Energy Efficient Packet Size Optimization for Wireless Ad Hoc Networks

Mobin, Iftekharul

Energy Efficient Packet Size Optimization for Wireless Ad Hoc Networks

by

Iftekharul Mobin

Supervised by

Dr. Raul J. Mondragon

A thesis submitted to the University of London for the degree of
Doctor of Philosophy

The School of Electronic Engineering and Computer Science
Queen Mary University of London
United Kingdom

6^{th} September, 2013

TO MY MOTHER

Abstract

Energy efficiency is crucial for ad hoc networks because of limited energy stored in the battery. Recharging the nodes frequently is sometimes not possible. Therefore, proper energy utilization is paramount. One possible solution of increasing energy efficiency is to optimize the transmitted packet size. But, we claim that only optimal packet size can not boost the energy efficiency in the noisy channel due to high packet loss rate and overhead. Hence, to reduce the overhead size and packet loss, compression and Forward Error Correction (FEC) code are used as remedy. However, every method has its own cost. For compression and FEC, the costs are computation energy cost and extra processing time. Therefore, to estimate the energy-optimize packet size with FEC or compression, processing energy cost and delay need to be considered for precise estimation. Otherwise, for delay sensitive real time applications (such as: VoIP, multimedia) over ad hoc network, energy efficient optimal packet size can be overestimated.

We will investigate without degrading the Quality of Service (QoS) with these two different techniques FEC and compression, how much energy efficiency can be achieved by using the energy efficient optimal packet size for different scenarios such as: single hop, multi-hop, multiple source congested network etc. This thesis also shows the impact of time variable channel, packet fragmentation, packet collision on the optimal packet size and energy efficiency.

Our results show that, for larger packets, error correction improves the energy efficiency in multi-hop networks only for delay tolerant applications. Whereas for smaller packets, compression is more energy efficient most of the cases. For real-time application like VoIP the scope of increasing the energy efficiency by optimizing packet after maintaining all the constraints is very limited. However, it is shown that, in many cases, optimal packet size improves energy efficiency significantly and also reduces the overall packet loss.

Acknowledgments

My foremost gratitude belongs to my supervisor, Dr. Raul J Mondragon, who has provided academic supervision, support, and persistent encouragement. His extensive knowledge and ideas has been a valuable asset to me. My research benefited tremendously from our weekly discussions. With his effective guidance, constructive comments and detailed feedback helped me continuously to write this thesis and achieve my goal.

My sincere thank goes to Dr. Gareth Tyson for his valuable suggestions and reading this report. Also, I am thankful to Amna Abdul Wahid for her proof reading of this thesis. In addition I would like to thank Dr. Athen Ma for being my second supervisor. Their indispensable advice and critics helps me lot to refine my knowledge. Thanks to my fellow labmates in the Networks Research Group: Ammar Lilamwala, Xian Zhang, Lexi Xu, Rehana Kawsar, Xin Chen, Sabri Zaman etc.

I would like to extend my acknowledgement to my friends in the department and outside the university Monirujjaman Khan, Al Basir, Maruf Iqbal, Rashedul Alam, Hasan Amin etc. My studies would not have been so pleasant without their friendship.

I am highly obligated to my family for their endless love and inspiration that gave me enormous moral support and motivate me to complete my work in a distant Land. My heartiest thanks go to my parents for teaching me to be an honest and diligent person, and to my elder and younger sisters for their love and friendship. Finally, with my love and gratitude, I would like to dedicate this thesis to my mother for her continuous support and encouragement.

Table of Contents

List of Figures

List of Tables

List of Abbreviations

ACK	Acknowledgment
ADC	Analog to Digital Converter
AP	Access Point
ARQ	Automatic Repeat Request
BCH	Bose Chaudhuri Hocquengum
BER	Bit Error Rate
BM	BerlekampMassey
CBR	Constant Bit Rate
CDMA	Code Division Multiple Access
CODEC	COder DECoder
CRC	Cyclic Redundancy Check
CTS	Clear To Send
FEC	Forward Error Correction
GSM	Global System for Mobile Communications
ICT	Information Communication Technology
LAN	Local Area Network
MAC	Medium Access Control
NAM	Network Animator
NS2	Network Simulator 2
OMNET	Optical MicroNetworks
OPNET	Optimized Network Evaluation Tool

OSI	Open System Interconnection Model
OTCL	Object Tool Command Language
P2P	Point to Point
PER	Packet Error Rate
PHY	Physical
PSTN	Public Switched Telephone Network
QoS	Quality of Service
RF	Radio Frequency
RTCP	Real-time Control Protocol
RTP	Real-time Transport Protocol
RTS	Request To Send
SNR	Signal to Noise Ratio
SIP	Session Initiation Protocol
TCL	Tool Command Language
TCP	Transmission Control Protocol
TDMA	Time Division Multiple Access
UDP	User Datagram Protocol
VoIP	Voice over Internet Protocol
VoWLAN	Voice Over Wireless Local Area Network
WAN	Wide Area Network
WiFi	Wireless Fidelity
WiMAX	Worldwide Interoperability for Microwave Access
WLAN	Wireless Local Area Network
WNIC	Wireless Network Interface Card
WSN	Wireless Sensor Network
WWAN	Wireless Wide Area Network

List of Symbols and Notations

Symbol	Description	Location
	Chapter 02	
n	number of bits	section 2.4
τ	number of error checking bits	section 2.4
t	error correction strength	section 2.4
m	an integer value where $m \geq 3$	section 2.4
ℓ, h, L	payload, header and payload size	section 2.4
b	raw channel bit error rate	equation (5.4)
P_{new}	packet error rate with FEC	equation (5.4)
T_{dec}	time of decoding	equation (2.7)
$T_{add,mul}$	one addition or multiplication instruction execution time of microprocessor	equation (2.4) and (2.5)
t_{cycle}	one cycle time of microprocessor	equation (2.5)
I_{proc}, V_{proc}	required current and voltage to process one block of data by microprocessor	equation (2.7)
E_{dec}	total energy for decoding a block of data packet	equation (2.7)
$\ell^h_{com}, \ell^p_{com}$	compressed header or payload size	section 2.7
L_h, L_p	Whole packet size with compressed header or compressed payload size	section 2.7 and equation (2.10) , (2.11)
r_{com}	reliability function with the compressed packet	equation (2.10) and (2.11)

Symbol	Description	Location
Chapter 03		
α, ℓ, τ	header, payload, and trailer size	equation (3.1)
per, ber	packet error rate, bit error rate	equation (3.1)
SNR	signal to noise ratio	equation (3.2)
R_{mod}	highest bit rate of modulation scheme	equation (3.2)
B_t	system bandwidth	equation (3.2)
E_{tx}, E_{rx}	transmission or reception energy	equation (3.4)
$E_{enc-dec}$	encoding decoding energy	equation (3.4)
R	data rate	equation (3.2)
B_t	system bandwidth	equation (3.2)
P_{tx}, P_{rx}	transmission power and reception power	equation (3.5)
γ	path loss exponent	equation (3.6)
λ, D	wave length, distance between the nodes	equation (3.6)
NF_{rx}	internal circuit noise	equation (3.7)
F_{mar}	fading margin	equation (3.7)
k	boltzmann constant	equation (3.7)
R_{th}	receiver threshold	equation (3.9)
$P_{ber}^{gg}, P_{ber}^{bb}$	packet error rate in good channel condition and bad channel condition	equation (3.11)
ber_{high}, ber_{low}	bit error rate in bad channel condition and good channel condition	equation (3.11)
P_{gg}, P_{bb}	probability of staying in good condition, and bad condition	equation (3.12)
P_{gb}, P_{bg}	probability of changing states from good to bad and bad to good condition	equation (3.12)
P_{tran}	transition probability	section 3.2
π_g, π_b	steady state probability of being in good and bad state	section 3.2

Symbol	Description	Location
P_{GE}	average packet error rate with Gilbert-Elliot model	equation (3.14)
P_{error}	packet error rate in multi-hop network	equation (3.15)
N, n	number of hops, number of nodes	equation (3.15)
	Chapter 04	
k_j	constant to represent the required energy for a single bit communication cost	equation (4.1)
M	maximum transfer unit size (MTU) packet	equation (4.5)
m_{pay}	payload size of MTU packet	equation (4.5)
n	number of MTU packets	equation (4.5)
s	total bits to transmit	equation (4.5)
η_{mtu}	energy efficiency with the MTU packet size	equation (4.8)
per_{mtu}	packet error rate for the MTU packet	equation (4.8)
η_{GE}	energy efficiency with the Gilbert-Elliot model	equation (4.8)
r_{GE}	reliability with the Gilbert-Elliot model	equation (4.8)
$E_{in}^{mul}, E_{out}^{mul}$	total energy input and output with the multiple hop	equation (4.10)
E_t, E_r	transmission and reception energy in the multi-hop network	equation (4.10)
N	number of hops	equation (4.12)
hop	number of hops in a multi-hop network	equation (4.13)
per_{multi}	packet error rate in multi-hop network	equation (4.14)
per_1, per_2	packet error rate between the nodes of first hop and second hop in a multi-hop network	equation (4.14)
per_{N-1}	packet error rate between the last link last hop in a multi-hop network	equation (4.14)

Symbol	Description	Location
	Chapter 05	
η_{re}	energy efficiency with the ARQ protocol	equation (5.1)
num_{ret}	number of retransmission with ARQ protocol	equation (5.1)
r_{mul}^{fec}	reliability function for the multi-hop topology with FEC	equation (5.8)
r_{edge}^{mul}	reliability function of using FEC in only the source and destination node (edge nodes)	equation (5.11)
ϕ	compression ratio of any data payload size	equation (5.12)
L_{com}^{p}, L_{com}^{h}	compressed packet size with the compressed payload and compressed header	equation (5.12) and (5.14)
α_{com}	Compressed header size	equation (5.12)
	Chapter 06	
BA	backoff window size	equation (6.1)
CW	contention window size	equation (6.1)
S	slot size in physical layer	equation (6.1)
$p_{r/c/d/a}$	packet error rate of RTS, CTS, DATA and ACK packet	equation (6.3)
$L_{rts/cts}$, $L_{data/ack}$	packet size of RTS, CTS and DATA, ACK packet	equation (6.4)
$P_{err}^{rc}, P_{err}^{b}$	packet error rate with the RTS CTS scheme and with the Basic scheme	equation (6.5)
$\overline{BK}$	average backoff window size	equation (6.10)
P_{col}	collision probability	equation (6.10)
$\overline{BK}_{rts-cts}$	average backoff window size with the RTS CTS scheme	equation (6.11)
$\overline{BK}_{basic}$	average backoff window size with the BASIC scheme	equation (6.12)

Symbol	Description	Location
Chapter 06		
p^s_{rc}, p^s_{bas}	successful packet transmission probability after i times of collision with RTS CTS and Basic scheme	equation (6.11) and (6.12)
$p^r c_{col}, p^b_{col}$	probability of collision with RTS CTS and Basic scheme	equation (6.13)
$P^{suc}_{rts-cts}, P^{suc}_{basic}$	probability of successful transmission with RTS CTS and Basic scheme	equation (6.11) and (6.14)

Chapter 1

Introduction

The Information Communication Technology (ICT) sectors of the whole world consumes almost 3%-8% energy of the total global energy consumption [HVPD09, FZ08, PVD+08]. With the rapid development in the ICT sectors the users are growing very fast and as a consequence the energy need in the ICT sectors is also increasing [VVHC+10, Web08]. Since, the energy consumption is the main cause of global CO2 emissions, the increasing energy need raise the CO2 emission level [VVHC+10, Par11, KDR+12]. Among the whole energy consumption globally for the ICT sectors about 50% of the total energy used in the ICT sector is consumed by the wireless access equipments [Sch10, LPV+08]. For this reason, one of the major concern of the wireless network providers and operators are to reduce the energy cost and CO2 emissions. The greener technologies and methods for the wireless networks are getting higher priority to improve efficiency. The energy consumption estimation to minimise the wireless network energy cost become highly demand-able research area. This thesis is about the energy efficient packet size estimation for the wireless ad hoc network.

1.1 Motivation

Nowadays, wireless networks have been achieved drastic popularity as an ad hoc network [AWSC02, PP00]. Ad hoc network is a kind of Wireless Local Area Network (WLAN) that can be formed with different technologies such as: zigbee, bluetooth, WiFi etc. It is more flexible than traditional wireless networks and provides lots of extra facilities such as: low-cost, self configuring and required little power to operate. The nodes are capable of communicating over infrastructure-less environment. The routing path can be reconfigured dynamically. The nodes can act like a router to forward the data to the destination by adopting multi-hop route.

Recently, ad hoc networks have been emerged as a promising prospect with a wide range of potential applications, such as: environment monitoring, wildlife habitat monitoring, vehicular object tracking, health care patient monitoring, security surveillance etc [WHE02, AKK04, LPV$^+$08, YK10]. Moreover, ad hoc networks can be implemented as a temporary network for disaster recovery or collaboration between the team members to communicate in an infrastructure-less environment during emergency.

In spite of having many advantages of ad hoc networks, still there are lots of problems to resolve [MFAJC13, San12] to implement the network in real-time. Ad hoc nodes are highly resource constrained (e.g. limited battery, small memory, short transmission range, computational complexity etc). They are often affected by interference, packet loss, security threats and many other problems. But one of the crucial part of design and implementation of ad hoc network for longer service time is energy efficiency[HCB00, PFSK08]. Because one of the main reason is Wireless Network Interface Cards (WNIC) are notorious for quickly depleting energy supplies whereas the nodes are equipped with limited battery energy. Hence, during the data communication the nodes run out of energy very quickly. One simple solution could be having high capacity batteries. But unfortunately the battery technology has not been improved compared with the growing need of ad hoc applications. Another reason is since it is a cheap technology there is not enough concern about the energy consumption. Therefore, proper energy utilization is paramount and extensive research is still needed to improve the energy efficiency for ad hoc networks.

1.2 Research Background

It is already stated that the WNIC consumes the maximum amount of energy of ad hoc devices for the data transmission and reception. In many research papers it has been proposed to turn off/on (sleep/wakeup) WNIC to save energy [NG10, CSE04, CHXHH09, Joh04, GPSN06, SBYG11]. The main strategy is to keep the most power consuming components WNIC card in sleep mode as much time as possible to save energy. Thus the energy consumption by the WNIC will be less during the idle period. But it is difficult to predict when exactly the node has to wake up to receive packets. To tackle this problem some other solution has been proposed [NG10, ACW+07, LDL10, PFSK08].

Including the WNIC active/sleep scheduling engineering, extensive literature has been published on how to improve the energy efficiency for ad hoc networks [CM99, CLL05, SSHI+01, WHE02, NG10, SAM03, WhZhYp07, JGS02, JXA04] in different layers of the protocol stack. These energy efficient protocols and methods can be categorised mainly in to four areas: energy efficient application layer protocols [TDV08], energy efficient routing protocols [HCB00, AKK04], energy efficient scheduling protocols [WHE02, NG10], and energy efficient physical layer protocols [SSHI+01]. Different protocols have been adopted different techniques and some methods have improved the energy efficiency impressively. However, every technique and methods have their own pros and cons.

We are particularly interested in a distinct approach packet size optimization to increase the energy efficiency for the ad hoc network [DSSY07, VA08, YF09, YK10, Dom11, BPPS10]. The cost of energy per bit is estimated in [LS98, VA08, XJA05] for different packet sizes. In [LS98] it is showed that by optimizing packet size at least 50% power can be saved. Since, packet is a small unit of data for communication, our presumption is, if energy efficiency can be improved by packet size optimization then significant energy gain can be possible.

Packet size can be optimized for various technologies such as: Zigbee, WiFi or bluetooth etc [VA08, YF09]. We are particularly interested in IEEE 802.11 standard (details in chapter 6) which is mostly known as WiFi Network. Currently for different technologies there are Maximum Transfer Unit (MTU) threshold sizes [IEE99, Wir07] which defines the maximum transferable bytes in one packet. Whenever the transmitted packet size become bigger than the threshold size the packet is split into multiple packets. Otherwise the packet size is determined according to the payload generated from the application layer.

The payload is added with the headers and trailers of other layers. Hence, to obtain an optimal size, either packets needs to split or aggregate to achieve the specific size. For smaller packet size that is less than the optimal, whole packet becomes an accumulated optimal packet which is formed by multiple small payloads. Whereas a packet size larger than optimal is chopped off into multiple packets. If payload size is not multiple of the optimal packet size the accumulator waits for a certain time and then send it through the MAC layer [WXC+10].

That means, optimal packet size acts like a MTU size (see chapter 2 for details). MTU size can be changed manually by using *ifconfig* command (e.g. Unix machine). It changes the frame length of the link layer directly. One problem of this approach is if TCP application is used, it would need to restart the TCP connection multiple times for dynamic optimal packet size [LS98]. Hence, UDP connection is preferred for packet size optimization [LS98, VA08, YF09]. Another way of defining the packet size is to set the packet size in transport layer. The IP packet will be fragmented and chopped of according to the defined size and then reassembled in the received end. However, the problem of this method is if only one IP fragment is lost or corrupted whole frame is considered as error packet. Most upper layer application layer packet size optimization has been proposed in [WXC+10]. In this approach there is algorithm to define the optimal payload size in the application layer. In our research application layer packet size optimization is followed.

During the transmission or reception of a packet, large portion is only wasted for the data administration purpose for the headers and trailers. Since, header and trailers are fixed in size for particular OSI layer, larger frames become more efficient. However, larger packets are more error prone in the noisy channel. The optimal packet size can make a balance between packet overhead, packet loss, and thus energy efficiency increases [VA08, YF09]. The optimized packet can reduce the wastage and minimise the energy consumption. But the wireless radio is dictated by various parameters. Hence, optimized packet size can be changed according to different conditions such as: channel noise, network topology scenario, network environment, number of hops or number of sources etc. Therefore, packet size optimization is not straight forward. Many research paper has been published on how to optimize the packet size considering different parameters for different scenarios [BPPS12, VA08, JA11a, JA11b, LMIS12, CG06].

1.3 State of the Art

The relationship between the nodes distance, channel noise with the packet size has been explored in [JA11b]. The same authors of [JA11b] has been proposed an algorithm to adopt the optimal packet size from a list of packet size look up table which are obtained from theoretical analysis in [JA11a]. However, the problem of [JA11a] and [JA11b] are the multiple source or multi-hop topology is not being considered. It is mentioned in [JA11b] that for the data transmission scheduling for the multiple source CSMA protocol has been applied. But how the channel condition and optimal packet size interact with the scheduling is not being analyzed.

Application oriented packet size optimization has been done in [Dom11, LQK^{+}12]. In these two papers the optimal packet sizes are estimated for the wearable sensor devices for the body sensor networks. The sensors nodes are scattered throughout the body to collect the patient information. The gateway nodes collect all the information, forward it to the monitoring station and can be accessible via internet. But synchronization between the nodes, how the aggregator node collects data, data forwarding schemes are not mentioned clearly in [Dom11]. On contrary more realistic approach has been given in [LQK+12]. In this paper [LQK+12] the sensor nodes (Zigbee) collect the information from the patient body. A WiFi node pull the data from the sensor nodes by using TDMA or CSMA data scheduling protocol and aggregates the collected information. The WiFi aggregator node estimates the energy efficient optimal packet size based on the packet delivery ratio and request the Zigbee nodes to adopt the optimal packet size.

In [VA08, YF09] an optimization framework has been designed to estimate the optimal packet size considering multi-hop routing, error correction, packet collision for multiple source etc. Hence, these two paper [VA08, YF09] have covered most of the essential scenarios of ad hoc networks for packet size optimization. However, in [YF09] the packet size is optimized specially for the Zigbee technology for the tiny sensor nodes whereas the [VA08] is for the WiFi network. The [VA08] is criticised for not considering the interference for densely populated network and this drawback is recovered in [BPPS12]. Specifically the energy efficient optimal packet size is estimated for the underwater sensor networks in [BPPS12]. The impact of the optimal packet size on two different scheduling protocol is investigated for underwater ad hoc communication. The energy efficiency, latency and deployment area has been considered to estimate the optimal packet size. Moreover, collision and multiple hops are also taken into consideration.

From the above literature review one thing is clearly stated that packet size can be optimized for different ad hoc networking scenarios. Whenever the network scenario changes, the optimal packet size also changes accordingly. That means the optimal packet size needs to be dynamic to achieve maximum efficiency in different situation. Very few [LQK+12, JA11a, WXC+10, KHZ11] have been mentioned on how to adopt the optimal packet size at runtime dynamically.

In [WXC+10] packet size optimization has been done dynamically based on the link condition. To estimate the link quality the Packet Reception Rate (PRR) is considered. A dynamic packet size transmission module has been assumed that check, whether the application message is less than or greater than the Optimal Packet Size (OPS). If the application message is less than OPS packet is aggregated or fragmented and after certain timeout limit the packet is transmitted. An optimal packet size adaptation algorithm also has been proposed in [KHZ11]. Beside the optimization algorithm their main contribution is packet collision has been taken into consideration to estimate the packet loss rate for adopting dynamic packet size. The access point broadcast channel occupancy within the network and based on the channel condition the neighbor nodes adopt the best packet size during runtime.

But one of the main problem to obtain dynamic packet size is some of the ad hoc nodes (e.g. Zigbee) do not have very high computation power and highly resource constraints. Moreover the dynamic packet size adaptation needs some extra energy to adopt the algorithm. Furthermore, adopting the dynamic packet size includes extra overhead to carry the channel condition information to the neighbour nodes [WXC+10, CG06]. For an example the signal to noise ratio [CG06] or packet delivery ratio [WXC+10] etc. Hence, there is still extent of improvement.

1.4 Research Objectives

In this research we want to find out optimal packet size for different network scenarios so that these results can be used for dynamic packet size optimization. Further dynamic optimal packet size technique will be adopted to implement energy efficient system.

Packet size optimization has been proposed by many researchers before [BPPS12, VA08, JA11a, JA11b, LMIS12, CG06]. The major problem of improving energy efficiency by packet size optimization is packet loss and overhead. Previously researchers have been adopted the error correction method to improve the energy efficiency by reducing packet loss rate [SAM03, VA08]. But overhead problem has not been considered. To reduce the overhead packet compression is proposed in this thesis for optimal packet size to improve energy efficiency even more. It is obvious that compression will reduce the overhead, increase throughput and hence efficiency will increase. However, there is a trade-off. Because the compression has its own costs. Moreover, it increases the delay of the network. Hence, whenever the packet processing cost become smaller than packet transmission cost and delay remain within the tolerance level for any particular application only then it will be efficient.

- Previously researches have been illustrated separately on packet size optimization, FEC on noisy channel, compression on different traffic etc. In this thesis we put together these three techniques and observe the impact on energy efficiency and network performance.

- However, adding a new protocol adds extra cost (such as: extra processing energy, processing time etc). After applying the compression we will analyse the impact on the energy efficiency and the Quality of Service (QoS) of the network considering delay as well. Our investigation shows most energy efficient optimal packet size for different scenarios considering FEC, compression and QoS delay.

- For delay tolerant ad hoc networks it is fair to assume that Quality of Service (QoS) is not vital [CLL05, SSHI+01, WHE02]. But for real time applications (such as Voice over IP (VoIP) or multimedia) QoS parameters such as: delay, packet loss, bandwidth etc need to be taken into consideration strictly. Packet size optimization for the VoIP service has been studied in [MT09, HA06, FK11, KRJ08, Obe08]. However, energy efficiency has not been taken in to consideration before. We will investigate after maintaining a standard QoS parameter delay and packet loss, how much energy efficiency can be achieved by using optimal packet size for VoIP service.

- As a highly QoS sensitive application VoIP is used as a reference in this research. Including packet loss, packet size also has an impact over delay. Because the end nodes take more time to process and transmit larger packets. In contrast smaller packet sizes are affected by the extra overhead of packet controlling bits (header and trailer). On this account in this thesis the packet size optimization benchmark is energy efficiency, packet loss and delay.

1.5 Research Methodology

The beginning of this thesis energy efficient optimal packet size is estimated for very basic scenario without considering service delay, channel noise etc. After that the estimation is extended considering noise, MTU, multiple source, multiple hops etc. In several research papers Forward Error Correction (FEC) has been proposed to correct the error packet and improve energy efficiency [VA08]. Therefore, we have analyzed the existing research to cross check at first after that analysis is extended further. The FEC and compression are adopted to estimate energy efficient packet size for single hop, multi-hop topology. To observe the impact of delay on a delay sensitive application, VoIP is taken into consideration as a reference and optimal packet size is estimated for various scenarios.

At first it is assumed two nodes are in one hop distance and there is only a single source in a noise free channel. There is enough bandwidth and does not have any packet loss or congestion. The packet is transmitted by the source node and received by the destination properly. There are many network parameters such as: congestion, channel noise, coherence time of connectivity can significantly affect the performance results. However, to start the estimation, simplest scenario is assumed and then more constraints are imposed to observe the impact on optimal packet size.

The summation of total energy that has been used for the packet generation, transmission, reception and processing are considered as an energy input to communicate one single packet from one end to another end. On the destination end from the total energy input, user can get the useful information. But a whole packet contains control bits, data bits together. Control bits define the protocol specifications which are used only for the packet administration purpose. Only data bits (payload) contain useful information. Therefore, out of the whole energy input only the energy for the payload can be obtained as an output energy.

If the packet size increases the ratio of energy input and output, the throughput will be higher. But, in the noisy channel due to channel error larger packets have the more probability to be corrupted and lost. Considering packet error probability, if energy efficiency is estimated for a noisy channel, the energy efficiency must be lower due to excessive packet loss. Energy efficiency also can be lower for shorter packet size due to heavy overhead. Hence, for a specific packet size energy efficiency attains the maximum level which is the optimal packet size.

But optimal packet size does not reduce the PER, its only makes a balance between packet loss and overhead. Hence, still there will be packet loss due to corrupted packet and energy can be wasted. Hence, if error correction coding [KSPR04, KS06] is used to correct error bits, it is expected that energy efficiency will improve.

However, FEC [HSI06, JS02, WWWL10, WH02] increases the overhead by adding extra error correction code with each packet and extra processing cost is also needed to correct the error bits. Hence, after using the FEC, overhead and processing cost are reconsidered.

It is stated on the previous paragraph that larger packet sizes increases the packet error rates and as a remedy one possible solution could be adopting the FEC method. But due to overhead (header and trailer) still some energy will be spent that increases the input energy cost for the packet. Hence, compression method is applied [Jon05, Com06]. Hence, the overhead energy loss will be less and energy efficiency should increase. However, every technique has its' own cost and for the compression the cost would be the energy cost for compression and decompression [Jon05] and processing delay for the compression-decompression algorithm. Therefore, packet size is not only optimized considering energy efficiency, also optimized considering delay.

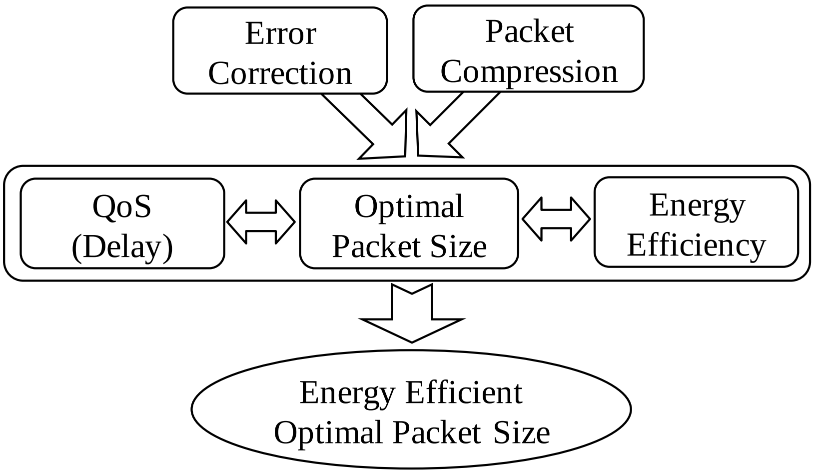

Figure 1.1: Research Scheme

This thesis estimates the packet size for some scenarios which is not estimated before. For an example, packet fragmentation is taken into consideration, energy efficient optimal packet size is estimated for a time varying channel, compression method for the optimal packet size estimation etc. The figure 1.1 shows a vivid illustration of the proposed energy efficient packet size optimization scheme clearly.

1.6 Structure of the Thesis

Chapter 2 describes the packet structure of WLAN and various protocols. Packet overhead, header trailer descriptions are given in terms of OSI layer protocol stack. It proposes the error correction technique for the noisy channel, error correcting codes, overhead and processing energy cost of error correction code.

It also shows the packet compression technique. It illustrates various compression methods and applications and processing energy cost for the packet compression.

Chapter 3 shows the impact of channel noise over packet size. It explains the transmission power and reception power estimation methodology according to channel condition and the effect of noise on energy cost per bit communication.

Chapter 4 illustrates the energy efficient packet size estimation methodology, and energy efficiency metric for the ideal channel and error pone channel. Then energy efficiency is estimated for the time variable channel, multi-hop network etc.

Chapter 5 shows the energy efficient packet size estimation with FEC and Compression methods. FEC and Compression comparison are illustrated in vivid.

Chapter 6 represents multiple source contending busy channel and energy efficient optimal packet size estimation. It describes how optimal packet size is estimated considering packet collision, packet error rates together.

Chapter 7 proposes the energy efficient packet size for wireless VoIP protocol considering error correction, header compression for various scenarios. The trade-off between the energy efficiency and QoS for the VoIP is analyzed extensively for these two methods considering optimal packet size.

Chapter 8 draws conclusions of the work of this thesis, contributions and points out the areas for possible future research and extension of this work.

Appendix A shows the simulator verification, process and simulation methodology. Existing error model and error correction technique of the simulator. Simulation output analysis procedure and the source code modification.

References

[ACW+07] Y. Agarwal, R. Chandra, A. Wolman, P. Bahl, K. Chin, and R. Gupta. Wireless wakeups revisited: energy management for VoIP over WiFi smartphones. In *Proceedings of the 5th international conference on Mobile systems, applications and services*, MobiSys '07, pages 179–191, New York, NY, USA, 2007. ACM.

[AKK04] J.N. Al-Karaki and A.E. Kamal. Routing techniques in wireless sensor networks: a survey. *Wireless Communications, IEEE*, 11(6):6–28, 2004.

[AWSC02] I.F. Akyildiz, S. Weilian, Y. Sankarasubramaniam, and E. Cayirci. A survey on sensor networks. *Communications Magazine, IEEE*, 40(8):102 – 114, August 2002.

[BPPS10] S. Basagni, C. Petrioli, R. Petroccia, and M. Stojanovic. Choosing the packet size in multi-hop underwater networks. In *OCEANS 2010 IEEE - Sydney*, pages 1–9, 2010.

[BPPS12] S. Basagni, C. Petrioli, R. Petroccia, and M. Stojanovic. Optimized packet size selection in underwater wireless sensor network communications. *Oceanic Engineering, IEEE Journal of*, 37(3):321–337, 2012.

[CG06] S. Choudhury and J.D. Gibson. Payload length and rate adaptation for throughput optimization in wireless LANs. In *Vehicular Technology Conference, 2006. VTC 2006-Spring. IEEE 63rd*, volume 5, pages 2444 –2448, may 2006.

[CHXHH09] Z. Chenyuan, Y. Hui, W. Xinbing, and C. Hsiao-Hwa. Improvement of capacity and energy saving of VoIP over IEEE 802.11 WLANs by a dynamic sleep strategy. In *Global Telecommunications Conference, 2009. GLOBECOM 2009. IEEE*, pages 1 –5, 30 2009-dec. 4 2009.

[CLL05] Y.P. Chen, A.L. Liestman, and Jiangchuan Liu. Energy-efficient data aggregation hierarchy for wireless sensor networks. In *Quality of Service in Heterogeneous Wired/Wireless Networks, 2005. Second International Conference on*, pages 7 – 7, August 2005.

[CM99] S. Corson and J. Macker. Mobile Ad hoc Networking (MANET): routing protocol performance issues and evaluation considerations, 1999.

[Com06] Cisco Technical Support Community. RTP Header-Compression or Compressed RTP (cRTP). `http://www.cisco.com/en/US/tech/tk652/tk698/technologies_tech_note09186a0080094ae2.shtml`, Feb 2006.

[CSE04] Y. Chen, N. Smavatkul, and S. Emeott. Power management for VoIP over IEEE 802.11 WLAN. In *Wireless Communications and Networking Conference, 2004. WCNC. 2004 IEEE*, volume 3, pages 1648 – 1653 Vol.3, march 2004.

[Dom11] M.C. Domingo. Packet size optimization for improving the energy efficiency in body sensor networks. In *ETRI Journal*, volume 33, pages 299–309, June 2011.

[DSSY07] W. Dalei, C. Song, H. Sharif, and Y. Yang. Packet size optimization for goodput enhancement of multi-rate wireless networks. In *Wireless Communications and Networking Conference, 2007.WCNC 2007. IEEE*, pages 3575–3580, 2007.

[FK11] T. Frantti and M. Koivula. Fuzzy packet size control for delay sensitive traffic in ad hoc networks. *Expert Syst. Appl.*, 38:10188–10198, August 2011.

[FZ08] G. Fettweis and E. Zimmermann. ICT energy consumption-trends and challenges. *Communications*, (Wpmc 2008):2006–2009, 2008.

[GPSN06] B. Gleeson, D. Picovici, R. Skehill, and J. Nelson. Exploring power saving in 802.11 VoIP wireless links. In *Proceedings of the 2006 international conference on Wireless communications and mobile computing*, IWCMC '06, pages 779–784, New York, NY, USA, 2006. ACM.

[HA06] M. Hassan and D.F. Alekseevich. Variable packet size of IP packets for VoIP transmission. In *Proceedings of the 24th IASTED international conference on Internet and multimedia systems and applications*, pages 136–141, Anaheim, CA, USA, 2006. ACTA Press.

[HCB00] W.R. Heinzelman, A. Chandrakasan, and H. Balakrishnan. Energy-efficient communication protocol for wireless microsensor networks. In *System Sciences, 2000. Proceedings of the 33rd Annual Hawaii International Conference on*, pages 10 pp. vol.2–, 2000.

[HSI06] S.L. Howard, C. Schlegel, and K. Iniewski. Error control coding in low-power wireless sensor networks: when is ECC energy-efficient? *EURASIP J. Wirel. Commun. Netw.*, 2006:29–29, April 2006.

[HVPD09] W.V. Heddeghem, W. Vereecken, M. Pickavet, and P. Demeester. Energy in ICT - trends and research directions. In *Advanced Networks and Telecommunication Systems (ANTS), 2009 IEEE 3rd International Symposium on*, pages 1–3, 2009.

[IEE99] Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications. IEEE Standard 802.11, June 1999.

[JA11a] L.T. Jung and A. Abdullah. Underwater acoustic communications: Optimizing data packet size with respect to throughput efficiency, BER, and energy efficiency. In *Proceedings of International Conference on Computer Communication and Management (ICCCM 2011)*, 2011.

[JA11b] L.T. Jung and A.B. Abdullah. Underwater acoustic communications: Relationship between data packet size, throughput, BER, and distance. In *AIP Conference Proceedings*, volume 1337, page 155, 2011.

[JGS02] L. JangYeon, K. GyeYoung, and P. SungKwon. Optimum UDP packet sizes in ad hoc networks. In *High Performance Switching and Routing, 2002. Merging Optical and IP Technologies. Workshop on*, pages 214 – 218, 2002.

[Joh04] K. John. An analysis of energy-efficient Voice over IP communication in wireless networks, March 2004.

[Jon05] L. E. Jonsson. RObust Header Compression (ROHC): Requirements on TCP/IP Header Compression. RFC 4163 (Informational), August 2005.

[JS02] W. Jiang and H. Schulzrinne. Comparison and optimization of packet loss repair methods on VoIP perceived quality under bursty loss. In *Proceedings of the 12th international workshop on Network and operating systems* *support for digital audio and video*, NOSSDAV '02, pages 73–81, New York, NY, USA, 2002.

[JXA04] Y. Jun, W. Xiaodong, and D.P. Agrawal. Optimal packet size in error-prone channel for IEEE 802.11 distributed coordination function. In *Wireless Communications and Networking Conference, 2004. WCNC. 2004 IEEE*, volume 3, pages 1654 – 1659 Vol.3, March 2004.

[KDR+12] G. Karina, B. Dejene, R. Roberto, R. Tinku, M. Daniele, and G. Fabrizio. Measurement-based modelling of power consumption at wireless access network gateways. *Computer Networks*, 56(10):2506 – 2521, 2012. Green communication networks.

[KHZ11] M.N. Krishnan, E. Haghani, and A. Zakhor. Packet length adaptation in WLANs with hidden nodes and Time-Varying channels. In *Global Telecommunications Conference (GLOBECOM 2011), 2011 IEEE*, pages 1 –6, dec. 2011.

[KRJ08] D. Kumar, Y. Ryu, and H. Jang. Quality of service (QoS) of voice over MAC protocol 802.11 using ns-2. In *Proceeding of the 1st ACM international workshop on Communicability design and evaluation in cultural and ecological multimedia system*, CommunicabilityMS '08, pages 39–44, New York, NY, USA, 2008. ACM.

[KS06] Z.H. Kashani and M. Shiva. Bch coding and multi-hop communication in wireless sensor networks. In *Wireless and Optical Communications Networks, 2006 IFIP International Conference on*, pages 5 pp. –5, 2006.

[KSPR04] H. Karvonen, Z. Shelby, and C. Pomalaza-Raez. Coding for energy efficient wireless embedded networks. In *Wireless Ad-Hoc Networks, 2004 International Workshop on*, pages 300 – 304, June 2004.

[LDL10] W. Lifu, Z. Dasheng, and M. Lei. An energy efficient WLAN Skype deployment using GSM wakeup signals. In *Green Computing and Communications (GreenCom), 2010 IEEE/ACM Int'l Conference on Int'l Conference on Cyber, Physical and Social Computing (CPSCom)*, pages 470 –473, dec. 2010.

[LMIS12] K. Lendvai, A. Milankovich, S. Imre, and S. Szabo. Optimized packet size for energy efficient delay-tolerant sensor networks. In *Wireless and Mobile Computing, Networking and Communications (WiMob), 2012 IEEE 8th International Conference on*, pages 19–25, 2012.

[LPV+08] C. Lubritto, A. Petraglia, C. Vetromile, F. Caterina, A. D"Onofrio, M. Logorelli, G. Marsico, and S. Curcuruto. Telecommunication power systems: Energy saving, renewable sources and environmental monitoring. In *Telecommunications Energy Conference, 2008. INTELEC 2008. IEEE 30th International*, pages 1–4, 2008.

[LQK+12] Y. Li, X. Qi, M. Keally, Z. Ren, G. ZHOU, D. Xiao, and S. Deng. Communication energy modeling and optimization through joint packet size analysis of BSN and WiFi Networks, 2012.

[LS98] P. Lettieri and M.B. Srivastava. Adaptive frame length control for improving wireless link throughput, range, and energy efficiency. In *INFOCOM '98. Seventeenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, volume 2, pages 564–571 vol.2, 1998.

[MFAJC13] M. Martnez, J. Friginal, D. Andres, and R. Juan-Carlos. Open challenges in the resilience evaluation of Ad Hoc networks. In *Dependable Computing*, pages 194–197. Springer, 2013.

[MT09] E.S. Myakotnykh and R.A. Thompson. Effect of packet size and compression variation on quality of VoIP communications. In *Security and Communication Networks (IWSCN), 2009 Proceedings of the 1st International Workshop on*, pages 1 –6, may 2009.

[NG10] V. Namboodiri and L. Gao. Energy-efficient VoIP over wireless LANs. *IEEE Transactions on Mobile Computing*, 9(4):566–581, April 2010.

[Obe08] S.A. Obeidat. *Cross-layer opportunistic adaptation for voice communications over wireless ad hoc networks.* PhD thesis, Tempe, AZ, USA, 2008.

[Par11] V. Paruchuri. Greener ICT: Feasibility of successful technologies from energy sector. In *Advanced Communication Technology (ICACT), 2011 13th International Conference on*, pages 1398–1403, 2011.

[PFSK08] G.P. Perrucci, F.H.P Fitzek, G. Sasso, and M. Katz. Energy Saving Strategies for Mobile Devices using Wake-up Signals. In *4th International Mobile Multimedia Communications Conference(MobiMedia 2008)*, Oulu, Finland, July 2008. ICTS/ACM.

[PP00] C.E. Perkins and C. Perkins. *Ad Hoc Networking*. Addison-Wesley Professional, December 2000.

[PVD+08] M. Pickavet, W. Vereecken, S. Demeyer, P. Audenaert, B. Vermeulen, C. Develder, D. Colle, B. Dhoedt, and P. Demeester. Worldwide energy needs for ICT: the rise of power-aware networking. In *ANTS: 2008 2ND INTERNATIONAL SYMPOSIUM ON ADVANCED NETWORKS AND TELECOMMUNICATION SYSTEMS*, pages 1–3. IEEE, 2008.

[SAM03] Y. Sankarasubramaniam, I.F. Akyildiz, and S.W. McLaughlin. Energy efficiency based packet size optimization in wireless sensor networks. In *Sensor Network Protocols and Applications, 2003. Proceedings of the First IEEE. 2003 IEEE International Workshop on*, pages 1 – 8, 2003.

[San12] P. Santi. *Mobility Models for Next Generation Wireless Networks: Ad Hoc, Vehicular and Mesh Networks*. Wiley. com, 2012.

[SBYG11] J. Sung-Bong and K. Young-Gab. An energy-efficient delay reduction technique for supporting WLAN-based VoIP in smartphone. *Journal of Systems Architecture*, 57(10):934–944, 2011.

[Sch10] H.O. Scheck. ICT & wireless networks and their impact on global warming. In *Wireless Conference (EW), 2010 European*, pages 911–915, 2010.

[SSHI+01] E. Shih, C. Seong-Hwan, N. Ickes, R. Min, A. Sinha, A. Wang, and A. Chandrakasan. Physical layer driven protocol and algorithm design for energy-efficient wireless sensor networks. In *MobiCom '01: Proceedings of the 7th annual international conference on Mobile computing and networking*, pages 272–287, New York, NY, USA, 2001. ACM.

[TDV08] N. Tsiftes, A. Dunkels, and T. Voigt. Efficient sensor network reprogramming through compression of executable modules. In *Sensor, Mesh and Ad Hoc Communications and Networks, 2008. SECON '08. 5th Annual IEEE Communications Society Conference on*, pages 359 –367, june 2008.

[VA08] M.C. Vuran and I.F. Akyildiz. Cross-layer packet size optimization for wireless terrestrial, underwater, and underground sensor networks. In *INFOCOM 2008. The 27th Conference on Computer Communications. IEEE*, pages 226–230, 2008.

[VVHC+10] W. Vereecken, W. Van Heddeghem, D. Colle, M. Pickavet, and P. Demeester. Overall ICT footprint and green communication technologies. In *Proceedings of the 4th International symposium on Communications, Control and Signal Processing (ISCCSP 2010)*, page 6. IEEE, 2010.

[Web08] M. Webb. *SMART 2020: enabling the low carbon economy in the information age, a report by The Climate Group on behalf of the Global eSustainability Initiative (GeSI)*. 2008.

[WH02] J. Wenyu and S. Henning. Comparisons of FEC and CODEC robustness on VoIP quality and bandwidth efficiency. In *PROC. OF THE ICN CONF*, pages 1–12, 2002.

[WHE02] Y. Wei, J. Heidemann, and D. Estrin. An energy-efficient MAC protocol for wireless sensor networks. In *INFOCOM 2002. Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, volume 3, pages 1567 – 1576 vol.3, 2002.

[WhZhYp07] H. Wen-hua, G. Zhi-hui, and H. Yu-ping. Optimizing UDP packet sizes in ad hoc networks. In *Wireless Communications, Networking and Mobile Computing, 2007. WiCom 2007. International Conference on*, pages 1617 –1619, September 2007.

[Wir07] Approved draft amendment to IEEE standard for Information technology-Telecommunications and information exchange between systems-PART 15.4:Wireless Medium Access Control (MAC) and Physical Layer (PHY) specifications for Low-Rate Wireless Personal Area Networks (LR-WPANs): Amendment to add alternate PHY (Amendment of IEEE Std 802.15.4).

IEEE Approved Std P802.15.4a/D7, Jan 2007, pages –, 2007.

[WWWL10] L. Wang, M. Wu, L. Wei, and M. Li. An adaptive forward error control method for voice communication. In *Networking and Digital Society (ICNDS), 2010 2nd International Conference on*, volume 2, pages 186 –189, may 2010.

[WXC+10] D. Wei, L. Xue, C. Chun, H. Yuan, C. Gong, L. Yunhao, and B. Jiajun. DPLC: Dynamic Packet Length Control in wireless sensor networks. In *INFOCOM, 2010 Proceedings IEEE*, pages 1 –9, march 2010.

[XJA05] W. Xiaodong, Y. Jun, and D.P. Agrawal. Effects of contention window and packet size on the energy efficiency of wireless local area network. In *Wireless Communications and Networking Conference, 2005 IEEE*, volume 1, pages 94–99 Vol. 1, 2005.

[YF09] Z. Yan and S. Feng. Packet size optimization for goodput and energy efficiency enhancement in slotted IEEE 802.15.4 networks. In *Wireless Communications and Networking Conference, 2009. WCNC 2009. IEEE*, pages 1–6, 2009.

[YK10] N. Yaakob and I. Khalil. Packet size optimization for congestion control in pervasive healthcare monitoring. In *Information Technology and Applications in Biomedicine (ITAB), 2010 10th IEEE International Conference on*, pages 1–4, 2010.

Chapter 2

Packet Structure, Error Correction and Compression

This chapter illustrates the packet structure of most common Wireless Local Area Network (WLAN) Wi-Fi network. As a real time application communication scenario VoIP packet format over WiFi is shown. Since wireless channel is noisy. Packet error correction is illustrated for noisy channel. We found that error correction incorporate extra overhead on packet. Hence, packet compression is investigated over WLAN to reduce packet overhead.

2.1 Wireless Local Area Network (WLAN)

WLAN is a network used by small portable devices such as PDA, laptop, smart phones etc. WLAN can also be used for wireless sensor networks [ASSC02]. Whether the WLAN is accessed using powerful laptop or a resource constrained sensor node the packet structure and protocol mechanism is the same for specific standard. The IEEE 802.11 family are the standard specifications for wireless LANs which is mainly defined in MAC and PHY layers [mac12].

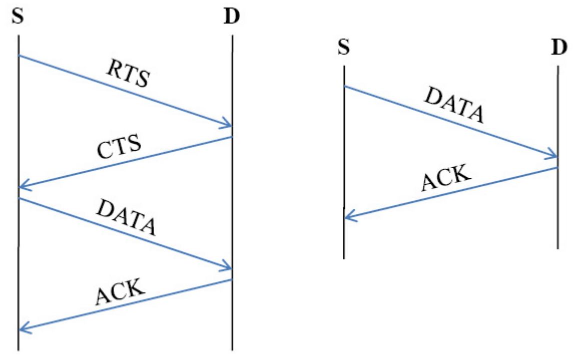

Figure 2.1: DCF four way handshaking and two way handshaking policy

For the PHY layer, IEEE 802.11 defines the encoding schemes for data transmission in the wireless medium. The most common signal to data encoding schemes are Frequency Hopping Spread Spectrum (FHSS), Direct Sequence Spread Spectrum (DSSS), and Orthogonal Frequency Division Multiplexing (OFDM). The FHSS transmission scheme operates on the frequency range of 2.4 to 2.5 GHz and bit rate $2Mbps$. There are some other standards 802.11b, 802.11a, and 802.11g which supports higher bit rates. Among these 802.11b uses DSSS to provide higher bit rates up to $11Mbps$ and also enable to provide $5.5Mbps$, $2Mbps$ and $1Mbps$ according to the application requirements. Basically 802.11b is the standard used for Wi-Fi which is used in laptops, smart phones etc.

2.1.1 Network Management and Control Packets of WLAN

The WLAN 802.11 protocol is called distributed coordination function (DCF) which is a random channel access scheme [Bia00]. The DCF is based on the carrier sense multiple access with collision avoidance (CSMA/CA) protocol. To avoid the frame collision between the contending nodes it provides a backoff rules. If more than one source nodes transmit their packets at the same time, collision happens between the frames and the packets are drop/lost. To reduce the collision probability the nodes restrain their transmission according to random binary exponential backoff rule. DCF employs two methods to access the channel and frame delivery (see figure 2.1).

Table 2-A: RTS packet format of 802.11 protocol

Frame Control	Duration	Receiver Address	Transmitter Address	FCS
2	2	6	6	4

- Two-way handshaking (basic scheme)
- Four way handshaking (RTS/CTS scheme)

In the basic scheme, data packet from the source node is preceded by an ACK packet by the destination node. Since in a wireless medium the sender cannot determine the successful reception of the transmitted data ACK packet ensures the confirmation of a successful reception. In four way handshaking method the Request To Send (RTS) packet is broadcasted by the source (if the channels remains free or idle for a specific time period). Then the destination node replies back by a Clear To Send (CTS) packet. After that the data packet is transmitted and a confirmation Acknowledgment (ACK) packet is sent by the destination node. Through this way a successful transmission is preceded. During one whole transmission period (RTS, CTS, DATA and ACK) the neighbour nodes remains quiet and suppress their transmission for the entire period. The RTS, CTS and ACK frame structures are shown in table 2-A and 2-B. Although the CTS and the ACK frames are completely identical the value of subtype field in frame control is different 1100 and 1101 respectively. The data frame structure is shown in section 2.1.2.

2.1.2 WLAN Configuration

WLAN can be configured in two ways

Table 2-B: CTS and ACK packet format of 802.11 protocol

Frame Control	Duration	Receiver Address	FCS
2	2	6	4

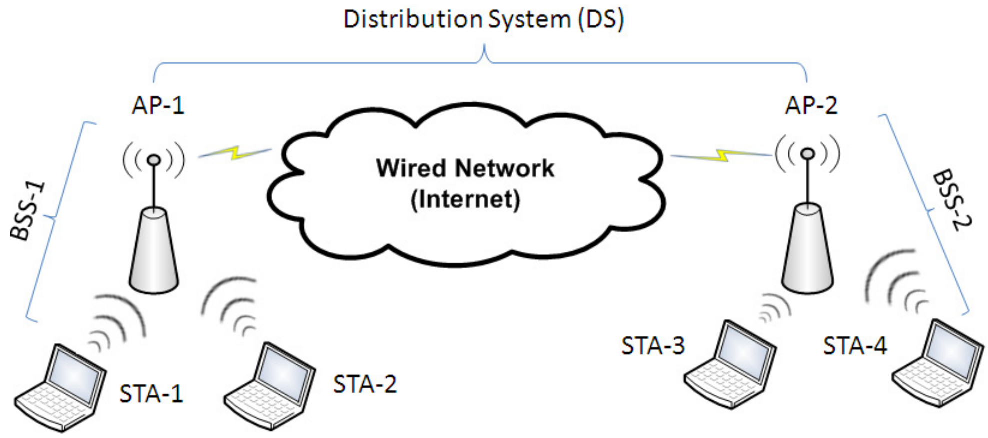

Figure 2.2: WLAN infrastructure network with different components

- Infrastructure less independent mode
- Infrastructure mode

The independent mode is called Ad hoc or peer to peer (P2P) network. The wireless nodes must be configured explicitly to use ad hoc mode. In this mode each node can directly communicate with all other nodes within its transmission range and create a mesh network [WPW06] for communication. This type of network is useful to connect wireless nodes, if there is no wireless Access Point (AP) available. The wireless nodes act as an AP and takes some of the responsibilities of the AP such as: periodic beaconing which announce the presence of WLAN, authentication of new nodes within the network etc. The nodes do not act as a bridge to forward information to its neighbour nodes. Maximum nine nodes can communicate among themselves in an ad hoc 802.11 WLAN mode [Tea03].

In infrastructure mode, there is at least one wireless AP and one client node. The client node uses the AP to access the resources of a traditional wired network (Internet). That's why whenever the node wants to communicate with other node, packets would be sent to the AP first which will be forwarded to the destination node. Figure 2.2 represents the WLAN with infrastructure configuration mode.

Table 2-C: IEEE 802.11 WLAN (Ethernet) Frame Format for Ad hoc Network

Frame Control	Duration ID	Address 1 (RA)	Address 2 (TA)	Address 3 (AP)	Sequence Number	Address 4 (N/A)	Pay load	FCS
2	2	6	6	6	2	6	0-2312	4

2.1.3 WLAN Packet Format

The WLAN frame contains all the necessary information within the frame for its delivery to the destination. The frames consists of several field that contain the required data. The 802.11 MAC frame format for the WLAN is given in table 2-C.

The frame [Tea03, com12] starts with a control field which is the most important part of the MAC data frame. It contains information about the protocol version, type of the frames for an instance whether its a data frame, network management frames or control frame, determines the retry limit for a corrupted or lost frame, determines whether the transmitting node is in active mode or power save mode, maintain orders of the frames, indicates if there is more fragments of any frames etc. Moreover, frame control field also indicates whether the frame is for the distributed system (DS) or from the DS. It also indicates if any encryption or extra authentication is used for a particular frame.

The Duration/ID Field is used to indicates the remaining duration needed to receive the next transmission. The Sequence Control field determines the fragment number of frames and the sequence number of the frames. FCS (Frame Check Sequence) is used to check the error fields within the frame.

There are four address fields within the MAC frame that represents four MAC address fields: Destination Address (DA), Source Address (SA), Receiver Address (RA) and Transmitter Address (TA).

The transmitted frames can carry different types of application data inside. The data can be for the real time applications or for the delay tolerant sensor networks [ASSC02].

According to the applications data structure on the upper layer can be changed but it does not have impact on the lower layers format. Only the size varies according to the application data. For the smaller data units in the application layer the overhead percentage for the whole frame is high. For a bigger data packets the overhead is lower. Whenever the data packets sizes are higher than the maximum transfer limit the packet is split into two parts and become multiple frames. Usually real time applications uses smaller packet sizes for faster delivery. For an example VoIP uses small packet sizes for data transmission. In the next section the packet structure of a real time application VoIP is described in detail.

2.2 VoIP Protocol Stack and WLAN

VoIP [GK03] is a set of protocols defined in different layers that allow the transmission of voice via packet switched networks [KBS$^+$98]. Voice communication is carried out using the Internet Protocol (IP) and allows using the network as the transmission medium for telephone calls [DPG00, Goo02]. But unlike with the traditional Public Switched Telephone Network (PSTN) it sends voice data in packets using IP rather than circuit transmissions [DKJ02]. In an IP network (Packet Switching environment), voice data is digitized and bundled into packets. Then the packets are sent across the network. Traditional phone networks (PSTN) uses circuit-switching in which resources are reserved along the entire communication channel for the duration of the call whereas in IP packet-switching there is no dedicated channel. The packets know their destination, and may arrive there via different paths.

VoIP communication needs an audio input device (microphone), like as ordinary PSTN system, to send the audio signal. An analog-to-digital (ADC) converter is used to transform audio signal into digital bytes. Then the packetizer allocates chunk of bytes (payload) for each packet. After that the payload is added with the headers to make a complete IP packet. VoIP service can run by using wired or wireless network. In table 2-D the VoIP protocol stack for the wireless network is illustrated.

2.2.1 Encoding and Decoding of VoIP

Voice is analogue, whereas the data network is digital. In order to overcome the problems associated with analog-digital conversion engineers have developed CODECs (an algorithm) which are able to translate signals and convert them from one category to another. The process to sample analogical waves into digital information is made by an encoder-decoder in short CODEC [KP09]. In addition, the CODEC can compress the digital data. The compression of the waveform saves bandwidth that enable to provide more VoIP connections at the same time. Also CODEC does the silence suppression, not to send any packet when there is not any voice during the conversation and saves bandwidth and other system resources.

The attributes of some well known CODEC is given in table 2-E [Com11, CXSM06, KP09]. In this table 2-E the payload size and bit rate is given as a standard format of VoIP codec. The voice payload size must be a multiple of the CODEC sample size. For example, G.729 packets can use 10, 20 ... 60 bytes of voice payload size. G.711 CODEC generates 80 bytes of sample payload for 10ms voice, such as: 80, 160, 240 bytes of voice payloads with an interval of 80 bytes.

Table 2-D: VoIP Protocol Stack for the wireless local area network

Application Layer	Voice			
	RTP	RTCP	SIP	H.323
Transport Layer	UDP			
Network Layer	IP			
Link Layer	802.11 MAC			
Physical Layer	Ethernet			

Table 2-E: Attributes of some common CODECs [Com11]

Codec Name	Sample Interval (ms)	Sample Size (Bytes)	Voice Payload Size (ms)	Voice Payload Size (Bytes)	Bit Rate (Kbps)	PPS	Bandwidth Ethernet (Kbps)
G.711	10	80	20	160	64	50	87.2
G.729	10	10	20	20	8	50	31.2
G.723.1	30	24	30	24	6.3	33.3	21.9
G.723.1	30	20	30	20	5.3	33.3	20.8
G.726	5	20	20	80	32	50	55.2
G.726	5	15	20	60	24	50	47.2
G.728	5	10	30	60	16	33.3	31.5

Packet Per Second (PPS) = Codec Bit Rate / Voice Payload Size,
Minimum bandwidth = Packet size (including headers) × PPS.

2.2.2 Wireless LAN VoIP

Wireless VoIP [SA07, KP09] applications and services can run over wireless local area network (WLAN). The MAC layer (section 2.1) mainly defines the type of network whether the packet is going to be transmitted by the wired or wireless network. VoIP packet remains inside the data (payload) portion of the frame. The whole packet format for the VoIP WLAN is given more explicitly in figure 2.3 [Gas05].

The frame starts with preamble which is the header for the physical layer and then followed by MAC header fields (see table 2-C). The middle section of the frame consists of payload data from the upper layers (e.g. Network layer, Application layer) and after that it ends with the FCS fields. At the physical level, the frame is transmitted as a series of signal representing the bits. On the destination device, the physical layer reassembles the bits into a data frame.

In figure 2.3 the preamble is used for the physical layer header. Physical layer header is called PLCP (Physical Layer Convergence Protocol) header which is 24 bytes. PHY header start with a preamble followed by PLCP header and then the payload from the upper layers. Preamble alerts the receiver nodes whether there is any potential receivable signal in the channel or not. Therefore it determines whether the channel is free to

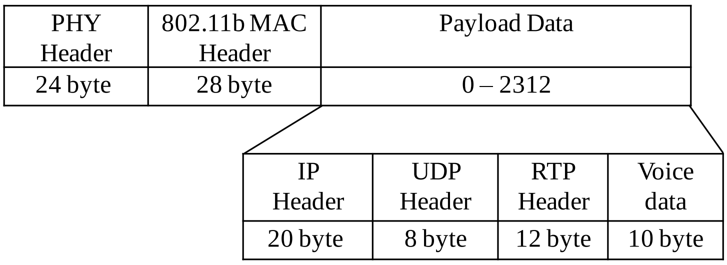

Figure 2.3: VoIP packet format in a wireless packet frame after adding the headers and trailers from different layers

transmit frame or not. It also defines the modulation type, transmission frequency and the remaining time length of the channel to be busy. The PHY headers are transmitted in lower data rate to increase reliability and synchronization with all other nodes.

From figure 2.3 it is revealed that WLAN VoIP carries huge overhead during the transmission. To transmit a single voice of $10ms$ considering all the headers and trailer the overhead is 92 bytes (24 bytes PHY, 28 bytes MAC, 20 bytes IP, 8 bytes UDP and 12 bytes RTP) which is 90% overhead.

2.3 Error Correction Coding for Wireless Network

In wireless data communication system, data is transmitted through the ether. Since, there is no direct link between the transmitter and receiver, unwanted noise from other wireless devices, obstacles between the line of sight can disrupt the communication. Therefore, wireless channels are usually more error prone comparing to wired network. When data packets are transmitted over the wireless channel, it may experience many impairments, such as fading, interference, random noise etc. Due to these impairments, data bits of the transmitted packets are corrupted. Sometimes the corrupted packets are lost in the medium and sometimes reach the destination with error bits. Whenever any corrupted packet is received at the receiver end, the node can check the packet condition by using Cyclic Redundancy Check (CRC) codes (see chapter 2 table 2-A). The nodes can detect the error and identify the corrupted packets. Either the corrupted packets are discarded or recovered at the destination.

To recover lost packets or correct the error packet information there exist the following two distinct approaches [TYL08, PSF10]:

- Automatic Repeat Request (ARQ) [HSI06]: In this protocol whenever the receiver node detect an error by the CRC code, it drops the error packet and asks a re-transmission from the source by sending a negative acknowledgment packet. If the negative acknowledgment packet is also lost in the medium, after certain timeout limit, the source node retransmits the packet to the receiver. This process continues until the maximum number of retransmissions limit is reached. Through this way ARQ increases the reliability of packet delivery. But, if the channel condition remains poor, error rate becomes higher and large number of retransmissions may be required. Therefore, it increases the end to end latency for packet transmission. As a result, ARQ protocol become incompatible for the delay sensitive real-time applications (such as: VoIP, multimedia etc). Since, the destination node sends extra acknowledgment packets, this whole procedure consume extra resources.

- Forward Error Correction (FEC): To retrieve the lost information in a packet or recover the corrupted bits within a packet, the most common scheme in ad hoc network is forward error correcting codes [LC04, KBP07, WWWL10]. Error correcting codes can recover the lost packet in a bit-exact form. But it contains redundant bits or parity bits to recover the lost/error packet. Hence, the encoded packet become larger than the original packet size. The encoded packet is generated from the sample payload by using predefined error correction algorithm. After adding the error correction code the packet become larger than it's normal size. As a result to transmit the packet with FEC code and process the packet for correcting errors some extra delay is introduced with the packet transmission time. Also extra energy is needed for the encoding-decoding and for the transmission of parity bits.

2.4 Error Correcting Codes

There are several types of error correcting codes [LC04, PSF10]. The first major classification is linear vs. nonlinear codes. Linear codes are categorized in block codes and convolution codes [HAM50]. Convolutional code operates on streams of data bits. It processes the encoded message in a bit by bit order. Convolutional codes are useful and efficient error correction techniques [PSF10, SAM03]. A convolutional code is implemented at the hardware level, uses a set of shift registers and adders. The decoder requires highly complex hardware according to the data correction capability [LC04]. Since, software level implementation is more flexible and comparatively easy to install, convolutional techniques are not considered for further analysis in this research.

Block codes [LC04, PSF10] are different from the convolutional codes. They break the information into chunks and append redundant bits that are used to detect and correct errors. The data and check bits together are called a codeword or block. The payload data from the application layer is encoded into multiple discrete blocks and processed block by block. Compared to convolutional codes block codes are simple to implement in software level. Therefore, the block coding technique is widely used in wireless communication environments. Most popular block codes are RS and BCH codes.

2.4.1 BCH Error Correction Codes

BCH (Bose Chaudhuri - Hocquenghem) code [LMZG97] was invented in 1959 by Hocquenghem, and independently in 1960 by Bose and Ray-Chaudhuri. BCH code is a powerful random error correcting codes for multiple error correction. This code can handle randomly located errors in a data streams according to its inherent limitations. BCH decoders correct up to a certain number of errors, specified by the user [LMZG97]. For any block length n and positive integers ($m \geq 3$), binary BCH code is referred as (n, k, t) code with the following parameters [Mas65, McE84]:

- Total number of bits to be protected: $n = 2^m - 1$
- Number of error checking (parity) bits: $(n - k) = \tau \geq mt$,
- Size of information bits: $k = (n - \tau) = (\ell + h)$
- Maximum error correction strength: $t < 2^{m-1}$

where ℓ is the payload and h is the size of the packet header. BCH code (n, k, t) can correct up to t-bit errors, and thus it is also referred to as a t error correcting code. From the block length constraints it is stated that the length of the block needs to be 7, 15, 31, and 63. The number t of reliably correctable bits in a block of length n bits depends on the actual coding scheme.

A similar type of error correcting code is Reed Solomon (RS) code [McE84] which is well known for correcting burst error of data communication channel. BCH code uses bits as symbols whereas the RS code consider eight bits (one byte) as one symbol to correct an error [GR05, Wic94]. The RS code is comparatively more complex to design and implement. When the channel is too noisy and the number of error bits becomes large at that time BCH code performs better than RS code by using less number of parity (overhead) bits. Moreover it is showed in some papers that BCH code performs better and is more energy efficient than the RS code [LPK13, BYJK07, VA09].

2.4.2 Error correction coding and packet error rate

Whenever encoded packets are transmitted through the channel, it may corrupt due to the channel noise. Some bits become lost or erroneous. Lets assume the average bit error rates of any noisy channel is b and due to the bit error rates the average packet error rates p, then after using the error correction code the new packet error rate (p_{new})

will be [SAM03, KSPR04, PSF10]

$$p_{new} = 1 - \sum_{i=0}^{t} \begin{pmatrix} n \\ i \end{pmatrix} b^i (1-b)^{L-i} \tag{2.1}$$

where $\begin{pmatrix} n \\ i \end{pmatrix}$ is a orderless combination of n and i which can be represented as

$$\frac{n!}{i!(n-i)!} \tag{2.2}$$

where n is the number of total bits within the encoded packets and all the other variables are according to section 2.4.1.

2.5 Cost of Error Correction Coding

Error correction code can correct a certain number of errors. Hence, it reduces packet loss rate, increases the channel reliability and improves performance. But error correction coding comes with price. The price are the cost of

- Encoding: energy cost to generate the encoded message (payload)
- Decoding: processing of the encoded message
- Parity bits: cost of transmitting redundant bits
- Delay: additional transmission time for the extra parity bits and the encoding-decoding

The FEC decoding algorithm is typically a more complex operation than encoding. Several researches [SAM03, KSPR04, GS99] proved that encoding energy needs very low amount of energy [VA09] compare to decoding, and therefore can be ignored. Mainly the major cost of the FEC is decoding and the additional latency incurred for the parity bits. The energy efficiency of three well known error correction codes (BCH, RS and Convolutional) has been studied in [BYJK07, SAM03]. This analysis has revealed that BCH code outperforms the other error correcting codes due to its low encoding-decoding energy cost. For this reason in the next section the energy consumption is presented only for the BCH codes.

2.5.1 Energy Consumption of BCH Code

Ad hoc network is used for various applications and also can be formed with various types of equipments such as smart phone, laptop, sensor nodes etc. Hence, Power consumption for specific application or equipment could be higher in some equipments and lower for others. As a reference point we have used the Micaz power estimations for the BCH FEC [SP10a, ZDQ08, VA09, NRKS12]. This power consumption model has been used in many papers and accepted by many other researchers.

The energy consumption of error correction code is dependent upon the error correction strength (t) and the length of the block size (n) [VA09, SP10b, NK11]. When the strength of the BCH code is increased the complexity of the code increases exponentially [SP10a, ZDQ08]. Consequently the energy cost and latency for the code also grows exponentially. The decoding latency (T_{dec}) for the BCH code is given by [SP10a, ZDQ08, NRKS12]

$$T_{dec} = (2nt + 2t^2)(T_{add} + T_{mul}) \tag{2.3}$$

where T_{add} and T_{mul} are the latencies for addition and multiplication.

A 8-bit micro-controller unit of ATmega128 [Sup12] microprocessor of MicaZ or Mica sensor node [NRKS12] can perform addition and multiplication of 8 bits in one and two cycles [SP10a, ZDQ08, VA09, NRKS12]. Hence,

$$T_{add} = \left\lceil \frac{m}{8} \right\rceil t_{cycle}, \quad T_{mul} = 2 \left\lceil \frac{m}{8} \right\rceil t_{cycle} \tag{2.4}$$

Therefore, for a low power 8-bit micro-controller for one cycle it is computed in [SP10a, ZDQ08, VA09, NRKS12] as

$$T_{dec} = (2nt + 2t^2)3\left\lceil \frac{m}{8} \right\rceil t_{cycle} \tag{2.5}$$

where $m = \lfloor \log_2 n + 1 \rfloor$ for error correction (BCH) code [SAM03], t_{cycle} is one cycle duration of the microprocessor. According to the data sheet of the MicaZ processor [Sup12] t_{cycle} is approximately $250ns$ [VA09, NRKS12]. From the latency of the error correction code the decoding energy consumption for the BCH code is expressed as [SP10a, ZDQ08, VA09]

$$E_{dec} = I_{proc}V_{proc}T_{dec} \tag{2.6}$$

where I_{proc} and V_{proc} are the required current and voltage for the processor to decode of n bits. It is assumed that the execution of each instruction consumes approximately the same amount of voltage and current [ZDQ08, VA09, NRKS12, TYL08]. Hence, the delay increases according to the error correction strength and the packet size.

To keep the complexity low the decoding energy (E_{dec}) for the BCH code is outlined in [SP10a, ZDQ08, GS99] as follows:

$$\begin{aligned} E_{enc} &= 2t(n - 2t)(E_{add} + E_{mul}) \\ E_{dec} &= (2nt + 2t^2)(E_{add} + E_{mul}) \end{aligned} \tag{2.7}$$

Where, E_{add} and E_{mul} is the energy consumption for addition and multiplication to process the decoding. It is computed in [MV02] that to complete an addition (E_{add}) and multiplication (E_{mul}) instruction a Strong ARM 100 processor takes $12.9\,ns$ and $20.8\,ns$ and spent $0.253\,W$ and $0.291\,W$ respectively. That yield

$$\begin{aligned} E_{add} &= 3.26 \times 10^{-9} J \\ E_{mul} &= 6.05 \times 10^{-9} J \end{aligned} \tag{2.8}$$

This equation (2.7) will not provide the accurate estimation based on current, power or latency but the complexity will be less. In this thesis the rest of the chapters for the decoding energy is estimated from equation (2.7).

2.6 Packet Compression

Compression methods can reduce the transmitted data. Usually the energy cost of transmitting one bit is higher than a single bit processing in an ad hoc node. One bit communication energy cost is approximately 485-1267 add instruction of energy [BA06]. Hence, if the compression reduces the total transmitted bits, then the network would be more energy efficient. Based on this strategy compression has been used to improve energy efficiency [SGGN09, TDV08, SM06, YLL^{+}09, YLYW10, KL05, NMQN08]. In this chapter the pros and cons of the compression are investigated and the impact on the network is discussed.

2.7 Compression Method

Compression is an algorithm that transform the original packet (bytes) into smaller size packet. Whenever the compressed packet is reached at the receiver end, the receiver node reverses the process, reverting the packet back to its original size. The smaller packets takes lower time to transmit. Hence, the end to end packet transmission speed increases. It improves the bandwidth utilization. Since, compared to smaller packet sizes the bigger packet sizes are more prone to channel noise. The compressed packets have lower chance to be corrupted in a noisy channel. Packet error/loss rate remains low. For a compressed header ℓ^{h}_{com} and a compressed payload ℓ^{p}_{com} packet error rate for packet size L_h and L_p will be [SAM03, KS06]

$$P^h_{com} = 1 - (1 - b)^{L_h}$$
$$P^p_{com} = 1 - (1 - b)^{L_p} \tag{2.9}$$

where P^h_{com} and P^p_{com} are the packet loss rate of compressed header and compressed payload size packets. It seems worthy of implementing compression in ad hoc networks, but there are several things that are needed to be considered before implementing the compression algorithm. Such as: choice of the compression algorithm, compression ratio, error rate of compression, delay etc. Also in which layer of the protocol stack the compression can be implemented, whether it can be done on hardware level or software etc.

Hardware level implementation is more expensive than software level implementation [SGGN09]. Also there is difficulty in modifying or updating the compression algorithm after the installation etc. Therefore, software level compression algorithm implementation is preferred for the ad hoc networks and in this thesis as well.

2.8 State of the Art for Compression

Several researches have been done about the compression arbitration between the choice of the algorithm based on compression type, energy consumption, algorithm time and space complexity, transmission power, number of nodes etc [KL05, YLYW10, RCHS09]. A summary of different compression algorithms have been published in [YLYW10] where the algorithms are differentiated and listed according to compression name, characteristics (lossless compression or lossy), evaluation index (the analysis type whether it has been done to investigate the compression ratio, error, energy etc) and implementation platform (within real time ad hoc network or simulator). A comprehensive description and analysis have been done for different compression algorithm in [RCHS09, KL05].

However, the compression method does not remain efficient in various scenarios.

The computational energy for the compression has been considered first in [SM06] and then [RCHS09]. After that in [YLL+09] it is showed that compression does not always ensure energy reduction, even though the amount of transmission data reduces. Hence, to achieve maximum energy efficiency compression arbitration modelling and adaptation mechanism is proposed in [YLL+09, YLYW10]. These arbitration methods mainly predicts the energy savings for the compression and decides whether to compress the data before transmission or not. In [YLL+09, YLW10] a compression ratio forecasting is proposed where the arbitration is done between compression ratio, error level and energy efficiency. However, the energy efficient packet size optimization for compression arbitration has not been done yet. Only in [CM11] compression ratio for various packet size has been estimated and the throughput gain of the channel is observed. In this thesis energy efficient packet size is estimated with the compression method in chapter five.

2.9 Compression Types and Scope

Compression method can be categorized in to three types:

1. Header compression
2. Payload compression
3. Bulk compression or whole packet compression

2.10 Header Compression

Header compression mechanism compress the header of a packet [NMQN09, NMQ+09, NMQN08]. Since the compressed header size is reduced in size, it speeds up the trans-

Table 2-F: Header compression of different layers [TF03, CJ99, BBD+01]

Protocol Header Name	Layer Name	Header Size	Compressed Header Size
IP4/TCP	Network and Transport	$40B$	$4B$
IP4/UDP	Network and Transport	$28B$	$1B$
IP4/UDP/RTP	Network, Transport and Application layer	$40B$	$2B$-$4B$

mission and also reduces overhead size. It was show in chapter 2 that the packet header consumes a large portion of a packet. If the overhead size decreases due to compression then the packet administration cost should be lower and energy efficiency would increase. From last chapter it was show that a packet can contain large header for example WLAN packets. As headers are added in different layers of the protocol stack. Hence, compression is applied in specific layer. The Internet Engineering Task Force (IETF) has defined the standardization of header compression schemes in [CJ99, BBD+01, DNP99, Jac90]. According to the standard header compression is specified in table 2-F for different layers.

Currently, header compression is mostly available for the network, transport and application layer. In these layers the compression can be implemented without affecting the underlying layers [CM11, SGGN09]. Although the lower layers MAC, data link or PHY layer provides higher compression ratio but implementation is quite difficult compared to upper layers.

2.10.1 Types of Header Compression

Mainly there are four types of header compression [TF03, CJ99, BBD+01, DNP99, Jac90]

- Van Jacobson Header Compression (VJHC) [Jac90]: It was developed by V. Jacobson in 1990 and most commonly known as VJ compression. It can compress IPv4 and TCP header together as a combined header field and the remaining fields within the packet remains same as before the compression.

- IP Header Compression (IPHC) [DNP99]: The main difference between IPHC and VJHC compression is, IPHC compress only the IP header.
- Compressed RTP [Com06, CJ99]: It compresses the header of IP/UDP/RTP packet. Usually CRTP is used for the real time applications like multimedia (audio, video) or real-time group conferencing services such as: VoIP etc. RTP protocol belongs to the application layer. It provides source identification, synchronization support, QoS feedback such as: loss detection, content identification etc. The payload size of RTP protocol for an audio is usually 20-128 bytes and header consists of IP, UDP and RTP header. Hence, a large portion of the RTP packet is used as a header, and therefore sometimes the overhead size becomes larger than the data (payload) portion. Hence, it is inefficient to transmit RTP packets without compression. RTP compression compress the IP header portion and squeeze the original header size to smaller size header.
- Robust Header Compression (ROHC) [BBD+01, NMQN08]: This compression algorithm can compress IP/UDP packet flows, IP/UDP/RTP headers to only one byte. This compression was developed in 2001 and can work efficiently in severe noisy channel. It was developed specially for the wireless networks to reduce overhead, packet loss and improve interactive response time.

In some research [NMQN09, NMQ+09, NMQN08] header compression algorithms are customized by using packet aggregation to increase the system efficiency. Since the aim of this research is to increase energy efficiency by packet size optimization and compression is used to reduce overhead only, off the shelf compressions are used to reduce complexity.

2.11 Payload Compression

Payload compression compress the payload generated in the application layer. In the application layer there are several applications such as: file sharing, web browsing, real time applications: multimedia, VoIP etc. The application generates payload data for transmission. The compression algorithm can be used to compress the payload. After that the compressed payload is passed through the other layers and added up with the headers of other layers. At the other end the receiver node reverse back the process and decompress the payload.

Since compression has been proved as an efficient technique, numerous compression algorithm have been proposed [YLYW10, BA06, KL05, SM06] in diverse area. Among these compression algorithms for payload compression. Some well known algorithms are Gzip, bzip2, zlib, rzip, LZW etc. According the result analysis and investigation of [RCHS09, KL05, TDV08] in total 11 types of compression algorithms are compared considering CPU, network peripherals, Ethernet card, memory. It is found that LZO is the most energy efficient algorithm. Moreover, in terms of complexity and usage it is the most useful compression algorithm and easy to implement. LZO provides the highest compression ratio, and lowest compression, decompression time compared to the other compression algorithms. It needs less instructions to compress and decompress, only 3 add instructions for the web and 9 for text data. However, considering the throughput it shows the lowest performance. The highest throughput shows by PPMD then bzip2, after that compress, zlib and then LZO compression. But it can compress very fast with lowest error rate which is necessary for faster packet transmission without compression error.

2.12 Bulk Compression

In bulk or whole packet compression [Com95, TF03] the entire packet is treated as a block of information and compressed using compression algorithm. The compressor constructs a common sequence dictionary according to the packet information, and matches the block sequences with a shorter representation. The new representation of the packet is used for transmission as a compressed packet. This bulk compression uses predefined dictionary which is needed to be installed beforehand. Whenever the compressed packet is reached at the destination node the receiver node must need to use an identical dictionary for decompression. It provides high compression ratios but it has two major drawbacks. The compression dictionary must be synchronised with each other and they have to be identical, another problem is compression dictionary consumes a large memory space.

To overcome this synchronization problem Packet by Packet dictionary algorithm is proposed [Com95]. However, it needs smaller dictionary and the compression technique is almost similar to bulk compression. To suppress the use of the dictionary based compression later Guess-Table-Based compression has been invented [Com95]. Algorithm is used to predict the next incoming bytes if the guess become correct the data is not transmitted otherwise the data is transmitted. Guess based compression is used for the VoIP service to improve the delivery speed and overall performance of the network.

2.12.1 How Voice Compression Works

In VoIP service the end user talks in front of the microphone and generates audio signal. To convert the analogue audio signal into digital signal the audio signal is digitized into pulse code modulation (PCM) signal [Com06]. The voice encoder-decoder process does the job of altering the analogue to digital PCM signal. After that, the PCM signal is passed through the compression algorithm and then the compressed packets are forwarded to the destination. The receiver node uses the reverse technique to generate the original audio.

2.13 Resource Consumption for the Compressor

Just like as every method and protocol compression decompression algorithms have specific cost. Such as: computational energy, computational complexity, time complexity or delay, space complexity or memory space requirements etc. However, the main cost are extra delay and processing cost which are considered in this thesis. Hence, the compression technique becomes only advantageous if the overall delay remains within the delay limit and the processing energy cost of compression remains lower than the amount of energy saved by the compression technique.

2.13.1 Energy Cost Estimation For Compression Decompression

The compression decompression energy cost generates from the computational energy cost from the microprocessor. In some research it is stated that computational energy is thousand times lower than the transmission energy cost [BA06]. Hence, computational energy cost must be very low. To estimate the energy consumption several researchers have followed different ways. The authors of [BA06] Kenneth and Krste have used one specially designed research based computer "Skiff" which is equipped with 233 MHz Strong ARM SA-110 processor, 802.11 ethernet card and other peripherals. From this machine it is estimated that the energy required to compress 1 bit of web data with LZO compression algorithm needs 3 instructions. Two instructions are for the compression and one is for the decompression. The energy cost of one instructions is equal to the estimated energy cost for one addition instruction which is $6.59 \times 10^{-9}\,J$.

In [YLYW10] one of the most popular ad hoc node MicaZ node is used to estimate the energy consumption for the compression methods which is equipped with a 8-bit Atmel ATmega 128L, 8 MHz microcontroller with CC2420 radio unit. Energy is estimated by considering a fixed power for the microcontroller with ATMEL AVR Studio and after that the energy efficiency is estimated. In [YLL+09] also dealt with the energy consumption of compression. But in their estimation they calculated the energy consumption of the microcontroller for the compression as a constant value and used in the simulator. After that energy efficient new compression arbitration method is proposed. In [TDV08] seven different off the shelf compression is analysed and the trade off between dissemination time and energy consumption has been estimated. They have used MSP430F1611 microcontroller, 10 kB of RAM, 48 kB of internal and 1 MB of external flash, and a CC2420 radio. By using an oscilloscope current, voltage is measured and energy is estimated. In [SM06] energy consumption for different algorithm is estimated for fixed compression algorithm. They have used MSP4 microcontroller to mimic the 8-bit Atmel ATmega128L microcontroller of MicaZ node.

Hence, in several research energy consumption has been estimated in several ways. Therefore, which one should be considered as a reference point for this thesis is an issue. Basically in this thesis our major concern is to use the off-the-shelf fixed compression algorithms to reduce header overhead and increase efficiency. This research is about the energy efficient packet size optimization where compression will be used as a supplement to reduce wastage. Hence, rather to do main experiment for the compression algorithm and estimation of energy cost for the compression, the results of LZO compression algorithm [BA06] and the energy usage of this algorithm is used directly in this thesis.

2.13.2 Delay Estimation for the Compression

In ad hoc networking application some of the application are delay tolerant such as environment monitoring but some of them are delay sensitive such as real time applications like multimedia or video. Hence, for the real time application delay estimation is important. Since, LZO algorithm was already chosen in the last section for computation energy cost estimation. Delay time for the compression is considered for the compression and decompression time of LZO. From [RCHS09] its found that to compress and decompress with LZO algorithm takes less than 1 second only for 3KB to 8MB of application data. Hence, this smallest delay would not create any severe service interruption.

2.14 FEC and Compression

Header compression compress the header and reduces the overhead of whole packet. Since, compression reduces the overhead, this technique could be an useful method for the FEC (see chapter 2.3). FEC improves the reliability of the channel but large portion of a packet is wasted due to data administration purposes for example the overheads of parity checking. Hence, if these two methods are used together the overhead can be reduced by using compression and reliability can be improved by FEC. With the compressed header ℓ_h the reliability of any packet size L with the FEC method can be given as below [SAM03, KS06]

$$r_{com} = \sum_{i=0}^{t} \begin{pmatrix} L_h \\ i \end{pmatrix} BER^i (1 - BER)^{L_h - i} \tag{2.10}$$

where, L_h is the total packet size after the compressed header size. BER is the bit error rates of the channel. Here, $L_h = \ell + \ell_h^{com}$ the sum of compressed header and the payload.

If payload compression is used then the compressed payload size will be ℓ_{com}^{pay}. Then with the compressed payload and FEC code the reliability of the packet will be

$$r_{com} = \sum_{i=0}^{t} \begin{pmatrix} L_p \\ i \end{pmatrix} BER^i (1 - BER)^{L_p - i} \tag{2.11}$$

where, L_p is the total packet size after the compressed payload size. Here, $L_p = \ell_h + \ell_{com}^{pay}$ the sum of compressed payload and the header.

2.15 Conclusion

The packet structure for the WLAN in general and also for the WLAN VoIP are observed. BCH FEC is investigated and its mechanism is illustrated to reduce packet loss in noisy channel. However, it shows, FEC and header fields incorporate a large portion within a packet only for the packet administration as an overhead. Hence, packet compression is discussed to reduce overhead. The cRTP and LZO compression algorithm are considered as reference to investigate further. In this chapter only those data is considered as a reference which have already been used in many researches. In chapter 5 and 7 the energy efficiency is estimated considering these protocols.

References

[ASSC02] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. Wireless sensor networks: a survey. *Computer Networks*, 38(4):393 – 422, 2002.

[BA06] K. C. Barr and K. Asanović. Energy-aware lossless data compression. *ACM Trans. Comput. Syst.*, 24:250–291, August 2006.

[BBD+01] C. Bormann, C. Burmeister, M. Degermark, H. Fukushima, H. Hannu, L-E. Jonsson, R. Hakenberg, T. Koren, K. Le, Z. Liu, A. Martensson, A. Miyazaki, K. Svanbro, T. Wiebke, T. Yoshimura, and H. Zheng. RObust Header Compression (ROHC): Framework and four profiles: RTP, UDP, ESP, and uncompressed . RFC 3095, July 2001.

[Bia00] G. Bianchi. Performance analysis of the IEEE 802.11 distributed coordination function. *Selected Areas in Communications, IEEE Journal on*, 18(3):535–547, mar 2000.

[BYJK07] G. Balakrishnan, M. Yang, Y. Jiang, and Y. Kim. Performance analysis of Error Control Codes for wireless sensor networks. In *Information Technology, 2007. ITNG '07. Fourth International Conference on*, pages 876 –879, april 2007.

[CJ99] S. Casner and V. Jacobson. Compressing IP/UDP/RTP Headers for Low-Speed Serial Links. RFC 2508, February 1999.

[CM11] S. Chawla and B.S. Manoj. Dynamic data compression in wireless networks. In *Advanced Networks and Telecommunication Systems (ANTS), 2011 IEEE 5th International Conference on*, pages 1 –3, dec. 2011.

[Com95] HP Technical Committee. HP case study: WAN link compression on HP routers, 1995.

[Com06] Cisco Technical Support Community. RTP header-compression or compressed RTP (cRTP). `http://www.cisco.com/en/US/tech/tk652/tk698/technologies_tech_note09186a0080094ae2.shtml`, February 2006.

[Com11] Cisco Technical Community. Voice Over IP - Per Call Bandwidth Consumption, Document ID: 7934. `http://www.cisco.com/en/US/tech/`

tk652/tk698/technologies_tech_note09186a0080094ae2.shtml, Dec 2011.

[com12] WildPackets Inc committe. 802.11 WLAN Packets and Protocols. http://www.wildpackets.com/resources/compendium/wireless_lan/wlan_packets/printable, Dec 2012.

[CXSM06] L. Cai, Y. Xiao, X. Shen, and J. W. Mark. VoIP over WLAN: voice capacity, admission control, QoS, and MAC: Research Articles. *Int. J. Commun. Syst.*, 19:491–508, May 2006.

[DKJ02] L. Dang, D. Kelly, and C. Jennings. *Practical VoIP Using Vocal.* O'Reilly & Associates, Inc., Sebastopol, CA, USA, 2002.

[DNP99] M. Degermark, B. Nordgren, and S. Pink. IP Header Compression. RFC 2507, February 1999.

[DPG00] J. Davidson, J. Peters, and B. Grace. *Voice over IP fundamentals.* Cisco Press, first edition, 2000.

[Gas05] M. S. Gast. *802.11 Wireless Networks: The Definitive Guide, Second Edition.* O'Reilly Media, Inc., 2005.

[GK03] S. Garg and M. Kappes. An experimental study of throughput for UDP and VoIP traffic in IEEE 802.11b networks. In *Wireless Communications and Networking, 2003. WCNC 2003. 2003 IEEE*, volume 3, pages 1748 –1753 vol.3, March 2003.

[Goo02] B. Goode. Voice over Internet Protocol (VoIP). *Proceedings of the IEEE*, 90(9):1495 – 1517, sep 2002.

[GR05] V. Guruswami and A. Rudra. Limits to list decoding Reed-Solomon codes. In *Proceedings of the thirty-seventh annual ACM symposium on Theory of computing*, STOC '05, pages 602–609, New York, NY, USA, 2005. ACM.

[GS99] M. Goel and N.R. Shanbhag. Low-power channel coding via dynamic reconfiguration. In *Acoustics, Speech, and Signal Processing, 1999. ICASSP '99. Proceedings., 1999 IEEE International Conference on*, volume 4, pages 1893 –1896 vol.4, March 1999.

[HAM50] R.W. HAMMING. Error detecting and error correcting codes. *BELL SYSTEM TECHNICAL JOURNAL*, 29(2):147–160, 1950.

[HSI06] S.L. Howard, C. Schlegel, and K. Iniewski. Error control coding in low-power wireless sensor networks: when is ECC energy-efficient? *EURASIP J. Wirel. Commun. Netw.*, 2006:29–29, April 2006.

[Jac90] V. Jacobson. Compressing TCP/IP headers for low-speed serial links. RFC 1144, February 1990.

[KBP07] J.H. Kleinschmidt, W.C. Borelli, and M.E. Pellenz. An analytical model for energy efficiency of error control schemes in sensor networks. In *Communications, 2007. ICC '07. IEEE International Conference on*, pages 3895 –3900, june 2007.

[KBS+98] T. J. Kostas, M. S. Borella, I. Sidhu, G. M. Schuster, J. Grabiec, and J. Mahler. Real-time voice over packet-switched networks. *Network, IEEE*, 12(1):18 –27, Feb 1998.

[KL05] N. Kimura and S. Latifi. A survey on data compression in wireless sensor networks. In *Information Technology: Coding and Computing, 2005. ITCC 2005. International Conference on*, volume 2, pages 8 – 13 Vol. 2, april 2005.

[KP09] S. Karapantazis and F. Pavlidou. VoIP: A comprehensive survey on a promising technology. *Computer Networks*, 53(12):2050–2090, August 2009.

[KS06] Z.H. Kashani and M. Shiva. BCH coding and multi-hop communication in wireless sensor networks. In *Wireless and Optical Communications Networks, 2006 IFIP International Conference on*, pages 5 pp. –5, 2006.

[KSPR04] H. Karvonen, Z. Shelby, and C. Pomalaza-Raez. Coding for energy efficient wireless embedded networks. In *Wireless Ad-Hoc Networks, 2004 International Workshop on*, 2004.

[LC04] S. Lin and D.J. Costello. *Error Control Coding (2nd Edition)*. Prentice Hall, 2 edition, June 2004.

[LMZG97] H. Liu, H. Ma, M.E. Zarki, and S. Gupta. Error control schemes for networks: an overview. *Mob. Netw. Appl.*, 2(2):167–182, October 1997.

[LPK13] F.R. Lone, A. Puri, and S. Kumar. Performance comparison of reed solomon code and bch code over rayleigh fading channel. *CoRR*, abs/1307.6930, 2013.

[mac12] IEEE draft standard for local and Metropolitan Area Networks - specific requirements - part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications - amendment 3: Enhancements for very high throughput in the 60 GHz band. *IEEE P802.11ad/D8.0, May 2012 (Draft Amendment based on IEEE 802.11-2012)*, pages 1 –667, 5 2012.

[Mas65] J. Massey. Step-by-step decoding of the Bose Chaudhuri Hocquenghem codes. *Information Theory, IEEE Transactions on*, 11(4):580 – 585, oct 1965.

[McE84] R.J. McEliece. *Encyclopedia of Mathematics and Its Applications: The Theory of Information and Coding: A Mathematical Framework for Communication*. Cambridge University Press, New York, NY, USA, 1984.

[MV02] S. Mitali and K.P. Viktor. System-level energy tradeoffs for collaborative computation. In *In International Conference on Communications (ICC), Workshop on Integrated Management of Power Aware Communications, Computing and NeTworking*. Kluwer, 2002.

[NK11] A. Nandi and S. Kundu. Energy level performance of error control schemes in wireless sensor networks. In *Devices and Communications (ICDeCom), 2011 International Conference on*, pages 1 –5, feb. 2011.

[NMQ+09] A.G. Nascimento, E. Mota, S. Queiroz, L. Galvao, and E. Nascimento. Towards an efficient header compression scheme to improve VoIP over wireless mesh networks. In *Computers and Communications, 2009. ISCC 2009. IEEE Symposium on*, pages 170 –175, july 2009.

[NMQN08] A. Nascimento, E. Mota, S. Queiroz, and E. Nascimento. Header compression for VoIP over multi-hop wireless mesh networks. In *Computers and Communications, 2008. ISCC 2008. IEEE Symposium on*, pages 286 –291, july 2008.

[NMQN09] A.G.O. Nascimento, E. Mota, S. Queiroz, and E. Nascimento. An alternative approach for header compression over wireless mesh networks.

In *Advanced Information Networking and Applications Workshops, 2009. WAINA '09. International Conference on*, pages 165 –169, may 2009.

[NRKS12] M.Y. Naderi, H.R. Rabiee, M. Khansari, and M. Salehi. Error control for multimedia communications in wireless sensor networks: A comparative performance analysis. *Ad Hoc Netw.*, 10(6):1028–1042, August 2012.

[PSF10] M. Pellenz, R. Souza, and M. Fonseca. Error control coding in wireless sensor networks. *Telecommunication Systems*, 44:61–68, 2010. 10.1007/s11235-009-9222-5.

[RCHS09] A. Reinhardt, D. Christin, M. Hollick, and R. Steinmetz. On the energy efficiency of lossless data compression in wireless sensor networks. In *Local Computer Networks, 2009. LCN 2009. IEEE 34th Conference on*, pages 873 –880, oct. 2009.

[SA07] P. Stuedi and G. Alonso. Wireless ad hoc VoIP. In *Proceedings of the 2007 Workshop on Middleware for next-generation converged networks and applications*, MNCNA '07, pages 8:1–8:5, New York, NY, USA, 2007. ACM.

[SAM03] Y. Sankarasubramaniam, I. F. Akyildiz, and S. W. McLaughlin. Energy efficiency based packet size optimization in wireless sensor networks. In *Sensor Network Protocols and Applications, 2003. Proceedings of the First IEEE. 2003 IEEE International Workshop on*, pages 1 – 8, 2003.

[SGGN09] A. Sharma, L. Golubchik, R. Govindan, and M. J. Neely. Dynamic data compression in multi-hop wireless networks. In John R. Douceur, Albert G. Greenberg, Thomas Bonald, and Jason Nieh, editors, *SIGMETRICS / Performance*, pages 145–156. ACM, 2009.

[SM06] C. M. Sadler and M. Martonosi. Data compression algorithms for energy-constrained devices in delay tolerant networks. In *Proceedings of the 4th international conference on Embedded networked sensor systems*, SenSys '06, pages 265–278, New York, NY, USA, 2006. ACM.

[SP10a] J. Singh and D. Pesch. Energy efficient soft-decision error control in wireless sensor networks. In *Wireless and Mobile Networking Conference (WMNC), 2010 Third Joint IFIP*, pages 1 –6, oct. 2010.

[SP10b] J. Singh and D. Pesch. Energy efficient soft-decision error control in wireless sensor networks. In *Wireless and Mobile Networking Conference (WMNC), 2010 Third Joint IFIP*, pages 1 –6, oct. 2010.

[Sup12] Atmel Technical Support. Atmega128 datasheet, December 2012.

[TDV08] N. Tsiftes, A. Dunkels, and T. Voigt. Efficient sensor network reprogramming through compression of executable modules. In *Sensor, Mesh and Ad Hoc Communications and Networks, 2008. SECON '08. 5th Annual IEEE Communications Society Conference on*, pages 359 –367, june 2008.

[Tea03] Microsoft Technical Team. How 802.11 Wireless Works. `http://technet.microsoft.com/en-us/library/cc757419(v=ws.10).aspx`, March 2003.

[TF03] C. S. Tye and Dr. G. Fairhurst. A review of IP packet compression techniques. PGNet 2003, 2003.

[TYL08] Zhen Tian, Dongfeng Yuan, and Quanquan Liang. Energy efficiency analysis of error control schemes in wireless sensor networks. In *Wireless Communications and Mobile Computing Conference, 2008. IWCMC '08. International*, pages 401 –405, aug. 2008.

[VA09] M.C. Vuran and I.F. Akyildiz. Error control in wireless sensor networks: A cross layer analysis. *Networking, IEEE/ACM Transactions on*, 17(4):1186 –1199, Aug 2009.

[Wic94] S.B. Wicker. *Reed-Solomon Codes and Their Applications*. IEEE Press, Piscataway, NJ, USA, 1994.

[WPW06] X. Wang, A. Patil, and W. Wang. VoIP over wireless mesh networks: challenges and approaches. In *Proceedings of the 2nd annual international workshop on Wireless internet*, WICON '06, New York, NY, USA, 2006. ACM.

[WWWL10] L. Wang, M. Wu, L. Wei, and M. Li. An adaptive forward error control method for voice communication. In *Networking and Digital Society (ICNDS), 2010 2nd International Conference on*, volume 2, pages 186 –189, may 2010.

[YLL+09] B. Ying, W. Liu, Y. Liu, H. Yang, and H. Wang. Energy-efficient node-level compression arbitration for wireless sensor networks. In *Advanced Communication Technology, 2009. ICACT 2009. 11th International Conference on*, volume 01, pages 564 –568, feb. 2009.

[YLW10] B. Ying, Y. Liu, and H. Wang. Improved adaptive compression arbitration system for wireless sensor networks. *Tsinghua Science and Technology*, 15(2):202 – 208, 2010.

[YLYW10] B. Ying, Y. Liu, H. Yang, and H. Wang. Evaluation of tunable data compression in energy-aware wireless sensor networks. *Sensors*, 10(4):3195–3217, April 2010.

[ZDQ08] T. Zhen, Y. Dongfeng, and L. Quanquan. Energy efficiency analysis of error control schemes in wireless sensor networks. In *Wireless Communications and Mobile Computing Conference, 2008. IWCMC '08. International*, pages 401 –405, aug. 2008.

Chapter 3

Packet Size as a Function of Channel Noise

The channel noise has a direct influence over the packet size. As the packet size increases it becomes error prone and consequently happen more packet loss. Because the bigger packet sizes are more likely to be affected by the noise. The channel noise is dependent on various parameters such as: distance between the nodes, interference, transmission power etc. This chapter describes the relationship between the packet size, channel noise and various parameters that have impact on the packet loss.

3.1 Channel Noise and Packet Size

Since the overhead ratio of bigger packet is less than the smaller size packets. In the noise free channel, the bigger packets are found more efficient. But bigger packets are more error prone in a noisy environment. Due to noise in the channel the bits inside the packets are corrupted and sometimes lost. The probability of corrupting the bigger packets is comparatively higher than smaller packets.

Bit error rate (BER) and transmitted packet sizes (L) are related with each other. Packet error rate (PER) per is dependent on BER (ber).

For a Additive White Gaussian Noisy (AWGN) channel with DQPSK modulated data, and assuming independent bit errors, the relationship between per, ber and L can be described as [VA06, SAM03, KS06, VA06, ZDQ08]

$$\begin{aligned} per &= 1-(1-ber)^L \\ &= 1-(1-ber)^{(\alpha+\ell+\tau)} \end{aligned} \tag{3.1}$$

Where, α, ℓ and τ represent the header, payload and trailer size.

The BER depends on various parameters such as: transmitted signal strength, interference etc. Usually signal power level is fixed for most of the ad hoc nodes. Hence, whenever the channel noise changes, signal to noise ratio (SNR) changes as well. The signal to noise ratio (snr) is obtained by [MHS+07, EAK+02]

$$SNR = E_b/N_0(R_{mod}/B_t) \tag{3.2}$$

where E_b is the energy required per information bits, N_0 is the thermal noise, R is the highest bit rate ($2\,Mbps$) of modulation scheme and B_t is the system bandwidth ($2\,MHz$). According to general rule of thumb greater than or equal to $20\,dB$ is considered as good quality SNR [VK12]. Hence, SNR value is assumed $20\,dB$. In this equation (3.2) E_b/N_0 specifies the energy per bit relative to the noise power.

Considering an AWGN and Rayleigh fading channel with DQPSK modulated data, the relationship between BER and E_b/N_0 is given by [CBB10, EAK+02]

$$\begin{aligned} ber &= \frac{1}{2}e^{-\frac{E_b}{N_0}} \\ \Rightarrow E_b/N_0 &= -ln(2ber) \end{aligned} \tag{3.3}$$

Hence, whenever the bit error rate increases or decreases the energy per bit value changes.

3.1.1 Energy Per Bit Calculation

To calculate the energy per bit we have used the energy model which is used in [SAM03, KSPR04, KS06]. Based on the radio power consumption, the energy required to communicate one single bit across a single hop is given by

$$E_b = E_{tx} + E_{rx} + \frac{E_{enc-dec}}{\ell} \tag{3.4}$$

where, E_b is the energy per bit, $E_{enc-dec}$ represents the total processing energy cost of a packet to encode and decode. In general the encoding energy is considered as very low energy and ignored from the energy per bit calculation [WN10, HWT$^+$11a, VA08, YFD09]. Decoding energy is also low and only included in [SAM03, KSPR04] whenever the error correction method is used. Error correction method is described in chapter 3. In our further analysis the encoding energy cost is not considered in $E_{enc-dec}$ and only the decoding cost (E_{dec}) is included whenever error correction method is used as a error correction processing energy cost. Among the rest of the parameters in equation (3.4) ℓ is the payload size of a packet and E_{tx} and E_{rx} are the transmission and reception energy cost for a single bit. In terms of power consumption transmission energy E_{tx} and E_{rx} can be expressed as,

$$E_{tx} = \frac{1}{\ell}\left((P_{tx})\,\frac{L}{R}\right) \ \mathit{and} \ E_{rx} = \frac{1}{\ell}\left((P_{rx})\frac{L}{R}\right) \tag{3.5}$$

where,

- $L = \ell + \alpha + \tau$ is the total packet length in bits. α and τ represent the length of the header and trailer size
- P_{tx}, P_{rx} are the transmission and reception power

- R is the data rate in bits/sec.

3.1.2 Transmission and Reception Power Estimation

The source node transmits with a fixed transmission power. The strength of the signal received by the receiver antenna depends on the path loss, fading, noise etc. Path loss defines the signal power reduction due to the distance between the transmitter and receiver antennas [ZP98, ESWW00]. Since, radio waves propagates in free space, the power level disseminate as a square of distance [ZP98, ESWW00]. This happens due to the spreading of radio waves as the range increases. The path loss exponent γ can be defined as below [ZP98, ESWW00]

$$\gamma = 20log_{10}(4\pi D/\lambda) \tag{3.6}$$

where, D is the distance between transmitter and receiver. Line-of-sight propagation holds only the first few meter distance between the transmitter and receiver. Only for about the first $6m$ [ZP98]. Beyond $6m$, propagation losses in indoors dense area increase at up to 30dB per $30m$ (such as dense office environments). Usually in our home or office environment the WLAN router provides better signal strength up to $6m$ approximately. In this thesis it is assumed the transmitter and receiver nodes are within $6m$ distance and the maximum transmission range is $30m$.

In equation (3.6) λ is the wave length (meter) that is expressed as c/f where c is the speed of light ($3 \times 10^8\, ms^{-1}$) and f is the transmitted signal frequency that is 2.4 GHz. For the value of $\lambda = 0.125\, m$ the γ will be $69.6\, dBm$.

The transmitter power P_{tx} denoted in [EAK+02, JEW02, ESWW00, TJV+09] as

$$P_{tx} = N_0 + NF_{rx} + F_{mar} + SNR + \gamma \tag{3.7}$$

where, the thermal noise N_0 can be estimated as below

$$N_0 = kTB_t \tag{3.8}$$

where k is a Boltzmann constant ($1.38 \times 10^{-23}\, J/K$), T is the system temperature (kelvin) and B is the channel bandwidth (Hz). For the $20\,^oC$ ($293\,K$) room temperature thermal noise N_0 is $8 \times 10^{-12}\, mW$ or $-111\, dBm$. The receiver noise NF_{rx} is generated due to internal circuit noise and it is assumed to be $10\, dBm$ [ESWW00].

Fade margin specifies the fading quantity of the transmitted signal that is expressed as

$$F_{mar} = P_{rx} - R_{th} \tag{3.9}$$

where, F_{mar} is the fade margin and R_{th} is a receiver threshold. Less than the receiver threshold value the receiver node cannot receive the signal. Hence, it is assumed that R_{th} is configured such that it can satisfy fading margin and also able to detect low signals. For a reliable channel, fade margin should be $20\, dB - 30\, dB$ [ZP98, ESWW00].

It is estimated that the transmission power P_{tx} is $19\, dBm$. Typical WiFi transmission power of wireless LAN devices stays between $10\, dBm - 30\, dBm$ [FN01, FN01]. Therefore, the assumptions considered here are correct for the WLAN transmission power estimation.

The receiver power sensitivity P_{rx} of equations (3.5) is given in [ESWW00] as

$$P_{rx} = N_0 + NF_{rx} + SNR \tag{3.10}$$

After calculating all the values the receiver power is estimated $-81\, dBm$ from equation (3.10). The typical range of wireless received signal power over a network (WiFi 802.11) varies between $-60\, dBm$ to $-100\, dBm$ [CYLC10]. Therefore, to estimate the receiver power the values used for NF_{rx}, SNR and F_{mar} are correct for the WLAN interface.

From the values of transmission power and reception power the energy per bit can be estimated from equation (3.4). After that the energy per bit and noise level BER can be estimated.

Most of the researchers of WLAN network have followed BER level 10^{-4} to 10^{-6} [ZP98, HWT+11b, WXC+10, SPRKH05]. For an example, [HWT+11b] has used BER level 10^{-5} for the physical layer parameters optimization and optimal distance estimation. After that the BER level 10^{-4} to 8×10^{-4} have been assumed as a time varying channel condition in [WXC+10] for the dynamic optimal packet size optimization. The BER level 10^{-4} has also been considered in [SPRKH05]. Other than the above examples in [ZP98] BER is considered 10^{-6} to estimate transmission and reception power. From our analysis and following the above references we have decided to use the BER level 10^{-4} as a high noisy channel, 10^{-5} medium and 10^{-6} as a good channel condition.

However, what if the channel condition changes or fluctuates frequently. How to estimate the packet loss rate in that condition. The next section describes this issue.

3.2 Variable Noise Channel Condition and Average Packet Loss Estimation

In ad hoc network the links and channel condition changes frequently. As a result the error rate also changes randomly with time [GS06, ZRM95]. Therefore, estimating the energy efficient packet size for time varying channel condition is very important. To simulate a time varying error prone channel there are some models available such as: Uniform, Gilbert-Elliot, multi-state Markov error model etc [ZRM95, JPAIB99, RRC01]. However, among these Gilbert-Elliot error model is popular to simulate a time varying characteristics of an error prone channel which is described in the next section.

Gilbert-Elliot model is widely used to simulate the bursty packet loss behavior where error rate varies with time [ZRM95, JPAIB99, RRC01]. It is also known as two-state Markov model. The figure 3.1 illustrates a state diagram of a two-state Markov model of Gilbert-Elliott channel. In the "good" state packet losses occur with low probability P_{ber}^{gg} while in the "bad" state packet losses happen with high probability P_{ber}^{bb}. Hence, this two error probability defines the packet loss rate of two different states of a same channel.

$$\begin{aligned} P_{ber}^{gg} &= 1 - (1 - ber_{low})^L \\ P_{ber}^{bb} &= 1 - (1 - ber_{high})^L \end{aligned} \tag{3.11}$$

where ber_{low} is the lower BER and ber_{high} is the higher BER of the channel. The Gilbert-Elliot model has four probabilities. Based on the probabilities the average packet loss is estimated.

- The probability of staying in good condition (P_{gg})
- The probability of staying in bad condition (P_{bb})
- The probability of changing states from good to bad condition (P_{gb})
- The probability of changing states from bad to good condition (P_{bg})

The relationship between the probabilities is given below

$$\begin{aligned} P_{gb} &= 1 - P_{gg} \\ P_{bg} &= 1 - P_{bb} \end{aligned} \tag{3.12}$$

To find out the average packet loss rate P_{GB} from these four parameters of Gilbert-Elliot model, two more probabilities are needed [ZRM95]. One is transition probability and another one is steady state probability. After multiplying these two probabilities P_{GB} is estimated. The transition probability P_{tran} for the Gilbert-Elliot model is described

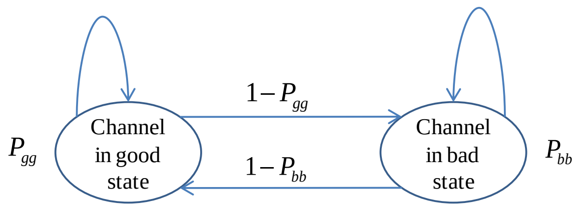

Figure 3.1: Two state Markov model of an error prone channel

as [JPAIB99]

$$P_{tran} = \begin{vmatrix} P_{gg} & P_{gb} \\ P_{bg} & P_{bb} \end{vmatrix}$$

The steady state probabilities (π_g, π_b) of being in states good and bad are stated below

$$\begin{vmatrix} \pi_g & \pi_b \end{vmatrix} \times \begin{vmatrix} P_{gg} & P_{gb} \\ P_{bg} & P_{bb} \end{vmatrix} = \begin{vmatrix} \pi_g & \pi_b \end{vmatrix}$$

where, $(\pi_g + \pi_b) = 1$. If we simplify this equation we can get

$$\pi_g = \frac{P_{bg}}{P_{bg} + P_{gb}} \quad and \quad \pi_b = \frac{P_{gb}}{P_{bg} + P_{gb}} \tag{3.13}$$

The average packet loss rate (P_{GE}) is given by [GO08]

$$P_{GE} = (P_{ber}^{gg}\pi_g + P_{ber}^{bb}\pi_b) = \frac{P_{ber}^{gg}P_{bg} + P_{ber}^{bb}P_{gb}}{P_{gb} + P_{bg}} \tag{3.14}$$

The average packet loss rate P_{GE} is shown in figure 3.2 for the corresponding packet sizes.

To do the simulation of time variable Gilbert Elliot channel model the two state Markov error model of NS-2 is used as a packet loss function. Rather comparing all the theoretical values stated for the P_{gb} and P_{bg} only one situation (P_{gb} and P_{bg} values are set 0.4 and 0.9) is considered in figure 3.2. It is assumed that the channel stays 60% time remains in the good state and 10% time remains in the bad state.

To compare the simulation with the theoretical results figure 3.3 is shown for P_{gb} and

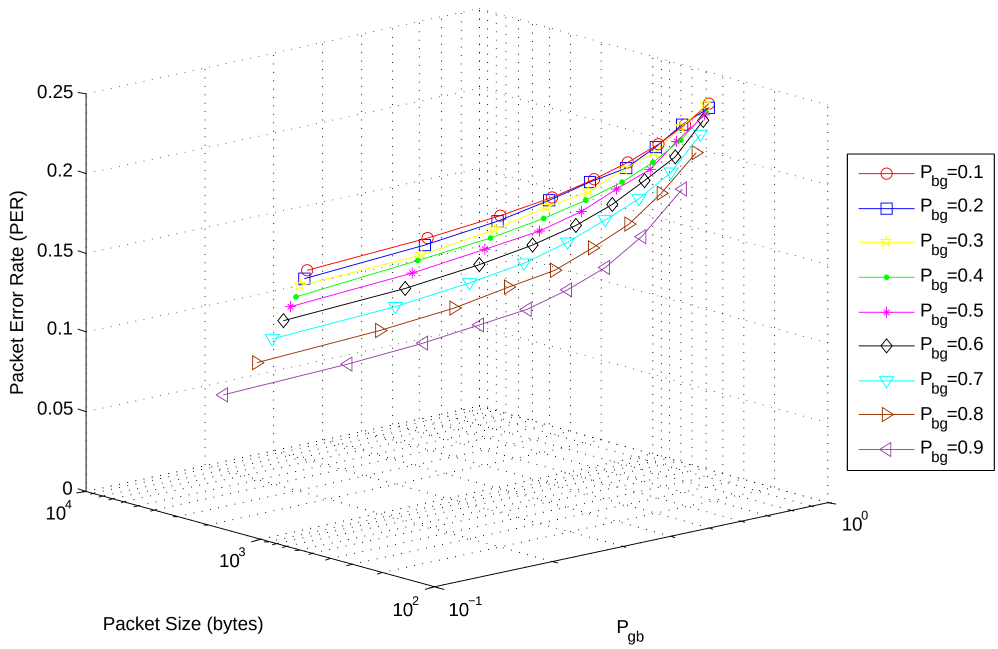

Figure 3.2: Packet Error Rate (PER) estimation for a time varying channel for different state changing values

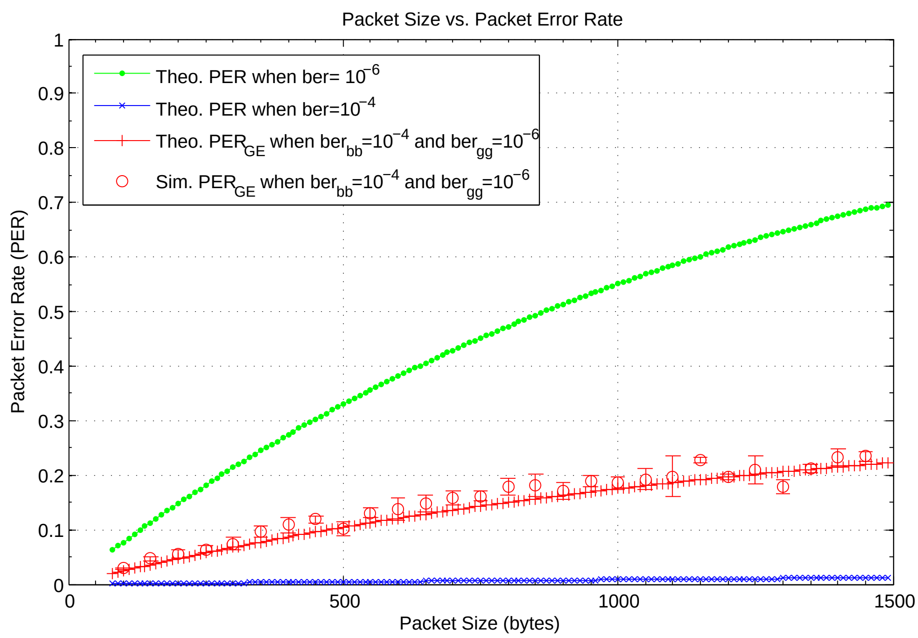

Figure 3.3: Packet error rate comparison of Gilbert-Elliot channel model and normal channel

P_{bg}, 0.4 and 0.9 respectively. The value of P_{GE} is compared with the two state markov packet loss model of the NS-2 in figure 3.3. Also the highest and lowest PER of a normal channel is shown in this figure for better comparison. From figure 3.3 it is found that simulation results are matched with the theoretical results.

3.3 Multi-hop Network and Packet Loss Estimation

In an ad hoc network the packets are traversed by multiple hops to reach the destination node. Since, the nodes do not have high transmission range to communicate directly with the destination node, The neighbor nodes act as a router to forward the data to the sink.

Whenever packets traverse from one node to another the distance between the nodes might be different. If the distance increases the transmitted signal strength decreases consequently it decreases the signal to noise ratio (see equation (3.6) and (3.2)). Hence, there will be different BER (equation (3.2)) for different links between the hops. To estimate the PER a linear multi-hop network topology has been considered in many research papers [KSPR04, SPRKH05] where the BER between the links is same.

Figure 3.4 shows the multi-hop linear chain topology, where n number of nodes are placed in a line with equal distance. This type of network is identical with large border area network, where the border line is covered with the sensor nodes to detect the intruder.

The number of hops have a relation with the PER [KSPR04, SPRKH05]. For a multi-hop linear chain topology packet error rate and reliability function (r_{mul}) follows the equation given below [KS06, KSPR04, SPRKH05, VA06]

$$\begin{aligned} P_{error} &= 1-(1-per)^{n-1} \\ &= 1-(1-ber)^{LN} \ \ and \\ r_{mul} &= 1-P_{error} = (1-ber)^{LN} \end{aligned} \tag{3.15}$$

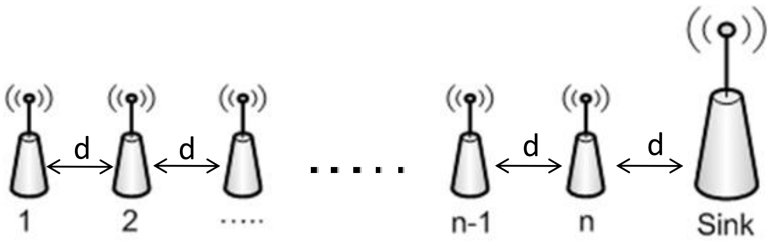

Figure 3.4: Chain like topology where all nodes are placed with equal space

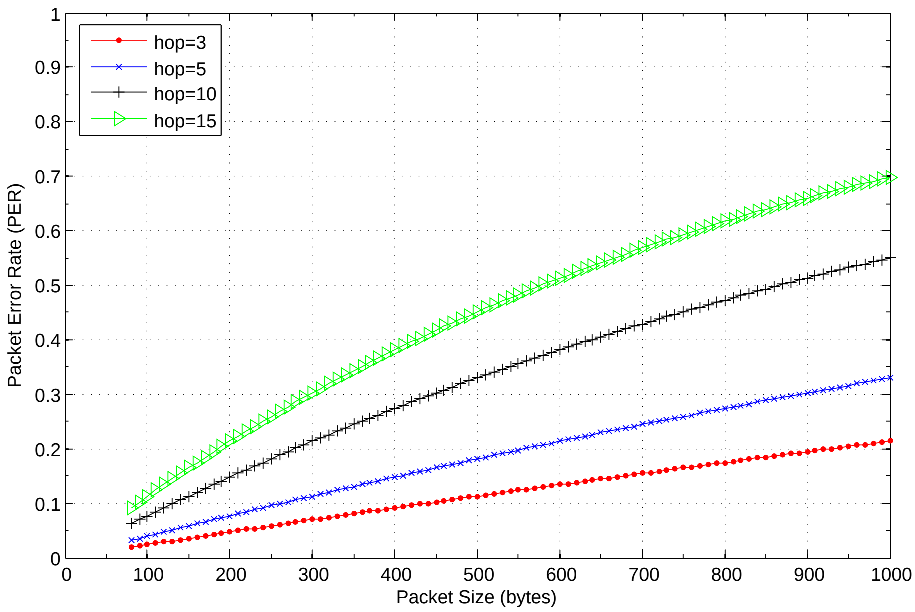

Figure 3.5: Packet error rate for the corresponding packet sizes for multi-hop linear chain network for $BER = 10^{-5}$

where, N is the number of hops, n is the number of nodes and P_{error} is the PER for the multi-hop linear chain topology. By using the equations (3.15) PER is plotted in figure 3.5. From this figure it is found that in a multi-hop network for a fixed BER, packet loss rate changes according the number of hops.

3.4 Conclusion

So far this chapter shows how the physical layer parameters the transmitted signal, channel noise, receiver noise etc related with the overall bit error rate and the packet loss. After that for a variable channel condition and multi-hop network scenario packet loss rate is estimated. It is found that noise and packet size has a strong relation and packet size has a direct impact on packet loss rate. Next chapter describes how the packet loss rate influences energy efficiency and the impact on the optimal packet size.

References

[CBB10] A. Chandra, D. Biswas, and C. Bose. BER performance of coherent PSK in rayleigh fading channel with imperfect phase estimation. In *Recent Trends in Information, Telecommunication and Computing (ITC), 2010 International Conference on*, pages 130–134. IEEE, 2010.

[CYLC10] Y. Chen, H. Yang, B. Liu, and J. Cheng. Transmission power optimization algorithm in wireless Ad Hoc networks. In *Communications and Mobile Computing (CMC), 2010 International Conference on*, volume 3, pages 358 –363, april 2010.

[EAK+02] J. Ebert, S. Aier, G. Kofahl, A. Becker, B. Burns, and A. Wolisz. Measurement and simulation of the energy consumption of an WLAN interface. *Technical University of Berlin, Telecommunication Networks Group, Tech. Rep. TKN-02-010*, 2002.

[ESWW00] J.P. Ebert, B. Stremmel, E. Wiederhold, and A. Wolisz. An energy-efficient power control approach for WLANs. 2000.

[FN01] L.M. Feeney and M. Nilsson. Investigating the energy consumption of a wireless network interface in an ad hoc networking environment. In *INFOCOM 2001. Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, volume 3, pages 1548–1557. IEEE, 2001.

[GO08] H. Gerhard and H. Oliver. The Gilbert-Elliott model for packet loss in Real Time services on the Internet. *Measuring, Modelling and Evaluation of Computer and Communication Systems (MMB), 2008 14th GI/ITG Conference* -, pages 1 –15, 31 2008-april 2 2008.

[GS06] R.K. Guha and S. Sarkar. Characterizing temporal SNR variation in 802.11 networks. In *Wireless Communications and Networking Conference, 2006. WCNC 2006. IEEE*, volume 3, pages 1408 –1413, april 2006.

[HWT+11a] M. Holland, T. Wang, B. Tavli, A. Seyedi, and W. Heinzelman. Optimizing physical-layer parameters for wireless sensor networks. *ACM Trans. Sen.*

Netw., 7:28:1–28:20, February 2011.

[HWT+11b] Matthew Holland, Tianqi Wang, Bulent Tavli, Alireza Seyedi, and Wendi Heinzelman. Optimizing physical-layer parameters for wireless sensor networks. *ACM Transactions on Sensor Networks (TOSN)*, 7(4):28, 2011.

[JEW02] G. Kofahl A. Becker B. Burns J. Ebert, S. Aier and A. Wolisz. Measurement and simulation of the energy consumption of an WLAN interface. technical report tkn-02-010, technical university berlin, telecommunication networks group, Jun 2002.

[JPAIB99] E. Jean-Pierre, W. Andreas, A.W. Ing, and T. Berlin. A Gilbert-Elliot bit error model and the efficient use in packet level simulation, March 1999.

[KS06] Z.H. Kashani and M. Shiva. BCH coding and multi-hop communication in wireless sensor networks. In *Wireless and Optical Communications Networks, 2006 IFIP International Conference on*, pages 5 pp. –5, 0-0 2006.

[KSPR04] H. Karvonen, Z. Shelby, and C. Pomalaza-Raez. Coding for energy efficient wireless embedded networks. In *Wireless Ad-Hoc Networks, 2004 International Workshop on*, 2004.

[MHS+07] P. Mahasukhon, M. Hempel, H. Sharif, T. Zhou, S. Ci, and H. Chen. BER analysis of 802.11b networks under mobility. In *Communications, 2007. ICC'07. IEEE International Conference on*, pages 4722–4727. IEEE, 2007.

[RRC01] T. Rancurel, D. Roviras, and F. Castanie. Expression of the capacity for the Gilbert channel in presence of interleaving. In *Acoustics, Speech, and Signal Processing, 2001. Proceedings. (ICASSP '01). 2001 IEEE International Conference on*, volume 4, pages 2533 –2536 vol.4, 2001.

[SAM03] Y. Sankarasubramaniam, I.F. Akyildiz, and S.W. McLaughlin. Energy efficiency based packet size optimization in wireless sensor networks. In *Sensor Network Protocols and Applications, 2003. Proceedings of the First IEEE. 2003 IEEE International Workshop on*, pages 1 – 8, 2003.

[SPRKH05] Z. Shelby, C. Pomalaza-Raez, H. Karvonen, and J. Haapola. Energy optimization in multihop wireless embedded and sensor networks. *International Journal of Wireless Information Networks*, 12:11–21, 2005.

[TJV+09] E. Tanghe, W. Joseph, L. Verloock, L. Martens, H. Capoen, K. Van Herwegen, and T. Buysschaert. Statistical validation of WLAN range calculated with propagation models for industrial environments by chipset-level received signal strength measurements. *Science, Measurement Technology, IET*, 3(3):244 –255, may 2009.

[VA06] M.C. Vuran and I.F. Akyildiz. Cross-Layer analysis of error control in wireless sensor networks. In *Sensor and Ad Hoc Communications and Networks, 2006. SECON '06. 2006 3rd Annual IEEE Communications Society on*, volume 2, pages 585 –594, sept. 2006.

[VA08] M.C. Vuran and I.F. Akyildiz. Cross-layer packet size optimization for wireless terrestrial, underwater, and underground sensor networks. In *INFOCOM 2008. The 27th Conference on Computer Communications. IEEE*, pages 226 –230, 2008.

[VK12] S. Varade and K. Kulat. BER comparison of Rayleigh fading, Rician fading and AWGN channel using chaotic communication based MIMO OFDM system. *International Journal of Soft Computing and Engineering (IJSCE) ISSN: 2231*, 2307, Jan 2012.

[WN10] S. Wang and J. Nie. Energy efficiency optimization of cooperative communication in wireless sensor networks. *EURASIP J. Wirel. Commun. Netw.*, 2010:3:1–3:8, January 2010.

[WXC+10] D. Wei, L. Xue, C. Chun, H. Yuan, C. Gong, L. Yunhao, and B. Jiajun. DPLC: Dynamic Packet Length Control in wireless sensor networks. In *INFOCOM, 2010 Proceedings IEEE*, pages 1 –9, march 2010.

[YFD09] M. Younis, O. Farrag, and W. D'Amico. Packet size optimization for increased throughput in multi-level security wireless networks. In *Proceedings of the 28th IEEE conference on Military communications*, MILCOM'09, pages 649–655, Piscataway, NJ, USA, 2009. IEEE Press.

[ZDQ08] T. Zhen, Y. Dongfeng, and L. Quanquan. Energy efficiency analysis of error control schemes in wireless sensor networks. In *Wireless Communications and Mobile Computing Conference, 2008. IWCMC '08. Interna-*

tional, pages 401 –405, aug. 2008.

[ZP98] J. Zyren and A. Petrick. Tutorial on basic link budget analysis. Application note: AN9804, Harris Semiconductor, April 1998.

[ZRM95] M. Zorzi, R.R. Rao, and L.B. Milstein. On the accuracy of a first-order Markov model for data transmission on fading channels. In *Universal Personal Communications. 1995. Record., 1995 Fourth IEEE International Conference on*, pages 211 –215, nov 1995.

Chapter 4

Energy Efficient Optimal Packet Size Estimation

In this chapter energy efficient optimal packet size is estimated. The highest energy efficient packet size is the optimal packet size. The energy efficiency and optimal packet size can be changed according to various conditions and scenarios. Such as: error free and error prone channel, multiple sources, time varying channel condition, number of hops etc. In this chapter energy efficient packet size is investigated for different scenarios.

4.1 Energy Cost of a Packet

Bits are the unit of information hence, a single bit can be represented as the unit of energy. Since in communication networks packets are formed by bits. The energy of packet can be estimated by using the energy for a single bit and then estimate the energy cost of different size packets. To estimate the energy budget when transmitting packets there are several things that are needed to be considered. Such as: distance between the nodes, frequency, modulation, noise level, fading etc. Moreover, ad hoc nodes can change their position with time and the link condition changes. As a result, the energy cost varies for packet transmission and reception. Therefore, it is difficult to estimate an exact energy budget for a single packet. However, assuming a standard fixed value of various parameters the energy cost for a single packet can be estimated [SAM03, KS06, VA08, ESWW00, HCB00, WN10].

4.2 Energy Efficiency Metric

Energy efficiency metric can provide us the way to determine energy efficient optimal packet size. This metric is estimated from the required energy to communicate a whole packet and payload [SAM03, ESWW00, KS06]. Equation (3.4) provides the energy per bit that can be used to estimate the required energy for a single packet and payload.

The energy per bit E_b is (see section 3.1.1)

$$\begin{aligned} E_b &= E_{tx} + E_{rx} \\ &= \frac{1}{\ell}\left((P_{tx})\frac{L}{R}\right) + \frac{1}{\ell}\left((P_{rx})\frac{L}{R}\right) \\ &= \frac{L}{\ell}\left(\frac{P_{tx}+P_{rx}}{R}\right) \\ &= \frac{L}{\ell}k_j \end{aligned} \tag{4.1}$$

Where,

$$k_j = \frac{P_{tx}+P_{rx}}{R} \tag{4.2}$$

k_j is the energy required to communicate a single bit. The values of P_{tx} and P_{rx} are estimated from section 3.1.2. The value of k_j can be changed according to the noise level of the channel. However, in several papers [SAM03, KS06, HCB00, KSPR04] the value of k_j is assumed constant to reduce complexity of energy efficiency estimation. Also, in this thesis the value of k_j is considered a fixed value. As k_j gives the combined energy to transmit and receive a single bit, the amount of energy of the payload of a single packet

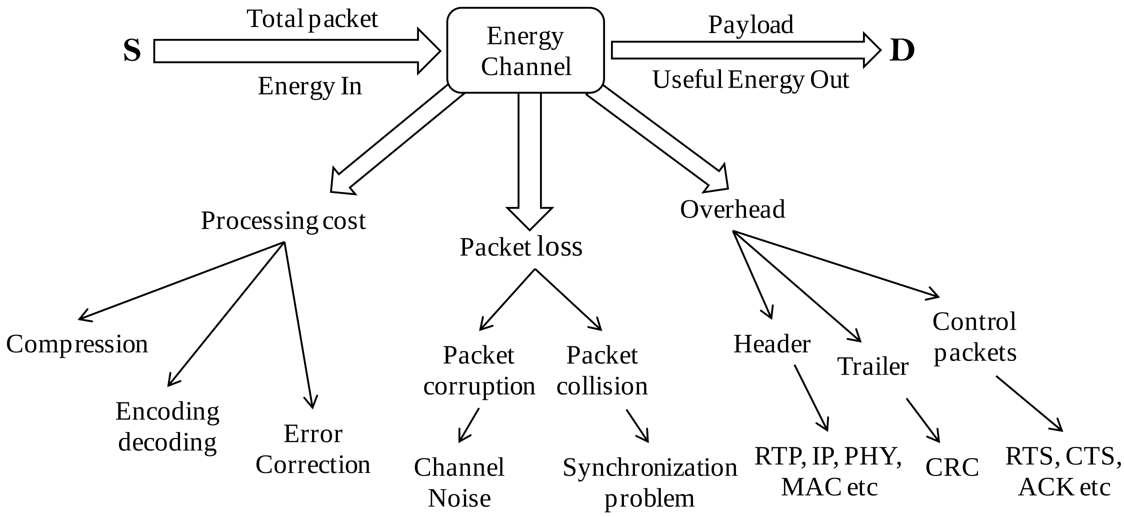

Figure 4.1: The notion of energy channel that leads us to estimate the energy efficiency

is $k_j\ell$ where the energy of a whole packet is $k_j(\ell + \alpha + \tau)$.

Based on the input energy and output energy an energy budget is drawn in figure 4.1. This figure helps understand the energy efficiency metric. In figure 4.1 the 'S' and 'D' are the source and destination nodes. In this figure the compression and error correction will be discussed in the next chapter. Packet collision, RTS, CTS overhead are explained in chapter 7. To start with a simple scenario in this chapter it is assumed the network is collision free, there is only header and trailer overhead.

According to figure 4.1 packet loss is considered as an energy wastage and the energy input for encoding decoding and control bits can not be considered as an output energy. If we assume that the channel is noise free and there is not packet loss. On the basis of figure 4.1, a suitable energy efficiency metric is defined as below [KW05, WCSY07, FMS10]

$$\begin{aligned} \eta &= \frac{k_j\ell}{k_j(\ell + \alpha + \tau)} \\ &= \frac{\ell}{(\ell + \alpha + \tau)} \end{aligned} \tag{4.3}$$

Here, the processing cost is assumed very low and it is not included in the efficiency equation (4.3). Since the control bits and processing energy cost are used only for the packet controlling and administration purpose; the energy $k_j(\alpha + \tau)$ that is used for the header and trailer (see chapter 2) can not be considered as output energy in the energy efficiency estimation.

The equation (4.3) is basically a goodput between useful data size from the application layer and whole packet size. Since, the header and trailer control portion of the packet are fixed in size, for large packet sizes the useful portion of the packet is higher which increase the energy efficiency. Figure 4.2 shows that the large packet sizes significantly increase the energy efficiency. However, if the packet size is more than the Maximum Transfer Unit (MTU) the packet will be split into two individual packets. Hence, whenever the optimal packets become bigger than the MTU it will be two separate packets with individual header and trailer. Therefore, the efficiency equation (η) for the optimal packets more than the MTU will be

$$\eta = \frac{nm + \ell}{nM + L} \tag{4.4}$$

where,

$$\begin{aligned} n &= \left\lfloor \frac{S}{M} \right\rfloor \\ M &= (m_{pay} + \alpha + \tau) \\ L &= \ell + \alpha + \tau \\ &= (S - Mn) + \alpha + \tau \end{aligned} \tag{4.5}$$

where m denotes the payload of MTU packet size, ℓ is the payload of fragmented packet, n is the number of MTU size packets and L is the packet size generated from the fragmented parts. S is the total packet size without fragmentation, M is the MTU packet size. According to IEEE 802.11 MAC protocol MTU size is 1500 bytes but to keep the computational complexity lower MTU is considered 1000 bytes [CG06b]. From figure 4.2 it is found that the MTU packet size is the most efficient packet size. Since,

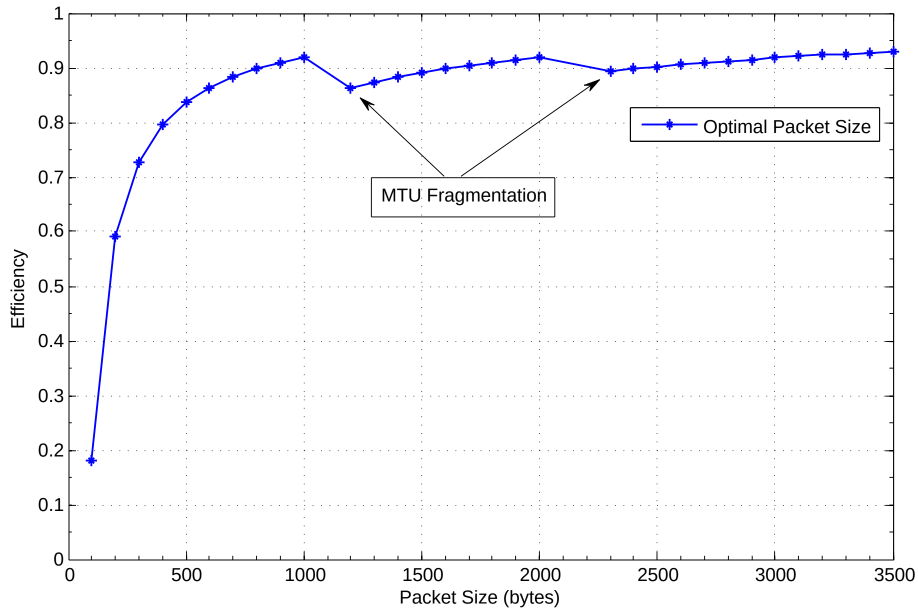

Figure 4.2: Optimal packet size estimation and efficiency measurements in a noiseless channel

energy efficiency metric equation (4.3) is a energy ratio between energy cost of payload and whole packet size. In figure 4.2 the efficiency rises very sharply with the packet size but after that it becomes steady and reaches a maximum and, whenever the packets become bigger than MTU, the packet is split in to two and due to the high overhead the efficiency drops. If the packet is increased in size even further the energy efficiency increases until the second MTU split size. Due to overhead of the headers, the bigger packets cannot achieve the maximum efficiency in a noise free channel.

4.2.1 Energy Efficiency Metric in Noisy Channel

Channel noise has a great influence over the estimation of the optimal packet size. In the last chapter the relationship between the packet size and channel noise is shown. In the noise free channel, bigger packet are more efficient than the smaller size packets. But bigger packets are more error prone in a noisy environment. Due to noise in the channel the bits inside the packets are corrupted and corruption rate is higher for bigger packets than smaller packets. The relationship between BER, PER and transmitted packet sizes

(L) are shown in the last chapter.

However, the MTU packet split is not considered in our previous discussion. If the packet size become more than MTU then the packet is split into multiple packets. Since, the fragmented parts need new header and trailer, the overhead bytes at least double. Hence, packet error rate of fragmented packets will be different than equation (3.1). For the MTU size packet there will be an error rate and for the fragmented part packet size there will be another error rate. If the fragmented parts of the packet or any parts of the packet is corrupted, the whole packet is considered as a corrupted packet. The PER of bigger packets more than the MTU (per_{mtu}) can be expressed as

$$\begin{aligned} per_{mtu} &= \Big(1-(1-ber)^{(\alpha+m+\tau)}\Big)^n \times \Big(1-(1-ber)^{(\alpha+\ell+\tau)}\Big) \\ &= \big(1-(1-ber)^M\big)^n \times \big(1-(1-ber)^L\big) \end{aligned} \tag{4.6}$$

Considering packets less than the MTU, the output energy will be $k_j\ell(1-per)$. $(1-per)$ determines the packet acceptance rate or reliability of the channel. Thus, $k_j\ell(1-per)$ estimates the total output energy of non-corrupted information bits. Similar to equation (4.3) the energy efficiency (η) for the noisy channel can be expressed as

$$\begin{aligned} \eta &= \frac{\ell}{(\ell+\alpha)}(1-per) \\ &= \frac{\ell}{(\ell+\alpha)} \times (1-ber)^{(\ell+\alpha+\tau)} \end{aligned} \tag{4.7}$$

The efficiency equation for the optimal packet size larger than the MTU (η_{mtu}) will be

$$\eta_{mtu} = \frac{n \times m + \ell}{(n \times M + L)} \times (1-per_{mtu}) \tag{4.8}$$

Whenever the channel remains in good condition, the PER becomes very low (see figure 4.4) and the optimal packet size becomes higher than the MTU size. Equation (4.8) is used for efficiency estimation for the bigger packet sizes larger than the MTU. But for smaller than MTU, equation (4.7) will be applied to estimate the efficiency and optimal

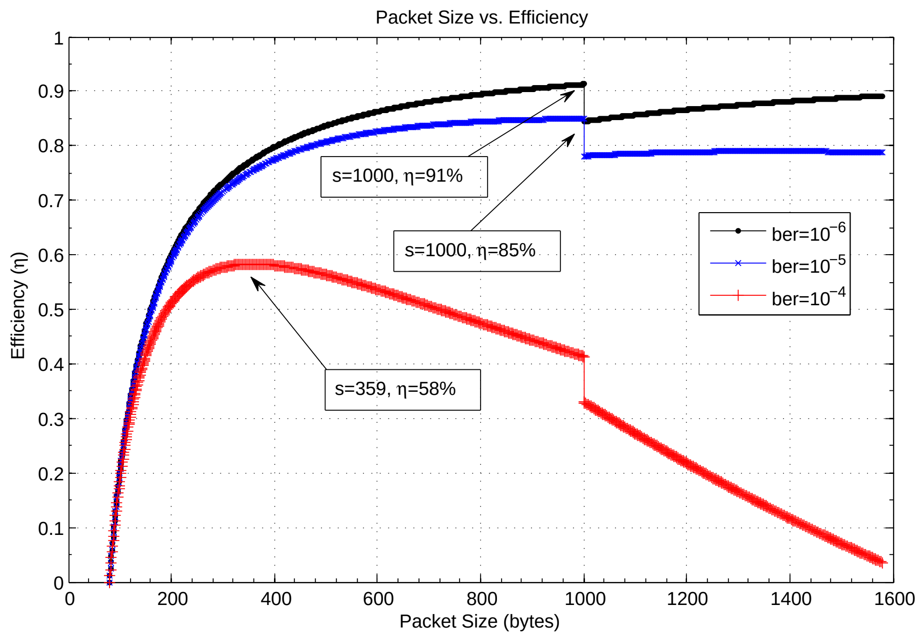

Figure 4.3: Optimal packet size estimation in a noisy channel for three different BER

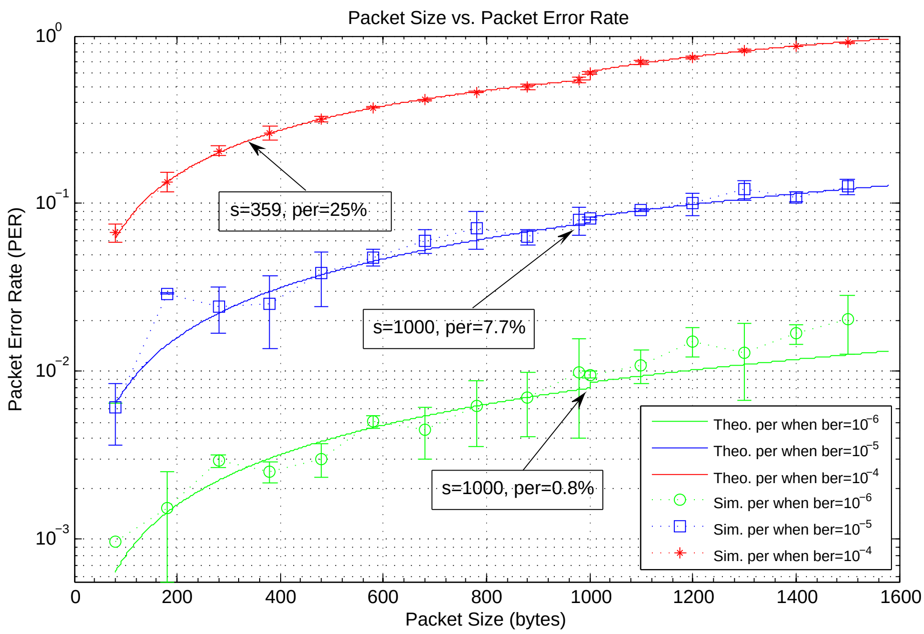

Figure 4.4: Packet error rates of the optimal packet sizes in a noisy channel for three different BER

packet size. By using equation (4.7), (4.8) and (4.6) two graphs are plotted in figure 4.3 and 4.4 respectively. The figure 4.3 shows the efficiency against packet size and the PER is shown for the corresponding optimal packet sizes in figure 4.4. The most energy efficient optimal packet sizes are shown in the text box denoted as $'s'$ and pointed with arrows. The text box specifies the corresponding energy efficiency and PER for the optimal packet size. The results of figure 4.3 are matched with the results of [SAM03, KS06]. Hence, the calculated values are correct for this figure.

To validate the results with the simulation, a simple topology is created in NS-2 simulator. Only two nodes are in the network. One node has one poisson source and another node act as a base station or sink node. The channel noise is fixed. The source node sends packets and the receiver node receives the packets. Transmission power, reception power, data rate are set according to the estimation given in section 3.1.2. Simple data broadcast is considered in the simulation for data transmission. There is not overhead for the control packets of data routing. Simulation is done for 3600 seconds. More details of the simulation procedure are described in chapter 8. From the simulation efficiency cannot be estimated directly. The efficiency is evaluated from the parameters.

Figure 4.4 shows the comparison between theoretical and simulated PER. From the simulation, it is found that the simulation results matches with the theoretical model results. Figure 4.4 shows that for high PER efficiency is lower. Because high BER increases the PER and corrupted packets are either discarded in the destination end or lost. Hence, the input energy for the packet is completely wasted and the efficiency goes down. Unlike the efficiency reported in figure 4.2, figure 4.3 shows that for a noisy channel the efficiency level reaches a maximum. After that it decreases as the packet size increases. The reason is, in a noisy channel the PER remains small for the smaller packets but packet overhead remains high and due to high overhead the efficiency becomes lower. Whereas for the bigger packets the overhead remains small but due to high packet loss rate the efficiency becomes low.

Energy efficient optimal packet size is found from figure 4.3 for three different channel condition. However, this estimation is done for a stable channel condition considering a fixed BER, whereas wireless channel condition might change with time. In the next section, energy efficient packet size optimization for time varying channel is estimated.

4.3 Energy Efficient Packet Size Estimation for Variable Noise Channel Condition

The link condition of an ad hoc network might change or fluctuates with time. Gilbert-Elliot error model can simulate time varying characteristics of an error prone channel. To estimate an average packet loss rate Gilbert-Elliot error model is a renowned method. Average packet loss rate estimation methodology for any time varying channel condition is shown in the last chapter. In this section only energy efficient estimation is shown for a time varying noisy channel for the Gilbert-Elliot model.

4.3.1 Gilbert-Elliot Error Model and Energy Efficient Optimal Packet Size

The average packet loss rate (P_{GE}) is estimated in equation (3.14). From the average packet error probability P_{GE} the reliability of the Gilbert-Elliot channel (r_{GE}) is estimated which is denoted as $(1-P_{GE})$. After applying this reliability the energy efficiency (η_{GE}) for the two nodes topology will be

$$\begin{aligned} \eta_{GE} &= \frac{k\ell \times r_{GE}}{k(\ell+\alpha+\tau)} \\ &= \frac{\ell \times r_{GE}}{(\ell+\alpha+\tau)} \end{aligned} \tag{4.9}$$

From this equation (4.9) for different values of P_{bg} and P_{gb} state changing probability efficiency is estimated and shown in figure 4.5. In figure 4.5 every single point shows the optimal packet size and energy efficiency for different values of P_{bg} and P_{gb}.

To compare the theoretical packet loss model with the simulated results two state Markov error model of NS-2 is used. The topology and other setting is used similar to section 4.2.1. Rather comparing all the theoretical values stated for the P_{gb} and P_{bg} in figure 4.5 only one situation is considered for the simulation to save simulation time.

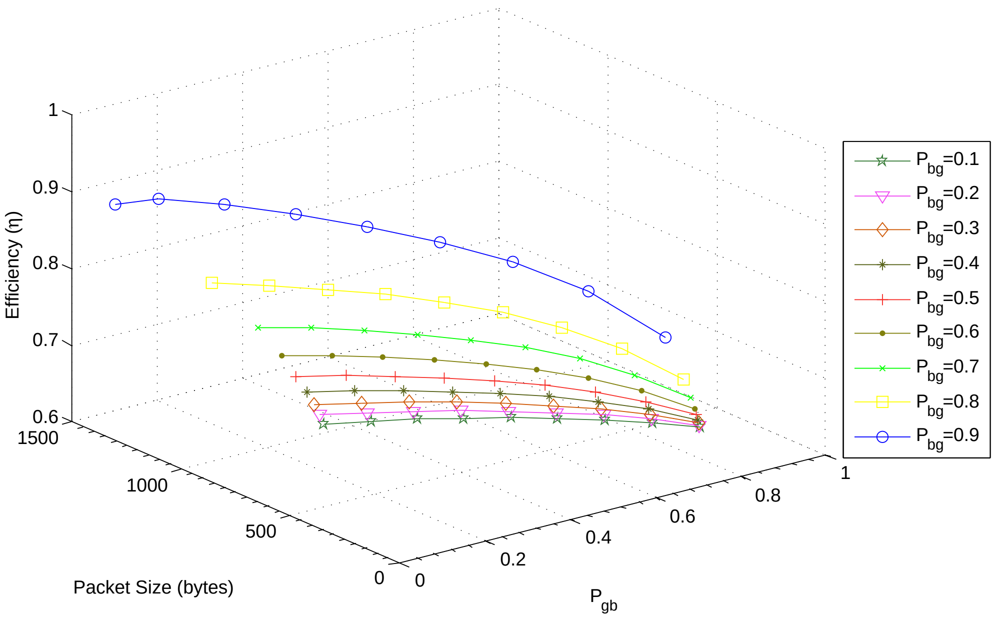

Figure 4.5: Maximum efficiency estimation for a time varying channel for different state changing values

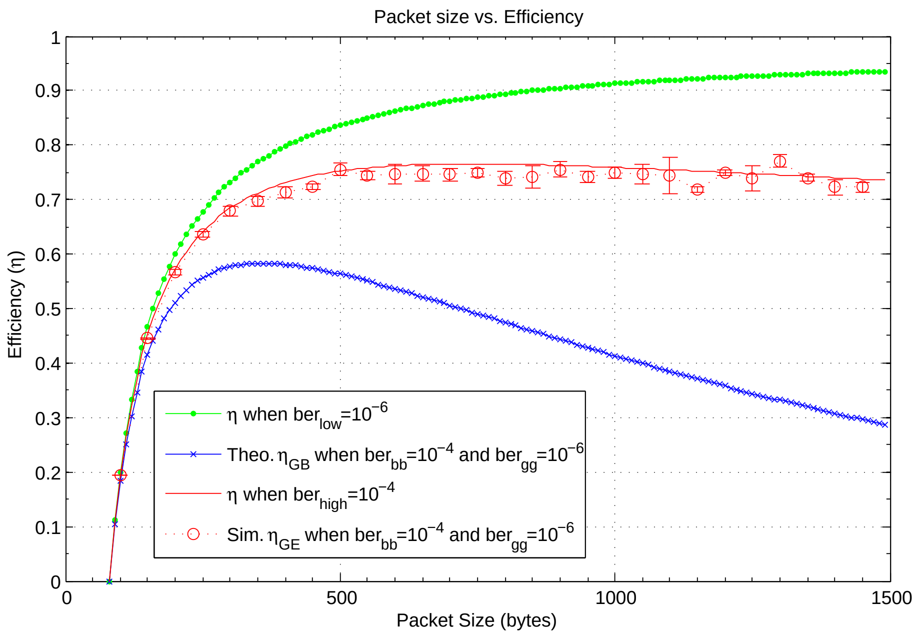

Figure 4.6: Energy efficient packet size comparison of Gilbert-Elliot channel and normal channel for two nodes topology

The P_{gb} and P_{bg} values are set 0.4 and 0.9 considering that the channel stays 60% time remains in the good state and 10% time remains in the bad state. Theoretically the most efficient optimal packet size is 1000 bytes for P_{gb} and P_{bg} values of 0.4 and 0.9 (shown in figure 4.5). For a fixed values of P_{bg} and P_{gb} efficiency is shown in figure 4.6. It is found that the simulated results matches with the theoretical results.

The optimal packet size from the Gilbert-Elliot model provides an average estimation. If the ad hoc node does not have the optimal packet size adaptation mechanism. In that case fixed packet size needs to be set by the network administrator. Hence, depending on the scenario and channel experience the packet size needs to be chosen. But if the channel condition is unpredictable or changes very frequently or out of control in that case the theoretical model of optimal packet size estimation will not give a good solution.

4.3.2 Dynamic Energy Efficient Optimal Packet Size and Time Varying Channel

In the time varying noise channel condition the PER does not remains same all the time. Hence, the optimal packet size can be estimated theoretically but it might not be the best for a particular time period. In the time varying channel sometimes the energy efficiency can be overestimated or underestimated. For that reason there some researchers have devised suitable techniques to optimize the packet size at runtime [WXC$^+$10, CG06a]. But a dynamic packet size adaptation policy consumes more energy. Because to run the optimal packet size estimation algorithm at runtime and adapt the best packet size cost some energy. In [KHZ11, WXC$^+$10, CG06a] dynamic packet size optimization technique have been discussed.

In [KHZ11] the research focused on ad hoc wireless LAN, where a packet size adaptation algorithm is proposed to increase throughput and reduce packet loss. The WLAN nodes observe the channel at specific time intervals and then adopt the optimal packet size. Payload size adaptation is also illustrated in [CG06a] where a theoretical framework has been given to achieve the maximum throughput by adopting optimal packet size at runtime. However, their analysis did not consider the energy efficiency and real time implementation is not shown. In [WXC+10] packet size optimization has been done dynamically. To estimate the link condition or channel noise 2 bytes additional overhead is included in each packets. Their estimation includes aggregation and fragmentation services for smaller and bigger packet sizes and most importantly the optimization algorithm is implemented in a ad hoc sensor node operating system test bed. To estimate the link quality the packet reception rate is estimated. However, what type of computation power is needed to adopt the optimization algorithm and how much delay will be added is not shown. Since energy efficient packet size optimization is the major concern of this thesis; before adopting the dynamic packet size algorithm, the optimal packet size is investigated for different scenarios. To adopt the optimized packet size dynamically the algorithm needs to consider all the parameters and possible common scenarios that has an effect on the optimization.

4.4 Energy Efficient Packet Size for Multi-hop Network

The ad hoc nodes does not have high transmission range. To cover long distance communication the nodes uses the neighbor nodes as a router and forward the data to the destination. The application scenario for the multi-hop topology is assumed border line area monitoring. Large number of nodes will be placed in the border line and made a linear chain like multi-hop topology to detect the intruders. If the nodes detect anything (intruder) around within its' range, an alarm message (packets) will be send to the base station via multi-hop linear chain and border guard will be informed.

Here, one important observation is the coherence time of connectivity of multiple number of hops is ignored. Since the application is considered as a delay tolerant application. It is fair to assume that the connection time and overall delay or transmission is negligible [KSPR04, SPRKH05]. However, the coherence time for the delay sensitive real-time application is vital and it is considered in our chapter 7 for real-time application VoIP.

Whenever, a packet traverse multiple nodes, if the packet gets corrupted at the first hop, only the energy to send the packet from a source to a specific node is lost. But when the packet is corrupted after few more hops, much more energy can be wasted. That's why the importance of a energy efficient optimized packet in a multi-hop network is paramount.

According to [KSPR04, SPRKH05] the energy required for a multi-hop equally spaced chain like topology (see previous chapter section 3.3) is given by

$$\begin{aligned} E_{in}^{mul} &= L \times (nE_t + (n-1)E_r) \\ E_{out}^{mul} &= \ell(nE_t + (n-1)E_r) \times r^n \end{aligned} \tag{4.10}$$

where E_{in}^{mul} and E_{out}^{mul} are the energy required for the multi-hop chain topology. n is the number of nodes and E_t and E_r represents

$$E_t = \left(\frac{P_{tx}}{R}\right) \; and \; E_r = \left(\frac{P_{rx}}{R}\right) \tag{4.11}$$

Whenever packets are transmitted from source node to the destination node via multiple hops, packet error rates (PER) become different than the single hop PER (see equation 3.1 of the last chapter).

From equation (4.10) and (3.15) the energy efficiency equation for the multi-hop linear chain topology is illustrated as below

$$\begin{aligned} \eta &= \frac{\ell(nk - E_r) \times (1 - ber)^{LN}}{L \times (nk - E_r)} \\ &= \frac{\ell \times (1 - ber)^{LN}}{(\alpha + \ell + \tau)} \end{aligned} \tag{4.12}$$

By using the equations (4.12) energy efficiency is plotted for different packet sizes in figure 4.7. This figure shows that in a multi-hop network with fixed noise when the number of hops increases energy efficiency level decreases.

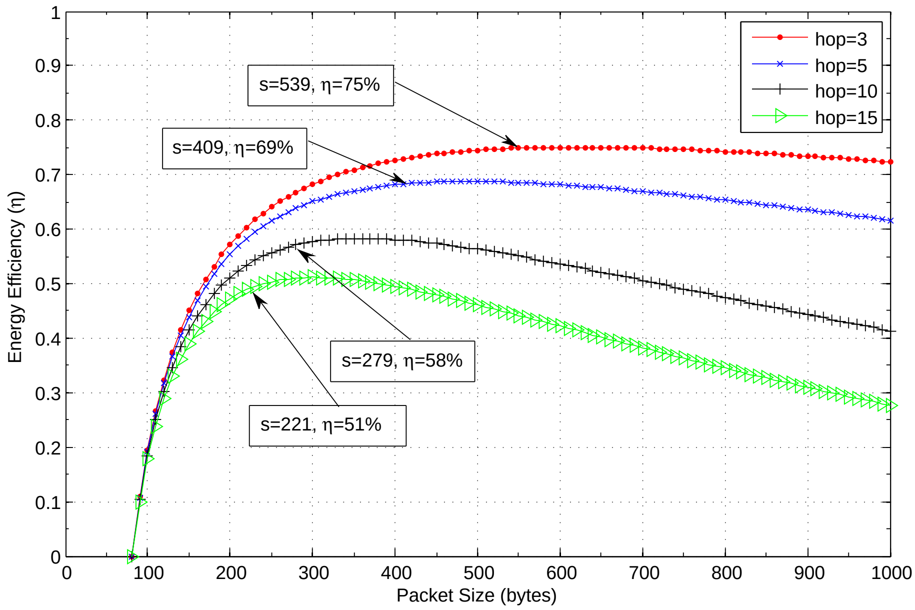

Figure 4.7: The efficiency analysis for a multi-hop chain like topology when $BER = 10^{-5}$

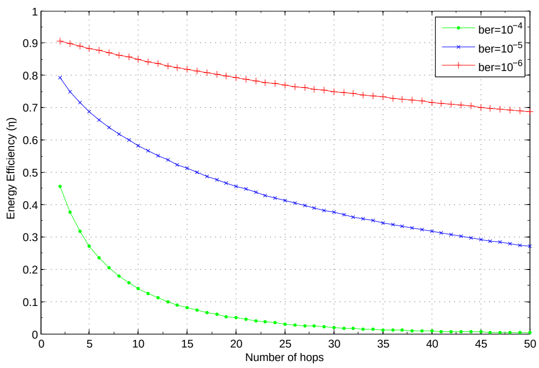

Figure 4.8: Energy efficiency analysis for a multi-hop chain like topology in three different channel condition

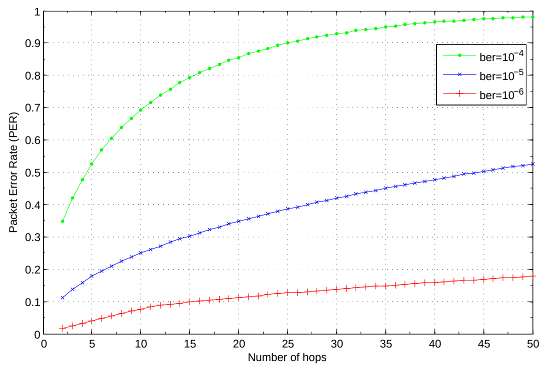

Figure 4.9: PER analysis of a multi-hop chain like topology in three different channel condition

This phenomenon is shown in figure 4.8. In this figure energy efficiency is estimated against various number of hops for three different BER. Figure 4.8 shows the maximum energy efficiency for optimal packet size for different number of hops and the corresponding PER is shown in figure 4.9.

Whenever the PER becomes low in a good channel condition the optimal packet size becomes more than the MTU packet size (see equation (4.6)). In that case the PER (per_{mtu}) for the optimal packet size more than the MTU for the multi-hop linear topology can be expressed as below

$$
\begin{aligned}
per_{mtu} &= \left(\left(1-(1-ber)^{(\alpha+m+\tau)}\right)^{n}\right)^{N} \times \left(1-(1-ber)^{(\alpha+\ell+\tau)}\right)^{N} \\
&= \left(1-(1-ber)^{nM}\right)^{N} \times \left((1-(1-ber))^{L}\right)^{N} \qquad (4.13)
\end{aligned}
$$

Where, N is the number of hops in the multiple hop linear chain topology network.

4.4.1 Energy Efficient Packet Size for Multi-hop Network with Different Link Condition

The distance between the nodes might not be same all the time in the linear chain like multi-hop topology. Also, the channel condition might be different among different links. Hence, there are three possibilities

- The link condition is the same between all the nodes (BER is same) and channel condition remains fixed. For example, in a three hops scenario, BER is 10^{-5} in each link.
- BER is different between the links but BER level remains fixed. For example, in a three hops scenario, the BER is 10^{-4}, 10^{-5} and 10^{-6} between the links.
- The link condition changes with time (BER level fluctuates) and also BER between the nodes are different. For the same scenario, in a three hops topology network BER fluctuates between the hops. For an instance BER changes between 10^{-4} to 10^{-6} from first hop node1 to node2. For the second hop it changes between 10^{-5} to 10^{-6} and so on.

The first situation is shown in the previous section. For the second case if the channel condition change with time then there will be different BER in different link. Therefore, if different PER is known between the links then the average packet error rate per_{multi}

for the multi-hop network can be estimated as below

$$per_{multi} = per_1 \times per_2 \times .... \times per_{N-1} \tag{4.14}$$

where, $per_{1,2,..,N}$ are the different PER between the links.

For the third scenario Markov error model is considered. The linear chain of hops is considered as a Markov chain where the error rates changes randomly between different links.

For an instance, assume that the BER between node1 and node2 of figure 3.4 changes with time with certain probability. The BER between node2 and node3 also have different BER and changes with time with certain probability. Only the difference with the Gilbert-Elliot model is, in Gilbert-Elliot model there are only two states and two state Markov chain is used. The channel condition changes between only two states. But multi state Markov chain will provide an estimation of PER for a time varying channel condition of multiple channel condition.

A four state Markov model has been shown in [TJKJ00] for a satellite channel. Whenever the Markov chain become larger than two state (see figure 3.1) the mathematical model become complex and large. Hence, the theoretical model is not shown here.

To estimate the energy efficiency the PER of equation (4.14) or PER of the Markov chain will be included in equation (4.12) and energy efficiency can be estimated.

4.4.2 Transmitting Only Header Packets in the Multi-hop Topology

In the multi-hop linear topology the increasing number of hops reduces the energy efficiency and also the optimal packet size decreases. Moreover, the PER becomes higher as the packets travels through more number of hops. Hence, to keep the PER low and travel many hops error correction has been used in some papers [KSPR04] which is discussed in our next chapter. We have analysed an distinct approach transmitting only header

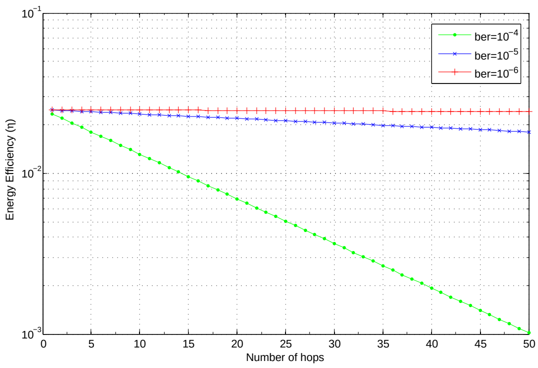

Figure 4.10: Energy efficiency estimation while traversing only header packets in a multi-hop linear chain network

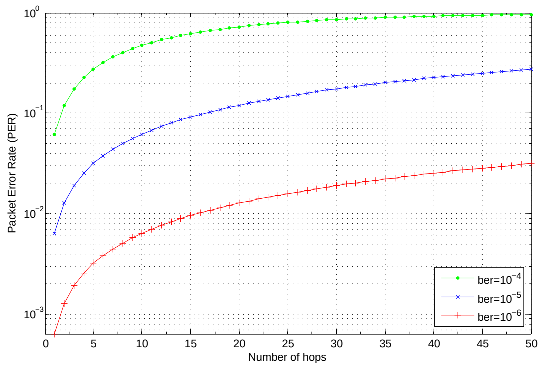

Figure 4.11: PER estimation with only header packets in a multi-hop linear topology network

packets to traverse large multi-hop linear chain network.

Since, smaller packets does not have high PER, therefore if lowest PER is given the highest priority, then by using small packets a large linear chain multi-hop topology can be traversed. However, the overhead will be very high, and as a consequence energy efficiency will be low.

The smallest header is 8 bytes UDP. Hence, the UDP header can be used as a header packet. But, if the packet size does not have any payload ($\ell = 0$) according to the energy efficiency estimation metric (see section 4.2) the energy efficiency becomes zero. Since the energy efficiency is estimated from the goodput of total input energy for the whole packet and payload is considered as useful output energy. That is why to estimate the energy efficiency for the UDP header (which is 8 bytes) 6 bytes are considered as useful output information. Because among the whole (UDP) header 4 bytes provide the destination and the source address and another 2 bytes provide the length of the header size.

According to the estimation given in this section only header packet is used as an input energy in equation (4.12) and 2 bytes are used as an useful output energy. In figure 4.10 and 4.11 the energy efficiency and PER is estimated against different number of hops. From these two figures it is found that with only header, the energy efficiency is low but PER also remains lower. Specifically, for multi-hop topology network while transmitting header packets the PER becomes lower than transmitting whole packets. For an instance, the PER for the 5 hops linear chain topology is 27.4% for 10^{-4} BER whereas for the whole packet it is 57%.

4.5 Conclusion

In this chapter energy efficient optimal packet size is described for different ad hoc network scenarios. The impact of PER, MTU, multi-hop and variable noise are investigated on optimal packet size. It is found that the MTU packet split has vital impact on energy efficiency and PER of the network. In the noisy channel smaller packet sizes are more energy efficient whereas in the good channel condition MTU is revealed as most energy efficient packet size. Next chapter will describe how the PER can be minimized with error correction and overhead can be reduced by compression.

References

[CG06a] S. Choudhury and J.D. Gibson. Payload length and rate adaptation for throughput optimization in wireless LANs. In *Vehicular Technology Conference, 2006. VTC 2006-Spring. IEEE 63rd*, volume 5, pages 2444 –2448, may 2006.

[CG06b] Sayantan Choudhury and Jerry D Gibson. Payload length and rate adaptation for throughput optimization in wireless lans. In *Vehicular Technology Conference, 2006. VTC 2006-Spring. IEEE 63rd*, volume 5, pages 2444–2448. IEEE, 2006.

[ESWW00] J.P. Ebert, B. Stremmel, E. Wiederhold, and A. Wolisz. An energy-efficient power control approach for WLANs. 2000.

[FMS10] T. Frantti, M. Majanen, and T. Sukuvaara. Delay based packet size control in wireless local area networks. In *Ubiquitous and Future Networks (ICUFN), 2010 Second International Conference on*, pages 332 –337, 2010.

[HCB00] W.R. Heinzelman, A. Chandrakasan, and H. Balakrishnan. Energy-efficient communication protocol for wireless microsensor networks. In *Proceedings of the 33rd Hawaii International Conference on System Sciences-Volume 8 - Volume 8*, HICSS '00, pages 8020–, Washington, DC, USA, 2000. IEEE Computer Society.

[KHZ11] M.N. Krishnan, E. Haghani, and A. Zakhor. Packet length adaptation in WLANs with hidden nodes and Time-Varying channels. In *Global Telecommunications Conference (GLOBECOM 2011), 2011 IEEE*, pages 1 –6, dec. 2011.

[KS06] Z.H. Kashani and M. Shiva. BCH coding and multi-hop communication in wireless sensor networks. In *Wireless and Optical Communications Networks, 2006 IFIP International Conference on*, pages 5 pp. –5, 0-0 2006.

[KSPR04] H. Karvonen, Z. Shelby, and C. Pomalaza-Raez. Coding for energy efficient wireless embedded networks. In *Wireless Ad-Hoc Networks, 2004 International Workshop on*, 2004.

[KW05] J. Korhonen and Y. Wang. Effect of packet size on loss rate and delay in wireless links. In *Wireless Communications and Networking Conference, 2005 IEEE*, volume 3, pages 1608 – 1613 Vol. 3, 2005.

[SAM03] Y. Sankarasubramaniam, I.F. Akyildiz, and S.W. McLaughlin. Energy efficiency based packet size optimization in wireless sensor networks. In *Sensor Network Protocols and Applications, 2003. Proceedings of the First IEEE. 2003 IEEE International Workshop on*, pages 1 – 8, 2003.

[SPRKH05] Z. Shelby, C. Pomalaza-Rez, H. Karvonen, and J. Haapola. Energy optimization in multihop wireless embedded and sensor networks. *International Journal of Wireless Information Networks*, 12:11–21, 2005.

[TJKJ00] T. Tao, L. Jianhua, G. Ke, and G. Jun. A four-states Markov model for burst error analysis in satellite communications. In *Communication Technology Proceedings, 2000. WCC - ICCT 2000. International Conference on*, volume 1, pages 930–934 vol.1, 2000.

[VA08] M.C. Vuran and I.F. Akyildiz. Cross-layer packet size optimization for wireless terrestrial, underwater, and underground sensor networks. In *INFOCOM 2008. The 27th Conference on Computer Communications. IEEE*, pages 226 –230, 2008.

[WCSY07] D. Wu, Song Ci, H. Sharif, and Yang Yang. Packet size optimization for goodput enhancement of multi-rate wireless networks. In *Wireless Communications and Networking Conference, 2007.WCNC 2007. IEEE*, pages 3575 –3580, 2007.

[WN10] S. Wang and J. Nie. Energy efficiency optimization of cooperative communication in wireless sensor networks. *EURASIP J. Wirel. Commun. Netw.*, 2010:3:1–3:8, January 2010.

[WXC+10] D. Wei, L. Xue, C. Chun, H. Yuan, C. Gong, L. Yunhao, and B. Jiajun. DPLC: Dynamic Packet Length Control in wireless sensor networks. In *INFOCOM, 2010 Proceedings IEEE*, pages 1 –9, march 2010.

Chapter 5

Optimal Packet Size Estimation with FEC and Compression

In this chapter energy efficiency of the optimal packet size is estimated when using forward error correction code (FEC) and compression techniques. These two methods are often used, to reduce the PER and overhead. Since, it is observed from our analysis in the previous chapter that PER and overhead decreases the energy efficiency of optimal packet size, in this chapter these two methods are applied to improve efficiency. We will investigate the impact of FEC and compression on the energy efficiency and the optimal packet size.

5.1 Error Correction and Energy Efficient Packet size

In a noisy channel large numbers of packets can become corrupted and hence lost. Therefore, energy is wasted and efficiency level decreases. In order to improve the reliability of channel and reduce packet error rate, two familiar approaches are used in general Automatic Repeat Request (ARQ) and Forward Error Correction (FEC). FEC and ARQ are described in chapter 3. In this chapter efficiency is estimated for these two methods.

5.1.1 Efficiency Estimation of Automatic Repeat Request (ARQ)

Whenever any packet is lost or become corrupted the ARQ protocol of the destination node requests the source node to retransmit the packet again. As a result, the overall energy cost for packet communication doubles. Considering the energy efficiency (η_{re}) for the retransmission technique yield as [SAM03, KS06]

$$\begin{aligned} \eta_{re} &= \frac{k_j \ell(1-per)}{num_{ret} k_j(\alpha+\ell+\tau)} \\ &= \frac{\ell(1-per)}{num_{ret}(\alpha+\ell+\tau)} \end{aligned} \tag{5.1}$$

where num_{ret} is the number of retransmission for a single packet. If the requested packet is reached at the destination node correctly without any error in first attempt then num_{ret} is 1. Hence, it can be written as

$$\eta_{re} \leq \eta \tag{5.2}$$

Where η is the energy efficiency without retransmission (see chapter 4 section 4.2.

Thus, ARQ is not an energy efficient method. It can be concluded as retransmission technique improves the reliability but doesn't improve the efficiency. ARQ method could be customized (e.g. Hybrid ARQ) and further research can be done in this area. Since, ARQ is revealed as an energy inefficient method it is not considered for further analysis.

5.1.2 Energy Efficiency of the Forward Error Correction Code (FEC)

Forward Error Correction Code (FEC) is used to retrieve the lost information and correct the corrupted bits of the packets. It reduces packet error rates and increases channel reliability. However, it adds extra energy cost (E_{dec}) for encoding decoding and error correcting overhead bytes (τ) to correct errors (see chapter 3).

The reliability of the channel with the FEC method is given by [SAM03, KS06, VA06, ZDQ08].

$$r = 1 - per = \sum_{i=0}^{t} \begin{pmatrix} L \\ i \end{pmatrix} ber^i (1 - ber)^{L-i} \tag{5.3}$$

where t is the maximum error correction strength that represents how many numbers of bit errors FEC can correct. Based on the error correcting capability (t) the value of τ and the decoding cost E_{dec} are changed accordingly (see equation (2.7) in chapter 2.3). Considering the error correcting code, recalling the energy efficiency equation (4.7), η can be rewritten as below

$$\eta = \frac{k\ell}{k(\ell + \alpha + \tau) + E_{dec}} \times \sum_{i=0}^{t} \begin{pmatrix} L \\ i \end{pmatrix} ber^i (1 - ber)^{L-i} \tag{5.4}$$

Since according to FEC constrains (see section 2.4.1 of chapter 2.3), to use the error correction code packet size need to be $L = 2^m - 1$. On this account, in figure 5.1 and 5.2 the packet sizes are defined for different values of m and afterwards the energy efficiency is estimated for different error correction strength. In figure 5.1 the most energy efficient optimal packet sizes are pointed with arrows for given BER. In the chart 's' is the highest energy efficient packet size.

Figure 5.1 shows that the value of error correction strength increases the energy efficiency of bigger packet sizes but for smaller packets it is not an efficient approach. It is possible to achieve even higher efficiency without the FEC for packets less than 1000 bytes.

For the higher values of error correction strength t, the efficiency becomes lower and steady. The reason of lower efficiency is the increasing cost of decoding. For the packet sizes more than 500 bytes efficiency becomes steady and efficiency decreases very slowly. Because error correction strength reduces the PER and retain the similar value of reliability for the larger packet sizes. Only the increasing cost of decoding decreases the efficiency very slowly.

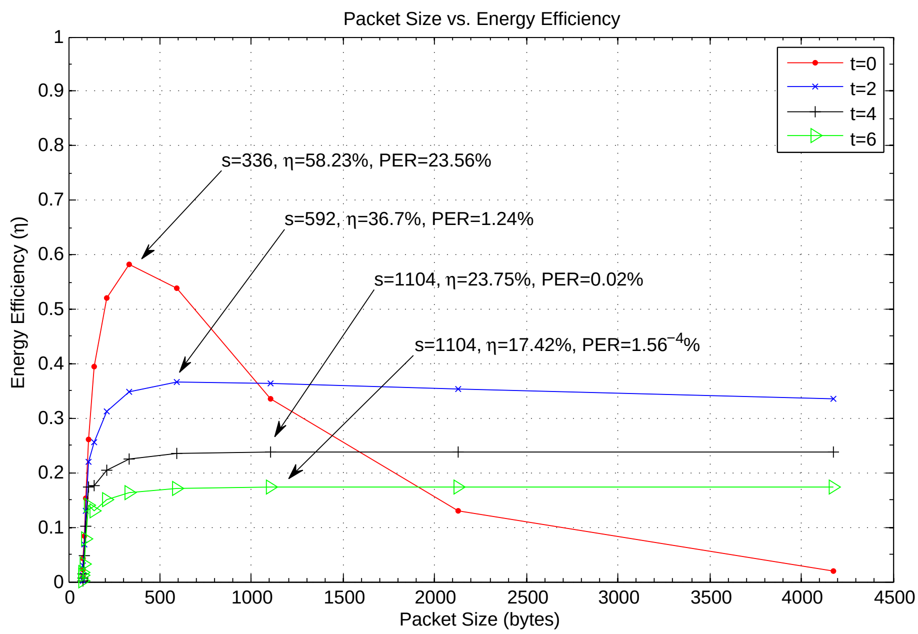

Figure 5.1: Energy efficiency estimation with FEC with different error correction strength for BER=10^{-4}

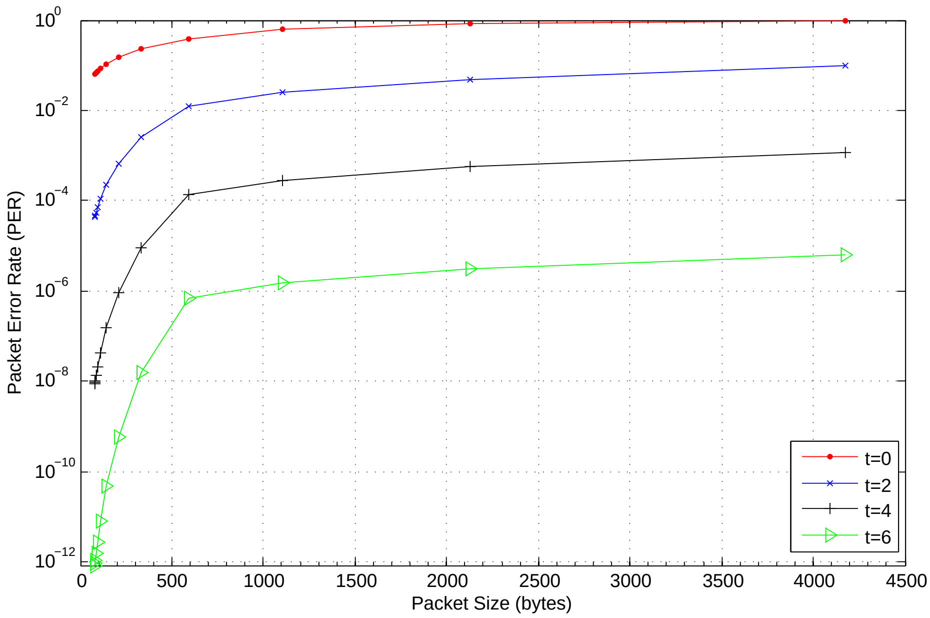

Figure 5.2: PER estimation with FEC for different error correction strength when BER=10^{-4}

The corresponding PER for the packet size is given in figure 5.2 which shows that FEC reduces the PER significantly. From figure 5.1 and 5.2 it is revealed that if FEC is used large packet size can be used with high efficiency and lower PER.

5.1.3 Energy Efficient Packet Size Estimation with FEC for Packets Bigger than MTU

With the FEC, bigger packets can be transmitted with lower PER. But whenever a packet becomes bigger than the MTU size, it is split in to multiple packets.

The payload size for the FEC code follows the constrains of $(2^m - 1)$. If 100 bytes is considered as MTU size, whenever the value of m become more than 12, the splitting occurs. Since, the payload size is a power function, it creates multiple number of same size packets. For instance, if the value of m is 14 and payload size is $(2^{14} - 1)$ which is 2047 bytes. For 2047 bytes payload there will be 4 individual packet of 511 $(2^{12} - 1)$ bytes each.

The reliability function for the MTU split packets for the FEC will be

$$r_{mtu} = 1 - per = \left(\sum_{i=0}^{t} \begin{pmatrix} L_{max} \\ i \end{pmatrix} ber^i (1 - ber)^{L_{max} - i} \right)^{mtu} \tag{5.5}$$

where L_{max} is the maximum transferable bytes which is $(2^{12} - 1)$ and the header and trailer bytes with the FEC considering the MTU. mtu is the number of split packets. Hence,

$$mtu = \left\lceil \frac{L}{L_{max}} \right\rceil \tag{5.6}$$

The energy efficiency equation with the FEC for the packet sizes more than MTU is expressed as

$$\begin{aligned} \eta &= \frac{mtu \times k\ell}{mtu \times k(\ell + \alpha + \tau) + mtu \times E_{dec}} \times r_{mtu} \\ &= \frac{k\ell}{k(\ell + \alpha + \tau) + E_{dec}} \times r_{mtu} \end{aligned} \tag{5.7}$$

The number of payload and packet on the numerator and the denominator in equation (5.7) are same. Hence, the energy efficiency function remains the same like as

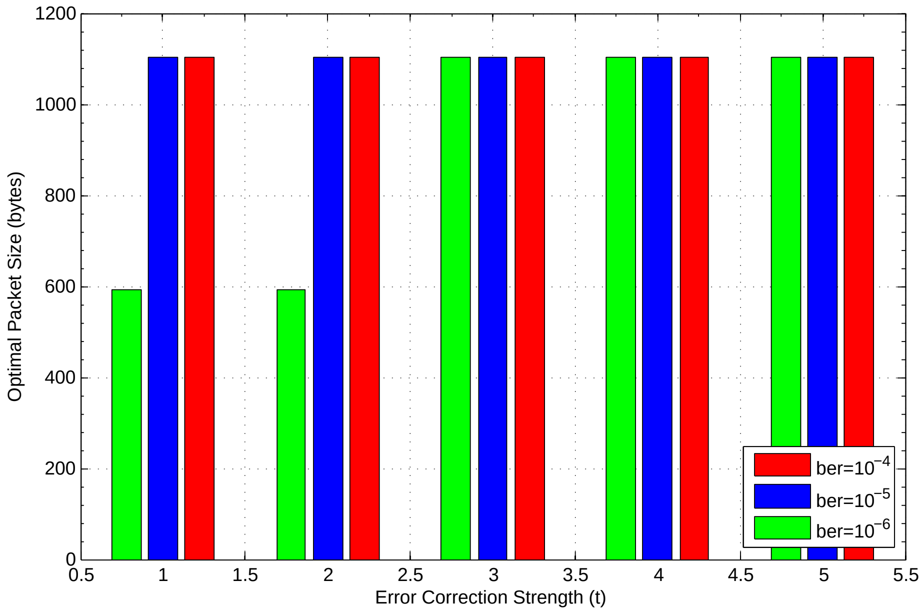

Figure 5.3: Optimal packet size estimation with different error correction strength

equation (5.4) for the FEC for the bigger packets more than MTU. Only difference is the reliability function r_{mtu} which is stated in equation (5.5).

5.1.4 Energy Efficient Packet Size with High Strength FEC

Lets assume that FEC can able to correct all the error bits within a packet according to the error correction strength t. Hence, by using FEC, bigger packets can be transmitted to the destination without any error.

Since, according to the FEC constrains, the payload size needs to be $2^m - 1$, if the value of m becomes more than 12 then the payload size will be more than the MTU and split in to multiple packets. Hence, after using the FEC the maximum payload size would be 511 ($2^{12} - 1$) bytes. Therefore, if error correction strength (t) increases continuously more overhead of the FEC will be added and the efficiency level goes down. Because as the error correction strength increases, it adds more overhead and extra decoding cost (see section 2.4.1 of chapter 2.3).

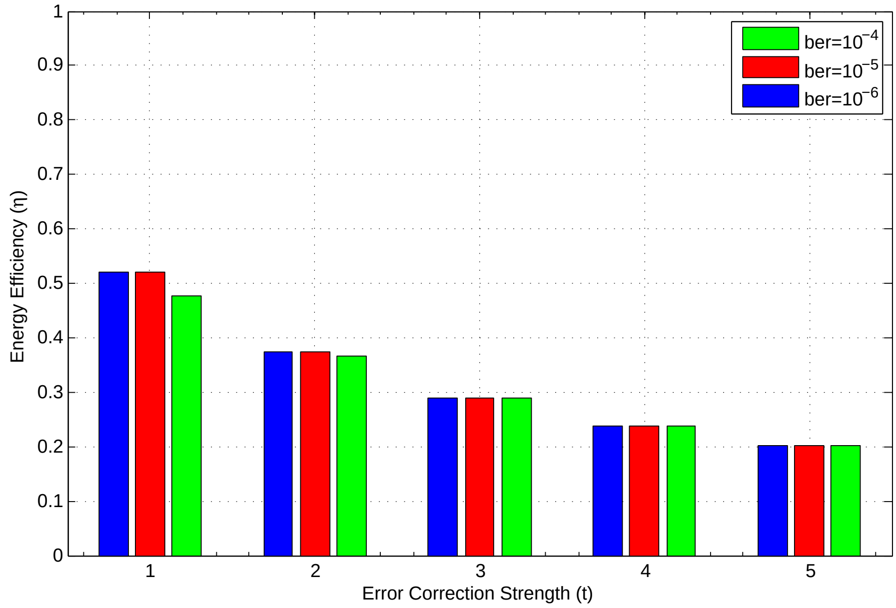

Figure 5.4: Maximum energy efficient error correction strength estimation for the optimal packet size

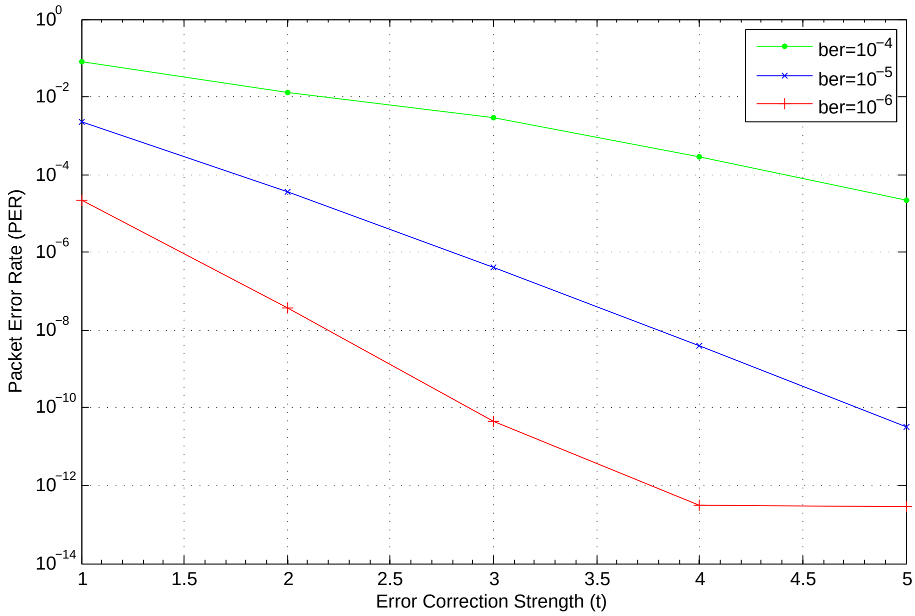

Figure 5.5: PER estimation for optimal packets for different error correction code strength

Hence, for a specific value of (t) energy efficiency becomes highest. After that if the value of error correction strength increases the energy efficiency value becomes low due to extra overhead of FEC and decoding cost.

Figure 5.3 shows the energy efficient optimal packet sizes for different error correction strength. In this figure MTU splitting is not considered for optimal packet sizes. Because if MTU splitting is considered the maximum payload size would be 511 bytes which is already shown in previous section. Hence, without packet splitting if error correction strength is increased, from figure 5.3 the optimal packet size can be found. The energy efficiency and PER are estimated for the corresponding packet sizes in figure 5.4 and figure 5.5.

From figure 5.4 it is revealed that high error correction strength does not increase energy efficiency. If error correction strength increases, energy efficiency decreases gradually. However, since the PER become extremely low (see figure 5.5), in the multi-hop topology network to traverse multiple nodes in noisy channel high strength FEC can be useful.

5.2 Energy Efficient Packet Size with FEC and Multi-hop Network

In the previous chapter 4.4 it is revealed that increasing number of hops reduce energy efficiency. Since the number of hops increase packet error rate, as a consequence energy efficiency decreases. In section 5.1.2 FEC method is used to improve the energy efficiency for the single hop topology and found that FEC increases the energy efficiency only for bigger packets. But FEC decreases PER and increases reliability of the channel. Therefore by using the FEC method PER can be kept low for multi-hop network. Hence, longer distance can be traversed with bigger packets and high efficiency.

Since, the number of hops have a relation with the channel noise and PER of the channel (see section 3.3). The reliability function of multi-hop network is different to the single hops' reliability function. The reliability function for the multi-hop topology

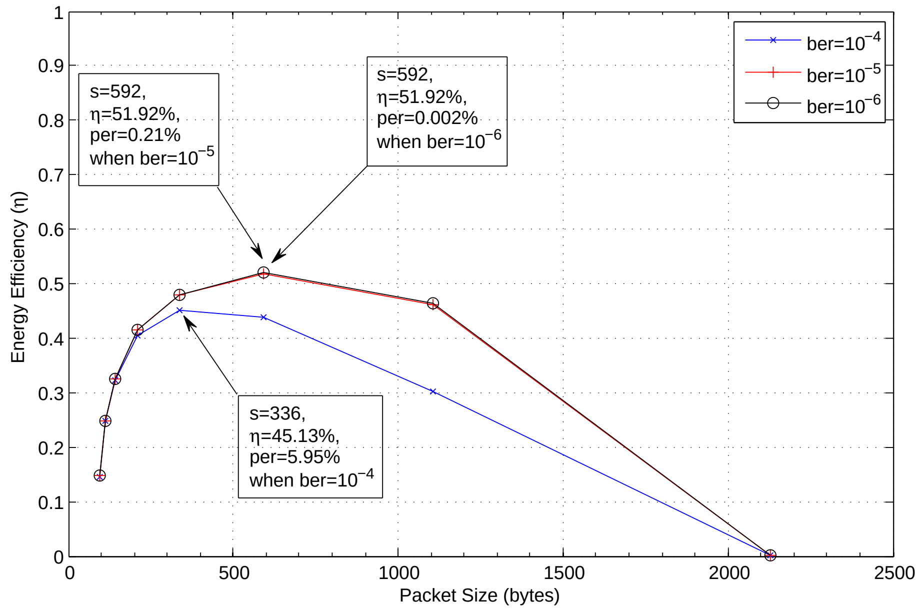

Figure 5.6: Energy efficient packet size for 2 hops chain topology with FEC and $t = 1$

with FEC is given by [KSPR04]

$$r_{mul}^{fec} = \left(\sum_{i=0}^{t} \begin{pmatrix} L \\ i \end{pmatrix} ber^i (1 - ber)^{L-i} \right)^N \tag{5.8}$$

where, r_{mul}^{fec} represents the reliability of the channel and N is the number of hops. Similar to equation (4.12) energy efficiency estimation for the multi-hop linear chain with FEC method is expressed as

$$\eta = \frac{\ell(nk - E_r) \times r_{mul}^{fec}}{L \times (nk - E_r) + (n-1)E_{dec}} \tag{5.9}$$

From equation (5.9) energy efficiency is plotted against packet size in figure 5.6. Comparing this figure with figure 4.8 it is found that for the two hops network FEC is only energy efficient for bigger packets. But after using the FEC method PER diminishes which is an advantage as it increases the reliability of the network.

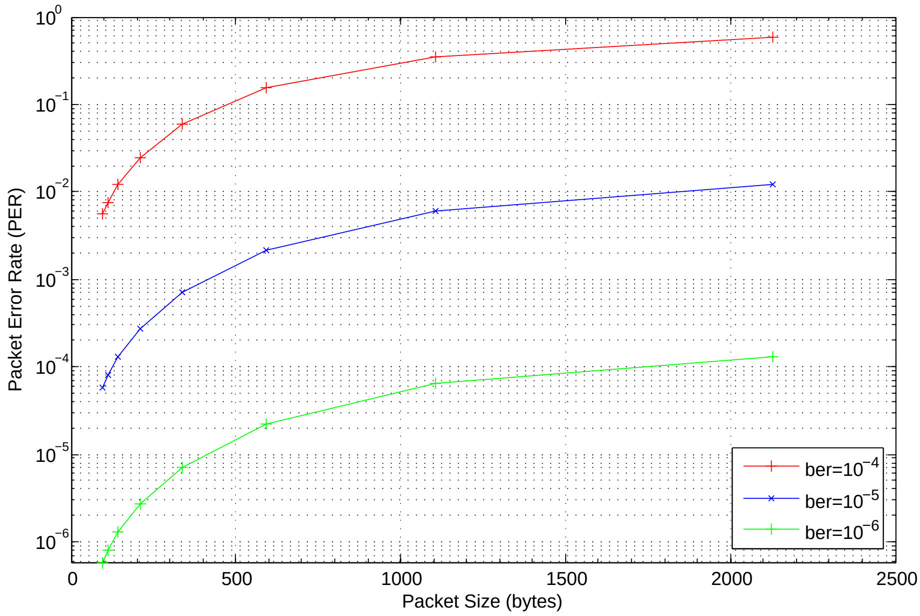

Figure 5.7: Packet error rate estimation for 2 hops chain topology with FEC and $t = 1$

5.2.1 FEC Code at the Destination Node and Energy Efficient Optimal Packet Size

FEC is an energy efficient method only for bigger packets and multi-hop topology networks. For the smaller packet sizes and single hop topology network, due to extra overhead of FEC and decoding cost, it is not an efficient system. However, to traverse multiple hops and to handle the increasing PER with the number of hops, FEC is a reliable and efficient method.

Hence, to reduce the energy cost of FEC in a multi-hop network, rather decoding the packets in every intermediate nodes in a multi-hop network, FEC can be used at the edge nodes (source and destination nodes) only. The source node encodes the packet and whenever the packet will reach the destination end, the destination node will use the FEC to correct errors. The intermediate nodes will not use any FEC.

For the multi-hop linear chain like topology, energy efficiency estimation equation for the FEC in every node has been shown in equation (5.9). For FEC in edge nodes the

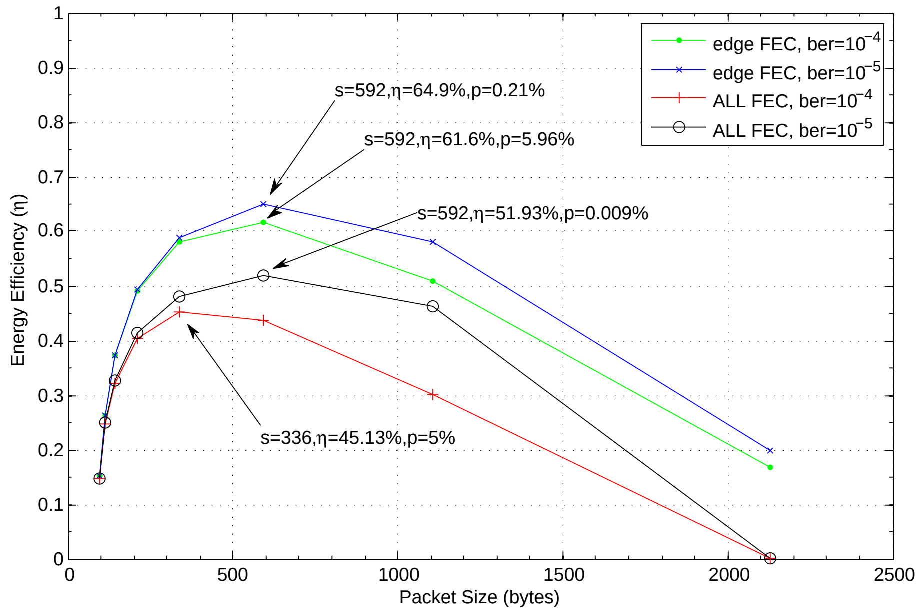

Figure 5.8: Energy efficiency estimation in a 5 hops chain topology to compare FEC at each node and only at the edge nodes

energy efficiency estimation equation is

$$\eta = \frac{\ell(nk_1 - E_r) \times r_{edge}^{mul}}{L \times (nk_1 - E_r) + E_{dec}} \tag{5.10}$$

where n is the number of nodes and r_{edge}^{mul} is the reliability of using the FEC in only the source and destination node. The r_{edge}^{mul} yield as below

$$r_{edge}^{mul} = (1 - PER)^N + r_{fec} \times (1 - (1 - PER)^N) \tag{5.11}$$

where $N = (n - 1)$ the number of hops and r_{fec} is the error correction rate which is similar to equation (5.3).

From figure 5.8 it is revealed that FEC at the edge nodes are more energy efficient than the FEC at all nodes because for the FEC in all nodes, each nodes uses FEC algorithm that increases the cost of packet encoding decoding. In figure 5.8 and 5.9 energy efficiency and PER are compared with FEC at all nodes and FEC at the edge nodes. From these two figure it is found that applying the FEC at the edge nodes are

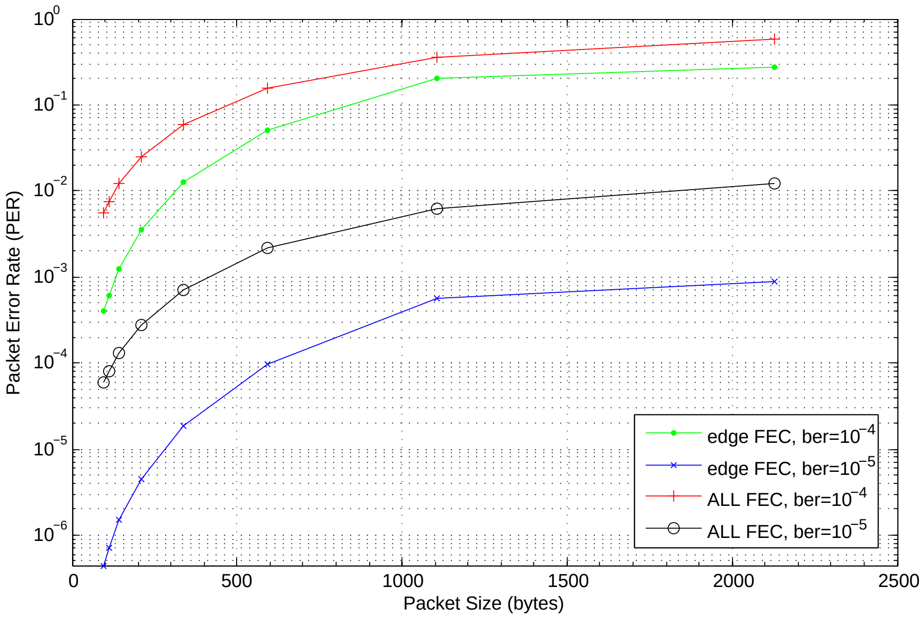

Figure 5.9: PER estimation in a 5 hops chain topology to compare FEC at each node and only at the edge nodes

more energy efficient and PER remains low.

5.2.2 Energy Efficiency with High Strength FEC in Large Multi-hop Network

In a multiple hop network, if a high strength error correction code is used then it is expected that large number of hops can be traversed. Also, by using the FEC larger packets can be transmitted through the multi-hop network. It is shown in section 5.1.2 that for the single hop topology network, high strength FEC increases energy efficiency only for bigger packet sizes. Since, the bigger packets have high PER and FEC reduces the PER.

In a multi-hop network, whenever the packets traverse multiple hops, PER increases (see section 3.3) gradually. To keep PER lower and traverse multi-hop network, only header packet is proposed in the previous chapter. Since, smaller packets have lower PER. However, smaller packet carries a large portion of overhead. Hence, if bigger

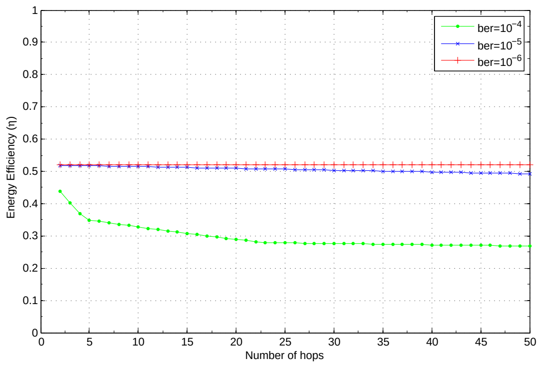

Figure 5.10: Maximum energy efficiency estimation with 511 bytes payload and high strength FEC in a large chain topology

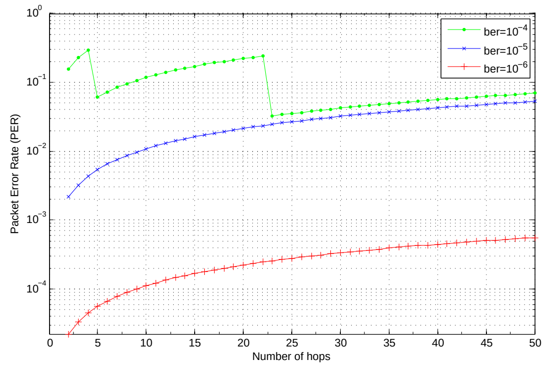

Figure 5.11: PER estimation in a large chain topology with 511 bytes payload and high strength FEC

packets are transmitted overhead will be less but PER will be higher. Since, it is found that FEC reduces the PER significantly, therefore by using high strength FEC large multiple hops can be traversed with lower PER with bigger packets.

FEC maintains specific constrains that the payload size must be $2^m - 1$. Considering the MTU packet split it is showed in section 5.1.2 that the maximum payload size with the FEC is 511 bytes ($2^{12} - 1$ bits).

Hence, for specific number of hops most energy efficient error correction strength for FEC is estimated for 511 bytes payload. In figure 5.10 maximum energy efficiency is shown for 511 bytes payload for various number of hops. The corresponding PER is also estimated in figure 5.11 and the optimal FEC strength is shown in figure 5.12. From this figures it is clearly revealed that by using the FEC large number of hops can be traversed with very low PER.

In figure 5.11 PER for the noisy channel with 10^{-4} BER changes the trend pattern multiple times. The reason behind this, from figure 5.12 it is found that only for the 10^{-4} BER, most energy efficient error correction strength changes with the increasing number of hops. Therefore, whenever there is a change of error correction strength the PER trend changes in figure 5.11.

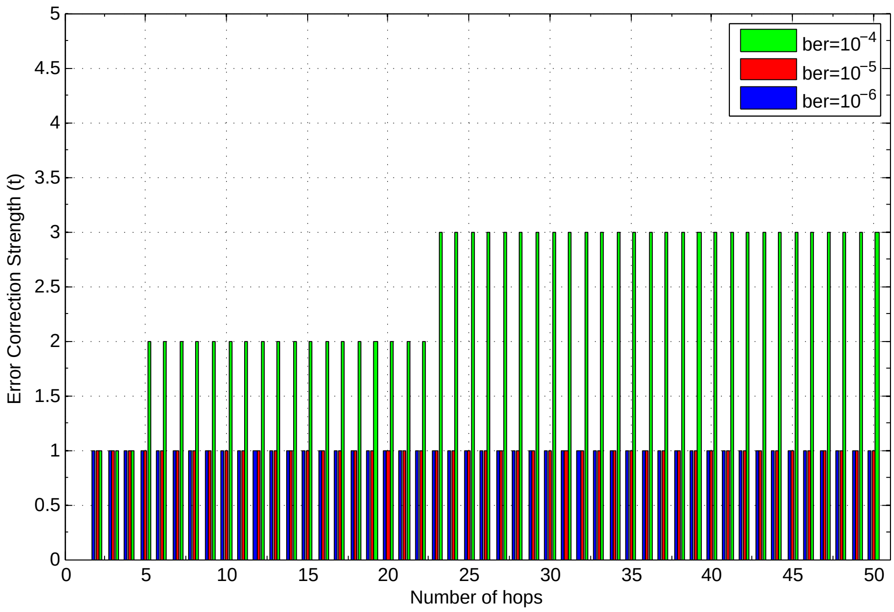

Figure 5.12: Energy efficient FEC strength estimation for 511 bytes payload in multi-hop network

5.3 Packet Compression and Energy Efficient Optimal Packet Size

In the last sections it is observed that FEC method improves the channel reliability and keeps the PER low in multi-hop network. However, FEC incorporates overhead bits and extra encoding decoding cost. After using the FEC large portion of a packet is wasted only for data administration purpose. To reduce the overhead, compression can be used as a remedy. Compression is a renowned method to reduce overhead which is described in chapter 3. In this section the energy efficiency of packet compression is analyzed and optimal packet size is estimated.

5.3.1 Energy Efficiency Estimation with Compression Method

Just like as every method or protocol, packet compression also has specific energy cost. It is already showed and explained in the packet compression chapter that LZO compression is an efficient compression algorithm. It consume less energy, takes very low processing time for compression decompression and also compression ratio is 38.5% approximately. Hence, the LZO compression algorithm is considered to estimate energy efficiency and optimal packet size.

Three different types of compression algorithm are illustrated in chapter 2.6 header, payload and whole packet compression. Since, the whole packet compression needs another dictionary to install in to the nodes, in this chapter, only the header and payload compression is analyzed.

Recalling the equation (5.4) the energy efficiency for the payload compression for a single hop topology can be rewritten as below

$$\eta = \frac{(\phi\ell \times k)}{L^{p}_{com}k + E_{com}\ell}(1 - ber)^{L^{p}_{com}} \tag{5.12}$$

where, E_{com} is the compression and decompression energy cost. ϕ is the compression ratio, L^{p}_{com} is the compressed packet size with the compressed payload, that yield

$$L^{p}_{com} = \alpha + \phi\ell \tag{5.13}$$

where, $\phi\ell$ is the compressed payload and α is the header bytes. For header compression, energy efficiency equation can be expressed as below

$$\eta = \frac{(\ell \times k)}{L^{h}_{com} \times k + E_{com}\alpha}(1 - ber)^{L^{h}_{com}} \tag{5.14}$$

where, L^{h}_{com} is the compressed packet size with the compressed header, that yield

$$L^{h}_{com} = \alpha_{com} + \ell \tag{5.15}$$

where, α_{com} is the compressed header and ℓ is the payload bytes. For the multi-hop linear chain topology in every single intermediate data forwarding node the packets need to be decompressed at first and after that compressed again to forward it towards the destination. Hence, the compression-decompression cost higher than for a single hop.

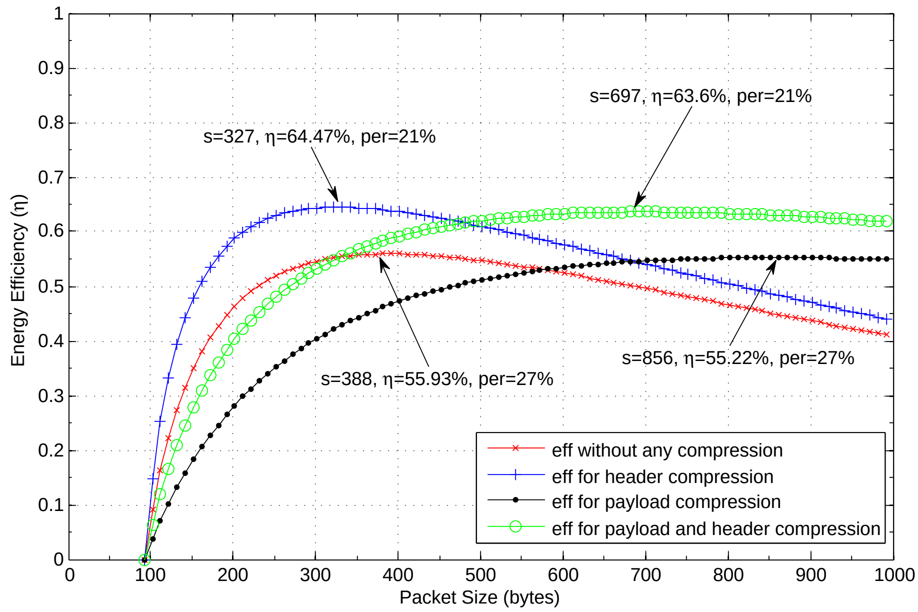

Figure 5.13: Energy efficiency comparison for different types of compression with 10^{-4} BER

From equation (4.12) for the multi-hop linear chain like topology energy efficiency for the payload compression method can be expressed as

$$\eta_{com} = \frac{\phi\ell \times (nk - E_r) \times (1 - ber)^{N \times L_{com}^p}}{L_{com}^p \times (nk - E_r) + nE_{com}\ell} \tag{5.16}$$

For header compression, total packet size is L_{com}^h and header size is α_{com}. In figure 5.13 and 5.14 the energy efficiency and PER are shown for various compression methods. After analysing the two figures it is found that header and payload compression methods are more energy efficient and at the same time more reliable than any other compression. Header compression is little more energy efficient than the header and payload combined compression method but in terms of transmitting larger packets, header compression is inefficient because of high PER. The PER of header compression is almost similar to PER without compressed packet.

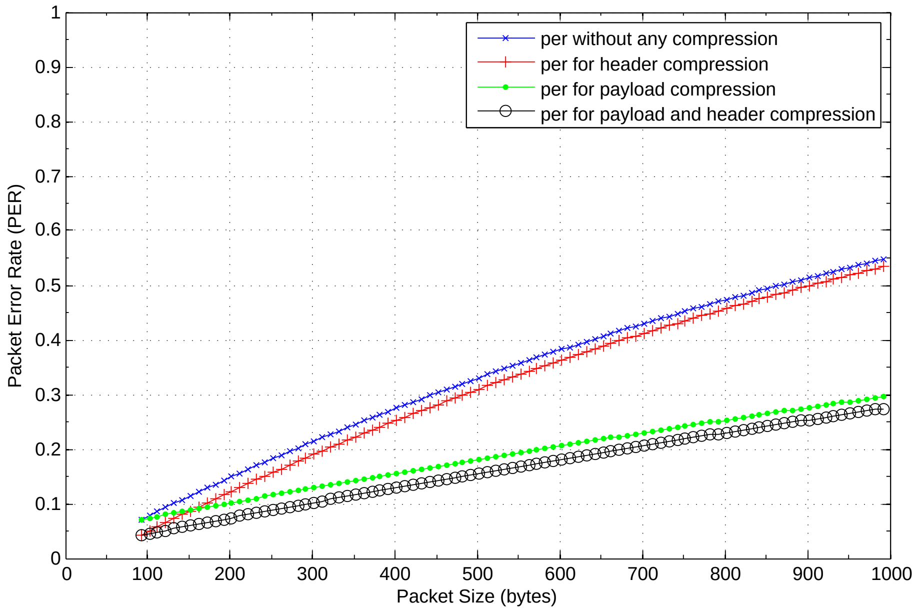

Figure 5.14: PER comparison of different types of compression for 10^{-4} BER

5.3.2 Energy Efficiency Comparison Between Compression and FEC

Wireless network is usually error prone. Hence, the compressed packet size can be corrupted. To reduce the compressed packet size corruption rate, FEC method can be used. Usually the FEC method is applied in the application layer (see chapter 2). Compression can be on the application layer, network or transport layer. If FEC method is used to correct the compressed packet size then the reliability function becomes similar to equation (5.3) and the packet size (L) will be different. Similar to equation (5.12) and (5.14) packet size L will be L^p_{com} and L^h_{com} for the payload and header compression respectively.

The FEC reliability equation (r_{com}) of only header compression method for the multi-hop topology is given below

$$r_{com} = \left(\sum_{i=0}^{t} \binom{L^h_{com}}{i} ber^i (1 - BER)^{L^h_{com} - i} \right)^N \tag{5.17}$$

From figure 5.13 and 5.14 it is clear that the header and payload combined compression

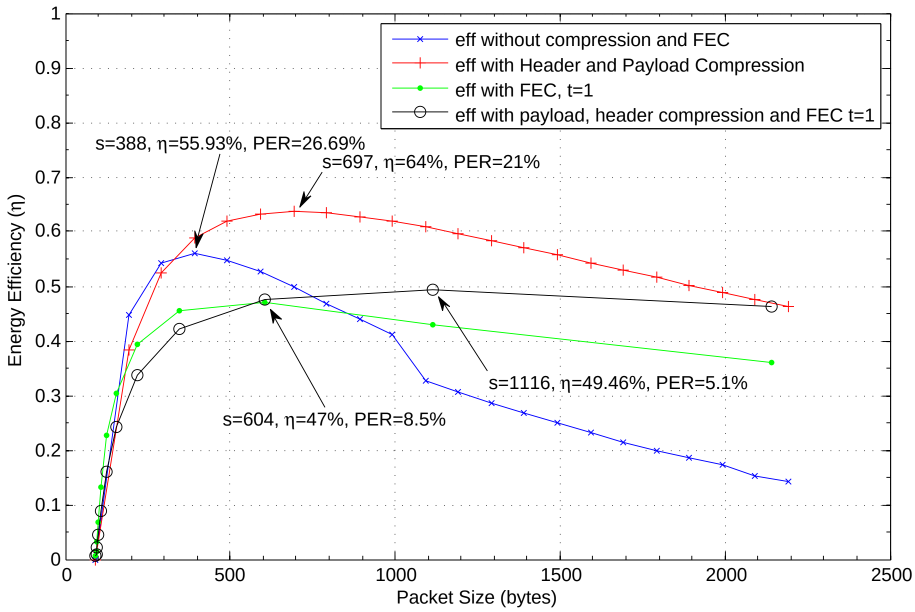

Figure 5.15: Energy efficiency comparison of different methods when BER=10^{-4}

is the best compression method. Therefore, to compare the combined compression with FEC two figures are plotted. In figure 5.15 energy efficiency is plotted against packet size for compression, FEC, combined and without any of them. From these two graphs it is found that the combined method compression and FEC does not increase the energy efficiency. The payload and header compression shows the most energy efficiency. However, in terms of PER the FEC method with the compression has the lowest PER. As a consequence, higher efficiency can be achieved for larger packets (here MTU splitting is not considered). If MTU (100 bytes) splitting is considered then for the FEC, 511 bytes will be maximum transferable packet size (see section 5.1.3). To analyze the impact of FEC and compression combined method and the compression methods are compared for the multi-hop network in figure 5.16. From figure 5.16 it is found that whenever the number of hops increases the energy efficiency becomes higher for the FEC than compression. Hence, it can be stated that compression is energy efficient and better for the smaller packet sizes and single hop network whereas the FEC and compression combined method is efficient for the multi-hop topology network.

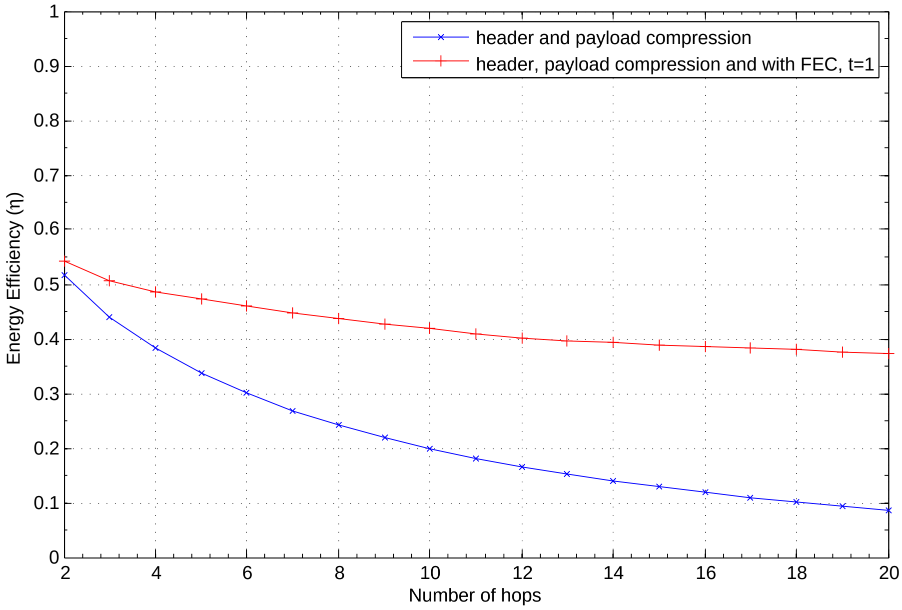

Figure 5.16: Comparison of energy efficiency for multiple hops between compression and compression-error correction combined method when BER=10^{-4}

5.4 Conclusion

In this chapter the impact of the compression and FEC is investigated in terms of energy efficiency. It is found that FEC is an inefficient method for the smaller packets and single hop topology network. Whereas for the bigger packets and multi-hop topology network it is reliable and energy efficient. Hence, larger MTU size would be better for the FEC method. Then, packet split will occur for very large packets and high energy efficiency can be achieved for the bigger packets which is not possible in a noisy channel without FEC.

Compared with compression method header and payload combined compression is found as the most efficient method for the smaller packet sizes in a single hop topology network. But for the multi-hop and bigger packet sizes, the energy efficiency is less than FEC. Therefore, according to the requirements and network scenario different optimal packet size needs to be defined and techniques need to be adopted to increase energy efficiency.

References

[KS06] Z.H. Kashani and M. Shiva. BCH coding and multi-hop communication in wireless sensor networks. In *Wireless and Optical Communications Networks, 2006 IFIP International Conference on*, pages 5 pp. –5, 0-0 2006.

[KSPR04] H. Karvonen, Z. Shelby, and C. Pomalaza-Raez. Coding for energy efficient wireless embedded networks. In *Wireless Ad-Hoc Networks, 2004 International Workshop on*, 2004.

[SAM03] Y. Sankarasubramaniam, I.F. Akyildiz, and S.W. McLaughlin. Energy efficiency based packet size optimization in wireless sensor networks. In *Sensor Network Protocols and Applications, 2003. Proceedings of the First IEEE. 2003 IEEE International Workshop on*, pages 1 – 8, 2003.

[VA06] M.C. Vuran and I.F. Akyildiz. Cross-Layer analysis of error control in wireless sensor networks. In *Sensor and Ad Hoc Communications and Networks, 2006. SECON '06. 2006 3rd Annual IEEE Communications Society on*, volume 2, pages 585 –594, sept. 2006.

[ZDQ08] T. Zhen, Y. Dongfeng, and L. Quanquan. Energy efficiency analysis of error control schemes in wireless sensor networks. In *Wireless Communications and Mobile Computing Conference, 2008. IWCMC '08. International*, pages 401 –405, aug. 2008.

Chapter 6

Optimal Packet Size Estimation for Congested Channel

In this chapter energy efficient optimal packet size is estimated for multiple sources in a contending congested channel. Energy efficiency estimation for optimal packet size with multiple sources is significant. In an ad hoc network, whenever multiple nodes start to transmit at the same time, the packets from different nodes collided with each other. The collided packet does not reach to its destination; it becomes lost in the medium. As a result packet loss rate increases and performance deteriorates. Dynamic scheduling is needed to synchronize the packet transmission and fair access allocation time between the nodes. The IEEE 802.11 is a standard protocol for the WLAN provides the dynamic scheduling for multiple nodes. In this chapter collision probability is investigated and energy efficient optimal packet size is estimated for multiple sources of WLAN.

6.1 IEEE 802.11 MAC Protocol for Scheduling and Synchronization

IEEE 802.11 protocol [Bia00, TC01, TSKC96, MCBT05] is defined in two layers: MAC and PHY layer. The MAC layer determines how the channel is going to be used between the ad hoc nodes. It synchronizes the packet transmission time and fair access scheduling between WLAN nodes. Specifically the 802.11 MAC is called Distributed Coordination Function (DCF). DCF [MCBT05] uses two types of scheduling scheme to prevent data collision. One is called basic scheme which is defined by default another one is called RTS-CTS. The Basic scheme is a two way handshaking policy where data packet from the source node is preceded by an ACK packet from the destination node. Since in the wireless medium source node cannot determine successful reception of the transmitted data, ACK packet ensures the confirmation.

In RTS-CTS method a RTS packet is broadcasted by the source node (if the channels remains free or idle for a specific time period). Then the destination node replies back with a CTS packet. After that the data packet is transmitted and then a confirmation acknowledgment packet is sent by the destination node. Through this way a successful transmission is preceded. During one whole transmission period (RTS-CTS-DATA-ACK) the neighbor nodes remains quiet and suppress their transmission for the entire period. In a busy channel the source nodes wait and listen the channel until its become free. If multiple nodes become active at the same time and start to transmit their packet, the packet collides with each other with the neighbor nodes and transmission becomes unsuccessful. Whenever any transmission becomes unsuccessful or any transmission attempt is failed, the node starts waiting for an exponential random backoff time. Random backoff counter time (BA) can be described as below [TSKC96]

$$BA = (CW * random()) * S \tag{6.1}$$

where S represents the slot time, which is one unit of time defined in the physical layer.

The CW defines the contention window size, $random()$ is a random number generator function which generates value between 0 and 1. The typical value of CW stays between 32 (CW_{min}) to 1024 (CW_{max}). For any i times of unsuccessful transmission the window size will be CW_i which is given by [Bia00, TC01]

$$CW_i = 2^i CW_{min};\ i = [0, 1, ...m] \tag{6.2}$$

Hence, for each failed attempt the backoff window time increases exponentially. Whenever it reaches to maximum ($m = 5$) the backoff window size becomes constant. If the retry limit exceeds the maximum retry limit (5 for Basic scheme and 7 for RTS-CTS) the frame is discarded.

Whenever the waiting node senses the channel is free, it starts decreasing the backoff counter. The waiting node holds back transmission until the backoff counter reaches zero. If the channel becomes busy within the backoff decrement countdown, the node freezes the timer and waits for the channel to be free again. After every successful packet transmission, each node waits for a random backoff time before the next transmission attempt. It gives chance the other waiting nodes to get access to the channel and allow fair channel access allocation among the neighbour.

6.2 Multiple Nodes Transmission of WLAN in Noisy Channel Condition

The ad hoc wireless network is comparatively noisy and error prone. Since, the 802.11 MAC protocol needs to operate in the noisy channel therefore to estimate the energy efficiency and optimal packet size, the PER needs to be estimated. Let us consider a noisy network with n number of ad hoc nodes with random topology. All the nodes are within their transmission range. With the IEEE802.11 MAC protocol DCF function RTS-CTS or Basic scheme, each node can observe one of the following conditions in a noisy channel

1. RTS/CTS/DATA/ACK frame corruption due to bad channel condition and due to fail transmission the nodes start to wait for random backoff time and restart the whole process (RTS-CTS-DATA-ACK) again.

   For the basic scheme, DATA/ACK frame corruption due to bad channel condition. The corrupted packet will be discarded and random backoff will commence.

2. RTS frame collision due to two or more nodes transmitting RTS frames within the same time slot and then start to wait for a random backoff time.

   For the Basic scheme, DATA-DATA frame collision due to two or more nodes transmitting DATA frames in the same time slot and the collided nodes begin a random backoff countdown.

3. Successful transmission of a frame (RTS, CTS, DATA, ACK transmitted and received successfully) or successful transmission of a frame with the Basic scheme (DATA, ACK transmitted and received successfully).

Hence, whether it is Basic or RTS-CTS scheme, every transmission is dependent on multiple number of packets. For the basic scheme the transmission of one data packet depends on the correct transmission of the DATA and ACK packet. For the RTS-CTS scheme, one successful transmission depends on the successful transmission of RTS, CTS, DATA and ACK packet. Hence, to estimate the packet loss rate of these two scheme in a noisy channel, the probability of different packet loss (RTS, CTS, DATA, ACK) is needed to define for a single packet transmission.

Considering the noisy channel condition for the RTS-CTS scheme four conditions can possible which is defined as below

- Either the RTS is corrupted: p_1
- RTS successful but CTS corrupted: p_2

- RTS-CTS successful but Data corrupted: p_3
- RTS-CTS-Data successful but ACK corrupted: p_4

That yield as below [SYHH09, SYHH10]

$$\begin{aligned} p_1 &= p_r \\ p_2 &= p_c(1-p_r) \\ p_3 &= p_d(1-p_c)(1-p_r) \\ p_4 &= p_a(1-p_d)(1-p_c)(1-p_r) \end{aligned} \quad (6.3)$$

Where p_r, p_c, p_d and p_a are the error probabilities (PER) for the RTS, CTS, DATA and ACK packets. These parameters can be simplified as

$$p_{r/c/d/a} = 1-(1-ber)^{L_{rts/cts/data/ack}} \quad (6.4)$$

Where, L_{data} is $(\alpha + \ell + \tau)$ the total packet size of data packet (see equation 4.3 for detail). Similarly $L_{rts/cts/ack}$ are the packet sizes of RTS, CTS and ACK packets.

The total PER for the RTS-CTS (P^{rc}_{err}) and Basic scheme (P^{b}_{err}) is given by [SYHH09, SYHH10]

$$\begin{aligned} P^{rc}_{err} &= p_1+p_2+p_3+p_4 \\ P^{b}_{err} &= p_d+p_5 \\ p_5 &= p_a(1-p_d) \end{aligned} \quad (6.5)$$

6.3 Energy Efficiency Estimation with 802.11 MAC

To estimate the energy efficiency for WLAN 802.11 MAC protocol, at the beginning it is assumed, MAC protocol works perfectly and there is not packet collision within the network. Since, the wireless channel is error prone, in a noisy channel there will be packet loss due to packet corruption. In the previous section in equation 6.3 and 6.5 the error probabilities for RTS, CTS, DATA and ACK packet is described for a collision free noisy channel. In this section, energy cost of those different scenarios is estimated to calculate the energy efficiency.

For the Basic scheme three cases can be possible for a single packet transmission successfully in a noisy channel.

1. n number of DATA packets can be corrupted for a single successful packet transmission. The average PER for the DATA packet can be estimated from p_d (see equation (6.4)). Lets assume, the energy cost for this amount of packet loss (p_d) is x.

2. Then, if the DATA packet is received successfully after that n number of ACK packets can be corrupted. Similarly, the packet error rate for the ACK is p_a. Then, considering the packet loss for p_a the energy cost is y.

3. After that successful transmission of a DATA and ACK packet. The energy cost for this type of event is z.

Above three different energy cost x, y and z is expressed as

$$\begin{aligned}
x &= (k_j \times L_{data} \times p_d) \\
y &= (k_j \times ((1 - p_d) \times L_{data} + p_5 \times L_{ack})) \\
z &= (k_j \times ((1 - p_d) \times L_{data} + (1 - p_d) \times (1 - p_a) \times L_{ack})) \qquad (6.6)
\end{aligned}$$

where k_j is the energy required to communicate a single bit (see equation (4.2) of section 4.2).

The energy efficiency is a energy throughput of energy out and energy in. To estimate the energy efficiency the equation (4.3) and (4.7) of chapter 4 are followed. The numerator in the energy efficiency equation is a product of payload and reliability of the channel which is similar to equation (4.7). But the denominator is different. Because multiple number of RTS/CTS/DATA/ACK packets can be lost for a single packet transmission successfully. The energy efficiency of Basic and RTS-CTS scheme are defined as η_{basic} and $\eta_{rts-cts}$. That yield

$$\eta_{basic} = \frac{k_j \times \ell \times (1 - P^b_{err})}{(x + y + z)} \tag{6.7}$$

Similar to η_{basic} to estimate the energy efficiency for the RTS-CTS ($\eta_{rts-cts}$), it is needed to define how many number of packets can be corrupted to transmit a single data packet successfully in the noisy channel (which is showed in previous section in equation (6.3)). Hence, energy cost is needed to estimate for each possible scenarios. For the RTS-CTS scheme five cases can be possible. The energy cost for these scenarios is expressed as

1. n number of RTS packet can be corrupted for a single successful packet transmission. It is assumed, the energy cost for this event a

2. After that if the RTS packet is received successfully then n number of CTS packet can be corrupted. The energy spent for this event is assumed b

3. Next, if the RTS, CTS packets are received successfully after that n number of DATA packets can be corrupted. The amount of energy cost is considered c.

4. If the RTS, CTS, DATA packets are received successfully then n number of ACK packets can be corrupted. The energy cost is d.

5. Finally, one successful transmission of a RTS-CTS-DATA-ACK packet. The energy cost for this successful event is e.

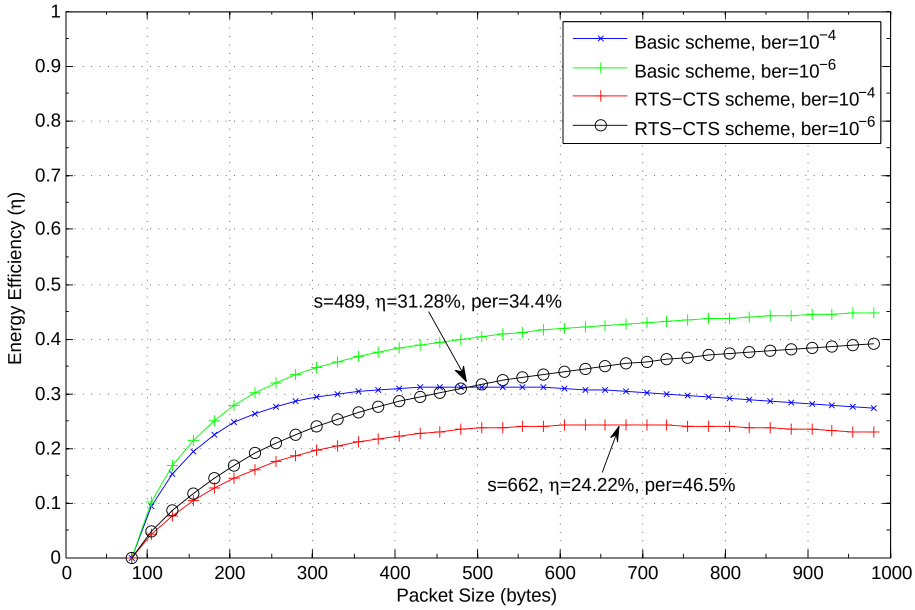

Figure 6.1: Energy efficiency analysis with the MAC protocol for the Basic and RTS-CTS scheme for two different BER

The above energy cost for different scenarios is simplified as

$$
\begin{aligned}
a &= k_j \times L_{rts} \times p_r \\
b &= k_j \times (L_{rts} \times (1-p_r) + L_{cts} \times p_2) \\
c &= k_j \times (L_{rts} \times (1-p_r) + L_{cts} \times (1-p_r)(1-p_c) + L_{data} \times p_3) \\
d &= k_j \times (L_{rts} \times (1-p_r) + L_{cts} \times (1-p_r)(1-p_c) + L_{data}(1-p_r)(1-p_c)(1-p_d) \\
&+ L_{ack} \times p_4) \\
e &= k_j \times (L_{rts} \times (1-p_r) + L_{cts} \times (1-p_r)(1-p_c) + L_{data}(1-p_r)(1-p_c)(1-p_d) \\
&+ L_{ack}(1-p_r)(1-p_c)(1-p_d)(1-p_a))
\end{aligned} \quad (6.8)
$$

Therefore, the energy efficiency equation for the RTS-CTS scheme $\eta_{rts-cts}$ yield as

$$\eta_{rts-cts} = \frac{\ell \times (1-P_{err}^{rc})}{a+b+c+d+e} \quad (6.9)$$

In figure 6.1 the energy efficiency is compared between the RTS-CTS and for the Basic methods. Energy efficient optimal packet size is pointed with arrows and $'s'$ represents the optimal packet size. Only for the noisy channel condition (when BER is 10^{-4}) optimal packet size is shown with arrows. For the good channel condition the most energy efficient optimal packet size is the MTU size packet. Since, it is showed in chapter 4 that more than the MTU size, packets are split apart and become multiple packets with individual headers and trailers. The larger packets more than the MTU not only increases the overhead also increases PER. Therefore, in the good channel condition, the MTU becomes the most efficient packet size.

One observation from this figure is for the noisy channel with high BER, the energy efficiency become flat for a large range of packet sizes. Therefore, it is stated that although there is an optimum point for a specific packet size from 450 bytes to 1000 bytes all the packets are more than 20% energy efficient. Whereas the most energy efficient packet size is 662 bytes that has a 24% energy efficiency. Therefore, if the packets have more than 400 bytes almost maximum efficiency can be achieved without applying the exact optimal packet size. The same phenomenon happens for the Basic scheme as well which is an advantage. Because without using exact optimal packet size, by using an optimum range of packet size will provide more flexibility to choose any packet size from a range, those have almost similar energy efficiency.

The energy consumption of the theoretical model is compared with the simulated network model. The theoretical energy consumption for a single packet is given in the denominator part of equation (6.7) and (6.9) for the Basic and RTS-CTS scheme. To compare the energy cost, it is assumed the channel has very low noise, there is not any packet loss (due to packet error or packet collision).

The energy cost for a single data packet is estimated from the simulation and showed in figure 6.2. It is found that the results matches properly for the smaller packet sizes and the standard deviations are also low. But for the higher packet sizes the deviations are large. In figure 6.2 it is found that for the packet loss free channel with multiple source node the theoretical model matches with the simulated model.

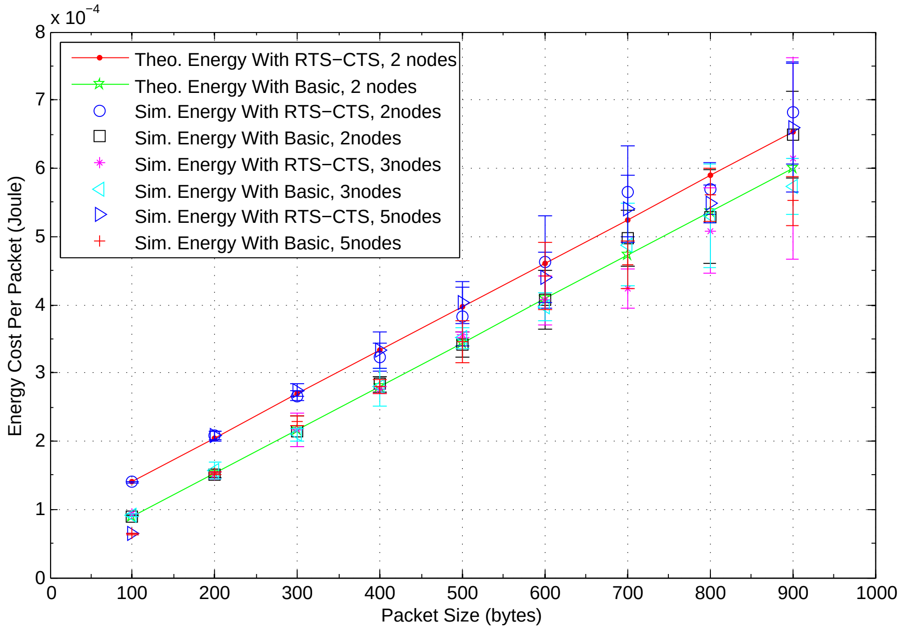

Figure 6.2: Energy cost comparison with the theoretical model and the simulated network with BER 10^{-5}

6.3.1 PER Comparison for the Simulated 802.11 WLAN

To compare the theoretical PER results of equation (6.5) with simulation results a small network is created with three nodes. The nodes are within their transmission range and each node act as a source node with poisson source UDP traffic. Since, it is assumed in this model that there is not any collision between the nodes, only packet loss happens due to channel noise for the corresponding packet error. In the simulation the nodes are set with low data rate and all the nodes are within the transmission range of each other. There is no hidden nodes in the network. Hidden node is the situation where neighbour nodes are not visible to each other but during the transmission their packets can collide. The Hidden nodes problem can be resolved by the RTS-CTS scheme [TC01, Bia00, HT06]. But, it costs two more control packets. Hence, for the Basic scheme the simulation scenario is set such a way that all the nodes are within their transmission range. The data rate is very low compared to bandwidth limit and queue size in each node is very large. Packet loss can only occur due to packet errors and no collision happens.

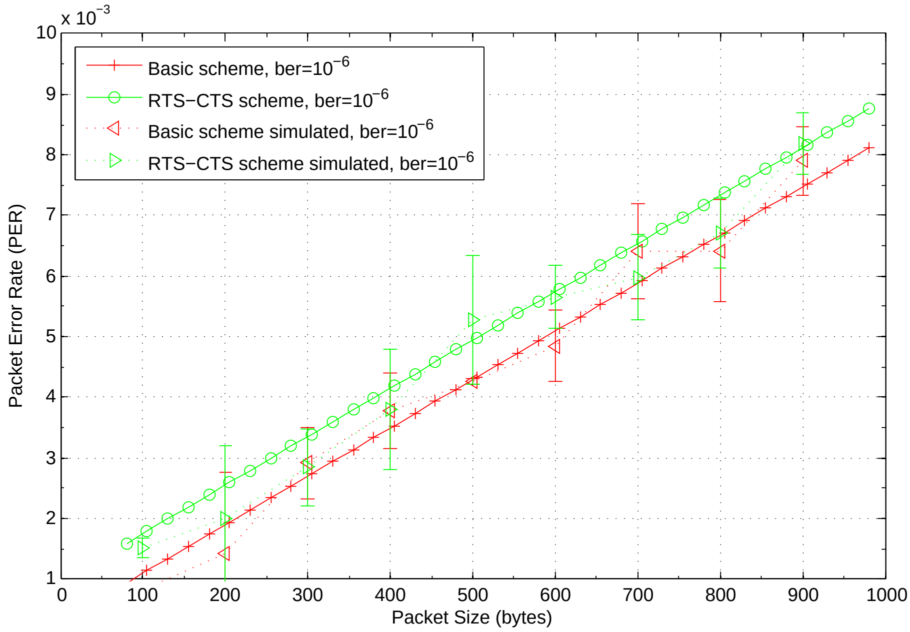

Figure 6.3: PER comparison between the theoretical model and the simulated network of a low noisy channel

In figure 6.3 the packet loss rate between the simulated results and theoretical results for the corrupted packets are shown. From the comparison it is found that for the good channel condition the PER matches with the theoretical model with low deviation. In figure 6.4 the PER is compared for the RTS-CTS and Basic methods for various number of nodes for comparatively more noisy channel (BER 10^{-5}). In this figure 6.4 within the legend R/C and BS represents RTS-CTS and Basic scheme. Figure 6.4 shows that increasing the number of nodes does not increase the PER. Since, for the multiple nodes MAC scheduler allows only one node to transmit packets at a certain time. Therefore, the overall packet loss rate remains the same and it does not create any impact on the packet loss rate for different number of nodes. However, for the noisy channel with high PER it is found that the theoretical model does not match properly with the simulation. For lower PER the theoretical results match with the simulated results with low deviation. For high PER, the simulated PER is higher than the theoretical estimated PER and also the deviation is very high which is identical to figure 6.2.

For the larger packets to achieve a fixed given data rate, less number of packets are

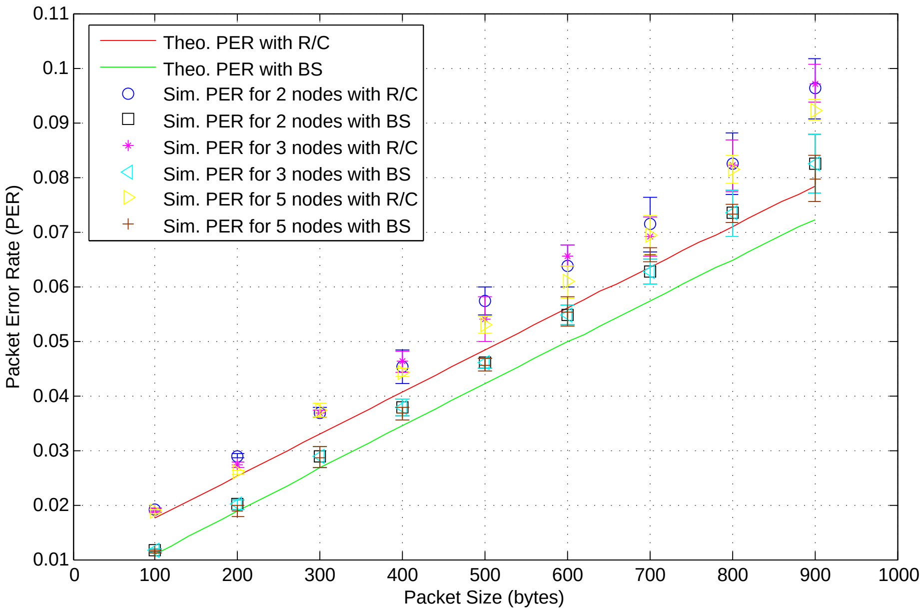

Figure 6.4: Energy cost comparison with the theoretical model and the simulated network for BER 10^{-5}

needed to generate by the source node. Whereas for the sorter packet more number packets are needed to achieve the same data rate. Therefore, for the bigger packets due to less number of packets deviation become larger.

6.4 Packet Loss Rate Estimation for Multiple Sources Considering Packet Collision and Packet Corruption

To estimate the collision probability, it is vital to measure the total packet loss rate of the WLAN networks. For the collision probability estimation of WLAN networks, most of the previous research [Bia00, TC01, TSKC96, MCBT05] follows similar strategy. However, there are some differences in average backoff time estimation. In this section the collision probability is estimated and put into context with previous research.

In [TC01] the channel has been assumed as saturated, where some packets always remains in the nodes queue to be transmitted. Each transmission has collision probability p, hence, the success probability is $(1-p)$. If the first transmission attempt fails, the probability of transmitting the packet successfully on the second attempt will be $p(1-p)$. Every packet transmission is preceded after a backoff. Each collision creates a backoff countdown which repeats until it reaches the maximum limit. Therefore, average backoff counter can be estimated from $\frac{CW-1}{2}$. Each node transmits a packet multiple times to send it successfully. Whenever there is an unsuccessful transmission, the backoff window size increases. If the transmission attempt fails continuously then the window size increases till the maximum value of m becomes CW_{max}. Hence, it has been assumed that the number of transmissions to transmit a packet successfully is a geometrical distribution.

The average backoff window size has been considered $\overline{BK}$. The probability of collision between two nodes would be $\frac{1}{\overline{BK}}$. The probability of getting a free time slot during transmission when all the neighbour nodes are quiet is given by $(1-\frac{1}{\overline{BK}})^{(n-1)}$. Therefore, the probability of packets are getting collided due to multiple nodes are trying to transmit at the same time is given by [TC01, HT06]

$$P_{col} = 1 - \left(1 - \frac{1}{\overline{BK}}\right)^{(n-1)} \tag{6.10}$$

In [TC01, HT06] the average backoff window $\overline{BK}$ is considered. However, in their estimation the retry limit has not been considered. Considering the retry limit the average backoff window has been estimated in [XA04, JXA05]. In [XA04] the average backoff timer is $\frac{(CW_i-1)}{2}$ where CW_i is the ith number of backoff. Hence, for the four-way handshaking with RTS and CTS, when $(i > m)$, the number of maximum backoff retry limit is 7, which is defined as SRC. For m number of collisions $\overline{BK}$ can be computed for the RTS-CTS as below [HT06]

$$\overline{BK}_{rts-cts} = \sum_{i=0}^{m} \frac{CW_i - 1}{2} p_{rc}^{s} + \sum_{i=m+1}^{SRC} \frac{CW_m - 1}{2} p_{rc}^{s} \tag{6.11}$$

For the Basic scheme the retry limit is 4 defined as LRC and expressed in [TC01, HT06] as below

$$\overline{BK}_{basic} = \sum_{i=0}^{LRC} \frac{CW_i - 1}{2} p^s_{bas} \tag{6.12}$$

Where, p^s_{bas} and p^s_{rc} are the probability to send a packet successfully after i times of unsuccessful transmission due to collision which is given by [XA04, CCG00]

$$\begin{aligned} p^s_{bas} &= (P^b_{col})^i (1 - P^b_{col}) \\ p^s_{rc} &= (P^{rc}_{col})^i (1 - P^{rc}_{col}) \end{aligned} \tag{6.13}$$

Considering that the occurrence of a collision and packet error are independent. The probability of successful transmission in the error-prone channel can be given by [XA04, CCG00] as below

$$\begin{aligned} P^{suc}_{basic} &= (1 - P^b_{err})(1 - P^b_{col}) \\ P^{suc}_{rts-cts} &= (1 - P^{rc}_{err})(1 - P^{rc}_{col}) \end{aligned} \tag{6.14}$$

Hence, P^{suc}_{basic} and $P^{suc}_{rts-cts}$ represents the reliability of the channel of Basic and RTS-CTS schemes.

6.4.1 Energy Cost Estimation of Basic and RTS-CTS schemes for Various Scenarios

To estimate the energy efficiency, energy cost for the different scenarios is needed to be estimated. Similar to section 6.2 for the Basic scheme three cases can be possible for a single packet transmission successfully. At the following, three different cases for the Basic scheme is explained and energy cost is estimated.

1. Either n number of DATA packet can be corrupted or collided for a single successful packet transmission. For the n number of DATA packet loss, lets assume the energy

cost is x.

2. If DATA packet is received successfully after that n number of ACK packets can be corrupted. Since, the average PER for the ACK packet is known p_a. Hence, the average PER for the ACK can be estimated and assume that the estimated energy cost is y.

3. One successful transmission of a DATA and ACK packet. The energy cost for this event is z.

The energy cost (x, y, z) are simplified as

$$\begin{aligned} x &= k_j \times (L_{data} \times p_d \times P^b_{col} + L_{data} \times (1 - p_d) \times P^b_{col} + L_{data} \times p_d \times (1 - P^b_{col})) \\ y &= k_j \times ((1 - p_d)(1 - P^b_{col}) \times L_{data} + p_5 \times L_{ack}) \\ z &= k_j \times ((1 - p_d)(1 - P^b_{col}) \times L_{data} + (1 - p_d) \times (1 - p_a) \times L_{ack}) \end{aligned} \quad (6.15)$$

It is assumed, if the data packet is received successfully from source node to the destination node, then all the neighbour nodes are acknowledged that the channel is occupied. Therefore, the neighbour nodes suppress their packet transmission until successful delivery of ACK packet from the destination node to the source node. As a result, there will not be any collision for the ACK packet. Similarly, for the RTS-CTS scheme there will be no collision for the CTS, DATA and ACK packets.

The energy efficiency equation for the Basic scheme will be similar to equation (6.7). After using the equation (6.15) and for high collision probability (50%) the efficiency is estimated and plotted in figure 6.5.

To estimate the energy efficiency for the RTS-CTS scheme, also similar procedure is followed. The energy cost for different scenarios are estimated separately and after that the energy cost is simplified. For the RTS-CTS five cases can be possible for a single packet transmission successfully which are given below

1. At the beginning of the handshaking process n number of RTS packet can be corrupted or collided for a single successful packet transmission. The energy cost for this event is assumed a.

2. Then, if RTS packet is received successfully after that n number of CTS packet can be corrupted for a single successful packet transmission. The energy cost is b

3. If RTS, CTS packets are received successfully after that n number of DATA packets can be corrupted.The energy cost for this event is assumed c.

4. After that, if RTS, CTS, DATA packets are received successfully consequently after that n number of ACK packets can be corrupted. This is assumed as d.

5. Finally, one successful transmission of a RTS-CTS-DATA-ACK packet. The energy cost is considered e.

Hence, the energy cost can be estimated as below

$$
\begin{aligned}
a &= k_j \times L_{rts} \times p_r \times P_{col}^{rc} \\
b &= k_j \times (L_{rts} \times (1-p_r)(1-P_{col}^{rc}) + L_{cts} \times p_2) \\
c &= k_j \times (L_{rts} \times (1-p_r)(1-P_{col}^{rc}) + L_{cts} \times (1-p_r)(1-P_{col}^{rc})(1-p_c) + L_{data} \times p_3) \\
d &= k_j \times (L_{rts} \times (1-p_r)(1-P_{col}^{rc}) + L_{cts} \times (1-p_r)(1-P_{col}^{rc})(1-p_c) \\
&+ L_{data}(1-p_r)(1-P_{col}^{rc})(1-p_c)(1-p_d) + L_{ack} \times p_4) \\
e &= k_j \times (L_{rts} \times (1-p_r)(1-P_{col}^{rc}) + L_{cts} \times (1-p_r)(1-P_{col}^{rc})(1-p_c) \\
&+ L_{data}(1-p_r)(1-P_{col}^{rc})(1-p_c)(1-p_d) \\
&+ L_{ack}(1-p_r)(1-P_{col}^{rc})(1-p_c)(1-p_d)(1-p_a))
\end{aligned} \tag{6.16}
$$

The energy efficiency for the RTS-CTS scheme is estimated from equation (6.9). From figure 6.5 it is found that the energy efficiency does not decrease even after including the collision probability. Also similar to figure 6.1 the energy efficiency remains flat for larger range of packet sizes for the noisy channel and for the good channel condition the

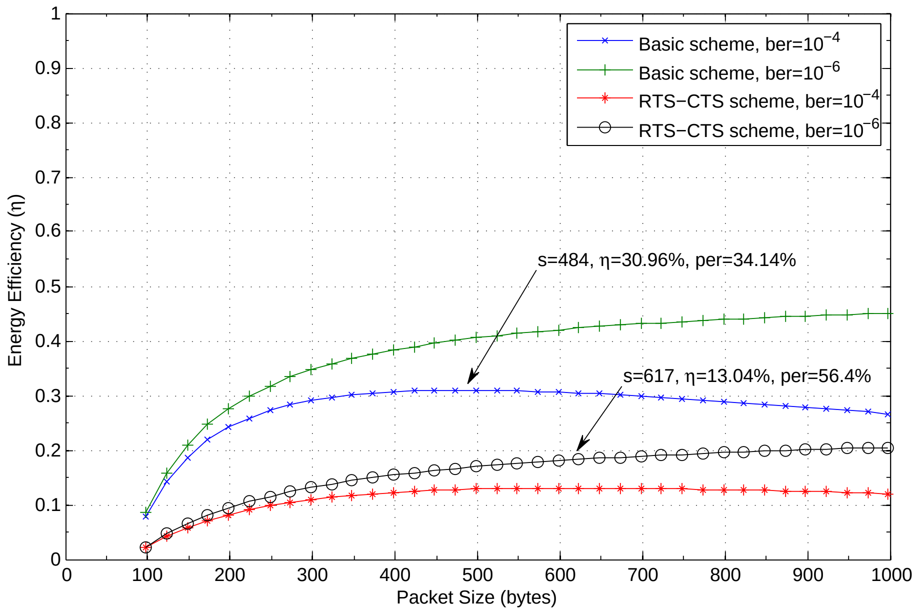

Figure 6.5: Energy efficient packet size estimation with the MAC protocol for different channel condition

optimal packet size becomes the MTU size packet.

In figure 6.1 and figure 6.5 energy efficient optimal packet size are estimated. However, from the numerical estimation model, it found that for a large range of packet size energy efficiency remains similar. Hence, this range of packet size can be used as an optimal size. To compare the numerical results with the simulated results, packet collision probability needs to investigate. In the next section the collision probability is investigated considering various parameters and different scenario.

6.5 Collision Rate Investigation for Various Packet Sizes Through Simulation

The packet collision rate has impact on the packet loss rate. Since, the packet loss rate depends on the packet size therefore packet size can be an effective parameter for packet collision. However, it is already stated by some researches that collision probability depends on the number of contending nodes within the channel, backoff window size,

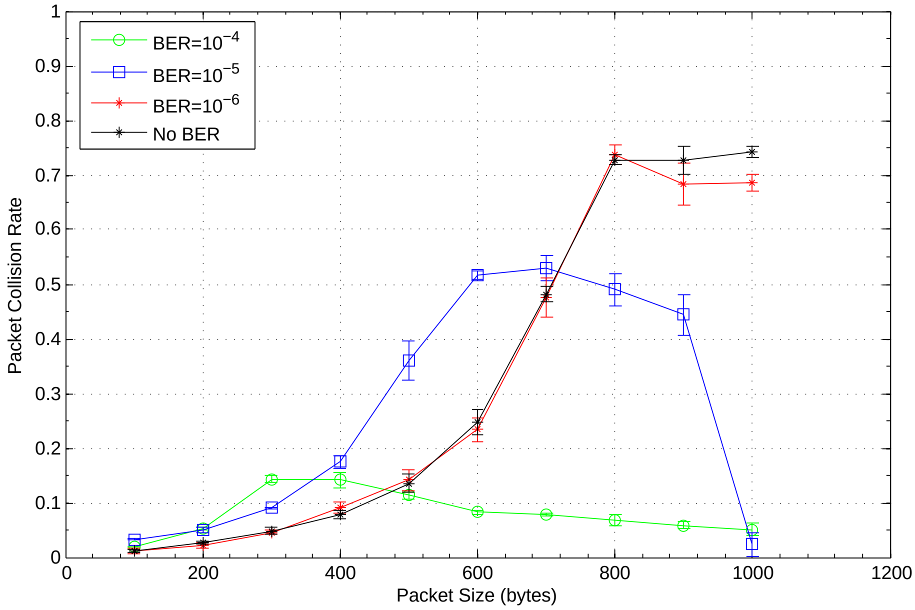

Figure 6.6: Collision probability investigation for different channel condition for RTS-CTS method with 5 nodes star topology

retry count and slot size [TC01, HT06]. Collision probability does not depend on the packet size. But in [VA08] Vuran and Akayldiz has been showed that packet size has an impact over the collision probability. Since packet size has an impact over the traffic rate and transmission period.

Packet size has a direct relation with the packet loss rate and due to packet loss in the channel the packets are retransmitted by the MAC scheduler. The retransmitted packet increased the total number of packets and therefore increases the chance of packet collision. To find out the relationship between the collision rate and the packet size in the error prone channel several simulations were done. The results are plotted in figure 6.6. In figure 6.6 the number of collisions are shown for various packets. For the good channel condition whenever BER is 10^{-6} or BER is very low (no BER), 800 bytes packet size shows the highest collision rate. After that the collision rate starts to decrease or become flat. The reason is, 800 bytes or more than 800 bytes packet size, the channel become congested. In the congested channel there is not more packets to collided with. In this experiment the channel bandwidth is considered $2Mbit/sec$ and fixed packet interval

0.036 is used. Hence, it generates 27 packets per second. For the higher bandwidth there will be very low collision rate and changing the packet interval rate increase or decrease the bandwidth utilization. For higher interval rate collision probability increases. But at the same time maximum channel utilization level is reached very quickly and for the lower rate it will show the opposite characteristic. Therefore, the bandwidth and packets per second generation rate is chosen such a way that channel saturation become visible clearly for different packet size.

Since the packet interval rate is fixed, whether it is smaller or bigger packets, same number of packets generate by the nodes. As a result, for the bigger packets the channel become congested. Due to the channel congestion extra packets can not be transmitted, it remains in the nodes' queue. Hence, whenever the packet size become 800 bytes or more, the number of collision become saturated.

In the noisy channel condition when BER is 10^{-5}, the highest collision rate is shown by 700 bytes packet size. After that it starts to decrease sharply. Because whenever there is packet loss due to the noisy channel, the nodes go to backoff stage for each packet loss. As the packet size increases in the noisy channel, the PER increases gradually. If there is more packet loss, the nodes remains most of time in backoff stage and the chance of a collision happening are very low. Therefore, for the larger packets the collision rate become low.

For BER 10^{-5} collision rate is higher than BER 10^{-4}. The reason is with 10^{-4} BER the PER becomes so high that the nodes remains most of the time in backoff stage and therefore there is less packets to collide with. For 10^{-5} BER the saturation is reached at 600 bytes because of excessive packet loss and, therefore after that, increasing the packet size only decreases the packet collision rate. Hence, one observation from this figure is that whenever the PER is high the collision rate is low and for low PER the collision rate is high.

Another observation is collision rate increases for the high data transmission rate

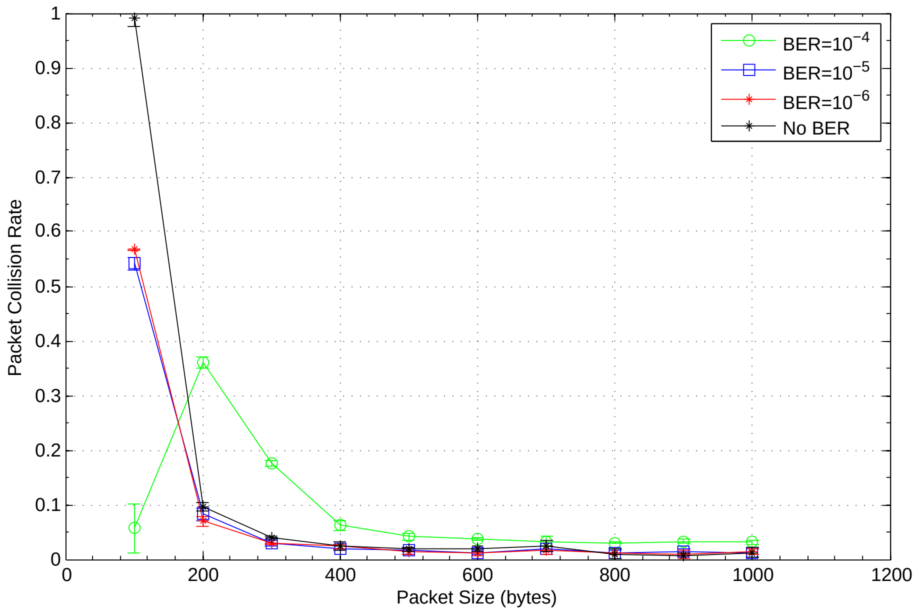

Figure 6.7: Collision probability investigation in different channel condition with 40Kb data rate for RTS-CTS method with 5 nodes star topology

within the congested channel with less BER. Hence, if fixed data rate is used in that case for the larger packets the collision rate should be less than the smaller packet sizes. Because to achieve the given data rate with the smaller packet sizes the nodes need to send large number of packets. More number of packets will increase the probability of packet collision.

To estimate the collision probability with fixed data rate for various packet size experiment is done and results are shown in figure 6.7. From figure 6.7 it is found that for the high BER 10^{-4}, since the PER becomes very high, as a consequence the nodes remains most of the time in back off state and the collision rate becomes very low. But, for the smaller packet sizes the collision rate is very high. The collision rate follows linear trend for bigger packet size more than the 400 bytes. Therefore, for a fixed data rate application, if 400 bytes or more than 400 bytes packet size is used, then it would generate maximum 5% collision rate only. Hence, it can be stated that more than 400 bytes packet sizes are more efficient in terms of collision rate.

Hence, we have found that PER has an inverse impact over packet collision rate.

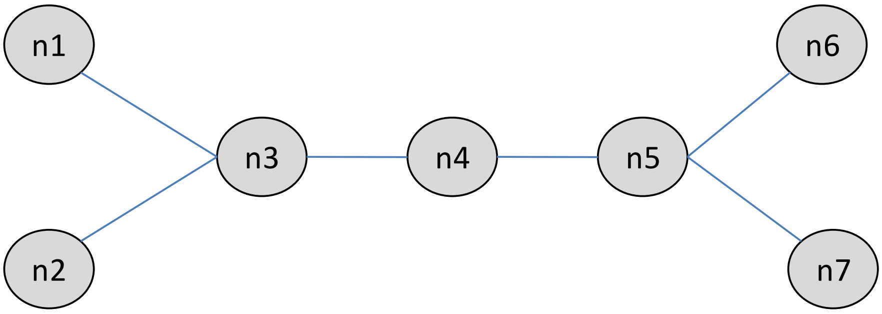

Figure 6.8: A dumbbell topology to observe the packet collision and packet corruption

Lower PER increases the collision probability and higher PER decreases the collision probability. In a fixed data rate condition smaller size packets generate high collision rate (figure 6.6) whereas for the fixed packets per second condition bigger packets generates high collision rate (figure 6.7). Since, packet size has a direct relation with the PER. Therefore, packet size has a relation with collision probability.

6.6 Simulation Based Optimal Packet Size Estimation for Multiple Hops and Multiple Source Scenario

For the linear chain multi-hop scenario energy efficient optimal packet size is estimated in section 4.4 in chapter 4. However, that estimation only includes single source multi-hop transmission without MAC scheduler. In this section optimal packet size is estimated for multiple sources for the multi-hop scenarios with the MAC protocol. To estimate an optimal packet size for multi-hop multi source scenario a simulation based distinct approach is shown.

A dumbbell topology is created like figure 6.8. In total 7 nodes and 3 sources. Two nodes (node n1 and n7) are communicating at the longest distance and at the middle, there is one pair of node (node n4 and n5) communicating with each other. Each source node is attached with a poisson source UDP traffic. The channel bandwidth is limited to $2Mbit/sec$ and data rate is fixed. Hence, during the transmission between the nodes, channel become congested as the packet size increases.

Since it is showed in the previous chapter that in the congested channel, collision rate reaches to its maximum then it starts to deteriorate. Through this experiment for various packet sizes collision rate is investigated for multi-hop scenario. After that for the same network packet error rate is estimated for different packet sizes. It is already explained in the previous sections that PER and collision rate shows opposite characteristics for various packet sizes. Hence, if a chart is plotted against packet size for packet collision and packet corruption, this two types of packet loss rate should intersect each other and we can get an optimal packet size where packet loss rate for collision and corruption will be low.

From the experiment results a chart is drawn in figure 6.9. From the simulation results from figure 6.9 it is found that roughly 200 to 400 bytes packet sizes are the most efficient packet size and have low packet loss considering packet collision and corruption.

From the previous literature study, we have found that in [VA08] Vuran and Akyildiz and in [TC01] Wang and Yin estimated optimal packet size in a congested channel 250 bytes and 400 bytes. Hence, our simulation based result matches with the previous research.

6.7 Conclusion

Since 802.11 MAC is a well known scheduling protocol, optimal packet size is investigated for multiple source ad hoc networks for 802.11 WLAN. It is found that for the good channel condition bigger packet sizes are more energy efficient whereas for the noisy channel condition the energy efficiency become flat for a longer range of packet sizes. Hence, using bigger packet sizes more than 400 bytes can be more efficient.

It is found that collision probability does not depend on the packet size directly but it has an influence over the packet loss rate. Therefore, it can be stated that packet

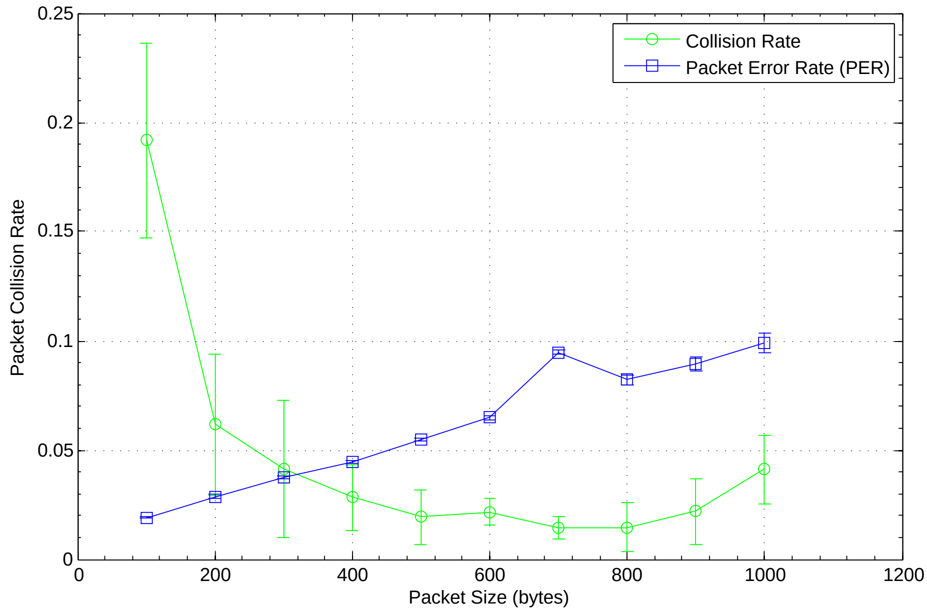

Figure 6.9: Collision probability investigation with 40Kb data rate for RTS-CTS method with 7 nodes dumbbell topology

size has a relation with the data collision rate. In this chapter along with the multiple sources multi-hop scenario is also analyzed. From simulation results, optimal packet size is investigated for multi-hop scenarios. Our investigation indicates that approximately 200-400 bytes packet size is the most efficient packet sizes in terms of energy and packet loss.

References

[Bia00] G. Bianchi. Performance analysis of the IEEE 802.11 distributed coordination function. *Selected Areas in Communications, IEEE Journal on*, 18(3):535–547, 2000.

[CCG00] F. Calì, M. Conti, and E. Gregori. Dynamic tuning of the IEEE 802.11 protocol to achieve a theoretical throughput limit. *IEEE/ACM Trans. Netw.*, 8(6):785–799, December 2000.

[HT06] L.V. Hai and S. Taka. Collision probability in saturated IEEE 802.11 networks. In *In Australian Telecommunication Networks and Applications Conference*, 2006.

[JXA05] Y. Jun, W. Xiaodong, and D.P. Agrawal. Energy efficiency evaluation of wireless LAN over bursty error channel. In *Global Telecommunications Conference, 2005. GLOBECOM '05. IEEE*, volume 6, pages 5 pp.–3632, 2005.

[MCBT05] H. Manshaei, G.R. Cantieni, C. Barakat, and T. Turletti. Performance analysis of the IEEE 802.11 MAC and physical layer protocol. In *World of Wireless Mobile and Multimedia Networks, 2005. WoWMoM 2005. Sixth IEEE International Symposium on a*, pages 88–97, 2005.

[SYHH09] S. Sayed, Y. Yang, G. Haiyou, and H. Honglin. Energy efficiency analysis of cooperative access with relay's data algorithm for multi-rate WLANs. In *Personal, Indoor and Mobile Radio Communications, 2009 IEEE 20th International Symposium on*, pages 1532–1536, 2009.

[SYHH10] S. Sayed, Y. Yang, G. Haiyou, and H. Honglin. Analysis of energy efficiency of a busy tone based cooperative MAC protocol for multi-rate WLANs. In *Wireless Communications and Networking Conference (WCNC), 2010 IEEE*, pages 1–6, 2010.

[TC01] Y.C. Tay and K.C. Chua. A capacity analysis for the IEEE 802.11 MAC protocol. *Wirel. Netw.*, 7(2):159–171, March 2001.

[TSKC96] H. Tien-Shin and C. Kwang-Cheng. Performance analysis of IEEE 802.11 CSMA/CA medium access control protocol. In *Personal, Indoor and Mo-*

bile Radio Communications, 1996. PIMRC'96., Seventh IEEE International Symposium on, volume 2, pages 407–411 vol.2, 1996.

[VA08] M.C. Vuran and I.F. Akyildiz. Cross-layer packet size optimization for wireless terrestrial, underwater, and underground sensor networks. In *INFOCOM 2008. The 27th Conference on Computer Communications. IEEE*, pages 226–230, 2008.

[XA04] W. Xiaodong and D.P. Agrawal. Analysis and optimization of energy efficiency in 802.11 distributed coordination function. In *Performance, Computing, and Communications, 2004 IEEE International Conference on*, pages 707–712, 2004.

Chapter 7

Energy Efficient VoIP Packet Size

In this chapter we investigate, without degrading the QoS, how much energy efficiency can be achieved by optimizing packet size for VoIP. FEC and compression are also taken into consideration to estimate packet size optimization. VoIP service can run on different wireless networking technologies from traditional cellular network to ad hoc network. We are particularly interested in WLAN ad hoc network.

7.1 Voice over IP (VoIP)

VoIP [Goo02, DPG00, Com10] is a revolutionary technology and the market of this service is growing very fast. A study estimates that there are approximately 110 million VoIP handset phone subscribers in the world [Pau10]. Another estimation says that the number of wireless (mobile) VoIP users will exceed 100 million by 2013 [Com10]. The main reason is this VoIP provides the cheapest price for calling. This service is run by desktop based software applications (such as: Skype [Lim10], Google Talk [Goo10], Vonage [Von10] etc) or any other VoIP handset phone [Tea11] over wired or wireless network.

Table 7-A: Delay Standard Specifications for VoIP

Delay Range	Description of the service quality
0 - 150 ms	Considered as a good quality service. Acceptable for most VoIP user applications.
150 - 250 ms	Acceptability provided that administrators are aware of the transmission delay and impact on quality.
250 - 400 ms	Severe service interruption and unacceptable.

7.1.1 Challenges of WLAN VoIP

By using the wireless medium, the users can move freely and use the VoIP service via wireless networks. Compared to the wired network, wireless VoIP provides more flexible service. However, it has some drawbacks. One of the main problem of wireless VoIP is QoS, another is the energy constraints of wireless ad hoc nodes [WSL05, CXSM06, SV10, And09].

7.1.2 QoS

Quality of service (QoS) measures the service based on different parameters like delay, jitter, packet loss, throughput etc. Since the wireless channels are noisy, QoS is not as good compared with the wired networks. Hence, QoS improvements for VoIP is a key research for the network researchers. Moreover, it is very import to ensure that voice packets are not dropped or lost during the transmission. Some QoS constraints of wireless VoIP are described below.

Delay: VoIP is a time sensitive technology. Voice packets must arrive at their destination without interruption and on time. If the end to end delay rises, talkers and listeners become un-synchronized, and often they speak at the same time, or both wait for the other to speak. The International Telecommunication Union (ITU) defines three bands of standard for one-way delay that is shown in Table 7-A [Com]. The overall delay contains the total one way communication time from bit generation to reaching to the destination node. It contains voice encoding and digitization of the analog voice, payload generation, serialization of the bits and packet formation by adding the headers at different layers. For good quality of VoIP the total delay should be less than 150ms. One example of delay estimation is shown in table 7-B [Com]. In this example, packet is transmitted from VoIP phone to router (see figure 7.1) and then toward the network clouds where the packet reaches to the destination via different routes.

A Short description of different delays are described below [Com]:

- **Coder delay:** Coder delay is actually the delay caused to process the voice by the processor and CODEC. It is product of the algorithmic delay, encoding time per block and decoding time per block multiplied with number of blocks in frame. G.729 CODEC generates sample payload size of 10 bytes within 10 ms interval. For an example, if the payload size is 30 bytes then the encoding delay will be 10ms and decoding time 1ms per block. Hence, decoding time will be 3ms. According to the example given in [Com], the algorithmic delay for G.729 CODEC is 5ms. Hence, the total coder delay becomes 18ms for 30 bytes of 30ms voice coding.

- **Packetization delay:** This delay is the time taken to fill a packet payload with encoded/compressed speech. It is a function of the sample block size required by the encoder and the number of blocks placed in a single frame.

- **Serialization delay:** The time is taken to place a packet on the physical medium for transmission. For the high speed LAN serialization delay is as low as 1ms. But

Table 7-B: Overall Delay Estimation Example for G.729 Codec

Delay Type	Fixed (ms)	Variable (ms)
Coder	18	
Packetization	30	
Queue/Buffering		8
Serialization	5	
Network	40	25
De-jitter Buffer	45	
Total Delay	138	

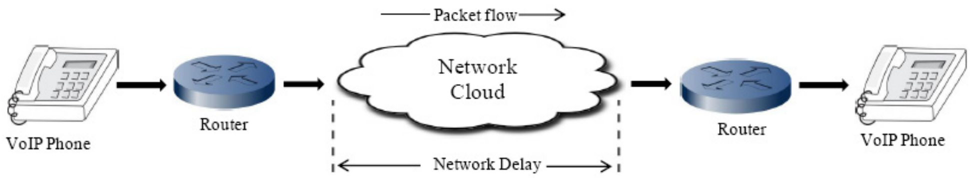

Figure 7.1: An instance of a network to estimate the VoIP delay

for the low rate LAN serialization delay cost is higher. WLAN can support high data transmission rate hence it is reasonable to assume very low delay $1ms$ only.

- **De-jitter Buffer delay:** The de-jitter buffer transforms the variable delay into a fixed delay. It holds the first sample received for a period of time before it plays it out.

- **Network delay:** The network delay contains several delays such as channel contention delay, routing delay, propagation delay etc. In the example given in table 7-B the packets have been forwarded from source node to the destination via network clouds. The network delay is assumed $45ms$. However it will be different for different network scenarios. In our case the network delay is estimated separately.

Another QoS factor is the packet loss. Since, the wireless channel is frail to noise. During the noisy channel condition, the packet loss rate becomes higher than the wired network. A small percentage of packet loss ensures a high quality of VoIP. According to Karapantazis and Stylianos [KP09] and from other research papers [WSL05, CXSM06, SV10, And09] it is estimated that the packet loss rate for the VoIP should not be more than 3%.

7.1.3 Energy Constraints

VoIP service can run on different technologies such as: wired or wireless, WiFi or cellular. Due to the availability and the high usages we are interested in the WLAN WiFi networks. WLAN WNIC is a highly resource consuming component. To run the VoIP service for longer time continuously energy efficiency is crucial.

After maintaining the QoS the WLAN VoIP service needs to be energy efficient. To improve the energy efficiency several approaches have been proposed [JSAC01]. But specifically for the wireless VoIP service energy efficient methods are very few [NG10, CSE04]. Most of the research major concern for VoIP technology was about Quality of Service (QoS) [Har03]. Energy awareness has not been a major research concern for VoIP in WLANs. Among the few researches for the energy efficient WLAN VoIP, the sleep and wake up algorithm has been published in [NG10, NG08]. The basic strategy is to turn on and off the WNIC card and keep the most power consuming component in sleep mode as much time as possible to save energy. Hence the energy consumption by the WNIC will be less during the idle period. But it is difficult to predict when exactly the node has to wake up to receive packets. If the node does not wake up at the right time the service will be interrupted and as a consequence the technique would be worthless for VoIP. A standard solution Power Saving Mode (PSM) protocol has been proposed in [NG10, IEE99] where PSM allows the node to go to lower power sleep mode while it is not transmitting or receiving any packet and during that time access point (AP) [IEE99] buffers the packet destined for the sleeping node. So, their [NG10] presumption was that AP is reconfigurable and can be modified which is not practical in real case [LDL10]. In order to save energy the authors of [ACW+07, LDL10] have shown that powering off the WNIC and wake it up increases the energy efficiency significantly. But their proposed method is relied on the traditional GSM network [LDL10] and 3G technology [ACW+07]. Hence, in their estimations [ACW+07, LDL10, NG10] ad hoc network is actually dependent on traditional infrastructure [ACW+07, LDL10] or fixed AP [NG10]. Therefore, ad hoc network in the infrastructure-less environment such as temporary VoIP network for collaboration between volunteers for disaster recovery or in the battle field these energy efficient methods would not be compatible. Hence an optimization is needed to guarantee standard QoS and increase the energy efficiency for the wireless VoIP networks. In this thesis optimized packet size is estimated for the wireless VoIP to increase energy efficiency without degrading the QoS.

7.2 Energy Cost Estimation of a Single VoIP Packet

Energy consumption for a single bit is estimated in this section. Our analysis starts with the assumption that two VoIP users are communicating with each other in a WLAN. The nodes are within their communication range. The energy consumption to transmit a VoIP packet of m bits over distance d, can be expressed as [ZR04]

$$E_L(m,d) = E_{tx}(m,d) + E_{rx}(m,d) + nE_{codec}(m) \tag{7.1}$$

where, E_{tx}, E_{rx} are the radio transceiver and receiver energy consumption and E_{codec} represents the CODEC energy required to encode and decode one packet respectively. n is the number of CODEC sample block [NNP10] within a packet. Based on the radio power consumption expressed in [SAM03, KS06, KSPR04] energy (E_b) is required to communicate one single bit across a single hop is given by

$$E_b = E_{tx} + E_{rx} + \frac{E_{codec}}{\ell} \tag{7.2}$$

where ℓ represents the size of the payload (bits) of a VoIP packet for a single block (here $n = 1$). To estimate the energy cost of E_b in equation (7.2) energy cost for the E_{tx}, E_{rx} and E_{codec} are needed to estimate. E_{tx} and E_{rx} are simplified later in this section.The energy cost for the VoIP CODEC E_{codec} is estimated according to the following paragraph.

The authors of [NNP10] Naeem and Vinod have been estimated the energy consumption of G.711 CODEC is 45.56 mWh (per minute) for Lenovo SL400 Laptop. G.711 CODEC rate is 64 $Kbps$ and sends 50 $packets/sec$ in average. Hence, from their estimation the energy cost of G.711 CODEC E_{codec}/bit is estimated 3.3×10^{-12} J/bit. Similarly the energy cost of G.729 (8Kbps) and G.723 (6.3Kbps) CODECs are estimated 2.35×10^{-11} J/bit and 2.63×10^{-11} J/bit respectively. However, in this chapter all the analysis is estimated only for the G.711 CODEC. Because this CODEC generates the largest sample payload within short time interval from input voice. Therefore, the overhead of this CODEC is low (see section 2.2.1 of chapter 2) comparing to the other CODEC. The energy per bit E_b of equation (7.2) can be simplified as below

$$\begin{aligned} E_b &= E_{tx} + E_{rx} + \frac{nE_{codec}}{\ell} \\ &= \frac{1}{\ell}\left((P_{tx})\frac{L}{R}\right) + \frac{1}{\ell}\left((P_{rx})\frac{L}{R}\right) + \frac{nE_{codec}}{\ell} \\ &= \frac{L}{\ell}\left(\frac{P_{tx}+P_{rx}}{R}\right) + \frac{nE_{codec}}{\ell} \\ &= k_j\left(\frac{\alpha+\ell+\tau}{\ell}\right) + \frac{nE_{codec}}{\ell} \\ &= k_j + k_j\left(\frac{\alpha+\tau}{\ell}\right) + \frac{nE_{codec}}{\ell} \end{aligned} \tag{7.3}$$

Where,

$$k_j = \frac{(P_{tx}+P_{rx})}{R} \tag{7.4}$$

The value of k_j is estimated from section 4.2 of chapter 4.

7.3 Energy Efficiency Metric for VoIP Packet

For a good channel condition in a specific time, if there is no packet loss then similar to equation (7.5) of chapter 4 the energy efficiency metric (η_v) for WLAN VoIP can be expressed as

$$\eta_v = \frac{k_j\ell}{k_j(\ell+\alpha) + nE_{codec}} \tag{7.5}$$

Since the codec processing cost (E_{codec}) is small, larger packets will be more energy efficient. Because for the larger packets, $k\ell$ becomes higher (see section 4.2 of chapter 4). But, larger packet sizes are frail to noise and increase delay. Whereas small

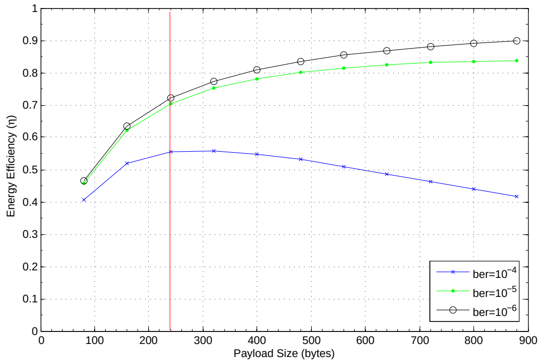

Figure 7.2: Energy efficiency estimation for WLAN VoIP in different channel condition

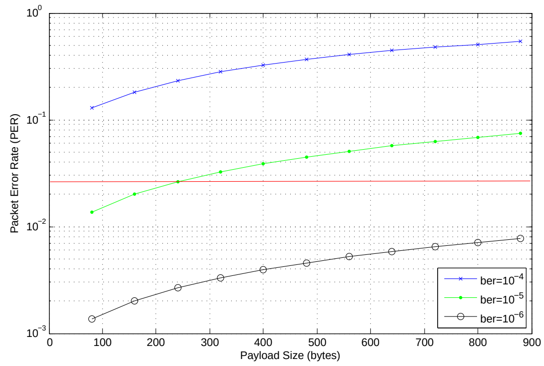

Figure 7.3: PER estimation for WLAN VoIP in different channel condition

packet sizes are comparatively robust with channel noise but consume higher energy due to high overhead. Since the usual wireless channel always has some noise during the communication. The larger packet sizes have higher probability of being corrupted. The corrupted packets might be lost or dropped by the destination node. That causes call drop or unwanted noise in VoIP communication. So, a trade-off is needed to find out the energy efficient optimal packet size that does not degrade the service quality and at the same time increase energy efficiency. Energy efficient packet size and channel error relation is given in section 4.2.1 of chapter 4. Considering the channel noise and bit error rate, the energy efficiency metric (η) can be expressed as below

$$\begin{aligned} \eta &= \frac{k_j \ell}{k_j(\ell + \alpha) + nE_{codec}}(1-p) \\ &= \frac{k_j \ell}{k_j(\ell + \alpha) + nE_{codec}}(1-b)^L \end{aligned} \tag{7.6}$$

where all the expression (p, b, ℓ, α) in equation (7.6) are similar to equation (4.3) and (3.1). Here, p and b represents the PER and BER respectively. In figure 7.2 and 7.3 energy efficiency and PER are plotted for various payloads of VoIP considering equation (7.6). The red line in these figures determine that more than the marked level higher payload sizes breach the QoS, 3% PER. In figure 7.3 for 10^{-5} BER only 240 bytes is the maximum limit, more than 240 bytes the PER becomes more than 3%. With 10^{-6} very high payload sizes can be sent and PER remains less than 3%. Whenever channel BER is 10^{-4}

there will be highly service interruption due to high packet loss rate. Hence, considering the QoS PER limit for an average channel condition almost 70% energy efficiency can be achieved and optimal packet size is 240 bytes.

7.3.1 Delay Budget and Energy Efficiency

VoIP is a delay sensitive application. For the larger packet size delays become higher. The larger packets take more time for generation, packetization, transmission etc. As a result overall delay increases. The International Telecommunication Union (ITU) defines three bands of standard for one-way delay that is shown in table 7-A. One example of overall delay estimation for the G.729 CODEC is given in table 7-B. But the given example is for the wired infrastructure. The packet is transmitted from VoIP phone to router and then toward the network clouds where the packet might be reached to the destination via different routes. Whereas in our estimation it is assumed that the packet is transmitted over the wireless channel and nodes have the routing capability to forward the packet to the destination. Hence, the coding, packetization, algorithm delays still can be comparable but the network delay estimation will be different.

Network delay is estimated form the total delay of packet propagation and transmission delay [GL96]. If the channel is considered contention free then packet propagation delay can be estimated from (distance between the nodes/speed of light) that gives very low delay. The transmission delay [GL96] can be estimated from L/R where L is the packet size in bits and R is the transmission rate (bits/sec). So transmission delay will be varied according to the payload sizes.

In the CODEC table 2-E (in chapter 2) it is stated that different CODECs generate different sample payload size. Now, the G.711 CODEC generate 80 bytes of sample payload for $10ms$ voice such as: 80, 160, 240, 320, 400, 480, 560 bytes of voice payloads with an interval of 80 bytes. All the other CODECs generate lower size of sample payload. Hence, using the G.711 CODEC large payload size can be obtained in short

Table 7-C: Delay estimation for G.711 CODEC for various payloads

Payload Size (bytes)	80	160	240	320	400	480
Codec delay (ms)	16	27	38	49	60	71
Packetization delay (ms)	10	20	30	40	50	60
Network delay (ms)	16	23	30	38	45	52
De-jitter Buffer delay (ms)	45	45	45	45	45	45
Total delay (ms):	87	115	143	172	200	228

N.B. The fraction values are rounded off to the nearest milliseconds.

period. Therefore, by using the G.711 CODEC higher energy efficiency can be achieved and delay will be less. Because for the large payloads output energy will be higher (see section 4.2 of chapter 4).

In table 7-C, for different payloads delays are estimated for G.711 VoIP CODEC considering Ethernet data rate $87.2Kbps$. In this table, serialization delay is not included. As WLAN allows high data transmission rate and for the high transmission link have negligible serialization delay (less than 1 ms) [Com]. From the delay estimation table 7-C it is found that more than 240 bytes the delay becomes more than $150ms$ and breaches the QoS. Hence, from figure 7.2 and table 7-C we can conclude 240 bytes is the optimal packet size for VoIP. Because only 240 bytes is the maximum payload that maintains the PER and delay constraints of VoIP QoS.

7.3.2 FEC and Energy Efficient VoIP Packet Size

In wireless networks packets are lost due to either excessive bit errors caused by unwanted noise, or congestion in the IP network, or simply excessive delay that cause the receiver to ignore the corresponding packets. Packets are corrupted due to channel noise and the bits inside the packet are altered or can be lost totally. In order to correct errors in VoIP system, there exist the two distinct approaches ARQ (Automatic Repeat Request) and FEC (Forward Error Correction). In chapter 4 it has been showed that the ARQ technique is an inefficient system in terms of energy. Whereas FEC is revealed as an energy efficient method for very large packet size or large multi-hop network without

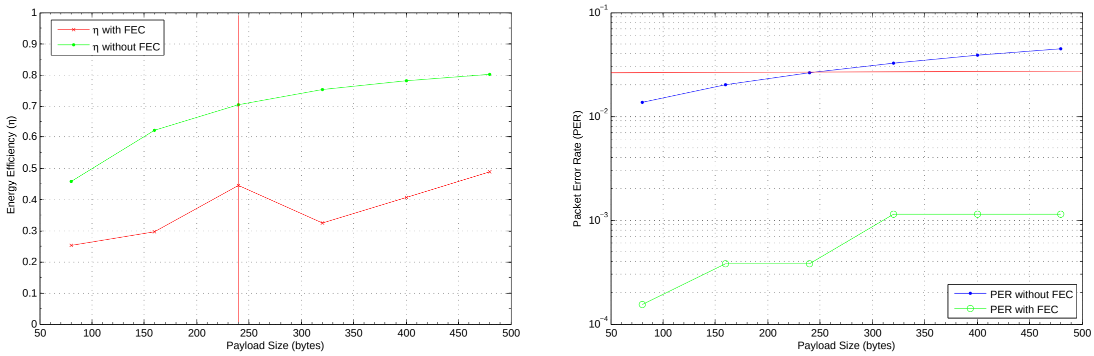

Figure 7.4: Energy efficiency and PER comparison for WLAN VoIP with FEC and without FEC when BER=10^{-5}

considering delay. Hence, this section illustrates is it possible to use the FEC for VoIP to increase energy efficiency and also delay will be kept within the tolerance level. FEC also increases delay to generate extra error correction code, packetization, code transmission, error correction etc. Before starting to estimate the delay for the FEC, the energy efficiency is estimated at first. Recalling the equation (5.4) from chapter 4 the energy efficiency η for the VoIP packet can be rewritten as below

$$\eta = \frac{k_j \ell}{k_j(\ell + \alpha + \tau) + nE_{codec} + E_{dec}} \times \sum_{i=0}^{t} \binom{L}{i} b^i (1-b)^{L-i} \tag{7.7}$$

From equation (7.7) energy efficiency and PER are estimated in figure 7.4. Figure 7.4 shows that with FEC energy efficiency decreases and lower than without FEC for WLAN VoIP. Since, considering the QoS constraints it is found that most energy efficient packet size is 240 bytes. In figure 7.4 red line is drawn to mark the level of 3% packet loss constraints of VoIP QoS and 240 bytes.

The right hand side graph shows that with the FEC, PER becomes very low compared with optimal packet size without FEC. But with the FEC, only 44% efficiency can be achieved whereas without using the FEC 25% more energy efficiency can be obtained. The reason is, if we consider one block of voice payload of G.711 CODEC, then sample payload will be 80 bytes (640 bit). Considering the block length constraints of FEC

($n = 2^m - 1$) the value of m need to be 10 or $m \geq 10$. As a result, the block size will be ($2^{10} - 1 = 1023$). Hence, if k_j is the voice payload and n is the total packet size after the FEC code then the extra overhead will generate $(n - k) = 383$ that means 37.4% overhead. Hence, this huge overhead reduces the energy efficiency. Additionally, FEC includes extra overhead for error correction. As a consequence efficiency drops from 70% to 44%. Since, in this thesis our main aim is to increase the energy efficiency whereas FEC has shown poor performance. Therefore, delay is not estimated for FEC further.

7.4 VoIP Packet Compression and Energy Efficiency

Compression involves mathematical algorithms that encode the original packet into a smaller string of bytes. After sending the smaller encoded packet, the decompression algorithm on the other end of the link reverses the process, revert the packet back to its original state. The radio transceiver and receiver needs less bits to transfer which reduce the energy cost. Moreover, this process increases the chance of utilizing more bandwidth. As the compressed packet size become smaller than its original form so PER become less (see section 4.2.1 of chapter 4) as a consequence energy efficiency should be increased.

It has been showed in Cisco website [Com12], header compression method increases efficiency of VoIP service. Because, the packet header consumes a large amount of portion in the packet (see chapter 2). It increases the cost for packet transmission. Hence, by using the compression if the overhead size become less then packet controlling cost will be lower and that will increase efficiency. In this section header and whole packet compression is described in terms of energy efficiency and delay.

7.4.1 Energy Cost of Compression Decompression Method

Compression decompression algorithm adds extra delay and processing energy cost. Hence, the compression technique will be only advantageous if it is found that the overall

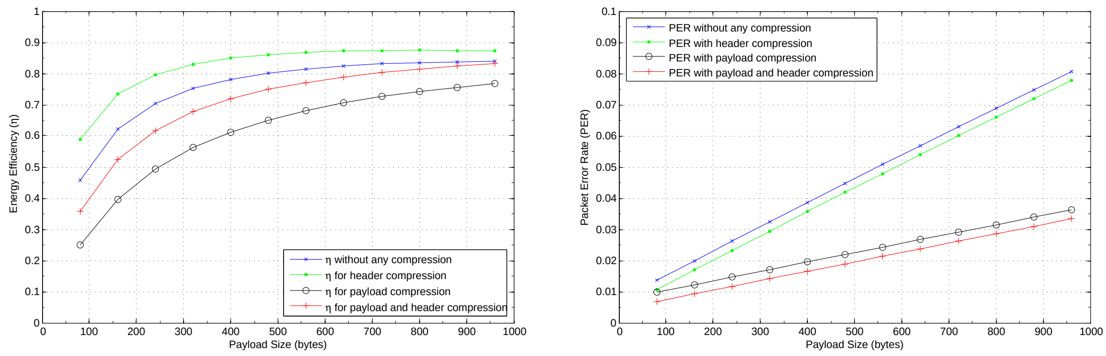

Figure 7.5: Energy efficiency and PER comparison for compression with BER 10^{-5}

delay is within the delay limit (see table 7-A of chapter 2) and the processing energy cost of compression is lower than the amount of energy saved by the compression technique. Because when the packet will be compressed the energy cost for the packet transmission will be lower than without compression. But compression algorithm will add processing energy cost too to compress the packet. To estimate the processing energy cost of the compression algorithm, LZO compression algorithm [BA06] is considered (see chapter 2.6 for details). According to [BA06] LZO is a well known efficient compression algorithm that needs 3 instructions for compressing and decompressing one bit. (two instructions are for the compression and one is for the decompression). It is showed that a 233MHz Strong ARM SA-110 processor for one add instructions consumes $6.59 \times 10^{-9}J$ in total including CPU, memory and peripheral computation energy cost. From their estimation it is found that the energy to compress and decompress only one bit will be

$$E_{comp} = 1.98 \times 10^{-8}J \tag{7.8}$$

where E_{comp} is the total compression decompression energy cost.

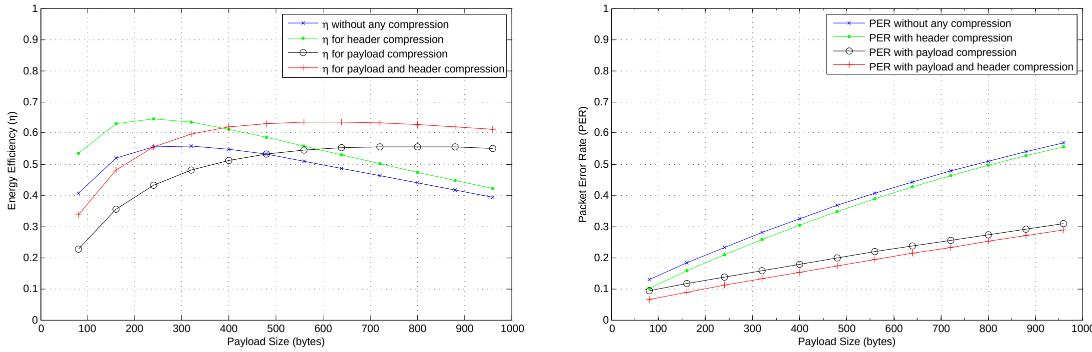

Figure 7.6: Energy efficiency and PER comparison for compression with BER 10^{-4}

7.4.2 Energy Efficient VoIP Packet Size Estimation with Compression

After including the energy cost for the compression decompression the energy efficiency (η_{com}) will be

$$\eta_{com} = \frac{k_j \ell_{com} \times (1-b)^{L_{comp}}}{k_j L_{comp} + nE_{codec} + E_{comp} L_{comp}} \tag{7.9}$$

where L_{comp} is the compressed packet size, $L_{comp} = (\alpha_{comp} + \ell_{com})$ where α_{comp} is the compressed header size and ℓ_{com} is the compressed payload size. By using the compression technique 40 bytes of IP, UDP, and RTP headers can be reduced to 2-4 bytes [Com12] (see section 2.10 of chapter 2.6). If only header compression is considered then energy efficiency (η^h_{com}) for the compressed header will be

$$\eta^h_{com} = \frac{k\ell \times (1-b)^{L^h_{comp}}}{k_j L^h_{comp} + nE_{codec} + E_{comp}\alpha_{comp}} \tag{7.10}$$

In this equation (7.9) L^h_{comp} is the total packet size with the compressed header size, $L^h_{comp} = (\alpha_{comp} + \ell)$. From equation (7.9) and (7.10) energy efficiency is estimated and plotted in figure 7.5 and 7.6.

Figure 7.5 and 7.6 each contains two graphs, the figure at left shows the energy

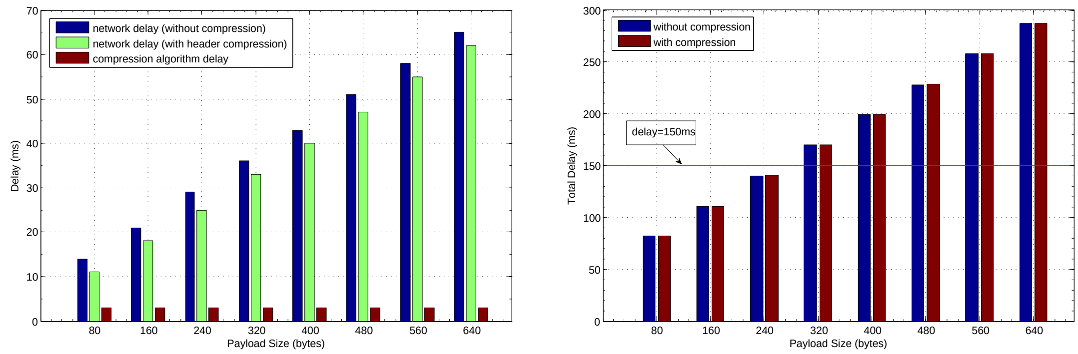

Figure 7.7: Delay comparisons between header compression and with out compression for various VoIP payload sizes

efficiency against payload sizes and another one at right shows packet error rate of corresponding payload sizes. It is found from figure 7.5 that header compression is the most energy efficient method. In a more noisy channel payload and header compression shows more energy efficiency but due to high PER (more than 3%) it exceed the QoS constraints (see figure 7.6).

Since, header compression is observed as the most energy efficient method, delay is estimated only for header compression further. For header compression technique, coding and packetization delay remain the same like without compression. Because coding and packetization delays related with payload not header. But other delays have impact on header compression such as: serialization, queuing, and network delay decreases [OC04, OC12]. However, in our estimation as high data rate WLAN is considered so serialization does not have much effect. But, the network delay will be varied. The network delay will reduce for header compression because the packet transmission time (packet size/data rate) with the short header will be lower. The figure 7.7 shows two bar graphs, the left hand side graph shows the delays of network delays and compression algorithm delays and the graph at right shows the total delays comparison between header compression and without header compression considering all the delays (including coding, packetization, de-jitter buffer delays). The figure 7.7 shows that header compression does not improve the overall delays much but due to transmission (network) delay reduction the compression algorithm delay is equalized. However, after using header compression as an advantage more energy efficiency can be achieved. Therefore, header compression method is a delay tolerant and energy efficient technique for VoIP.

7.5 Energy Efficient VoIP Packet Size for Multi-hop Network

In ad hoc WLAN the nodes do not have a high transmission range. Hence, the source node forward their packet via multiple hops to send the packet to the destination. Hence, a long chain can be formed. The intermediate nodes act like as a router to send the packet to its destination. When packets traverse multiple hops the delay time increases [Com]. Moreover, in the multi-hop route when the packets are forwarded to the destination through multiple hops the error rate changes (chapter 4) according to the number of hops and distance between them. As the VoIP service is very sensitive to the delay this imposes a restriction, how far the packets can travel via a multi-hop route. Moreover, for the VoIP service the packet error rate needs to be less than 3%.

The packet error rate and reliability function for the linear chain multi-hop topology has been given in equation (3.15) of chapter 4. Similar to that equation the energy efficiency for the multiple hops (η_{mul}^{voip}) can be rewritten for the VoIP as below

$$\eta_{mul}^{voip} = \frac{\ell(Nk_j - E_{rx}) \times (1 - PER)^N}{L \times (Nk_j - E_{rx}) + nE_{codec}} \tag{7.11}$$

where N is the number of hops traveled by the packet. Table 7-D shows the delay, energy efficiency and PER for multiple hops for the linear chain topology. From this table it is found that if VoIP packets traverse more than 2 hops then QoS will not be satisfied. Also 80 bytes is the most energy efficient packet size considering the PER and delay constrains of VoIP QoS. Because only for the 80 bytes payload, for the two nodes multi-hop linear chain, the PER remains within 3% tolerance level 3% and delay is also within the limit of $150\,ms$.

Table 7-D: Energy efficient VoIP packet size estimation for G.711 CODEC for multiple hops

Payload Size (bytes)	80	80	160
Number of hops	2	3	2
Codec delay (ms)	16	16	27
Packetization delay (ms)	10	10	20
Network delay (ms)	32	48	46
De-jitter buffer delay (ms)	75	105	75
Total delay (ms)	133	179	168
Packet Error Rate (PER)	2.7%	4%	4%
Energy efficiency (η)	45%	45%	61%

N.B. The fraction values are rounded off to the nearest milliseconds.

In section 7.3.2 it was show that FEC is not an energy efficient method for the WLAN VoIP for 80 bytes payload. If FEC is applied for the multi-hop network it is found that PER become 0.02% from 2.5% and energy efficiency become 25.4% from 49.4%. Since, the FEC constrains ($\ell = 2^m - 1$) overhead become so high that it reduces the energy efficiency. For the other payload sizes the delay become higher than the tolerance level ($150\,ms$) and does not have any scope to use FEC for multi-hop network because the delay will be increased above the $150\,ms$ limit.

7.5.1 Compression for Energy Efficient VoIP Packet Size in Multi-hop Network

From the previous section 7.4.2 it is found that header compression is an energy efficient method for the WLAN VoIP and from section 7.5, 80 bytes packet size is found as the only energy efficient packet size for the two nodes multi-hop network. Hence, if header compression is applied for the multi-hop network then energy efficiency should improve. With header compression method the energy efficiency for the multi-hop network can be expressed as

$$\eta_{h-com}^{mul} = \frac{\ell(Nk_j - E_{rx}) \times (1 - ber)^{(\alpha_{com}+\ell)N}}{L_{com}^{h}(Nk_j - E_{rx}) + NE_{com}\alpha_{rtp} + nE_{codec}} \tag{7.12}$$

Table 7-E: Header compression comparison for multi-hop network considering energy efficiency, PER and delays

Header Compression	without	with	with
Payload Size (bytes)	80	80	80
Number of hops	2	2	3
Codec delay (ms)	16	16	16
Packetization delay (ms)	10	10	10
Network delay (ms)	32	25	37
De-jitter buffer delay (ms)	75	75	105
Compression delay (ms)	0	6	6
Total delay (ms)	133	132	174
Packet Error Rate (PER)	2.7%	2.1%	3.16%
Energy efficiency (η)	45%	58.4%	57.8%

N.B. The fraction values are rounded off to the nearest milliseconds.

where, α_{rtp} is the RTP header size which is 40 bytes and α_{com} is the compressed header size which is 2 bytes. From table 7-E it is found that if packet traverse more than two hops then header compression become inefficient considering the QoS constraints of packet loss and delay limit. Moreover, for the multiple hops the processing energy cost become higher than the energy saves by header compression. Hence, for the multi-hop linear chain topology when the packet traverse more than two multiple hops, header compression become inefficient.

7.6 Conclusion

In this chapter the energy efficient packet size is estimated for WLAN VoIP considering FEC and compression methods. While estimating the energy efficiency PER and delay is also considered to maintain standard QoS. It is found that FEC method is not an energy efficient method as it includes high overhead with VoIP packet. On the other hand compression is revealed as an energy efficient method for the single hop network and for the multi-hop network. But for the multi-hop network whenever the packet size is traversed between two hops due to the increasing packet loss and delay, compression becomes an inefficient system. Hence, to improve the energy efficiency by optimizing the packet size for ad hoc network for the VoIP service is very limited.

References

[ACW+07] Y. Agarwal, R. Chandra, A. Wolman, P. Bahl, K. Chin, and R. Gupta. Wireless wakeups revisited: energy management for VoIP over WiFi smartphones. In *Proceedings of the 5th international conference on Mobile systems, applications and services*, MobiSys '07, pages 179–191, New York, NY, USA, 2007. ACM.

[And09] A. Andres. *A study of mobile VoIP performance in wireless broadband networks*. PhD thesis, Helsinki University of Technology,(Espoo, Finland), October 2009.

[BA06] K.C. Barr and K. Asanović. Energy-aware lossless data compression. *ACM Trans. Comput. Syst.*, 24:250–291, August 2006.

[Com] Cisco Technical Community. Understanding delay in packet voice networks, Document ID: 5125, February.

[Com10] Juniper Research Committee. Press release: Mobile VoIP users to exceed 100 million by 2012 finds new Juniper research report. `http://juniperresearch.com/viewpressrelease.php?pr=187`, Dec 2010.

[Com12] Cisco Technical Support Community. *Header Compression - Networking Software (IOS & NX-OS)*, June 2006 (accessed: March 03, 2012).

[CSE04] Y. Chen, N. Smavatkul, and S. Emeott. Power management for VoIP over IEEE 802.11 WLAN. In *Wireless Communications and Networking Conference, 2004. WCNC. 2004 IEEE*, volume 3, pages 1648 – 1653 Vol.3, march 2004.

[CXSM06] L. Cai, Y. Xiao, X.S. Sherman, and J.W. Mark. VoIP over WLAN: voice capacity, admission control, QoS, and MAC: Research Articles. *Int. J. Commun. Syst.*, 19:491–508, May 2006.

[DPG00] J. Davidson, J. Peters, and B. Grace. *Voice over IP fundamentals*. Cisco Press, first edition, 2000.

[GL96] J.F. Gibbon and T.D.C. Little. The use of network delay estimation for multimedia data retrieval. *Selected Areas in Communications, IEEE Journal* *on*, 14(7):1376 –1387, sep 1996.

[Goo02] B. Goode. Voice over Internet Protocol (VoIP). *Proceedings of the IEEE*, 90(9):1495 – 1517, sep 2002.

[Goo10] Google. Google talk software. `http://www.google.com/talk/`, December 2010.

[Har03] W.C. Hardy. *VoIP service quality: measuring and evaluating packet-switched voice*. McGraw-Hill, New York, 2003.

[IEE99] Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications. IEEE Standard 802.11, June 1999.

[JSAC01] C.E. Jones, K.M. Sivalingam, P. Agrawal, and J.C. Chen. A survey of energy efficient network protocols for wireless networks. *Wirel. Netw.*, 7:343–358, September 2001.

[KP09] S. Karapantazis and F. Pavlidou. VoIP: A comprehensive survey on a promising technology. *Computer Networks*, 53(12):2050–2090, August 2009.

[KS06] Z.H. Kashani and M. Shiva. BCH coding and multi-hop communication in wireless sensor networks. In *Wireless and Optical Communications Networks, 2006 IFIP International Conference on*, pages 5 pp. 1–5, 2006.

[KSPR04] H. Karvonen, Z. Shelby, and C. Pomalaza-Raez. Coding for energy efficient wireless embedded networks. In *Wireless Ad-Hoc Networks, 2004 International Workshop on*, pages 300 – 304, June 2004.

[LDL10] W. Lifu, Z. Dasheng, and M. Lei. An energy efficient WLAN Skype deployment using GSM wakeup signals. In *Green Computing and Communications (GreenCom), 2010 IEEE/ACM Int'l Conference on Int'l Conference on Cyber, Physical and Social Computing (CPSCom)*, pages 470 –473, dec. 2010.

[Lim10] Skype Limited. Skype. `http://www.skype.com/intl/en-gb/home`, December 2010.

[NG08] V. Namboodiri and L. Gao. Towards energy efficient VoIP over wireless LANs. In *Proceedings of the 9th ACM international symposium on Mobile ad hoc networking and computing*, MobiHoc '08, pages 169–178, New York, NY, USA, 2008. ACM.

[NG10] V. Namboodiri and L. Gao. Energy-efficient VoIP over wireless LANs. *IEEE Transactions on Mobile Computing*, 9(4):566–581, April 2010.

[NNP10] M. Naeem, V. Namboodiri, and R. Pendse. Energy implication of various VoIP CODECs in portable devices. In *Local Computer Networks (LCN), 2010 IEEE 35th Conference on*, pages 196 –199, oct. 2010.

[OC04] W. Odom and M. Cavanaugh. *Cisco QOS Exam Certification Guide (IP Telephony Self-Study) (2nd Edition)*. Cisco Press, November 2004.

[OC12] W. Odom and M. Cavanaugh. *Foundation Topics Payload and Header Compression*, November 2004 (accessed: March 03, 2012).

[Pau10] T. Paul. Hard VoIP users top 100 million. `http://www.theinquirer.net/inquirer/news/1593216/hard-voip-users-100-million`, December 2010.

[SAM03] Y. Sankarasubramaniam, I.F. Akyildiz, and S.W. McLaughlin. Energy efficiency based packet size optimization in wireless sensor networks. In *Sensor Network Protocols and Applications, 2003. Proceedings of the First IEEE. 2003 IEEE International Workshop on*, pages 1 – 8, 2003.

[SV10] K.O. Stoeckigt and H.L. Vu. VoIP capacity analysis, improvements, and limits in IEEE 802.11 Wireless LAN. *Vehicular Technology, IEEE Transactions on*, 59(9):4553 –4563, nov. 2010.

[Tea11] Cisco Technical Team. Cisco Unified Wireless IP Phone 7920 Version 3.0. `http://www.cisco.com/en/US/prod/collateral/voicesw/ps6788/phones/ps379/ps5056/product_data_sheet09186a00801739bb.html`, Dec 2011.

[Von10] Vonage Broadband Phone Company. Vonage. `http://www.vonage.co.uk/`, December 2010.

[WSL05] W. Wei, C. Soung, and V.O.K. Li. Solutions to performance problems in VoIP over a 802.11 wireless LAN. *Vehicular Technology, IEEE Transactions on*, 54(1):366 – 384, jan. 2005.

[ZR04] M. Zorzi and R.R. Rao. Coding tradeoffs for reduced energy consumption in sensor networks. In *Personal, Indoor and Mobile Radio Communications,* *2004. PIMRC 2004. 15th IEEE International Symposium on*, volume 1, pages 206 – 210 Vol.1, sept. 2004.

Chapter 8

Conclusions and Future Work

Efficient usage of energy is a desired need to extend the service time of wireless devices. Driven by this challenge, optimal packet size is proposed to increase energy efficiency. Optimal packet size increases the energy efficiency. But this efficiency depends on various parameters such as: channel condition, number of nodes and hops, and applications etc. Hence, it is not possible to estimate a fixed optimal packet size for different scenarios.

This research analysed diverse ad hoc network scenarios and constraints to estimate the energy efficient optimal packet size for: fixed and time varying channel condition, single hop and multi-hop, QoS sensitive and delay tolerant applications, FEC and compression methods etc.

8.1 Conclusions

Without considering any specific circumstances MTU is the optimal packet size. But wireless channel condition might fluctuate and consequently the optimal packet size also changes. Hence, for a time variable channel, instead of a fixed optimal packet size a range of packet size would be more energy efficient.

There is no fixed optimal packet size solution for different scenarios. In the noisy channel, larger packet size have high packet loss rate, as a result smaller packet sizes have higher energy efficiency. However, small optimal packet size have higher overhead. Hence, the use of compression methods is studied in this thesis. Analysis shows that after using compression, efficiency improves 8% and overhead reduces from 24% to 16.5% in the optimal packet size (see figure 5.13). If header and payload are both compressed this can provide a further reduction in overhead 7.7%. Therefore, we suggest to use compression technique to increase energy efficiency (figure 5.15) in ad hoc networks where noise is low.

However, the compression method is no longer an energy efficient technique for optimal packet size in a heavy noisy channel or in a multi-hop QoS constraint application like VoIP. Maintaining the QoS constraints of VoIP, compression remains energy efficient only for up to two hops. In a noisy channel multi-hop network, due to excessive packet loss rate, error correction technique becomes more energy efficient than compression (figure 5.16). Error correction technique does not increase the energy efficiency with low noise. But, it can can be useful to obtain highly reliable channel (see figure 5.5). If FEC is only adopted at the source and destination nodes, this would give higher energy efficiency (see figure 5.8).

However, FEC is completely inefficient for VoIP. After maintaining the VoIP and FEC constraints, it is observed that energy efficiency becomes very low (see figure 7.7 and 7.4). Therefore, it should not be used for VoIP.

Table 8-A demonstrates a summary of various results from this thesis. It shows our experiments and results of optimal packet size in different scenarios. From our analysis it is found that in a good channel condition MTU is the optimal packet size, except for the VoIP. In an average channel condition also MTU is a good option, however not for the time variable channel condition. If FEC is used then energy efficiency becomes even lower. Most variation of optimal packet size occurs in a noisy channel condition. One observation from these results is, if the transmitted packet size is capped between 300

Table 8-A: Optimal Packet Size Estimation Results in Different Scenarios

Scenarios	Channel Condition			Efficincy
	Good	Average	Noisy	Range
In General	MTU	MTU	359B	58%-90%
Time Variable	MTU	540B	380B	60%-81%
Congested	MTU	MTU	300B	30%-45%
FEC	MTU	592B	336B	45%-52%
Compression	MTU	MTU	713B	63%-90%
VoIP	240B	240B	-	50%-73%

bytes to MTU, 30% to 90% energy efficiency can be achieved.

8.2 Summary

This chapter draws the conclusions based on the studies presented in previous chapters, mainly

- FEC is not an energy efficient approach for the wireless VoIP since it adds huge overhead after considering all the constrains of using FEC and VoIP CODEC (see chapter 7).
- However, FEC is an energy efficient method whenever packet loss rate becomes extremely high. For an instance, large multi-hop network communication or to transmit larger packet sizes in the noisy channel (chapter 5). By using FEC large number of hops can be traversed maintaining very low packet error rate without degrading the efficiency.
- To reduce the overhead of the packet size compression method is analysed in chapter 5. In contrast to FEC, the compression is revealed as an efficient approach in terms of energy and QoS.
- In chapter 6 in a congested channel with multiple source nodes, it is found that optimal packet size does not have direct impact on the packet collision.

Table 8-B: Optimal Packet Size Estimation Overview in Different Network Scenarios

Optimal Packet Size (OPS)		Single	Multi-hop	Multi-Source
Energy Efficiency	Only OPS	High	Low	Low
	FEC & OPS	Low	High	Low
	Comp & OPS	Very High	Low	High
PLR (Packet Loss Rate)	Only OPS	Low	High	High
	FEC & OPS	Very Low	Very Low	Low
	Comp & OPS	Low	High	High
Delay	Only OPS	Low	High	High
	FEC & OPS	High	Very High	High
	Comp & OPS	High	Very High	High

In table 8-B optimal packet size estimation overview is shown for different network scenarios. Here, Comp represents the compression method and OPS means optimal packet size. From this table network operator can choose the best method or technique according to the requirements.

8.2.1 Contributions

To estimate energy efficient optimal packet size, two important parameters are overhead and packet loss. To reduce the packet loss rate FEC has been proposed by many researchers. But overhead also imposes extra cost which reduces the optimal packet size and energy efficiency.

- In this research compression method is applied along with optimal packet size to reduce overhead and improve energy efficiency. This is the first attempt to estimate the energy efficient optimal packet size for diverse range of scenarios considering compression. Energy efficient optimal packet size is investigated and compared considering header compression, header and payload compression, then compression and FEC in combine.

- Another contribution is optimal packet size is estimated for the WLAN VoIP. Before this, many research has been done for the VoIP considering QoS but energy efficient optimal packet size has not been analysed before for VoIP.

Moreover, on the way of doing the experiments and estimating the energy efficient optimal packet size some equations are formed and techniques are generated. For an example,

- Energy efficiency estimation metric is derived from the main packet size estimation metric to estimate energy efficiency for the MTU packet size fragmentation. Our analysis investigates the larger optimal packet size more than MTU and its impact on energy efficiency.
- Also, for time varying channel condition Markov error model is taken into consideration to estimate the optimal packet size. One interesting result found that without using dynamic packet size optimization, if packet size is capped between a range, also high energy efficiency can be achieved.
- For congested channel with multiple source node, to estimate an optimal packet size a distinct and completely unique approach PER and packet collision intersection is shown. This technique can be used as an alternative to estimate the optimal packet size for congested channel.
- In our research optimal packet size, FEC and compression these three methods are analysed and compared for different ad hoc networking scenarios. Hence, our analysis can be used to help the network designer or administrator to determine the suitable techniques according to specific network scenarios and requirements.

8.3 Future Work

Based on the work presented in this thesis, the possible future research directions can be summarized in following areas:

- Since optimal packet size becomes different in different network condition and scenarios. Dynamic packet size optimization is the future interest of this research.

Dynamic packet size adaption technique has been proposed before. However, there are still room for improvement considering compression, FEC, various channel condition and application oriented packet size optimization such as considering VoIP.

- In the real scenario the bit error rate can change according to time. In this research the energy efficiency estimated for fixed bit error rate. This research could be illustrated for variable error rates, variable channel condition and dynamic packet size adaptation during runtime.

Appendix A

Simulation Methodology

Simulation is a prototype or model that mimics the operation of proposed systems. Simulations enable us to analyze or visualize the impact of experiments in a controlled and repeatable environment without real-time implementation. Most of the wireless network performance measurements have been done before by using the simulation tools. A few simulators have been broadly used in the wireless network research community, such as: NS-2 [KK07], OPNET [Cha99], GloMoSim [ZBG98], OMNeT++ [Var01] etc. But among all these popular network simulators specifically NS-2 is chosen because of the following reasons:

1. Since this research is about the optimal packet size estimation for the ad hoc networks, it is crucial to change the packet size frequently during the simulation. With the NS-2 simulator, packet size defining policy is straightforward and can be changed during the simulation runtime.

2. The error model of NS-2 can characterize the bursty packet loss behavior of noisy channel. Bit Error Rate (BER) or Packet Error Rate (PER) follows the theoretical equations according to the estimation given in chapter 3. Also it has time variable noisy channel model to set up Gilbert-Elliot channel.

3. It has built in FEC module and this module follows the same formula which is expressed in our chapter 2.3 as reliability.

4. The topology design, connection between the nodes, traffic pattern settings, WLAN set up etc is fairly easy. Moreover, NS-2 (version 2.34) has the built-in WiFi settings (IEEE 802.11 standard MAC and PHY layer) and Zigbee settings module to do the necessary simulation for WLAN.

5. The built in energy model of NS-2 can be used to estimate the energy consumption of particular node as well whole network.

However, it has some drawbacks too, for an example

- Installation process is quite complex and time consuming
- Documentation is often limited and source code customization is difficult
- Lack of simulation output files analyzing tools. To extract the simulation results, the simulation trace file need to be parsed with either other programming languages (awk, perl, excel etc) or manually.

But considering pros and cons, NS-2 is appeared to us the most favorable tool for doing simulation for this research. A brief description of NS-2 focusing our research is given in the following sections.

A.1 Network Simulator-2 (NS-2)

Network Simulator (version 2) widely known as NS-2 [IH08] is one of the most popular open source simulator which is used for performance analysis of various networks (wired, wireless). It's a discrete event simulator originally developed in 1995 at Lawrence Berkeley Laboratory at the University of California, as part of the Virtual Inter Network Testbed (VINT) project. The Monarch project at Carnegie Mellon University has extended the NS-2 in 2004. They have added the WLAN functionality for NS-2 also introduced the nodes' mobility and realistic OSI layer for wireless network. NS-2 simulator is written in C++ which is the core engine and uses OTcl, an object oriented version of Tcl language for command, configuration and simulation scripts. By using OTcl scripts one can easily set up network topologies, nodes' configuration, traffic pattern etc.

The reasons of using two languages are, both have their own advantages. C++ is slow to change but fast to run. OTcl is slow to run but easy to change. To compromise and work efficiently with these two different behaviors NS-2 offers an excellent combination of these two languages.

A.1.1 Simulation Process

The simulation is processed mainly in four steps. At first the simulation and parameter settings are described in a Tcl script. Then simulation is run and outputs are generated in the trace file. After that the trace file is analyzed with other programming languages. A framework of NS-2 simulator is given in figure 1.1. The results of the simulation is written in a trace file for every millisecond containing details information about the nodes' behavior (e.g. node id, position coordinate, packet size, traffic pattern etc). Table 1-A shows the different parameter specifications of a trace file. By processing this trace file (by using awk, perl etc) delay, throughput, packet loss, energy level, routing overhead etc are monitored. There is also a graphical interface called network animator (NAM) to visualize the simulation. NAM shows the packet propagation within the network with nodes movement (if the nodes are mobile), packet loss and display an overall sketch of the simulation.

A.1.2 NS-2 Energy Model

Energy model of NS-2 can keep track of radio energy consumption of the nodes. The energy model considers only four energy cost Transmit, Receive, Idle and Sleep states.

Table 1-A: NS-2 trace format different fields' specifications of the trace file

Event Information	**Specifications**	**Tag**
Event Type	Send, Receive, drop, forward	S, R, D,F
Time	Packet transmission time in milliseconds	-t
Source and destination Id	Source node Id and Destination node Id	-Hs , -Hd
Current Node ID	Intermediate node Id	-Ni
Nodes Position	X, Y, Z coordinate	-Nx, -Ny, -Nz
Nodes Energy	Nodes Energy level	-Ne
Trace Level	Router, Agent or MAC trace	-Nl
Reasons of packet drop	Queue full, TTL, MAC Error etc	-Nw
Packet Information IP Level	Packet type, size, TTL, flow Id etc	-Is,-It, -If, -Iv
Packet info MAC level	Ethernet address, type etc	-Ma, -Mt
Packet Info level	TCP, UDP, DSR, DSDV, AODV etc	-Po,-Ps, -Pp etc

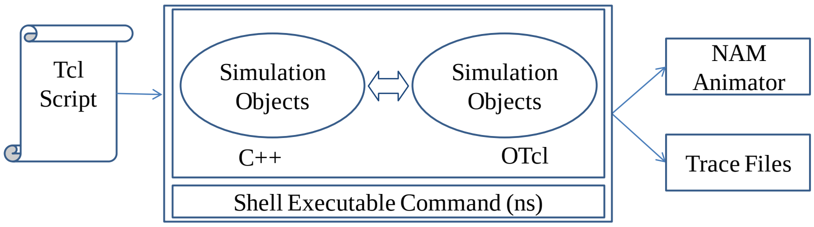

Figure 1.1: Framework of NS-2 simulation

Whenever the radio state changes, it updates the energy model by subtracting the appropriate amount of energy from the previous state. It provides interface to put radio into sleep state, wake it up also considers nodes transition energy (sleep to wake up and vice versa) consumption [KK07]. But NS-2 does not consider any processing energy cost. For an example it does not include decoding energy for error correction.

A.1.3 NS-2 Error Model

NS-2 imposes error on packet transmission by using its' error module. This module simulates packet error after receiving a packet. If the packet is simulated as an error packet either the packet is dropped or marked the error flag of the packet as a corrupted packet and forward it to its destination. Error model can be used for wired and wireless networks both. Errors can be generated randomly in a controlled way by only defining

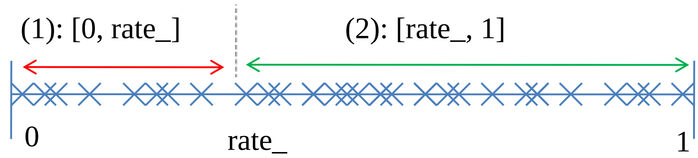

Figure 1.2: Bernoulli distribution for uniform random error of NS-2

the error rate [KK07] and error unit in the simulation (Tcl) script. The user only needs to describe the error procedure in Tcl code in the simulation scripts. An example is given in the appendix. How the simulator code works with the BER and drops packets is explained in the appendix in detail.

The Bit Error Rate (BER) is defined in the Tcl simulation script. The simulator drops packets according to the defined BER. According to the error rate defined in the simulation script, the simulator creates Bernoulli distribution with uniform random variable. The random variable values remain within 0 to 1. If (u) is considered as a random variable between 0 and 1 then packet is marked according to the following statements

1. If (u < PER)
2. return 1; (error).
3. else
4. return 0; (correct).

If every point within the range of 0 and 1 is picked with equal probability. The ratio will be 1:(1+2) which is basically PER:1 (see Figure 1.2). For corrupting the bits within a packet NS-2 uses the following formula:

$$var[m] = \begin{pmatrix} size \\ i \end{pmatrix} b^i (1-b)^{size-i} \tag{A.1}$$

Where $size$ is the total packet size, m is the iteration loop variable and b is the bit error rate. If ($var[0] \geq u$) only that time no bit error will happen because error function will return the value $i = 0$ which means no error. Otherwise for every cases ($var[1]$, $var[2]$ etc) if ($var[1] \geq u$) or ($var[2] \geq u$) packets are marked as corrupted. Only difference is when $i = 1$, only one bit is in error, while $i = 2$ two bits are error within a packet and so on. From the trace file only the number of packet loss and the reason of packet drop can be recorded.

The bit error count is only necessary for the simulator for the error correction process. Because in the error correction process the maximum error correction strength needs to be defined (see chapter 2.3). It determines how many numbers of bits can be corrected within a corrupted packet.

The PER can be estimated from the trace file by calculating the ratio between drop and sent packets. In this thesis the simulated PER is estimated just like the description above for different experiments and after that it is compared with the theoretical PER.

A.2 Simulation Strategy of this Research

The energy efficient optimal packet size estimation simulation starts with simplest ad hoc network scenario. Only two nodes are within their transmission range and communicating with each other. The channel condition is very good; there is not any packet loss. One node is transmitting data on a particular time frame and other node remains silent. The communication continues for certain time period in a vice versa manner. After that the experiments are extended for the noisy channel. Then time variable channel is introduced. Next, the error correction and compression methods are analyzed. Also the impact of multi-hop linear chain topology and multiple source topology networks over the packet size is investigated (see section 1.5 and chapter 6 section 6.5).

Since energy efficiency can not be estimated from NS2 simulator directly. Energy is estimated and compared with the numerical model in chapter 6. At the beginning simulation results are compared with the numerical packet error rate only for fixed channel

Table 1-B: Parameters Settings for the simulation

Simulator	NS-2 Version 2.34	
	Network Size	1000m x 1000m
	Number of Nodes	Variable
	Duration	10,000s
Physical Layer	Signal Propagation model	Two Ray Ground
	Max Transition Range	250m
	Antenna Model	Omni Antenna
Mac Layer	MAC protocol	802.11
	Link Bandwidth (data Rate, RTS/CTS)	1Mb, 1Mb (by default)
Traffic Model	Date Rate	20 Kb
	Traffic Type	Poisson, UDP
	Traffic Interval Rate	Variable
	MTU size	1000 byte (default)
Queue	Type	Drop Tail/PriQueue
	Size	50 (default)

condition with two nodes, one source. Then simulation results are also compared with variable noisy channel. In chapter 4, 5 and 7 simulation results are compared considering packet error rates assuming simple broadcasting communication protocol. Then considering multiple source and multiple hops for particular technology IEEE 802.11 results are plotted in chapter 6.

A.2.1 Parameter Settings

Most of the parameters used in our simulation are default parameter settings of NS-2. Some of the default parameter settings are shown in table 1-B. The packet size, network topology and channel condition are changed to investigate the packet drop of different packet sizes frequently. Except this the other configuration remains the same most of the time for the entire simulation such as: UDP traffic, data rate, bandwidth, source type (Poisson) remains the same.

Detail example of simulation scripts and configuration are shown in the appendix with comments. During the simulation, trace files generate. By using a variety of scripts (e.g. perl, awk, shell etc), the trace files are processed. Further analysis is done with

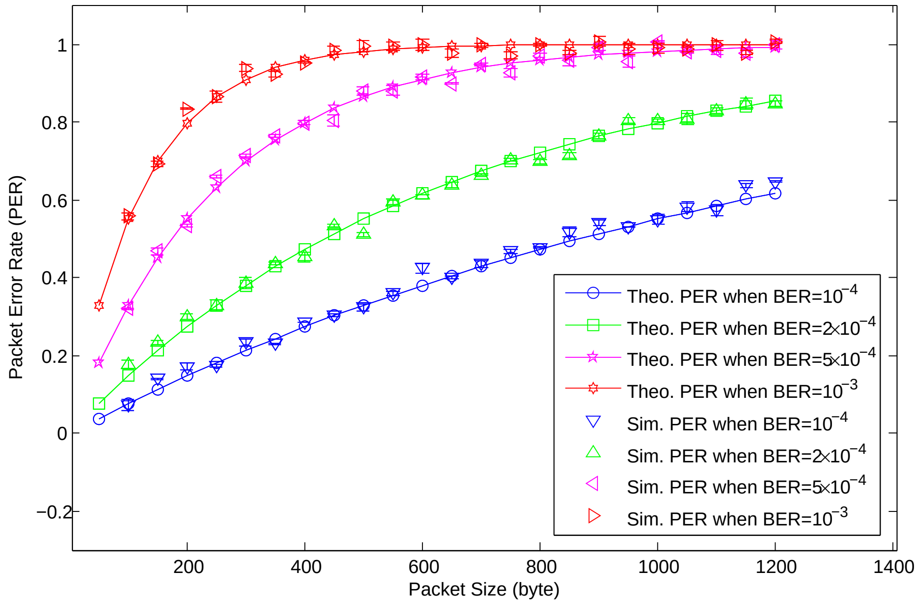

Figure 1.3: For different BER theoretical PER and simulated PER is compared

Matlab or Microsoft Excel to produce the charts shown in this report. In figure 1.3 for different BER the theoretical PER and simulated PER is shown with error bar. This figure shows that with constant data rate, PER does not varies much compared with the simulated results.

After that packet error is estimated for different types of traffic with average data rate 20 Kb/sec. From figure 1.4 it is observed that using different traffic pattern does not vary the average packet error rate theoretically with simulation results. From the experiments it is found that theoretical error rates matches with the simulation results with low error bar. It is also revealed that for the low average data rate theoretical PER and simulated PER remains almost same for various traffic flow.

Simulation is also done for the multi-hop linear chain scenario. Two nodes are at the edge and in the middle of the two nodes there is another node. The edge node are outside the transmission range of each other. Hence, the middle node act as an intermediate data forwarding nodes in between the two nodes (source and destination). It works as a router for packet forwarding. In a noisy wireless channel environment, the PER of a multi-hop linear chain network follows the equation (3.15) of chapter 4.

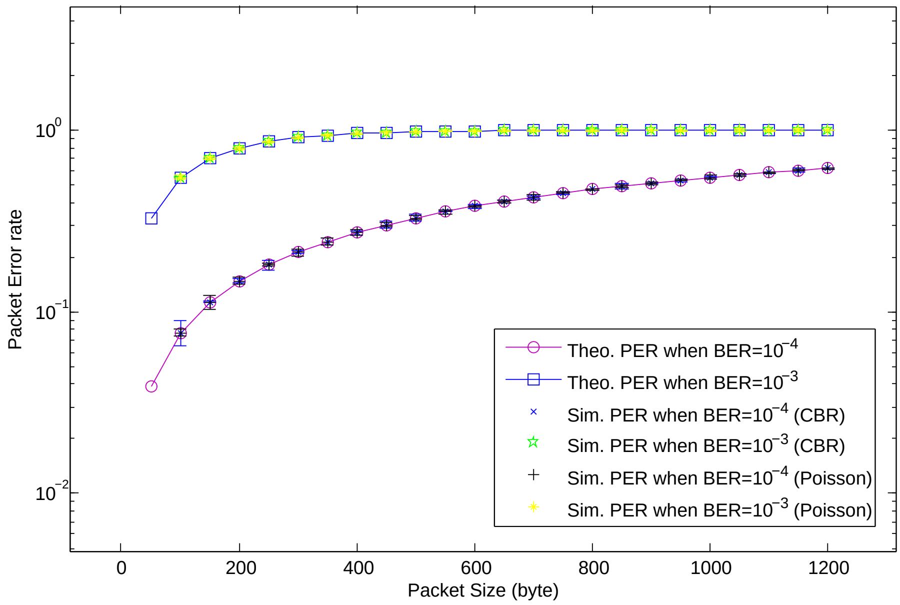

Figure 1.4: Theoretical PER and simulated PER comparison with different traffic flow

The simulation is run for various packet size and took the average values for the sent packets, received packets and packet loss rate. In the simulation packet loss can only happen because of the error packet. From the simulation results it is found that for various packet size the packet error rate follows the multi-hop equation (3.15). For the multi-hop topology the error rate is compared with figure 1.5. It is found that NS-2 results also matches with the theoretical results for the multi-hop linear chain topology network.

A.3 Conclusion

This chapter specifies particularly why the NS-2 simulator is chosen for this research. The advantages and disadvantages of NS-2 are described. The NS-2 architecture, working framework, simulation procedure is illustrated in brief. Also the error module and energy module of NS-2 are extended in two different sections. After that simulation procedure and parameter setting are expressed for this research. Finally, some results from the simulator error module is compared with the theoretical error formula for single hop,

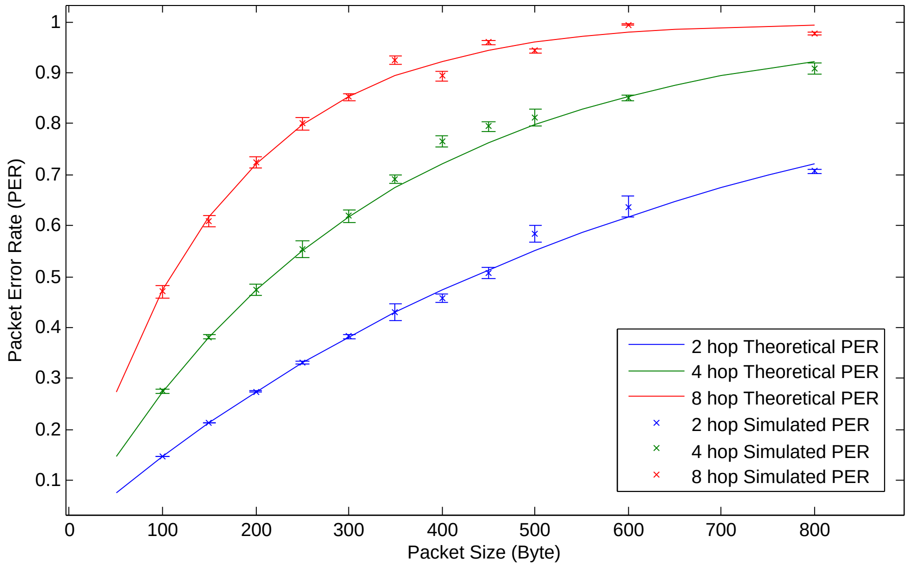

Figure 1.5: Theoretical PER and simulated PER comparison in the multi-hop linear chain topology when BER=10^{-4}

multi-hop and different types of traffic sources. It shows that the theoretical packet error rate matches with the simulation results. The deviations between the results are very low which proves that the simulated results are fairly accurate and comparable.

References

[Cha99] X. Chang. Network simulations with OPNET. *Proceedings of the 31st conference on Winter simulation: Simulation—a bridge to the future*, 1:307–314, 1999.

[IH08] T. Issariyakul and E. Hossain. *Introduction to Network Simulator NS2*. Springer Publishing Company, Incorporated, 1 edition, 2008.

[KK07] F. Kevin and V. Kannan. The Network Simulator NS-2 online documentation. `http://www.isi.edu/nsnam/ns/doc/index.html`, 2007.

[Var01] A. Varga. The OMNeT++ discrete event simulation system. *Proceedings of the European Simulation Multiconference (ESM'2001)*, June 2001.

[ZBG98] X. Zeng, R. Bagrodia, and M. Gerla. GloMoSim: a library for parallel simulation of large-scale wireless networks. In *Parallel and Distributed Simulation, 1998. PADS 98. Proceedings. Twelfth Workshop on*, pages 154 –161, May 1998.