Improved Multi-Point Communication for Data and Voice Over IEEE 802.11b

A Thesis Submitted to the College of

Graduate Studies and Research

in Partial Fulfillment of the Requirements

for the Degree of Master of Science

in the Department of Electrical Engineering

University of Saskatchewan

Saskatoon, Saskatchewan

By

Dongmei Zhao

PERMISSION TO USE

In presenting this thesis in partial fulfillment of the requirements for a degree of Master of Science from the University of Saskatchewan, the author agrees that the libraries of this University may make it freely available for inspection. The author further agrees that permission for copying of this thesis in any manner, in whole or in part, for scholarly purposes may be granted by the professor or professors who supervised this thesis work or, in their absence, by the Head of the Department or the Dean of the College of Graduate Studies and Research. It is understood that any copying or publication or use of this thesis or parts thereof for financial gain shall not be allowed without the author's written permission. It is also understood that recognition shall be given to the author and to the University of Saskatchewan in any scholarly use which may be made of any material in this thesis.

Requests for permission to copy or to make other use of material in this thesis in whole or part should be addressed to:

Head of the Department of Electrical Engineering,
57 Campus Drive,
University of Saskatchewan,
Saskatoon, Saskatchewan, Canada
S7N 5A9

ABSTRACT

There is a growing demand for faster, improved data and voice services in rural areas without modern telecom infrastructure. A wireless network is often the only feasible solution for providing network access in this environment, due to the sparse populations and difficult natural conditions.

A system solution that incorporates the Multipoint Communication System (MCS) algorithm created by TR*Labs* into the available IEEE 802.11b Wireless Local Area Network (WLAN) devices was proposed and studied in this thesis. It combines the advantages of both systems, that is, the MCS' capability of integrating Voice over Internet Protocol (VoIP) and data services and the IEEE 802.11b standard, currently the most widely used in WLAN products.

A system test bed was set up inside Network Simulator-2 (NS-2). The data and VoIP performance was tested. Modifications to the original MCS algorithm to improve system performance were made throughout this thesis.

In a constant rate radio channel, data performance (throughput and transmission efficiency) was measured using the original MCS algorithm, which was comparable to the standard Distribution Coordination Function (DCF) operation of IEEE 802.11b when both were simulated at similar conditions. On an 802.11b platform, the Automatic Rate Fallback (ARF) feature was incorporated into the original MCS algorithm. However, when clients with different data rates were present in the same channel, all the clients involved received unacceptably low and equal data throughput, dragged down by the low rate clients. A modified MCS data polling algorithm was proposed with the capability of repeated polling, which eliminated the negative effect of low rate clients in a multi-rate channel.

In addition, the original MCS algorithm was modified to be more efficient in the voice polling process. The voice performance and data throughput were tested at various conditions. However, the one-by-one polling still resulted in very low voice transmission efficiency. The time wasted became more severe with increasing relay distance and channel rate (only 8.5% in an 11 Mbps channel at 30 km). A new voice handling process similar to Time Division Multiple Access (TDMA) mode was implemented and simulated. Its voice efficiency can be kept at 25% at any setting of relay distance and channel rate. Data transmission in the same channel can also benefit from using the new voice scheme. The normalized saturation throughput could be improved by 13.5% if there were 40 voice clients involved in an 11 Mbps channel at the relay distance of 15 km, compared to the original MCS algorithm. More improvement in voice efficiency, voice capacity, and data throughput can be achieved at longer relay distance, or with more voice calls set up.

ACKNOWLEDGEMENTS

I would like to thank my supervising professor, Dr. Dave Klymyshyn, for providing me the opportunity to study for my Master’s degree in Canada and for his guidance during my graduate program. I benefited much from his knowledge, especially from his attitude toward doing research.

Special thanks go to the Telecommunications Research Laboratories (TR*Labs*) for providing me with funding, as well as a very conductive working environment. Thanks to Jack Hanson and Trevor Hamm, for sharing their knowledge on my research, and other people in TR*Labs*, all of whom made my stay there enjoyable and rewarding.

Finally, I would like to give my deep thanks to my dearest parents, Xu, Huanchun, and Zhao, Fengbao. It is their love, encouragement, and supports that have been accompanying me to get over all the difficulties I have ever had in my life.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

ACK	ACKnowledgment
ADSL	Asymmetrical Digital Subscriber Line
AP	Access Point
ARF	Automatic Rate Fallback
ATM	Asynchronous Transfer Mode
BER	Bit Error Rate
BSS	Basic Service Set
CBR	Constant Bit Rate
CCD	Complementary Cumulative Distribution
CCK	Complementary Code Keying
CDMA	Code Division Multiple Access
CFP	Contention Free Period
CM	Call Manager
codec	COder/DECoder
CP	Contention Period
CRTP	Compressed Real Time Protocol
CSMA	Carrier Sense Multiple Access
CSMA/CA	Carrier Sense Multiple Access with Collision Avoidance
CSMA/CD	Carrier Sense Multiple Access with Collision Detection
CTS	Clear To Send
CW	Contention Window
DBPSK	Differential Binary Phase Shift Keying
DCF	Distributed Coordination Function
DIFFServ	DIFFerentiated Services
DIFS	DCF InterFrame Space
DQPSK	Differential Quadrature Phase Shift Keying
DSSS	Direct Sequence Spread Spectrum
ETSI	European Telecommunications Standards Institute

FCC	Federal Communications Commission
FCS	Frame Check Sequence
FDMA	Frequency Division Multiple Access
FHSS	Frequency Hopping Spread Spectrum
FTP	File Transfer Protocol
FWA	Fixed Wireless Access
HIPERLAN	HIgh-PERformance Local Area Network
HTTP	Hyper Text Transfer Protocol
IBSS	Independent BSS
IC	Industry Canada
IEEE	Institute of Electrical and Electronics Engineers
IFS	InterFrame Space
IP	Internet Protocol
IR	InfraRed
ISM	Industrial, Scientific, and Medical
ITU	International Telecommunication Union
LAN	Local Area Network
LLC	Logic Link Control
MAC	Medium Access Control
MCS	Multipoint Communication System
MPDU	MAC Protocol Data Unit
MSDU	MAC Service Data Unit
NACK	Negative ACKnowledgment
NAV	Network Allocation Vector
NS-2	Network Simulator-2
OFDM	Orthogonal Frequency Division Multiplexing
OSI	Open System Interconnect
OTcl	Object oriented Tool command language
PC	Point Coordinator
PCF	Point Coordination Function
PER	Packet Error Rate
PIFS	PCF InterFrame Space

PLCP	Physical Layer Convergence Procedure
PLR	Packet Loss Rate
PMD	Physical Medium Dependent
PPDU	PLCP Protocol Data Unit
PSDU	PLCP Service Data Unit
PSTN	Public Switched Telephone Network
QoS	Quality of Service
RTP	Real Time Protocol
RTS	Request To Send
SIFS	Short InterFrame Space
SNAP	SubNetwork Access Protocol
TCP	Transmission Control Protocol
TDD	Time Division Duplex
TDMA	Time Division Multiple Access
TIM	Traffic Indication Map
TR*Labs*	Telecommunication Research Laboratories
UDP	User Datagram Protocol
VoIP	Voice over Internet Protocol
VoWLAN	Voice over Wireless LAN
WLAN	Wireless Local Area Network

CHAPTER 1

INTRODUCTION

1.1 Overview of WLANs

The Internet has fundamentally changed the way people live and has become an indispensable tool in modern life. During over three decades of development, the Internet itself has also changed dramatically, especially in the access technologies, also called the “last mile”. The conflict of cost and environmental conditions over the problem of “last mile” has plagued service providers and users for long time. Compared to the speed of backbone networks, access networks are slow and often the bottleneck of the whole network.

The Local Area Network (LAN) is the most significant access technology providing network connection for group users in a single building (or on a corporate or university campus). Ethernet, also known as the IEEE 802.3 LANs, is currently by far the most prevalent of LAN technologies.

It is apparent that the convergence of wireless and Internet usage is already underway. Wireless Local Area Network (WLAN) is a rapidly emerging LAN technology in recent years. It offers services similar to Ethernet for mobile or portable users within a limited geographical area like a building or campus, using the microwave radio spectrum. Accessing the Internet via wireless methods is becoming more and more attractive largely due to its flexibility. According to the prediction of the Gartner Group, a market research firm, the number of WLAN users will rise from 4.2 million in 2003 to 31 million by 2007 in the USA alone [1].

Compared to its wired counterpart, WLAN has the following advantages:

- ❑ Mobility/Greater Flexibility

WLANs provide network connection with mobility and greater flexibility. People can freely reach the Internet anywhere at any time in service areas without being tethered by a wired network.

- ❑ Faster Deployment

  It is easier and faster to set up a new WLAN than a wired network, because WLAN requires no physical wire to be installed.

- ❑ Lower Installation and Maintenance Cost

  The cost of installation and deployment has to be evaluated case-by-case and the actual costs depend on the particular circumstances. Where the installation and maintenance of a wired network is costly or impractical, for example, in the buildings with structures difficult to wire (such as concrete buildings, warehouse, and historical buildings), or in sparsely populated areas, wireless access is a more cost effective option.

Of course wireless LANs also face certain challenges and constraints occurring in a few areas, because of the nature of radio channel.

- ❑ Security

  It is more difficult to secure a wireless network, since the transmission medium is open to anyone within the range of a transmitter. Data privacy is accomplished using encryption, but usually at the expense of increased cost and decreased performance [2].

- ❑ Bandwidth

  In general, the bandwidth of wireless LAN is much less than Ethernet. Typical Ethernet rates are 10 Mbps, 100 Mbps, or even up to 1 Gbps, while most available WLAN products only provide data rate ranging from 1 Mbps to 11 Mbps. Higher data rates up to 54 Mbps are also available now for wireless LANs, however, with much smaller service cells.

- ❑ Interference Sensitivity

  Interference is inevitable in the radio channel. The utilization of radio frequency spectrum is regulated to avoid interferences. For wireless LANs operating in the unlicensed Industrial, Scientific, and Medical (ISM) bands, effective anti-interference transmission techniques must be employed.

Mobility and broadband multimedia applications will continue to be two major themes of the next generation IT network. Although it has been acknowledged that broadband wired networks will form the backbone of future high-rate multimedia applications, the demands for wireless networks will no doubt keep growing. They are never a replacement to wired networks, but a necessary complement or extension, and sometimes the only feasible solution.

Currently two series of WLAN standards are being used extensively: the IEEE 802.11 WLAN and the European Telecommunications Standards Institute (ETSI) HIgh-PERformance Local Area Network (HIPERLAN). The IEEE 802.11 WLAN project was initiated in 1990 and has evolved to the most widely used WLAN standards family involving 802.11 [7], 802.11b [8], 802.11a, etc. Among those, 802.11b WLAN has been well received, and is currently becoming very prevalent in home, corporate, and academic environments.

The emergence of Internet telephony makes it possible to combine separate data and voice networks into a single transport mechanism. However, the fundamentally different service requirements suggest that a network architecture that has been designed primarily for data communication may not be well suited for supporting real-time voice. Traditional services on the Internet are based on best-effort delivery and fair access for every user receives more consideration than does Quality of Service (QoS). This becomes a drawback for the Internet (WLANs included) when it comes to supporting real-time traffic and imposes a big challenge for implementing multimedia applications. With a lack of mechanisms to guarantee required voice delay, the currently available 802.11b products for Internet access primarily support data traffic. The Voice over Wireless LAN (VoWLAN) market is still an extremely small, immature market, albeit one with a significantly growing amount of interest and potential.

1.2 Project Motivation

It has also been realized that the coverage of modern telecommunication services is far from sufficient, especially in some rural areas. There is a growing disparity between services available to rural and urban users.

Rural areas are characterized by extended expanses of land where the population is sparse and scattered. Some of them are geographically distant from major population centers and often separated by difficult terrain. Due to the limitation of local economy or environments, there is little modern telecom infrastructure and the best service available in these areas is often a single phone line running to an entire village. Even in developed countries, some rural residents still face the similar challenges, for example, the Yukon area in Canada [3].

Meanwhile a demand for faster, improved data and voice services is growing in rural communities to drive the local economy forward. Another demand is the exposure of a new generation of children to high-speed network access in schools. However, most rural phone lines are incapable of handling high-speed services. Without enough communication links such as fiber optic or co-axial cable, rural residents are often faced with very few (or no) practical options for receiving high-speed access to the Internet.

The requirements for communication networks in these areas could include:

- Robust network access for high-speed data and real-time voice services
- Low investment cost
- Inexpensive hardware at the user sites
- Rapid deployment, easy operation and maintenance
- Scalability

Obviously network access in such areas faces enormous technical challenges. High bandwidth “wired” data access methods (cable modems, ADSL, etc), and even ordinary modem access through telephone lines are often not cost effective to deploy due to the sparse populations and difficult natural conditions. Wireless access methods (satellite access is included) are more feasible in such environment.

TR*Labs* has created the Multipoint Communication System (MCS) to provide point to multipoint wireless data and Voice over Internet Protocol (VoIP) for rural areas. Since IEEE 802.11b is by far the most widely used WLAN technology (maybe not the best solution technically, but the best one practically up to now) with the biggest market share and more cost effective products, a system scheme that combines MCS and IEEE 802.11b would be advantageous. Thus, configured with ordinary 802.11 PC cards, rural residents can reach the Internet like an IEEE 802.11b WLAN, providing not only the Internet data service, but also with VoIP integrated.

1.3 TR*Labs*' MCS Solution

A Fixed Wireless Access (FWA) network provides point to multi-point transmission of voice and data traffic between stationary devices and a hub station. Belonging to this type of wireless application, TR*Labs*' MCS was designed to be a cost-effective solution to provide two-way wireless Internet access and IP telephone service for rural areas over long distances.

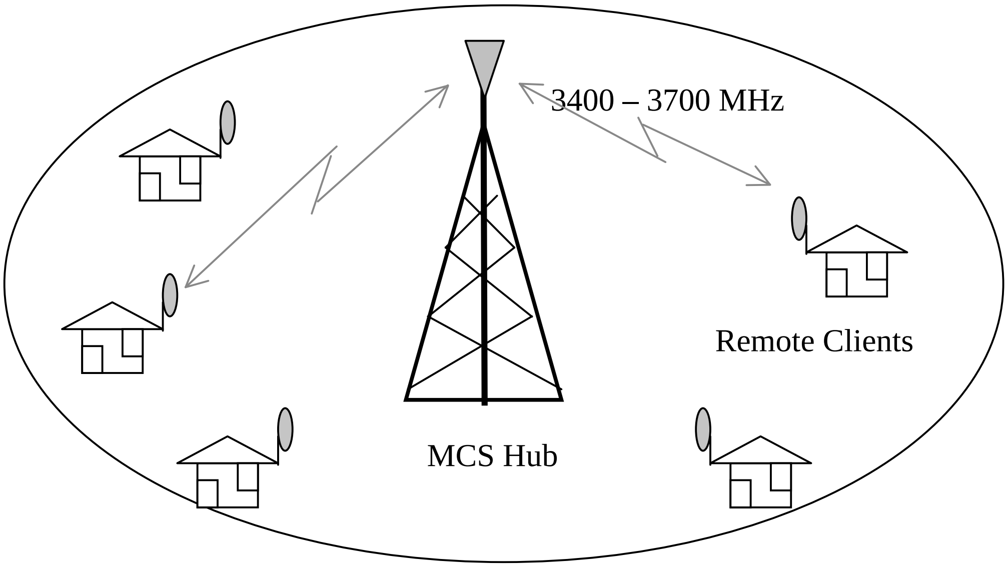


Figure 1.1 TR*Labs*' MCS Time Division Duplex (TDD) communication

Figure 1.1 shows one radio cell of MCS involving an MCS hub station, which can support as many as 100-200 remote client stations on a single radio channel. The hub station can be installed in the nearby city or town with access points to the Internet and the Public Switched Telephone Network (PSTN), while clients are scattered within a large geographical area. The cellular footprints can vary in size from several kilometers to over 30 km. Each client station can support one data device (computer) or one voice device (VoIP phone), or both.

A common frequency channel is used by both the hub and the remote stations. At any time, only the hub or one of the remote stations is allowed to transmit. The hub sequentially polls each client, passing a software token along with a data or voice packet to the client. All remote clients receive it, but only the one that the packet is addressed to can be allowed to reply with its packet, the others ignore and discard the token. The MCS polling is characterized by prioritized token issuing and the highest priority is allocated to voice packets. The hub and remote clients time share the common frequency for the forward (hub to remote) and return (remote to hub) channel, known as Time Division Duplex (TDD), rather than separate frequency channels. This is more bandwidth efficient, as it allows for dynamic bandwidth allocation in both directions, especially suitable for asymmetric Internet data.

The data rate for one MCS channel was initially defined at 2 Mbps, although this is somewhat arbitrary. If one single channel is inadequate to service an entire cell, this system allows full scalability, simply by adding additional common radio channels. TR*Labs*' MCS radio was designed to operate in the FWA 3400-3700 MHz band. Differential Quadrature Phase Shift Keying (DQPSK) is used as the modulation technique to achieve bandwidth efficiency in the order of 1.6 bps/Hz. [3][4]

1.4 Proposed System Solution

TR*Labs*' MCS was designed specifically for rural areas, being able to prioritize delay sensitive voice while still transporting data efficiently. However, it was felt that the "non-standard" nature of the original MCS prototype was detrimental to its acceptance for broad commercial development.

In contrast, as more and more people enjoy the convenience of wireless surfing, IEEE 802.11b products from various vendors are widely used by laptop users, due to their easy availability, rapid deployment, and good prices. Unfortunately, rural residents are not the targeted users of IEEE 802.11b, mainly because of the small coverage on the order of tens or hundreds of meters. Another limitation for the commercially available 802.11b products is their incapability of supporting real-time voice service.

On the recognition of the advantages and drawbacks of both systems, a new system solution is proposed, combining the attractive features of TR*Labs*' MCS (including effective provision of both voice and data) with the increasingly accepted

IEEE 802.11b. It is implemented by incorporating an MCS algorithm into available IEEE 802.11b products, with the capability of supporting both data and voice services for rural residents using the standard IEEE 802.11b hardware. The service coverage provided by the new system can be extended from the limited 802.11b cell into a large geographic area, which is “seamless” to the end users, so that they can access the network in the same way as in a local IEEE 802.11b WLAN.

1.4.1 System Architecture

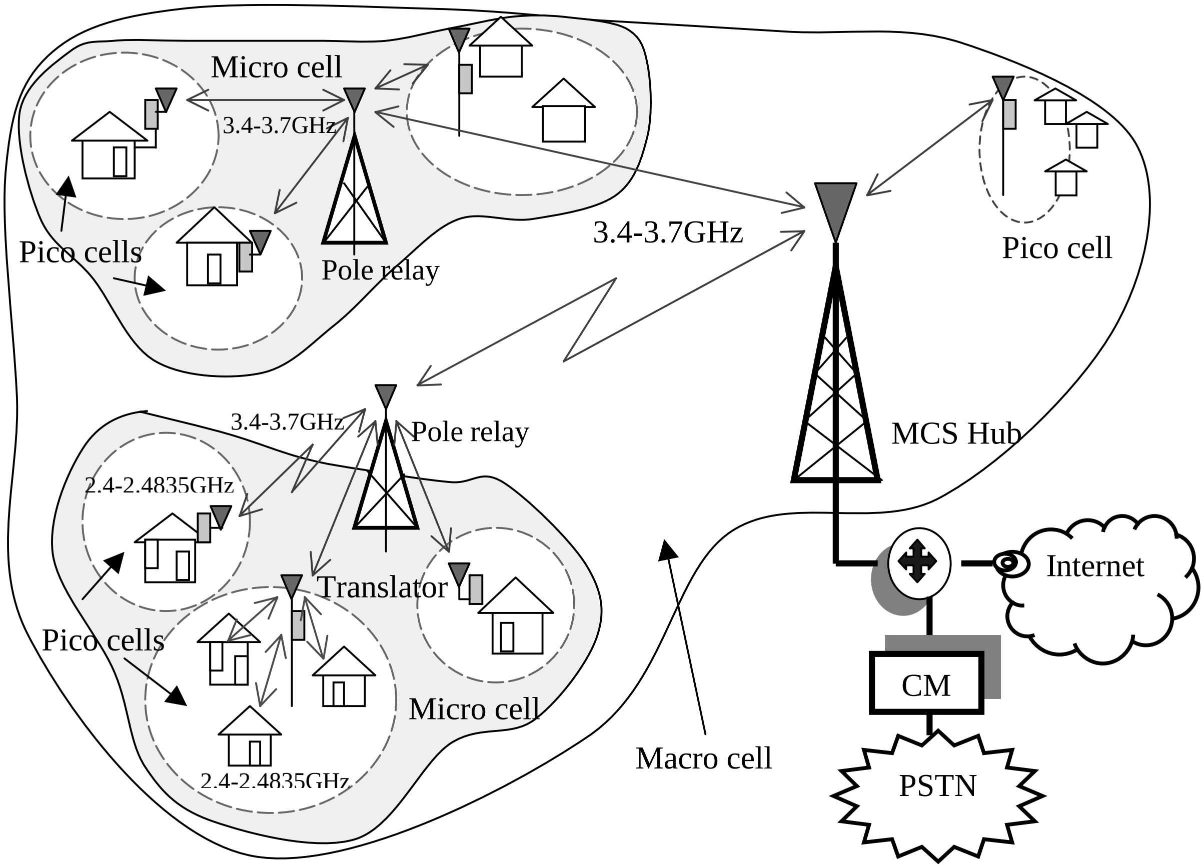


Figure 1.2 Proposed system architecture

One possible scenario of the proposed system solution is depicted in Figure 1.2. Three types of cells are used in the order from the smallest: pico cell, micro cell, and macro cell. A pico cell consists of client stations equipped with PC cards complying with the IEEE 802.11b standard, similar to the Basic Service Set (BSS) of IEEE 802.11b. The change is the local 802.11b Access Point (AP) is moved to the MCS hub. In its usual position is a frequency translator that is installed close to clients in the pico cell.

Clients use one of the standard data rates of IEEE 802.11b: 1, 2, 5.5, 11 Mbps in the ISM frequency band of 2.4 GHz inside pico cells. The frequency translator is used to convert the frequency from the unlicensed 2.4 GHz ISM band in the pico cells to a licensed MCS band (for instance, 3.4-3.7 GHz) for reliable long distance propagation, and vice versa in the opposite direction. The distance from the frequency translator to clients is within the useful range of 802.11b devices.

The MCS hub station usually is installed where there are interfaces to the backbone of PSTN (through the Call Manager (CM)) and the Internet. Some pico cells might be close to the hub station, while some others are located very far from it, for example, 20-30 km. In this case, a relay station is necessary to amplify the signal between clients and the MCS hub station. Several pico cells using one relay station comprise a micro cell. The area controlled by one MCS hub can be regarded as a macro cell. Usually all the clients included in a macro cell share a single channel. New common channels can be added for full scalability.

1.5 Research Objectives

The overall project aims to provide two-way, high-speed, integrated data and VoIP voice service to rural areas, in a robust but cost effective manner. This thesis focuses on the performance improvement and protocol optimization at a system level. The detailed tasks are listed below:

- Propose a feasible system scheme integrating the features of TR*Labs*' MCS in an IEEE 802.11b platform.
- Set up a system test bed inside Network Simulator-2 (NS-2).
- Test the data and VoIP performance using the MCS algorithm in the proposed network topology. The multi-rate feature of IEEE 802.11b will be incorporated and simulated. The analyses and evaluation of system performance will be made.
- Improve system performance, especially the data transmission in multi-rate applications and the inefficiency of MCS' voice transmission.

1.6 Thesis Organization

The thesis organization follows:

Chapter 1 gives a short overview of wireless LANs. The status of telecommunication in rural areas is briefly described, and from that the motivation for this project is introduced. A system solution that combines the features of TR*Labs*' MCS in an IEEE 802.11b platform is proposed. The objectives of this thesis are also summarized.

Chapter 2 explains relevant background theory in networking and the metrics for characterizing VoIP performance. The access methods of the IEEE 802.11 MAC layer and the features of physical layer are briefly introduced. Finally, recent research to support real-time voice services over IEEE 802.11 is reviewed.

Chapter 3 explains the polling algorithm of the TR*Labs*' MCS prototype and the C++ simulator modules implemented previously.

Chapter 4 concentrates on data performance of applying the original MCS algorithm in the proposed network topology. The Automatic Rate Fallback (ARF) of IEEE 802.11b is implemented in the MCS' simulation. Simulations are conducted using the NS-2 simulator under various channel conditions. The feasibility of the original MCS algorithm to support multi-rate data transmission is explored through simulations. A modified MCS data polling method with the capability of repeated polling is proposed and simulated in multi-rate environment.

Chapter 5 first makes a modification in the voice polling of the original MCS algorithm. Using the improved MCS polling algorithm, the voice end-to-end delay, voice capacity and transmission efficiency are evaluated from simulation results.

In Chapter 6, a new TDMA-like voice transmission scheme is proposed to improve the voice transmission efficiency of the MCS algorithm. A new voice processing module is implemented in C++ code and incorporated with the other MCS modules. Simulation results are given to show the improvement achievable under various channel conditions.

Chapter 7 presents a summary of the results and conclusions. Also, future work is outlined.

CHAPTER 2

BACKGROUND

Relevant background theories in networking are explained in this chapter, including the layered Internet model, commonly used protocols, and the prevalent multiple access methods. QoS Metrics of Internet telephony are also explained. Following that is the brief introduction to the IEEE 802.11 standard including network architectures, the medium access operations and the features of the 802.11b physical layer. Finally, recent research on the capability of IEEE 802.11 to support real-time service is reviewed.

2.1 OSI and Internet Models

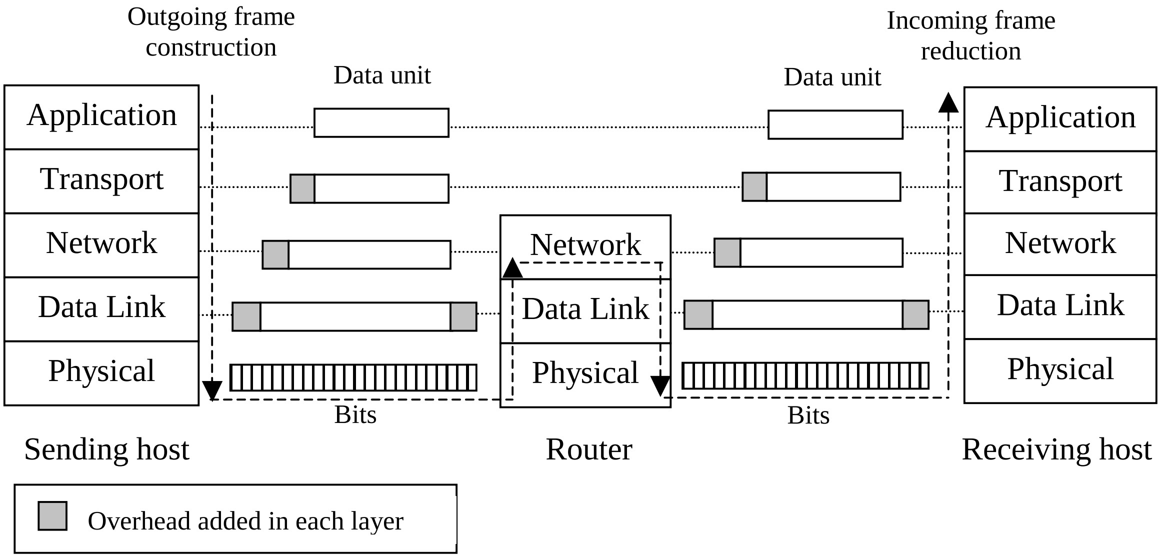


Figure 2.1 Data encapsulation and decapsulation in the Internet protocol stack

The Internet protocol stack is the simplified version of the Open System Interconnect (OSI) model. The flowing of data within the network can be seen as the process of being passed up and down from layer to layer, as shown in Figure 2.1, illustrating how data units of different layers are encapsulated in turn at the end of sender and the inverse process, decapsulation, occurring at the end of receiver, in the layered Internet model. The process of encapsulation increases the overhead involved in transmitting data and results in transmission inefficiency. It is necessary to achieve a balance between information and overhead when designing a protocol.

A data unit is first encapsulated at the transport layer. Transmission Control Protocol (TCP) and User Datagram Protocol (UDP) are two distinct transport-layer protocols. A 20 byte header is added by TCP, which is used to provide a connection-oriented service along with the TCP algorithm, including guaranteed delivery of application-layer messages and mechanism for flow control. UDP, on the other hand, provides an unreliable, connectionless service and attaches only an 8 byte UDP header.

The Internet Protocol (IP) is the most common protocol used at the network layer, and is responsible for routing data packets from one node (host or router) to another. An additional overhead (for example, 20 bytes in the IP version 4 (Ipv4)) is added.

The data link layer has the job of moving a network-layer data unit, similar to the transport layer, but over a single link from node to node. The services provided at this layer depend on the specific link-layer protocol. The physical layer concentrates on how a single bit is moved from one node to the next. Its protocols strongly depend on the actual transmission medium of the link.

2.2 Multiple Access Methods

Broadcast channels are often used in LANs (including WLANs) where multiple sending and receiving stations are all connected to the same, single, shared channel. Traditional multiple access protocols such as Frequency Division Multiple Access (FDMA), Time Division Multiple Access (TDMA), and Code Division Multiple Access (CDMA) are suited for real-time traffic because of the capability of eliminating collisions and providing fair, dedicated access for every user. However, for bursty data traffic on the Internet like file transfer and web browsing, it is impossible to predict the traffic load and the number of users, since users' requests are randomly generated. Moreover, the bandwidth of the Internet is a critical resource. More flexible protocols are required for such applications. ***Random access protocols*** and ***taking-turns protocols*** are two categories of multiple access protocols to deal with channel sharing in LANs.

Carrier Sense Multiple Access (CSMA) protocols, belonging to the random access methods, are being widely used in the wired Ethernet and the IEEE 802.11 WLANs. Users are always allowed to transmit at the full rate of the channel. When there is a collision, each user involved in the collision repeatedly retransmits its packet until the packet gets through. The delay for a user to wait until retransmission is randomly selected. To efficiently decrease the possible collisions, the “carrier sensing” is performed. That is, a user is required to detect the channel before transmitting. If the channel is sensed to be idle, it then begins transmitting. Otherwise, it waits a random amount of time and then senses the channel again.

CSMA with Collision Detection (CSMA/CD) is the dominant access mode of Ethernet, using collision detection to further minimize the collisions. It requires the transmitting user to listen to the channel while it is transmitting. Once detecting another transmission, it stops right away and attempts to transmit after an exponential backoff selected at random. However, collision still might occur during transmission due to the end-to-end channel propagation delay. The longer this propagation delay, the larger the chance that a user is unable to sense a transmission that has already been initiated by another user in the network.

CSMA with Collision Avoidance (CSMA/CA) was developed from CSMA/CD to adapt to wireless transmission media and was specified as the fundamental access method of the IEEE 802.11 WLANs, which will be introduced in the next section.

Polling protocol is an example of the ***taking-turns protocols***. Different from random access protocols, it allows each client to access the channel by “taking turns” in a cyclic manner, and not by contention. One of the stations is designated as a master, which is in charge of polling every other station. Only after receiving the polling from the master station can one station be allowed to transmit, otherwise it has to wait until being polled. Polling protocol eliminates the collision in the random access protocols. It is generally more robust and a better option for voice over IP service, but introduces a polling delay [5] and also reduces data throughput due to overhead.

2.3 IP Metrics for Voice Over IP

IP telephony provides interactive voice communication over the Internet that can be used as an alternative to the traditional telephone system, but normally at much lower cost. The worldwide telephone network uses 64 kbps pulse code modulation to transmit human voice (mostly falls in the frequency band from 300 Hz to 3400 Hz) with high quality. However, the bandwidth of the Internet is shared by numerous users and supports versatile multimedia services, so modulation techniques with high compression ratio and acceptable voice quality are desirable for IP telephones. For example, G.729 (8 kbps), and G.723.3 (both 6.4 kbps and 5.3 kbps) are popular compression techniques for speech [5]. Among those, G.729 was designed to support a toll-quality 8 kbps speech for wireless applications [13].

Quality of Service (QoS) refers to the capability of a network to provide priority to selected network traffic over various technologies, including Frame Relay, Asynchronous Transfer Mode (ATM), Ethernet, the IEEE 802.11 networks, and IP-routed networks, including dedicated bandwidth, controlled jitter and latency, and improved loss characteristics.

Voice quality is directly affected by three major factors: ***delay***, ***jitter***, and ***loss***. Data service is very sensitive to the packet loss, but somewhat insensitive to delay. In contrast, IP telephony is highly sensitive to the end-to-end delay and delay variation, but can tolerate occasional loss of packets.

End-to-end delay, a widely used latency measure in networking. If the total path from source to destination consists of a number of physical links and network nodes, the end-to-end delay is the accumulation of delay introduced in each link and node, including transmission delay, propagation delay and some other delays such as nodal processing delay, queuing delay, etc. If the encoding delay introduced by the codec is taken into account, the total end-to-end delay can be expressed as

$$\delta_{end\_to\_end} = Q(\delta_{trans} + \delta_{prop} + \delta_{proc} + \delta_{que}) + \delta_{enc} \tag{2-1}$$

Where Q( ) expresses the accumulation of the delay introduced in all the links the packet crosses during the end-to-end transmission [5].

Transmission delay (δ_{trans}), also called the store-and-forward delay, is the amount of time required to transmit all of the packet’s bits into a link. When passing through a store-and-forward network device such as a router or switch, the entire packet must be received before the first bit is transmitted onto the outbound link. Transmission delay is a constant function of link speed and packet size, expressed in *L/R* (in seconds), where *L* (in bits) is the length of the packet, and *R* (in bits/s) is the transmission rate of the outbound link [5].

Propagation delay (δ_{prop}) is the time required for signal to travel from the sending point to the receiving point of a physical link once the bits are pushed onto the link. It is the link distance (*D*) divided by the propagation speed (*S*) of the radio wave in the physical media, denoted by *D/S* (in seconds) [5]. The propagation speed in the wireless channel is very close to the light speed in vacuum (3×10^8 m/s). Actually, propagation delay is comparatively small and can be usually ignored in the IEEE 802.11 WLANs, but must be considered in wide-area wireless networks.

Processing delay (δ_{proc}) and queuing delay (δ_{que}) are variable delays inside network nodes, and are difficult to be described in formulae. Processing delay is the time required to examine the packet's header and determine where to direct the packet. In high-speed routers, it is typically on the order of microseconds or less. Queuing delay is the time a packet at the queue waits to be transmitted onto the link and can vary significantly from packet to packet. It could be zero if the queue is empty and the packet can be transmitted onto the link without delay. It could be very long if the traffic is heavy [5].

The G.729 codec compresses voice to 8 kbps but at the expense of encoding delay (δ_{enc}) (10 ms per frame and 5 ms look-ahead) [24] in Equation (2-1).

Network congestion can be encountered at any time within a network and the flow of packets may take different routes through the Internet, so the end-to-end delay varies from packet to packet. Packet jitter is the variability of packet delays within the same packet stream. Jitter buffers are used to remove jitter by adding extra delay to the quickest packets, so that the delay variation is turned into a constant value and voice can be played out smoothly.

The loss of voice packets can occur in three cases. The first one is network congestion in heavy traffic. When a queue is full, new incoming packets are discarded. Secondly, those packets with the end-to-end delay greater than the system requirement are also dropped. The third possible cause for voice packet loss is the bit errors introduced by any non-ideal channels. Usually the time critical traffics such as Internet phone and real-time interactive video are sent over UDP instead of TCP. Since UDP has no support for error correction and acknowledgement, the corrupted voice packets are dropped right away after being detected, without performing re-transmission. The concept of Packet Loss Rate (PLR) is defined as the ratio of dropped packets to all the transmitted packets.

Users' tolerance for delay varies. The International Telecommunication Union (ITU) standard G.114 states, that the maximum one way delay, 150 ms is acceptable for most user application, and the delay 150-400 ms is tolerable provided that administration are aware of the transmission time impact on the transmission quality of user applications. It has been further suggested by Kostas [14] that for toll quality voice, delay should be kept less than 100 ms with a PLR less than 5%. A more stringent requirement of 1% for PLR should be met [13], if no additional "concealment" technique is implemented to compensate for missing packets.

2.4 IEEE 802.11b Standard

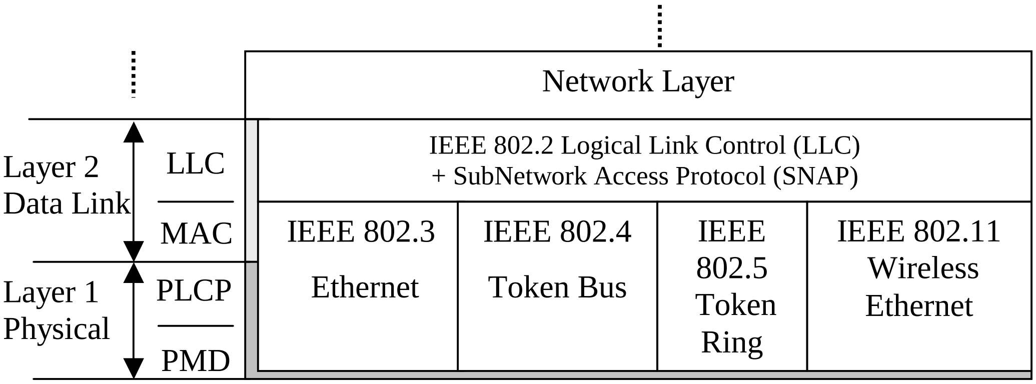
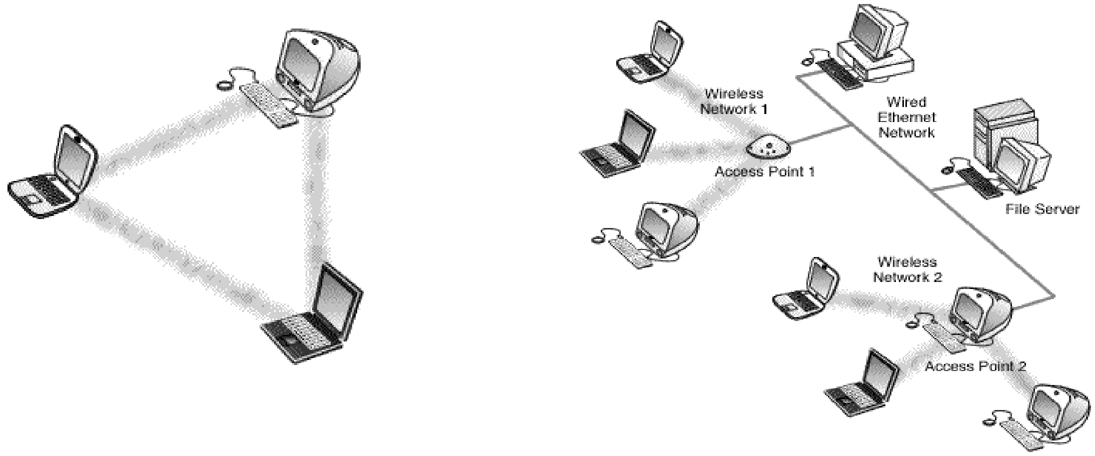


Figure 2.2 IEEE 802 standards in the Internet protocol stack

The data link layer in the Internet protocol stack can be split into two sublayers, Logic Link Control (LLC) layer and Medium Access Control (MAC) layer, as can the physical layer, Physical Layer Convergence Procedure (PLCP) and Physical Medium Dependent (PMD) (Figure 2.2). IEEE 802.11 is located at the parallel position with other wired IEEE 802 standards, specifying the functions of MAC layer and physical layer. The 802.11 MAC layer provides functionality to allow reliable data delivery for the upper layers over the wireless physical medium. For compatibility purposes, it must appear to the upper layers of the network as a "standard" 802 LAN. How the MAC layer works is crucial to the operations of the 802.11 WLANs and also is the most important concern for the MCS.

2.4.1 Architectures



(a) Ad-hoc (b) Infrastructure

Figure 2.3 Basic architectures in IEEE 802.11 [6]

Basic Service Set (BSS), the fundamental building block in IEEE 802.11, is a group of stations controlled by a single coordination function of the MAC layer. There are two basic architectures defined in IEEE 802.11, shown in Figure 2.3: the independent BSS (IBSS, also called ad-hoc network) and the infrastructure network. In an ad-hoc network, any station can establish a direct communications session with any other station in BSS, but there is no connection point to the “outside world”. On the other hand, the infrastructure network is the interconnection of several BSSs, and the connection of BSSs and wired networks (non-IEEE 802.11 LAN) through a centralized Access Point (AP), where the wireless users can reach the Internet via AP.

2.4.2 DCF and PCF

Two different functions are supported by the IEEE 802.11 MAC layer: Distributed Coordination Function (DCF) and Point Coordination Function (PCF). DCF, the fundamental access method of IEEE 802.11 MAC, is the implementation of CSMA/CA. Different from the CSMA/CD of Ethernet, it does not perform collision detection because of two facts. First, it is impossible for a wireless station to listen to the channel for collisions while transmitting without full duplex radio (double bandwidth, full duplex devices, etc.) supported, which can be extremely expensive.

The second reason is the "hidden terminal problem". Even if a wireless station has collision detection and senses no collision when sending, a collision could still occur at the receiver. For example, due to physical obstructions in the environment (such as a mountain), station A and C in Figure 2.4 cannot hear each other's transmission, but they can communicate with the third station B. In this situation, even if station A and C have collision detection, they could not sense the transmission from each other and a collision could occur at the receiver [5].

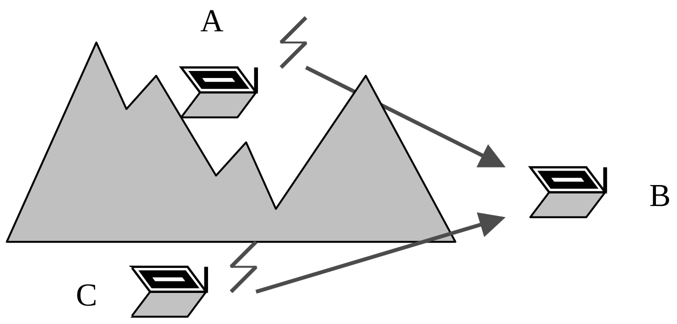


Figure 2.4 Hidden terminal problem in WLANs

Collisions avoidance was developed in the DCF instead, including the following measures:

- The indication of duration time in each frame
- The requirement of an explicit acknowledgment (ACK) returned from the receiver to the transmitting station on successful reception of a packet.
- The optional use of Request To Send (RTS) and Clear To Send (CTS).

Before transmission, a station first senses the medium to determine if another station is transmitting. If the medium is detected as idle, the transmission may proceed. If the medium is determined to be busy, the station will defer until the end of the current transmission, then enter a Contention Window (CW) with its Backoff Timer set to a random backoff time. The Backoff Timers of stations decrement during each CW. Transmission will commence whenever one Backoff Timer reaches zero. In case collisions occur, the stations involved in the collisions re-set their Backoff Timers. The backoff procedure is suspended if the channel is detected to be busy at any time.

The duration field contained in the IEEE 802.11 frame explicitly indicates the length of time that the frame will be transmitting on the channel (or how long the channel will be busy). Other stations receiving the valid frame update their Network Allocation Vector (NAV) based on it. How long one station should defer its access is indicated by its NAV. The NAV timers elapse and the contention for all the stations to access the channel starts again until the NAV value is decreased to zero.

The MAC Protocol Data Unit (MPDU) is encapsulated by adding a 30 byte MAC header and a 4 byte Frame Check Sequence (FCS) field to the Frame Body, as shown in Figure 2.5. The Frame Body is a variable length field containing a MAC Service Data Unit (MSDU) passed down from the LLC, or one of its fragments. For large MSDUs exceeding a Fragmentation Threshold, one MSDU is broken into multiple fragments. The FCS is used to detect transmission errors that might occur over the fields of the MAC header and the Frame Body [7].

Bytes:	2	2	6	6	6	2	6	0 – 2312	4
	Frame Control	Duration /ID	Address 1	Address 2	Address 3	Sequence Control	Address 4	Frame Body	FCS

MAC Header

Figure 2.5 MPDU Frame Format [7]

Two operation modes are supported in DCF according to the utilization of optional RTS/CTS: two-way hand shaking policy of Data/ACK and four-way hand shaking policy of RTS/CTS plus Data/ACK. Stations can choose to never use RTS/CTS, use RTS/CTS whenever the MSDU exceeds the value of RTS_Threshold, or always use RTS/CTS.

Priority access to the wireless medium is controlled through the use of InterFrame Space (IFS) time intervals between the transmissions of frames. Three types of IFS are defined in the IEEE 802.11 MAC layer: Short IFS (SIFS), PCF-IFS (PIFS), and DCF-IFS (DIFS), in the order from the shortest to the longest, which grant the access priority from the highest to the lowest to stations.

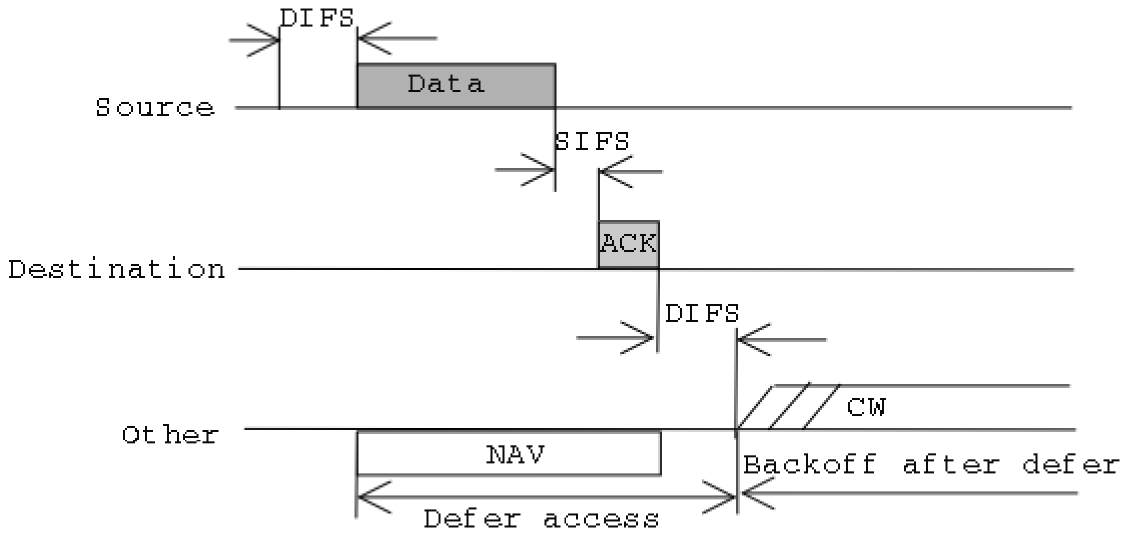


Figure 2.6 Transmission of an MPDU without RTS/CTS in DCF

Figure 2.6 illustrates the successful transmission of a data frame without utilizing RTS/CTS frames. Before transmission, a station first determines that the medium is idle for a DIFS period and then transmit an MPDU. After the destination receives the MPDU correctly, it only waits a SIFS interval to reply with an ACK frame. All the other stations hearing the data frame adjust their NAV values based on the duration field value, which includes the SIFS interval and the ACK following the data frame. Once the current transmission session is complete, all the stations enter the CW after a DIFS period. Whenever the Backoff Timer of a station elapses to zero, it is allowed to transmit and a new transmission session begins.

Figure 2.7 illustrates the transmission of an MPDU using the RTS/CTS mechanism to avoid the “hidden terminal problems”. At the beginning of a DCF transmission, a sender first sends a short RTS control frame to the receiver, indicating the duration of the data packet and the ACK packet. The receiver then responds with a short CTS control frame, giving the sender explicit permission to send. All the other station hearing the RTS or CTS will adjust their NAV values. For example, station A in Figure 2.4 can not hear the RTS from C, but the CTS from B can be received by both A and C. Hence, although station A is out of the range of C, it still can be informed that the channel is going to be occupied for some time and defer its own transmission. Even though there is a collision between RTS or CTS frames, the re-transmission time is much shorter than lengthy data packets because the RTS and CTS are small frames. However, RTS/CTS frames generate additional overhead, so they are only used with the transmission of large data packets.

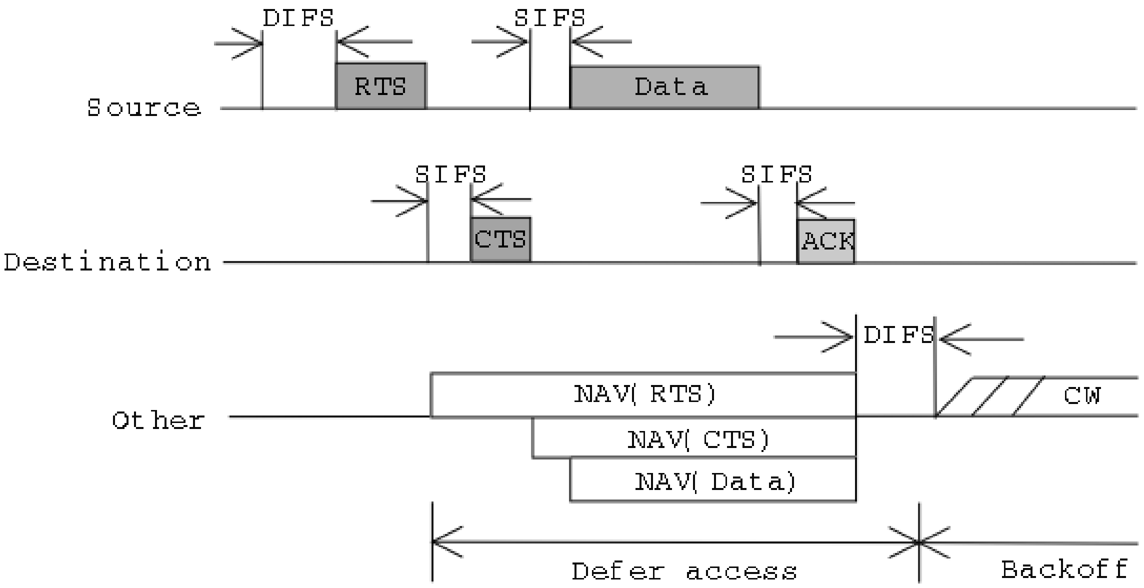


Figure 2.7 Transmission of an MPDU using RTS/CTS in DCF

The DCF can operate solely in an ad-hoc network, and either operate solely or coexist with PCF in an infrastructure network. PCF is an optional access method specified by the IEEE 802.11 MAC protocol to provide contention-free frame transfer for stations involved. The AP is required to perform polling as the Point Coordinator (PC) within each BSS. The polled station is enabled to transmit without contention. The Contention Free Period (CFP) controlled by PCF and Contention Period (CP) controlled by DCF often coexist in the same channel and comprise a superframe. PCF has higher priority over DCF since PIFS is shorter than DIFS.

Figure 2.8 illustrates the constitution of a superframe and how the PC communicates with clients in a polling list during the CFP. At the nominal start of the CFP, if the medium remains idle for a PIFS interval, the PC transmits a beacon frame (B) to initiate the CFP. After one SIFS the PC starts CF transmission by issuing CF-polling to clients in the sequence of the polling list. The PC first sends a CF-Poll + Data packet to Client 1. After receiving this packet correctly, the client sends an ACK back. Then the PC sends a CF-Poll + Data packet to next client. On successful reception Client 2 replies with an ACK together with its packet (U2) for the PC. Then a Data + CF-ACK + CF-Poll packet is sent from the PC. The CF-ACK portion acknowledges the receipt of the previous data packet from Client 2, and the Data and CF-Poll portion starts next polling to Client 3 at the same time. The ability of "piggybacking" an ACK to a CF-Poll + Data packet for different client can improve the transmission efficiency of PCF. If the PC fails to receive the ACK from Client 3, it continues polling the next client after a PIFS interval. Client 4 sends back a Data + CF-ACK packet to the PC. If Client 4 is the last client in the polling list, the PC sends a CF-end packet including an ACK to Client 4, indicating the end of a CFP [7][9].

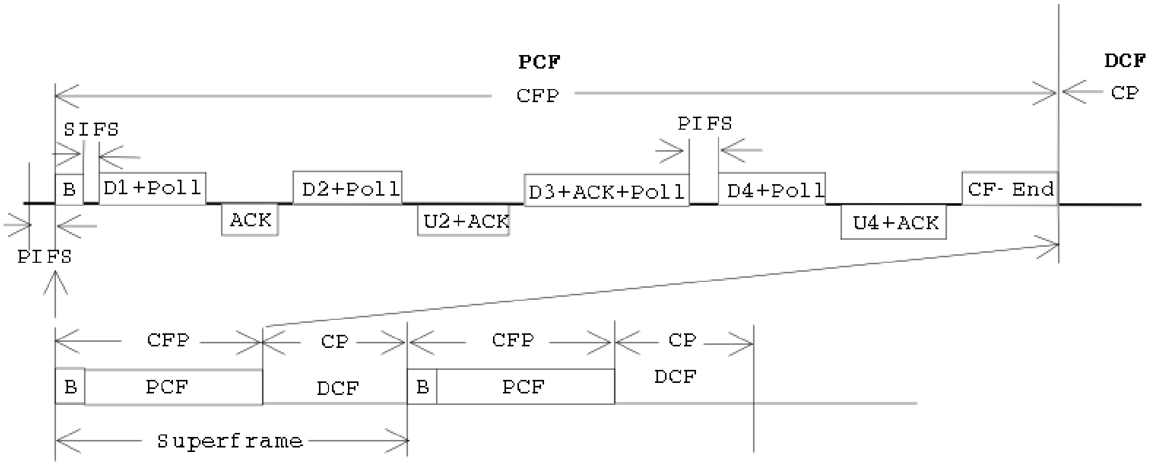


Figure 2.8 Coexistence of the PCF and DCF

2.4.3 Physical Layer

IEEE 802.11 WLAN operates in the unlicensed ISM frequency bands. The technology of Frequency Hopping Spread Spectrum (FHSS) or Direct Sequence Spread Spectrum (DSSS) is utilized to minimize the mutual interference with other transmission sources.

- 802.11 -- provides 1 or 2 Mbps transmission using FHSS, DSSS in the 2.4 GHz ISM band (2.4 – 2.4835 GHz in US, Canada, and Europe) or InfraRed (IR).
- 802.11b (also referred to as 802.11 High Rate or Wi-Fi) -- an extension to 802.11 that extends the data rates supported to 5.5 and 11 Mbps in the 2.4 GHz band and provides backward compatibility, using DSSS as the spread spectrum technology.

- 802.11a -- an extension to 802.11 that provides up to 54 Mbps in the 5 GHz band (5.15-5.35 GHz and 5.725-5.825 GHz). An Orthogonal Frequency Division Multiplexing (OFDM) encoding scheme is used rather than FHSS or DSSS.

IEEE 802.11b can provide Ethernet levels of performance, throughput, and availability for mobile users. Table 2.1 describes the data rates and modulations of 802.11b [8][10].

Table 2.1 IEEE 802.11b data rates and modulations

Data Rate	Spreading Code Length	Modulation
1 Mbps	11 chip Barker code	Differential Binary Phase Shift Keying (DBPSK)
2 Mbps	11 chip Barker code	DQPSK
5.5 Mbps	8 chip Complementary Code	Complementary Code Keying (CCK)
11 Mbps	8 chip Complementary Code	CCK

The PLCP Protocol Data Unit (PPDU) is generated at the PLCP sublayer, then transmitted to the PMD sublayer, which provides a means and method of transmitting and receiving data through wireless medium. The PPDU format shown in Figure 2.9 comprises three portions: a MPDU coming down from the MAC layer (also called PLCP Service Data Unit (PSDU)), preamble and PLCP header.

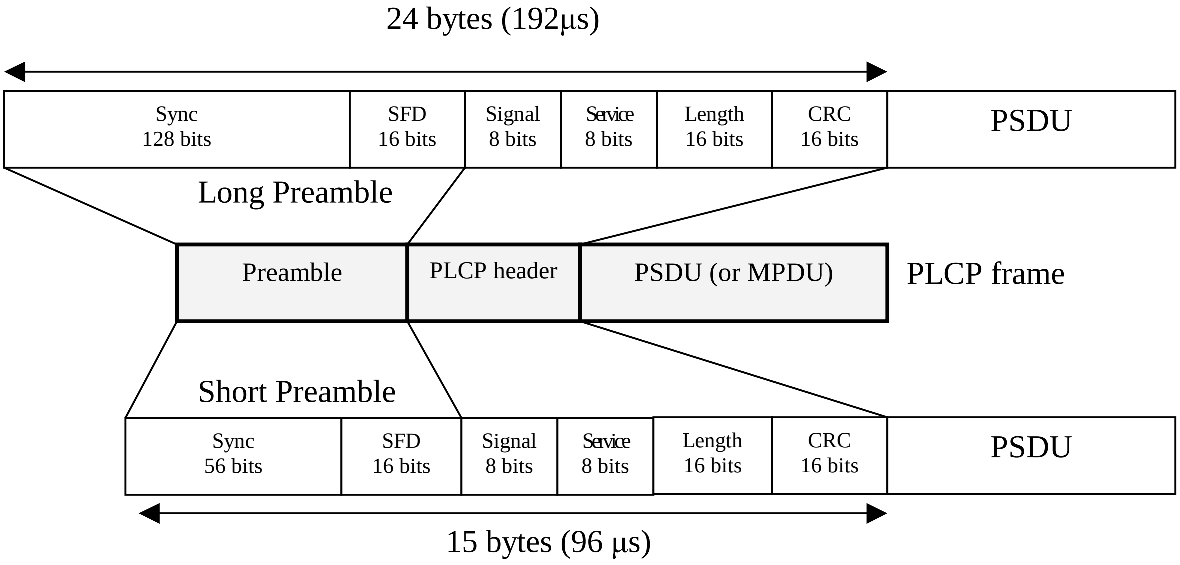


Figure 2.9 Short and long 802.11 preambles

There are two PLCP frame formats in IEEE 802.11b. The mandatory format uses the long preamble with a 128 bit sync field. The long preamble and PLCP header are always transmitted in 1 Mbps DBPSK and the duration is fixed to be 192 μs. An option in IEEE 802.11b is the short preamble with only a 56 bit sync field. While the preamble is still transmitted in 1 Mbps DBPSK, the PLCP header is transmitted in 2 Mbps DQPSK, which reduces the overhead duration to 96 μs, half of the long preamble PLCP. The short preamble format is intended to improve the efficiency of the wireless network for more "real-time" applications such as streaming video and VoIP telephony applications due to the small payload employed.

Another significant feature of the IEEE802.11b WLANs is the ability of dynamic rate shifting with the variation of the distance from device to the access point and/or in a noisy fading environment. The data rate can be automatically reduced to a lower one or upgraded to a higher one based on the channel conditions, in order to improve system performance. Stations with four data rates may coexist in one BSS and share one channel. Data rate shifting is purely a physical layer mechanism transparent to the higher layers.

Following the regulations of Federal Communications Commission (FCC) and Industry Canada (IC), the operating frequency range of 2.4 GHz to 2.4835 GHz in the IEEE 802.11b standard is partitioned into 11 channels. The bandwidth of each channel is about 20 MHz and the center frequencies of two adjacent channels are 5 MHz apart. In a multiple cell network topology, overlapping and/or adjacent cells use different channels with center frequencies at least 25 MHz apart to operate simultaneously [7][8].

2.4.4 Capability of the IEEE 802.11 to Support Voice Over IP

Given the growing popularity of real-time services and multimedia-based applications, it is critical that the IEEE 802.11 MAC protocols be tailored to meet their requirements. DCF is used to support asynchronous data transfer on a best effort basis, but is not suitable for delay sensitive services such as IP telephone. PCF, on the other hand, provides priority access for users with time sensitive traffic by keeping them in the polling list and issuing polling token every CFP, thus eliminating the contentions with others.

However, it is hard for polling to achieve high efficiency. The time delay of voice packets transmitted by the PCF is caused by the repetition time of the CFP interval, as shown in Figure 2.8, and the voice payload length is a trade-off between large overheads and long packetization delays [11]. Furthermore, several researchers have pointed out the IEEE 802.11 PCF poorly supports packet voice traffic [16][17]. Moreover, PCF does not describe a method for creating and maintaining the polling list. “APs may also implement additional polling list maintenance techniques that are outside the scope of this standard.”[7] PCF can be modified to obtain improved performance, for example, by dropping voice stations from the CFP if they are idle for a specific period of time, and setting appropriate voice payload length, etc. [9]. However, no venders currently support the PCF method for MAC.

At the same time quite a few people are concentrating on the DCF-based schemes to support voice traffic over 802.11 WLAN, either in the infrastructure WLANs, or the ad-hoc networks. To guarantee the required end-to-end delay of VoIP, voice users must be granted higher priority than asynchronous data users. Several methods were proposed, for instance, Liu and Wu [11] designed a modified DCF scheme that adapted the power-saved mode of the IEEE 802.11 specifications in such a way that it approached the TDMA access mode carrying voice traffic. Each active voice user was assigned scheduled transmission time in every periodical beacon frame and then transmitted at its allocated time slot. In another method with multi-priority proposed by Deng and Chang [15], the IEEE 802.11 DCF access method was modified to carry the prioritized traffic with four classes of priority available by giving shorter IFS and random backoff time to stations with higher priority.

New members of the IEEE 802.11 standards family are also under development. For example, 802.11e specializes in enhancing the current 802.11 MAC to expand support for wireless applications with QoS requirement, and in the capabilities and efficiency of the protocol, but isn’t specifically being extended to the networks with large coverage. It is being finalized and the first generation product is expected to be available in 2004. The new 802.11g WLAN standard has been approved in June 2003, which is the high rate extension of 802.11b to 54 Mbps and is backward compatible with 802.11b. With available products in the WLAN market, 802.11g will soon enable speeds as high as five times those of 802.11b, and wire-free multimedia content streaming. In the meantime, the standard QoS functions, such as IEEE 802.1p, 802.1q and DIFFerentiated Services (DIFFServ) for QoS will be implemented by most WLANs.

2.5 Summary

The MAC layer, providing control to access the wireless medium for users, is the critical part of IEEE 802.11. The emphasis of this chapter was on the description of DCF and PCF modes of the IEEE 802.11 MAC layer. The detailed channel access procedures were illustrated.

Various metrics for enabling delay sensitive voice on a network were studied. The capability of the IEEE 802.11 DCF and PCF modes to support VoIP was also discussed in the end. The best-effort delivery nature makes DCF unsuited for the IP telephony. Although the PCF was designed to eliminate the contentions though a polling policy, its performance was not satisfying and no commercial products are currently available. The research for the QoS support over IEEE 802.11 based on the DCF and recent development were reviewed.

More considerations about the Voice QoS provision were taken into the design of TR*Labs*' MCS prototype. Similar to the PCF mode, however, the MCS algorithm is thought perhaps more advanced in voice support. Besides, it was designed specifically to provide Internet access and voice service over large geographic coverage. How it works will be explained in the next chapter.

CHAPTER 3

TR*Labs*' MCS

TR*Labs* created the original MCS software algorithm and hardware prototype during the period 1998-2001. This concluded with D.R. Johnson who evaluated the performance of MCS in a realistic rural Yukon scenario [3]. MCS uses a prioritized polling scheme to access the radio channel, and supports integrated data and VoIP services. The MCS functions are implemented through an MCS header and the crucial parts are introduced in this chapter. The MCS behaviors are explained in detail through the packet exchange scenarios of data and voice. The packet format used in the proposed system solution is created by integrating the MCS header into the standard IEEE 802.11b frame. Three key C++ modules that have been implemented previously in NS-2 simulator are also introduced briefly.

3.1 TR*Labs*' MCS Prototype

3.1.1 Channel Access Method

In MCS, the hub controls the access to the radio channel in a sequence of prioritized polls to clients. Packets are differentiated into voice packets, data packets, system message packets, and registration packets. Voice packets have the highest polling priority, the next is data packets and system message, and the newcomer client registration has the lowest priority.

Figure 3.1 depicts the MCS radio channel utilization. The entire timeline is divided into successive time slots. The time slot, T, is somewhat arbitrary but is chosen to be small enough (for example, 30 ms for MCS) for small delay of voice packets. Each time slot is then broken into three sections for voice client polling (Tv), data client polling/system message (Td) and newcomer client registration (Tp). Voice packets are handled first, with the remainder of the time slot available for data packets, system messages, and newcomer polling.

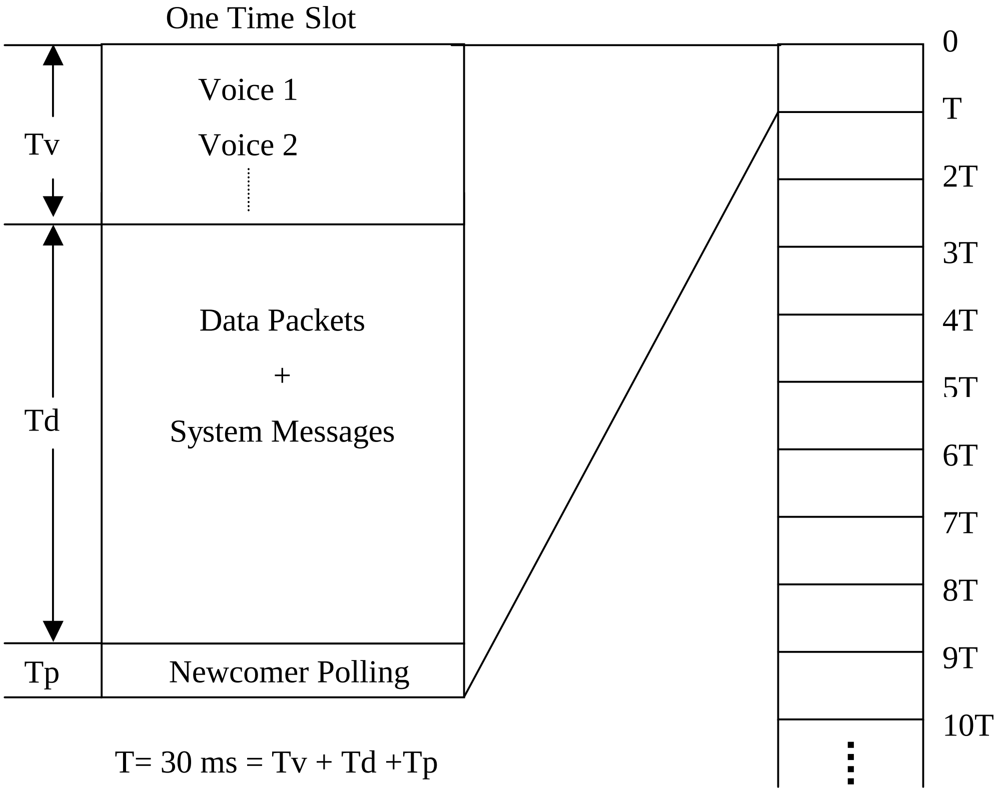


Figure 3.1 Time allocation in the MCS radio channel

Tv is dedicated to the voice packet delivery between the hub and voice clients, and is always at the start of each time slot. The length of Tv is variable, dependent on the number of voice clients.

The rest of the time left in one slot is allocated to deliver data packets and system messages between hub and clients (Td). Td starts only after the exchange of voice packets is over. The length of Td is made dependent on Tv. Since the length of Tv is variable, the boundary between Tv and Td shifts dynamically. System messages are exchanged between hub and client use to configure the overall system (i.e., setup or teardown of a voice call).

A newcomer client is the one who has been inactive (i.e., powered down) for an extended period of time. It must register with the hub to be placed on the active client list. With the lowest priority, those registration requests are dealt with during a small interval of the time slot (Tp). Tp is always placed at the end of each time slot [18]. The small part of Tp is ignored in the MCS simulations [3], since all the clients are assumed to be always in active states.

3.1.2 MCS Header Definition

The MCS hub-client pair operates at the data link layer and physical layer. The 15 byte MCS header is shown in Figure 3.2.

<- 4 bytes ->

<table>
<tr><td colspan="4">Preamble Flags</td></tr>
<tr><td>Station ID</td><td>Control 1</td><td>Control 2</td><td>UNUSED</td></tr>
<tr><td colspan="2">Parameter 1</td><td colspan="2">Parameter 2</td></tr>
<tr><td colspan="2">Parameter 3</td><td>Checksum</td><td></td></tr>
</table>

Figure 3.2 MCS header definition [18]

Following the MCS header is the Payload area with variable size (not shown in Figure 3.2). The functions of MCS are performed through the MCS header. The byte of “Control 2”, used to exchange system messages in actual MCS’ operation, has never been used in simulations [3] (the same as Parameter 1, 2, 3 [18]). Its functions are implemented through “Control 1” in simulation. Only the “Station ID” and “Control 1” are discussed here. The one-byte Station ID field contains the recipient station MCS address, for example, 255 indicates a broadcast packet, 0 is the hub’s address, and the rest are assigned to clients. Control 1 is used for the hub to indicate the client whether a voice or data token is being issued. It is also used to signal whether the Payload field contains a data packet, voice packet, or nothing at all.

The Control 1 byte is of key importance to understanding the operation of the MCS. The 8 bits in the Control 1 byte are defined individually in Figure 3.3. Bits 4 and 5 are used to perform the functions of “Control 2”.

When the MCS polling algorithm sends Client 1 a token, it sets the Station ID to 0000 0001, sets the TD or TV bit of Control 1 depending on whether a data or voice token is being issued. If a data token is being issued and there is a data packet queued for Client 1, then the IP data packet is added to the Payload area of the MCS frame and bits TD and D of Control 1 are set (Control 1 = 1000 1000). If there is no data packet for Client 1, the Payload area of the MCS frame is empty and bit D of Control 1 is not set

(Control 1 = 1000 0000). Similarly if a voice token is being issued and a voice IP packet is attached, bits TV and V of Control 1 are set (Control 1 = 0100 0100). If there is no voice packet for that client, only bit TV of Control 1 is set (Control 1 = 0100 0000). The MCS frame is then broadcast into the radio channel.

Bit 7	Bit 6	Bit 5 *	Bit 4 *	Bit 3	Bit 2	Bit 1	Bit 0
TD	TV			D	V		DA
Bit 7: Token Data, set by Hub to allow Client to reply with data. Bit 6: Token Voice, set by Hub to allow Client to reply with voice. Bit 5: Undefined Bit 4: Undefined Bit 3: Data, set by Hub or Client, indicates data packet included. Bit 2: Voice, set by Hub or Client, indicates voice packet included. Bit 1: Undefined Bit 0: Data ACK, set by Hub or Client indicating previous packet received in good order. It is used to signal reception of either a voice or data packet.							
* The simulated model in [3] actually uses bit 5 to signal a voice call setup and bit 4 to signal a call teardown.							

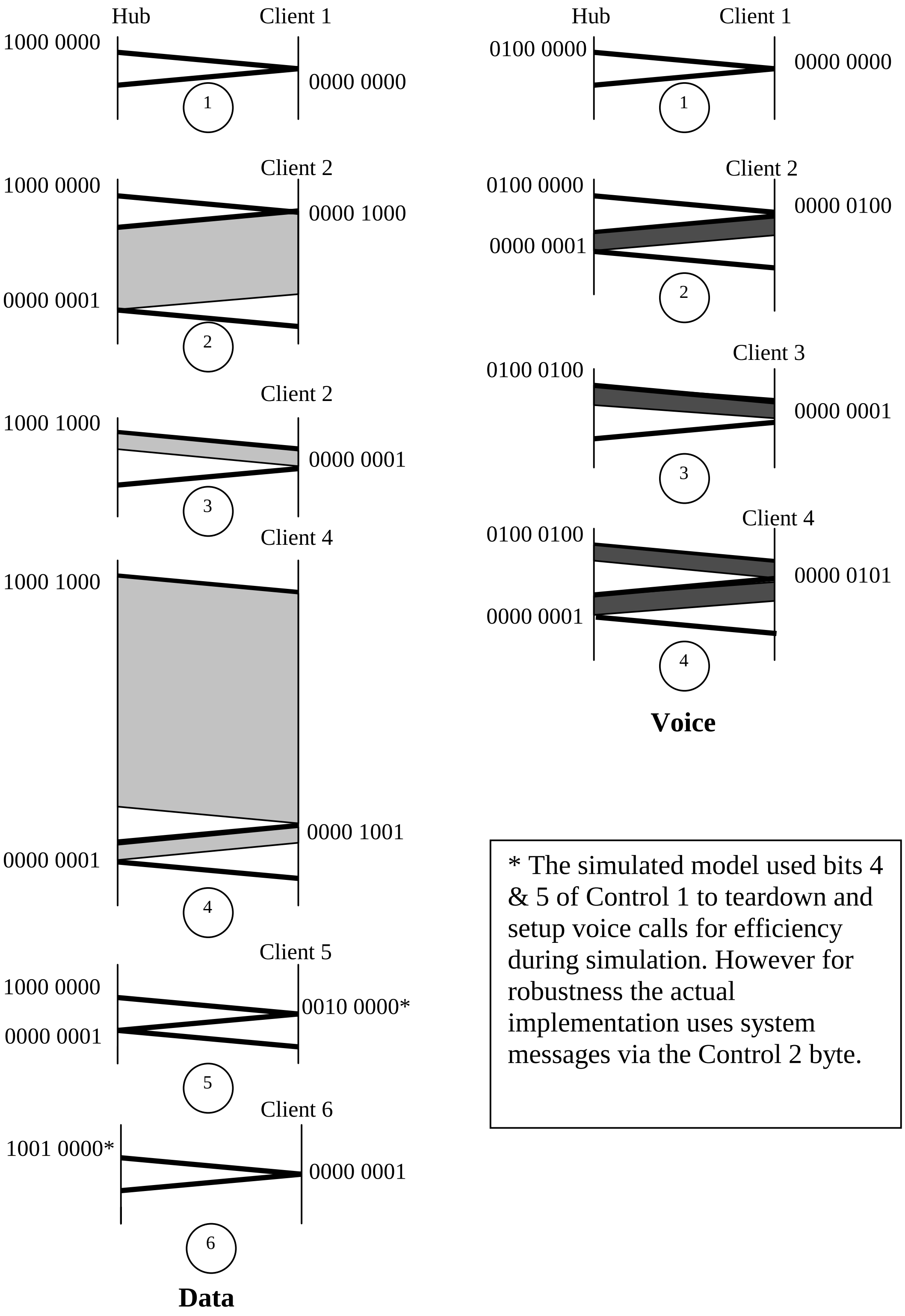
Figure 3.3 Control 1 byte of the MCS header [3]

When a client receives an MCS frame, the byte of Control 1 is checked, if either TD or TV is set, then the client gets permission to transmit a data or voice payload into the radio channel following the MCS header, with the bit D or bit V set.

Bit 0 of Control 1 can be set by either hub or client in case of acknowledging the successful reception of a received data or voice packet. The simulated model in [3] uses bits 5 and 4 of Control 1 to indicate the request of a call setup and teardown.

3.1.3 MCS Packet Exchange

The scenarios of error free packet exchange between the hub and clients are shown in Figure 3.4. These timelines are drawn roughly to scale with time increasing downward. The slope of the near horizontal lines represents the propagation delay. Since the MCS hub and client are store and forward devices, the entire frame has to be received



* The simulated model used bits 4 & 5 of Control 1 to teardown and setup voice calls for efficiency during simulation. However for robustness the actual implementation uses system messages via the Control 2 byte.

Figure 3.4 Packet exchange scenarios in MCS [3]

before any operations can be done. The thickness of the lines represents the transmission delay of an MCS frame into an MCS radio channel. The thinnest (black line) represents the transfer of a 15 byte MCS frame without payload.

On the left side are data exchange scenarios. In the first scenario, the hub sends Client 1 an empty MCS frame (no payload) with the TD bit of Control 1 set to indicate a data token is being passed. Client 1 has no data packet and returns only the empty frame. Some time later the MCS polling algorithm chooses Client 2. Client 2 happens to have a Hyper Text Transfer Protocol (HTTP) request to send and then is allowed to send it out after receiving the MCS data token. Then the hub sends back an empty MCS frame with the DA bit set, acknowledging the receipt of the HTTP request packet. Some time later, a small 40 byte TCP acknowledgement packet is sent to Client 2 in the 3rd scenario. Since no data is queued at Client 2, only an empty MCS frame with the DA bit set is returned to the hub. The payload part is empty so the hub does not need to return an acknowledgement.

The fourth data scenario is the most common. Data being downloaded to the client comes in large 576 byte chunks as payload of the MCS frame, with the TD & D bits of Control 1 set. When Client 4 receives the data packet it is ready to respond with a 40 byte TCP acknowledgment for the previous TCP data packet (not shown in Figure 3.4) it received. Then the hub replies to Client 4 with an empty MCS frame with the DA bit set as an acknowledgment.

The last two data scenarios of Figure 3.4 show the exchanges for two types of system messages, call setup and call teardown, during the data exchange process. System messages are always handled first because their priority is higher than the ordinary data exchange. Here it is assumed that all the voice calls are between an MCS client and a PSTN user. In this example the setup request is initiated by an MCS client, and the teardown request is initiated by a PSTN user through the Call Manager.

In the fifth data scenario, Client 5 responds to a data token with a voice call setup request instead of a data packet. Voice tokens are never issued to a client unless it has previously requested a call setup. Although no payload packets are passed in either direction, the hub still responds to Client 5 with an empty MCS frame acknowledging the voice call setup request and then adds Client 5 to the voice list.

If a call teardown request for Client 6 is generated, the hub sends a data token with the call teardown bit set, when polling Client 6. Client 6 responds with an empty MCS frame acknowledging the voice teardown request, as shown in the final data scenario. The hub then removes Client 6 from the voice list.

Four possible scenarios of voice packet exchange (in the right side of Figure 3.4) work in a similar way.

- ❑ Scenario 1 - No voice packets queued for both hub and client
- ❑ Scenario 2 - Only client has a voice packet to send
- ❑ Scenario 3 - Only hub has a voice packet to send
- ❑ Scenario 4 - Both have voice packets to exchange

The hub sets the TV bit instead of the TD bit to notify one client that it is time to exchange voice packets. A voice packet is appended if there is one addressed to this client. The client then responds with an MCS frame with the V bit of Control 1 set, and a voice IP packet is attached. In the four scenarios, the DA bit of Control 1 is set for hub or client to acknowledge the successful reception. In the original MCS design an explicit acknowledgment is always required for both voice and data exchanges.

The above discussion is about the successful scenarios of packet delivery. In a realistic MCS channel, it is possible that some packets sent from either the hub or clients are lost or received with errors. After the hub sends out a data or voice packet, it waits on a valid reply for a timeout period (roughly equal to the round trip time of flight plus the packet transmission time). If the timeout expires with no valid reply received, a successive error counter associated with that client is incremented, and the hub turns to the next client. The last unacknowledged data packet is re-transmitted at the next polling for the client until its successful reception. Voice packets are never re-transmitted. If the successive error counter exceeds a maximum allowable value, then the resources for the client are all torn down and the client is placed on the inactive list [18].

On the client side, the client takes no action if no valid token is received. The other behaviors are similar. If the client sends out a packet, it then waits on a valid acknowledgement for a timeout period (roughly equal to the round trip time of flight plus the empty frame transmission time). If the timeout expires with no valid acknowledgment received, the client updates its successive error counter [18]. If the unacknowledged packet is a data packet, it is re-transmitted at the next polling until it is received successfully.

In the MCS simulations, it is assumed that all the clients always stay in the active state. Each client may be in a voice-active state, or a data-active state, or both. Four active lists are maintained in the hub: voice list, hot list, data list, and quiet list. The voice list includes all the clients in conversations. The occurrence of call setup and teardown will cause the variation in the voice list.

Clients in the data-active state belong to one of hot list, data list, and quiet list, according to their activities, where the hot list is polled most frequently, next is the data list and the least is for those clients in quiet list. The data state of one client may jump among three lists, as explained in Figure 3.5. At the beginning of the polling algorithm, all clients are assumed in the quiet list. Any time a data device in the quiet list answers a poll with a non-empty MCS frame, it is moved to the hot list immediately. After 2 sec of inactivity the data device is moved from the hot list to the data list. After 360 sec of inactivity a data device on the data list is moved to the quiet list. As in the quiet list, once a data device in the data list answers a poll with a non-empty MCS frame, it is moved back to the hot list.

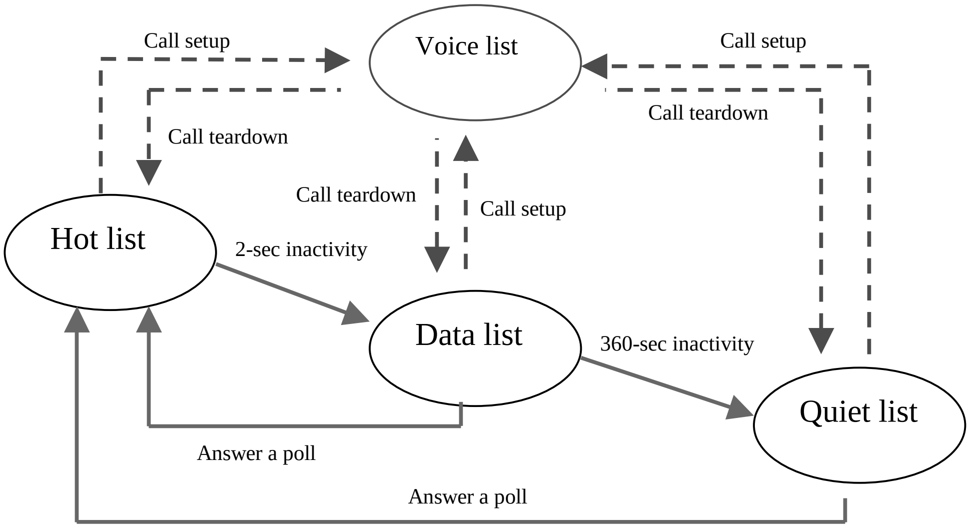


Figure 3.5 Dynamic switching of client states

The voice activity of a client is independent of its data activity. This behavior is not represented in Figure 3.5. A client can be in the voice list or one of three data lists simultaneously. During the polling process, once the hub receives a call setup request from a client (in the hot list, data list, or quiet list), the client is then added to the voice list until a call teardown signal is issued, while keeping its data state unaffected [3][18].

3.2 Incorporating the MCS into IEEE 802.11b

This section describes incorporating the MCS algorithm into the IEEE 802.11b platform. Table 3.1 lists some of the major characteristics of each.

Table 3.1 Summary of IEEE 802.11b and the original MCS

Specifications	**IEEE 802.11b**	**Original TR*Labs*' MCS**
Frequency band	Unlicensed ISM band (2.4-2.4835 GHz)	Licensed FWA band (3.4–3.7 GHz)
Data rates	1, 2, 5.5, 11 Mbps, with the ability of dynamic rate shifting.	2 Mbps
Access methods	CSMA/CA (or DCF), Polling protocol (PCF)	Polling protocol with dynamic adjustment on polling rate
Service	Provide data delivery and Internet access in DCF mode; Optional PCF mode for prioritized access.	Provide integrated data and VoIP services (voice packets have the highest priority.)
Range	Typical operating range is 50-100m (indoor) or 300-500m (outdoor).	Large geographical coverage exceeding 30 km
Product availability	Most widely used WLAN standard with mature, cost effective products, supporting roaming and mobility.	Non-standard prototype

3.2.1 PCF and MCS in Voice Service Support

The optional PCF mode of the IEEE 802.11b MAC layer was specified to support time critical services such as voice, similar to the TR*Labs*' MCS polling algorithm. It is more efficient than the MCS in how it issues ACKs, since it "piggy backs" an ACK for the last data transaction on the new poll, while MCS deals with them separately. Nevertheless, TR*Labs*' MCS is quite a bit more developed than the PCF, since PCF does not describe a method for creating and maintaining the polling list. Moreover, one MCS client can support data and voice simultaneously, because the MCS can differentiate the data packets and voice packets and treat them differently. However, the prioritization in PCF is performed at the station level. Separate data stations and voice stations are required. Stations granted high priority are put into a polling list and polled by the AP during every Contention Free Period (CFP), while others still access the channel by contention in the following Contention Period (CP).

The performance of PCF in supplying voice service is somewhat unknown in the real IEEE 802.11b WLANs, since no vendors currently support the PCF method for MAC.

3.2.2 DCF and MCS in Data Service Support

Random access protocols are always thought more suited for the Internet data transfer than polling protocols due to polling delay. However, the performance of DCF in supplying data service in wide area networks (not the typical WLANs) is still unclear and requires further exploration. Unfortunately, no research or simulation results are available for such specific application.

The MCS algorithm is more advanced and efficient than other polling protocols, because it performs dynamic adjustment of the polling rate based on the history of client activity. The polling token of the MCS, not an additional packet, is a part of MCS header sent together with or without a data or voice payload, which can further minimize the waste of bandwidth.

The system solution proposed in Chapter 1 is the combination of IEEE 802.11b and the TR*Labs*' MCS prototype realized by implementing the MCS algorithm on the IEEE 802.11b platform. In this thesis, the MCS prototype discussed above is referred to as the "original MCS". To support the hardware operation of IEEE 802.11b, modifications to the original MCS are inevitable.

3.2.3 Combined Header Format

To incorporate the MCS functions into IEEE 802.11b products, a new header format is created (shown in Figure 3.6) by incorporating the MCS header into the standard IEEE 802.11b frame. The original MCS header is shortened to 8 bytes and the redundant fields (Preamble, Station ID, Checksum, and an UNUSED byte) in Figure 3.2 are discarded, since these specific functions are effectively handled by 802.11b. The extra

MCS header is implemented in software to perform the MCS functions and no modification in the hardware is required. It is added to the Frame Body. They comprise a new “Frame Body” to the IEEE 802.11b hardware and an MPDU is then encapsulated at the MAC layer by attaching a 30 byte MAC header and a 4 byte FCS, as in Figure 2.5. A short PLCP preamble and header of 15 bytes is added at the physical layer (as in Figure 2.9) to achieve higher network efficiency. The total overhead added at the MAC layer and the physical layer has 57 bytes, including the MCS header.

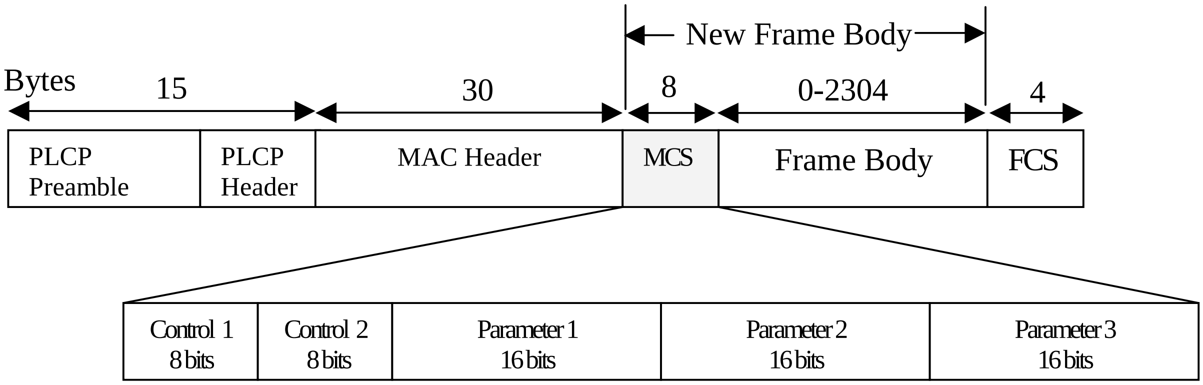


Figure 3.6 Modified IEEE 802.11b frame format

With the MCS functions incorporated into the IEEE 802.11b platform, the sending of packets is controlled by the MCS algorithm, not by contention. That is, the packets are blocked in the queues until receiving permission to send. The control of the MCS algorithm over the radio link can be realized in software and without any modification to the hardware part of the IEEE 802.11b devices.

3.3 Simulator Modules of the Original MCS

Network Simulator-2 (NS-2) is one of the most commonly used simulators today in networking research. The simulator was written in C++ and Object oriented Tool command language (OTcl) and consists of a large number of modules performing various network protocols (like HTTP, FTP, TCP, UDP). However, to mimic the behavior of the MCS, new modules have to be built inside NS-2 and incorporated with other modules. The C++ modules for the original MCS have been developed by D.R. Johnson and applied to a rural Yukon scenario [3]. The previous work provides a good start for this thesis. To understand the modifications made to the original MCS in later chapters, it is necessary to give a brief explanation about the original MCS modules in NS-2 here (For detailed information please refer to [3]).

In the modules built in [3], “MCSController”, “MCSQueue”, and “PingAgent” are crucial and they work together to implement the MCS polling algorithm. An MCS timer was created to schedule all the tasks associated with token sending. “MCSController” resides in the hub and behaves like a “commander”. It takes control of the issuing of voice or data token, knowing which client to poll and which kind of token (data or voice) to issue. “MCSQueue” mimics the behavior of the MCS links between hub and clients. “PingAgent”, modified from a built-in NS module, is responsible for the sending of voice and data tokens at the request of the “MCSController”.

A Client State Table built in the “MCSController” module (Table 3.2) contains the specific information of each client, including whether it is in voice state or not, the data state, and the time of last activity.

Table 3.2 Client State Table in the “MCSController” [3]

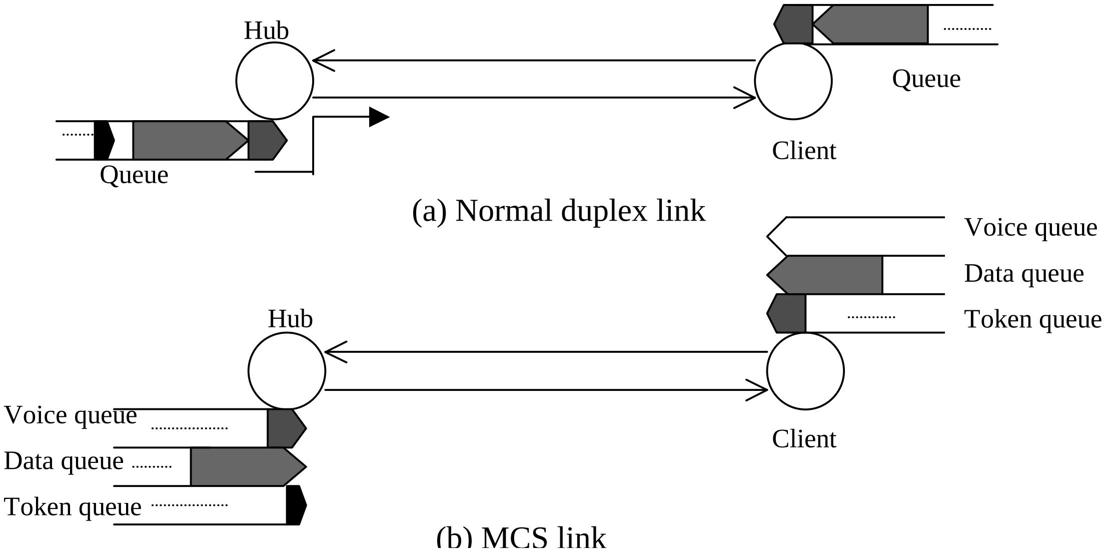
Index	**VoiceState**	**DataState**	**LastActivity**
[1]	NULL	QUIET_STATE	0.00000 s
[2]	VOICE_STATE	DATA_STATE	0.00833 s
[3]	VOICE_STATE	QUIET_STATE	0.01234 s
:	:	:	:
[60]	NULL	HOT_STATE	0.00652 s

Four process functions, ProcessQuietClient(), ProcessHotClient(), ProcessDataClient(), and ProcessVoiceClient(), are defined in the “MCSController” module, handling the packet exchange between the hub and the clients in the quiet list, hot list, data list, and voice list, respectively. Each entry of four functions is sequentially scanned until an expired time is found and the function associated with that entry is executed. After calling a function, the “MCSController” is scanned again starting from where it left off the previous time. The polling for voice clients occurs every 30 ms. The polling rate is approximately 2 polls/sec for a device on the quiet list, 20 polls/sec for a device on the data list, and the hot list is polled as frequently as the leftover bandwidth from voice allows. The function dealing with the hot list, “ProcessHotClient()” is the default process if no other time has expired.

The link model in NS-2 is used to characterize a link connecting two nodes. Link parameters (involving sending node, receiving node, bandwidth, distance and queue type) must be indicated. NS-2 has two classes of wired link: simplex link and duplex link. Packets are generated at the sending node in the end of a link and are first pushed into the queue of this link. From there they are sent out to the link following different rules. The receiving node works in the opposite way. According to the queue type used, NS links can be further divided into different types. For example, the most common queue type is “DropTail”, which sends out the queued packets in turn, following the rule of “first come first serve” and the newcomer is dropped immediately if the queue is full.

In simulation, the MCS wireless link between the hub and clients is realized by the wired links between the MCS hub and each client, and they emulate TDD under the control of the “MCSController”, just like in a wireless medium. However, no queue type has been emulated in all the contributed codes for NS-2, which is able to differentiate packet types and treat them differently in the way of the MCS polling algorithm. The “MCSQueue” used in the MCS link was created from the “DropTail”, by adding two queues to store data and voice packets separately (Figure 3.7). The original queue is used for the MCS tokens. Data and voice packets are not allowed to transmit into the link unless an MCS token comes to the queue of that node connected to the link. The MCS token is actually the MCS header, but expressed in a separate packet so that it can be easily separated from the IP packet in MCS simulations.

Figure 3.7 shows a normal duplex link (a) and an MCS link (b) in NS-2 connecting the MCS hub and one of clients. The normal link only has one queue for one direction, so once a packet is received it is forwarded out on the link immediately regardless of the packet type. In the MCS link, however, the received packet is queued into one of three separate queues: voice queue, data queue, or MCS token queue, depending on the packet type. The MCS token queue still behaves like the “DropTail” queue, but the voice and data queues are “blocked” queues, meaning that they cannot forward the packet freely. After a data packet or voice packet is received, it is usually blocked from going to the MCS link, and waits in the queues until an MCS token for data or voice is received by the MCS link. When a voice token is received, a queued voice IP packet is sent out along the link following the voice token. When a data token is received, the first packet in the data queue is sent out to the link right after the token. If either the voice queue or data queue is empty, only the voice token or data token is sent into the link.



(a) Normal duplex link

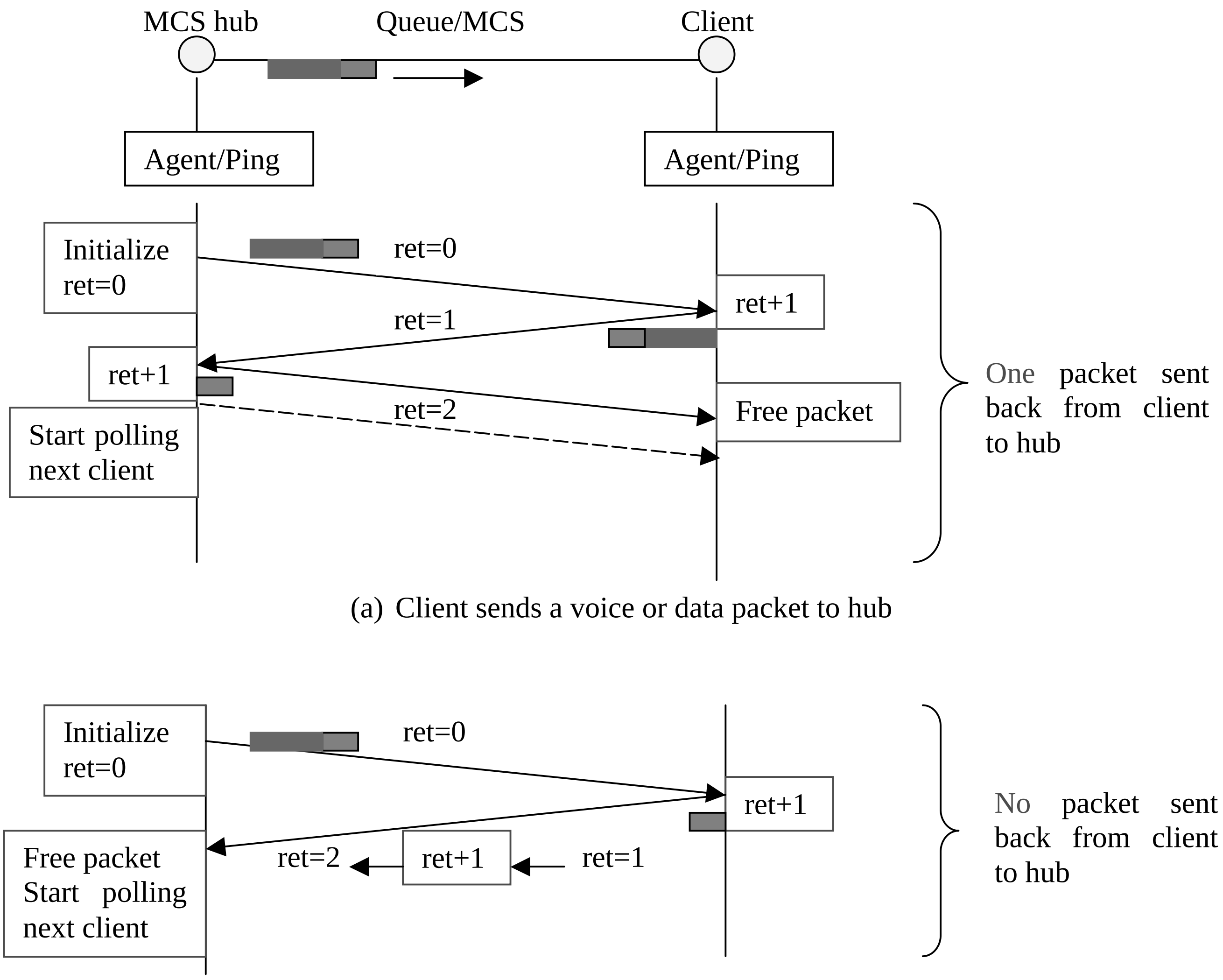
(b) MCS link

Figure 3.7 Duplex link and MCS link in NS-2

The "PingAgent" pair at the two nodes of each MCS link is responsible for generating the ping packets (used as the MCS headers). The ping packet has the ability of bouncing back and forth automatically between the pair of ping agents [19]. The "ret" field of the ping packet is used to indicate how many times this ping packet has been reflected. It is initiated to 0 after being generated and is increased by 1 when the other ping agent receives this ping packet, then sends it back, and so on, until its "ret" is accumulated to the maximum setting.

In MCS, the ping packet with "ret" equal to 2 is freed immediately after being received. Only the ping agent attaching to the hub can initiate a ping packet under the control of "MCSController". Figure 3.8 shows how the "ret" varies during one polling cycle. If the hub has a packet directed to the client, it sends the packet following the token. The client receives the token, increase the "ret" to 1 and then sends it back. If it has a data or voice packet for the hub, the packet is sent out following the token (see Figure 3.8(a)). After the hub receives the token, the "ret" is increased to 2 and the token is sent back to the client, notifying the client to free this token. Now one exchange scenario between hub and client is finished. The control of MCS returns to "MCSController" and the next polling cycle begins.

The "ret" is increased as 0->1->2 in the above process by the ping agent at the hub or client node. One exception (Figure 3.8(b)) occurs when the client receives the polling token from the hub, but has no packet to send back (data exchange scenario 1, 3, 6 and voice exchange scenario 1, 3 in Figure 3.4). When the hub gets the empty token, it is unnecessary to send back an ACK to the client. In this situation, the increasing of "ret" from 1 to 2 is implemented by the MCS link (not the ping agent), before the token arrives at the hub. Therefore, the token is to be discarded immediately after received by the hub and the next polling begins immediately.



(a) Client sends a voice or data packet to hub

(b) No voice or data packet is sent back from client to hub

Figure 3.8 The MCS token during the original MCS polling process

3.4 Summary

This chapter describes the channel access method of the MCS polling algorithm, which lies at the heart of the MCS and determines the data throughput for clients and ensures voice prioritization and delay minimization. Hot data users in busy status are polled much more frequently than quiet users, resulting in higher data throughput. However, data polls are routinely interrupted in order to process the voice list. All voice clients are then given the opportunity to pass the delay sensitive voice packets across the radio channel. The detailed polling process was illustrated through the scenarios of packets exchange for data and voice traffic.

The MCS is more advanced than the PCF to support voice service. The DCF was specially designed for the Internet data transfer in WLANs. A combined header format was created through inserting a shortened MCS header into the standard IEEE 802.11b data unit. The MCS algorithm is performed inside of 802.11b and takes control of the packet sending.

Key MCS modules implemented in NS-2 previously were reviewed. In the following chapters, the original MCS algorithm will be first directly applied in the proposed network topology. Necessary modifications to the original MCS algorithm will be made. The next chapter will simulate and discuss the performance of data transmission in the proposed rural network, using the MCS algorithm.

CHAPTER 4

MCS' DATA SIMULATION AND IMPROVEMENT

In this chapter the original MCS algorithm is applied to the network proposed in Section 1.4.1 for rural areas. The performance of data transmission is tested and evaluated in constant rate radio channels. On an 802.11b platform, the Automatic Rate Fallback (ARF) feature is incorporated into the original MCS algorithm. However, when clients with different data rates use one common channel, the MCS algorithm is found not suited for the multi-rate data transmission.

A modified data polling scheme is proposed based on the original MCS. After polling one client, the hub can keep polling the same client before turning to the next one. The number of repeated polling cycles for one client depends on its data rate. The modified scheme is simulated adjusting the repeated polling cycle setting for each client. A more realistic scenario integrating the FTP, HTTP, and voice clients scattered in four regions is also simulated.

4.1 Network Topology in NS-2 for MCS Simulation

The topology of the proposed network in NS-2 consists of the links and nodes, as shown in Figure 4.1. A relay station is used to connect the frequency translators to the MCS hub station. The wireless link between clients and the MCS hub includes three parts: the transmission from a client to a frequency translator inside a pico cell (50-100 m), the transmission from the frequency translator to the relay station (about 1 km), the relay transmission from the relay station to the MCS hub (may exceed 30 km). The third one is the major part of the whole link, and is called the "relay distance", together with the second part.

The web servers connected to the MCS hub are used to generate data traffic. The Call Manager (CM) provides the interface to the PSTN. Since only nodes working above the physical layer (the store-and-forward network devices) are provided in NS-2, a new node type performing the functions of the physical layer is created and used in the simulations, for example, the relay station and the frequency translators in Figure 4.1, which perform the power amplification and frequency translation at the physical layer.

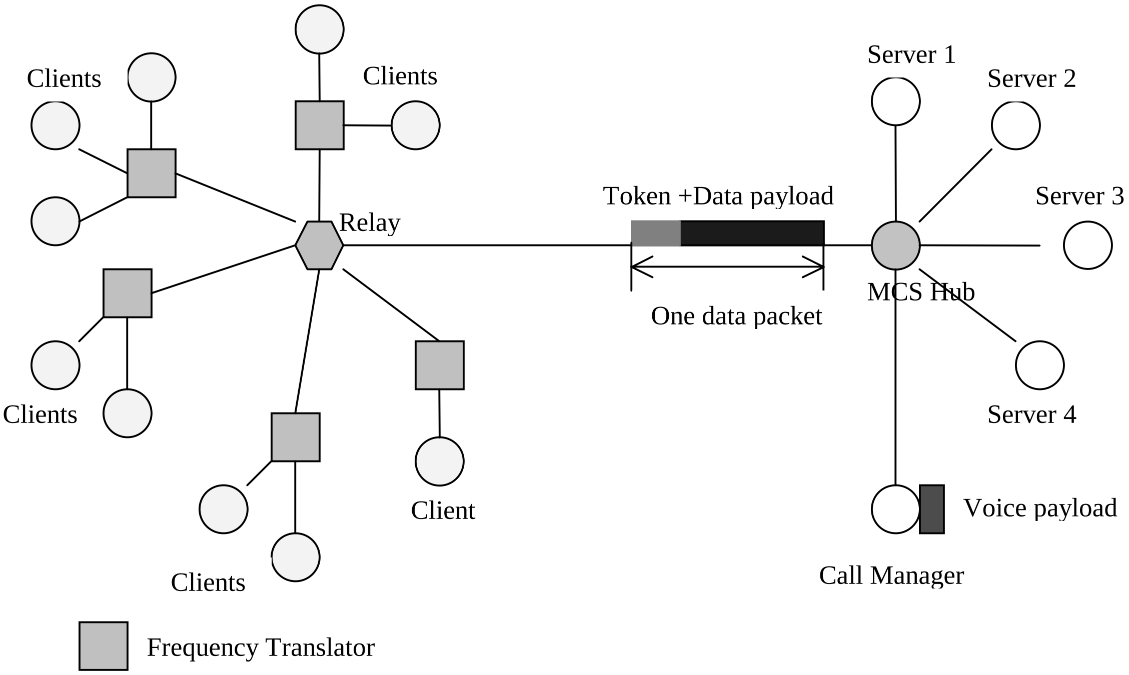


Figure 4.1 Network topology in NS-2

D.R. Johnson simulated data performance using the 576 byte and 1500 byte data IP datagrams in [3]. Although “increasing the data payload increased the MCS channel efficiency”, it was concluded that 1500 byte data packets perform much worse under higher BER conditions [3]. Therefore, the IP datagrams in the length of 576 bytes are used in the following simulations of the Internet data applications, which are passed down from the network layer to the LLC layer.

An 8 byte overhead (3 byte LLC + 5 byte SNAP) is added at the LLC layer to encapsulate the IP datagram into a 584 byte MSDU, which is then handed down to the MAC layer. The 40 byte TCP acknowledgment packet [3] is encapsulated into a 48 byte MSDU. The final packets are formed by adding the 57 byte combined overhead of the IEEE 802.11b and MCS. The 57 byte overhead is treated as a separate packet in the simulations, called the token.

4.2 Simulation Results in Constant Rate Channels

Initially, all the clients in Figure 4.1 are assumed to access the channel at the same data rate (one of 1, 2, 5.5, or 11 Mbps). The variation of the data throughput versus the data offered load is tested under different conditions. The parameters used in the simulation results are explained below.

The data throughput and data offered load, expressed in bits per second, are universally used in data communications. Their definitions are different from layer to layer in the Internet model. For example, in an IEEE 802.11 wireless network, the offered load is commonly defined to be the average number of bits per second passed down to the MAC sublayer at the source, and the data throughput is the average number of bits per second passed up from the MAC sublayer at the destination [9]. If not specifically indicated, the offered load and data throughput used in the following simulations comply with these definitions. The MSDU, which is the payload part at the MAC layer, is used to calculate the data throughput and the data offered load. Here the data throughput is the aggregate throughput received by all the clients. Approximate throughput per station can be calculated by dividing the aggregate throughput by the total number of data stations involved [9].

Another definition of the data throughput perhaps more useful to the end users is the net throughput, which is the data throughput received at the application layer. It is used to indicate how many bits of the “pure” data without any overhead can actually be received per second at the destination.

In the data throughput simulations, the traffic sources are chosen to be Constant Bit Rate (CBR) supported by the TCP at the transport layer. While the packet size is kept constant in the simulations, the packet interval between two consecutive packets can be adjusted to offer the variable data load.

4.2.1 Effect of Relay Distance on Data Throughput

For the initial test, it is assumed that 10 clients are downloading files from the servers (the CBR traffic sources) through the hub in Figure 4.1 and no voice calls are set up during the data transmission. The aggregate throughput is measured while the offered data load is gradually increased to the channel rate. Figure 4.2 shows the aggregate throughputs achieved at three settings of relay distance (0.5 km, 15 km, and 30 km), assuming the channel rate is 11 Mbps. When the sending rate is far below the channel rate, the data throughput is equal to the offered load and increases proportionally with it, until reaching the maximum value. After that the throughput stays at the maximum value, in spite of the continual growth of the load.

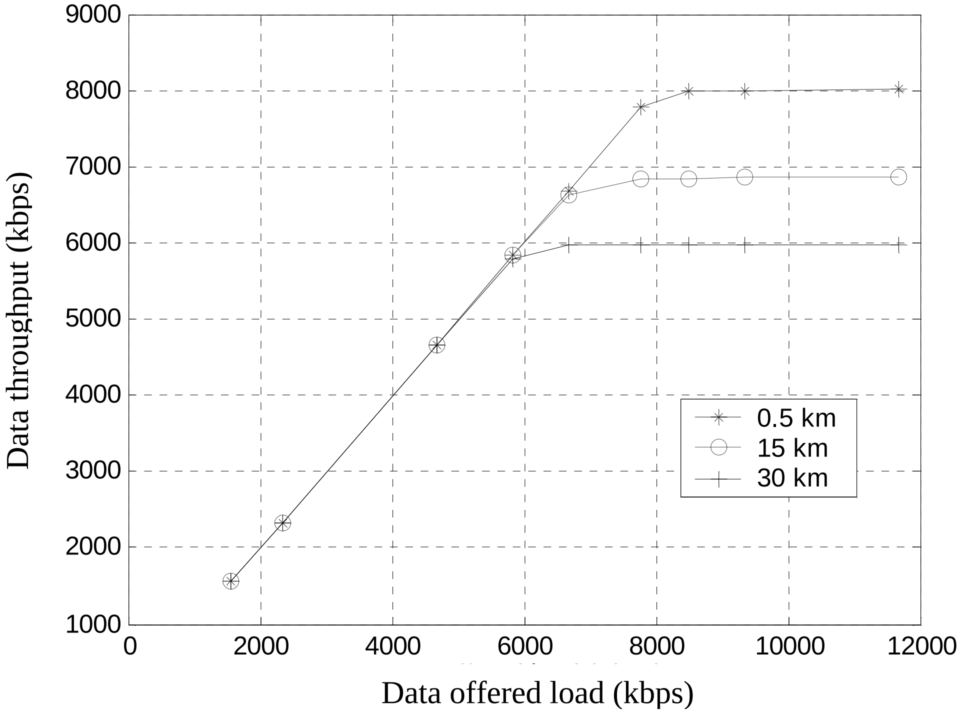


Figure 4.2 Effect of relay distance on data throughput

The maximum throughput that the system can carry is defined as the saturation throughput, indicating the transmission capability of a channel. The saturation throughput is always less than the channel rate due to the presence of delay and overhead. The saturation throughput divided by the channel rate is defined as transmission efficiency or normalized saturation throughput.

The relay distance adversely affects the saturation throughput, as shown in Figure 4.2, because the hub has to take more time to poll each client at the longer relay distance. The highest saturation throughput among three curves, 8008.6 kbps, is achieved at the distance of 0.5 km, but in the rural environment (such as 15 km or 30 km) the saturation throughput can only reach 6856.3 kbps or 5976.7 kbps respectively. (The approximate throughput per station is therefore about 685.6 kbps or 597.7 kbps.) The saturation throughput at the application layer is calculated to be 6290.3 kbps and 5485.5 kbps, using the data payload at the application layer (by deducting the 20 byte IP header and 20 byte TCP header from the 576 byte IP datagram), for 15 km and 30 km, respectively.

4.2.2 Effect of Number of Voice Calls on Data throughput

The exchange of voice packets has higher priority than data packets in each time slot. Increasing the number of voice clients means less time is left for data transmission and lower saturation throughput is expected. In an 11 Mbps channel at the relay distance of 15 km, the saturation throughput, 6856.3 kbps, 5496.2 kbps, or 4137.3 kbps, is achieved when 0, 20, or 40 clients have set up calls through the CM, as shown in Figure 4.3.

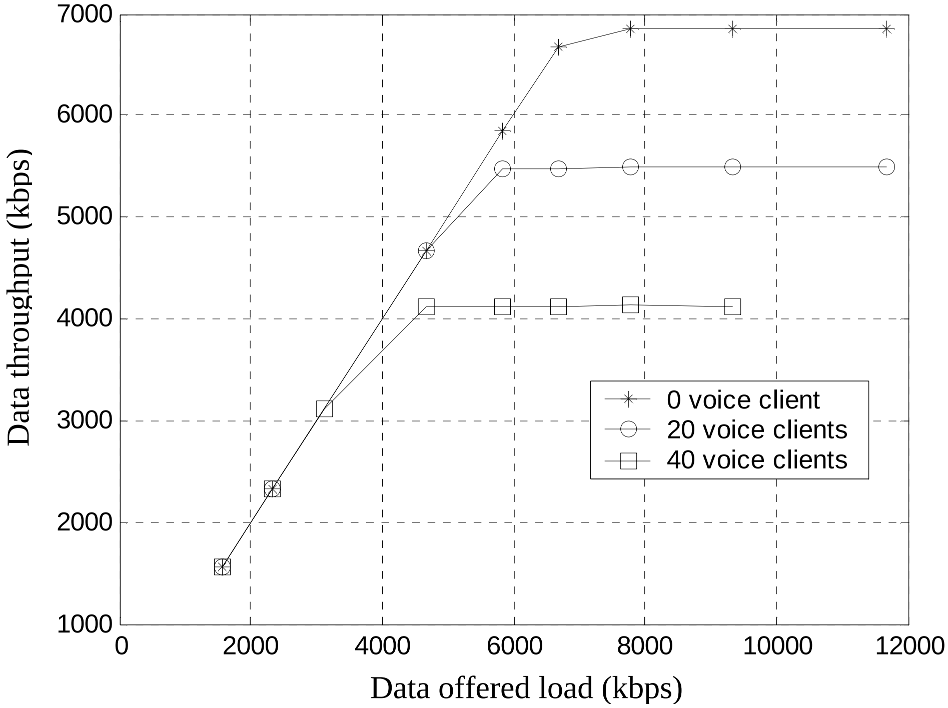


Figure 4.3 Effect of the number of voice calls on data throughput

4.2.3 Effect of Channel Rate on Data Throughput

In IEEE 802.11, different data rates are achieved in the wireless channels, depending on the channel conditions. The throughput at four data rates is shown in Figure 4.4, for a relay distance fixed at 15 km and no voice calls. As expected, the saturation throughput depends largely on channel rate, and an 11 Mbps channel having the highest saturation throughput.

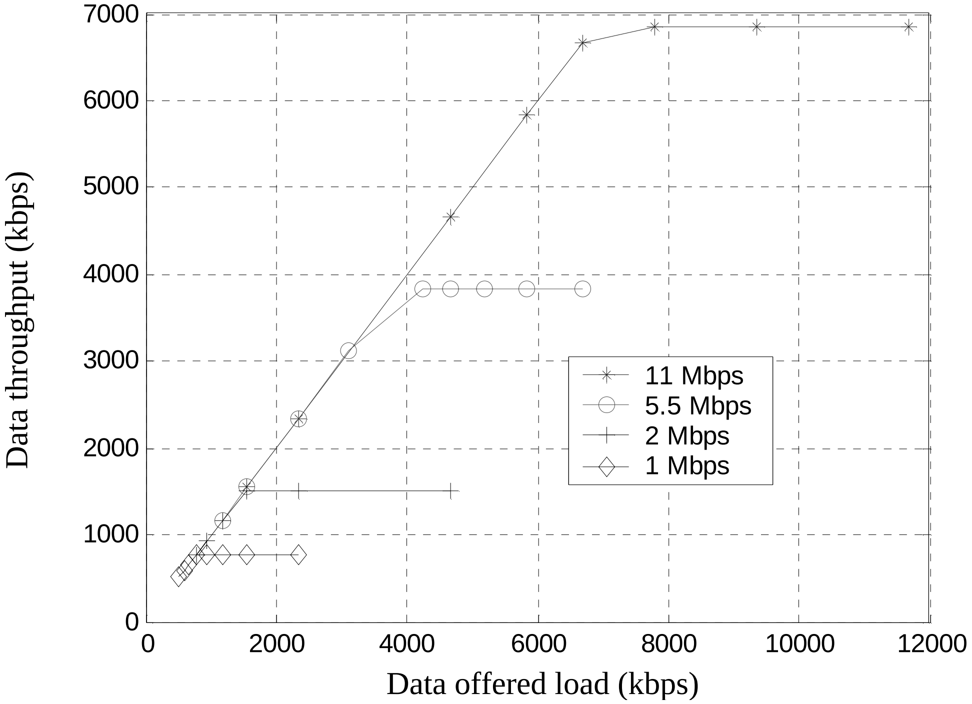


Figure 4.4 Effects of channel rate on data throughput

The saturation throughput available at the MAC layer and the application layer (the net throughput) at four data rates are listed in Table 4.1 and shown in Figure 4.5.

Table 4.1 Saturation throughputs and efficiencies at four data rates

Data rate **Mbps**	**Throughput (MAC)** **Mbps**	**Efficiency (MAC)**	**Net throughput** **Mbps**	**Net Efficiency**
1	0.72	72.0%	0.66	66.2%
2	1.42	71.0%	1.30	65.2%
5.5	3.70	67.4%	3.40	61.8%
11	6.86	62.3%	6.29	57.2%

Table 4.1 also includes the transmission efficiency at the application layer and the MAC layer. Obviously the saturation throughput goes up with data rate, but the efficiency is degraded quite noticeably. Throughput efficiency (η) for a system is defined as the ratio of time taken successfully transmitting payload bits to overall time [3]. It is

used to measure the efficiency of data and voice transmission, and is different from layer to layer.

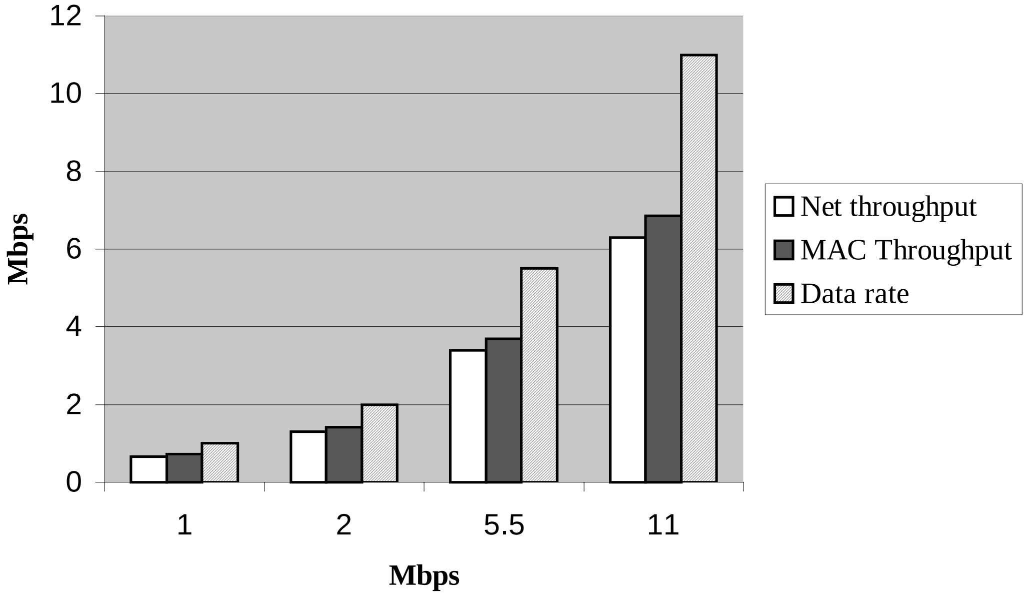


Figure 4.5 Saturation data throughputs in four Channels

The adverse impact of increasing the MCS channel rate on the efficiency can be explained using Data Scenario 4 of Figure 3.4. Its efficiency at the MAC layer is expressed in Equation (4-1). The third propagation delay is not factored into the denominator, since the hub does not have to wait for the client to receive the empty MCS frame before starting the next packet exchange.

$$\eta = \frac{\delta_{trans}(584B)}{\delta_{trans}(584B) + \delta_{trans}(48B) + 3\times\delta_{trans}(57B) + 2\times\delta_{prop}(15km)} \tag{4-1}$$

The efficiency at the MAC layer with data rate 1, 2, 5.5, or 11 Mbps is 71.6%, 70.5%, 67.0%, or 62.1%, calculated by Equation (4-1) (listed in Table B.2, APPENDIX B), which are very close to Table 4.1.

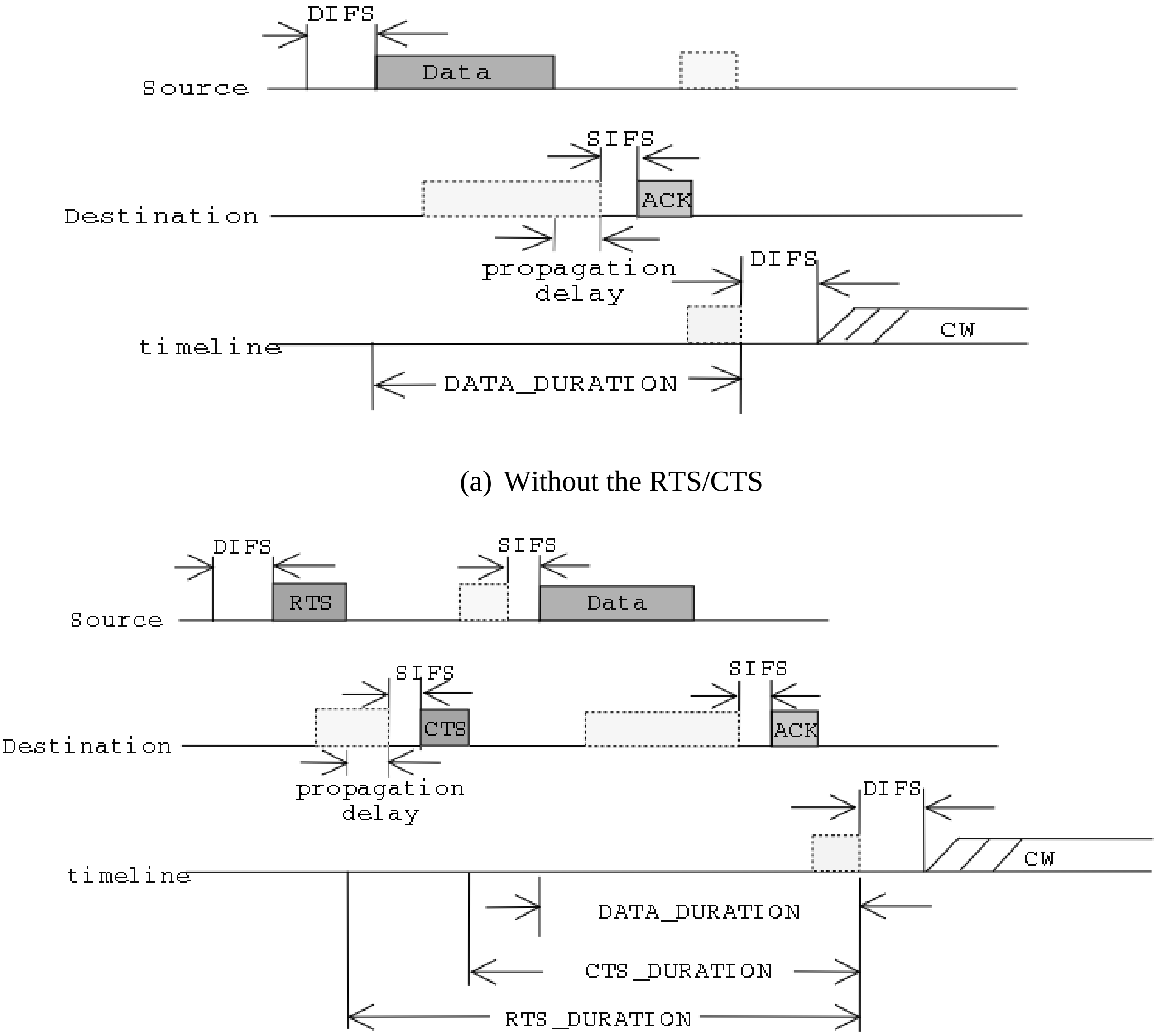
Using higher data rate can effectively decrease the transmission delay (δ_{trans}) in the numerator and denominator of Equation (4-1), but the propagation delay (δ_{prop}) in the denominator cannot be decreased, since the propagation distance is kept constant in this simulation. Therefore, the efficiency is lower when the higher channel rate is used. It must be indicated that the PLCP short preamble is always transmitted at 1 Mbps, and the PLCP header is always transmitted at 2 Mbps in a practical network, regardless of the transmission rate of the remaining portion. However, in our simulation the entire packet including the PLCP preamble and header are all transmitted at the same channel rate. Therefore, the simulated data throughput and efficiency presented in this thesis could be a little higher than in a realistic system, especially in the high data rate applications of 5.5 or 11 Mbps.

4.2.4 Results Discussions

Since no formulae, or published simulation results have been found for the similar network topology of Figure 4.1, to evaluate the above simulated data performance, the simulation results of DCF operation in an IEEE 802.11 infrastructure network in [11] and [13] are used as the references. With the average MSDU length of 1000 bytes and the channel rate of 1 Mbps, the efficiency at the MAC sublayer is on the order of 75.0% in [11] and [13] in the case of 10 data users and 0 voice users involved. In Figure 4.2, at the relay distance of 0.5 km, the saturation throughput in an 11 Mbps channel is 8008.6 kbps, giving the efficiency at the MAC sublayer by 72.8% after divided by the channel rate. A higher efficiency, 82.4%, can be achieved, if the MCS algorithm is simulated in the similar conditions to [11] and [13], for example, with no relay distance added, the channel rate lowered to 1 Mbps, and the 1000 byte MSDU used. It shows that the MCS algorithm is more efficient than the DCF mode in data transmission, if both are simulated in similar conditions.

The saturation throughput (or the data efficiency) in Figure 4.2 is degraded with increasing relay distance. For example, when the relay distance is 30 km, the efficiency available at the MAC sublayer in an 11 Mbps channel is only 54.3%. Although the DCF mode was designed for a typical WLAN application and not for long distance propagation, it is expected that the DCF operation also faces the similar situation, if used in a wide area network.

In the possible application of the DCF in rural areas, the effects of propagation delay, usually neglected in the WLANs because of the small range, must be considered and the MAC layer protocol also needs to be modified accordingly. Figure 4.6 shows the modified DCF operations without the RTS/CTS (a) and using the RTS/CTS (b), when the propagation delay (about 50 µs –200 µs from 15 km to 60 km) is included.



(a) Without the RTS/CTS

(b) Using the RTS/CTS

Figure 4.6 Transmission of an MPDU in the modified DCF

Compared to the original DCF (shown in Figure 2.6 and 2.7), after the hub sends a packet out, the receiving station waits some time to receive the first bit because of the propagation delay between them. The additional waiting time slows the entire process down and results in lower transmission efficiency. The longer the relay distance is, the lower the efficiency will be. The duration field, used by other stations to set their NAV timers (indicated by DATA_DURATION, RTS_DURATION, and CTS_DURATION in the DCF module of NS-2), must be modified to include the additional propagation delay.

With the addition of voice users in the same channel, a portion of channel capacity is occupied because either the assignment scheme proposed in [11] [13] or the MCS algorithm guarantees higher priority for voice traffic. In the case of 10 data users, the MCS' data efficiency at the MAC sublayer is 62.3%, 50.0%, or 37.6%, respectively, with 0, 20, or 40 voice users included in an 11 Mbps channel at the relay distance of 15 km (Figure 4.3). With the improvement of MCS' voice transmission efficiency (explored in Chapter 5 and 6), more time in each MCS slot can be left to the data exchange and thus the data transmission efficiency can be further increased.

4.3 Incorporating the ARF into MCS

The original MCS prototype with a single data rate (2 Mbps) was designed to discard the erroneous packet immediately after it is detected, and retransmit the discarded packet during the next polling cycle. The IEEE 802.11b network also performs the retransmission. Besides, when the received signal quality falls below a pre-set level, the data rate is allowed to drop to a lower one automatically, using the Automatic Rate Fallback (ARF) algorithm that performs the dynamic rate switching of IEEE 802.11b.

4.3.1 Automatic Rate Fallback in the IEEE 802.11b Wireless LANs

Four data rates (11, 5.5, 2, 1 Mbps) are supported in the IEEE 802.11b wireless LANs. Each client station in a BSS is assigned one of the data rates to communicate with the AP according to the detected signal strength. The data rate is adjusted automatically during the communication when the channel conditions change (for example, when the station moves to another location, the propagation path is blocked by passing objects, some unpredictable interference occurs, etc.). To simplify the analysis without losing generality, the effects of all the factors are attributed to the location change of the stations. The BSS is then translated into different reliable communication ranges for different data rates, 1 Mbps gives the largest range, and 11 Mbps is only available in the smallest one.

Figure 4.7 illustrates a simplified BSS model consisting of four cell regions associated with four data rates. It is assumed that all the stations are scattered in four regions. The ARF algorithm was designed to ensure the usage of the highest practical data rate at each moment. It can cause a fallback to a lower data rate when a station wanders from an inner region to an outer region or receives higher level of interference, or an upgrade to a higher rate when it moves back into the inner region or in better channel conditions. The ARF functions when the ARF boundary is crossed in either direction [22].

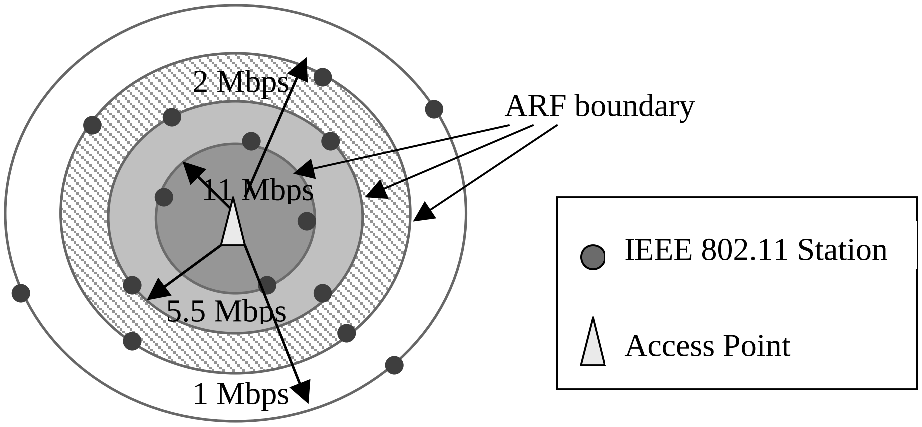


Figure 4.7 Relation between data rates and cell regions

4.3.2 Improvement in Data throughput using the ARF

Achieving the highest data throughput possible is the objective of a data communication network, either by improving the channel Bit Error Rate (BER) (to reduce the packet retransmission rate) or using higher data rates. However, the two approaches are conflicting. Simulations are needed to conclude if the ARF is helpful to the data throughput and how much improvement can be achieved.

With the assumption that BER is the only reason to cause the packet discarding, the Packet Loss Rate (PLR) is equal to the Packet Error Rate (PER). The condition for a packet to be received correctly is that every bit included is received without errors. Assuming that the bits are detected independently, the PER given in Equation (4-2) is a function of the channel BER and the packet length in bits (L). Examples of the conversion from BER to PER using Equation (4-2) are listed in Table 4.2.

$$PER = 1 - (1 - BER)^{L} \tag{4-2}$$

Table 4.2 Conversion from BER to PER

Payload	584Bytes	
Header	57Bytes	
Total Bits	5128bits	
BER	PER	1/PER
1.00E-09	0.001%	195008
1.00E-08	0.005%	19501
1.00E-07	0.051%	1951
1.00E-06	0.511%	196
1.00E-05	4.999%	20
1.63E-05	8.000%	12
4.35E-05	20.000%	5
7.91E-05	33.333%	3
1.00E-04	40.120%	2
1.35E-04	50.000%	2
1.00E-03	99.409%	1
1.00E-02	100.000%	1

On average, one retransmission occurs every 1/PER packets. In the simulation the retransmission is realized through preventing one data packet into the MCS link every 1/PER packets. This packet is being paused in the queue at the first polling to the client and is sent out at the next polling. The number of transmitted data packets is accumulated during the simulation. When the total reaches 1/PER, one retransmission is performed, the counter is reset, and a new count begins.

Since it is intended to realize the MCS algorithm over available IEEE 802.11b products, the ARF will work automatically as a default operation at the hardware level. To illustrate, the original MCS algorithm is simulated in an 11 Mbps radio channel without the ARF and then the ARF is enabled for comparison, using a wide range of BER: 10^{-9}, 10^{-8}, 10^{-7}, 10^{-6}, 10^{-5}, and 10^{-4} (see the un-shaded area of Table 4.2 for the corresponding values of PER). The saturation throughput shown in Figure 4.8 is not noticeably degraded in the range of BER from 10^{-9} to 10^{-5}, but for BER below 10^{-5}, it drops rapidly if the channel rate is always kept 11 Mbps.

The ARF is assumed to work at high BER. For example, in Figure 4.8 the data rate falls back to 5.5 Mbps at the BER, 10^{-4} and 10^{-3}, and the BERs can be upgraded to 10^{-8} and 10^{-6} (Table A.2, APPENDIX A), according to the BER curves of CCK (Figure A.1, APPENDIX A). It is seen that reducing the data rate can effectively improve the data throughput under bad channel conditions if all the clients always keep the same data rate.

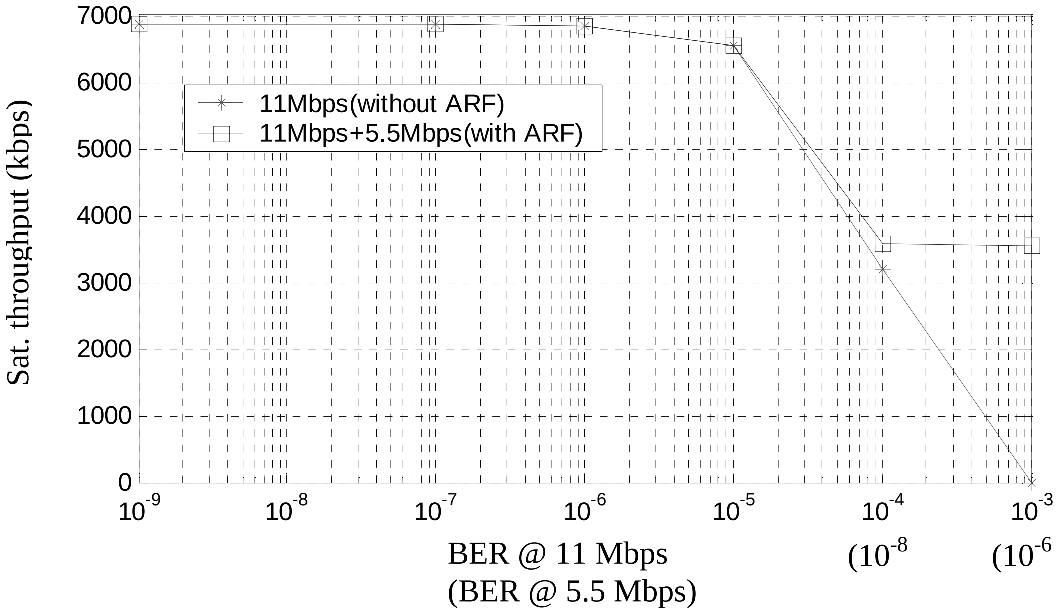


Figure 4.8 Improved throughputs using ARF at high BER

4.4 Simulation of the Original MCS in a Multi-Rate Channel

The above simulations were conducted assuming all the clients use the same data rate. However, in a realistic system, clients with different data rates may coexist in a BSS using one common channel. For the multi-rate simulations, it is assumed that the clients are distributed in the four regions of Figure 4.7. Four groups are used in the simulations and those clients using the same data rate belong to the same group. Given the experimental ranges associated with the four data rates in the IEEE 802.11b WLAN environments [22] (Table A.3 in APPENDIX A), the ranges in the “Semi Open Office” are used as the distances from clients to the frequency translator inside each pico cell of Figure 4.1. The distance from the frequency translator to the MCS hub is always fixed at 15 km in the following simulations.

The dynamic rate variation of clients is not considered and the operation of the whole network is assumed in a steady state. That is, the number of clients in each group is constant during the simulations and is known by the hub at all times. It is felt that this is a reasonable consideration in a fixed wireless network over the relatively short durations of the tests. Two possible cases are discussed here. In the first one all the clients keep the highest data rate (11 Mbps), but have different PERs, depending on their locations. The second case is opposite, where the clients use different data rates to keep the PERs relatively constant and always below some specific threshold. An optimistic case is used as a reference, assuming that the four groups have the same data rate and keep the PER always below the threshold.

Some vendors specify the receive sensitivity for their 802.11 adaptors (such as the D-Link products). When the fall back of the data rate occurs at some given received signal strength, the PER is no worse than 8% [23]. From Table 4.2, the PER 8% is given by the BER 1.63×10^{-5}, which can provide a saturation throughput without significant dropping from Figure 4.8. Therefore, the PER 8% is used as the PER threshold in the following simulations. In the first case, the four groups of Figure 4.7 are assumed to have the PERs listed in the shaded area of Table 4.2, 8.0%, 20.0%, 33.3%, and 50.0%, respectively, from the innermost region, while in the second case the ARF is used to cause the fall back of data rate. Four groups are assumed to keep PER 8% and be distributed in all four regions.

In the reference case, the data rate is set to 11 Mbps at first for all clients (Table 4.3(a)), and is changed to 5.5, 2, and 1 Mbps in turn for all clients. The simulation results corresponding to the four data rates are listed in Table 4.3(b). All the clients use the same data rate to access the channel. With the PER of 8% (corresponding to the BER of 1.63×10^{-5}), the aggregate throughput in an 11 Mbps channel, 6334.3 kbps, is degraded slightly from the highest saturation throughput achieved at a very low BER (6856.3 kbps, see Figure 4.8). Changing the data rate from 11 Mbps to 1 Mbps causes the drop of the data throughput received by each client from about 156.0 kbps to 17.5 kbps.

Table 4.3 Simulation of the reference case

(a) Simulated network parameters

Group No.	Group 1	Group 2	Group 3	Group 4
Distance to the center (m)	56	69	85	105
Data rate (Mbps)	One of 11, 5.5, 2, or 1 for all groups			
PER	8%	8%	8%	8%
Number of clients	10	10	10	10

(b) Simulation results of the saturation throughputs

Data rate (Mbps)	For single client (kbps)				For total clients (kbps)
	Group 1	Group 2	Group 3	Group 4	
11	156.0	155.8	155.4	155.0	6334.3
5.5	86.7	86.6	86.4	86.1	3521.7
2	34.1	34.0	33.9	33.8	1382.9
1	17.5	17.5	17.4	17.4	708.0

In Case 1, the highest data rate is maintained in the four groups, but the PER changes with the location. The data throughput of each client varies from 156.0 kbps to 89.8 kbps from Group 1 to Group 4 (Table 4.4), due to the increased retransmission rate caused by the PER, which also seems somewhat “fair” as the clients with the best channel conditions still receive the highest throughput. This is consistent with the MCS behavior observed by Johnson [3].

Table 4.4 Simulation of Case 1

(a) Simulated network parameters

Group No.	Group 1	Group 2	Group 3	Group 4
Dist. to center (m)	56	69	85	105
Data rate (Mbps)	11	11	11	11
PER	8%	20%	33%	50%
Number of clients	10	10	10	10

(b) Simulation Results of the saturation throughputs

For total clients (kbps)	5085.2			
For single client (kbps)	Group 1	Group 2	Group 3	Group 4
	156.0	137.4	116.2	89.8

The ARF is applied in Case 2, where the four groups in the same channel use different data rates and the PER 8% is maintained for all. However, in the simulation results listed in Table 4.5, all the clients have nearly equal data throughput, regardless of their actual data rates. Even though some clients have the data rate as high as 11 Mbps, they can only get throughput of about 37.5 kbps, the same as the other lower rate clients. The unexpected result reflects the weakness of the MCS algorithm in multi-rate application, as the throughput of the highest rate clients is dragged down toward the rate of the lowest rate clients in the system.

Table 4.5 Simulation of Case 2

(a) Simulated network parameters:

Group No.	Group 1	Group 2	Group 3	Group 4
Dist to center (m)	56	69	85	105
Data rate (Mbps)	11	5.5	2	1
PER	8%	8%	8%	8%
Number of clients	10	10	10	10

(b) Simulation Results of the saturation throughputs

For total clients (kbps)	1538.9			
For single client (kbps)	Group 1	Group 2	Group 3	Group 4
	37.5	37.4	37.4	37.3

In the above simulations, the clients are assumed equally distributed in four groups. The performance of MCS in multi-rate application is investigated further through altering the numbers of client in each group. In Table 4.6 the total number of the clients is still fixed at 40, but the client numbers in different groups are adjusted as listed in (a) and (b). It is seen that the throughput of each client is still almost equal in either (a) or (b), but (a) and (b) have different throughput for each client. Obviously, larger data throughput is achieved when there are more clients using higher data rates.

Table 4.6 Simulation results in different distributions

(a)

Group No.	Group 1	Group 2	Group 3	Group 4
Dist. to center (m)	56	69	85	105
Data rate (Mbps)	11	5.5	2	1
PER	8%	8%	8%	8%
Number of clients	20	10	8	2
Throughput each client (kbps)	65.3	65.1	64.9	64.7
Total throughput (kbps)	2678.8			

(b)

Group No.	Group 1	Group 2	Group 3	Group 4
Dist. to center (m)	56	69	85	105
Data rate (Mbps)	11	5.5	2	1
PER	8%	8%	8%	8%
Number of clients	2	8	10	20
Throughput each client (kbps)	25.6	25.6	25.5	25.4
Total throughput (kbps)	1041.6			

From the simulation results, it is found that when the original MCS algorithm is applied in the multi-rate application with the ARF incorporated, all the clients using the same channel have almost equal data throughput. Although some clients have the ability to support high data rates, they cannot get high data throughput. The inefficiency is caused by the polling policy of the MCS algorithm.

The data clients are prioritized in three priority levels (quiet list, data list, and hot list) for efficient bandwidth utilization, but the prioritization is only based on their activity, not their supported data rates. Thus, the clients in the same list are treated fairly by the hub and are polled equally in the sequence of the list. However, in the multi-rate application, with the equal polling rate, the clients with higher data rate take less time to finish their own packet exchange scenarios, but have to wait longer for the hub to poll those lower rate clients. Therefore, the final data throughput for all clients is dragged down toward that of the lowest rate clients. It is thus concluded that the original MCS algorithm is unsuitable for the multi-rate applications. To overcome the drawback, modifications must to be made to the original MCS data polling policy for better efficiency, with the consideration of the differentiation in data rate.

4.5 A Modified MCS Data Polling Method

In this section, a modified MCS data polling method is proposed and studied in the multi-rate application. While all the clients in the same list are polled equally in the original MCS, the higher rate clients receive more polling from the hub, so they are given more opportunities to exchange packets with the hub. After polling one client, the hub can choose to continue polling this client repeatedly, or turn to the next client, depending on the data rate of the client. In each polling a data token is issued to the client, followed by one MSDU, like in the original MCS. Therefore, those clients that are polled more times in each time slot can receive more data packets and get higher throughput. The detailed scheme is explained and simulated next.

4.5.1 Scheme Descriptions

Figure 4.9 shows the simplified data exchange timelines of the original MCS (left) and the modified scheme (right). It is assumed that Clients 1-4 in different ranges are downloading data files through the hub (with the detailed data exchange as in Data

Scenario 4 of Figure 3.4), using 11, 5.5, 2, and 1 Mbps, respectively. During the transmission four data users always stay in the hot list. A feature of the multi-rate exchanges in Figure 4.9 is the scenarios with variable width, from the thinnest one for 11 Mbps, to the thickest for 1 Mbps, since the transmission delay of the token (57 byte), data payload (584 byte), and the ACK frame (48 bytes) vary with different data rates. In the original MCS, the four clients are polled equally in turn, regardless of their data rates. The whole polling process is dragged down due to the long transmission delay when the low rate clients are being polled. Finally the data throughput can only reach a very low level for all the clients.

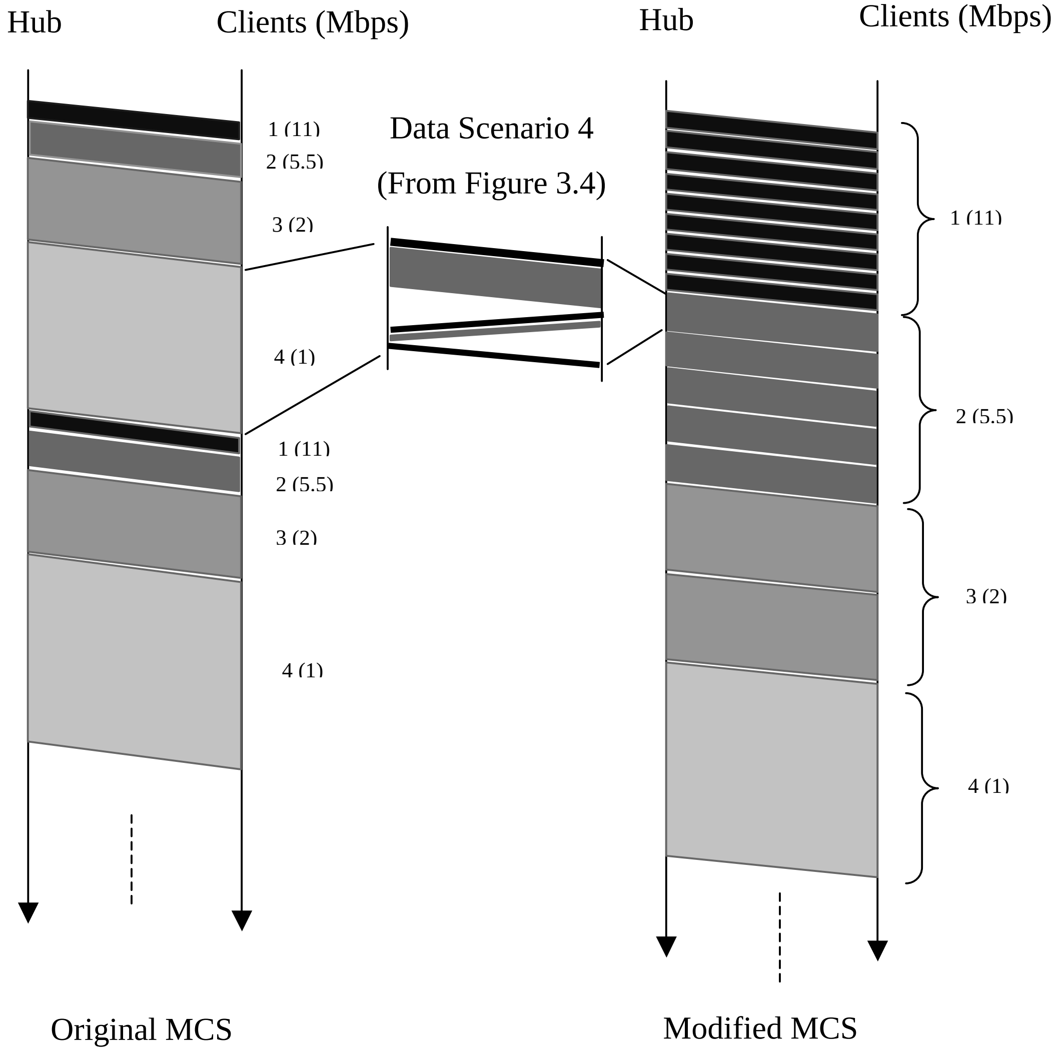


Figure 4.9 Modified data polling method of the MCS for multi-rate application

In the modified data polling scheme of the MCS, the hub is assumed to know the data rate of each client. After one polling cycle is over, the hub decides to poll the same client for more times or move to the next one, depending on the data rate of the client, whereas in the original MCS algorithm, the hub has no choice but to start polling the next client immediately. The number of repeated polling cycles for one client is decided by the data rate of this client. In Figure 4.9, Client 1 uses the highest data rate and receives 9 repeated polling cycles. The repeated polling cycles for Client 2, 3, and 4 are 5, 2, and 1 in this example, respectively. Only after the required polling cycles are finished can the hub start to poll the next client in the polling list. The total polling time for each client is basically equal. In this way, more polling tokens are sent to those clients with higher data rates and more data packets are delivered to them following the tokens. The repeated polling cycle is an adjustable variable in the following simulations.

It must be noted that sending voice tokens to a client in the voice list repeatedly is unnecessary, because only one voice packet is generated for one client in each 30 ms slot. Here the voice transmission is still as in the original MCS. How to improve the voice efficiency will be discussed in the following chapters.

In the original “MCSController”, a Client State Table is maintained to record the client activities. A new Client State Table is constructed for the hub to track and record the data rate of each client at all times in the modified data transmission scheme, by adding the “Access Rate”, shown in Table 4.7. The “MCSController” also has to be modified in the control of token issuing.

Table 4.7 A modified Client State Table

Index	Voice State	Data State	Last Activity	Access Rate
[1]	NULL	HOT_STATE	0.00833 s	11
[2]	NULL	HOT_STATE	0.00000 s	5.5
[3]	NULL	HOT_STATE	0.01234 s	2
[4]	NULL	HOT_STATE	0.00055 s	1
[5]	VOICE_STATE	QUIET_STATE	0.00000 s	11
:	:	:	:	:

4.5.2 Simulation Results

The “repeated polling cycles” in the simulations is used to indicate how many times the client is polled continually. Clients with higher data rates are given more repeated polling cycles, and have more chance to exchange packets with the hub than those with lower data rates. The “repeated polling cycles” for Groups 1-4 are expressed in an array, for example, [4, 3, 2, 1]. Table 4.8(a), (b), and (c) are tested by varying the values in the array of repeated polling cycles.

Table 4.8 Simulation results of the modified MCS algorithm

(a)

Group No.	Group 1	Group 2	Group 3	Group 4
Dist to center	56 m	69 m	85 m	105 m
Data rate (Mbps)	11	5.5	2	1
Repeated polling cycles	4	3	2	1
PER	8%	8%	8%	8%
Number of clients	10	10	10	10
Simulation Results of data throughput (kbps)				
For single client	91.4	68.7	46.0	22.9
For total clients	2242.0			

(b)

Group No.	Group 1	Group 2	Group 3	Group 4
Dist to center	56 m	69 m	85 m	105 m
Data rate (Mbps)	11	5.5	2	1
Repeated polling cycles	6	5	2	1
PER	8%	8%	8%	8%
Number of clients	10	10	10	10
Simulation Results of data throughput (kbps)				
For single client	113.0	94.4	38.0	19.0
For total clients	2595.4			

(c)

Group No.	Group 1	Group 2	Group 3	Group 4
Dist to center	56 m	69 m	85 m	105 m
Data rate (Mbps)	11	5.5	2	1
Repeated polling cycles	9	5	2	1
PER	8%	8%	8%	8%
Number of clients	10	10	10	10
Simulation Results of data throughput (kbps)				
For single client	154.4	86.4	34.7	17.5
For total clients	2877.7			

It is found the clients with more repeated polling cycles have higher data throughput. The data throughput of each client in Group 1 is the highest because they are polled the most times. Adjusting the repeated polling cycles can cause variation in the data throughput. With any of setting combinations tested, the clients with higher data rates always obtain larger data throughput using the proposed scheme. For example, if the repeated polling cycles for Groups 1-4 are set to [4, 3, 2, 1], the data throughputs of each client in Groups 1-4 are [91.4, 68.7, 46.0, 22.9] (in kbps) (listed in (a)). If the repeated polling cycles are changed to [6, 5, 2, 1], the data throughput of each client is also changed to [113.0, 94.4, 38.0, 19.0] (in kbps) for Group 1-4 (listed in (b)). With the setting of [9, 5, 2, 1] in (c), the throughput of each client, [154.4, 86.4, 34.7, 17.5] (in kbps), is very close to the data throughputs of each client in four separate channels of 11, 5.5, 2, and 1 Mbps, [156.0, 86.7, 34.1, 17.5] (in kbps) (the results of the reference case in Table 4.3).

The clients are distributed equally in four groups in Table 4.8. With the repeated polling cycles configured as [9, 5, 2, 1], if the numbers of clients in four groups are adjusted, as listed in Table 4.9, each client in the different group still receives the data throughput proportional to its data rate, consistent with Table 4.8(c).

Table 4.9 Simulation results in different distributions

(a)

Group No.	Group 1	Group 2	Group 3	Group 4
Dist. to center (m)	56	69	85	105
Data rate (Mbps)	11	5.5	2	1
PER	8%	8%	8%	8%
Number of clients	20	10	8	2
Throughput each client (kbps)	151.8	84.9	34.3	17.2
Total throughput (kbps)	4145.5			

(b)

Group No.	Group 1	Group 2	Group 3	Group 4
Dist. to center (m)	56	69	85	105
Data rate (Mbps)	11	5.5	2	1
PER	8%	8%	8%	8%
Number of clients	2	8	10	20
Throughput each client (kbps)	156.7	87.5	35.3	17.6
Total throughput (kbps)	1697.7			

Compared to the results in Table 4.6 simulated with the original MCS algorithm in the same distribution, it is obvious that the proposed modified scheme not only provides the reasonable throughput for each client, but also improves the total throughput significantly (4145.5 kbps vs. 2678.8 kbps for case (a) for example). It can be concluded that the modified scheme basically eliminates the effects of the low rate clients.

To simulate the operation of a more realistic system, the HTTP and FTP servers are used in the simulations to generate the Internet data traffic, instead of the CBR sources. It is assumed there are 60 clients in the rural area, about 15 km away from the MCS hub station. Four data rates are supported simultaneously in the common channel. All the clients are in the active state and the clients using the same data rate belong to the same group. Each group has one FTP client and the other users are doing web browsing. During the data exchange, 16 clients have set up voice calls through the CM with the PSTN users.

The original MCS and the modified scheme proposed above are simulated. The variations of throughput of four FTP clients and the aggregate throughput for the remainders (the total HTTP clients in the same channel) are graphed versus time (0-500 s) (Figure 4.10, 4.12). One of the HTTP clients in each group is taken as an example to show the variation of the data throughput of HTTP clients with time (Figure 4.11, 4.13). In Figure 4.10-4.13, Client 1 - 4 are the FTP users of Group 1- 4 with the data rates of 11, 5.5, 2, and 1 Mbps (indicated by FTP1 – 4). Client 5 - 8 represent one of HTTP users in Group 1-4, working at 11, 5.5, 2, and 1 Mbps (indicated by HTTP5 – 8), respectively.

The FTP clients always stay in the hot list, and the HTTP clients may switch their states dynamically among the hot list, data list, and quiet list with time. During the quiet period, clients do not respond to any polling. If the original MCS algorithm is used, the mean data throughput of each FTP client is almost equal, about 43.8 kbps (shown in Figure 4.10). The same situation faces the HTTP clients (each one gets about 13.7 kbps in Figure 4.11). It is pretty low for the clients working with high data rate, like 11 and 5.5 Mbps.

Figure 4.12 and Figure 4.13 show the simulation results of the proposed scheme with the repeated polling cycles [9, 5, 2, 1] for Group 1-4. The clients with higher data rate get higher throughput and the aggregate throughput for the remaining HTTP clients is also improved. The mean throughput for an FTP client with 11 Mbps is increased to 133.4 kbps, almost ten times of the client with 1 Mbps. From Figure 4.13, the higher rate

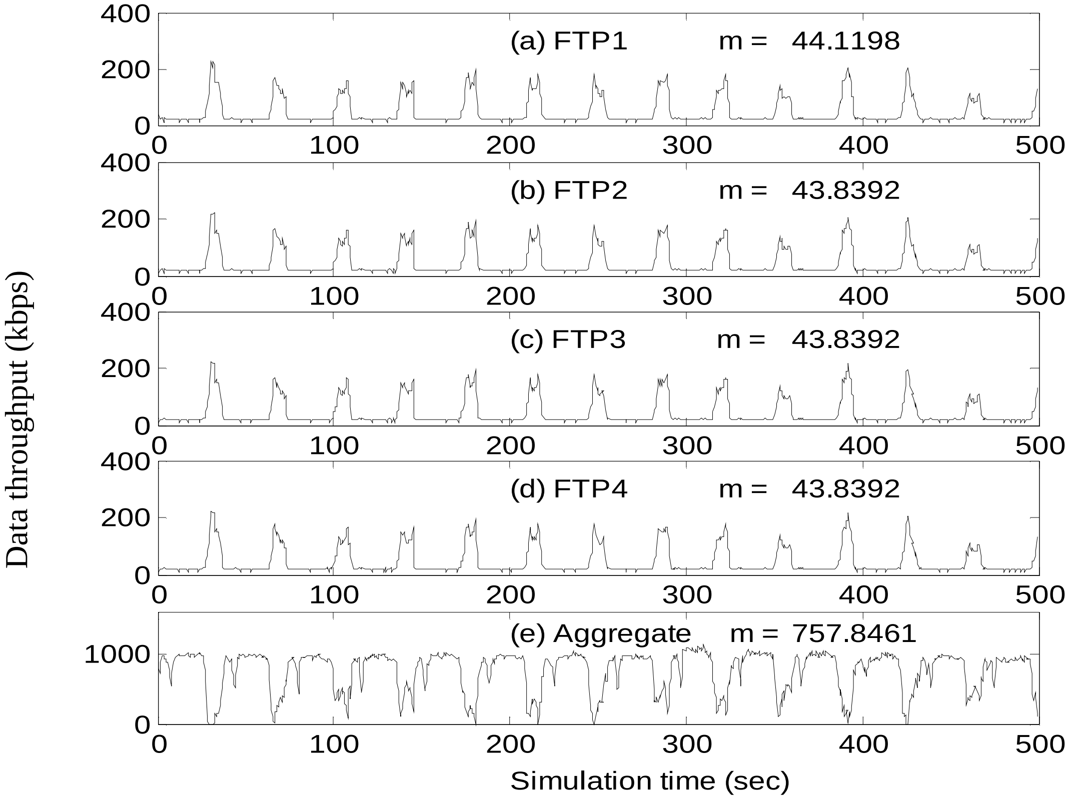


Figure 4.10 Data throughput of FTP users in original MCS

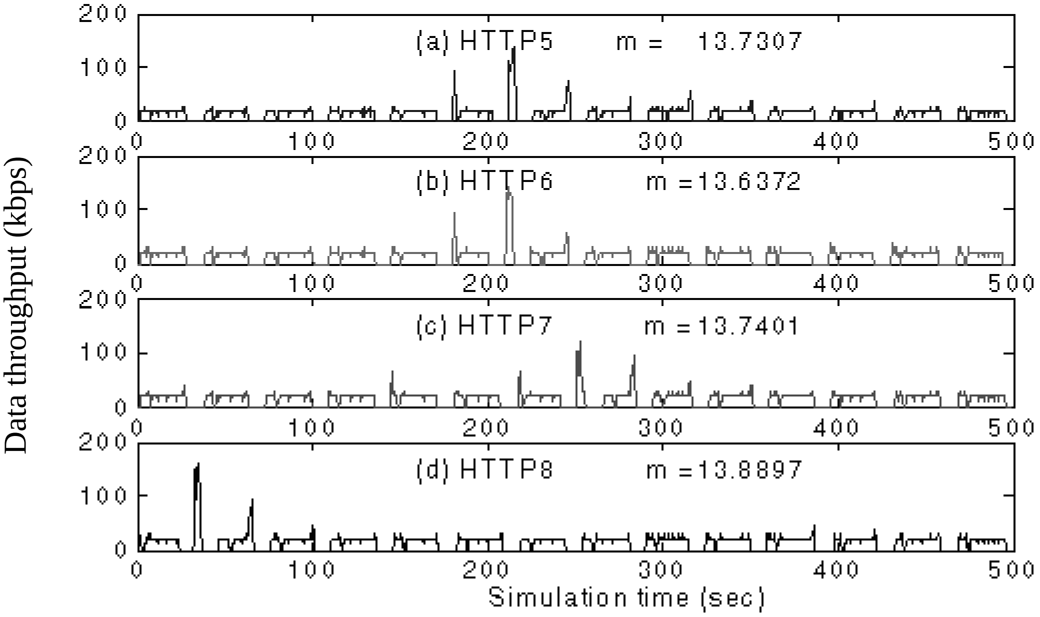


Figure 4.11 Data throughput of HTTP users in original MCS

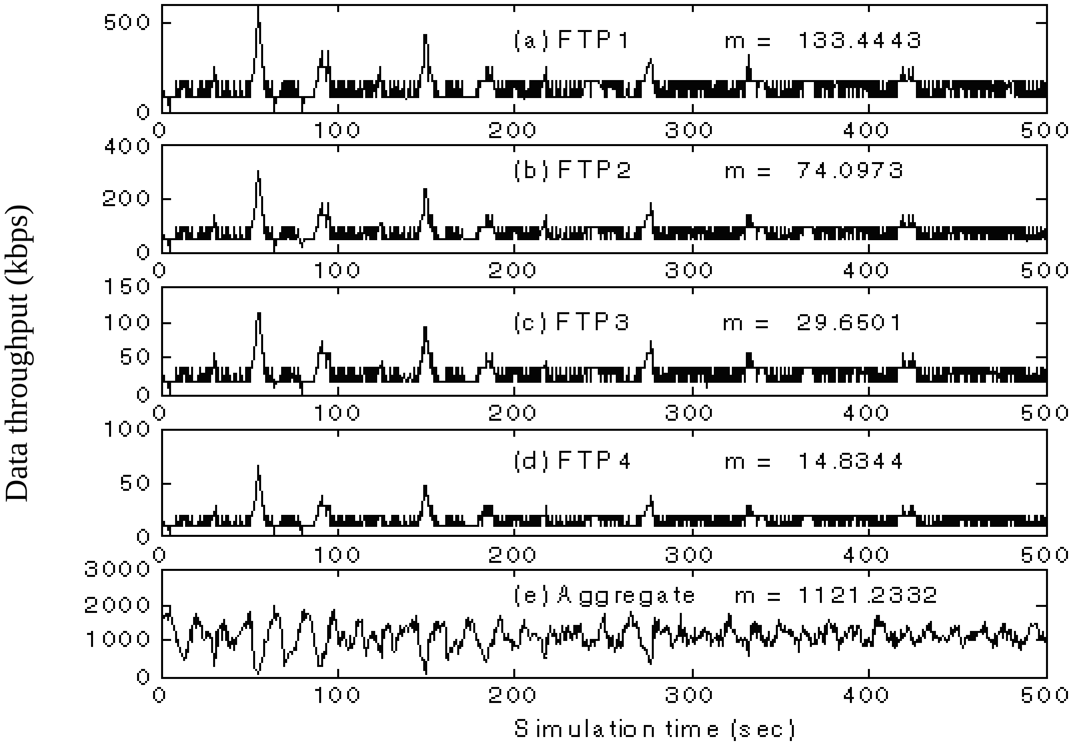


Figure 4.12 Data throughput of FTP users in modified MCS

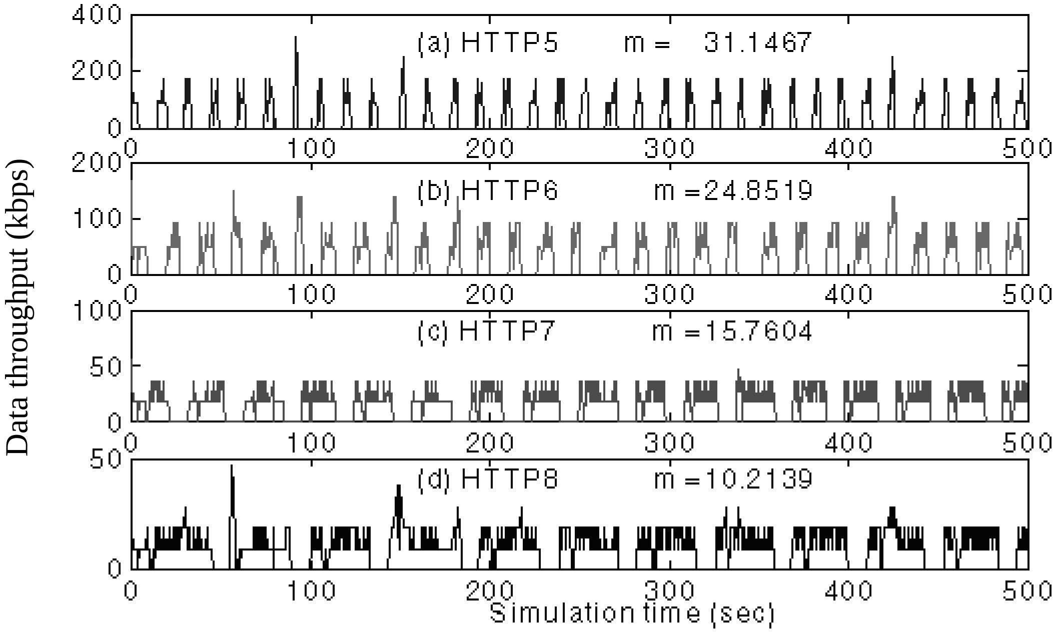


Figure 4.13 Data throughput of HTTP users in modified MCS

HTTP clients take less time to view the same content and achieve higher throughput than those with lower rate.

It is shown from the simulation results that the proposed scheme can basically overcome the drawback of the original MCS in handling the multi-rate application and can be considered as a feasible solution for the multi-rate data transmission in the proposed rural network over IEEE 802.11b.

4.6 Summary

The ability of the original MCS algorithm in supporting data transmission in the proposed rural network was examined in constant rate radio channels. The MCS saturation throughput simulated in the WLAN application was comparable to the DCF operation, but it was degraded with relay distance or the number of voice clients. The efficiency, 72.0%, 71.0%, 67.4%, or 62.3% was achieved at the MAC layer with the data rates 1, 2, 5.5, or 11 Mbps, at the relay distance of 15 km, very close to the theoretical results.

The ARF feature of IEEE 802.11b was incorporated into the MCS data simulations, which is able to improve the data throughput of an MCS system at bad channel conditions. However, when clients in the same channel worked at different data rates at the same time, the MCS algorithm was found to provide almost equal and low throughput for each data client involved. It was concluded that the original MCS algorithm is not suitable for the multi-rate data transmission.

The data polling method of the MCS was modified to give the higher rate clients more opportunities to be polled, so they could get higher data throughput. With appropriate setting of the repeated polling cycles for the clients according to their data rates, simulation results showed that the problem hindering the original MCS scheme in the multi-rate application is basically overcome. The new data polling method of the MCS algorithm can be used to support the data transmission of the proposed system over IEEE 802.11b.

The performance of voice transmission using the original MCS algorithm will be examined in the next chapter.

CHAPTER 5
VOICE PERFORMANCE OF MCS POLLING ALGORITHM

The primary objective of this chapter is to thoroughly test the MCS algorithm's ability to prioritize voice packets in the network proposed for rural areas in Figure 4.1. The voice polling part of the original MCS algorithm is modified to be more efficient. End-to-end delay and channel capacity for voice are measured under various conditions using the improved scheme. Finally, the MCS' inefficiency in voice transmission is analyzed and possible solutions are proposed for further discussion.

5.1 Voice Traffic Pattern in NS-2

The voice traffic has the characteristic of an ON/OFF process, and the voice users are either transmitting (ON) or listening (OFF). The amount of time for a voice user in the talk period or in silence follows the exponential distribution with the mean value of 1.0 s or 1.35 s, according to the statistical results [9][11]. In simulation, a source with the exponential ON/OFF distribution is used to generate the voice traffic stream. It sends the Constant Bit Rate (CBR) packets during ON period and stops sending during OFF period. The average burst time (ON) and average idle time (OFF) are set to 1.0 s and 1.35 s respectively.

UDP is used as the transport layer protocol to minimize the end-to-end delay. There is no positive acknowledgment (ACK) or negative acknowledgment (NAK) generated after the reception of a packet in the transport layer. Besides, UDP is more efficient to transport small packets because of its smaller overhead (8 bytes) than TCP (20 bytes overhead). However, a reliable connection with definite ACK or NAK is required to set up or tear down a voice call, so TCP is used for the requests of the call setup and teardown in the simulation.

The commonly used G.729 codec, providing the toll-quality 8 kbps voice for wireless applications, is modeled in the simulation. A 30 byte payload is generated at the application layer every 30 ms (one MCS time slot), and is passed down to the transport layer. Compressed Real Time Protocol (CRTP) compresses the Real Time Protocol (RTP), UDP, and IP headers to 5 bytes. An 8 byte header is added at the LLC layer (3 byte LLC and 5 byte SNAP). The total 43 byte is sent to the MAC layer as a voice payload. A 57 byte overhead that combines the MCS algorithm and IEEE 802.11b is attached to it. The 57 byte overhead is simulated using a separate packet (the token).

5.2 Estimation of the End-to-End Delay

Table 5.1 Budget of total delay for IP Voice

Positions	**Description of delay**		**Delay (ms)**
Clients	G.729 encoding delay (three 10 ms frames + 5 ms look-ahead)		35
	Packetization delay – included in coding delay		
MCS channel (from the hub to clients)	MCS' polling network access		<=30
	Voice packet serialization delay (at 1 Mbps)		0.8
	Propagation	Inside a pico cell (100 m)	0.00033
		From frequency translator to relay station (1000 m) *	0.0033
		Relay transmission (15 km)	0.05
CM	Jitter Buffer (matched to the MCS polling delay of 30 ms)		0
	Voice Decompression G.729		10
	Digital Switch		1.6
PSTN	Long distance PSTN propagation delay		20
		Total =	97.5

* The group delay of the Band Pass Filter (BPF) inside a frequency translator was tested to be about 70 ns for the Chebeshev BPF (filter order N=3) using ADS. Assuming the frequency translator has two BPFs in one direction, the BPF group delay is much less than 1 ms. In the simulation, its effect on the overall delay is neglected.

The G.729 codec compresses the voice to 8 kbps, but introduces the encoding delay (35 ms) (packetization delay is included). One voice packet is generated every 30 ms for one active IP phone and is sent out when it is polled by the MCS hub. If the MCS polling is completely out of sync with the IP voice packet generation, an additional polling delay of up to 30 ms may result.

It is assumed that in simulations all the phone calls are set up between the pairs of a remote MCS client and a PSTN user. This makes the prediction to the total end-to-end delay possible across a known network. If an Internet IP phone is used, instead of a PSTN user, variable delays and different paths associated with the unknown Internet make it difficult predicting voice quality, and therefore guaranteeing it, which is beyond our discussion. The estimated end-to-end voice delay in Table 5.1, 97.5 ms, is acceptable for most user applications under the ITU standard G.114. It is also possible for two remote clients to set up a call via the MCS hub station. Here we only consider the delay related to the MCS polling scheme, including the MCS polling delay, the serialization delay (or transmission delay) and the propagation delay (the shaded area of Table 5.1). For the distance from 15 km to 50 km, the propagation delay ranges from 0.05 ms to 0.167 ms (0.05% -0.17% of the total delay).

5.3 Modification to the Original MCS Voice Polling

In the original MCS algorithm, all the received packets are required to be acknowledged for the reliable transmission in the wireless medium. However, due to the low latency requirements, the discarded voice packets are never retransmitted. It is a waste of bandwidth to acknowledge the voice packet in the original MCS.

Figure 5.1 illustrates a shortened MCS scenario for voice transmission, which sends the token only twice in any case. After the voice token is returned from the client, the field “ret” of ping packet (see Section 3.3, it performs the function of the token in simulation) is increased from 1 to 2 in the MCS link before arriving at the hub, no matter if the client has a voice packet to send or not. When the hub receives the voice token with the “ret” equal to 2, it frees the token immediately and starts polling the next client in the voice list. In this way, even if the client has a voice packet for the hub, for example, in Voice Scenario 2 and 4 of Figure 3.4, no ACK packet is generated and sent back to the client and the time line for the scenario is shortened.

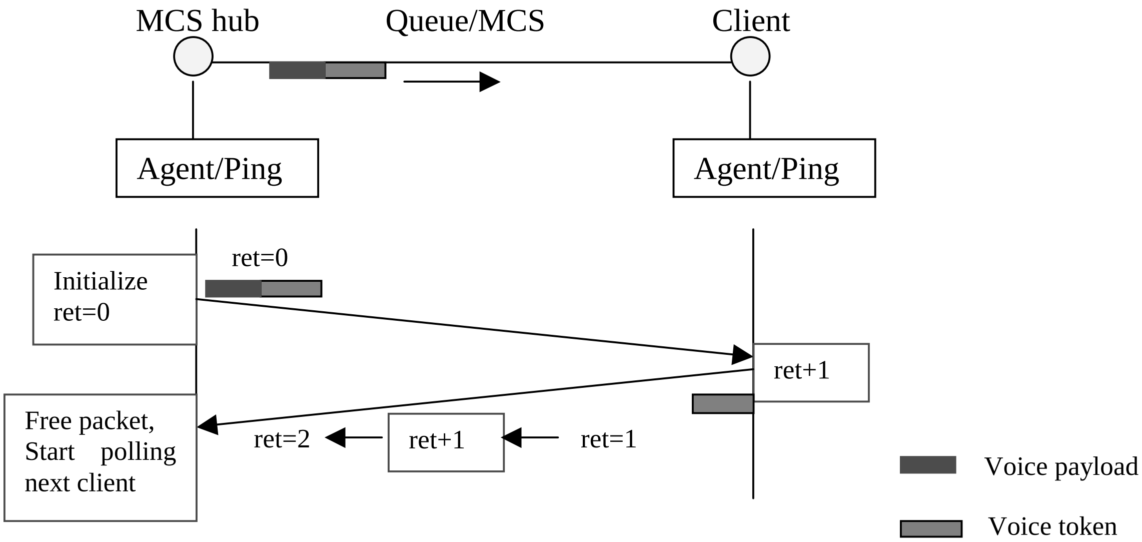


Figure 5.1 Improved MCS polling process for voice transmission

Seen from the MAC layer, the 57 byte voice token leads to system inefficiency. For example, the voice efficiency for the 4th MCS voice scenario in Figure 3.4 is given by Equation (5-1) in a 2 Mbps channel.

$$\eta = \frac{2\times\delta_{trans}(43B)}{2\times\delta_{trans}(43B)+2\times\delta_{prop}(15km)+3\times\delta_{trans}(57B)} = 30.5\% \qquad (5\text{-}1)$$

With the above modification, the efficiency is now improved to:

$$\eta = \frac{2\times\delta_{trans}(43B)}{2\times\delta_{trans}(43B)+2\times\delta_{prop}(15km)+2\times\delta_{trans}(57B)} = 38.2\% \qquad (5\text{-}2)$$

5.4 Simulation Results

5.4.1 Improvement in Data Throughput

In addition to the voice efficiency, the data throughput in the same channel can also be improved with the modification shown in Figure 5.1, because the improved MCS voice polling can effectively shorten the voice portion in each time slot, and leave more time to exchange data packets.

Using the improved MCS voice polling scheme, data throughput is tested as the offered load is increased gradually, at the exactly same channel conditions as Figure 4.3 (There are 10 data clients in an 11 Mbps channel, the relay distance is 15 km, and 0, 20, and 40 voice clients are included, respectively).

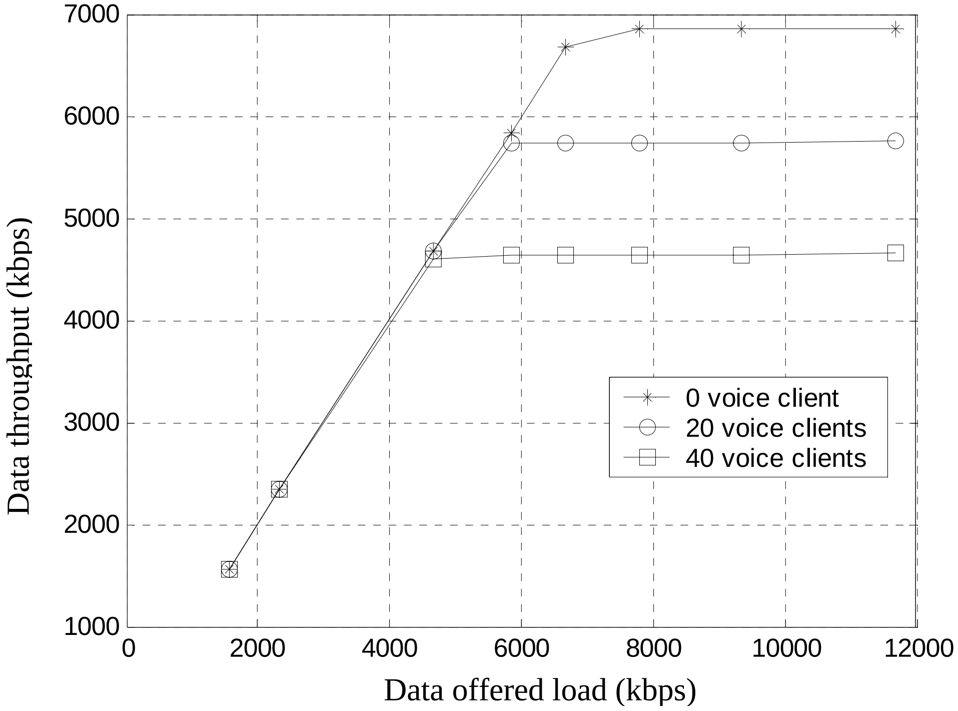


Figure 5.2 Data throughput using the improved MCS voice polling scheme

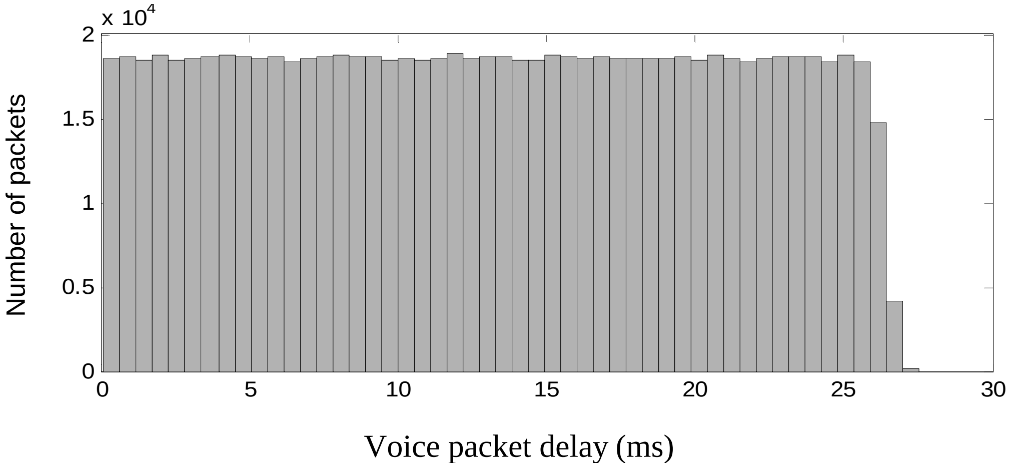
Like in Figure 4.3, the saturation throughput is inversely proportional to the number of active voice clients. The more voice clients set up calls in the channel, the lower data throughput is resulted. However, with the same number of voice clients involved, the saturation throughput of Figure 5.2 is higher than Figure 4.3. For example, with 20 voice clients, the saturation throughput is increased from 5496.2 kbps to 5749.9 kbps after the MCS voice polling scheme is improved. More improvement on the data throughput can be achieved when there are more clients in the voice list. The following simulations in this chapter are conducted using the improved MCS voice polling scheme.

5.4.2 Distribution of Voice Delay

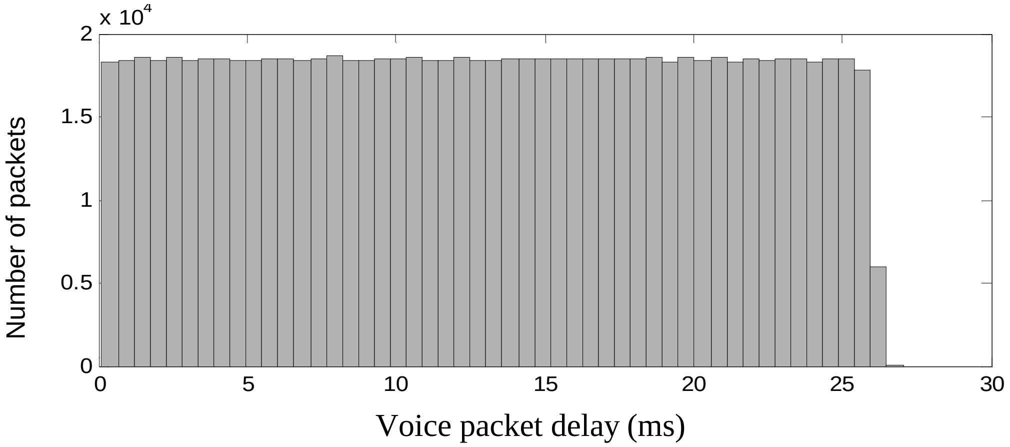
The delay discussed here is only a portion of the total end-to-end delay from a speaker to a listener, which is introduced between the CM and the clients, and the wireless MCS channel is a major part. During the simulation, the departure time (t_1) for every voice packet from the CM and the arrival time (t_2) at the destination client are recorded. The delay of every voice packet from the CM to any client can be calculated by subtracting t_1 from t_2, and vice versa.

One voice MSDU packet (43 bytes) is generated randomly at any time if the voice user is in the talk state. The function handling voice packet exchange is called every fixed interval in the "MCSController". The distribution of voice delay is basically uniform in the polling interval.

Figure 5.3 shows the voice delay distribution in an 11 Mbps channel, indicating the number of received voice packets with the delay equal to a given value in X-axis. The distance from the hub to the relay station is set to 15 km and the simulation time is set to 500 ms (The longer the simulation time, the result is more close to the real situation.). The polling interval for the voice processing function in the "MCSController" is set to 26 ms in simulation. There are 4 FTP users, 35 HTTP users, and 82 voice users (some clients support both data and voice at the same time) in the simulation of Figure 5.3(a). The distribution function of voice delay in a "pure" 11 Mbps voice channel without data traffic is shown in Figure 5.3(b).



(a) In a channel with data and voice integrated



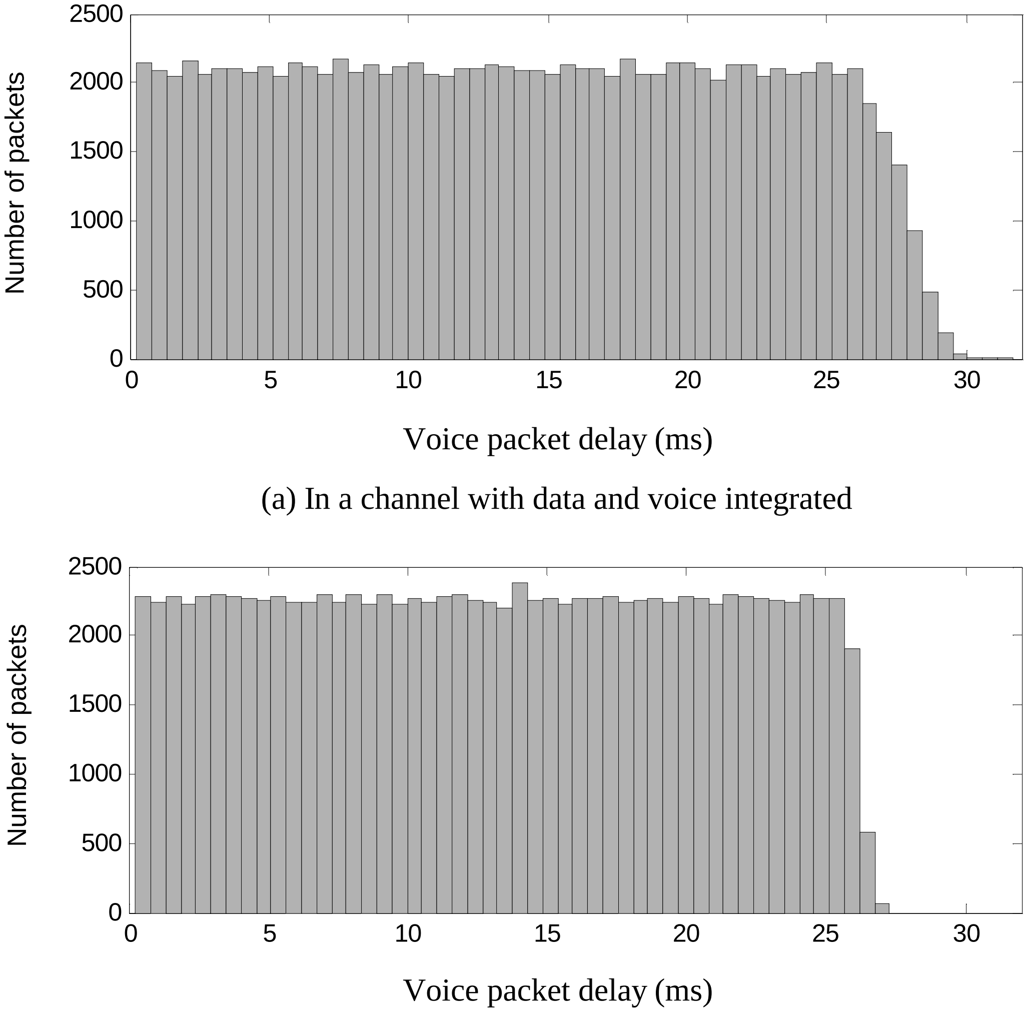
(b) In a channel without data traffic

Figure 5.3 Distribution of voice delay in an 11 Mbps channel

Both of them basically follow the uniform distribution in [0, 26 ms], but the maximum delay in (b) is slightly smaller and it is more close to the ideal uniform distribution.

The difference is more obvious in a 2 Mbps channel, as shown in Figure 5.4. There are 4 FTP users, 35 HTTP users and 10 voice users in Figure 5.4(a) and only 10 voice users in Figure 5.4(b). The other parameters are kept unchanged. The maximum delay in Figure 5.4(a) is about 30 ms and the edge is not as sharp as (b). Obviously, the latter is more close to the ideal curve.

In an integrated data and voice channel, polling data clients in the remaining slot time has a negative impact on the voice delay performance. The voice polling portion (Tv) was designed to start periodically (the interval is set fixed), but sometime is deferred because of a data polling scenario that has not finished yet. Only when the entire data exchange scenario is over can the control of the MCS channel be returned to the hub to start the voice transmission. Therefore the data traffic can cause some extra delay for the voice transmission, which is larger in a lower rate channel due to the longer data exchange scenario. This is the reason why the voice polling interval is set a little smaller than 30 ms, if the maximum polling delay of 30 ms is required.



(a) In a channel with data and voice integrated

(b) In a channel without data traffic

Figure 5.4 Distribution of voice delay in a 2 Mbps channel

5.4.3 Effect of Relay Distance on Voice Delay

The Complementary Cumulative Distribution (CCD) is the probability of a random variable x greater than a given value X, expressed in P{x>X}. For the variable, "voice delay", P{x>X} is the ratio of the number of packets with delay greater than X to the total packets received. It is the objective to achieve P{x>X} as small as possible for a given delay X. In several probability curves, the lowest one has the best delay performance.

A pure 11Mbps voice channel without data traffic is simulated with 10 voice clients. Three CCD curves are shown in Figure 5.5, corresponding to the relay distance of

1 km, 15 km, and 30 km. The delay performance becomes worse slightly as the relay distance increases because of the negative effects of propagation delay, which is a small part of the total delay. The effect of propagation delay can be shown more clearly from the variation of the voice capacity of a channel, which is described below.

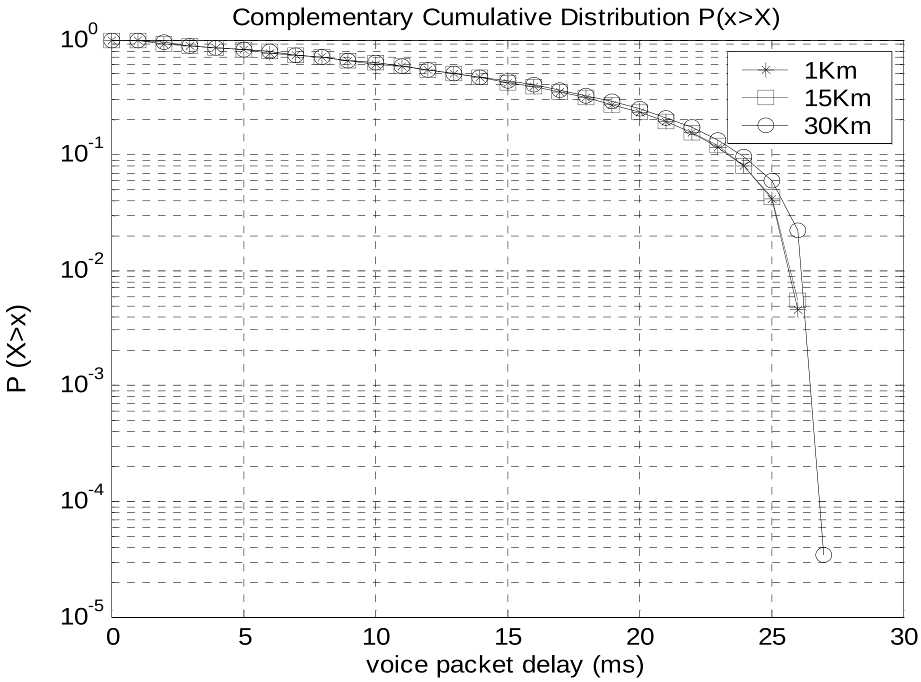


Figure 5.5 CCD curves of voice delay at different distances

5.4.4 Packet Loss Consideration

The Packet Loss Rate (PLR) is one of the metrics of voice quality. For quality voice service, the PLR less than 1% is acceptable [13].

Two factors can cause the voice packet loss. The first one is the overflow of the voice queue. When more and more voice clients have set up their calls, the hub takes more time to get through the whole voice list. The voice packets cannot be sent out immediately and have to wait in the voice queue. When the voice queue is full, the newcomers have to be dropped (under another rule, the voice queue accepts the new one, but drops the oldest). The second reason for the packet loss is the bit errors in a realistic channel, where the corrupted voice packets are discarded right away after being detected without performing the retransmission.

With voice packet size of 100 bytes (43 byte payload plus 57 byte header), a maximum PLR of 0.8% can be expected with radio channel BERs of up to 10^{-5}. For a radio channel in good conditions, the PLR caused by the BER is pretty low and is ignored in simulation. When more and more clients set up their calls, the overflow of voice queue occurs. “Voice capacity” is defined as the maximum number of calls that can be supported in one channel simultaneously, while the PLR is kept below the threshold. In our simulation, the number of voice clients is increased gradually. When it approaches the voice capacity, the dropping of voice packets begins. Although the quality voice traffic can tolerate some packet loss, for simplicity, the voice capacity is regarded as the number of voice clients when the packet dropping starts, or, the minimum number of voice clients that causes the packet dropping.

5.4.5 Voice Capacity under Various Conditions

Table 5.2 Simulation results of voice capacity

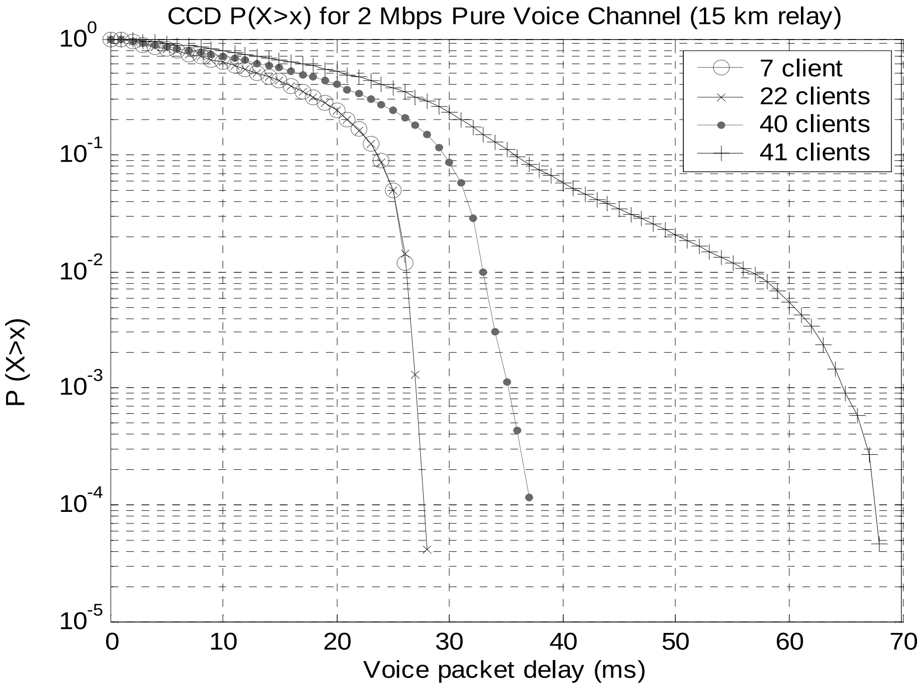
Simulation conditions			**Packet dropping starts @ the number of voice clients**
(a)	**Channel rate**	1 Mbps	22
		2 Mbps	41
		5.5 Mbps	94
		11 Mbps	148
(b)	**Relay distance**	1 km	47
		5 km	45
		15 km	41
		20 km	40
		25 km	38
		30 km	37
		50 km	32
(c)	**Data traffic (FTP + HTTP data clients)**	0 + 0	41
		1 + 0	37
		4 + 0	37
		4 + 5	37
		4 + 10	37
		4 + 30	36

Table 5.2(a) is the voice capacities tested at four channel rates without data traffic, at the relay distance of 15 km. The voice capacity increases with the channel rate. In a realistic network, the performance to support multi-rate voice transmission in a

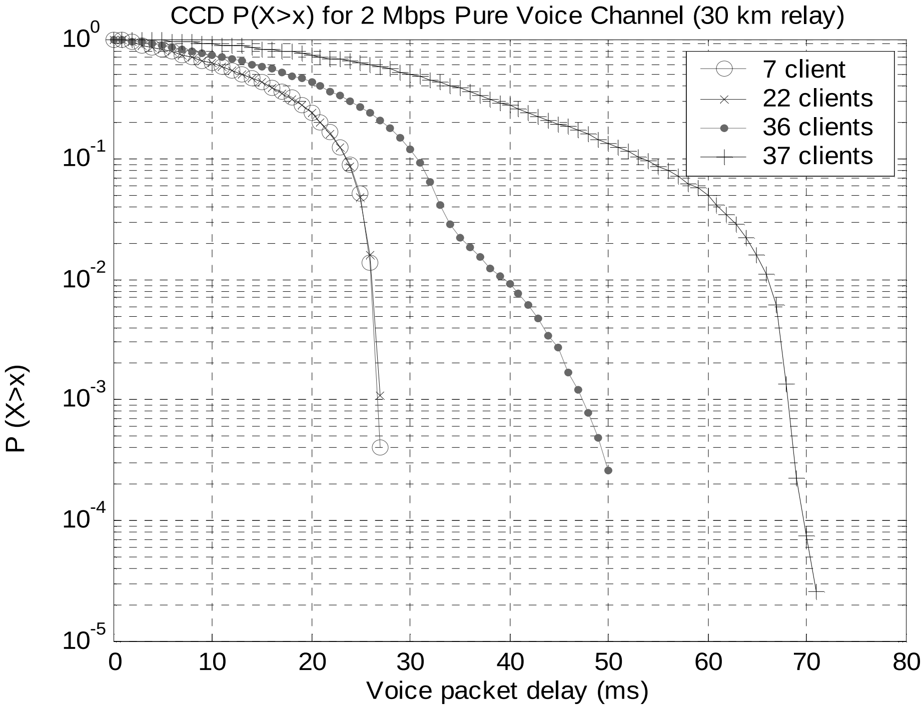
single channel will be discussed later. Table 5.2(b) and (c) are simulated taking a 2 Mbps channel as an example (the other channel rates can result in the same conclusion). The voice capacity is negatively affected by the propagation distance. In (b) as the relay distance is increased from 1 km to 50 km, the voice capacity of a 2 Mbps channel declines from 47 to 32.

Data traffic is included in a 2 Mbps channel (Table 5.2(c)). The joining of data traffic negatively influences the voice capacity. It changes from 41 to 37 at first, but changes little in spite of the continual growth of data traffic. This verifies the MCS ability of prioritizing the voice packets. In each time slot, only after all the clients in the voice list are polled, the exchange of data packets and system messages starts. The data traffic may defer the start of voice polling, but it does not do much in deteriorating the voice performance.

Figure 5.6 shows what happen to the CCD curves when more and more calls are set up to approach the capacity. The dropping of voice packets begins when the number of voice clients reaches 41 and 37, if the relay distance is set to 15 km and 30 km in a 2 Mbps pure voice channel. Obvious change of the CCD curves occurs when the packet dropping starts, showing that the delay performance becomes worse accompanying to the start of packet dropping.



(a) Relay distance is 15 km



(b) Relay distance is 30 km

Figure 5.6 CCD curves of voice delay with different number of voice clients

5.4.6 Multi-Rate Application of Voice Transmission

In the multi-rate mode, like the data transmission, the entire voice polling process is slowed down by those low rate clients. Since only one voice packet is generated in each time slot for each voice client, the throughput of voice packets is not our concern in the multi-rate application, but the voice capacity of a channel can be affected by the speed of polling process, which is related to the rate distribution of voice clients.

Table 5.3 Channel voice capacity in different distribution of voice clients

Relay distance	**Distribution A-B-C-D**	**Number of the total voice clients when the packet dropping starts**
15 km	25%-25%-25%-25%	48
	40%-30%-20%-10%	62
	10%-20%-30%-40%	34

Table 5.3 lists the simulation results in three distributions. The ratios of voice clients with 11, 5.5, 2, and 1 Mbps to the total voice clients in a channel (A-B-C-D) have

three different distributions. If the ratio of high rate clients is raised, more voice calls can be supported in the channel without packet dropping.

With shorter Tv in an integrated voice and data channel, more time can be allocated to the data portion (Td), and larger data throughput can be achieved. Assuming 20 voice clients have set up calls, the average length of Tv varies with the rate distribution.

-- 4.35 ms (All 20 clients use 11 Mbps.)

-- 28.17 ms (All 20 clients use 1 Mbps.)

-- 9.69 ms (8 clients use 11 Mbps, 6 clients use 5.5 Mbps, 4 clients use 2 Mbps, and 2 clients use 1 Mbps.)

-- 18.04 ms (2 clients use 11 Mbps, 4 clients use 5.5 Mbps, 6 clients use 2 Mbps, and 8 clients use 1 Mbps.)

5.5 Discussion to the Voice Inefficiency

The voice efficiency has been improved by not acknowledging the voice packets from clients, but it is still not satisfactory. Now the average efficiency in a 1, 2, 5.5, or 11 Mbps channel is 22.8%, 21.4%, 17.5%, or 13.7%, respectively, at the distance of 15 km (Table B.1, APPENDIX B). The reasons for the low efficiency of voice transmission can be attributed to two factors.

The first is the polling policy of the MCS algorithm. All the clients in the voice list are polled by the hub individually. Only after one entire polling cycle is over can the hub turn to the next one. For each polling, voice packets are exchanged between the hub and one client. Before receiving the response from the other side, the hub or the client cannot do anything but wait. Since the voice transmission does not require acknowledging the reception of received packets, it is a waste of time for the hub to poll each client individually and wait for the response. The time wasted is more severe at the longer propagation distance from the hub to clients.

The poor ratio of the voice payload (43 bytes) to the overhead (57 bytes) at the MAC layer also results in the voice inefficiency. Although the MCS header has been shortened and the short preamble of IEEE 802.11b is used, the 49 byte overhead of IEEE 802.11b is still too much for the payload part, but there is no way to shorten it currently. What we can do is to increase the size of voice payload.

The size of voice payload (43 bytes) is decided primarily by the length of the time slot (30 ms in the MCS algorithm), which also decides the maximum time delay introduced by the MCS polling access. If the time slot is prolonged to 60 ms, the size of voice payload is almost doubled (73 bytes), but an extra 30 ms will be added to the total end-to-end delay in Table 5.1. It is a feasible modification, since the total delay (about 150 ms) is still acceptable.

5.6 Summary

The MCS' ability of prioritizing voice packets was examined in this chapter. At the beginning, an improved voice polling scheme was proposed, which is more efficient in the voice polling, and can provide higher throughput for the data traffic in the same channel.

From the simulation results using the improved voice polling scheme, the delay of voice packets between the CM and the clients basically follows the uniform distribution. The maximum delay depends on the interval of voice polling. The coexistence of data traffic in same channel also affects the distribution of voice delay. The voice capacity was tested under different channel conditions. More voice clients can be supported simultaneously in the channel without any packet dropping, if higher channel rate is used and/or the relay distance is decreased. Integrating data traffic also has a negative impact on the voice capacity, but the voice capacity varies little with the continual growth of data load, which shows the highest priority of voice packets can be guaranteed in the MCS. The multi-rate application for the voice transmission was also discussed.

CHAPTER 6

TDMA-LIKE VOICE TRANSMISSION SCHEME

A TDMA-like voice transmission scheme is described and implemented in this chapter. A possible format of the beacon frame specific for the TDMA-like scheme is proposed. The voice efficiency is calculated and compared with the MCS polling algorithm under various channel conditions, including varying the payload size. The new voice handling process is programmed in C++ code as a part of the “MCSController”, and modifications are made to the other simulation modules to implement the new scheme inside NS-2. The voice performance and data throughput in various channel conditions are simulated using the new scheme and compared to the previous results. How to support the multi-rate voice transmission in the new scheme is also discussed.

6.1 Scheme Descriptions

Collisions can be avoided in the MCS polling algorithm, but the individual polling of each voice client places an excessive burden on the MCS resources, largely due to the propagation delay [3], and results in low efficiency. The propagation delay is added twice in Equation (5-1) and (5-2). Its ratio in the total time becomes larger and its negative effect is more noticeable with increasing propagation distance. The area most in need of improvement is the way the MCS algorithm issues voice tokens.

A more efficient way for voice service is to send a number of tokens sequentially for one polling, so the time wasted in waiting for replies can be minimized. A new scheme similar to the TDMA mode, called TDMA-like scheme, is developed from a possible solution to the MCS voice inefficiency first proposed in [3]. Figure 6.1 describes the exchange of voice packets in the Tv portion of a 30 ms slot. It is assumed in a steady state with 15 VoIP devices on the voice list.

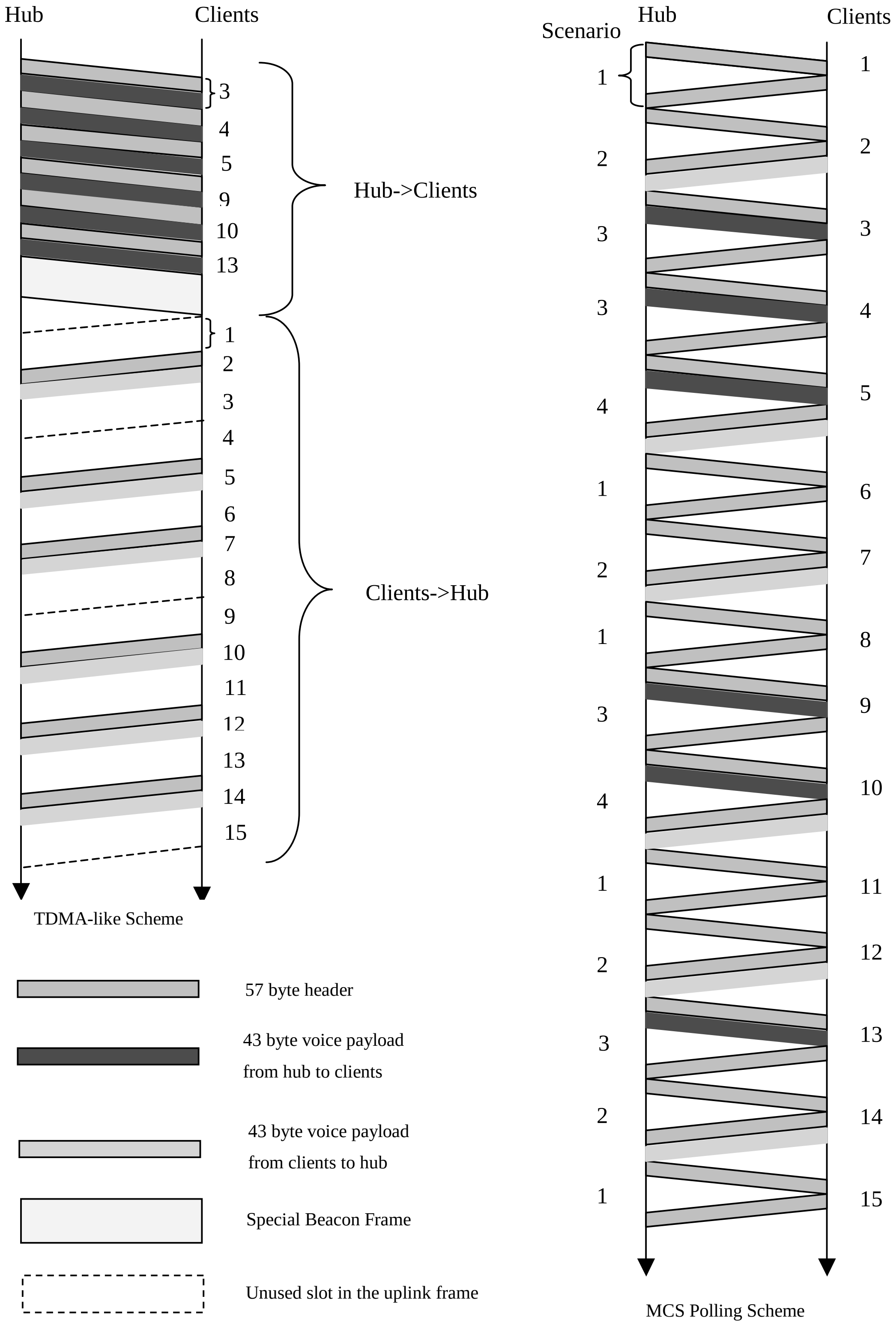


Figure 6.1 Comparison of TDMA-like Scheme and MCS Polling Algorithm

The timeline on the right represents the improved MCS polling algorithm. Although the last ACKs have been deleted, there is still considerable wasted space in the timeline diagram, since each client is individually polled and extra propagation delays are added.

The new TDMA-like voice transmission scheme is shown on the left. The entire voice exchange process comprises two parts, downlink transmission and uplink transmission. The downlink transmission is initiated by the hub. Since the hub has knowledge of which clients are due to receive a voice packet, it would continually transmit these packets. After that a special beacon frame is issued by the hub. The purpose of the beacon is to synchronize the clients on the voice list and provide information as to the upcoming client transmit order.

In the following uplink transmission, voice clients in turn transmit a voice packet in a predetermined slot providing they have a voice packet to transmit. The minimum width of time slot is equal to the transmission time of putting an entire voice packet (the header is included) into the outgoing link. It is assumed that there is no way for the hub to know ahead of time if a voice slot would be unused in the uplink direction, so each client in the voice list is assigned a slot, no matter if it is in talk period or silence. Those voice clients in silence do not transmit in their assigned slots and leave some slots unused in the uplink frame.

In addition, a small “guard time” is reserved between any two consecutive time slots in the uplink transmission (not shown in Figure 6.1) to prevent the possible collisions caused by inaccurate synchronization or distance deviation. No additional guard time is needed in the downlink, because all the packets are sent from the hub, and no collision occurs. The longer the guard time, the possibility of collisions is lower, but it can cause the dropping of voice transmission efficiency. If the distance from clients to the hub station varies in a large range (for example, some clients are located about 15 km away, while some others are only 1 km only), some distant clients would receive the beacon frame later than the others. A big enough guard time has to be set in such situation, which might negate the advantages of the TDMA-like scheme. However, we can avoid the problem by adjusting the channel allocation plan according to the geographical distribution of clients, for example, arranging the clients with big differences in the distance to the hub in separate channels.

The situation can be further improved by letting the hub know the geographical location of each client station. It is feasible for our proposed network, since all the clients involved are fixed. Therefore, the arrival time of beacon frame at each client can be calculated and known by the hub all the time. The uplink transmission order and schedule can be adjusted correspondingly, for example, the closer clients are allowed to transmit earlier, the slots at the behind of uplink frame can be assigned to those distant clients, and the guard time can be set flexible to avoid overlaps in the uplink transmission.

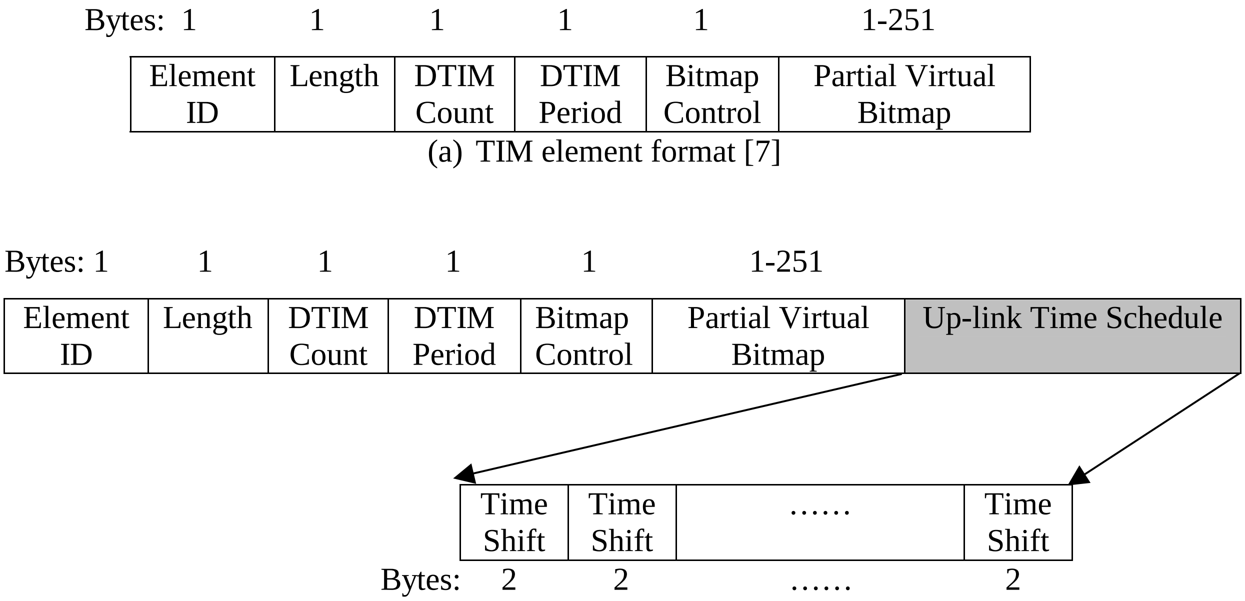
The problem caused by big distance deviation in one channel can be basically resolved using the above methods. However, a small guard time is still needed to tolerate some inaccuracy. It is assumed in the following discussion, all the clients are connected to the hub through one relay station and have a guard time of 1 μs. The results presented are likely valid for the guard times in the range of 1–5 μs without significant performance degradation, which correspond to the distance deviation from 300 m to 1500 m occurring in micro cell and pico cell.

In the TDMA-like scheme, the requests of call setup and teardown is handled during the period (Td) of polling data clients, that is, in the same way the original MCS algorithm does.

6.2 A Format of the Beacon Frame in the TDMA-like Scheme

The beacon frame is significant to the operation of the TDMA-like scheme. However, there is currently no frame available to perform such functions either in the MCS prototype or in the IEEE 802.11. Although the frame design is not the task of this thesis, a possible frame format is proposed here for simulation and theoretical calculation [13].

In the Beacon frame of IEEE 802.11 periodically generated by the AP, the Traffic Indication Map (TIM) element is the key portion of its Frame Body, illustrated in Figure 6.2(a). The detailed explanations for each field are given in [7] and only the "Partial Virtual Bitmap" is discussed here. Each bit of the Partial Virtual Bitmap indicates if there is traffic buffered for a specific station within the BSS that the AP is prepared to deliver at the time the beacon frame is transmitted. For example, if no frames are buffered for the station n, the bit number n is 0. Otherwise, the bit number n is 1. The special beacon used in the new scheme can be developed from the Beacon frame of IEEE 802.11. It has the extended TIM element format shown in Figure 6.2(b), with a new field of “Up-link Time Schedule” appended. The Partial Virtual Bitmap is modified to indicate which clients are in the voice list. For example, when the bit number 6 is set, it indicates that Client 6 has set up a call and has been put in the voice list. The similar modification was implemented by software in [13], while keeping the hardware part unchanged.



Bytes:	1	1	1	1	1	1-251
	Element ID	Length	DTIM Count	DTIM Period	Bitmap Control	Partial Virtual Bitmap

(a) TIM element format [7]

Bytes: 1	1	1	1	1	1-251	
Element ID	Length	DTIM Count	DTIM Period	Bitmap Control	Partial Virtual Bitmap	Up-link Time Schedule

	Time Shift	Time Shift	……	Time Shift
Bytes:	2	2	……	2

(b) Modified TIM element format

Figure 6.2 Modified TIM element format in the Beacon frame

The field of “Up-link Time Schedule” consists of sub-fields “Time Shift” in 2 bytes [13], as shown in Figure 6.2(b). The Time Shift fields are used for the hub to designate the scheduled transmission time for all the voice clients, and the bits corresponding to their client number are all set in the Partial Virtual Bitmap. To minimize the Time Shift length, the first Time Shift field indicates the time between the first voice user’s scheduled transmission time and the time indicated in the Time Stamp field in the beacon [7] (another field of Beacon frame not included in Figure 6.2). The Time Stamp field is used for time synchronization of stations in a BSS and is broadcast periodically. Stations in a BSS use this value to adjust its timer. The subsequent Time Shift fields indicate the time difference between two scheduled transmissions of voice users [13]. Only the clients on the voice list (the corresponding bit numbers are 1 in the Partial Virtual Bitmap) are assigned the Time Shift fields.

With the combined overhead of the MCS and IEEE 802.11b attached, the final beacon frame is encapsulated. Its length is variable, depending on the total number of clients and how many clients are in the voice list, and is always known by the hub. The length of time slot for the beacon frame can be calculated, which is used in the following calculation and simulations. The implementation in software is left for further study.

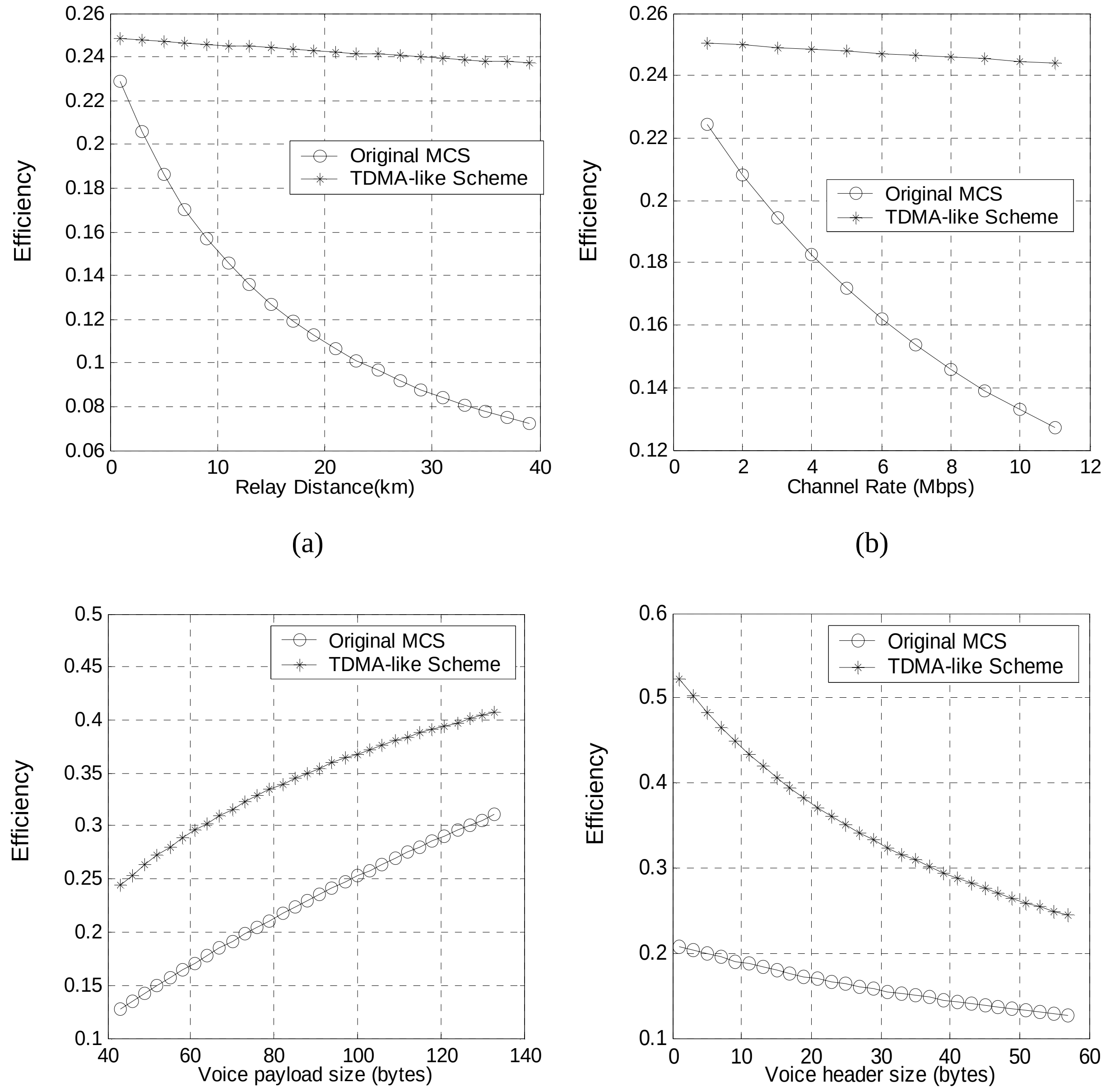
6.3 Improvement in Voice Efficiency

How much improvement can be achieved from the TDMA-like scheme depends on many factors such as the propagation distance, channel rate, the size of voice payload, etc. The efficiencies of the two voice transmission schemes are calculated and compared using the default parameters listed in Table 6.1. When one of parameters is varied in some range, the others are kept fixed.

Table 6.1 Default parameter setting in the efficiency calculation

Parameters	**Default Values**
Relay distance	15 km
Channel rate	11 Mbps
Voice payload	43 bytes
Voice overhead	57 bytes
Voice clients	50
ON/OFF interval	1 s /1.35 s
Guard time	1 μs

Figure 6.3 (a) to (d) show that the TDMA-like scheme is more efficient than the MCS polling algorithm at any conditions. The propagation delay is included in each scenario and slows down the MCS polling process, but its negative effect has been significantly reduced in the TDMA-like scheme. As a result, the new scheme is insensitive to the variation of distance and the inefficiency caused by the propagation delay is basically eliminated in the TDMA-like scheme. As shown in Figure 6.3(a), in an 11 Mbps channel, the efficiency of the MCS polling algorithm decreases quickly with increasing relay distance, whereas the TDMA-like scheme is basically constant. More improvement can be achieved at longer propagation distance. For example, the new efficiency (24.8%) is about 1.7 times of the old one (15.0%) at 10 km, and 2.8 times of the old one (8.5%) at 30 km in Figure 6.3(a). It shows that the TDMA-like scheme is more suitable to the networks with long propagation distance.



(a)

(b)

(c)

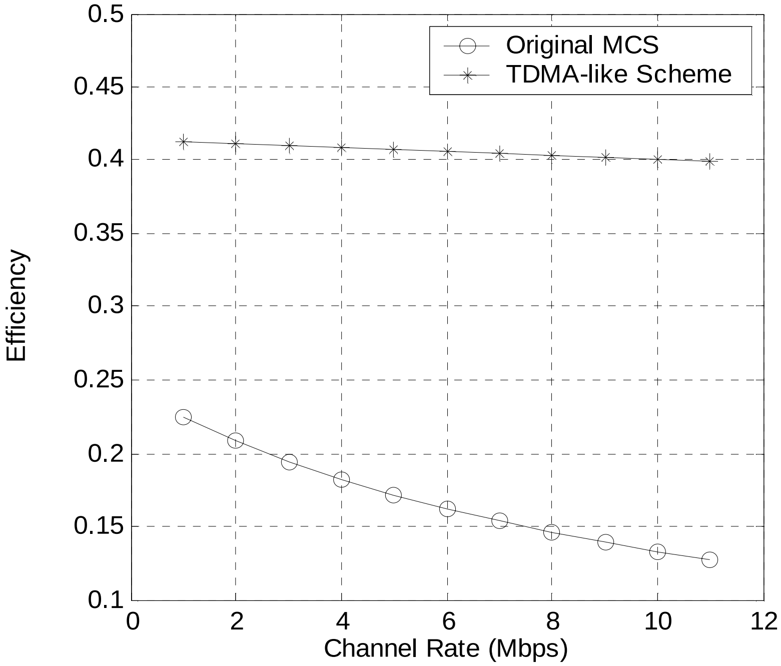
(d)

Figure 6.3 Comparison of efficiencies in both schemes

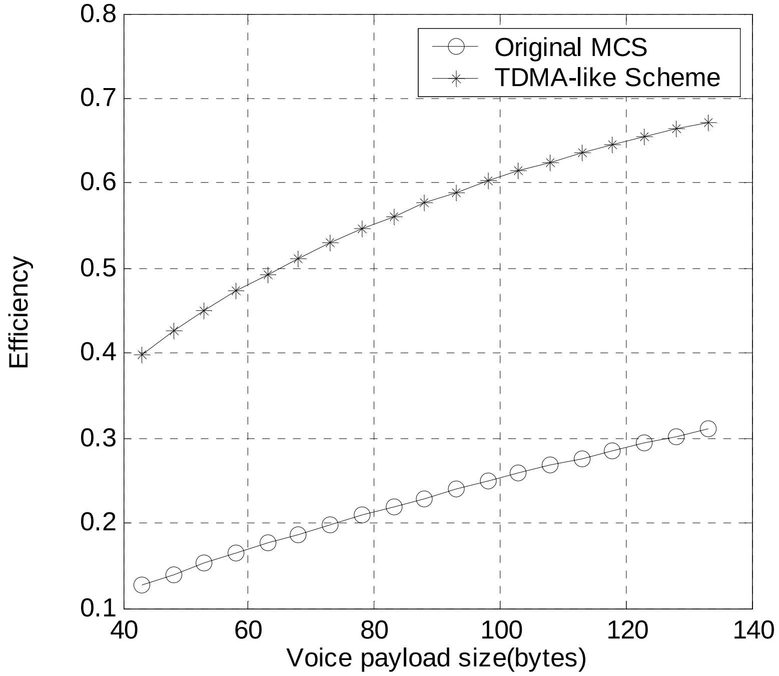
Changing the channel rate has much more impact on the MCS polling scheme than the TDMA-like scheme, as shown in Figure 6.3(b). The efficiency of the MCS polling algorithm is inversely proportional to the channel rate, because the negative effect of propagation delay becomes more obvious in a channel with higher rate. This is consistent with the discussion of the data efficiency in Section 4.2.3. With the minimized effect of propagation delay, the efficiency of the TDMA-like scheme is almost unaffected by the channel rate because the transmission time of payload and overhead is always changed synchronously. It can be concluded that the TDMA-like scheme is much more efficient than the MCS polling, especially working with high channel rate.

Although the negative effects of the propagation delay and the channel rate are basically eliminated in the new scheme, its efficiency is still below 26%, due to the poor ratio of the voice payload to the overhead. Increasing the size of voice payload is helpful to improve the efficiency of both schemes, for example, if the size of voice payload is increased from 43 bytes to 73 bytes, the efficiency can go up from 24.0% to 32.4% using the new TDMA-like scheme, shown in Figure 6.3(c). Reducing the overhead of voice packet is another option (Figure 6.3(d)), but it is infeasible in the current stage, because the 49 byte overhead pertains to the IEEE 802.11b products.

Another factor that limits the efficiency of the new scheme is the inevitable waste of bandwidth. Since in the realistic network the hub has no idea if it is in talk state or silence, each voice client is assigned a slot in the uplink frame, and quite a few slots are empty. The TDMA-like scheme in Figure 6.1 can be optimized by assigning slots only to the clients in talk state in the uplink frame. Its theoretical efficiency shown in Figure 6.4(a) is increased to around 41.0 % for the 43 byte voice payload, 3.2 times of the MCS



(a)



(b)

Figure 6.4 Efficiency of the optimized TDMA-like scheme

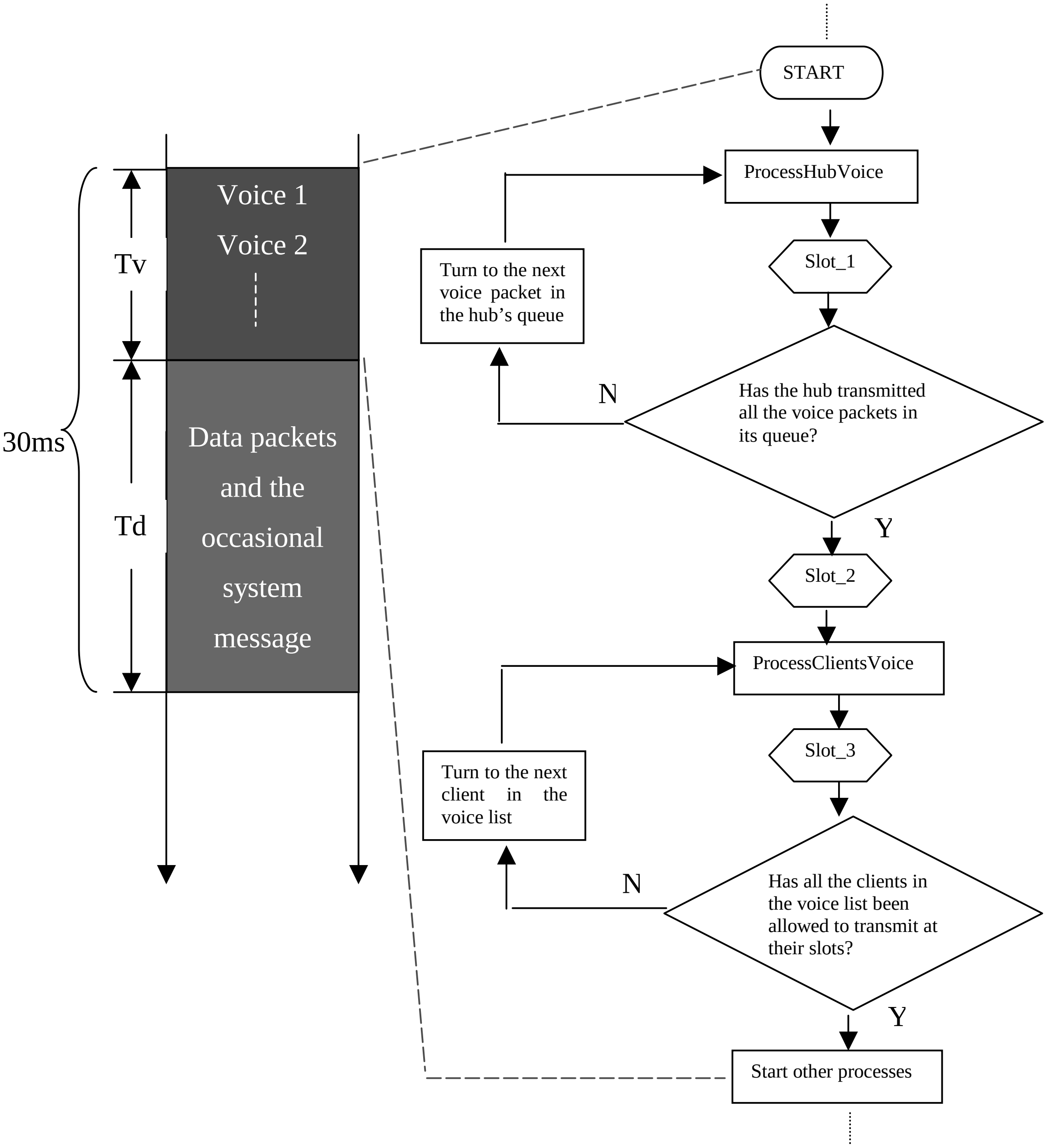
polling scheme in an 11 Mbps channel. If the 73 byte voice payload is used by doubling the MCS slot, the efficiency can go up to 54.0% (Figure 6.4(b)). However, due to the complexity in the practical implementation, only the TDMA-like scheme of Figure 6.1 with all slots assigned is simulated.

6.4 Implementation of the NS-2 Module of TDMA-like Scheme

New voice processing functions performing the TDMA-like scheme are written in C++ code in NS-2, with the flow chart described, and they are inserted into the simulated MCS model in place of the old one. Some necessary modifications have to be made to the other relevant modules.

6.4.1 Flow Chart of the New Voice Handling Process

Figure 6.5 is the flow chart for the TDMA-like voice transmission scheme. Without any change to the other functions that perform the polling of the data list, quiet list, and hot list, two voice processing functions (for the downlink and uplink transmissions respectively) written in C++ code are inserted into the "MCSController", taking the place of the original voice processing part. The parameters in Figure 6.5 are explained in Table 6.2, where δ_{trans} and δ_{prop} are the transmission delay and propagation delay.



Voice 1
Voice 2
Tv
30ms
Data packets and the occasional system message
Td
START
ProcessHubVoice
Slot_1
Turn to the next voice packet in the hub's queue
N
Has the hub transmitted all the voice packets in its queue?
Y
Slot_2
ProcessClientsVoice
Slot_3
Turn to the next client in the voice list
N
Has all the clients in the voice list been allowed to transmit at their slots?
Y
Start other processes

Figure 6.5 Flow chart for TDMA-like voice transmission scheme

When the function "ProcessHubVoice" is performed, the hub would continually issue voice tokens to those clients in turn, since it knows which clients are due to receive a voice packet. Each token is followed by a voice packet directed to the destination client. After the beacon frame from the hub is received by the clients, the uplink transmission begins. Another function, “ProcessClientsVoice” is performed allowing all the voice clients to respond in different scheduled slots assigned in the beacon. If a voice client has a packet to send, a voice token is issued from the client, followed by the voice packet in its slot. If the client happens to be in silence, it does not transmit in their slot. After all the uplink transmissions finish, the “MCSController” starts another process.

Table 6.2 Parameters used in the simulation of TDMA-like scheme

Parameters	Definitions	Expressions
Slot_1	The time interval between two consecutive transmissions from the hub to clients	δ_{trans} (voice payload + header)
Slot_2	The time interval between the ending of the last voice packet from the hub and the first up link transmission	δ_{trans} (beacon) + δ_{prop}
Slot_3	The time interval between two consecutive transmissions from clients to the hub	δ_{trans} (voice payload + header) + guard time

6.4.2 Modifications to the MCS Modules

In the simulation of the TDMA-like scheme, the 57 byte voice token (the combined overhead) is a modified ping packet, whose ability of bouncing back and forth between two nodes is disabled. Its only purpose is to notify the hub or clients to put their voice packets in the outgoing MCS link. Once arriving at the other side, it will be freed right away in simulation.

It is the responsibility of the “MCSController” to “command” the “PingAgent” to send voice tokens. However, it does not know which clients are due to receive a voice packet in the MCS simulated model discussed in Chapter 3, because the information of voice queues is contained in another module, the “MCSLink”. Modifications are required to implement the TDMA-like scheme in NS-2.

The ping packet sent by the Ping agent can be used as voice token or data token, by setting bit 6 or bit 7 in the Control 1 byte of the MCS header (Figure 3.3). Here, the third token type, named the “ask token”, is created by setting the unused bit 1 in the Control 1 byte (Control 1 = 0000 0010).

Every client is connected to the hub via an individual MCS link in the NS-2 topology. Each MCS link has two voice queues, one is in the client node, and the other is in the hub node. In the downlink transmission, before issuing a token to a voice client, the “MCSController” first checks the voice queue at the hub node by issuing an “ask token” to the “MCSLink”. Then the MCS link connected to the client returns the length of voice queue at the hub node to the “MCSController”. If the hub’s voice queue of the MCS link is not empty, it shows that the hub has a voice packet for that client. A voice token is then issued from the hub to the client, followed by the voice packet. If the voice queue is empty, it shows that the hub has no voice packet for that client. The “MCSController” just skips the client and turns to “ask” another MCS link connected to the next client in the voice list. By doing so, it can be realized in the simulation that the tokens and the voice packets are sent only to those clients due to receive voice packets in turn in the downlink transmission.

In the following uplink transmission, each client in the voice list is assigned a slot without being checked if its voice queue is empty or not. As a result, some slots are empty.

6.5 Simulation Results

Simulations are conducted using the topology of Figure 4.1, under various channel conditions, with the new voice simulated module incorporated (the guard time is set to 1 μs). All the simulation results are compared with the improved MCS polling algorithm. Lastly, the multi-rate voice transmission is discussed.

6.5.1 Improvement in Voice Capacity

The voice capacities at four channel rates when the distance from the hub to the relay station is kept 15 km, are listed in Table 6.3. The results of the improved MCS polling are also included for comparison. The voice capacity is denoted by N, the number of voice clients when the packet dropping starts in simulation. It goes up using the new scheme and the greatest improvement (a significant increase of 76.4%) is received in an 11 Mbps channel, which is consistent with Figure 6.3(b).

Table 6.3 Voice capacity and channel rate

Channel rate	N (MCS polling)	N (TDMA-like scheme)	Raised percentage
1 Mbps	22	23	4.55%
2 Mbps	41	46	12.2%
5.5 Mbps	94	128	36.2%
11 Mbps	148	261	76.4%

The CCD performance at the relay distance of 15 km in a 2 Mbps channel is taken as an example, as shown in Figure 6.6. With 41 voice clients involved, the delay performance has been improved compared to the MCS polling algorithm shown in Figure 5.6(a) and the maximum voice delay is decreased from the previous 68.0 ms to the current 31.5 ms. Moreover, in the simulation of the new TDMA-like scheme, voice packets begin to drop when the number of voice clients is increased to 46, while the number is 41 in the MCS polling algorithm (See Table 5.2(b)).

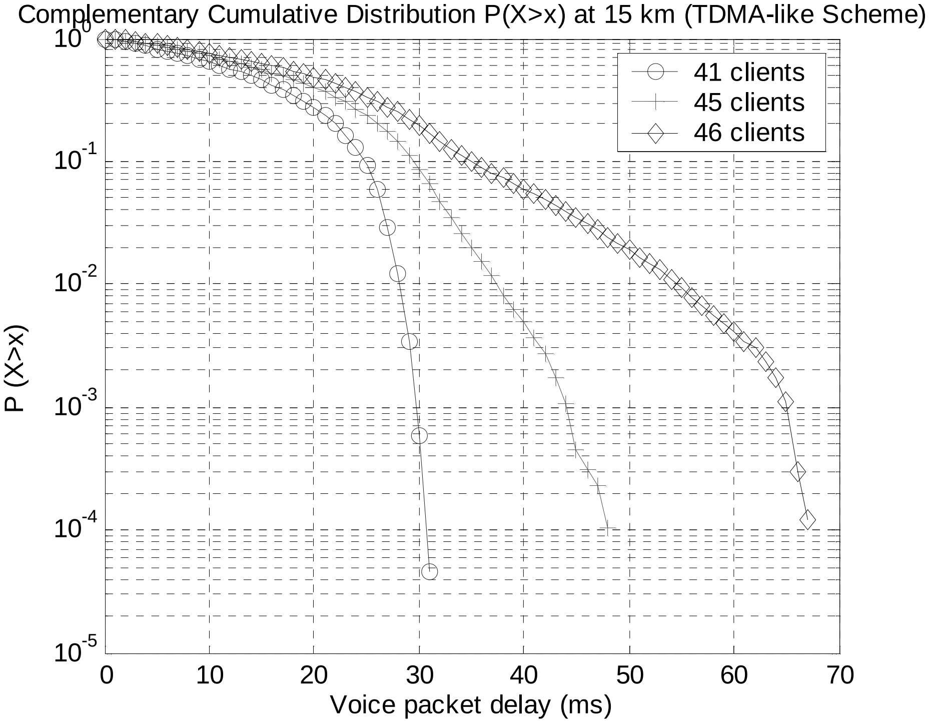


Figure 6.6 CCD curves of the TDMA-like scheme

The propagation distance also affects the voice capacity, in the same way as the voice efficiency (Figure 6.3(a)). The TDMA-like scheme has a larger voice capacity (N) than the MCS polling algorithm at the same distance, and the voice capacity keeps almost constant at different distances, listed in Table 6.4, which is tested in a 2 Mbps channel. Therefore the longer the propagation distance is, the more improvement can be obtained from this new scheme. For example, the improvement reaches 43.8% at 50 km.

Table 6.4 Voice capacity and relay distance

Distance from the hub to the relay station	N (MCS polling)	N (TDMA-like scheme)	Raised percentage
5 km	45	47	4.44%
15 km	41	46	12.2%
20 km	40	46	15.0%
25 km	38	46	21.1%
30 km	37	46	24.3%
50 km	32	46	43.8%

6.5.2 Improvement in Data Throughput

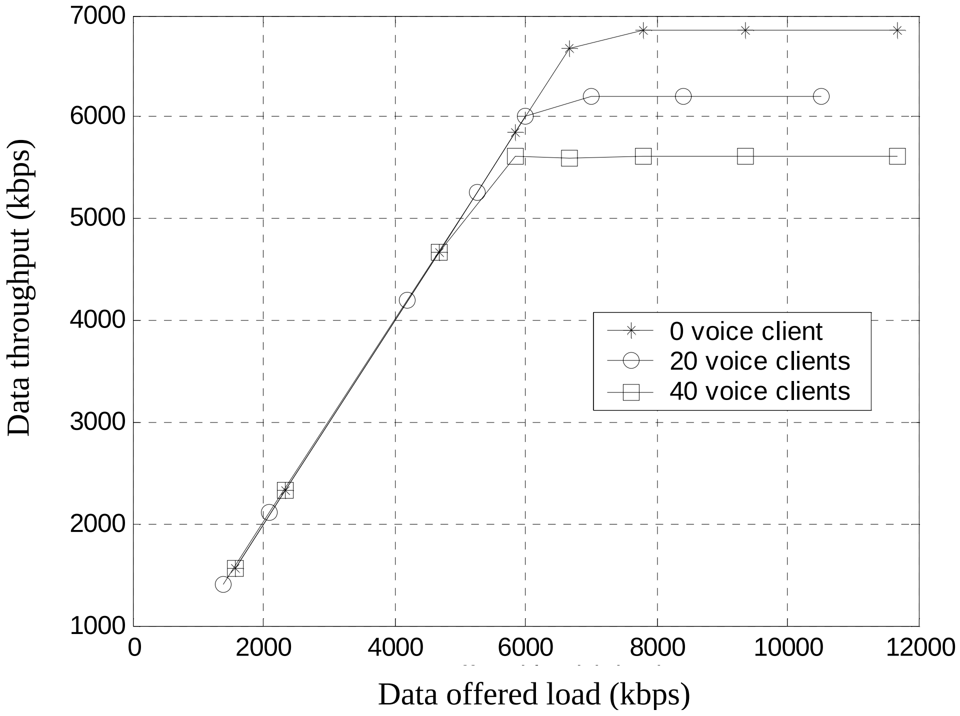


Figure 6.7 Data throughput in TDMA-like scheme with different number of voice clients

Data transmission can also benefit from the new scheme. Compared to the MCS polling algorithm, the new scheme takes less time to exchange voice packets in each time slot and leaves more time to transmit data packets. Figure 6.7 shows the variation of the data throughput versus the offered load in the TDMA-like scheme, while all the parameters are kept the same as in Figure 5.2 and Figure 4.3. From the three figures, it can be seen that using the TDMA-like scheme further improves the data throughput. For example, if 20 voice clients are involved in an integrated channel, the saturation throughput can be increased to 6211.3 kbps from 5749.9 kbps (Figure 5.2).

Compared to the improved MCS polling algorithm, the improvement achievable increases with the number of voice clients, (Figure 6.8) and relay distance (Figure 6.9). For example, in Figure 6.8, in an 11 Mbps channel at 15 km, the normalized saturation throughput (or the data efficiency) is increased by 2.1% with 10 voice clients included, and 13.0% with 60 voice clients included. In Figure 6.9, the number of voice clients is fixed at 20 and the channel rate is 11 Mbps. The data throughput achieved using the TDMA-like scheme decreases much lower than the MCS polling algorithm. The difference in the normalized saturation throughput becomes bigger with increasing propagation distance and can reach 9.4% at 50 km.

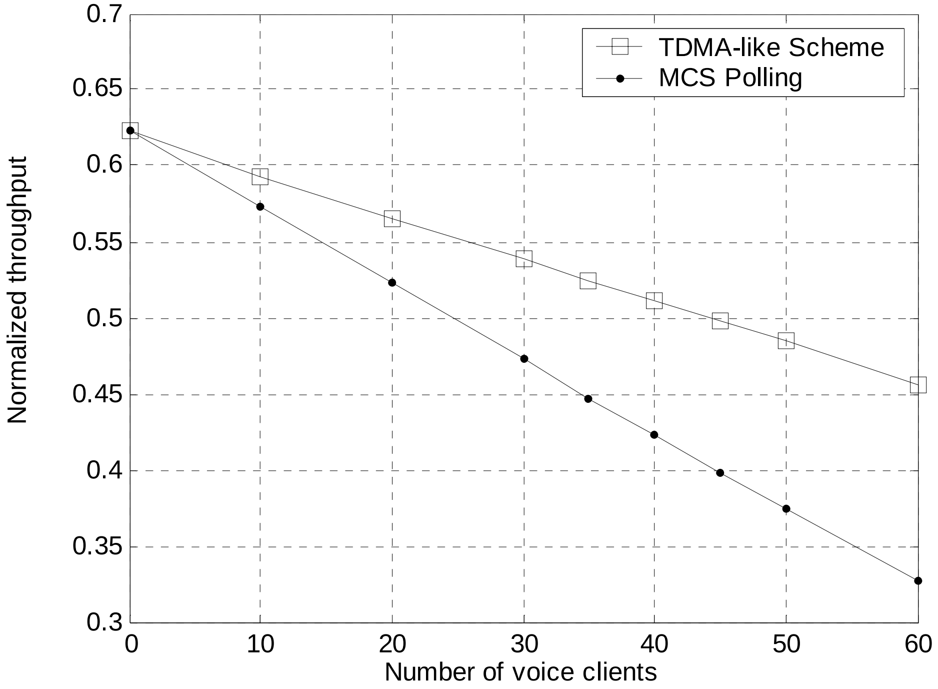


Figure 6.8 Saturation throughputs with different number of voice clients

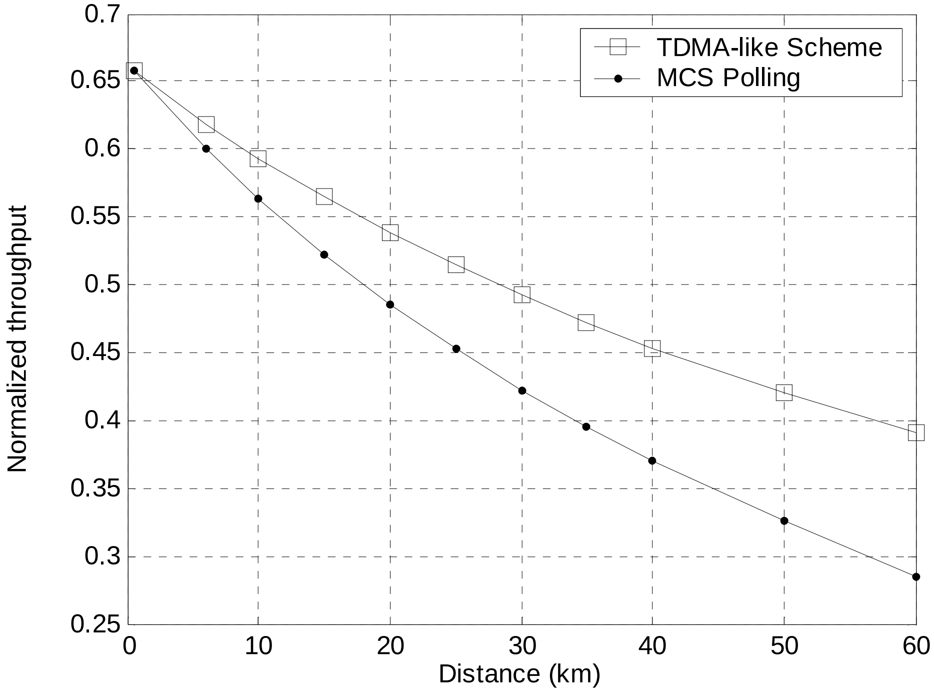


Figure 6.9 Saturation throughputs at different distances

6.5.3 Discussion to Multi-Rate Voice Transmission

In this application, the hub is able to communicate with voice clients using different access rates simultaneously. As stated in Section 4.6.1, the hub maintains a modified Client State Table, including the data rate of each voice client. In the downlink transmission, the hub controls its modem and transmitting rate, and sends the packets out continually according to the Client State Table. The beacon frame is always broadcast at 1 Mbps for reliable reception. In the uplink direction, the scheduled transmission time for each client is calculated in terms of its data rate and a “TDMA” frame with variable time slots is returned to the hub.

Assuming 20 voice clients have set up calls through the CM, simulation results show that the length of voice polling portion (Tv) in an MCS time slot depends on the distribution of voice clients.

-- 3.185 ms (All 20 clients use 11 Mbps.)

-- 25.658 ms (All 20 clients use 1 Mbps.)

-- 7.66 ms (8 clients use 11 Mbps, 6 clients use 5.5 Mbps, 4 clients use 2 Mbps, and 2 clients use 1 Mbps.)

-- 14.73 ms (2 clients use 11 Mbps, 4 clients use 5.5 Mbps, 6 clients use 2 Mbps, and 8 clients use 1 Mbps.)

The length of Tv can be shortened if the ratio of voice clients in higher rate is raised in a channel and more data packets can be exchanged in the remaining time. Compared to the simulation results of the MCS polling scheme in multi-rate application in Section 5.4.6, the voice portion (Tv) here is always shorter at the same distribution. It shows that the TDMA-like voice scheme can work more efficiently in the multi-rate application.

6.6 Summary

The TDMA-like scheme implemented in this chapter basically eliminated the negative effects of propagation distance and channel rate. Simulation results showed that it could improve the voice efficiency, and support more active voice users simultaneously in a channel than the MCS polling algorithm, assuming the guard time was kept in the range of 1-5 μs. The improvement is especially noticeable at long propagation distance and/or with higher data rates, where the MCS polling scheme can only get a very low efficiency.

With data traffic included in the same channel, the TDMA-like voice scheme can also effectively increase the data throughput. The longer the propagation distance from the MCS hub to the clients, the more improvement can be achieved by using the new scheme, which makes it more suitable for the wireless networks with large geographical coverage. More improvement in data throughput can also be achieved in higher rate channel, and/or with more voice clients involved.

However, the heavy burden of voice header limits the voice efficiency of both schemes. Two methods were verified to be able to further improve the efficiency. Doubling the MCS time slot is effective, but adds some delay. The uplink transmission of the TDMA-like scheme was optimized by discarding the empty slots and achieved a significant increase in voice efficiency (from 24.8% to 41.0%). Combining both methods can achieve the efficiency as high as 54.0%.

CHAPTER 7
SUMMARY AND CONCLUSIONS

7.1 Summary

There is a growing demand for faster, improved data and voice services in some rural areas of the world. Due to the lack of modern telecom infrastructure and the limitations of geographical conditions, wireless access maybe the only feasible solution.

IEEE 802.11b is currently by far the most widely used WLAN standard and has been well received. Another wireless access method, TR*Labs*' MCS prototype was designed to provide integrated Internet access and IP telephone service over large distances. A system solution that implements the MCS algorithm over the available IEEE 802.11b platform was proposed and studied in this thesis. It combines the advantages of both systems, that is, the MCS' capability of integrating VoIP and data services for rural areas and the standard IEEE 802.11b products widely available.

The IEEE 802.11 standard was briefly introduced. The polling algorithm of the original TR*Labs*' MCS prototype was described in detail and the simulation modules implemented in NS-2 Simulator previously were explained. The packet format used in the proposed system solution was created by integrating an MCS header into a standard IEEE 802.11b frame.

A system test bed was set up inside NS-2. The simulations to test the feasibility of the MCS algorithm in the proposed network were conducted and further modifications to improve system performance were made through this thesis.

The ability of the original MCS algorithm in supporting data transmission in the proposed rural network was examined in constant rate radio channels. The ARF feature of IEEE 802.11b was incorporated into the MCS data simulations, followed by the discussion about the unsuitability of the MCS algorithm for the multi-rate data transmission. To support this application, the original MCS data polling method was modified with the newly added capability of repeated polling, which was simulated at different cases.

The voice polling method of the original MCS algorithm was first improved, by skipping the sending of acknowledgement to the voice packets from clients. Using the improved MCS polling algorithm, the delay of voice packet between the CM and clients was measured and the final end-to-end delay was estimated. The number of active voice users supported in a channel when the packet dropping started (defined as the voice capacity) was also tested in various conditions. The voice inefficiency of the MCS polling algorithm was analyzed and discussed.

To further improve the voice efficiency, a new TDMA-like voice transmission scheme was implemented and simulated. A new processing module for the exchange of voice packets was implemented, without changing the processes of data exchange. The voice performance and data throughput in various channel conditions were measured and compared with the MCS polling algorithm.

7.2 Conclusions

TR*Labs*' MCS was designed to be able to prioritize delay sensitive voice while still transporting data efficiently, but its "non-standard" nature was felt to be detrimental for broad commercial development. Building the MCS algorithm over the most widely used IEEE 802.11b WLAN standard was considered as a promising solution.

The original MCS algorithm could provide reasonable data performance, which is comparable to the standard Distribution Coordination Function (DCF) operation of IEEE 802.11b, if both were simulated at similar conditions of constant rate radio channels. Without voice traffic involved, the MCS' data efficiency, 72.0%, 71.0%, 67.4%, or 62.3%, was achieved at the MAC layer with data rate 1, 2, 5.5, or 11 Mbps, at the relay distance of 15 km.

The Automatic Rate Fallback (ARF) feature of the 802.11b platform was incorporated into the MCS simulation. However, it was found that when clients with different data rates coexisted in one channel, the original MCS algorithm gave an unexpected result that every client got almost equally low throughput, because the whole process of packet exchange was dragged down by polling those low rate clients. It was concluded that the original MCS algorithm performed badly in multi-rate data transmission.

The modified data polling method of the MCS algorithm with the capability of repeated polling was verified to be able to provide efficient channel utilization for each data client. With the repeated polling cycles, such as [9, 5, 2, 1], for the clients with data rate 11, 5.5, 2, or 1 Mbps, the throughputs per client, 154.4, 86.4, 34.7, or 17.5 kbps could be achieved respectively, very close to the throughputs achieved in four separate channels of 11, 5.5, 2, or 1 Mbps. The problem that restricted the original MCS scheme in the multi-rate application was basically overcome.

Compressing the voice exchange scenarios in each MCS time slot not only made voice transmission more efficiently, but also improved data throughput from the original 5496.2 kbps to 5749.9 kbps when 20 voice clients were involved in an 11 Mbps channel at 15 km. The MCS polling delay was the major part of the voice delay introduced between clients and CM. The voice service provided by the system solution is acceptable (with the estimated end-to-end delay of 97.5 ms). More voice capacity could be supported when a higher channel rate or a shorter relay distance was used. The variation of data traffic had little effect on the voice traffic, which demonstrated the ability of MCS to prioritize the voice service. However, the one-by-one polling led to the inefficiency of MCS' voice transmission. For example, in an 11 Mbps channel, the voice efficiency was only 12.7%, at a distance of 15 km. The time wasted became more severe with increasing relay distance. At 30 km, the voice efficiency was as low as 8.6%.

Assuming the problem of big distance deviation between the hub and clients could be resolved and the guard time was kept in the range of 1-5 μs, the proposed TDMA-like voice transmission scheme could provide the voice efficiency stable at around 24.8% in any channel conditions, which basically eliminated the effects of propagation delay and channel rate. Using the TDMA-like scheme, a significant increase of 76.4% of voice capacity in an 11 Mbps channel was achieved at the relay distance of 15 km. At the same channel conditions, the normalized saturation throughput (or the data efficiency) was increased by 2.1% with 10 voice clients included, and 13.0% with 60 voice clients included, compared to the improved MCS voice polling. If the relay distance was varied and the number of voice clients was fixed at 20, the normalized saturation throughput was improved by 1.9% at 6 km and 9.4% at 50 km. Generally speaking, at longer relay distance, or using higher channel rate, more improvement could be achieved in the voice efficiency, voice capacity, and data throughput. Obviously, the new TDMA-like voice scheme was especially suitable for the rural access network where long relay distance was required.

Developed from the original MCS, the modified data polling scheme supporting multi-rate data transmission and the TDMA-like voice scheme, which is much more efficient in voice transmission, constituted a new “MCS” algorithm. Using the new “MCS” in the proposed rural access network, dramatic improvements could be achieved in both data and voice performance over the available IEEE 802.11b products and also over the original MCS algorithm. It could be considered as a feasible and cost-effective solution for providing the quality integrated voice and data services in sparsely populated rural areas.

7.3 Suggested Future Work

As was discussed in Chapter 5 and Chapter 6, the poor ratio of voice payload (43 bytes) to overhead (57 bytes) is one of reasons for the low voice efficiency. Although it has been verified theoretically that increasing the voice payload is helpful to improve the efficiency for both the MCS and the TDMA-like scheme, no further simulation has been performed. Future work could be done to extend the time slot from 30 ms to 60 ms in the MCS algorithm, which can almost double the size of voice payload. The simulated model has to be modified and all the simulations must be updated as well.

It has also been suggested that the efficiency of the TDMA-like voice transmission can be improved from about 24.8% to 41.0% by discarding the empty slots of the uplink “TDMA” frame. To achieve this, the hub is required to be informed whenever the state transition of voice users occurs. In the uplink, the More Data bit in the Frame Control field of the MAC header can be used as a piggyback to indicate to the hub that the client has more packets to be sent, and the hub assigns a specific time for the client to transmit at the following uplink frame. When a voice user changes its state from silence to talk period, it must let the hub know it is time to assign it a transmission slot in the next uplink frame [13]. Other than the requests of call setup and tear down, a new system message is required to inform the hub of the state transition. In addition, an “active voice list” is required in place of the original “voice list”, which involves only the voice clients in the talk state.

The setting of guard time is crucial to the TDMA-like voice transmission scheme. How to keep the guard time in a small range while avoiding the overlap in the uplink transmission at the same time also needs more consideration. Simulation and discussion can be conducted at various scenarios to propose a detailed implementation approach, which will make the TDMA-like scheme more complete and feasible in real networks.

REFERENCES

[1] “Big Increase in WLAN users”, available at http://www.silicontrust.com/home/news/86582ED3EBD34FEFB1C6137617A770EB.asp

[2] R.A. Dayem, Mobile Data & Wireless LAN Technologies, Prentice-Hall Inc., New Jersey, 1997.

[3] D.R. Johnson, “Performance of A Rural Wireless PTMP Network With Integrated Data and VoIP”, M.Sc Thesis, University of Saskatchewan, 2002.

[4] “#199901 Multipoint Communications System”, TR*Labs*’ Internal Document, TR*Labs* Saskatoon, SK, 1999, available at http://www.trlabs.ca/techshowcase/index.html, choose Wireless Communications, bulletin available to TR*Labs*’ sponsors.

[5] J.F. Kurose, K.W. Ross, Computer Networking-A Top-Down Approach Featuring the Internet, Addison Wesley Longman Inc., New York, NY, 2001.

[6] “Wireless Networking Q&A”, available at http://www.vicomsoft.com/knowledge/reference/wireless1.html.

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

[8] Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications: Higher-Speed Physical Layer Extension in the 2.4GHz Band, IEEE Std. 802.11b, 1999.

[9] B.P. Crow, I. Widjaja, J.G. Kim, P.T. Sakai, “IEEE 802.11 Wireless Local Area Networks”, IEEE Communications Magazine, pp. 116-126, September 1997.

[10] C. Andren, M. Webster, “CCK Modulation Delivers 11Mbps for High Rate IEEE 802.11 Extension”, Wireless Symposium/Portable By Design Conference Proceedings, Spring 1999.

[11] H. Liu, J. Wu, “ Packet Telephony Support for the IEEE 802.11 Wireless LAN”, IEEE Communications Letters, vol. 4, no. 9, pp. 286-288, September 2000.

[12] Cisco IP Telephony QoS Design Guide, Chapter 1, available at http://www.cisco.com/univercd/cc/td/doc/product/voice/ip_tele/avvidqos/qosintro.htm.

[13] H. Liu, J.C. Wu, “A Scheme for Supporting Voice over IEEE 802.11 Wireless Local Area Network”, Proc. Natl. Sci. Counc. ROC(A), vol.25, no.4, pp. 259-268, 2001.

[14] T.J. Kostas, et al., “Real-Time Voice Over Packet-Switched Networks”, IEEE Network, pp. 18-27, January/February 1998.

[15] D. Deng, R. Chang, “A Priority Scheme for IEEE 802.11 DCF Access Method”, IEICE Transaction on Communications, vol.E82-B, no.1, pp.96-102, January,1999.

[16] W.K.Kuo, C.Y.Chan, and K.C.Chen, “Time Bounded Services and Mobility Management in IEEE 802.11 Wireless LANs”, in Proc. 1997 IEEE Int. Conf. Personal Wireless Commun., 1997, pp.157-161.

[17] J. Sanchez, R.Martinez, and M.W.Marcellin, “A Survey of MAC Protocols Proposed for Wireless ATM”, IEEE Network, vol.11, pp.52-62, Nov-Dec,1997.

[18] L.L. Lipoth, “MCS Hub/Client System Software Overview Revision 1.01”, TR*Labs*’ Internal Document, TR*Labs* Saskatoon, SK, Oct, 1998.

[19] M. Greis, “Tutorial for the Network Simulator “ns””, available at http://www.isi.edu/nsnam/ns//tutorial/index.html.

[20] The ns Manual, available at http://www.isi.edu/nsnam/ns/tutorial/index.html.

[21] “ORiNOCO AP-200 Access Point Specifications”, available from http://www.proxim.com/products/wifi/ap/

[22] A.R. Prasad, “Performance Evaluation, System Design and Network Deployment of IEEE 802.11”, Wireless Personal Communications, vol.19, pp.57-79, 2001.

[23] “DWL-120+ Technical Information”,
available from http://www.dlink.co.uk/2105_2107.htm

[24] “ITU-T Recommendation G.729”, available from
http://www.ece.cmu.edu/~ece796/documents/g729.pdf

APPENDIX A

Tables and Figure from Data Sheet and Literatures

Table A.1 Range in ORiNOCO AP-200 Specifications [21]

RANGE (meters)	11 Mbps	5.5 Mbps	2 Mbps	1 Mbps
Open	160m	270m	400m	550m
Semi-open	50m	70m	90m	115m
Closed	25m	35m	40m	50m

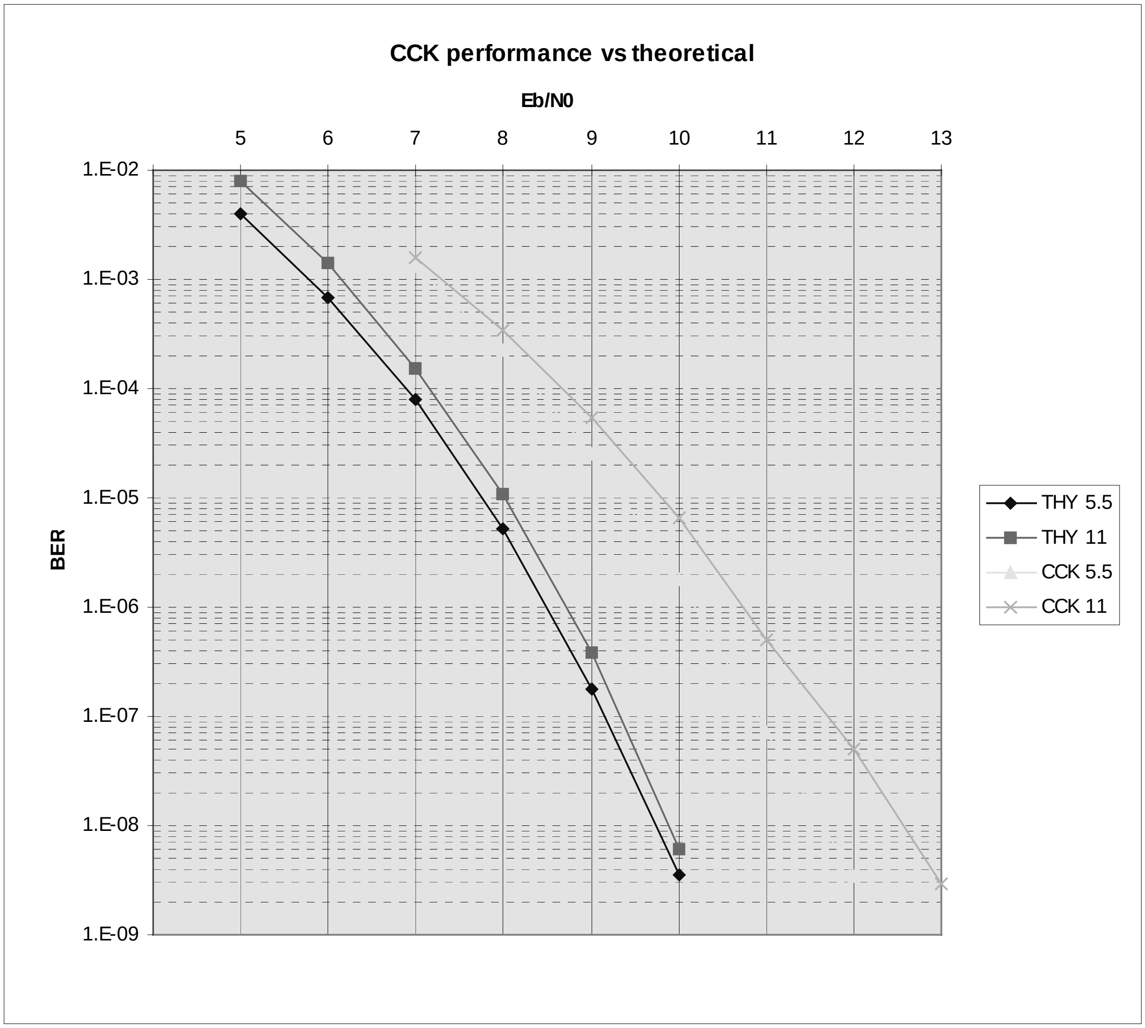


Figure A.1 Performance of the HFA3860B chip [10]

Table A.2 BER Improvement by the fallback of data rate from Figure A.1

CCK 11 Mbps		**CCK 5.5 Mbps**	
BER	Eb/No	Eb/No	BER
10^{-4}	8.7dB	11.7dB	10^{-8}
10^{-3}	7.3dB	10.3dB	10^{-6}

Table A.3 IEEE 802.11b Reliable ranges according to path loss models [22]

Data rate	1 Mbit/s	2 Mbit/s	5.5 Mbit/s	11 Mbit/s
Receiver sensitivity for BER 10^{-5}	-93 dBm	-90 dBm	-87 dBm	-84 dBm
Range covered 99% point TX power	15 dBm			
Open Plan Building	485 m	354 m	259 m	189 m
Semi Open Office	105 m	85 m	69 m	56 m
Closed Office	46 m	40 m	34 m	29 m

APPENDIX B

Theoretical Results of Efficiency

Table B.1 Voice efficiencies in original MCS and improved MCS polling schemes

Voice payload	43bytes
Header size	57bytes
Distance	15km

Channel rates (Mbps)	Transmission delay (µs)		Propagation delay (µs)	Scenario	Original MCS Polling	Improved MCS Polling
	Payload	Header				
1	344.0	456.0	50.0	1	0.0%	0.0%
				2	19.0%	25.4%
				3	25.4%	25.4%
				4	31.9%	40.5%
				Average	19.1%	22.8%
2	172.0	228.0	50.0	1	0.0%	0.0%
				2	18.0%	23.6%
				3	23.6%	23.6%
				4	30.5%	38.2%
				Average	18.0%	21.4%
5.5	62.5	82.9	50.0	1	0.0%	0.0%
				2	15.2%	19.0%
				3	19.0%	19.0%
				4	26.4%	32.0%
				Average	15.2%	17.5%
11	31.3	41.5	50.0	1	0.0%	0.0%
				2	12.2%	14.6%
				3	14.6%	14.6%
				4	21.8%	25.5%
				Average	12.2%	13.7%

Table B.2 Theoretical efficiency of Data Scenario 4

Data transmission efficiency						
Data payload	584	bytes				
ACK	48	bytes				
Header size	57	bytes				
Distance	15	km				
Channel rates (Mbps)	Transmission delay (μs)			Propagation delay (μs)	Scenario	MCS
	Payload	ACK	Header			
1	4672.0	384.0	456.0	50.0	4	71.6%
2	2336.0	192.0	228.0	50.0	4	70.5%
5.5	849.5	69.8	82.9	50.0	4	67.0%
11	424.7	34.9	41.5	50.0	4	62.1%