{2,k}}. \tag{4.7}$$

The two phases of $p_{1,k}$ and $p_{2,k}$ can be found by solving the equations below

$$\begin{cases} Re(s'_k) = \cos(p_{1,k}) - \cos(p_{2,k}), \\ Im(s'_k) = \sin(p_{1,k}) - \sin(p_{2,k}). \end{cases} \tag{4.8}$$

Since there may be multiple possible answers for these equations, $p_{1,k}$ and $p_{2,k}$ are chosen in a way that they are closest to $p_{1,k-1}$ and $p_{2,k-1}$ and increasing or decreasing linearly.

The frequency and phase offsets are then found by a solving a linear regression according to the following equations,

$$\begin{cases} p_{1,k} = (SpS)\omega_1 k + \phi_1, \\ p_{2,k} = (SpS)\omega_2 k + \phi_2. \end{cases} \tag{4.9}$$

4.2.4 MA receiver, preamble detection

A window with the length of the Sync preamble is always sliding on the received samples. Then each time the above CFO estimation algorithm is applied and the ϕ_i and ω_i $(i = 1, 2)$ are used to decode the next L_{SFD} symbols of the packet where L_{SFD} is the length of the SFD part. If the decoded SFD matches the expected SFD, which is XOR of SFD of node A and node B, then the packet is detected and the rest of the symbols are decoded using the same parameters. In our design, SFDs by nodes A and B are "1010 0110 1101 0110" and "1010 1010 0110 1011", respectively, so the SFD expected in the relay is "0000 1100 1011 1101".

Because of the presence of CFO and phase offset, the constellation is changing from one symbol to another. The right constellation for each received symbol is calculated using the ϕ_i, and ω_i and then used for decoding and mapping, According to the mapping function.

4.3 Performance Evaluations

This section, presents the simulation and experimental results for the PNC transmission. Since the real gain of PNC can been seen in multi-hop transmissions, the same implementation is used for measuring performance parameters in a Multi-hop Physical Layer Network Coding (MPNC). The main MPNC network design used for multi-hop experiments is adopted from [28]. Then using asynchronous experiments the same two hop performance parameters are measured for multi-hop PNC too.

4.3.1 Testbed Experiments

Experiment setting

USRP N210 sets with XCVR2450 daughterboards are used to run the PNC experiments. The software used to develop PNC is GNURadio. The transmission gain of transmitting USRPs is set to 18 dB and the receiver gain of receiving USRPs is set to 5 dB. The carrier frequency is chosen as 2.45 GHz. The length of the Sync preamble is 64 symbols and the SpS used in the experiments is four.

BER, two-hop

To begin with, we first built a two-hop testbed where two sources A and B exchange the information through a relay, as shown in Figure 4.2.

Fig. 4.3 (a) and (b) shows the BER in the MA communications and the end-to-end BER for the two-hop case, respectively. As anticipated, the BER performance deteriorates when the payload length increases, due to the non-stationary wireless channel and the CFOs. Thus, to maintain a low BER, preambles could be repeated periodically. A somewhat surprising result in Fig. 4.3 (a) is that there is a small drop of BER around 150-bit block length. This is because, due to CFO, the phase difference of the two transmissions superimposed at the relay changes periodically. BER with different phase shift is slightly different. After 100-bit was sent, the phase shift returns to the area corresponding to low BER.

The end-to-end BER results in Fig. 4.3 (b) include both the MA transmissions and the broadcast transmissions from the relay to the sources. We further compared the end-to-end BER performance of PNC with that of ANC reported in [15]. The relay in ANC uses an amplify-and-forward approach, instead of the mapping function. From Fig. 4.3 (c), MPNC slightly outperforms ANC, from the USRP testbed experiments. In general, the performance of amplify-and-forward degrades in the low SINR region.

Multi-hop performance

Next, we extended the two-hop scenario to multi-hop scenario, where the number of hops varied from two to five. The transmission and reception of the relays for multiple hops behave similarly as that in the two-hop case, except that the mutual interference and the error propagation problem will lead to a higher BER.

Figs. 4.4 (a) and (b) compare the end-to-end BER and normalized throughput,

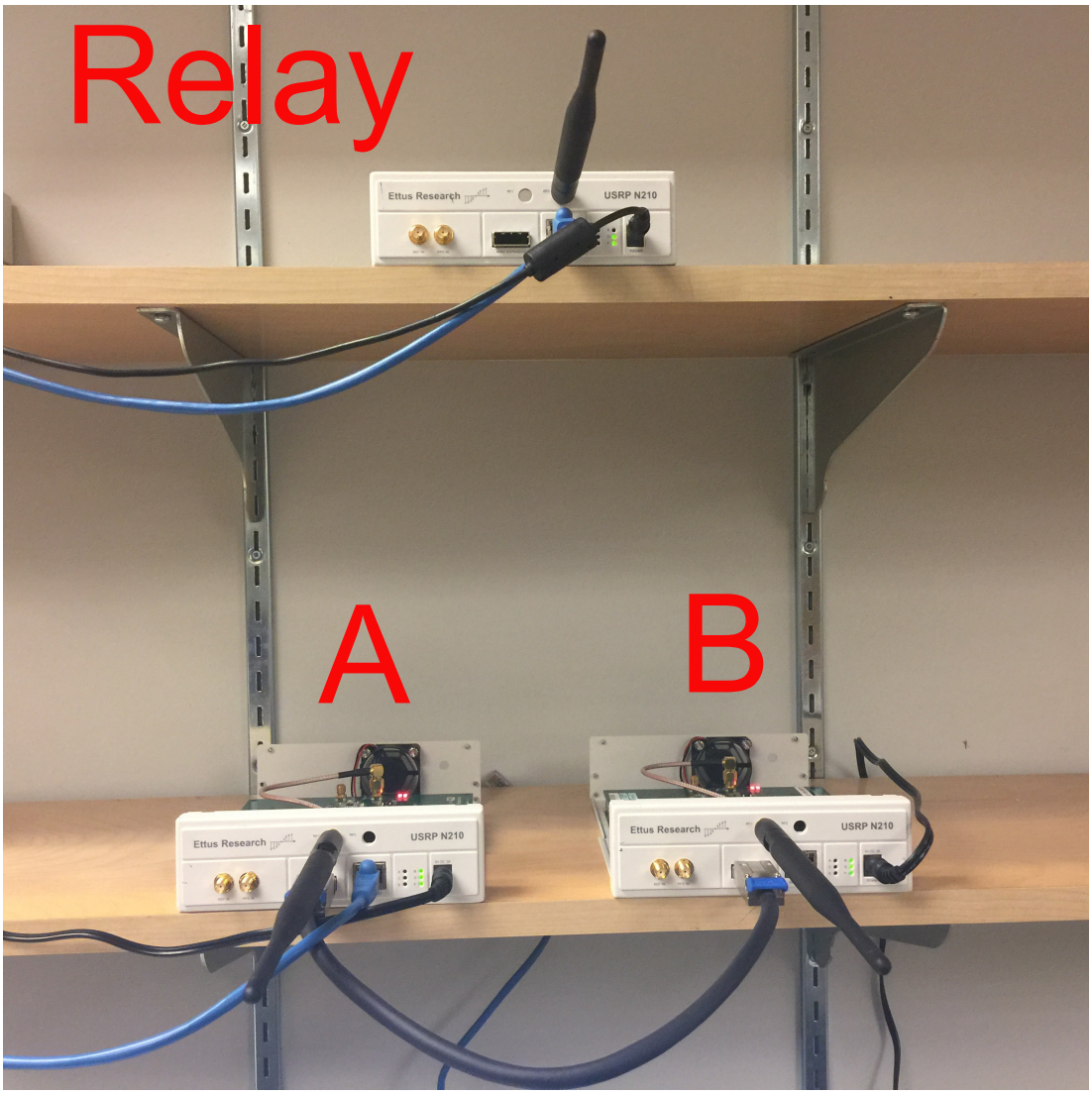
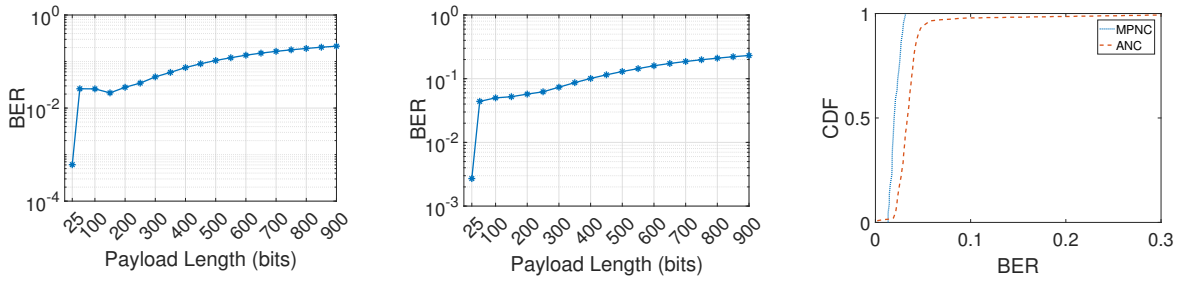


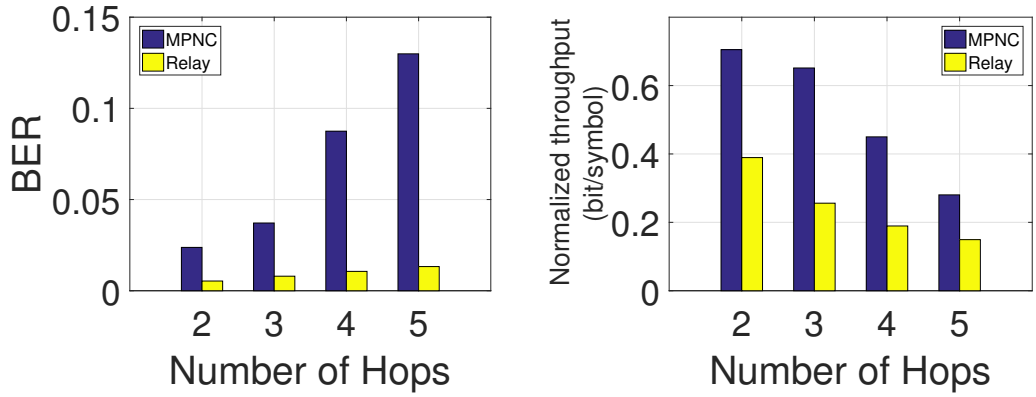
Figure 4.2: Testbed, two-hop scenario.

respectively of MPNC and that with hop-by-hop relay. The normalized throughput is the number of bits successfully received by the destination per symbol duration. From the figures, although the end-to-end BER of MPNC is much higher than that of hop-by-hop relaying, due to both the worse BER performance in MA transmissions and the error propagation problem (i.e., a single MA transmission error may affect multiple end-to-end bit errors in a multi-relay path), the end-to-end throughput of MPNC is much higher than that of hop-by-hop relay. Using the inexpensive USRP testbed, it is encouraging to notice that the performance gain of MPNC ranges from 75% to 80% for the two- and five-hop cases, and around 150% for the three- and



(a) MA transmission BER (b) End-to-end BER (c) MPNC vs. ANC

Figure 4.3: BER performance, two-hop.



(a) End-to-end BER (b) Normalized throughput

Figure 4.4: Multi-hop performance. Multi-hop End-to-End BER are calculated using formulations outlined in table 5.1 of [28].

four-hop cases.

4.3.2 Simulation Results

To evaluate and compare the performance of the proposed MPNC in controllable and repeatable environment with a wide range of settings, we used Matlab and GNURadio to conduct extensive simulations and the results are presented in this section.

Goodput performance comparison

In this section, the goodput of MPNC is studied and also compared with the other state-of-the-art solutions. As we compare the performance of different solutions with the same number of hops, different from the last subsection, here, the transmission power of each node and the hop distance remains constant. The goodput measures

the number of error-free packets received per slot. For the full-duplex relay solution, in the simulation, we made a strong assumption that the self-interference can be fully cancelled. Thus, the performance shown in our simulation can be viewed as the upper bound of it. As the existing PNC method for multi-relay path suffers from the infinite error propagation problem, to make it usable, after detecting each error, the destination must notify all other nodes and the pipeline should be flushed and restarted. The goodput for the PNC method is thus affected by the restarts. Note that with the infinite error propagation problem, an error detection coding have to be used for PNC, but the error correction coding cannot be applied, which is a severe disadvantage for it.

The goodputs are shown in Fig. 4.5. Comparing with the other state-of-the-art algorithms, MPNC has always a higher goodput and can maintain its performance as the number of hops increases. The goodput gains of MPNC over hop-by-hop relay, full-duplex, and PNC is about 3x times, 33%, and 68% when the number of hops is as high as eight. Also, from the figure, the goodput of MPNC remains close to 0.9 packet/slot when the number of hops varied from two to eight, showing that it is quite scalable to long paths.

Effect of CFO on BER

Fig. 4.6 (a) shows the impacts of CFOs on the BER performance. High CFO is set according to the practical number measured from the USRP devices in our testbed, and the low CFO is 0.1 of that. As it can be seen in the figure, CFO compensation is crucial to the performance of MPNC. The figure shows that when CFO is not compensated, the performance is poor (larger than 10% bits in error) no matter whether the CFO is high or low. On the other hand, using the proposed CFO compensation algorithm can improve the BER performance significantly.

Effect of Channel Estimation Error on BER

We studied the effect of channel estimation error on end-to-end BER performance. In Fig. 4.6 (b), δ represents the ratio of estimated channel gain and the real channel gain. From the figure, overall, the BER increases when the channel estimation error increases, while the algorithm can maintain a reasonable BER even with 10% to 20% estimation errors.

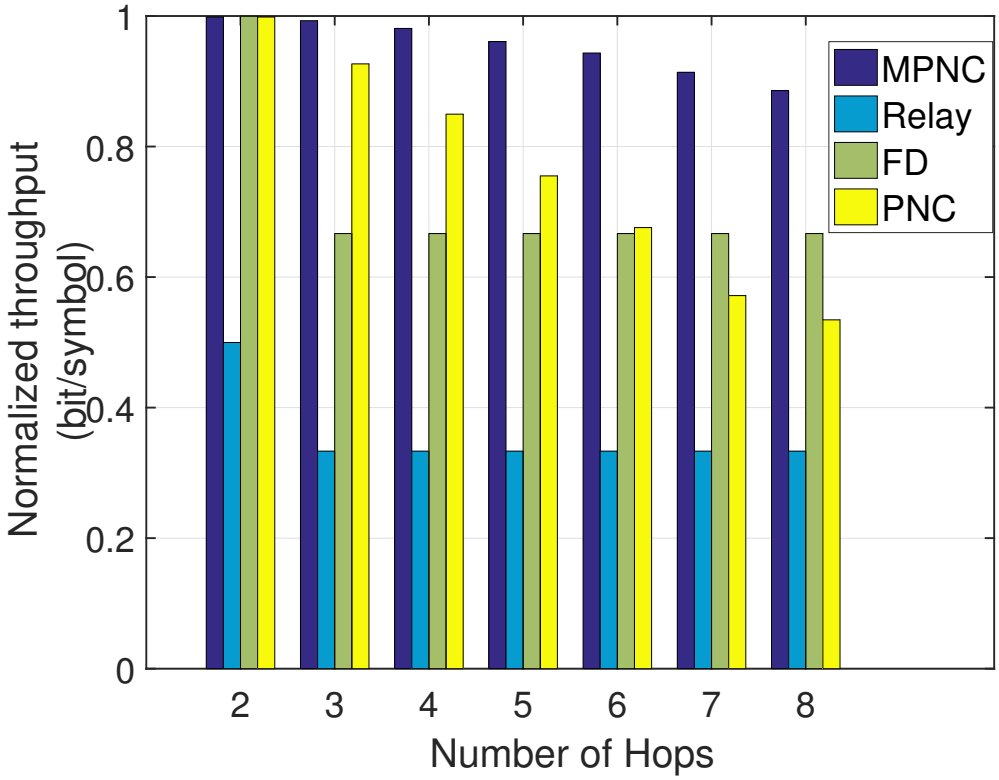
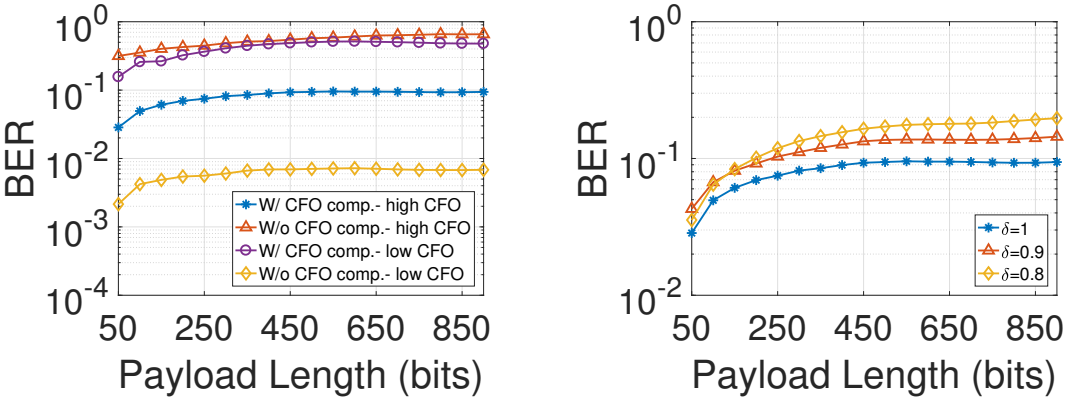


Figure 4.5: Goodput, per-hop SNR = 9, $\alpha = 4$, Payload=256 bits.



(a) CFO compensation (b) Channel estimation error

Figure 4.6: Effect of CFO and channel estimation error on BER.

Chapter 5

Conclusions

Two efficient wireless networking solutions supporting IoT have been the main focus for this thesis. In single-hop communications, super dense networks and IEEE 802.11ah as a promising standard for sensor networks are studied and the hidden terminal problem under GS-DCF has been addressed. An RSS-based grouping scheme has been proposed. The proposed scheme will categorize users in RSS-based groups so nodes in the same group can be close enough that the probability of having a hidden terminal is very low. The performance of RSS-based grouping has been then studied and compared to random and optimal grouping schemes using analytical and simulation results presenting different metrics.

Another problem in future infrastructure networks that can be solved with RSS-based groups is small message size of most sensing data exchanges, which will reduce channel efficiency. [4] showed that small packet size will reduce the overall throughput. RSS-based grouping schemes can easily utilize clustering algorithms where each group is a cluster and a cluster head in each group can act as a relay for other nodes. We intend to cover this application in our future work.

The second half of the thesis has focused on multi-hop transmissions and recognized physical layer network coding as a robust technology for many sensor network solutions. Since practical implementation has been missed in many PNC designs, this thesis proposed solutions to deal with the challenges that a real-life PNC system has to face and validates its feasibility using a testbed implementation on USRP devices. Experiment results show that MPNC can outperform the traditional relay with a throughput gain ranging from 75% to 80% for the two and five-hop cases, and around 150% for the three- and four-hop cases in the real testbed.

There are still many unsolved challenges and improvement areas around MPNC

in general and its practical implementation. Since the relay only receives a combined version of its surrounding nodes, unconventional channel coding algorithms are no longer useful. Design of new channel coding algorithms for the multiple access phase of PNC can improve its performance by a large margin. Multi-path channel and fast fading are also practical issues not addressed by this thesis. Following the same idea proposed in this thesis but modeling the channel as an FIR filter can overcome these challenges. Integrating MPNC with MIMO, accounting for random-hop distances, and uni-directional traffic are other areas that this work can be further improved.

Appendix A

Additional Information

In order to use the PNC implementaiton, first the GNURadio software has to be installed. It can be found in `https://www.gnuradio.org` for Linux, Mac Os, and Windows. In the GNURadio platfrom, any block that is not part of the build-in enviroment is called an Out of Tree (OOT) block. In the next step, blocks written for this thesis need to be imported in the GNURadio enviroment. Below is the solution to some common troubles that one might face while installing the PNC library into a new computer along with some tips for further developing the library in GNURadio platform.

A.1 Installing out of tree modules on Mac OS

If GRC complains that it can not find some blocks like:

```
Error: Block key "name of block" not found in Platform - grc(GNU Radio Companion)
```

Most likely you used a different *CMAKE_INSTALL_PREFIX* for the module than for GNU Radio. Therefore, the blocks of the module ended up in a different directory and GRC can't find them. You have to tell GRC where these blocks are by creating/adding to your /.gnuradio/config.conf something like

```
[grc]
local_blocks_path = /usr/local/share/gnuradio/grc/blocks
```

But with the directories that match your installation.

If this is your problem you have probably installed python modules correctly but GRC can not find modules.

How to find the above address?

If you look at you installation report after sudo make install command you can find out where is the location of *.xml files. it should be something like this:

```
-- Installing: /usr/local/share/gnuradio/grc/blocks/mac_simple_mac.xml
-- Installing: /usr/local/share/gnuradio/grc/blocks/mac_virtual_channel_encoder.xml
-- Installing: /usr/local/share/gnuradio/grc/blocks/mac_virtual_channel_decoder.xml
-- Installing: /usr/local/share/gnuradio/grc/blocks/mac_packet_framer.xml
-- Installing: /usr/local/share/gnuradio/grc/blocks/mac_packet_deframer.xml
-- Installing: /usr/local/share/gnuradio/grc/blocks/mac_packet_to_pdu.xml
-- Installing: /usr/local/share/gnuradio/grc/blocks/mac_802_3_tracker.xml
-- Installing: /usr/local/share/gnuradio/grc/blocks/mac_burst_tagger.xml
```

which indicates that the installation have copied xml files in */usr/local/share/gnuradio/grc/blocks*

A.1.1 UHD commands

UHD is the free and open-source software driver and API for USRP devices. It will be installed with GNURadio. There are some simple commands that one can use after installing GNURadio, specially if you want a simple spectrum analyzer or want to Record I/Q sample stream to a file and use in another software like MATLAB. They are listed below. They all include a "-h" switch which will display a help screen. More information can be found in `https://gnuradio.org/redmine/projects/gnuradio/wiki/HowToUse`

```
uhd_cal_rx_iq_balance
uhd_find_devices
uhd_siggen
uhd_cal_tx_dc_offset
uhd_image_loader
uhd_siggen_gui
uhd_cal_tx_iq_balance
uhd_rx_cfile
uhd_usrp_probe
uhd_fft
uhd_rx_nogui
```

A.1.2 Adding new modules to gnuradio in Mac OS

If you've installed your gnuradio with mac port, first you need to make sure that the mac port python is in use. then when you want to compile PNC modules use this:

```
cmake –DPYTHON_LIBRARY=/opt/local/lib/libpython2.7.dylib
–DCMAKE_INSTALL_PREFIX=/opt/local/ ../
```

after compiling the code(make and sudo make install) put these two lines in *.bash_profile* :

```
export PYTHONPATH=/opt/local/lib/python2.7/site-packages:\$PYTHONPATH
export GRC_BLOCKS_PATH=/usr/local/share/gnuradio/grc/blocks
```

A.1.3 Adding a library for linking

If an outside library has been used for a block. That library might be needed in linking phase of making. The standard place for libraries is */usr/local/lib* , if the library is already build and installed its file must be there. If the user has the .a or .so or .dylib file they may put a copy of it in that folder so the *gcc* can easily find it with the flag *-l<name of library>*. For example if the name of library is levmar, the filename is liblevmar.a and the flag would be *-llevmar*. In order to add the linking flag to all the codes, the function *target_link_libraries* in the file *CMakeLists.txt* found in the *lib* folder of the module has to be modified in the following order. For example in order to add the levmar library the line:

```
target_link_libraries(gnuradio-pnc ${Boost_LIBRARIES} ${GNURADIO_ALL_LIBRARIES})
```

will change to :

```
target_link_libraries(gnuradio-pnc ${Boost_LIBRARIES} ${GNURADIO_ALL_LIBRARIES} levm
```

The same thing happens to all other uses of function *target_link_libraries* in that file. Run the *cmake* command and make again after modifing this file.

A.1.4 OOT modules in Ubuntu

If after compilling and installing a new block in ubuntu, when everything is fine, you stil get this error:

```
AttributeError : 'module' object has no attribute 'block name'
```

In order to fix it, first check the enviroment variable *PYTHONPATH* and make sure it point to where the python blocks are, if it did not solve the problem, then it might be because SWIG is not installed. If that is the case, you must have got this error after cmakeing:

```
disabling swig because version check failed
```

in order to solve it, simply install swig on your ubuntu machine by running the below code:

```
sudo apt-get install swig
```

A.1.5 Blocks with Multiple Inputs

Put the *MAX_INT* parameter as −1. The *input_items* input of the *work* function is a vector of void stars. These void stars can be casted to the desired type of input, e.g. to *gr_complex** in case of complex input. Obviously the number of input ports can be obtained from the size of the vector. Example:

```
int number_of_inputs=input_items.size();
for(int i=0;i<number_of_inputs;i++)
{
 int this loop, (const gr_complex *) input_items[i] is the array for i_th input.
 const gr_complex *in_i = (const gr_complex *) input_items[0];
 now we can iterate in in_i for items of i_th input.
}
```

A.2 Working with *tagged_stream_blocks*

The definition for the *work* function of *tagged_stream_blocks* looks like the one for *general_work* of other blocks. But the difference here is that the input *noutput_items* is "not" number of output items to write on each output stream. In this case, this variable shows The size of the writable output buffer which can be a lot larger than the pdu size. This essentially makes this variable unusable. Instead use *ninput_items[i]* where i is the index of input to find the pdu length.

A.3 Positioning and Tabs with QT

In order to have tabs when using QT as your GUI engine, insert the block, *QT GUI Tab Widget* and put the number of tabs and a label for each tab. In order to put each QT component in the desired tab use the phrase tab@T:x,y in their GUI hint property, where T is the tab index, and x,y is the position you want the component to be at.

Bibliography

[1] Wireless LAN medium access control (MAC) and physical layer (PHY) specifications amendment 8: IEEE 802.11 wireless network management, September 2011.

[2] Zakhia Abichar and J Morris Chang. Group-based medium access control for IEEE 802.11n wireless LANs. *IEEE Transactions on Mobile Computing*, 12(2):304–317, 2013.

[3] Mohammadhossein Alvandi, Mustafa Mehmet-Ali, and Jeremiah F Hayes. Delay optimization and cross-layer design in multihop wireless networks with network coding and successive interference cancelation. *IEEE JSAC*, 33(2):295–308, 2015.

[4] Giuseppe Bianchi. Performance analysis of the IEEE 802.11 distributed coordination function. *IEEE Journal on Selected Areas in Communications*, 18(3):535–547, 2000.

[5] Lee Breslau, Deborah Estrin, Haobo Yu, Kevin Fall, Sally Floyd, John Heidemann, Ahmed Helmy, Polly Huang, Steven McCanne, Kannan Varadhan, et al. Advances in network simulation. *IEEE Computer*, (5):59–67, 2000.

[6] V. R. Cadambe and S. A. Jafar. Interference alignment and degrees of freedom of the k -user interference channel. *IEEE Trans. on Info. Theory*, 54(8):3425–3441, Aug 2008.

[7] B. Chen, V. Yenamandra, and K. Srinivasan. Interference alignment using shadow channel. In *Proc. IEEE INFOCOM*, pages 2128–2136, April 2015.

[8] Klaus Doppler, Mika Rinne, Carl Wijting, Cássio B Ribeiro, and Klaus Hugl. Device-to-device communication as an underlay to lte-advanced networks. *IEEE Communications Magazine*, 47(12), 2009.

[9] Patrick Ho Wang Fung, Sumei Sun, and Chin Keong Ho. Preamble design for carrier frequency offset estimation in two-way relays. In *Signal Processing Advances in Wireless Communications (SPAWC), 2010 IEEE Eleventh International Workshop on*, pages 1–5. IEEE, 2010.

[10] Mohammad Ghasemiahmadi, Yue Li, and Lin Cai. Rss-based grouping strategy for avoiding hidden terminals with gs-dcf mac protocol. In *Wireless Communications and Networking Conference (WCNC), 2017 IEEE*, pages 1–6. IEEE, 2017.

[11] Shyamnath Gollakota and Dina Katabi. *Zigzag Decoding: Combating Hidden Terminals in Wireless Networks*. SIGCOMM '08. ACM, 2008.

[12] IEEE 802.11ah Task Group. 11/1137r14 specification framework for TGah.

[13] IEEE 802.11ah Task Group. TGah functional requirements and evaluation methodology rev.5.

[14] Mayank Jain, Jung Il Choi, Taemin Kim, Dinesh Bharadia, Siddharth Seth, Kannan Srinivasan, Philip Levis, Sachin Katti, and Prasun Sinha. Practical, real-time, full duplex wireless. In *Proc. ACM MOBICOM*, MobiCom '11, pages 301–312, New York, NY, USA, 2011. ACM.

[15] Sachin Katti, Shyamnath Gollakota, and Dina Katabi. Embracing wireless interference: Analog network coding. *Proc. ACM SIGCOMM*, 37(4):397–408, 2007.

[16] Evgeny Khorov, Andrey Lyakhov, Alexander Krotov, and Andrey Guschin. A survey on IEEE 802.11 ah: An enabling networking technology for smart cities. *Computer Communications*, 58:53–69, 2015.

[17] Sumit Khurana, Anurag Kahol, Sandeep KS Gupta, and Pradip K Srimani. Performance evaluation of distributed co-ordination function for IEEE 802.11 wireless LAN protocol in presence of mobile and hidden terminals. In *Proceedings of 7th International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems, MASCOT'99*, pages 40–47, 1999.

[18] Yunbae Kim, Ganguk Hwang, Jungsun Um, Sungjin Yoo, Hoiyoon Jung, and Seungkeun Park. Optimal throughput analysis of a super dense wireless network with the renewal access protocol. In *IEEE International Conference on Communication Workshop, ICCW'15*, pages 2194–2199, 2015.

[19] Linghe Kong and Xue Liu. mzig: Enabling multi-packet reception in zigbee. In *Proc. ACM MOBICOM*, MobiCom '15, pages 552–565, New York, NY, USA, 2015. ACM.

[20] Lu Lu, Taotao Wang, Soung Chang Liew, and Shengli Zhang. Implementation of physical-layer network coding. *Physical Commun.*, 6:74–87, 2013.

[21] Dmitri Moltchanov. Distance distributions in random networks. *Ad Hoc Networks*, 10(6):1146–1166, 2012.

[22] Orod Raeesi, Juho Pirskanen, Ali Hazmi, Toni Levanen, and Mikko Valkama. Performance evaluation of IEEE 802.11 ah and its restricted access window mechanism. In *IEEE International Conference on Communications Workshops, ICC'14*, pages 460–466, 2014.

[23] Hariharan Rahul, Haitham Hassanieh, and Dina Katabi. Sourcesync: A distributed wireless architecture for exploiting sender diversity. In *Proc. ACM SIGCOMM*, SIGCOMM '10, pages 171–182, New York, NY, USA, 2010. ACM.

[24] Hsueh-Wen Tseng, Yao-Chung Fan, Shiann-Tsong Sheu, and Shaiu-Yi Ou. An effective grouping scheme for avoiding hidden node problem in IEEE 802.15. 4-based wireless sensor networks. *ACM SIGAPP Applied Computing Review*, 14(1):30–40, 2014.

[25] Athanasia Tsertou and David I Laurenson. Revisiting the hidden terminal problem in a CSMA/CA wireless network. *IEEE Transactions on Mobile Computing*, 7(7):817–831, 2008.

[26] Haitao Wu, Fan Zhu, Qian Zhang, and Zhisheng Niu. Wsn02-1: Analysis of IEEE 802.11 dcf with hidden terminals. In *IEEE Global Telecommunications Conference, GLOBECOM'06.*, pages 1–5, 2006.

[27] Sung-Guk Yoon, Jeong-O Seo, and Saewoong Bahk. Regrouping algorithm to alleviate the hidden node problem in 802.11 ah networks. *Computer Networks*, 105:22–32, 2016.

[28] Haoyuan Zhang. *Cross-layer design for multi-hop two-way relay network.* PhD thesis, 2017.

[29] Shengli Zhang, Soung Chang Liew, and Patrick P Lam. Hot topic: physical-layer network coding. In *Proc. ACM MOBICOM*, pages 358–365, 2006.

[30] Lei Zheng, Minming Ni, Lin Cai, Jianping Pan, Chittabrata Ghosh, and Klaus Doppler. Performance analysis of group-synchronized dcf for dense IEEE 802.11 networks. *IEEE Transactions on Wireless Communications*, 13(11):6180–6192, 2014.

[31] Lei Zhong, Yozo Shoji, Kiyohide Nakauchi, and Suyong Eum. BE-DCF: Barring-enhanced distributed coordination function for machine type communications in IEEE 802.11 networks. In *IEEE International Conference on Communications Workshops, ICC'14*, pages 467–471, 2014.