Voice Call Capacity Over Wireless Mesh Networks

Pradeep Gunda Bhat

Follow this and additional works at: https://egrove.olemiss.edu/etd

Part of the Electrical and Computer Engineering Commons

Recommended Citation

https://egrove.olemiss.edu/etd/124

This Dissertation is brought to you for free and open access by the Graduate School at eGrove. It has been accepted for inclusion in Electronic Theses and Dissertations by an authorized administrator of eGrove. For more information, please contact egrove@olemiss.edu.
VOICE CALL CAPACITY
OVER WIRELESS MESH NETWORKS

A Thesis
presented in partial fulfillment of requirements
for the degree of Master of Science
in the Department of Electrical Engineering
The University of Mississippi

by

PRADEEP GundA BHAT

JULY 29, 2011
Abstract

The goal of this thesis is to understand the voice call carrying capacity of an IEEE 802.11b/e based ad hoc network. We begin with the modelling of conversational speech and define a six state semi-Markov voice model based on ITU-T P59 recommendation. We perform a theoretical analysis of the voice model and compare it with results obtained via simulations. Using a Java based IEEE 802.11 medium access layer simulator, we determine the upper-bound for the number of voice calls carried by an ad hoc network. We use a linear topology with the ideal carrier sensing range and evaluate the number of calls carried using packet loss and packet delay as metrics. We observe that, for one, two, three and four hop, 5.5 Mbps IEEE 802.11 wireless links have an upper-bound of eight, six, five, and three voice calls respectively. We then consider a carrier sensing range and a path loss model and compare them with the ideal case. We observe, after considering a carrier sensing range with path loss model, there is a reduction in the number of calls carried by the linear networks. One, two, three and four hop 5.5 Mbps IEEE 802.11 wireless links support eight, five, four, and two voice calls respectively, when a carrier sensing range and a path loss model is considered. We also find that by adopting packet dropping policies at the nodes, we improve the call carrying capacity and quality of service on the network. In our simulations of a two hop network in path loss conditions, we find that, by adopting a time delay based packet dropping policy at the nodes, the number of calls supported simultaneously increased from five to six. In a four hop linear network we find that by total packet loss is reduced by 20%, adopting a random packet dropping policy and by 50% adopting a time delay based packet dropping policy. Although there is no change in number of calls supported, load on the network is reduced.
This work is dedicated to my parents, for all their support and encouragement.
Acknowledgements

I would like to thank my advisor, Prof. John N. Daigle for everything he has done during my graduate studies. Without his unwavering support it would have been impossible to complete my work. I would also like to thank Prof. Ronald A. Wagstaff for his academic support. I gratefully acknowledge the assistantship of my friends Nishchal Chaudhary, Himanshu Dwivedi and Mir M. Ali who have helped me.

Last but not the least, I am grateful to my parents and my brother for all their support, patience and encouragement.

University, Mississippi

Pradeep Gunda Bhat

July 2011
# Table of Contents

1 Introduction .................................................. 1
   1.1 Motivation for the Thesis ............................... 1
   1.2 Literature Survey ......................................... 2
   1.3 Organization of this Thesis ............................. 4

2 The IEEE 802.11 MAC and Physical Layer ................. 6
   2.1 Introduction to MAC layer ............................... 7
      2.1.1 Independent Basic Service Set ..................... 8
      2.1.2 Basic Service Set .................................. 9
      2.1.3 Extended Service Set ............................... 10
   2.2 Distributed Coordination Function ..................... 10
      2.2.1 Carrier Sense ....................................... 11
      2.2.2 The Acknowledgment Frame ......................... 15
      2.2.3 Hidden node and RTS/CTS ........................... 15
      2.2.4 Frame Fragmentation ................................. 16
   2.3 Point Coordination Function ............................ 16
   2.4 Enhanced Distributed Coordination Function .......... 18
      2.4.1 Transmission Opportunity ......................... 20
   2.5 Physical Layer ........................................... 20

3 Voice Conversation Model ................................. 21
   3.1 Overview ................................................. 21
   3.2 Voice Model ............................................. 21
      3.2.1 Six State Model ..................................... 22
      3.2.2 G 723.1 Voice Codec ............................... 26

4 Simulator and Simulation Setup .......................... 28
   4.1 Introduction ............................................ 28
   4.2 Simulator Overview ..................................... 28
   4.3 Link Layer Simulation Algorithm ...................... 29
      4.3.1 Transmitted Power .................................. 31
      4.3.2 Path Loss Model and Packet Error Rate ........... 31
      4.3.3 Node States in MAC ................................. 32
List of Tables

3.1 Temporal parameters in conversational speech . . . . . . . . . . . . . 25

5.1 Impact of packet dropping policies on a four hop 5.5 Mbps linear network under path-loss channel conditions . . . . . . . . . . . . . . . 41
List of Figures

2.1 IBSS/Ad hoc mode ........................................... 8
2.2 BSS mode .................................................. 9
2.3 ESS mode .................................................. 10
2.4 Carrier sense .............................................. 12
2.5 IEEE 802.11 DCF basic access mechanism .................. 13
2.6 Random back-off with distributed coordination function (DCF) medium access .............................................. 14
2.7 IEEE 802.11b/e basic access mechanism without RTS/CTS .... 15
2.8 RTS/CTS operation ........................................... 17
2.9 Basic EDCF operation in IEEE 802.11 ....................... 19

3.1 State transition model for conversation ....................... 22
3.2 Six state voice model ....................................... 23
3.3 Spurts and gaps over a period of time ........................ 25
3.4 Cumulative distribution of duration of talk spurt from of conversational speech as measured by simulation .......... 27
3.5 Cumulative distribution of duration of silence gap from of conversational speech as measured by simulation ........ 27

4.1 Packet flow in link layer simulator ........................... 30

5.1 Survivor function of one hop wireless link supporting 8 calls on a 5.5 Mbps under ideal channel conditions and minimal collisions .... 37
5.2 Survivor function of three hop wireless link supporting 5 calls on a 5.5 Mbps under ideal channel conditions and minimal collisions .... 38
5.3 Relation between total packet loss and the number of calls supported in a three hop 5.5 Mbps wireless link under ideal channel conditions and minimal collisions .................. 39
5.4 Upper-bound of simultaneous voice calls supported by a 5.5 Mbps wireless link under ideal channel conditions and minimal collisions as a function of hop count .................. 39
5.5 Comparison of survivor functions of packet delays in a three hop 5.5 Mbps linear network carrying six voice calls under ideal physical channel conditions ......................................................... 41
5.6 Four hop linear network topology ......................................................... 42
5.7 Comparison of survivor functions of packet delays in a four hop 5.5 Mbps linear network carrying three voice calls under path loss channel conditions. ................................................................. 42
5.8 Capacity of wireless multi-hop links under both ideal and path-loss channel conditions, with and without packet dropping policy. ........ 43
List of Abbreviations

AP  access point
APs  access points
ACK  acknowledgement
AC  access category
AIFS  arbitrary interframe space
AIFSN  arbitrary interframe space number
ATM  asynchronous transfer mode
BSS  basic service set
BER  bit error rate
CAC  call admission control
CBR  constant bit rate
CCK  complementary code keying
CSMA  carrier sense multiple access
CSMA/CA  carrier sense multiple access with collision avoidance
DCME  digital circuit multiplication equipment
CW  contention window
CTS  clear to send
DSSS  direct sequence spread spectrum
DCF  distributed coordination function
DIFS  distributed interframe space
ESS  extended service set
EDCA  enhanced distributed channel access
EDCF  enhanced distributed coordination function
EDCA-TXOP  enhanced distributed channel access - transmission opportunity
HCF  hybrid coordination function
IBSS  independent basic service set
MAC  medium access control
MPDU  MAC protocol data unit
NIC  network interface card
NAV  network allocation vector
PCF  point coordination function
PHY  physical
QoS  quality of service
RTS  request to send
SIFS  short interframe space
TDM  time division multiplexing
TXOP  transmission opportunity
VoIP  voice over IP
WLAN  wireless local area network
WLANs  wireless local area networks
ITU-T  international telecommunication union
Chapter 1

Introduction

The objective of this thesis is to contribute to the understanding and improvement of the voice call carrying capacity of an IEEE 802.11b/e based ad hoc network. The call carrying capacity of an ad hoc network refers to the number of simultaneous video voice calls that the network can support. Since the channel access in the network is contention based and the voice traffic is delay sensitive, in order to simulate voice calls over ad hoc network it is important to accurately model a voice conversation to understand the impact of contention on quality of service (QoS).

In this thesis, we start by modelling conversational speech as a six state semi-Markov process based on the ITU-T P59 recommendation. We then determine the upper-bound of number of calls voice calls that can be handled by multi-hop wireless link. We then propose packet dropping (load shedding) policies for a wireless link to reduce packet latency and improve QoS. The results of our research were obtained by using a Java based IEEE 802.11 medium access control (MAC) layer simulator.

1.1 Motivation for the Thesis

The wireless local area networks (WLANs) [1] utilize infrastructure mode that requires the use of one or more access points (APs). In the infrastructure mode, an access point (AP) provides an interface to a distribution system (e.g., Ethernet), which
enables wireless users to utilize corporate servers and Internet applications. However, there exists an ad hoc mode, which allows users to spontaneously form a peer-to-peer wireless network. Since an ad hoc network does not rely on a pre-existing infrastructure, the idea of using them at times of a calamity, or whenever the fixed communication infrastructure is not suitable, is attractive.

However, in ad hoc mode the nodes, themselves, must provide services for routing and address assignment, which causes significant packet delays and overheads since, IEEE 802.11 standards use a contention based MAC which also adds to the packet loss occurring due to wireless physical medium. The delays and overheads in transmitting packets and path loss conditions in channel can hinder in maintaining an acceptable QoS, especially for real-time applications. Hence, there is a need to understand and analyse the wireless link quality, propagation path loss, and service carrying issues of these networks in order to implement acceptable real-time applications.

1.2 Literature Survey

In this section we provide a brief summary of the literature relevant to voice carrying capabilities of wireless ad hoc networks.

In [2], the role of codec packetization on the channel efficiency of WLANs is examined, and through simulations show its significant impact on voice capacity. The dynamics of the collision probability, packet drop rate and medium activity, as more active voice users enter the system is also reported. It is shown that higher packetization rates may increase network capacity with little to no degradation in QoS.

In [3], a new performance model is proposed for the IEEE 802.11 WLAN in ad hoc mode. The ad hoc mode was chosen with the aim of using interconnected wireless local area network (WLAN) clusters where no base station exists. The effects of parameters
such as throughput, delay, packet fragmentation, buffer size and retransmission limit for a single hop scenario is determined and is extended to analyse clustered WLANs. It is shown that over a typical bit error rate (BER) of $\{10^{-4} - 10^{-6}\}$, multiple values of fragmentation factor, buffer size, and retransmission limit are found that optimize a specific performance measure or their joint weighted measure.

In [4], the capacity of an IEEE 802.11b network carrying voice calls is evaluated. A scenario where multiple users are connected to a single AP is considered to find the call carrying capacity of the AP using G.711, and G.729 voice encoding schemes and a range of voice packet sizes. It is concluded that selecting the packet size appropriately given the delay budget and channel conditions, the capacity can be maximized and in most cases optimum packet size selection can be made without knowledge of the channel conditions. The use of G.729 has been shown to allow greater capacity than the use of G.711 voice codec.

In [5], a modified MAC protocol supporting voice traffic over the IEEE 802.11 WLAN is proposed. The proposed scheme adapts the power-saved mode of the existing IEEE 802.11 specifications in such a way that its capacity approaches the time division multiplexing (TDM) access mode traffic capacity.

In [6], the conversational speech capacity of WLANs is simulated and the results analysed. It is found that the voice capacity is a strong function of the channel bandwidth, codec packetization interval, data traffic and packet size. It is found that by increasing the packetization interval, the channel efficiency increases by decreasing the number of packets generated per second. The use of request to send (RTS)/clear to send (CTS) procedure further reduces the call carrying capacity. Its also found that larger packets increase the packet inter-arrival times for the same data rate, and hence leave a greater share of resources for voice users.

In [7], the capacity of a carrier sense multiple access with collision avoidance
(CSMA/CA) WLAN with voice and data services using TCP/IP protocol is analysed to obtain a lower bound for the capacity of the wireless networks with voice and data services. A lower bound for the voice capacity of the wireless networks was calculated. Using UDP for carrying the voice and TCP/IP for the data traffic, the system was modelled with a non-preemptive priority queuing system and a carrier sense multiple access (CSMA) channel. The maximum number of voice users can be determined under different conditions for maximum allowable packet time delay, channel bandwidth, and a specified data traffic.

In [8], a measurement-based model that can accurately model the available capacity and guide call admission decisions and route selection procedures for voice over IP (VoIP) in a wireless mesh are proposed. It is found that in order to maintain QoS, call admission control (CAC) must be performed. However, without any reasonable model of multi-hop capacity of the network, the admission decisions cannot be taken. It is also concluded that because of the wireless interference, looking for a feasible route to accommodate an incoming call can be computationally hard. In order to overcome computational complexity, an assumption of the knowledge of the ratio of interference and carrier sensing ranges is introduced to ensure that path segments of constant length can be evaluated separately to determine feasibility in polynomial time.

In [9], both IEEE 802.11b and 802.11e multi-hop networks are simulated, voice traffic bottlenecks are identified and a burst queue scheme is proposed to cooperate with IEEE 802.11 MAC protocols.

1.3 Organization of this Thesis

In this thesis, we determine the upper bound of total number of calls carried by an ad hoc network as a function of hop count. The end-to-end packet delay and packet loss
for varying sizes of linear topology is measured. Survivor functions of packet delay, to
determine the fraction of packets delayed by more than 150 ms, is plotted. From the
packet loss and delays, we then go on to evaluate QoS. Two packet dropping (load
sheding) policies to decrease link latency and improve QoS are also proposed.

The organization of the thesis is as follows. In Chapter 2, The physical (PHY) and
MAC layer of IEEE 802.11 based networks are discussed. In Chapter 3, we discuss
a six state voice conversation model, which is derived from international telecommu-
nication union (ITU-T)-P59 specification [10]. In Chapter 4, we provide an overview
of simulator design and description of the simulation network set-up. The simulation
trace files which are obtained are then analysed and plotted in Chapter 5. The thesis
concludes with a summary of the results.
The IEEE 802.11 standard defines MAC and a PHY layer protocols for wireless digital data transmission in the 2.4, 3.6 and 5 GHz frequency bands. Its target application is WLAN access between wireless, including mobile, terminals and between wireless terminals and a fixed network infrastructure or connection-oriented backbone. Popular 802.11 standards include 802.11a, 802.11b and 802.11g. The IEEE 802.11 standard defines a physical layer and a MAC layer; four different technologies are used as an air interface at PHY for contention-based and contention-free access control: infrared, frequency hopping, direct sequence spread spectrum and orthogonal frequency division multiplexing. The MAC layer manages and maintains communications between IEEE 802.11 stations (radio network cards and AP) by coordinating access to a shared radio channel as well as utilizing protocols that enhance communications over a wireless medium. It uses the physical layer to perform the tasks of carrier sensing (to check if the medium is free), transmission, and reception of 802.11 frames.

The IEEE 802.11 standard initially specified two coordination mechanisms: distributed coordination function (DCF) and point coordination function (PCF). In the IEEE 802.11 standard, the DCF mode is defined for asynchronous data transmissions, while the PCF mode is used to support time-bounded data transfer such as voice or
video transmission. This PCF mode is used in wireless LAN networks where an access point is present. However, the IEEE 802.11 MAC algorithm is unable to support modern multimedia applications which require a certain level of QoS guarantees in terms of consistent, in time and reliable data transfer. This lack of QoS support was a big hurdle in the evolution of multimedia applications over IEEE 802.11 networks. Therefore, to enable QoS support in the IEEE 802.11 networks, an enhanced version of the 802.11 was proposed. This enhanced version was named 802.11e. The access mechanism of 802.11e is referred to as enhanced distributed channel access (EDCA). The EDCA replaces the distributed coordinated function access mechanism of IEEE 802.11. The EDCA assigns traffic priorities based on the QoS requirements of the traffic, and for each priority, uses a different set of medium access parameters to support QoS requirements. In addition to EDCA mode IEEE 802.11e also defines another access mechanism called hybrid coordination function (HCF). The HCF is a centralized co-ordination function that combines the aspects of DCF and PCF with enhanced QoS mechanisms to provide service differentiation. In this chapter we discuss the important functions specified in IEEE 802.11 standards.

2.1 Introduction to MAC layer

The IEEE 802.11-based WLANs use a MAC mechanism known as CSMA/CA. CSMA/CA is based on listen-before-talk mechanism. The transmitting station senses the medium for a carrier signal and waits until the channel is available before transmitting. In a wired Ethernet, a node is able to sense a collision in the medium by detecting the increase in signal level during a collision. However, IEEE 802.11 wireless stations do not have this capability, and thus medium access mechanisms must have the capability of avoiding collisions.

Networks based on IEEE 802.11 standards can be deployed using three different
2.1.1 Independent Basic Service Set

An independent basic service set (IBSS) consists of a group of IEEE 802.11 stations communicating directly with one another. An IBSS is also referred to as an ad hoc network because it is essentially a simple peer-to-peer WLAN. Figure 2.1 illustrates how two stations equipped with IEEE 802.11 network interface card (NIC) can form an IBSS and communicate directly with one another. This is used for ad hoc/mesh network purposes.
2.1.2 Basic Service Set

A basic service set (BSS) is a group of IEEE 802.11 stations communicating with one another. A BSS requires a specialized station known as an AP which is the central point of communications for all stations in a BSS. Client stations do not communicate directly with other client stations, but use a common AP through which frames are routed to their destination. Sometimes the AP is equipped with an uplink port that connects the BSS to a wired network. Figure 2.2 illustrates a typical BSS.

In CSMA/CA, a node has to first listen to the channel for a predetermined amount of time to determine whether or not another node is transmitting on the channel within its wireless range. If the channel is sensed *idle*, the node is allowed to begin transmission. If the channel is sensed *busy*, the node defers its transmission for a random period of time known as the *back-off* period. A node’s transmission is considered successful if it successfully receives an ACK frame from the destination. If the ACK is not received the transmitting node retries transmission after back-off.
2.1.3 Extended Service Set

Multiple infrastructure BSS can be connected via their uplink interfaces. In an IEEE 802.11 network, the uplink interface connects the BSS to the distribution system. The collection of BSS interconnected via distribution system is known as extended service set (ESS). Figure 2.3 shows a practical implementation of an ESS. The uplink to the distribution system does not have to be through a wired connection but according to IEEE 802.11 specification this can be wireless too.

2.2 Distributed Coordination Function

The IEEE 802.11 implementation of CSMA/CA is manifested in the DCF [1]. It is important to describe some key IEEE 802.11 CSMA/CA components:

- Carrier Sense
• Distributed Coordination Function

• Acknowledgement Frames

• RTS/CTS medium reservation

In addition, two other mechanisms are used in IEEE 802.11 medium access but, are not directly tied to CSMA/CA:

• Frame fragmentation

• Point coordination function

2.2.1 Carrier Sense

A station that wants to transmit on the wireless medium must sense whether the medium is in use or not. If the medium is in use, the station must defer its frame transmission until the medium becomes idle. The station determines the state of the medium using the following two methods:

• Check PHY to see whether a carrier is present.

• Use the virtual carrier-sense function also known as the network allocation vector (NAV).

NAV is a virtual carrier sensing mechanism that may be thought of as a counter. When the counter is zero, the virtual carrier sense indication is that the medium is idle; when non-zero, the indication is busy. For example, in an infrastructure BSS (see Figure 2.4), suppose Kate is sending a frame to Nish, because the wireless medium is a broadcast-based shared medium, Paola also receives the frame. The MAC layer fram header contains a duration field that specifies the transmission time required for the frame, in which time the medium will be busy. As Paola is listening on the
wireless medium she reads the duration field and sets her NAV, which is an indicator for her to determine the time to defer her access to the medium. As time progresses, NAV is decremented until it reaches zero, at which Paola can resume her medium access attempt.

NAV is updated only when the duration field of the current transmission is greater than the current NAV. For example, if Paola has a NAV of 10 ms, she will not update her NAV unless the duration is greater than 10 ms.

In DCF operation, a station must wait a specific amount of time after the medium becomes available before attempting medium access. This time duration is known as the distributed interframe space (DIFS). Once the DIFS interval elapses, the medium becomes available for contention.

Suppose Paola and Nish have data ready to be transmitted but are waiting for Kate’s transmission to complete. If their current NAV is same they will detect an idle channel together, at the same time, and attempt transmission resulting in a collision. To avoid this situation, DCF uses a random back-off timer through which they access the medium at different times. In Figure 2.5, Kate and Paola’s status is illustrated.
Paola has frozen her back-off counter until Kate completes her *two way handshake*.

The random back-off algorithm randomly selects a value from the range \( \{0, \text{CW}\} \) known as contention window. The default CW values vary by vendor and are stored in the station NIC. The upper value of the contention window exponentially increases upon failure to access medium with a minimum value of \( \text{CW}_{\text{min}} \) and a maximum of \( \text{CW}_{\text{max}} \). Figure 2.6 illustrates the \( \text{CW}_{\text{min}} \) and \( \text{CW}_{\text{max}} \) values for binary random back-off. A station randomly selects a value between 0 and the current CW. The random value selected determines the number of IEEE 802.11 slot times the station must wait before it can transmit. A slot time is a time value derived from the PHY based on RF characteristics of the BSS. For example, if Paola’s NAV has decremented to 0, and the PHY also indicates the medium is idle, she selects a random back-off time between 0 and contention window (CW) (in this case, CW is 7) and waits the selected number of slot times before transmitting. Figure 2.6 illustrates this process, with a random back-off value of four slot times. Once the four slot times elapses, Paola can transmit. If Nish’s station has a random back-off time of two time slots, Paola hears a new duration from Nish’s frame when he begins his transmission, and Paola updates her NAV with that new value. Paola must wait for her NAV to decrement to
0 and her PHY to report that the medium is available again before she can resume her back-off.

Assuming that Paola is able to defer transmission for all four slot times, she transmits the frame. The IEEE 802.11 specification requires that the receiving station must send an acknowledgement (ACK) frame to the transmitter. This ACK frame allows the sending station to indirectly determine if a collision has occurred. If the sending station does not receive an ACK frame, it assumes that a collision has occurred and then in an attempt to retransmit the packet updates its retry counters, increases the contention window range by doubling the CW value, and begins the medium access process again. Figure 2.7 summarizes the steps a IEEE 802.11 DCF station must iterate through to transmit a frame.
2.2.2 The Acknowledgment Frame

A station receiving a frame acknowledges error-free receipt of the frame by sending an ACK frame back to the sending station. ACK frames are allowed to skip the random back-off process and wait a short interval after the frame has been received. The short interval the receiving station waits is known as the short interframe space (SIFS). The SIFS interval is shorter than a DIFS interval by two slot times.

2.2.3 Hidden node and RTS/CTS

The hidden node problem occurs when a node A is visible from node B, but not from node C communicating with said node B. If the data to be transmitted is too large between node C and node B, collisions due to A’s transmission might result much of the channel bandwidth being wasted. The optional RTS/CTS access mode involves a four-way handshaking technique in which the sender first sends the RTS to reserve
the channel before its transmission, and upon receiving CTS from the receiver, packet transmission and ACK response procedure takes place. The header of RTS and CTS and data frames include the duration information which is the amount of time for which the wireless medium is to be reserved for transmission of data and returning ACK frames after the end of the current frame. All the stations who gets this RTS, CTS and/or data frames will use this information to set their NAV, that represents the time it has to defer its access to the medium. But if we send the short RTS frame before transmitting actual data, even if a collision occurs we lose only the control message packet and much of the channel bandwidth can be saved. Therefore, the advantage of exchange of short RTS/CTS frames before the actual data transmission is to avoid long collisions and also the amount of bandwidth that gets wasted in collisions. This four way handshake, as illustrated by Figure 2.8, is not suitable for real time voice traffic because it cause large end to end delay, thus affecting QoS.

### 2.2.4 Frame Fragmentation

Frame fragmentation is a MAC layer function that is designed to increase the reliability of frame transmission across the wireless medium. The premise behind fragmentation is that a frame is broken up into smaller fragments, and each fragment is transmitted individually. A network administrator can define the fragmentation size in an AP.

### 2.3 Point Coordination Function

A PCF is an IEEE 802.11 optional medium access mechanism that is used in addition to DCF. PCF is an access mechanism that provides contention-free frame delivery to and from the AP. Most vendors do not include PCF support because it increases the
Figure 2.8. RTS/CTS operation
protocol overhead of the BSS. As a result, it is not widely deployed.

2.4 Enhanced Distributed Coordination Function

The IEEE 802.11e is an extension of the IEEE 802.11 WLAN standard with provisioning for QoS. The new standard provides the means of prioritizing radio channel access within the infrastructure of a BSS. IEEE 802.11e introduces enhancement to DCF known as EDCA. Each station has eight traffic categories, or priority levels. Using EDCF, stations contend for channel access after waiting for a period of time (AIFS) defined by the corresponding traffic category. Traffic of higher priority will have a shorter AIFS than those of lower priorities.

Each access category (AC) has its own queue and channel access parameters, which include arbitrary interframe space (AIFS), \(CW_{\text{min}}\) and \(CW_{\text{max}}\) sizes and transmission opportunity (TXOP) limits. The AC can gain different priority for channel access by differentiating the parameters. Furthermore, each AC executes an independent back-off process to transmit its frames. The lower AIFS/\(CW_{\text{max}}/CW_{\text{min}}\) result in the higher probability of winning the channel contention [11].

The AIFS is at least DIFS, and can be enlarged individually for each AC. The arbitrary interframe space number (AIFSN) means AIFS number, which is a integer greater than zero, and SIFS is the time interval used for a CTS frame, MAC protocol data unit (MPDU), and an ACK frame. The Figure 2.9 shows the basic access access mechanism in enhanced distributed coordination function (EDCF).

Moreover, slot time is the constant number for a slot of back-off process. After waiting for AIFS, each back-off process is set. AIFS value for each AC is decided as following [12]:

\[
AIFS[AC] = SIFS + AIFSN[AC] \times SlotTime,
\]

(2.4.1)
The positions and sizes of the contention windows relative to each other, as defined per AC by the EDCA parameter set, are important factors to define relative priority in medium access per AC. The contention window increases upon unsuccessful frame exchanges, but never exceeds the value of \( CW_{max}[AC] \). This parameter is defined per AC as part of the EDCA parameter set. The smaller the \( CW_{max}[AC] \), the higher the medium access priority. However, a small \( CW_{max}[AC] \) may increase the collision probability. Furthermore, it should be highlighted that there are retry counters (similar to legacy IEEE 802.11) that limit the number of retransmissions. The IEEE 802.11e protocol also defines a maximum MSDU lifetime per AC, which specifies the maximum time a frame may remain in the MAC.
2.4.1 Transmission Opportunity

An IEEE 802.11e station (more precisely, a back-off entity) that obtains medium access must not utilize radio resources for a duration longer than a specified limit. This important new attribute of the IEEE 802.11e MAC is referred to as a transmission opportunity (TXOP). A TXOP is an interval of time during which a back-off entity has the right to deliver MAC service data units (MSDUs). A TXOP is defined by its starting time and duration. TXOPs obtained via contention-based medium access are referred to as enhanced distributed channel access - transmission opportunity (EDCA-TXOP). The duration of an EDCA-TXOP is limited by a parameter referred to as TXOPlimit. TXOPlimit is distributed regularly by the HC within an information field of the beacon.

2.5 Physical Layer

The IEEE 802.11 PHYs essentially provides wireless transmission mechanism for the MAC, in addition to supporting secondary functions such as, assessing the state of the wireless medium and reporting it to the MAC.

In IEEE 802.11b/e for 1 Mbps and 2 Mbps, the data is direct sequence spread spectrum (DSSS) modulated using an 11-bit Barker code whereas, in 5.5 Mbps and 11 Mbps operation IEEE 802.11b uses complementary code keying (CCK) to modulate data at a higher data rate. A shorter spreading sequence in used in CCK which reduces the spreading compared to 11-bit Barker spreading and hence results in an increased data rate [13].
3.1 Overview

In simulating packet voice networks, it is very important to consider the characteristics of voice sources used to generate packets. The accuracy of packet generation, directly impacts the evaluation of the voice call carrying capacity of the network. A source generating voice packets has talk-spurts (or active periods) and silent-spurts (inactive periods). Usually, for purposes of generating VoIP traffic a simple on-off model is used, where the on (talk-spurt) and off (silent-spurt) periods of a user at one end of a two way voice call is assumed to be independent of the user at the other end. However in a real conversation, the transition between the on and off periods of users is dependent upon one another. A model to generate on-off speech patterns, considering this dependence, was first proposed by P T Brady [14, 15, 16]. Present literature available in modelling on-off voice patterns, [10, 17, 18], are extensions of Brady’s model [16].

3.2 Voice Model

The model described in this chapter reproduces the on-off temporal characteristics of human conversational speech for characterizing speech processing systems which
have speech detectors, such as loudspeaker telephones, echo control devices, digital circuit multiplication equipment (DCME), packet systems, and asynchronous transfer mode (ATM) systems. This model reflects parameters of human conversation such as the length of the talk-spurt, pause, double talk, and mutual silence.

3.2.1 Six State Model

In human conversations, the durations and rates of talk-spurt and pause vary according to measurement conditions. In [10], the voice model proposed has talk-spurts, and silent-spurts based on the transition blocks shown in Figure 3.1. $P_1$, $P_2$, and $P_3$ described in the model, denote transition probabilities expressed in percent.

The artificial voice described in the ITU-T50 specifications [19] is generated during a talk-spurt. As per the specifications of [10], if the pause duration is less than 200 ms, the model chooses either a single talk or mutual silence state with probabilities of
50% until the pause duration exceeds 200 ms. This condition results in formation of two more pause states, PAUSE1 and PAUSE2, as shown in Figure 3.2. Hence in our simulations, a six state semi-Markov model (Figure 3.2), derived from [10], is used for generating voice packet time stamps and is used as the voice packet generator in our simulations.

In this model, the on-off speech patterns generated are exponentially distributed and the duration spent in single talk ($T_{st}$), double talk ($T_{dt}$), mutual silence ($T_{ms}$)
and pause \((T_p)\) states are given by the following equations.

\[
T_{st} = -0.854 \log_e (1 - x_1), \tag{3.2.1}
\]

\[
T_{dt} = -0.226 \log_e (1 - x_2), \tag{3.2.2}
\]

\[
T_{ms} = -0.456 \log_e (1 - x_3), \text{ and} \tag{3.2.3}
\]

\[
T_p = -0.456 \log_e (1 - 0.3551 x_4), \tag{3.2.4}
\]

where \(x_1, x_2, x_3, x_4 \in \{0, 1\}\) are independent random variables with uniform distribution. The time in these equations is expressed in seconds. The transition probabilities for this model are \(P_1 = 0.4, P_2 = 0.5, P_3 = 0.5, P_4 = 0.5, P_5 = 0.5, P_a = 0.5, \) and \(P_b = 0.5.\)

It is assumed that the two users, A and B, always exist in one of the six possible states as shown in Figure 3.2. It is also assumed that, the two users will never enter silent mode or talk mode simultaneously. There are no diagonal transitions between single talk and mutual silence or between double talk and mutual silence. From this, the state transitions can be described in the form of a matrix as shown below:

\[
P = \begin{bmatrix}
0 & 0.2 & 0.2 & 0 & 0 & 0.6 \\
0.5 & 0 & 0 & 0.5 & 0 & 0 \\
0.5 & 0.5 & 0 & 0 & 0 & 0 \\
0 & 0.2 & 0 & 0 & 0.2 & 0.6 \\
0 & 0.5 & 0 & 0.5 & 0 & 0 \\
0.5 & 0 & 0 & 0.5 & 0 & 0
\end{bmatrix}.
\]

Using this state transition matrix and the following equations,

\[
\pi = \pi P \tag{3.2.5}
\]

\[
\pi e = 1. \tag{3.2.6}
\]

Limiting probabilities were found to be equal to

\[
\pi_{theory} = \begin{bmatrix}
0.2381 & 0.1428 & 0.0476 & 0.2381 & 0.0476 & 0.2857
\end{bmatrix}.
\]
During simulation, the limiting probabilities were calculated as the ratio of the total number of transitions to each state to the total number of transitions occurred. The values obtained through simulations are as follows:

\[
\pi_{\text{simulation}} = \begin{bmatrix} 0.2350 & 0.1420 & 0.0452 & 0.2400 & 0.0520 & 0.2860 \end{bmatrix}.
\]

We see that the simulation values obtained, after sufficient number of iterations, tend towards the theoretical values of limiting probabilities. The talk spurts, mutual silence and pause states of the model, if plotted over a period of time, resemble as shown in Figure 3.3. Table 3.1 lists the values of Temporal parameters of six state voice model.

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Rate(%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Talk Spurt</td>
<td>51.50</td>
</tr>
<tr>
<td>Pause</td>
<td>10.20</td>
</tr>
<tr>
<td>Mutual Silence</td>
<td>16.00</td>
</tr>
<tr>
<td>Double Talk</td>
<td>21.20</td>
</tr>
</tbody>
</table>

Now to calculate the proportion of time spent in a state \( i \) \( (P_i, \text{normalized time} \)}
spent in a state), we can use the following equation:

$$P_i = \frac{\pi_i E[\tilde{t}_i]}{\sum_{j=1}^{6} \pi_j E[\tilde{t}_j]}.$$  \hfill (3.2.7)

Where $E[\tilde{t}_j]$ is used to denote the average amount of time spent in a state $j$. Theoretical calculation and simulation results for $P_i$ are as follows:

$$P_i\{\text{Theory}\} = \begin{bmatrix} 0.3707 & 0.1180 & 0.0111 & 0.3707 & 0.0111 & 0.1178 \end{bmatrix},$$

and

$$P_i\{\text{Simulation}\} = \begin{bmatrix} 0.3802 & 0.102 & 0.0091 & 0.3605 & 0.0131 & 0.1203 \end{bmatrix}.$$

We observe that with sufficient number of iterations the simulation results tend towards theoretical values. The Java code for the simulation of this six state semi-Markov model can be found in Appendix A.1. Figures 3.4 and 3.5 show the cumulative distribution of talk spurt and silent gap random variables obtained from simulation of conversational speech.

### 3.2.2 G 723.1 Voice Codec

To model a voice codec, we use the G 723.1 (at 5.3 kbps) codec [20] for packet creation. To simulate packet generation from the voice spurts we break up the voice activity into frames. In this case the voice packet traces are generated in 30 ms intervals for single talk and double talk using the six state semi-Markov model. The voice payload size is 20 bytes as per the codec specifications.
Figure 3.4. Cumulative distribution of duration of talk spurt from of conversational speech as measured by simulation

Figure 3.5. Cumulative distribution of duration of silence gap from of conversational speech as measured by simulation
Chapter 4

Simulator and Simulation Setup

4.1 Introduction

In the previous chapter, we presented a detailed description of the voice model used for our simulations. In this chapter, we discuss the ad hoc network simulator used for evaluating the upper-bound of call carrying capacity of the network. We begin with an overview of the simulator and later discuss the algorithm used in our simulation model. We conclude with a description of the setup used for the simulations.

4.2 Simulator Overview

A Java based IEEE 802.11e link layer simulator was used for simulations to determine the call carrying capacity. The simulator can be classified as a discrete event simulator as it consists of a set of discrete events, and a simulator object, which executes these events, in the order of occurrence. An event is defined as a broadcast of a packet from a node at a particular point in time. The simulator essentially consists of a node, with features that enable it to be a part of a multi-hop network. We create a network by creating logical connections between different nodes. We then schedule events for transmission of packets that have to be broadcast to a particular destination. The scheduled events are enqueued in an event queue of the simulator. Since we have
voice packets with MPDU size less than 100 bytes we use DCF without RTS/CTS to avoid any packet overhead at MAC layer.

4.3 Link Layer Simulation Algorithm

At the IEEE 802.11 link layer, the flow of a packet is as shown in Figure 4.1. A source node having a voice packet to send, inserts the packet into the network event queue. If the NAV of that particular node is 0, it broadcasts the packet after the back-off expires. The node then waits for an AIFS period to sense any current ongoing transmission in the channel. If it senses a transmission during its back-off period, it resets its NAV to the duration time of the transmission and freezes its back-off. In case of collision, the packet is re-transmitted and the duplicate MPDU packets are discarded. ACKs are transmitted only once for an MPDU. MPDUs can be rescheduled/retransmitted 10 times as per IEEE 802.11 specifications [1]. The flow in Figure 4.1 describes the operations of the IEEE 802.11 link layer simulator without RTS/CTS.
Figure 4.1. Packet flow in link layer simulator
4.3.1 Transmitted Power

For the ideal physical channel scenario, we assume that the physical channel is perfectly lossless at a distance of one hop and the received power is zero at a distance greater than one hop. The loss of packets in this case will purely be from collisions and delay in the MAC layer. This approach helps us in quantifying the loss due to MAC layer protocol restrictions and will thus provide a better understanding of the upper-bound of achievable capacity.

For the non-ideal channel scenario we assume that all nodes in the network transmit at maximum power. According to [21] the maximum transmission power is 90 mW.

4.3.2 Path Loss Model and Packet Error Rate

In a wireless communication system, the transmitted signal is affected by a channel that varies in space and time. These variations introduce impairments such as attenuation, multipath, linear distortion and noise in the transmitted signal. Three different and mutually independent propagation phenomena influence the power of the received signal: path loss, shadowing and multipath fading [22, 21]. For our simulations we consider the commonly used two-ray Rayleigh flat fading model described by the equation below.

\[ h_b(t) = \alpha_1 \exp(j\phi_1)\delta(t - t_1) + \alpha_2 \exp(j\phi_2)\delta(t - t_2), \quad (4.3.1) \]

where \( t_1 \) and \( t_2 \) denote the relative delays of the individual paths. We then compute the path loss of signal due to large scale path loss and fading. We use equation (4.3.2) from [22], where the path loss exponent is \( n = 3.1 \).

\[ \text{PL}(d) = \text{PL}(d_0) + 10 \times n \times \log_{10}(d/d_0) + \tilde{\chi}_\sigma + \tilde{\gamma}_{(dB)}, \quad (4.3.2) \]
where $PL(d_0)$ is the free space path loss at a distance of one meter, $d$ is the distance between 2 nodes and $\tilde{\chi}_\sigma$ is the zero mean Gaussian random variable and $\tilde{\gamma}_{(dB)}$ is the fading. The BER is then obtained from a lookup table in [21]. For calculating PER, we assume that bit errors are independent of each other. The probability of packet error is given by equation (4.3.3),

$$\text{PER} = 1 - (1 - \text{BER})^m,$$

where $m$ is the packet size in bits.

### 4.3.3 Node States in MAC

A node in the network toggles between two states.

- **Send**

- **Receive/Wait**

```java
START
1 frame = source.getMPDUvocoder();
2 destination = source.getMPDUDest();
3 if (!sendBuffer.isEmpty() || MPDUToSend != null)
4 {
5     sendBuffer.insert(frame, destination);
6     return;
7 }
8
9 MPDUToSend = frame;
10 targetToSnd = destination;
11 if (!nav.isEmpty())
12 {
13     return;
14 }
15 node.BackOffExpired();
16 setTimeout(ACKFrame);
17 node.sendUnicast(MPDUToSend, targetToSnd);
18
STOP
```

Listing 4.1. Algorithm describing the Send state of a node.
In our simulations, the voice packet source trace files are generated using the six state semi-Markov voice model described in the previous chapter. In the **Send** state, a node preparing to transmit gets the packet from the packet source and checks to see if the send buffer at the transmit end is free. If the buffer is free the packet is inserted into wait buffer and an MPDU is transmitted once the back-off expires. The pseudocode in Listing 4.1 describes the algorithm implementing the **Send** state of the node.

In the **Receive/Wait** state, a node receives a packet from the channel. The pseudocode described in Listing 4.2 implements the **Receive/Wait** state of a node. To include the impact of physical layer conditions on the packet, the boolean value of **PHYCorrupt**, returned by the PER evaluation of equation (4.3.3), is used to decide whether a packet has to be discarded or not.

The `handleNetworkLayerReceive()` function is used to handle incoming packets appropriately. If the packet received is an ACK, the `handleACK()` function handles it whereas, an MPDU packet is handled by the function `handleMPDU()`. Since we are using static routes on a linear network, re-routing is done by replacing the destination address in the packet and forwarding it.

### 4.3.4 Simulation Setup

In our simulation setup, to obtain an upper-bound for the number of voice calls supported we use linear networks with an ideal channel and zero routing overhead. We consider one, two, three and four hop scenarios. Distance between two consecutive nodes in our linear network is considered to be 100 meters. We use static routes for packet routing across the network. This setup is chosen as it will provide us the upper-bound of call carrying capacity of a wireless link in a multi-hop scenario with minimal collisions. For each simulation run we gradually increase the number of
calls until we obtain an upper bound. We also perform simulations to evaluate the call carrying capacity using a path loss model. Finally, we propose packet dropping policies at the nodes to reduce network traffic across the link and evaluate its impact.

```
START

if(PHYCorrupt==true)
{
    return;
}
else
{
    if(destination==MACAddress && !nav.isEmpty())
    {
        return;
    }
    setNAV();
    stopBackoff();
    if(destinationAddress==0 && frame==MPDU)
    {
        handleNetworkLayerReceive(Packet);
        return;
    }
    if(destinationAddress!=MACAddress && frame==MPDU && isPromiscuous)
    {
        handleNetworkLayerReceive(Packet);
        return;
    }
    if(destinationAddress!=MACAddress)
    {
        return;
    }
    handleNetworkLayerReceive(Packet);
    switch(packetType)
    {
        case MPDU:
            handleMPDU();
        case ACK:
            handleACK();
    }
STOP
```

**Listing 4.2.** Algorithm describing the Receive/Wait state of a node.
Chapter 5
Simulation Results and Observations

In this chapter, we discuss the methods used to analyse the simulation output and we present the results obtained from the simulations. The simulation results were obtained for linear networks consisting of 1, 2, 3 and 4 hops with varying number of voice calls. We conducted simulation tests with varying parameters like the transmission and interference range of the nodes, data rate of 5.5 Mbps under ideal path and path loss scenarios. We also proposed packet dropping policies to reduce network traffic and evaluate the impact on the QoS. We carry out all our simulations without using the RTS/CTS mechanism to avoid overhead, as the MPDUs are all less than 100 bytes.

For the node traffic input, the voice packet trace generated from voice model to mimic the G 723.1 codec. It is a constant bit rate (CBR) stream of time stamps stored and read out of the file during simulations. In this thesis, for all the simulations, the output trace is written to a text file. It consists of all the information about when, where and how many packets were sent in the network throughout the run time of the simulation. The text file is generated for each pair of nodes. It consists of four columns; Where the first column contains the packet number, the second column contains packet numbers generated by the vocoder, third column contains
the transmission times and the fourth column gives us the reception time. The name of the file contains the information about source and destination of the packet. All this information is written to the trace file periodically for all the simulation time. These trace files are then used for analysis of the simulated data and graphs of average end-to-end delay of packets and other parameters associated with the quality of the network.

5.1 Metrics of Interest

Real-time traffic transmitted over multiple wireless links are evaluated using latency and packet losses as metrics. In our simulations, one way packet loss of 1% is considered acceptable. Packet loss here is defined as a sum of packets loss due to PHY noise and packets delayed by the MAC layer. We shall discuss our simulation runs in relation to the above mentioned parameters in this chapter.

The results are noted to see the maximum call carrying capability of different sizes of linear network. ITU-T G.114 [23] recommends a maximum of a 150 ms one-way latency for voice calls, so we consider that packets whose end-to-end delay is more than 150 ms as lost. One way packet loss of 1% is considered acceptable [6]. For all the simulation results using IEEE 802.11e MAC, we use a default data rate of 5.5 Mbps with a send buffer size of 100 packets. We do not use RTS/CTS mechanism to transmit packets. This is due to the size of the voice payloads, which are 20-30 bytes for which RTS/CTS would cause significant delay and overhead. In almost all commercial cards, RTS/CTS mechanism is turned off by default and are only activated when the size of a packet cross a preset threshold.
5.2 Upper-Bound of Call Carrying Capacity

In our first set of simulations we determine the upper-bound of call carrying capacity of linear networks of size one, two, three and four can support at 5.5 Mbps. Figures 5.1 and 5.2 show the survivor functions of number of simultaneous voice calls supported by one hop and a three hop linear network, respectively, satisfying the delay and dropped packet constraints defined in [6].

Figure 5.3 depicts the relationship between the total packets lost and the number of calls. We can see from the graph that as the number of calls increase, the QoS deteriorates rapidly. The maximum number of calls a one hop and a three hop linear network can support simultaneously at 5.5 Mbps is 8 and 5 voice calls respectively.

Figure 5.4 summarizes the relationship between hop count and number of calls supported under ideal physical channel conditions for linear networks of size one, two, three and four hops.
5.3 Packet Dropping Policies and Call Carrying Capacity

In the previous section we evaluated the upper-bounds of linear networks of different sizes. In this section we propose simple packet dropping policies and also evaluate the performance of wireless multi-hop links under path loss. The primary goal of the packet dropping policies is to reduce traffic across a multi-hop link, while improving QoS of real-time traffic.

For our simulations we pick two types of packet policies.

- Random packet dropping policy.
- Time delay based packet dropping policy.

In the random packet dropping policy, a node randomly drops 0.1% of all packets it receives. The goal of this policy is to reduce the traffic carried on a link. This in
Figure 5.3. Relation between packet loss and the number of calls supported in a three hop 5.5 Mbps wireless link under ideal channel conditions and minimal collisions.

Figure 5.4. Upper-bound of simultaneous voice calls supported by a 5.5 Mbps wireless link under ideal channel conditions and minimal collisions as a function of hop count.
turn reduces the link latency as there are less packets waiting to be transmitted across the link. In the time delay based packet dropping policy, the goal is to preemptively discard packets which have a high probability of being delayed across the link. The node picks a packet to discard based on the delay threshold specified on the node. If the difference between the time stamp on the packet and the current time on the node crosses a certain time threshold \((t_d)\) it discards the packet. The time delay packet dropping policy that we employed in our simulations dropped packets that were delayed for more than 80 ms at the end of their first hop transmission, or 100 ms at the end of their second hop transmission.

It can be seen through Figure 5.5 that there is a loss of 10.12% of total packets transmitted without adopting any packet dropping policy. When a random packet dropping policy is used the packet loss reduces to 8.86% but by adopting a time delay based packet dropping policy the packet loss goes down to 3.53%. By adopting a time delay based policy the QoS improves significantly. In both the cases, while the network may not support more than 5 calls (the three hop upper bound), the overall degradation of QoS is less rapid, as observed from the plot of the survivor function in Figure 5.5.

In our simulations we also considered a four hop linear network (Figure 5.6) in path loss conditions and evaluated the performance. As we can see from the graph of survivor functions, the overall QoS of the link increases when packets are preemptively dropped. In the four hop case we see up to 20% decrease in overall packet loss, when using the random packet drop policy and up to 50% decrease in the case of time delay based packet drop policy. Table 5.1 compares the delay and packet loss metrics obtained after introducing dropping policies in the link, against the regular linear network performance metrics.
Figure 5.5. Comparison of survivor functions of packet delays in a three hop 5.5 Mbps linear network carrying six voice calls under ideal physical channel conditions.

<table>
<thead>
<tr>
<th>Packet Delayed (%)</th>
<th>Packets Dropped (%)</th>
<th>Total Packet Loss (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>8.3943</td>
<td>1.1033</td>
<td>9.4976</td>
</tr>
<tr>
<td>6.1462</td>
<td>0.9364</td>
<td>7.8266</td>
</tr>
<tr>
<td>2.2311</td>
<td>1.8225</td>
<td>4.4536</td>
</tr>
</tbody>
</table>

Table 5.1. Impact of packet dropping policies on a four hop 5.5 Mbps linear network under path-loss channel conditions.

Figure 5.8 summarizes the call carrying capacity of linear networks of different sizes, both with and without taking into account packet dropping policies and channel conditions.
Figure 5.6. Four hop linear network topology

Figure 5.7. Comparison of survivor functions of packet delays in a four hop 5.5 Mbps linear network carrying three voice calls under path loss channel conditions.
Figure 5.8. Capacity of wireless multi-hop links under both ideal and path-loss channel conditions, with and without packet dropping policy.
Chapter 6

Conclusions and Future Research

In this thesis, we evaluated the call carrying capacity of an IEEE 802.11 based ad hoc network. In this chapter, we summarize the major results obtained through simulations and propose further research directions.

First, we thoroughly understood the working procedure of the IEEE 802.11 standard that is being used for the medium access in our simulations. We proposed and modelled a six state semi-Markov voice conversation model for generating voice packets and used a Java based discrete event simulator to evaluate the voice call carrying capabilities of an IEEE 802.11 based ad hoc network. We performed simulations to determine the upper-bound of a wireless link’s call carrying capacity using a linear network topology of one, two, three and four hops. We compared the call carrying capacities of the ad hoc network by performing simulations under two different packet dropping schemes.

We found that for one, two, three and four hop, a 5.5 Mbps IEEE 802.11 wireless link has an upper-bound of eight, six, five, and three voice calls respectively. We also found that by adopting packet dropping policies at the nodes, we improve the call carrying capacity and quality of service on the network. In our simulations of a two hop network in path loss conditions, we observe that, by adopting a time delay based packet dropping policy at the nodes, the number of calls supported simultaneously
increased from five to six. In a four hop linear network we find that by using a random packet dropping policy the total packet loss is reduced by 20% and it increases by 50% when a time-delay based packet dropping policy is considered, however there is no increase in the number of calls supported by the network. This is because dropping packets whose delay would render them useless at the destination reduces network load without reducing call quality.

Further studies should be done to analyze the impact of using packet dropping policies in other types of networks supporting real-time traffic. Regardless of the type of network or system used, it appears that a suitable packet dropping policy can lead to improvement in capacity and QoS.
Bibliography


Appendix A

Code Listing for Six State Semi-Markov Voice Model

This appendix provides a listing of the Java code for the voice model used in determining the upper bound on the voice-call carrying capacity of an IEEE 802.11 based ad hoc network. The model itself is based on the ITU-T P59 standard [10].

```
import java.io.FileWriter;
import java.io.IOException;
import java.util.Vector;

public class ITUVoiceModel_P59 {
    double ASum = 0, BSum = 0, CSum = 0, DSum = 0, ESum = 0, FSum = 0, GSum = 0;

    public ITUVoiceModel_P59() throws IOException {
        double ConTiLine = 0;
        double a = 0, b = 0;
        double p1, p2, p3, p4, pa, p5, pb, temp1 = 0;
        p1 = Math.random();
        p2 = Math.random();
        p3 = Math.random();
        pa = Math.random();
        p4 = 2;
        p5 = 2;
```

int PrevCase = 0, CC1 = 0, CC2 = 0, CC3 = 0;
int CC4 = 0, CC5 = 0, CC6 = 0, CASE = 0;
double ToTime = 0;
boolean x;
CASE = 6;

while (ConTiLine < 500000) {
    switch (CASE) {
    case 1: {
        temp1 = (-.854 * Math.log(1 - Math.random())) * 1000;
        ConTiLine = ConTiLine + temp1;
        PrevCase = CASE;
        break;
    }
    case 2: {
        temp1 = (-.456 * Math.log(1 - Math.random())) * 1000;
        ConTiLine = ConTiLine + temp1;
        PrevCase = CASE;
        break;
    }
    case 3: {
        temp1 = (-.456 * Math.log(1 - (0.3551 * Math.random()))) * 1000;
        ConTiLine = ConTiLine + temp1;
        PrevCase = CASE;
        break;
    }
    case 4: {
        temp1 = (-.854 * Math.log(1 - Math.random())) * 1000;
        ConTiLine = ConTiLine + temp1;
        PrevCase = CASE;
        break;
    }
    case 5: {

temp1 = (-.456 * Math.log(1 - (0.3551 * Math.random()))) * 1000;
ConTiLine = ConTiLine + temp1;
PrevCase = CASE;
break;
}
case 6: {
    temp1 = (-.226 * Math.log(1 - Math.random())) * 1000;
    ConTiLine = ConTiLine + temp1;
    PrevCase = CASE;
    break;
}
default: {
    System.out.println("DID NOT ENTER ANY STATE");
    break;
}

if (CASE == 1) {
    CC1++;
}
if (CASE == 2) {
    CC2++;
}
if (CASE == 3) {
    CC3++;
}
if (CASE == 4) {
    CC4++;
}
if (CASE == 5) {
    CC5++;
}
if (CASE == 6) {
    CC6++;
}

ToTime++;

p1 = Math.random();
p2 = Math.random();
p3 = Math.random();
p4 = Math.random();
p5 = Math.random();
pa = Math.random();
pb = Math.random();

if ((p4 <= 0.5 && PrevCase == 3) || (p3 < 0.5 && PrevCase == 6)
    || (p2 < 0.5 && PrevCase == 2)) {
    CASE = 1;
}

if (((p1 - (p1 * pa)) >= 0.258 && (p1 - (p1 * pa)) < 0.4) && PrevCase == 1) ||
    (((p1 - (p1 * pb)) >= 0.258 && (p1 - (p1 * pb)) < 0.4) && PrevCase == 4) ||
    (p5 < 0.5 && PrevCase == 5) || ((1 - p4) < 0.5 && PrevCase == 3)) {
    CASE = 2;
}

if ((p1 - (p1 * pa)) < 0.258 && PrevCase == 1) {
    CASE = 3;
}

if (((1 - p2) <= 0.5 && PrevCase == 2)
    || ((1 - p3) <= 0.5 && PrevCase == 6)
    || ((1 - p5) <= 0.5 && PrevCase == 5)) {
    CASE = 4;
}

if ((p1 - (p1 * pb)) < 0.258 && PrevCase == 4) {
CASE = 5;
}

if (((p1) >= 0.6 && PrevCase == 4) || ((p1) >= 0.6 && PrevCase == 1)) {
    CASE = 6;
}
}

System.out.println ("AT , MS , P1 , BT , P2 , DT "
    +(float)(CC1/ToTime)+" , "
    +(float)(CC2/ToTime)+" , "
    +(float)(CC3/ToTime)+" , "
    +(float)(CC4/ToTime)+" , "
    +(float)(CC5/ToTime)+" , "
    +(float)(CC6/ToTime) );
}

public static void main(String args[]) throws IOException
{
    new ITUVoiceModel_P59();
}
Pradeep Gunda Bhat was born in Bangalore, India to Mr. A. G. Bhat and Mrs. Pushpa Bhat. He received a Bachelor of Engineering degree in Telecommunications from Visveswariah Technological University (VTU), Belgaum, India in 2003. In 2004, he enrolled into the Master of Engineering Science (Telecommunications) program at The University of Mississippi, University, MS. He is currently working as a Web Systems Engineer for Skolix, LLC.