QoS in Wireless Networks

Teresa Sheausan Tung

Electrical Engineering and Computer Sciences
University of California at Berkeley

Technical Report No. UCB/EECS-2006-152
http://www.eecs.berkeley.edu/Pubs/TechRpts/2006/EECS-2006-152.html

November 19, 2006

QoS in Wireless Networks

by

Teresa Sheausan Tung

B.S. (University of California, Berkeley) 2000
M.S. (University of California, Berkeley) 2003

A dissertation submitted in partial satisfaction of the
requirements for the degree of
Doctor of Philosophy

in

Engineering - Electrical Engineering and Computer Sciences

in the

GRADUATE DIVISION
of the
UNIVERSITY OF CALIFORNIA, BERKELEY

Committee in charge:

Professor Jean Walrand, Chair
Professor Pravin Varaiya
Professor J. George Shanthikumar

Fall 2006

The dissertation of Teresa Sheausan Tung is approved:

Chair Date

Date

Date

University of California, Berkeley

Fall 2006

QoS in Wireless Networks

by

Teresa Sheausan Tung

Abstract

QoS in Wireless Networks

by

Teresa Sheausan Tung

Doctor of Philosophy in Engineering - Electrical Engineering and Computer Sciences

University of California, Berkeley

Professor Jean Walrand, Chair

This thesis describes issues and solutions for providing service for applications requiring QoS guarantees in wireless networks. We focus on practical solutions that leverage existing wireless technologies and solutions for the wired network. We consider a range of wireless networks like the single domain case, the fixed multi-hop case, and the mobile ad-hoc case.

We perform an analytical study of the single domain WiFi network with a focus on the feasibility of providing QoS. Our approach extends the analytical model by Bianchi to consider more realistic network conditions such as non-saturated sources and rate adaptation. Our analysis finds that today's WiFi technology is not enough to guarantee QoS in the presence of elastic TCP traffic. We suggest solutions like an admission control scheme that are compatible with existing WiFi equipment.

The multi-hop and ad-hoc cases require additional consideration for routing. We describe a multi-path routing protocol designed for wired networks that may extend to a fixed multi-hop wireless network. We also study the use of clustering for routing over an ad-hoc network.

_____

Professor Jean Walrand
Dissertation Committee Chair

To my parents.

Contents

List of Figures

List of Tables

Acknowledgements

First, I would like to thank my advisor Professor Jean Walrand for his guidance and encouragement over the years. I am grateful for the opportunity to work as his student and hope to have also learned some of his patience and consideration. I also thank my dissertation committee members Professor J. George Shanthikumar and Professor Pravin Varaiya, and Professor Venkat Anantharam who served on my Qualifying Committee.

I would like to acknowledge the support of Ayman Fawaz and Raghuveer Chereddy from Siemens Technology-to-Business center in Berkeley: Ayman for providing the opportunity to work on the analysis of single domain WiFi and Raghuveer for his help in setting-up experiments.

Finally, I would like to express my gratitude to my fellow graduate students for their support, notably Eric Chi, Antonios Dimakis, Rajarshi Gupta, Linhai He, Paul Huang, Zhanfeng Jia, John Musacchio, Wilson So, Aaron Wagner.

Chapter 1

Introduction

The vision is of next generation networks that provide ubiquitous access for inelastic and elastic applications over a common wireless infrastructure. There is demand for such access as proven by the success of wireless technologies like cellular and DECT for carrying inelastic voice traffic, and WiFi for carrying elastic data traffic. However, instead of using a dedicated network for each type of traffic, we focus on supporting the transport of traffic with different quality of service (QoS) requirements over a common wireless infrastructure. The benefit lies in the reduced cost of managing and maintaining a single network.

1.1 Wireless Networks

The most basic wireless network is a single-hop network wherein all communication occurs between wireless client stations and an access point (AP) with a wired connection to the Internet. Today single hop WiFi networks are widely deployed. These WiFi networks are based on IEEE 802.11a/b/g specifications designed for carrying elastic data traffic. New technologies like 802.11e and WiMax are designed to handle the coexistence of inelastic and elastic traffic. However these technologies are just emerging: 802.11e is not commercially available and WiMax is still in limited deployment.

The next step from a single-hop scenario is a fixed multi-hop network. Here the client stations communicate to an AP via multiple wireless hops. Intermediate nodes relay the communication of other nodes. A fixed multi-hop set-up extends network coverage without the cost of adding wireline access. Today we are moving towards the deployment of fixed multi-hop wireless networks by universities and local governments to provide access

for a campus or within urban areas. An example is the TIER project that is deploying a large multi-hop WiFi network for providing Internet to remote villages in India.

Relaxing the condition that all nodes are fixed leads to the mobile ad-hoc case. Now even the intermediate nodes that relay transmissions may be mobile. There are no dedicated intermediate nodes that must be fixed as part of the network infrastructure. In the future, we envision mobile ad-hoc networks enabling communication between emergency rescue workers or soldiers on a battlefield. Indeed the demand for ubiquitous access drives the progression of wireless networks.

1.2 Defining Quality of Service

We wish to provide transport for the same types of traffic over wireless as the wired Internet. Today's wired Internet accommodates many types of traffic with different service requirements. We categorize these types as elastic and inelastic.

Elastic traffic refers to traffic of applications where the transmitted information is not time sensitive, but requires eventual correct delivery. Examples of applications that generate elastic traffic are email, web-browsing, file transfers (FTP), Telnet, and any application that can work without timely delivery. Protocols like TCP (the transmission control protocol) control the transmission rate of elastic traffic and allow for reliable transmission.

Inelastic traffic refers to that of real-time applications where the transmitted information is only useful if it is received within a small delay. Examples of applications that generate inelastic traffic are voice over IP (VoIP), video conferencing, and generally any application that requires small end-to-end delay.

We consider telephony-type traffic in the form of VoIP as a representative interactive application with inelastic requirements. When using the VoIP protocol, the source converts the voice signal into a bit stream, packetizes that voice stream, transmits the packets. The source generates fixed size packets over regular intervals. For instance, a VoIP application using the G.711 codec generates 80-bytes every 10ms which corresponds to 64Kbps of voice traffic [35]. The source packetizes the voice traffic by adding up to 74 bytes of headers used by the network to the 80 bytes of payload. The network delivers the packets to the destination, which reconstructs the bit stream and the voice signal for playback.

As it travels across the network, the stream of packets is subject to delays and losses

that influence the subjective quality of the voice service. The characteristics of the delays and losses determine the quality of service that the Internet offers to the VoIP application. The quality of service is acceptable if the delays and losses are small enough. The *end-to-end delay* of a packet is the time it takes to get from the source to the destination. This delay consists of propagation delay, queuing delay, and transmission delay introduced by network components. In order for interactive voice communication like that enabled by the public switched telephone networks to be feasible, the end-to-end delay of packets must be less than about 300ms. Video conferencing has similar requirements. The *loss probability* for a flow is the proportion of packets received incorrectly against the number of packets sent. Losses are caused by transmission errors and by routers that drop packets. The loss probability should be below 3% for acceptable voice quality [7].

1.3 Why not solved?

While the wired Internet accommodates both inelastic and elastic traffic, we cannot naively extend existing solutions for the wired Internet to the wireless setting. The difference is that the wireless channel is shared. In wired networks, bandwidth requirements are per-link constraints so that traffic through a link must be less than the link capacity. In wireless, transmissions interfere with neighboring links so there are additional constraints on channel access that is shared among interfering links.

1.3.1 Single Domain

First focus on a single domain network to understand the impact on the delay and loss metrics used to determine quality of inelastic connections. The wireless medium is shared meaning that a successful transmission requires that no other transmission occurs simultaneously. A station monitors the channel state and only operates when it senses an idle channel. So the delay a packet incurs includes the time it must wait for the channel to become idle, that is the time taken by transmissions of other stations that access the channel ahead of it. This delay may also include time taken by non-network components like microwave ovens whose operations jam the wireless channel. The delay across the wireless hop depends not only on link traffic, but also on the channel access of all sources within range to interfere.

More than simply waiting for the channel to become idle, interference increases

the delay and loss by causing transmission errors. In wired networks, transmission errors are very rare. For instance, the bit error rate of an optical fiber link is typically around 10^{-12}. For packets of a few thousand bits, this bit error rate results in a negligible packet error rate. However in wireless, transmission errors are inevitable since access of the shared wireless channel is not guaranteed. Although stations wait to sense the channel idle to operate there is no mechanism that forbids two stations from attempting to access the channel at the same moment. Transmission errors occur when transmission attempts fail due to noise on the channel or due to collisions. When there is too much noise on the channel, the transmission may not be received correctly resulting in a transmission error. Additional errors occur when transmissions collide. In WiFi, the transmissions between stations are not coordinated explicitly and the collision probability may be arbitrarily high depending on the number and behavior of the competing stations. Large delays occur when many transmission errors occur and multiple re-transmission attempts are necessary. Losses occur when the maximum re-transmission attempt is reached and the packet is dropped.

Moreover the situation becomes even worse as inelastic connections compete with even a small number of elastic connections. The competition is difficult for the inelastic connections because elastic connections tend to use TCP. Instead of transmitting one small packet periodically as a voice connection does, an elastic data connection may try to send large packets in rapid succession. The large amount of data transmissions create more interference leading to the large delay and loss caused by preventing access of the wireless channel as discussed above. Moreover additional contributions to delay and loss are caused by the bursts of TCP traffic that occasionally fill up the single queue shared by all traffic in a WiFi station. These queue overflows are inevitable since TCP is an integral part of Internet operations.

1.3.2 More complex networks

Yet even more problems arise with fixed multi-hop and mobile ad-hoc wireless networks in terms of difficulty dealing with interference between multiple wireless links and with mobility. Providing guaranteed service over wireless links requires consideration of the effects of interference from other links. In the single domain case, it is assumed that all network components are within one-hop away from the AP and a solution requiring explicit coordination is implementable. However this cannot be assumed in multi-hop networks. It

is common to estimate the interference range as twice the transmission range. Thus for some network topologies coordination amongst interfering links is impossible since the set of interfering links may not even be connected. Even for connected networks, coordination amongst interfering links is difficult since the effects of interference are far reaching.

In the case of mobile ad-hoc networks, coordinating the use of wireless channel has now become more complicated since the topology is no longer fixed. Even simple problems like network organization and locating a mobile client require adaptive solutions.

1.4 Approaches

The obstacles posed by wireless become more complicated as network topologies become more extreme. The scope of the problem is large and requires solutions at many levels. This work comprises of solutions for aspects of MAC, admission control, and routing that improve upon the existing technologies. Providing QoS over wireless requires solutions that involve Medium Access Control (MAC) and admission control. Additionally the multi-hop and mobile ad-hoc cases require solutions for routing. In each case we take a practical approach: solutions for the single hop case should be compatible with deployed WiFi equipment, solutions for the multi-hop scenario should be subject to a distributed implementation, and solutions for mobile ad-hoc networks must be highly adaptable.

1.4.1 MAC

Medium access control (MAC) regulates the access of the wireless channel. Generally we categorize MAC schemes as either contention-based or coordinated schemes. Contention-based schemes rely on a randomized contention protocol to determine channel access. These schemes are unscheduled but use a randomized scheme that allows for distributed contention for channel access. Theses schemes require no prior knowledge of demand and work well with elastic traffic. Examples of contention based schemes are 802.11a/b/g Distributed Coordination Function (DCF) used in today's WiFi networks and 802.11e Enhanced Distributed Coordination Function (EDCF). 802.11e is a contention-based protocol specifically designed to address QoS requirements. 11e provides relative priority so that inelastic transmissions have a higher chance of gaining channel access over elastic transmissions. Coordinated schemes like 802.11 Point Coordination Function (PCF)

and WiMax are slotted and scheduled thus providing a guaranteed rate. These schemes are ideal for transporting inelastic traffic whose requirements are regular and known.

There has been much work on designing new MAC protocols as a means to provide QoS over wireless. Indeed coordinated schemes specifically designed to meet inelastic requirements will result in improvements over WiFi that was designed for the transport of elastic traffic. However WiFi equipment is widely deployed and commercial products for WiMax are only beginning to be produced at the time of writing. Thus our contribution focuses on the feasibility of providing QoS over WiFi networks. We use an analytical model to understand the inefficiencies and obtain the capacity limit of WiFi. Our analysis captures a realistic network scenario where the link speeds change via the rate adaptation protocol as is implemented by card vendors and the traffic models represent the behavior of voice and TCP sources. The results of the analysis are consistent with published experimental results of other researchers. Understanding the limits of current deployed WiFi provides insights to new schemes for supporting inelastic and elastic traffic that are compatible with existing WiFi equipment.

1.4.2 Admission Control

Admission control regulates the access of inelastic flows. Recall that inelastic flows have fixed requirements that do not adapt to network performance. Instead they require an all-or-nothing type service where the network guarantees a specified performance if possible. Thus admission control solutions for serving inelastic connections like VoIP should work in the way the public switched telephone network admits phone calls: Incoming connections request the network for resources. If the resources are available, the network admits the connection and provide the required QoS for the duration of the connection. Otherwise the network responds with a busy signal. The user defers and attempts establishing the connection at a later time. In this manner the service of on-going calls is always protected.

The decision to admit a new connection requires an estimate of available resources and a method to determine the impact of admitting a new call. Existing schemes use reservations based on approximate capacity estimation which are inaccurate and tend to be too conservative. Unfortunately, an accurate theoretical capacity estimation suffers from two practical problems. First, it requires complete and accurate knowledge of all interfering links – this may be difficult to obtain in practice. Second, it only proves the existence of

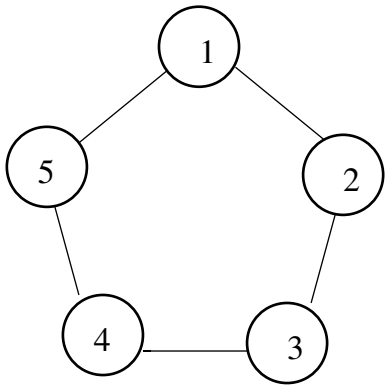

Figure 1.1: Conflict Graph

a feasible schedule to support the flows in the network, but does not provide the means to achieve it. In fact, distributed scheduling mechanisms like 802.11 are seen to be quite far from the optimally feasible schedule.

The following example exposes a pitfall with using an admission control scheme based on capacity estimates. Figure 1.1 shows a "conflict graph" that models the interference relationships between the different wireless links of a network. A node in the conflict graph represents a wireless link, Figure 1.1 represents 5 wireless links numbered 1 through 5. An edge connects two nodes in the conflict graph if the nodes correspond to wireless links that interfere. For example, link 1 interferes with links 2 and 5. That is transmission on one link interferes with the other so that only one link may be active at any time. Capacity estimates using only localized link constraints suggest a valid allocation of 50% utilization on each link, in reality only 40% on each link is achievable since at most two out of the five links may be active simultaneously. In this example, the capacity estimate based on localized information is not achievable due to the network structure. A method based solely on capacity estimation must account for global knowledge of interference relationships.

Instead of using capacity estimates, our approach to admission control uses trial flows. We study a scheme first suggested for the wired setting. Using trial flows simplifies admission control since it guarantees the performance experienced by the trial flows and it eliminates the need to calculate available resources.

1.4.3 Routing

In the case of multi-hop networks, the routing protocol determines a path to the destination. In fixed multi-hop networks there is much work on new routing protocols. For example, offline methods may try to maximize network throughput given traffic demand. However such a problem is NP complete. Many online methods attempt to find a path satisfying QoS requirements for given the current network state. Typical strategies for online methods involve dynamic programming or heuristics. These online solutions are often complicated requiring accounting of network statistics like utilization that is difficult due to the shared nature of the wireless medium.

In this work we consider approaches based on solutions for the wired network. First we describe a multi-path routing scheme developed for wired networks to achieve stability. We discuss why this scheme may extend to fixed multi-hop wireless networks where network performance is greatly determined by the network load.

Then we consider the use of heirarchical routing where stations are clustered. The advantage of organization is in terms of scalability and is seen in the wired setting. Scalability may be even more important in wireless where there is less capacity for transmitting update messages. We study the use of clustering for organization and its impact on QoS routing.

1.5 Overview

We have described a vision of supporting QoS applications over wireless networks. The problem is difficult even in the single-hop network and becomes even more complicated after introducing multiple hops and mobility. The complexity requires solutions that address MAC, admission control, and routing. We present work that addresses some of the issues involved. Figure 1.2 summarizes the contributions.

This thesis is organized as follows. For single domain networks Chapter 2 describes a feasibility study of providing voice traffic over single domain WiFi. We extend an analytical model of other researchers to account for more realistic network conditions. The outcome provides an understanding of what is required to support inelastic traffic like VoIP over today's WiFi technology. Instead of suggesting a completely different technology, we suggest solutions that are compatible with the widely deployed WiFi equipment. For in-

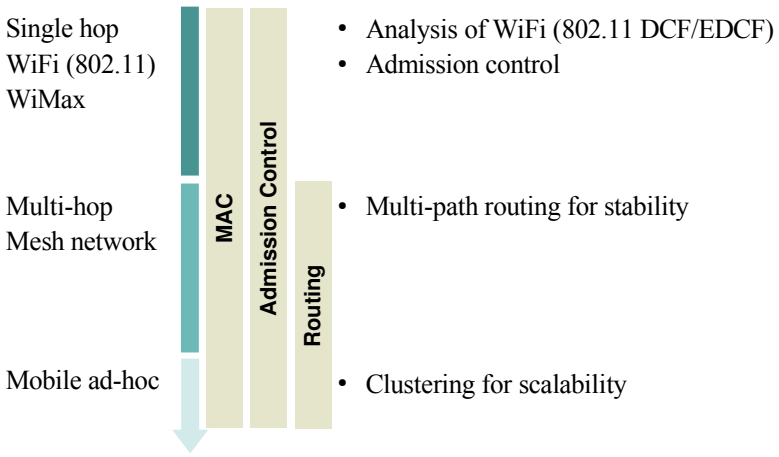

Figure 1.2: Summary of Contributions

stance, we consider an admission control scheme that uses trial flows first studied for wired networks.

Then we describe work for multi-hop and ad-hoc cases. Chapter 3 describes a multi-path routing scheme for wired networks that may extend to the fixed multi-hop scenario. Chapter 4 describes the impact of clustering on routing in an adhoc network.

Chapter 2

Single Domain WiFi

2.1 Introduction

In recent years there has been wide deployment of single domain WiFi networks in home and office settings. The success of WiFi for supporting best effort data traffic motivates studying whether such networks can also support traffic requiring QoS guarantees. For example, in an office setting it may be desirable for the same network to support both data and voice traffic. This chapter describes a feasibility study for supporting inelastic voice traffic over single domain WiFi networks.

The scenario is a single domain WiFi network like the one illustrated in Figure 2.1. The network consists of wireless client stations and a wireless access point (AP) with a wired connection to the Internet. To simplify the analysis we assume that the client stations are either soft-phones or laptops. So a client either transmits voice traffic with inelastic QoS requirements or data traffic with no specialized requirements. The clients are one hop away from the AP and all traffic passes through the AP. The wireless technology is WiFi, the popular name for IEEE 802.11-based technology; we focus on the popular 802.11b and 802.11g standards and the 802.11e standard designed to support QoS. The goal is to understand the wireless capacity in terms of the maximum number of voice connections supported over such networks.

In these networks the capacity is not simply a function of the transmission rate. Experimental results confirmed by those of other researchers [13] and [11] show that the number of possible voice connections corresponds to a small fraction of the nominal rate of the network: For an 802.11b network, the maximum number of bi-directional G.711 VoIP

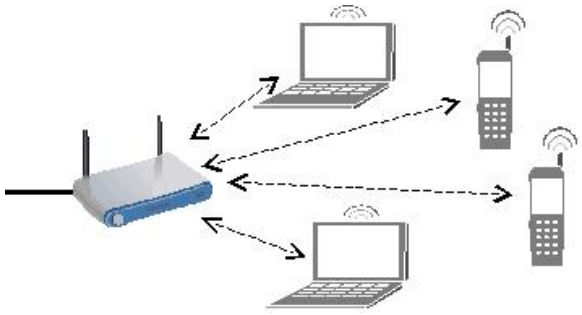

Figure 2.1: A Single Domain WiFi Network

connections with acceptable quality is 6. These 6 connections correspond to an aggregate bit rate of voice signals equal to $2 \times 6 \times 64\text{kbps} = 728\text{kbps}$, or about 7% of the possible 11Mbps channel rate. The result is surprising until one studies the many causes of inefficiency of Wi-Fi when transporting voice calls. One first cause is the large header and preamble that every small VoIP packet carries. A second cause is the idle time that the MAC protocol inserts between two packet transmissions. The third cause is the additional random delay that a station must wait for after a collision. The probability that these random delays are excessive for voice connections is substantial even for a small number of connections.

The situation gets worse as soon as the voice connections compete with even a small number of elastic data connections. The competition is difficult for the voice connections because data connections tend to be persistent and to send large packets. That is, instead of transmitting one small packet periodically as a voice connection does, a data connection may try to send large packets in rapid succession. To understand the voice capacity in WiFi we must characterize the impact of data traffic.

This chapter is organized as follows. In Section 2.2, we present a review of the 802.11 MAC protocol and a summary of related work. In Section 2.3, we explain the model and we outline the methodology that we used for analysis of voice traffic. In Section 2.4, we describe how to extend this analysis for data traffic. In Section 2.5, we present the results of the analytical study. We conclude in 2.6 with a summary of the results and with suggestions for improving the service of voice over WiFi.

Codec	GSM 6.10	G.711	G.723.1	G.726-32	G.729
Bit rate (Kbps)	13.2	64	5.3/6.3	32	8
Framing interval (ms)	20	10	30	20	10
Payload (Bytes)	33	80	20/24	80	10
Packets/sec	50	50	33	50	50

Table 2.1: Voice Codecs

2.2 Overview

We are interested in finding the voice capacity in terms of the number of simultaneous calls whose QoS requirements are sustained. This number depends on the type of calls, the presence of competing data traffic, and the particular 802.11 protocol. This section provides an overview of these scenario details. In 2.2.1 we describe assumptions on the traffic types. Then we briefly recall the key features of the 802.11 MAC protocols in 2.2.2 through 2.2.4. And in 2.2.5, we describe the related work for modeling the 802.11 MAC.

2.2.1 Traffic Types

We consider voice connections in the form of conversations or calls. A voice call is a bi-directional connection between the client station and the AP. In each direction, the voice traffic is voice over IP (VoIP) that generates fixed sized packets periodically. Table 2.1 lists common codecs for packetizing VoIP. For instance, G.711 codec generates 80byte packets every 10ms resulting in a bit rate of 64Kbps of traffic for one direction, or 128Kbps of bi-directional traffic for the conversation. The analysis assumes the generalized assumption that VoIP generates constant bit rate traffic as described. However the results focus on when VoIP uses the popular G.711 or G.729 codecs.

We assume that elastic data connections use the TCP mechanism. The source sends data packets and the destination responds with TCP acknowledgments. For these connections the rate of acknowledgments is approximately the same as the rate of data packets.

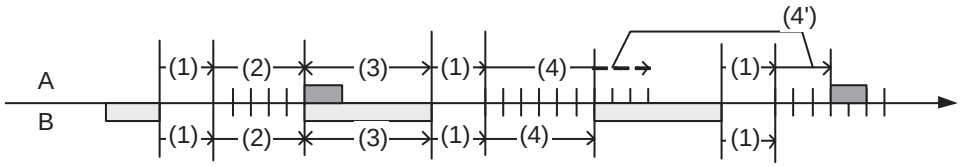

Figure 2.2: The 802.11 MAC Protocol

2.2.2 802.11 DCF MAC Protocol

The basic 802.11 protocols use the Distributed Coordination Function (DCF) to manage access of the wireless channel. Figure 2.2 illustrates the operations of the 802.11 DCF. It corresponds to a situation where two stations A and B compete for a common channel. In the figure, (1) is the initial delay that a station must wait for after the channel becomes idle before it can start transmitting. This delay is the Distributed Interframe Spacing (DIFS) and is equal to 28μs in 11g. We will explain shortly the goal of this delay. In (2) the stations pick a random delay uniformly from the set $\{0, 1, 2, \ldots, CW_{min} - 1\} \times IDLE$ before the first transmission attempt of a packet. Here CW_{min} and $IDLE$ are parameters of the protocol. In 11g, $CW_{min} = 16$ and $IDLE = 9\mu$s and the delay is chosen from $\{0, 1, \ldots, 15\} \times 9\mu$s. The figure assumes that A and B pick the same delay. In (3), A and B transmit and their packets collide. The stations then repeat step (1) and wait for the channel to be idle for a duration equal to the initial delay DIFS. The goal of this delay is to wait for the acknowledgment of a successful transmission that is sent after waiting a Short Interframe Spacing (SIFS) period that is shorter than the DIFS. In (4), the stations again pick a random delay uniformly, but in a set that doubles after each collision. Thus in 11g, after one collision the set is $\{0, 1, 2, \ldots, 31\} \times 9\mu$s, after a second collision the set is $\{0, 1, 2, \ldots, 63\} \times 9\mu$s, and so on. The figure assumes that A picks a delay that is 3 idle slots longer than that of B. Both A and B decrement their delay timer whenever they see an empty time slot. As the figure shows, B starts transmitting when A's timer still shows 3 slots. A's count-down timer remains frozen during B's transmission. When the channel becomes idle again, the stations wait for the initial delay before they can resume counting down. Three time slots later, indicated by (4'), A transmits.

Summing up, the operations of one station are as follows: The station has a timer that it decrements when it sees an empty slot. The timer is frozen when the channel is busy and resumes counting down a fixed delay after the channel is idle again. When the timer

	802.11b	802.11g
$DIFS$	50μs	28μs
$SIFS$	10μs	10μs
$IDLE$	20μs	9μs
CW_{min}	32	16
CW_{max}	1024	1024
Supported rates	1,2,5.5,11 Mbps	1,2,6,9,12,18,24,36,48,54 Mbps
ACK	248	24

Table 2.2: 802.11 parameters

reaches zero, the station transmits. If the transmission collides, the station selects a new timer value in a set of values that doubles in size after every collision until the transmission is successful or until a maximum retransmission limit is reached. The station chooses the initial delay from the set $\{0, 1, 2, \ldots, CW_{min}\} \times IDLE$.

We described the 802.11 MAC protocol using parameters specific to 802.11g. 802.11a and 802.11b follow the same general procedure but with different units for the following:

- $DIFS$ is the amount of time to wait after the channel becomes idle before a station may resume competition for the channel (1),
- $SIFS$ is the amount of time to wait after a successful transmission before responding with a MAC layer acknowledgment whose transmission lasts duration ACK,
- $IDLE$ is the duration of the idle slot,
- CW_{min} is the size of initial set from which the random back-off is chosen (2),
- CW_{max} is the size of the maximum set from which the random is chosen,
- and the supported transmission rates.

Table 2.2 lists values for 802.11 parameters.

2.2.3 802.11e EDCF MAC Protocol

While the 802.11 DCF treats all traffic types equally, the 802.11e Enhanced Distributed Coordination Function (EDCF) provides differentiated service for voice and data traffic. The 802.11e standard provides such service by regulating the competition for the

channel in a differentiated way for stations transmitting different traffic types. Within a station voice traffic is given priority over data. However across stations voice traffic is given relative priority over data. The key idea is that a station transmitting data must wait longer than a station transmitting voice after the channel becomes idle before it can compete. Moreover, after a collision, a data station must wait a random delay that tends to be larger than that of voice station.

Again we refer to the situation illustrated in Figure 2.2 to discuss the details. 802.11e provides differentiated service by varying the delays in (1) and (2) per traffic class. In 11e, the initial delays (1) are different for voice (28 μs) and data (37 μs). The shorter delay gives a slight edge to voice transmissions: a station transmitting a voice packet starts counting down its random delay 9μs before a data station does. The difference between the delays for voice and data is no coincidence and is the equal to the duration of an idle slot ($IDLE = 9\mu$s) in 11e. Thus the channel must be empty for an additional idle slot before a data station may begin back-off operations. Additional advantage may be given to voice over data by requiring data transmissions to wait more idle slots. This initial delay is the Arbitrated Interframe Spacing (AIFS) and is different for each class of traffic.

Like before the duration of (2) is a random delay chosen uniformly from set $\{0, 1, 2, \ldots, CW_{min} - 1\} \times IDLE$ for the first transmission attempt of a packet. However, now each class of traffic has an associated value of CW_{min} so the size of the set is different for different traffic types. For example the initial delay is chosen from $\{0, 1, \ldots, 3\} \times 9\mu$s for voice and $\{0, 1, \ldots, 15\} \times 9\mu$s for data. The remaining operations are as described for DCF above: The stations again pick a random delay uniformly, but from a set that doubles after each collision. The process continues until the packet is successfully transmitted or until the maximum window size CW_{max} is reached and the packet is dropped.

The 802.11e rules protect the voice connection, but they do so only partly. As a result, the acceptable number of voice connections in the presence of data is larger under 11e than under 11g. However, these benefits of 11e are only possible if all the devices in the network use that protocol. This condition is unlikely given the large number of 11b/g equipment already deployed.

2.2.4 Packet overhead

The per packet overhead in 802.11 networks is large and is due to the MAC mechanism described above and to the headers used for transport. Each packet requires 74 bytes of header information (RTP, UDP, IP, MAC) which is large compared to the VoIP payload that ranges from 10 bytes to 160 bytes (Table 2.1 lists various codecs used for VoIP; Table 2.3 lists the sizes of the headers). Furthermore, each packet incurs a MAC overhead for successful transmission or collision. For example, the time taken by a successful transmission consists of the time to transmit a preamble (PRE) for the packet, the packet transmission time, a $SIFS$ period, the time to transmit a preamble for the MAC acknowledgment, the transmission time of the MAC acknowledgment (MAC_ACK), and a $DIFS$ period that other stations must wait before counting down back-off counters or accessing the channel. The preamble is the time to transmit the PLCP preamble and physical header. Given the link rate R, the following equations give the time a successful transmission or a collision occupies the channel:

$$T_V^S = PRE + PACKET + SIFS + PRE + MAC_ACK + DIFS, \quad (2.1)$$

$$T_V^C = PRE + PACKET + DIFS, \quad (2.2)$$

$$PACKET = \frac{RTP + UDP + IP + MAC + PAYLOAD}{R}. \quad (2.3)$$

Default values for the preamble are $PRE = 192\mu s$ for $R = 1$Mbps and $PRE = 96\mu s$ for other transmission rates. Table 2.2 gives the remaining values. Note that we do not consider the additional RTS/CTS virtual carrier sense mechanism. RTS/CTS is known to be inefficient for small packets like those generated by VoIP.

Since we also consider data traffic in the scenarios below, let us define the following. Let T_D^S be the duration of a successful transmission of a data packet, T_D^C be the duration of a collision of a data packet, T_A^S be the duration of a successful transmission of a TCP acknowledgment (ACK), and T_A^C be the duration of a collision of TCP ACK packets. These are similarly defined as

$$T_n^S = PRE + PACKET + SIFS + PRE + MACACK + DIFS, \quad (2.4)$$

$$T_n^C = PRE + PACKET + DIFS, \quad (2.5)$$

$$PACKET = \frac{TCP + IP + MAC + PAYLOAD}{R}, \quad (2.6)$$

for $n = A$ and D.

RTP	16bytes
UDP	8bytes
MAC	30bytes
IP	20bytes
MAC ACK	14bytes

Table 2.3: Packet Headers for VoIP

2.2.5 Related Work

There are two primary approaches for analyzing the 802.11 DCF described above: via the Bianchi model or the p-persistent model. The p-persistent approach approximates 802.11's random back-off procedure by a geometrically distributed back-off so that a station transmits with probability p and defers otherwise. See [4] for a summary of details and references for the p-persistent approach.

The more typical approach uses Bianchi's formulation described below in Section 2.3.1 which more accurately models the 802.11 back-off procedure. Bianchi's original formulation models a single domain 802.11 network where all the stations are saturated, that is they always have packets to send. Much related work extends Bianchi's basic model to account for more realistic network conditions like non-saturated traffic sources and a more accurate representation of the MAC protocol. To account for non-saturated stations, [12] adds an extra number of idle states geometrically distributed with parameter λ between packet departures. The drawback is the relationship between desired traffic load and λ is not defined. A similar approach is taken by [8] and [29]. [12] also corrects the original model to include an idle slot before each transmission attempt and accounts for sources with different transmission rates by using average values. [36] corrects the original model to account for the retransmission limit.

This work extends the work of other researchers to account for more realistic behavior by the traffic sources and link rate adaptation. The model considers traffic sources like VoIP and TCP and follows a simplified approach as in [34] and [21]. In [34], the authors model non-saturated traffic sources by introducing a probability of not having traffic to send, but their approach requires estimates to obtain this probability via simulation. [21] uses the same approach but is able to relate this probability to VoIP traffic. Our work expands their approach to also account for TCP. Then we also account for the Auto-Rate Fallback (ARF) link rate adaptation algorithm implemented in commercial WiFi products. Finally

we integrate our extensions with the work by [30] to account for the differentiated service for 802.11e. We outline our approach in the next sections.

2.3 Analysis for Voice only

We obtain analytical results based on extensions of Bianchi's model of the 802.11 MAC protocol; we review this model in 2.3.1. Then we describe our extensions that cover voice traffic types and changes in link speed via the Auto-Rate Fallback (ARF) algorithm. Later in Section 2.4 we describe the extensions that include data traffic with voice traffic as well as the 11e variations.

2.3.1 Basic 802.11 DCF MAC Protocol Model

The Bianchi model is an analytical model of the 802.11 DCF protocol. In the original formulation [3], Bianchi considers the case of a saturated station that always has packets to transmit. The formulation is based on the approximation that the probability a station n finds the channel busy in a slot c_n is independent and constant. We know this assumption is not accurate since c_n depends on the back-off procedure of the other stations. However the independence assumption may be reasonable when there are many contending stations. This assumption allows us to decouple the behavior of the stations.

The basic model is a discrete model where time is divided into slots. The innovation is that slots are not uniform in size: each slot may be a transmission, a collision, or an idle slot. If the slot is idle, then the back-off timer decrements; otherwise the slot is busy and the timer freezes. Using this generic notion of slots and the independence assumption, we model the MAC back-off mechanism for a single station by a Markov chain and obtain the probability that the station transmits in a given slot. By using this probability and studying events that occur in a generic slot, we can calculate the expected delay values used as our criterion for voice quality.

We model the MAC mechanism of one station n as a two-dimensional Markov chain given by (k, b), where $k \in \{0, 1, 2, \ldots\}$ is the number of retransmissions and

$$b \in \{0, 1, \ldots, \min\{CW_{min}2^k, CW_{max}\} - 1\}$$

is the current value of the back-off timer. Recall in 11g, $CW_{min} = 16$ and $CW_{max} = 1024$. Figure 2.3 depicts the this Markov chain for 802.11g.

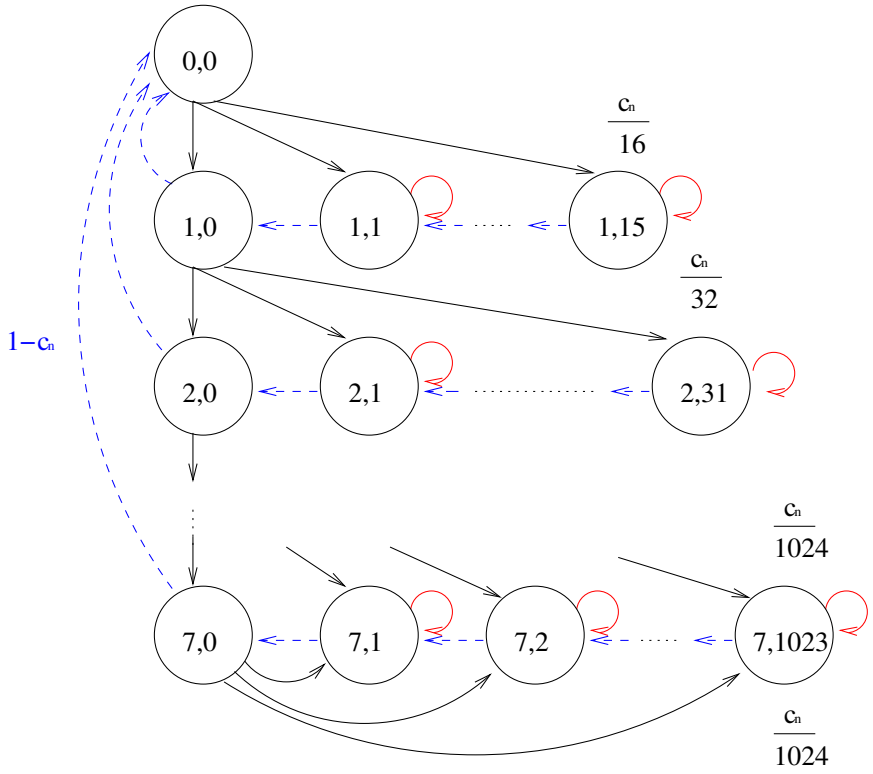

Figure 2.3: **Markov chain for 802.11g**: The dashed transitions occur with probability $1-c_n$, otherwise the transitions to a new back-off value occur with the labeled probabilities or the self-transitions occur with probability c_n.

Under the independence assumption the transition probabilities for the Markov chain are

$$\begin{aligned} P(k, b-1|k, b) &= 1 - c_n, b \neq 0, \\ P(k, b|k, b) &= c_n, b \neq 0 \\ P(0, 0|k, 0) &= 1 - c_n, \\ P(k+1, b|k, 0) &= \frac{c_n}{2^{k+1} CW_{min}}, k \neq K, \\ P(K, b|K, 0) &= \frac{c_n}{CW_{max}}, \end{aligned}$$

where K is such that $2^K CW_{min} = CW_{max}$.

We find the stationary distribution π of this Markov chain and calculate the probability that n transmits in a given slot $p_n = \sum_n \pi(k, 0)$. Specifically,

$$p_n = \frac{2(1-2c_n)(1-c_n)}{(CW_{min}+1)(1-2c_n) + CW_{min} c(1-(2c)^m)}.$$

We refer the reader to [3] and [12] for more details.

Now for each station n, we have an expression for p_n in terms of c_n. The next step is express c_n in terms of the p_n's. For example, when all stations are saturated and all the values of p_n and c_n are the same, then we have that

$$c_n = 1 - (1 - p_n)^N$$

where N is the total number of stations. Below we describe how to setup the expressions for non-saturated stations like voice. Then we solve for the fixed point values of p_n and c_n which can be plugged into straightforward expressions to determine the conversation capacity.

2.3.2 Extension to Non-Saturated Stations

Much of the related work extends the original Bianchi model to consider non-saturated traffic sources. The approaches in [34] and [12] do not apply to a purely analytical setting since they require measurements to tune the parameters. The approach from [36], [8], [29] is complicated since it expands the Markov chain to include states for when there are no packets to transmit.

We take a simpler approach as used by [21]. Instead of adding states to the Markov chain, our solution thins the original Bianchi model by the probability that a station n has a packet to send λ_n. For saturated data stations λ_n is 1, otherwise λ_n is the average fraction of time it takes n to transmit its load of packets, which is just a function of p_n and constant system parameters. Then $\lambda_n p_n$ is the probability that station n attempts transmission. This model approximates VoIP's periodic transmissions with random (Bernoulli) transmissions that have the same rate. Then we have the probability a station senses a busy channel as a function of the p_m's,

$$c_n = 1 - \prod_{m \neq n} (1 - \lambda_m p_m).$$

Voice only

For concreteness, we describe the extension for the voice only scenario. We illustrate the approach described above and focus on how to reconcile the effects of the generic slot model. The analysis for scenarios including data follow the same approach and is described later.

When the network carries only voice traffic a station n may be either the AP or one of N_V voice-type stations. Each type of station exhibits the same behavior. Therefore we assign the same values p_n and c_n for each station of a given type. Below, we designate by c_A the value of c_n when n is the AP, by c_V the value of c_n when n is a voice-type station, and similarly for p_A, p_V, and so forth. (Later when we consider data we denote c_D, p_D, and so forth for data stations).

We find the probabilities of finding the channel busy as a function of the transmission probabilities of the other stations:

$$c_A = 1 - (1 - \lambda_V p_v)^{N_V}, \tag{2.7}$$

$$c_V = 1 - (1 - \lambda_A p_A)(1 - \lambda_V p_V)^{(N_V - 1)}. \tag{2.8}$$

Recall that p_n is the transmission probability given by the Markov analysis for a saturated station that always has packets to send. However our voice-type sources have one packet to send every period T, and the AP has N_V packets for every period T. We thin the saturated case by λ_V or λ_A, the probability that a voice-type station or the AP respectively has a voice packet to send. We approximate this probability as the fraction of time the station

requires to send its load of voice packets:

$$\begin{aligned}\lambda_A &= \min\{\frac{N_V E[d_A]}{T}, 1\},\\ \lambda_V &= \min\{\frac{E[d_V]}{T}, 1\},\end{aligned}$$

where d_n is the time it takes for n to transmit a voice packet from when it arrives at the head of the line. Let T_V^S be the time taken by a successful transmission of a voice packet, T_V^C be the time take by a collision of voice packets, and T^I be the time of an 802.11 idle slot T^I (9 μs in 11g). The durations of T_V^C and T_V^S depend on the transmission rate and are discussed earlier in 2.2.4. d_n consists of the back-off time, the time taken by collisions, and the time to successfully transmit,

$$d_n = \sum_{k=1}^{K_n}\sum_{i=1}^{U_k} S_n^{i,k} + (K_n - 1)T_V^C + T_V^S.$$

Here K_n is the number of channel attempts needed to transmit the packet and is geometrically distributed with mean $\frac{1}{1-c_n}$. U_k is the number of back-off slots at the kth transmission attempt chosen uniformly from $\{0, 1, \ldots, \max\{2^k CW_{min}, CW_{max}\}\}$. And $S_n^{i,k}$ is the duration of the ith back-off slot in the kth transmission attempt and depends on the actions observed by station n.

In terms of our generic slot definition, each $S_n^{i,k}$ may be a duration T_V^S, T_V^C, or T^I from 2.2.4. By our assumption of independence, the distribution of $S_n^{i,k}$ only depends on the type of station and not on i, k, thus $E[S_n] = E[S_n^{i,k}]$ for $n = A, V$. Then the probability a slot is used for successful transmission as seen by the AP or the voice-type station respectively are

$$\begin{aligned}t_A &= N_V \lambda_V p_V (1 - \lambda_V p_V)^{N_V - 1},\\ t_V &= \lambda_A p_A (1 - \lambda_V p_V)^{N_V - 1} + (N_V - 1)\lambda_V p_V (1 - \lambda_V p_V)^{N_V - 2}(1 - \lambda_A p_A),\end{aligned}$$

and

$$\begin{aligned}E[d_n] &= E[S_n]\frac{CW_{min}}{2}[\frac{1 - (2c_n)^m}{1 - 2c_n} + \frac{(2c_n)^m}{1 - c_n}] + \frac{c_n}{1 - c_n}E[T_V^C] + E[T_V^S],\\ E[S_n] &= (1 - c_n)T^I + t_n T_V^S + (c_n - t_n)T_V^C,\end{aligned}$$

for $n = V$ and A. These expressions relate c_A and c_V to p_A and p_V.

2.3.3 Maximum average delay

Our criterion for determining acceptability for voice flows is if the maximum average delay for a voice packet is less than 20 ms. This criterion seems reasonable since a typical average delay of 10ms is acceptable per hop over the wired network. Recall that all stations transmitting the same class of traffic have an equal chance of gaining channel access. So we satisfy the average delay metric for the wireless network if we satisfy the delay on the downlink from the AP which must send the most voice traffic.

Let D_A be the delay experienced by a packet from the time it arrives at the AP until it is successfully transmitted. We approximate the joint arrival process from the N_V voice conversations as a Poisson process with arrival rate of $\frac{N_V}{T}$, then the delay D_A is that of an M/G/1 queue. The average service time is $E[d_A]$ which we computed above from the steady state values of c_A. Let W_V be the random time a voice packet spends waiting from arrival until service begins at the AP, so $D_A = d_A + W_V$. Have that $W_V = W_V^r + W_V^a$ where W_V^r is the random delay due to residual service in progress and W_V^a is the random delay due to other voice packets queued in the system as seen by an arriving voice packet. By Little's Law, $E[W_V^a] = \frac{N_V E[d_A] W_V}{T}$ and we can solve

$$E[W_V] = \frac{E[W_V^r]}{1 - \frac{N_V E[d_A]}{T}}.$$

Thus the expected packet delay is given by

$$E[D_A] = E[d_A] + \frac{N_V E[d_A^2]}{2(1 - \lambda_A)T},$$

where $E[d_A]$ is given above and a straight-forward computation gives $E[d_A^2]$. For 802.11b, the average delay results from this computation are comparable to average delay results from ns-2 simulations as given in [21].

2.3.4 Extension to varying link rates

The existing works assume each station transmits with a fixed link rate. However commercial 802.11 products may vary the transmission rate from packet to packet. Ideally by using this behavior a station discovers the best transmission rate supported by the current channel conditions. We extend the model to consider the behavior of the auto-rate fallback (ARF) algorithm [39].

The ARF algorithm is as follows. When there is a packet to transmit, the station consults a table to determine which rate to use depending on the destination. Initially, the value for a new destination is the maximum possible rate for the protocol. If the transmission is not acknowledged, then the station decreases the table value for the destination to the next lower rate. After ten consecutive acknowledged transmissions to the destination, the station increases the table value for the destination to the next higher rate.

We formulate a two-dimensional Markov chain to model ARF where each state (R, t) is the current link rate R and the number of successful transmissions t needed to increase the link rate. Figure 2.4 depicts the Markov chain for 802.11b when the maximum transmission rate is 11 Mbps. The transitions where R increases or remains the same occur with probability $1 - c_n$ when there is a successful transmission. The transitions where R decreases occur with probability c_n when the transmission is unsuccessful.

For a given maximum possible link rate, we find the stationary distribution from which we find the average link rate. For example, for the 11b case when the maximum possible link rate is 11Mbps the stationary distribution is as follows:

$$\begin{aligned}
\alpha &= 1-(1-c_n)^k, \\
\pi_{11} &= \frac{1}{1+\alpha+\alpha^2+\frac{\alpha^3}{(1-c_n)^k}}, \\
\pi_{5.5} &= \alpha\pi_{11}, \\
\pi_2 &= \alpha^2\pi_{11}, \\
\pi_1 &= \frac{\alpha^3\pi_{11}}{(1-c_n)^k}.
\end{aligned}$$

Then the average rate R is

$$R = 11\pi_{11} + 5.5\pi_{5.5} + 2\pi_2 + \pi_1$$

and is a function of c_n. We use this average link rate to compute the time required for collisions or successful transmissions.

2.4 Analysis for Voice and Data

We now discuss the addition of data traffic to the model. Recall that data traffic uses the TCP mechanism for transport, and we model the TCP differently for responsive or bursty connections. We first describe the extensions for the 802.11 DCF case. Then in 2.4.2

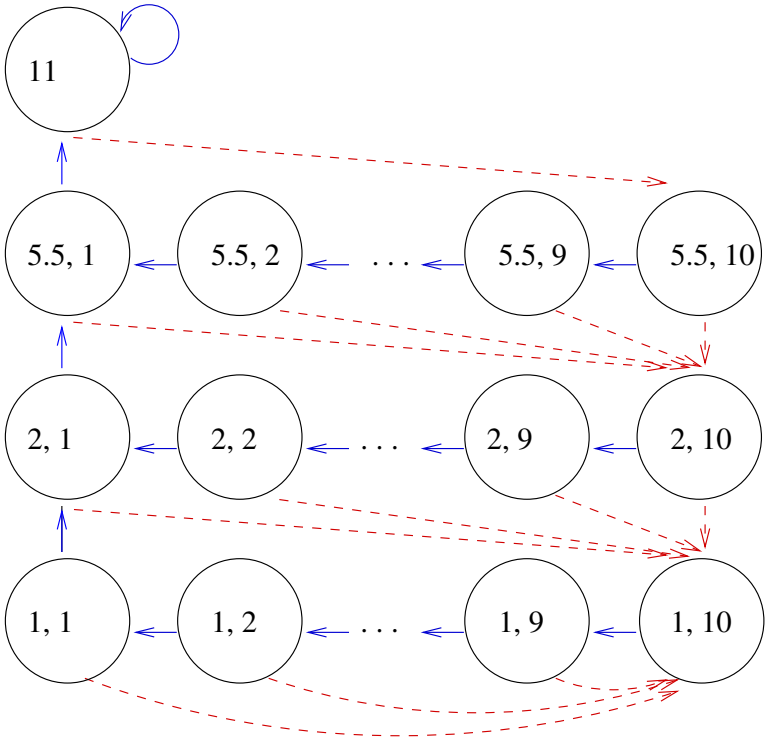

Figure 2.4: **Markov chain for ARF in 802.11b**: The dashed transitions occur with probability c_n, the solid occur with probability $1 - c_n$. The state labeled 11 refers to when the link rate is 11Mbps and any number of successful transmissions.

describe the approach for 802.11e EDCF. While using the 11e model is not difficult, setting up the required fixed-point equations is tedious. We save those details for the interested reader in 2.4.3.

2.4.1 802.11 DCF

TCP Connections

Recall our assumption that data connections are TCP. The TCP data connections are throttled by the available utilization at the AP. The behavior (whether an upload or download) is aggressive and saturates the utilization of the AP which is the bottleneck. Even if we neglect the effect of packet sizes, the constant demand of an arrival rate near 1 results in unacceptable delay at the AP under our M/G/1 model since the queue is shared for all traffic in DCF.

Traffic Shaped Connections

Here we introduce a model for data connections when they are traffic shaped to have constant bit rate features similar to voice traffic. This model is used later to compare against published experimental results for 11b. We consider the case when a data connection is traffic shaped to transmit M packets over T. We model the symmetric case when in addition to the N_V voice connections there are N_D traffic shaped data connections.

In these scenarios a station n may be either the AP, one of N_V voice-type stations, or one of N_D data-type stations. Below, we designate by c_A the value of c_n when n is the access point, by c_V the value of c_n when n is a voice-type station, and by c_D the value for a data-type station, and similarly for p_A, p_V, p_D, and so forth.

Our voice flows generate one packet every time T, we consider data connections where each generates M packets during this period. So each data station has a packet to send a fraction $\min\{\frac{ME[d_D]}{T}, 1\}$ of time where d_D is the time to transmit a packet from the head of the line at the data station. In the case of uploads, d_D is the time to transmit a data packet; in the case of downloads, d_D is the time to transmit a TCP ACK. $\min\{\frac{N_D ME[d_{A,D}]}{T}, 1-\lambda_A\}$ is the fraction of time the AP transmits its load of data packets (ACK packets in the case of uploads and data packets in the case of downloads). $d_{A,D}$ is the delay experienced at the head of the line at the AP for the transmission of a data or ACK packet as appropriate. We approximate the probability a data station or the AP has a data packet to send as λ_D

or $\lambda_{A,D}$ respectively. Then the probabilities of sensing the channel busy are

$$\begin{aligned} c_A &= 1-(1-\lambda_V p_V)^{N_V}(1-\lambda_D p_D)^{N_D}, \\ c_V &= 1-(1-(\lambda_A+\lambda_{A,D})p_A)(1-\lambda_V p_V)^{N_V-1}(1-\lambda_D p_D)^{N_D}, \\ c_D &= 1-(1-(\lambda_A+\lambda_{A,D})p_A)(1-\lambda_V p_V)^{N_V}(1-\lambda_D p_D)^{N_D-1}, \end{aligned}$$

where λ_A and λ_V are defined as before in the voice only case. We have the same expression for d_n as in the voice-only case. It only remains to find the average slot duration viewed by the AP, voice stations, and the data stations.

The average slot durations in the case of traffic shaped uploads are as follows. For the AP, we have

$$\begin{aligned} t_{A,1} &= 1-c_A, \\ t_{A,2} &= N_V\lambda_V p_V(1-\lambda_V p_V)^{N_V-1}(1-\lambda_D p_D)^{N_D}, \\ t_{A,3} &= (1-\lambda_V p_V)^{N_V} N_D p_D(1-\lambda_D p_D)^{N_D-1}, \\ t_{A,4} &= (1-(1-\lambda_V p_V)^{N_V} - N_V\lambda_V p_V(1-\lambda_V p_V)^{N_V-1})(1-\lambda_D p_D)^{N_D}, \\ E[S_A] &= t_{A,1}T^I + t_{A,2}T_V^S + t_{A,3}T_D^S + t_{A,4}T_V^C + (1-t_{A,1}-t_{A,2}-t_{A,3}-t_{A,4})T_D^C, \end{aligned}$$

Similarly for the voice-type station we have

$$\begin{aligned} t_{V,1} &= 1-c_V, \\ t_{V,2} &= ((1-(\lambda_A+\lambda_{A,D})p_A)(N_V-1)\lambda_V p_V(1-\lambda_V p_V)^{N_V-2}+... \\ &\quad \lambda_A p_A(1-\lambda_V p_V)^{N_V-1})(1-\lambda_D p_D)^{N_D}, \\ t_{V,3} &= (1-(\lambda_A+\lambda_{A,D})p_A)(1-\lambda_V p_V)^{N_V-1}N_D p_D(1-\lambda_D p_D)^{N_D-1}+\ldots \\ &\quad \lambda_{A,D}p_A(1-\lambda_V p_V)^{N_V-1}(1-\lambda_D p_D)^{N_D}, \\ t_{V,4} &= (1-(1-(\lambda_A+\lambda_{A,D})p_A)(1-\lambda_V p_V)^{N_V-1} - \lambda_A p_A(1-\lambda_V p_V)^{N_V-1}-\ldots \\ &\quad (1-(\lambda_A+\lambda_{A,D})p_A)(N_V-1)\lambda_V p_V(1-\lambda_V p_V)^{N_V-2})(1-\lambda_D p_D)^{N_D}, \\ E[S_V] &= t_{V,1}T^I + t_{V,2}T_V^S + t_{V,3}T_D^S + t_{V,4}T_V^C + (1-t_{V,1}-t_{V,2}-t_{V,3}-t_{V,4})T_D^C. \end{aligned}$$

Finally for data type station we have

$$\begin{aligned} t_{V,1} &= 1-c_D, \\ t_{V,2} &= ((1-(\lambda_A+\lambda_{A,D})p_A)N_V\lambda_V p_V(1-\lambda_V p_V)^{N_V-1}+... \\ &\quad \lambda_A p_A(1-\lambda_V p_V)^{N_V})(1-\lambda_D p_D)^{N_D-1}, \end{aligned}$$

$$
\begin{aligned}
t_{V,3} &= (1-(\lambda_A+\lambda_{A,D})p_A)(1-\lambda_V p_V)^{N_V}(N_D-1)p_D(1-\lambda_D p_D)^{N_D-2}+\ldots \\
&\quad \lambda_{A,D}p_A(1-\lambda_V p_V)^{N_V}(1-\lambda_D p_D)^{N_D-1}, \\
t_{V,4} &= (1-(1-(\lambda_A+\lambda_{A,D})p_A)(1-\lambda_V p_V)^{N_V}-\lambda_A p_A(1-\lambda_V p_V)^{N_V}-\ldots \\
&\quad (1-(\lambda_A+\lambda_{A,D})p_A)N_V\lambda_V p_V(1-\lambda_V p_V)^{N_V-1})(1-\lambda_D p_D)^{N_D-1}, \\
E[S_D] &= t_{V,1}T^I + t_{V,2}T_V^S + t_{V,3}T_D^S + t_{V,4}T_V^C + (1-t_{V,1}-t_{V,2}-t_{V,3}-t_{V,4})T_D^C.
\end{aligned}
$$

Similarly in the case of downloads, we can enumerate the possible events each station sees to obtain $E[S_n]$, for $n = A, V, D$.

Here we have c_n as a function of p_n. From Section 2.3.1, we have p_n as a function of c_n. Using these values we find the average delay for the AP D_A under the M/G/1 approximation. In this case, the arrivals can come from voice or data connections, where voice arrivals have rate $\frac{N_V}{T}$ and data arrivals have rate $\frac{N_D M}{T}$. As before, $E[D_A] = E[d_A] + E[W_V]$ where

$$
E[W_V] = \frac{E[W_V^r]}{1-\frac{N_V E[d_A]}{T}-\frac{N_D M E[d_{A,D}]}{T}}, \tag{2.9}
$$

$$
E[W_V^r] = \lambda_A \frac{E[d_A^2]}{2E[d_A]} + \lambda_{A,D}\frac{E[d_{A,D}^2]}{2E[d_{A,D}]}. \tag{2.10}
$$

Now the expected residual service time seen by an arriving voice packet is averaged by whether the current packet in service is voice or data.

2.4.2 Extension to 802.11e EDCF

To extend the model to handle 11e, we adopt the approach taken in [30]. Recall that using 11e data senders must wait longer than voice senders before decrementing back-off timers or transmitting. We classify our generic slots according to this situation. Type A slots immediately follow a busy period and are reserved exclusively for stations transmitting voice packets. The remaining slots are type B slots where stations carrying any packet may operate. We use a Markov chain to find the distribution of the slot types.

Suppose there must be L idle slots following a busy period before all stations can operate. These L idle slots are the difference in the initial delays between voice and data slots introduced in 11e ((1) in Figure 2.2). Consider a Markov chain that tracks the number of successive idle slots as depicted in Figure 2.5. The state s takes on values from $\{0, 1, \ldots, L-1, \text{Type B}\}$. When the chain is in states $0, 1, \ldots, L-1$ the slots are type A.

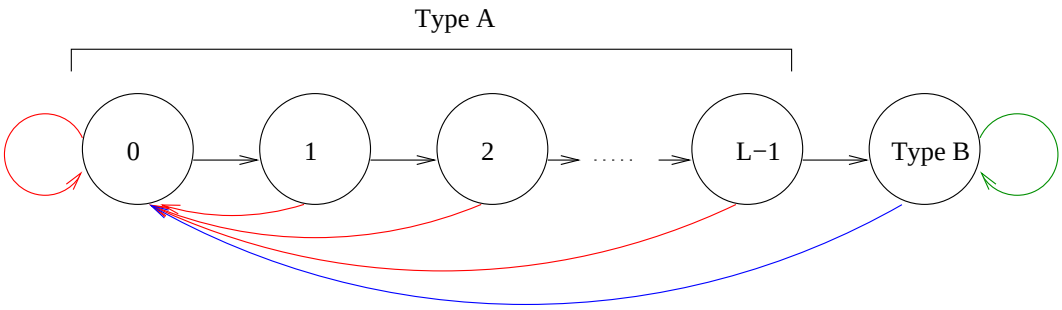

Figure 2.5: Markov chain for the classification of slots in 802.11e

After L successive idle slots, the slots are type B. The transition probabilities are

$$\begin{aligned} P(b+1|b) &= (1-\lambda_A p_A)(1-\lambda_V p_V)^{N_V}, b \neq B \\ P(0|b) &= 1 - P(b+1|b), b \neq B \\ P(B|B) &= (1-\lambda_A p_A)(1-\lambda_V p_V)^{N_V}(1-\lambda_D p_D)^{N_D}, \\ P(0|B) &= 1 - P(B|B), \end{aligned}$$

where $\lambda_n p_n$ is the probability of transmission for a station of type n (recall $n = A$ for the AP, $n = V$ for one of N_V voice stations, or $n = D$ for one of N_D data stations). Writing $q_A = (1-\lambda_A p_A)(1-\lambda_V p_V)^{N_V}$ and $q_B = (1-\lambda_D p_D)^{N_D} q_A$ the stationary distribution of a slot being type A or B are, respectively:

$$\begin{aligned} \pi_A &= \frac{1 + q_A + \ldots + q_A^{L-1}}{1 + q_A + \ldots + q_A^{L-1} + \frac{q_A^L}{1-q_B}}, \\ \pi_B &= \frac{\frac{q_A^L}{1-q_B}}{1 + q_A + \ldots + q_A^{L-1} + \frac{q_A^L}{1-q_B}}. \end{aligned}$$

The probability of finding the channel busy is a function of the size and composition of the set of competing stations, thus it must differ among slot types. Since the model does not track progress within a transmission period, our solution uses an average conditional probability, defined in terms of the contention specific values for and the distribution of transmission attempts across slots of different contention types. For example, the probability the AP with voice packets to transmit finds the channel busy is

$$c_A = \pi_A(1-(1-\lambda_V p_V)^{N_V}) + \pi_B(1-(1-\lambda_V p_V)^{N_V}(1-\lambda_D p_D)^{N_D}).$$

Generally the equations for c_n depend on the traffic scenarios and are complicated. We leave the details of the fixed point equations for 2.4.3.

2.4.3 Details of 802.11e

Following [30], we analyze the 802.11e system by categorizing the generic slots in the model by contention types: type A where only stations transmitting voice packets can attempt to use the channel, and type B where stations carrying any packet may transmit. In Section 2.4.2 we found the stationary distribution of slot types π_A and π_B as functions of the transmission probabilities p_n. Here we setup expressions for c_n in terms of p_n's that allow us to solve for the fixed-point as before in the case of 802.11 DCF.

In these scenarios a station n may be either the AP, one of N_V voice-type stations, or one of N_D data-type stations. Below, we designate by c_A the value of c_n when n is the AP, by c_V the value of c_n when n is a voice-type station, and by c_D the value for a data-type station, and similarly for p_A, p_V, p_D, and so forth.

We focus on the case of a bi-directional TCP connection where the packet sizes are the same in each direction. Moreover we argue that the connection is backlogged at the AP. Our reasoning is as follows. On average, nodes of the same class gain an equal number of accesses to the channel regardless of packet size. However, the AP must send voice packets with priority, it then uses the remaining time to send TCP packets. The AP has less opportunity to contend for the channel than data stations that constantly contend. Thus even the transmissions from just one data station are greater than the transmissions from the AP. Thus there will be a backlog of TCP packets at the AP.

As before, $\lambda_V = \frac{E[d_V]}{T}$ is the fraction of time the voice station has packets to transmit. The AP serves voice packet with priority and sends voice packets a fraction of time $\lambda_A = \frac{N_V E[d_V]}{T}$ and TCP data packets the remaining $\lambda_{A,D} = 1 - \lambda_A$. Data stations are busy a fraction λ_D of the time. For simplicity we assume that

$$\lambda_D = \frac{1 - \lambda_A}{N_D}$$

since the TCP mechanism regulates the rate of the up and downlink connections to be about the same: the number of ACKs sent is the number of data packets received, and the number of data packets sent is paced by the ACKs are received. Then the probabilities of sensing a busy channel for the AP with a voice packet to transmit, the AP with a data packet to transmit, a voice-type station, and a data-type station respectively are

$$\begin{aligned} c_A &= \pi_A(1 - (1 - \lambda_V p_V)^{N_V}) + \pi_B(1 - (1 - \lambda_V p_V)^{N_V}(1 - \lambda_D p_D)^{N_D}), \\ c_V &= \pi_A(1 - (1 - \lambda_A p_A)(1 - \lambda_V p_V)^{N_V - 1}) - \ldots \end{aligned}$$

$$\pi_B(1-(1-\lambda_A p_A)(1-\lambda_V p_V)^{N_V-1}(1-\lambda_D p_D)^{N_D}),$$

$$c_D = 1-(1-\lambda_A p_A)(1-\lambda_V p_V)^{N_V}(1-\lambda_D p_D)^{N_D-1}.$$

The AP may use any slot when it has a voice packet to transmit, otherwise the AP may only use slots of type B. The voice station operates all the time, but the data station may only operate in slots of type B.

So as before we must obtain the average head of line delay ($E[d_A]$ and $E[d_V]$) to calculate λ_A and λ_V. The expressions remain the same except for values of the average observed slot time for the AP and voice stations, $E[S_A]$ and $E[S_V]$. To find $E[S_A]$ and $E[S_V]$ we simply enumerate the possible states a station might find the channel. For the AP, we have

$$\begin{aligned}
t_{A,1} &= 1-c_A,\\
t_{A,2} &= N_V\lambda_V p_V(1-\lambda_V p_V)^{N_V-1}(\pi_A+\pi_B(1-\lambda_D p_D)^{N_D}),\\
t_{A,3} &= \pi_B(1-\lambda_V p_V)^{N_V}N_D p_D(1-\lambda_D p_D)^{N_D-1},\\
t_{A,4} &= (1-(1-\lambda_V p_V)^{N_V}-N_V\lambda_V p_V(1-\lambda_V p_V)^{N_V-1})(\pi_A+\pi_B(1-\lambda_D p_D)^{N_D}),\\
E[S_A] &= t_{A,1}T^I+t_{A,2}T_V^S+t_{A,3}T_D^S+t_{A,4}T_V^C+(1-t_{A,1}-t_{A,2}-t_{A,3}-t_{A,4})T_D^C.
\end{aligned}$$

Similarly for the voice-type station we have

$$\begin{aligned}
t_{V,1} &= 1-c_V,\\
t_{V,2} &= [(1-\lambda_A p_A)(N_V-1)\lambda_V p_V(1-\lambda_V p_V)^{N_V-2}+\lambda_A p_A(1-\lambda_V p_V)^{N_V-1}]\times\ldots\\
&\quad (\pi_A+\pi_B(1-\lambda_D p_D)^{N_D}),\\
t_{V,3} &= \pi_B(1-\lambda_A p_A)(1-\lambda_V p_V)^{N_V-1}N_D p_D(1-\lambda_D p_D)^{N_D-1},\\
t_{V,4} &= [1-(1-\lambda_A p_A)(1-\lambda_V p_V)^{N_V-1}-p_A(1-\lambda_V p_V)^{N_V-1}-\ldots\\
&\quad (1-p_A)(N_V-1)\lambda_V p_V(1-\lambda_V p_V)^{N_V-2}](\pi_A+\pi_B(1-\lambda_D p_D)^{N_D}),\\
E[S_V] &= t_{V,1}T^I+t_{V,2}T_V^S+t_{V,3}T_D^S+t_{V,4}T_V^C+(1-t_{V,1}-t_{V,2}-t_{V,3}-t_{V,4})T_D^C.
\end{aligned}$$

Maximum average delay

As before the maximum average delay for voice packets occurs at the AP. Again we compute the delay experienced by at the AP (D_A) by approximating the joint arrival process from the voice conversations as a Poisson process with arrival rate of $\frac{N_V}{T}$, additionally we

have data packets arriving at the AP with rate 1. As before, $E[D_A] = E[d_A] + E[W_V]$ where

$$E[W_V] = \frac{E[W_V^r]}{1 - \frac{N_V E[d_A]}{T}}.$$

Only, now the expected residual service time seen by an arriving voice packet is averaged by whether the current packet in service is data or voice:

$$E[W_V^r] = \lambda_A \frac{E[d_A^2]}{2E[d_A]} + (1 - \lambda_A) \frac{E[d_{A,D}^2]}{2E[d_{A,D}]}.$$

2.5 Results of Study

We present the results separately for 802.11b, voice only in 802.11g, voice plus saturated data connections in 802.11g and 802.11e, then voice plus bursty data transfers in 802.11g and 802.11g+e. We consider voice traffic using the G.711 and G.729 codecs.

2.5.1 G.711 calls

First we compute analytical results for G.711 calls for the purpose of comparing with experimental results to check the accuracy of the model. Our experiments used 3Com USB wireless adapters and a D-Link AP and were performed in an office setting. The measured background traffic on the channel at the begining and end of the tess were always less than 6%. Our experimental results show a capacity of 6 G.711 calls using 11b, and 15 G.711 calls using 11g. In the case of the capacity for G.711 over 11b, results from other researchers [13] and [11] are available and are consistent. We could not find published experimental results to confirm our measurements for the case of 11g.

802.11b

For now, we focus on 11b scenarios and show that our analytical results are consistent with out experimental results as well as those from other researchers. Recall that 11b uses a modulation scheme that transmits at rates 1, 2, 5.5, and 11 Mbps, depending on the packet error rate.

First consider when traffic consists only of G.711 voice conversations, each generating 80bytes every 10ms. Figure 2.6 shows the average delay for packets from the AP. We find that the average delay is less than 7ms for less than 6 conversations, but increases to infinity otherwise. The result that 11b networks support 6 conversations is consistent

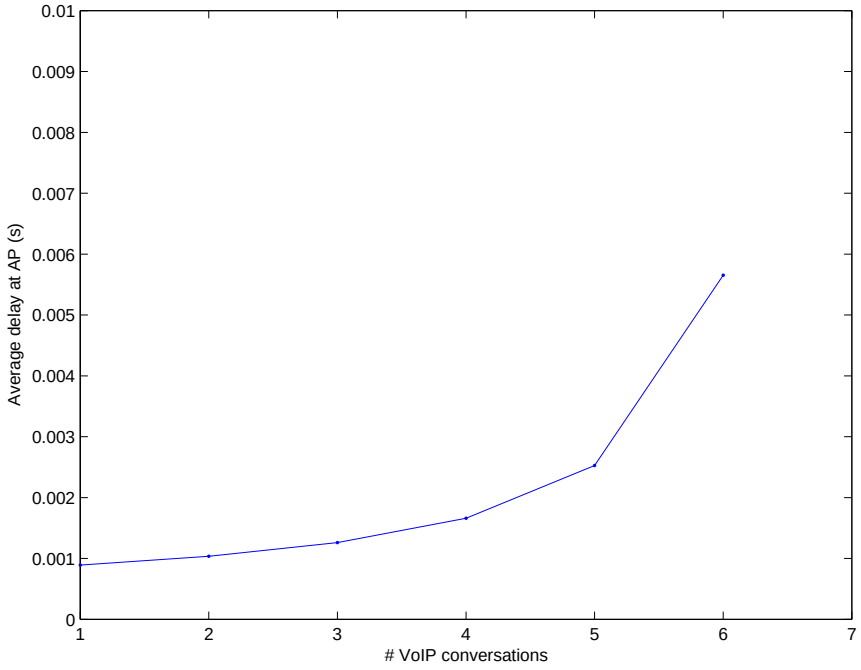

Figure 2.6: Average delay from the AP for G.711 voice packets using 802.11b

with our experiments as well as the experiments in [13] and [11]. The experiments find that the system has acceptable jitter, delay, and loss values when less than 6 conversations are present, but adding the seventh conversation results in an unacceptable number of packets lost or late on the downlink from the AP. Indeed, the bottleneck is at the AP which handles traffic from all conversations whereas individual clients only handle traffic from one conversation.

We also compare results of our analysis with the experiments in [1] and [14] where both VoIP and constant bit rate (CBR) data traffic are present. Now there is the additional presence of a single CBR upload from a client to the AP. In [14] the authors determine the maximum rate of the CBR uplink while maintaining the jitter and loss values for a fixed number of voice conversations. Table 2.4 lists the results of their experiments with the expected results from our analytical model. One reason our expected results are more optimistic than the experimental results is because our criterion of average delay is more lenient than the jitter requirements.

G.711 VoIP conversations	Measured CBR	Predicted CBR
1	5.15Mbps	5.70Mbps
2	4.26Mbps	4.84Mbps
3	3.28Mbps	3.88Mbps

Table 2.4: G.711 VoIP conversation capacity in the presence of uplink UDP

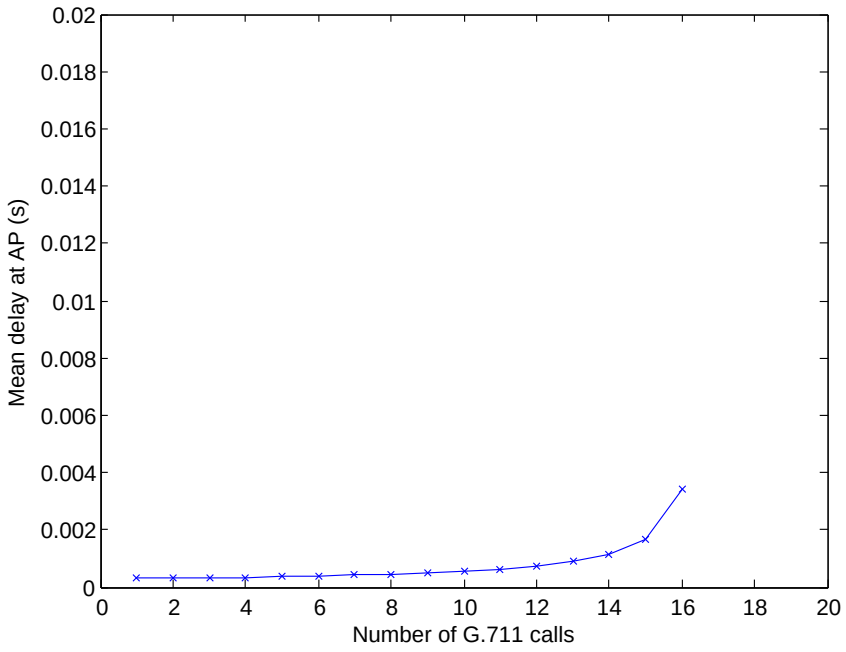

Figure 2.7: Average delay from the AP for G.711 voice packets using 802.11g

802.11g

Now consider the case of an 802.11g network when all traffic is G.711 calls. Figure 2.7 shows the average delay given by the analytical model as a function of the number of simultaneous G.711 calls over 802.11g. The analytical results show that 15 calls are sustained when the delay through the AP is less than 20ms. This capacity is consistent with the number from our experiments.

The model seems correct since our analytical results are close to our experimental and published experimental results. The results confirm that voice capacity is low in 802.11b networks, and better in 802.11g. In this thesis we are interested in the maximum conversation capacity over WiFi. Thus we focus the remainder on 802.11g networks that have a maximum transmission rate of 54Mbps. Also in our calculations we focus the case

when all voice traffic uses the G.729 codec which generates a smaller load of all codecs listed in Table 2.1.

2.5.2 G.729 calls over 802.11g and 802.11e

In G.729, the voice connection sends 20 bytes every 20 ms for each direction of the conversation. We determine the conversation capacity as the maximum number N of voice connections so that the maximum average delay for voice packets is less than 20ms.

Voice only

802.11g uses modulation schemes with rates from 1 to 54 Mbps. Not surprisingly, the link rate impacts the capacity. Figure 2.8 shows the average delay at the AP as a function of the number of simultaneous G.729 calls for link rates 1, 6, and 36 Mbps. Assuming that QoS requirements are met when average delay is less than 20ms, the capacities are 8, 25, and 36 G.729 calls for link rates 1, 6, and 36 Mbps. For lower link rates, the transmission time is a greater factor in determining capacity. As link rates increase, the per packet overhead dominates the calculation. Figure 2.9 shows this effect. We find the maximum number of voice connections when all links use a given fixed rate as shown as the curve in Figure 2.9.

Ideally, the rate is chosen depending on the channel conditions to reduce transmission errors. Commercial 802.11 products may vary the transmission rate from packet to packet to discover the appropriate rate. The typical scheme (as in Atheros products) adjusts the transmission rate via the auto-rate fallback (ARF) algorithm. Using ARF, the conversation capacity is 34 when the maximum link rate is 54Mbps, this capacity is the horizontal line in Figure 2.9. In the following results, we assume that ARF is used and the maximum possible link rate is 54Mbps.

Voice with TCP data connections

Now we consider traffic from voice conversations and TCP data connections. We find that 802.11g cannot support any voice flows in the presence of any TCP data connections. First consider a download: the TCP source is at the AP and the destination is a wireless client. The likely bottleneck for the connection is the wireless link. TCP will continue increasing the transmission rate until the buffer at the AP overflows. At this time,

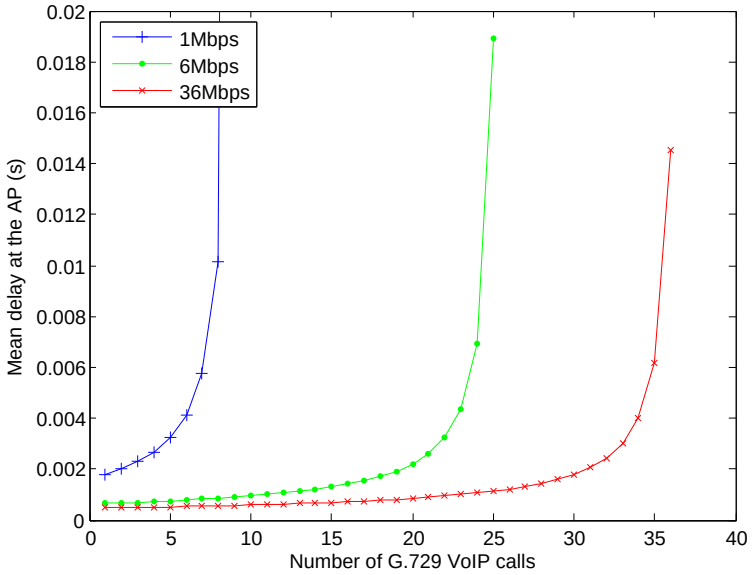

Figure 2.8: Average delay at the AP as a function of G.729 calls in 802.11g

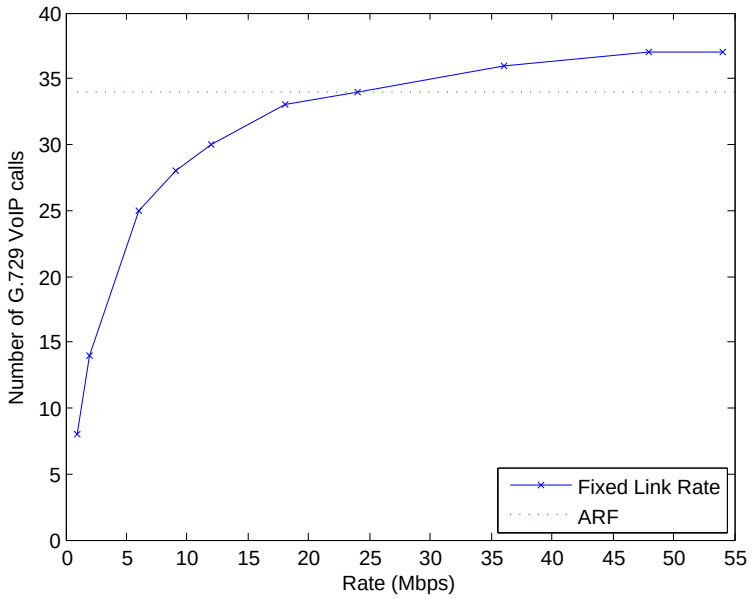

Figure 2.9: G729 conversation capacity in 802.11g

any voice packet will face a large delay since all traffic share a common queue. We have a similar situation in the case of a single saturated upload from a wireless client to the AP. Again the bottleneck for voice packets occurs at the AP where the voice packets must share service with many TCP acknowledgments. Since the data source is saturated, the many TCP acknowledgments saturate the AP. 11g cannot accommodate voice with the presence of a single data upload or download.

2.5.3 802.11e

Voice only

We model the situation that corresponds to the IEEE 802.11e recommendations [38]. The initial back-off is chosen from a set of size $CW_{min} = 4$, and the maximum set size is $CW_{max} = 8$. Since there is only one re-transmission attempt, packet loss due to collisions becomes a factor than delay for determining QoS. Figure 2.10 shows the average loss experienced at the AP as a function of the number of G.729 calls for fixed link rates 1, 6, and 36 Mbps. In this case, we compute the average loss as c_A^2 where c_A is the probability the AP finds the channel busy.

Under these default parameters, the capacity of the wireless channel is not reflected by the utilization at the AP. Figure 2.11 shows the utilization of the AP as a function of G.729 calls. The values for which packet loss of 1% is reached is circled. The circled values are all below 15% utilization at the AP. In the presence of data traffic, the remaining utilization will be used to serve TCP. Using the default 11e parameters, the AP continues to transmit data traffic adding more traffic to the wireless channel even when capacity is already reached. Thus even when using 11e one must choose the 11e parameters carefully to better reflect channel capacity.

Voice and TCP data

We assume that TCP sources send 1000-byte packets. Table 2.5 shows the number of voice conversations compatible with a number of TCP connections for various choices of 11e parameters. For voice packets we vary CW_{min} the size of the initial set from which the back-off is chosen, and M the number of retransmission attempts. We use the default values of $CW_{min} = 16$ and $M = 5$ for data transmissions.

For instance, the entry for shown in boldface shows the capacity attained when

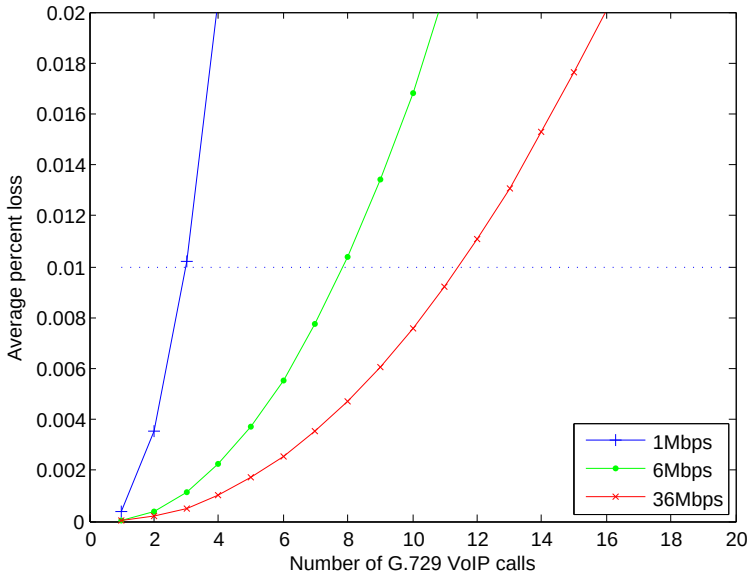

Figure 2.10: Average loss at the AP as a function of G.729 calls in 802.11e

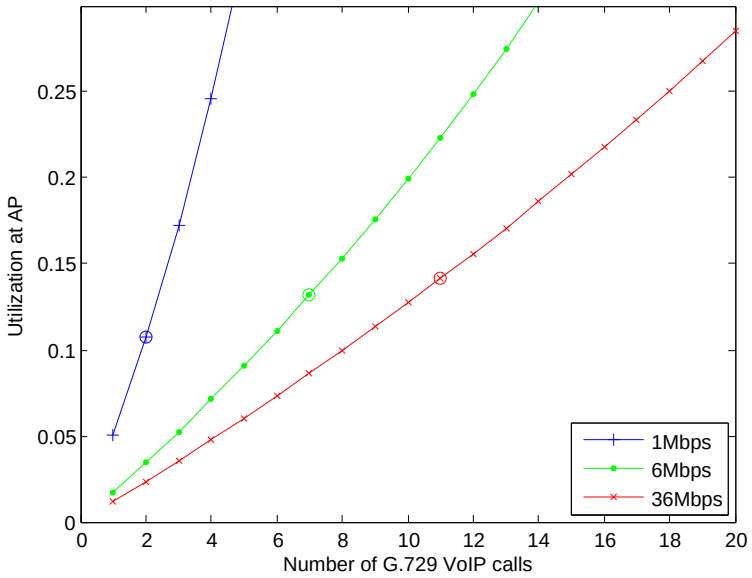

Figure 2.11: Utilization by the AP as a function of G729 calls in 802.11e

CW_{min}/M	1	2	3	4	5	6
2	0/0/7	4/9/16	14/19/25	23/27/31	30/32/36	34/36/38
4	0/0/11	4/11/22	17/23/31	26/31/37	28/32/38	28/32/38
8	0/0/16	5/14/29	18/25/36	19/25/36	19/25/36	19/25/36
16	0/0/22	5/15/34	11/17/34	11/17/34	**11/17/34**	11/17/34

Table 2.5: Capacity of G.729 calls with 2/1/0 TCP connections

802.11g type values are used for voice as well as data ($CW_min = 16$ and $M = 5$). The entry shows that 34 G.729 calls are maintained with voice only, 17 calls when 1 TCP connection is present, and 11 calls when there are 2 TCP connections. The improvement over the 11g case when no calls were maintained in the presence of TCP can be explained mainly by the addition of the priority queue at the AP.

Table 2.5 shows the improvement gained by using an appropriate choice of 11e back-off parameters. Using $CW_{min} = 2$ and $M \geq 6$ for voice lessens the impact of the TCP connections. A station transmitting voice traffic chooses a random back-off from small sets and is allowed to retransmit many times. By increasing the retransmission attempts, the utilization at the AP more accurately reflects the capacity of the channel. Then the priority queue at the AP is more useful for protecting voice traffic from TCP.

2.6 Solutions

We considered the question of "how many VoIP connections can a single-domain Wi-Fi network support?" We used analytical models to answer that question and find the following answers:

- In 802.11b
  With no data traffic: 6 G.711 calls

- In 802.11g
  With no data traffic: 34 G.729 calls
  With any TCP data connections: No VoIP calls

- In 802.11e with default parameters
  With no data traffic: 11 G.729 calls
  With any TCP data downloads: No VoIP calls

- In 802.11e with optimized parameters
  With no data traffic: 38 G.729 calls
  With 1 or 2 TCP connections: 36 or 34 G.729 calls respectively

These results suggest that other mechanisms are necessary to guarantee voice calls in the presence of TCP data connections. Even in the unlikely event that all the stations implement the 802.11e standard, additional mechanisms like admission control and traffic shaping are needed to protect on-going voice connections.

The good news is that changes are not necessary at all stations. Simple changes to the AP should be enough to protect voice service in the single domain case. Given the poor performance of 802.11g with TCP traffic, it is necessary to implement a priority queue at the AP. However more complex mechanisms are also possible since all traffic passes through the AP. For instance an implementation of admission control and traffic shaping mechanisms at the AP is effective for controlling all traffic within the single wireless domain.

Consider the following admission control scheme from the wired setting first described in [15] and studied in [33]. The voice application is changed so that before a call is established the application must first transmit trial packets of the same characteristics as those used by the call. The AP is changed so that trial packets are dropped when the delay for voice traffic through the AP exceeds some threshold (say 20ms). The trial lasts for a short duration of about 1 second or until it notices a packet was dropped. At the end of the trial period, if none of the packets were dropped then the call is admitted. Otherwise, the call is blocked and tries again later. The success of the trial period guarantees the quality of admitted voice traffic. This scheme requires changes for the voice application and the AP but is otherwise compatible with today's off-the-shelf WiFi clients.

To limit data traffic within the single domain network, implementing traffic shaping at the AP should be enough. We rely on the assumption that data traffic uses TCP which is paced by the traffic transmitted by the AP. Downstream data packets generate 1 upstream TCP ACK, and downstream ACKs generate at most 2 upstream data packets. Then by controlling the downstream data traffic we can also control the upstream traffic. Via traffic shaping, the AP controls the amount of data traffic transmitted over the wireless channel to preserve the quality of the voice traffic.

While priority queuing, admission control, and traffic shaping are necessary to enable the coexistence of voice with data, other changes may be added to improve performance.

For instance, using PCF at the AP to schedule the downlink transmissions back-to-back minimizes the overhead due to contention. Or changing the Maximum Segment Size (MSS) used by TCP would minimize the impact of large data transmissions whose packet size can be more than 10 times larger than voice packets. These schemes should work with legacy WiFi equipment and client stations would not even be required to implement 11e.

Chapter 3

Multipath routing

3.1 Introduction

This chapter describes a multi-path routing protocol designed for wired networks. The purpose of this protocol is to preserve load stability for a willingness to accept slightly suboptimal routing. Preserving load stability makes it easier to predict behavior since the amount of re-routing reflects the amount of change in the path length. This routing algorithm may be applicable in fixed multi-hop wireless networks when multiple paths exist.

In shortest path routing, a change in a link's length may result in the rerouting of many of the flows that pass through that link. Consider when the length of a link changes from d to d'. Rerouting is more appropriate for flows for which d is a large fraction of the total path length, and less appropriate for flows for which d is a small fraction. In the case of routes where d is a small fraction of the path length, the new path may be over a set of completely different intermediate links. The load fluctuation is large since all the traffic is re-routed over a completely new set of links. This drastic fluctuation is difficult for network operators to predict since the cause of the change d is only a small fraction of the total path length. The problem becomes worse if the length of d changes often resulting in large amounts of path flapping. There is a trade-off between the stability and the optimality of the route.

We study a scheme that performs probabilistic routing where the probability a path is chosen depends on the path's length. By using multiple paths, a change in a link's length does not effect all paths and not all flows are rerouted. Indeed, [24] notes that using multiple routes improves network reliability. We accept that the shortest path is

not necessarily used for every flow; however, on average we achieve close to shortest path routing. [5] and [18] describe similar probabilistic schemes that use multiple paths. [5] describes a source-based routing scheme that chooses amongst various paths using game theoretic models. [18] describes a distributed scheme for wireless networks where routing decisions are performed on a per packet basis. We focus on a scheme that uses routing tables and for which routes are the same for a fixed connection, that is all packets from the same connection travel along the same route.

This chapter is organized as follows: We discuss the probabilistic routing scheme we call *soft routing* in Sections 3.2-3.4. Section 3.2 describes the simple recursive procedure for computing the routing tables. Section 3.3 describes a method of resampling the routing table when a link's length changes. And Section 3.4 suggests a particular implementation of forwarding using the computed routing tables. Then in Section 3.5 we define costs relating the optimality and the stability of routing. We show simulation examples to validate the performance of and to explore the tunability of soft routing in Section 3.6. Finally we conclude in Section 3.7.

3.2 Soft Routing

We call our routing scheme soft because the network's reactions to changes are soft. When a link's length changes, soft routing limits the amount of rerouting performed by the network: soft routing strives to preserve the stability of the routing tables. In our scheme, the amount of change in the routing table reflects the amount of the change in the path lengths. We describe the details of soft routing below.

3.2.1 Simple Case

First consider the simple scenario illustrated in Figure 3.1. Assume that a source S sends traffic to a destination D and that it can choose among N paths $\pi_1, \ldots, \pi_N$. Each path π_j has distance d_j to the destination. The source sends a fraction p_i of the traffic along path π_i with p_i given by

$$p_i = \frac{q_i}{\sum_{j=1}^{N} q_j} \text{ where } q_j := e^{-\alpha d_j}.$$

In other words given flows of the same size, S routes each flow independently along path π_i with probability p_i. α is a parameter that tunes the routing decisions toward path

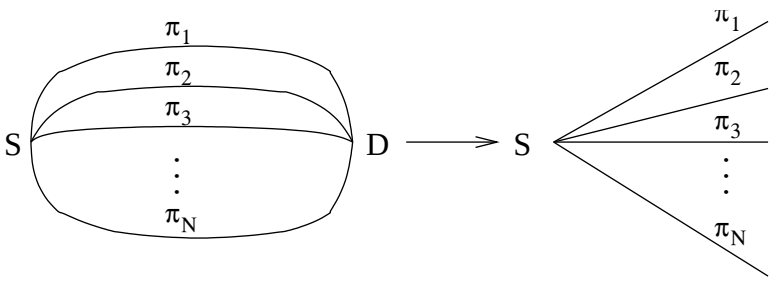

Figure 3.1: **Simple Network** We represent each path between the source and destination in the network on the left as an edge in the tree on the right. Every edge of the tree terminates at the destination.

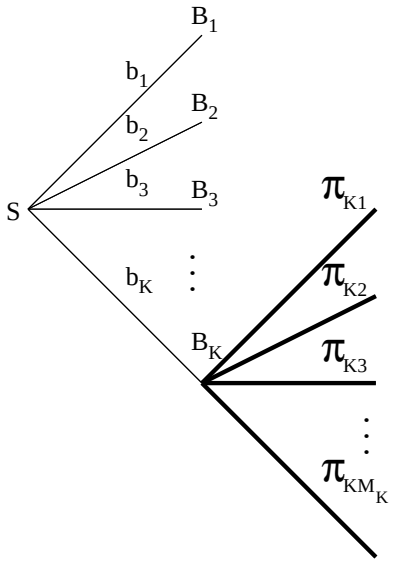

Figure 3.2: **Directed Acyclic Network** The source can choose to forward amongst K next hops. At each next hop B_k, there are choices $\pi_{k1}, \ldots, \pi_{kM_k}$.

optimality or stability. We discuss the role of α in more detail later.

3.2.2 Recursive Construction

To begin, we consider a connected directed acyclic network, where there are no backward edges [9]. Consider a source-destination pair (S, D) in a connected directed acyclic network. The source S has neighboring nodes $B_1, \ldots, B_K$, with link length b_k from S to B_k for $k = 1, \ldots, K$. Node B_k has paths $\pi_{k1}, \ldots, \pi_{kM_k}$ to node D. Assume that path π_{km} has length d_{km}. This setup is illustrated in Figure 3.2. We recursively construct the probabilities as follows.

Step 1: Each node B_k calculates the fraction p_{km} of the traffic that it sends along path π_{km} as before:

$$p_{km} = \frac{q_{km}}{\sum_{r=1}^{M_k} q_{kr}} \text{ where } q_{kr} := e^{-\alpha d_{kr}}.$$

Step 2: Each node B_k $(k = 1, \ldots, K)$ sends to node S the value of $\beta_k := \sum_{r=1}^{M_k} q_{kr}$. The source's own value of β is consistent, $\beta = \sum_{k=1}^{K} \beta_k e^{-\alpha b_k}$. (Note that S sends this β to its upstream neighbors to further the recursion.)

Step 3: Node S sends a fraction a_k of its traffic to node B_k where

$$a_k = \frac{e^{-\alpha b_k} \beta_k}{\sum_{j=1}^{K} e^{-\alpha b_j} \beta_j}.$$

We have that

$$a_k p_{km} = \frac{e^{-\alpha(b_k + d_{km})}}{\sum_{j=1}^{K} \sum_{r=1}^{M_k} e^{-\alpha(b_j + d_{jr})}},$$

is the fraction of traffic sent on the path from S to B_k then along π_{km}. This fraction is consistent with that from the simple case for which S considers all possible paths. Notice that S does not require knowledge of all possible paths, but uses summaries from downstream neighbors.

3.2.3 Construction in a General Network

The above recursion requires that there are no backward edges between the source and the destination. In fact, the above construction works only for a directed acyclic graph. Consider the network in Figure 3.3 [37]. Without the constraint on edges, nodes 2 and 3 have viable paths to node 10 through each other. Soft routing achieves stability by routing flows along all viable paths; however, if we allow all possible paths, there may be loops

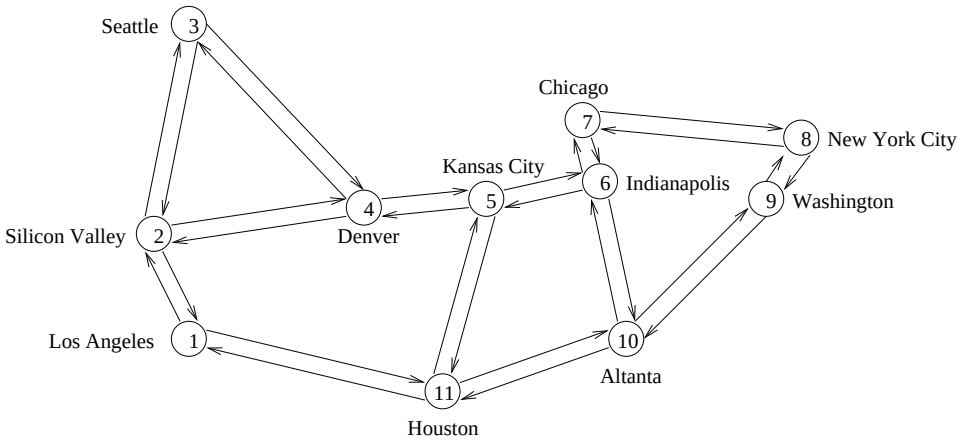

Figure 3.3: Abilene Network

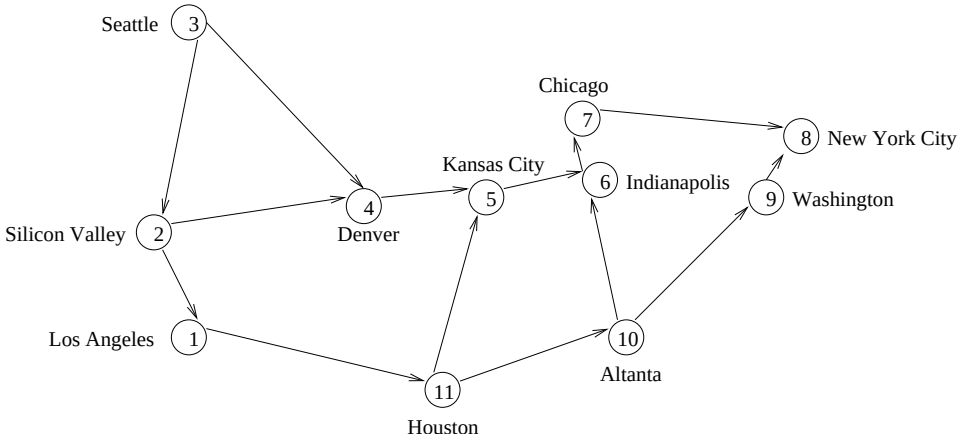

Figure 3.4: Abilene Network representation for computing routing tables for New York City (8)

since nodes may have feasible routes through each other. It is necessary to prune backward edges.

For each destination, we construct a directed acyclic version of the network without backward edges. The following is a distributed procedure for computing such a network for a destination D.

Step 1: Compute the shortest path from each node n to D, denote the distance by $S_D(n)$. We find $S_D(n)$ via a distributed algorithm like the Bellman-Ford distance vector algorithm [9].

Step 2: Each node m includes the edge to neighbor n, called edge (m, n), if any of the

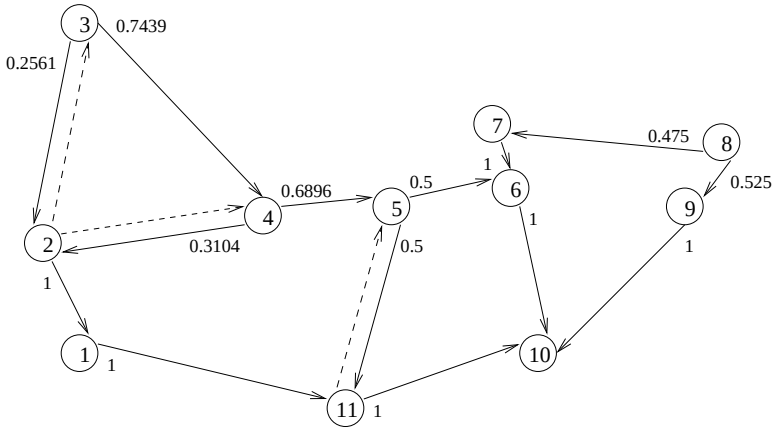

Figure 3.5: Abilene Network representation for computing routing tables for Atlanta (10) shown in solid lines

following holds.

- **Case 1** $S_D(m) > S_D(n)$
- **Case 2** $S_D(m) = S_D(n)$ and $d(m,n) < d(n,m)$, where $d(m,n)$ and $d(n,m)$ are the weights of edges (m,n) and (n,m) respectively.
- **Case 3** $S_D(m) = S_D(n)$, $d(m,n) = d(n,m)$, and $m > n$.

These cases define a unique ordering of the nodes. We include edges from nodes with a higher ordering to nodes of a lower one. Notice that there are no backward edges under this construction. Also note that the edges in all shortest paths to node D are always included. The multiple paths that soft routing uses allow for increased stability. Figure 3.5 is an example of a pruned version of the Abilene network for destination 10 where all the links in Figure 3.3 have unit length.

For routing calculations, each node keeps a link mask to record which links to use for a given destination. Then routing computation occurs for the directed acyclic network as in the previous section. Repeat the above procedure for each destination to complete the routing table.

3.3 Resampling

3.3.1 Simple Case

When a link length changes, we do not recalculate all the probabilities. Instead we *resample* the original probabilities and only reroute flows effected by the change. First revisit the simple case from Figure 3.1. The resampling idea is as follows.

Link Length Increases

Imagine that the length of path π_1 increases from d_1 to $d_1 + \epsilon$ and that the lengths of the other paths do not change. Accordingly, the probabilities p_j of the paths become p'_j in a way that $p'_1 < p_1$ and $p'_j > p_j$ for $j = 2, \ldots, N$. We write $p'_1 = p_1(1 - \delta)$ and $p'_j = p_j + p_1 \delta P(1, j)$ for $j = 2, \ldots, N$. Every flow that was routed along path π_1 has now a probability δ of being rerouted. Moreover, if the flow is rerouted, it goes along path π_j with probability $P(1, j)$. The flows that were not routed through path π_1 are not rerouted. Thus, if δ is small, there are few changes.

Link Length Decreases

If the length of π_1 decreases instead, then $p'_1 > p_1$ and $p'_j < p_j$ for $j \geq 2$. Write

$$p'_j = p_j(1 - \Delta_j) \text{ for } j = 2, \ldots, N, \text{ and } p'_1 = p_1(1 + \delta), \text{ where } \delta = \frac{\sum_{j=2}^{N} p_j \Delta_j}{p_1}.$$

δ is the total fraction of traffic that is rerouted as in the case when the link length increases. We reroute a flow traveling on path π_j, $j \neq 1$, to π_1 with probability Δ_j.

3.3.2 Amount of Rerouting

The amount of rerouting due to resampling decreases as one moves further away from the link change. We show this result for when a link's length increases. There is a similar result for a decrease in a link's length.

Consider Figure 3.6. The link length increases from d_{01} to $d_{01} + \epsilon$. d_i is the length of the upstream link i hops away. At each node i, the change occurs on the first path. Let $\eta_i := \sum_{j \neq 1} q_{ij}$ denote the cumulative weight of all other links out of i. Each node i has γ_i

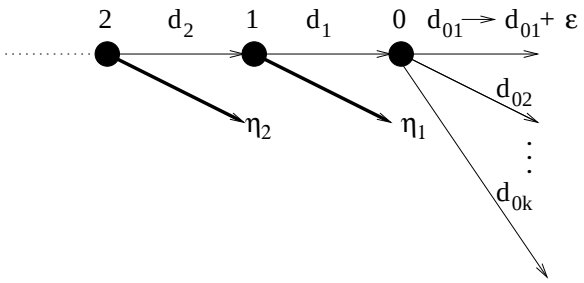

Figure 3.6: The change occurs at link d_{01}.

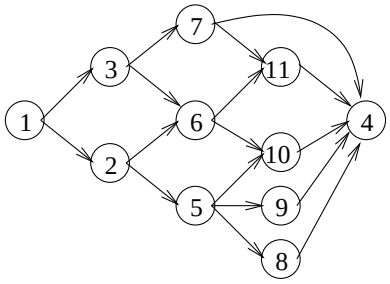

Figure 3.7: Directed acyclic graph for destination 4

and γ_i' as follows,

$$\gamma_0 = \sum_{i=1}^{k} e^{-\alpha d_{0i}} \text{ and } \gamma_0' = e^{-\alpha(d_{01}+\epsilon)} + \sum_{i=2}^{k} e^{-\alpha d_{0i}} \text{ and } \gamma_i = e^{-\alpha d_i}\gamma_{i-1} + \eta_i \text{ and } \gamma_i' = e^{-\alpha d_i}\gamma_{i-1}' + \eta_i.$$

Then

$$\delta_i = 1 - \frac{\gamma_{i-1}'\gamma_i}{\gamma_{i-1}\gamma_i'}$$

where δ_i is the rerouting probability at i hops away. Some algebra gives that $\frac{\gamma_{i-1}'\gamma_i}{\gamma_{i-1}\gamma_i'} \uparrow 1$ as $i \uparrow \infty$. There is less rerouting needed further away from the change.

3.3.3 Resampling in a General Network

Resampling in our simple case extends recursively to a directed acyclic network. In such a network, only nodes upstream from the link change need to be resampled. Adapting to changes in link lengths is more complicated in the general network. A change in a link's length not only changes the proportion of traffic sent along each link, but may also change which links are used. When the shortest path changes the computed directed acyclic

network changes. However, changing the topology used in routing results in more rerouting. But if we do not change the topology, the shorter paths may not be used.

We suggest the following compromise to limit the effect of short-term anomalies in link lengths. Consider the calculation for a destination D. Assume there are discrete time intervals of some period during which the link weights do not change. Let $S_D^k(n)$ denote the shortest path from node n to D during the kth interval. Fix a parameter $\kappa < 1$. We use the standard practice of exponential forgetting [25]. Initially, let $\sigma_D^0(n) = S_D^0(n)$. Then

$$\sigma_D^k(n) = S_D^k(n)\kappa + (1-\kappa)\sigma_D^{k-1}(n).$$

During each time interval, use σ_D^k instead of S_D to generate the directed acyclic topology. The choice of κ dampens the effects of short lived changes. We explore this choice below in the simulations.

3.4 Implementation Issues

We have shown how for a given destination, each node calculates a fraction of traffic p_k to send along each next hop k. The node sends this particular fraction of traffic along each path by independently forwarding each incoming flow with probability p_k along link k. Here we describe the implementation of the forwarding.

Assign each flow a unique flowid: for example the ith flow between source S and destination D has flowid (S, D, i). Assume that given the space of all possible flowids, we have a hash function h that maps uniformly the flowids onto a space of numbers $\{1, 2, \ldots, N\}$. For each number $n \in \{1, 2, \ldots, N\}$, we have another mapping R that specifies the next hop. We assign the map R independently per n: each $R(n)$ is assigned next hop k with probability p_k. For any flowid f, $R(h(f))$ specifies the next hop. For large N, this routing approximates the desired behavior.

Suppose the length of path m increases. The node computes new values p'_k and a resampling probability δ. Resampling only requires recomputing the values where $R(n) = m$. Whenever $R(n) = m$, $R(n)$ is reassigned with rerouting probability δ, where $R(n)$ is rerouted according to the new values p'_k. This setup is illustrated in Figure 3.8 where there are three outgoing links. Similarly, if the length of path m decreases. Resampling occurs at all $R(n) \neq m$.

This forwarding hash table has the advantage that we only reroute flows when

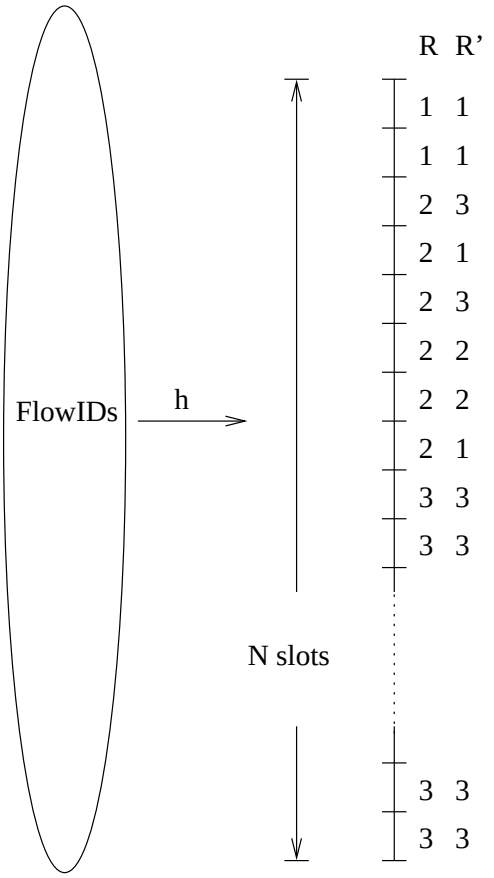

Figure 3.8: **Routing Table** R denotes the original path assignment per slot. Suppose the length of path π_2 increases. R' denotes the resampled path assignments where flows originally destined to π_2 are rerouted.

necessary: When the length of path π increases, we reroute flows from π to other paths. When the length decreases, we reroute flows from other paths to π. The resampling process does not reroute any other flows.

3.5 Cost of Scheme

3.5.1 Cost of Routing

We define the *cost of routing* for a source S and destination D to be the expected length of the $S-D$ routes. A source sends a fraction p_j of traffic along a path of length d_j. The cost of routing is $C(\alpha)$ where

$$C(\alpha) = \sum_{j=1}^{N} d_j p_j = \frac{\sum_{j=1}^{N} d_j e^{-\alpha d_j}}{\sum_{j=1}^{N} e^{-\alpha d_j}}.$$

For $\alpha = 0$, this routing is done uniformly over all routes. As α increases, this average cost approaches the shortest path length, $\min\{d_1, \ldots, d_N\}$. Routing is more optimal for large α, and more stable for small α.

3.5.2 Cost of Rerouting

The cost of rerouting between a source S and destination D is the fraction of routes that change when a link's length changes. For a fixed directed acyclic network, we define the cost of rerouting to be the sum over all the changes in the routes from S to D,

$$R(\alpha, \kappa) = \sum_{i \in U} \delta_i,$$

where U is the set of nodes in paths from S to D upstream from the link change. This cost is the total fraction of flows that need to be rerouted.

Recall that for a general network changes in link length may result in topology changes in the addition or removal of the links. The addition of a link is a change in the link's weight from ∞ to some $l < \infty$, the removal of a link is the opposite. A change in topology is a change of multiple link weights. Thus we define the cost of rerouting as

$$R(\alpha, \kappa) = \sum_{i \in \cup_{j \in L} U_j} \delta_i,$$

where L is the set of links that change weight, and U_j are the nodes upstream from the link j in the new topology. This cost can be greater than 1 reflecting that some flows are rerouted at multiple nodes.

Large κ generates a routing topology that is more responsive to current network conditions. The shortest path is more likely to be used at the expense of more rerouting. So R increases with κ and α.

3.6 Simulations

3.6.1 Choosing α

First we focus on the directed acyclic network for one source-destination pair to demonstrate the choice of α. Consider the network shown in Figure 3.7. Let $d(i,j)$ denote the length of link i to j. First suppose that $d(i,j) = 1\ \forall i,j$. Call this scenario the *original setup*. In this setup, the shortest path is $1 \to 3 \to 7 \to 4$. Consider the following three changes to the original setup: $d(3,7)$ increases to 2, $d(7,4)$ increases to 2, $d(1,2)$ increases to 2.

Figure 3.9 shows the difference between routing cost and the optimal, $(C(\alpha) - C(\infty))$, as a function of α. Notice that the average route length approaches the shortest path length as α increases. Figure 3.10 shows the cost of rerouting, the fraction of traffic that is rerouted. When $d(7,4)$ increases, this fraction approaches one because the shortest path changes so all flows need to be rerouted. There is less rerouting when $d(3,7)$ increases since the rerouting only needs to happen at upstream nodes. The least amount of rerouting is needed when $d(1,2)$ increases because it is not on the shortest path and it is the furthest upstream. One chooses α depending on how common the changes are and how much suboptimality one is willing to incur.

One chooses α depending on how much suboptimality and how much rerouting one is willing to accept. Given the network topology and knowledge of the costs incurred per link failure, we can determine an optimal α in a centralized manner as in [5]. However, we wish to choose α without the centralization of information. Generally, α specifies the sensitivity of the rerouting. Recall for paths π_i with lengths d_i, we route the fraction of traffic $p_1 = \frac{e^{-\alpha d_1}}{e^{-\alpha d_1} + \sum_{i=2}^{N} e^{-\alpha d_i}} = \frac{1}{1 + \sum_{i=2}^{N} e^{-\alpha(d_i - d_1)}}$ along π_1. α determines the sensitivity of p_i since it scales the difference of the path lengths. By specifying α, one specifies the amount

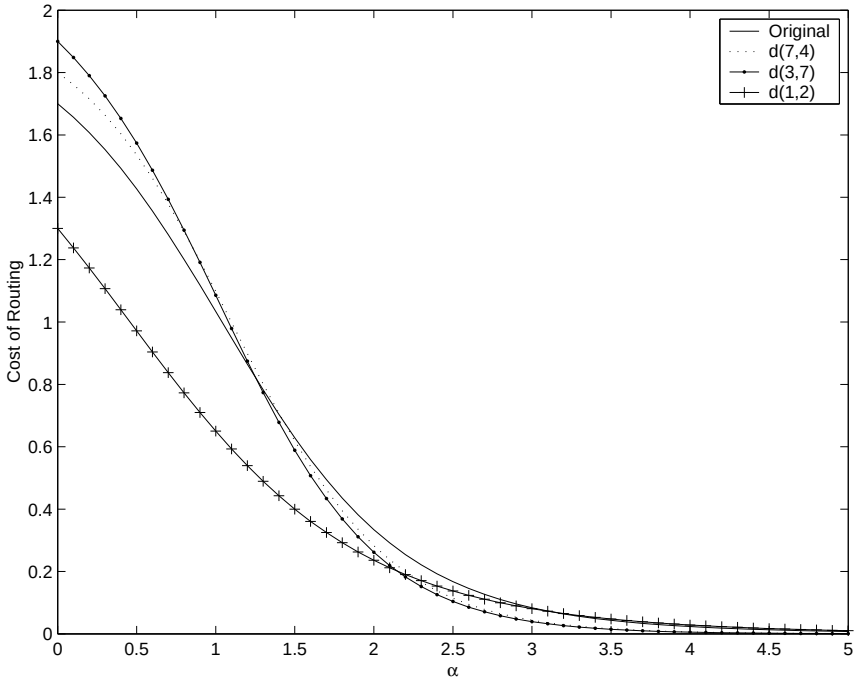

Figure 3.9: $C(\alpha) - C(\infty)$

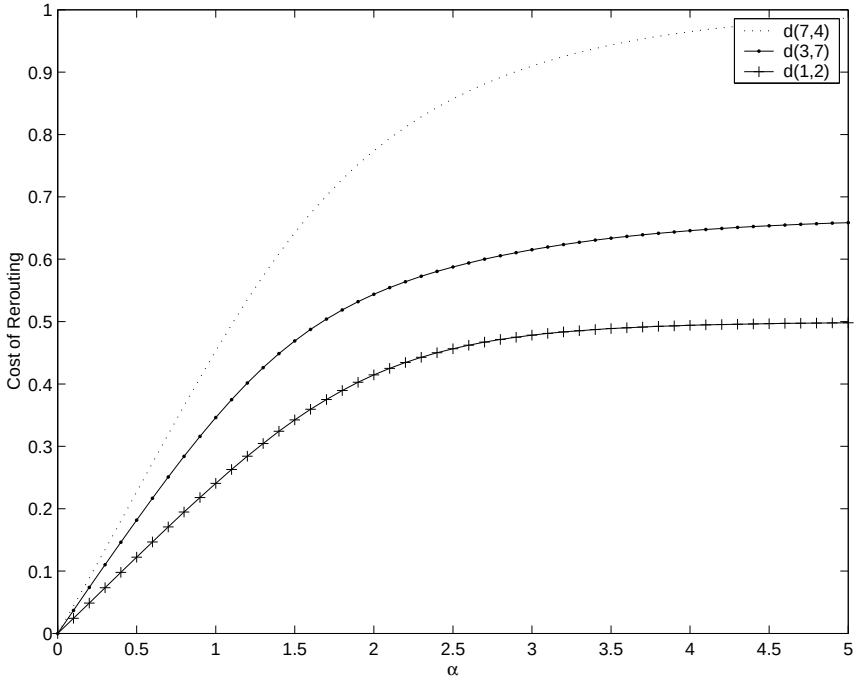

Figure 3.10: Cost of Rerouting

Label	$d(11,10)$	Topology Change
a	1	No
b1	1.1	No
b2	1.1	(2,4)
c1	2	No
c2	2	(5,11),(2,4),(2,3)
d1	5	No
d2	5	(5,11),(2,4),(2,3)

Table 3.1: Changes to $d(11,10)$ and associated topology changes. In cases b1, c1, d1 we assume κ is such that there are no topology changes. In cases b2, c2, and d2 assume κ is such that the topology changes.

of rerouting incurred by a change in the difference of path lengths.

3.6.2 Choosing κ

Consider the network shown in Figure 3.3.

Let $d^k(i,j)$ denote the length of the link (i,j) in time interval k. Initially assume all the links have equal weight $d^0(i,j) = 1$. Consider changing the length of one link, say link $(11,10)$. Figure 3.5 shows the original directed acyclic network for destination 10 in the bold edges and the resulting routing probabilities. This topology changes when $d^k(11,10)$ changes; however, there are only two changes possible. Varying $d^k(11,10)$ affects $\sigma^k_{10}(11), \sigma^k_{10}(1)$, and $\sigma^k_{10}(2)$. Thus there are only three possible changes to the topology: when $\sigma^k_{10}(11) \geq 2$ link $(11,5)$ replaces $(5,11)$, when $\sigma_{10}(2) > 3$, link $(2,4)$ replaces $(4,2)$, when $\sigma_{10}(2) > 4$, link $(2,3)$ replaces $(3,2)$. These changes are denoted by the dotted edges in the figure. The choice of κ determines how much current changes to $d^k(11,10)$ affect routing topology by controlling the their affect on $\sigma^k_{10}(11)$. Figure 3.14 shows $\sigma^k_{10}(2)$ for various values of κ as $d(11,10)$ changes from 1 to 3 at 8, a step change. Figure 3.15 shows $\sigma^k_{10}(2)$ where $d^8(11,10) = d^9(11,10) = 3$ and $d^k(11,10) = 1$ otherwise, an impulse change.

We show the results from changes in Table 3.1 and in Figures 3.11,3.12,3.13.

3.7 Conclusion

We studied a probabilistic routing scheme we call soft routing as an alternative to shortest path routing when stability of network load is desired. This scheme limits the amount of rerouting required when path lengths change subject to a willingness to accept

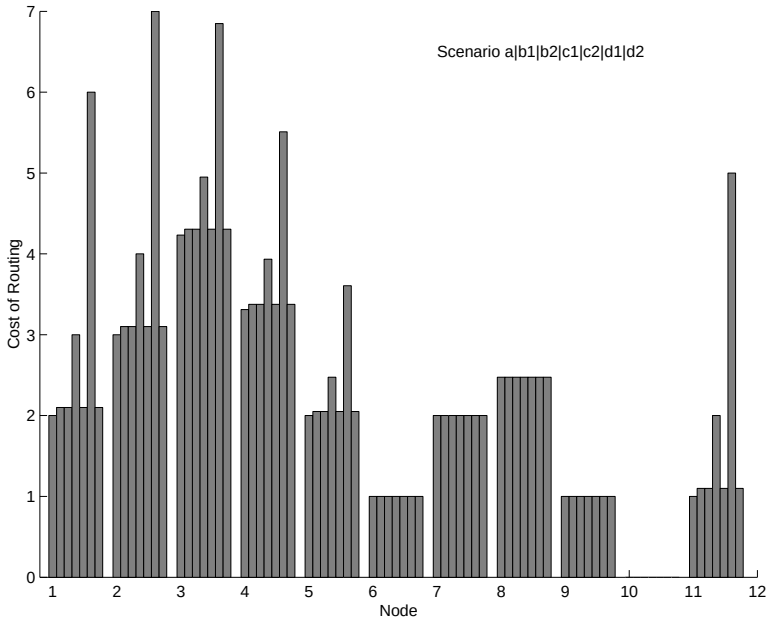

Figure 3.11: This figure shows the cost of routing from each node to destination 10 where $\alpha = 0.1$. On the x-axis there is a cluster of bars for each node. This cluster shows the cost of routing from that node to 10 for all the scenarios.

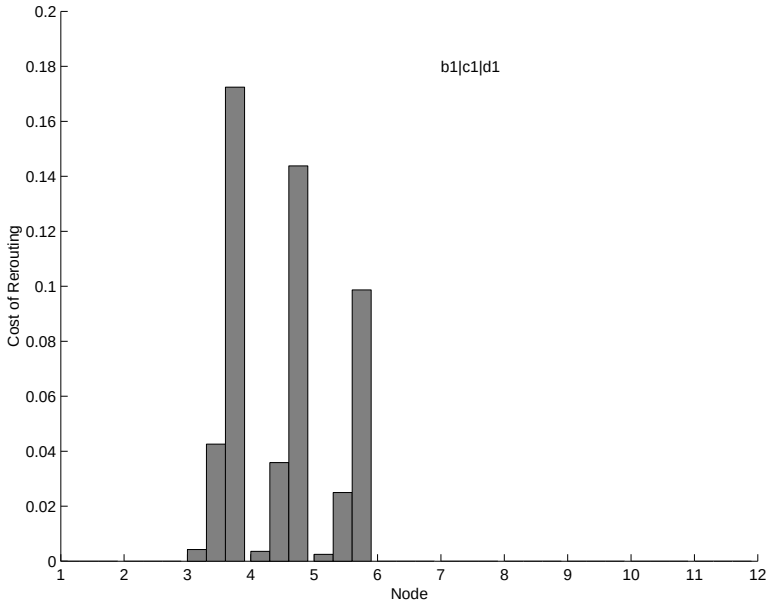

Figure 3.12: Cost of Rerouting from each Node to Destination 10 where $\alpha = 0.1$ and there are no topology changes. On the x-axis there is a cluster of bars for each node. This cluster shows the cost of rerouting from that node to 10 for the scenarios b1,c1, and d1.

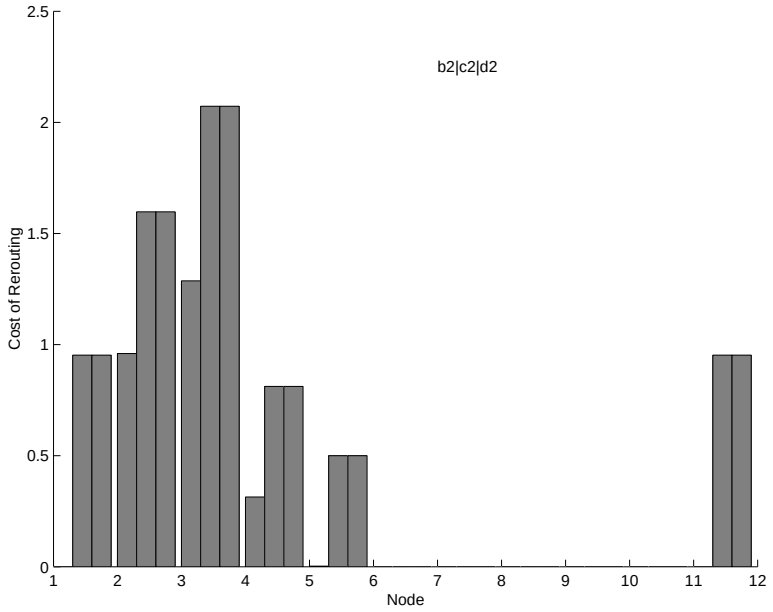

Figure 3.13: Cost of rerouting from each node to destination 10 where $\alpha = 0.1$ and there are topology changes. On the x-axis there is a cluster of bars for each node. This cluster shows the cost of rerouting from that node to 10 for the scenarios b2,c2, and d2.

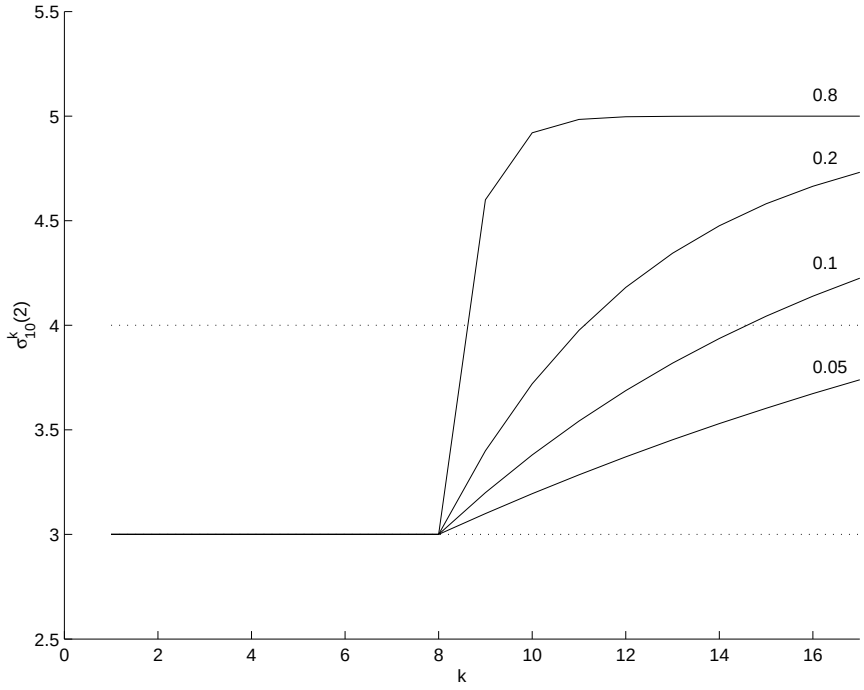

Figure 3.14: $\sigma_{10}^k(2)$ for Step Change

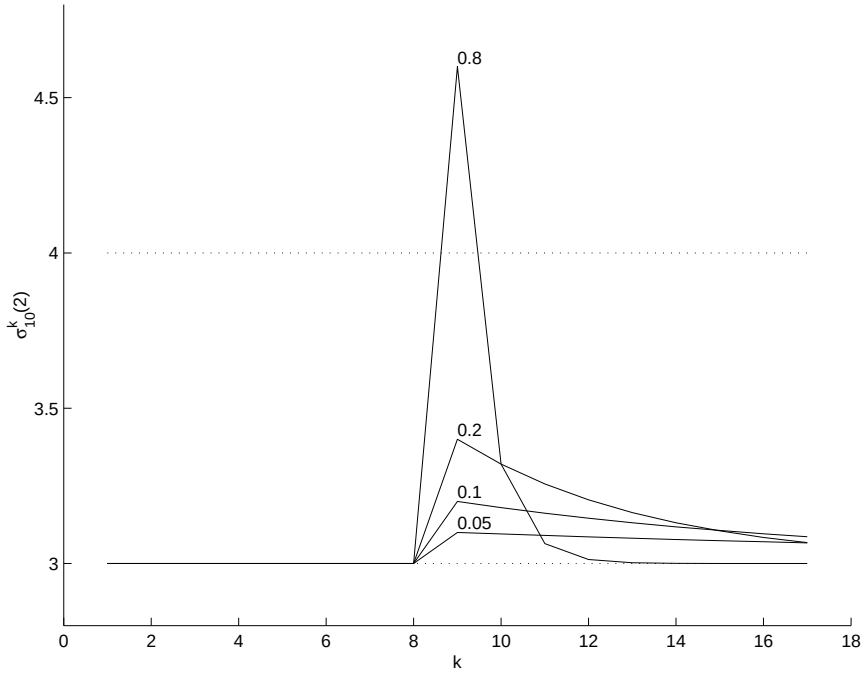

Figure 3.15: $\sigma_{10}^k(2)$ for Impulse Change

suboptimal routes. Here we considered the cost of routing in terms of path length. For future work we consider a metric based on link utilization.

Chapter 4

Clustering

4.1 Introduction

We study the impact of clustering on QoS routing over adhoc wireless networks. By clustering we refer to grouping stations to form a structure on the network. The structure allows for heirarchical routing as in today's wired networks.

Existing QoS routing strategies for wireless either ignore the interference effects ([31], [6]) or require detailed accounting of network resources ([22], [28], [26], [20], [19]). There has been much work involving theoretical models of available capacity ([17], [16], [10]). However, using these models to account for utilization requires communication amongst all stations that interfere. This communication is impossible if the interfering stations are disconnected. Thus detailed accounting of network resources is complex and sometimes impossible.

Instead we consider an approach based on the observation that differences in performance between interfering routes are typically negligible since they are limited by a common constraint. We study the use of clustering to introduce scalability as in wired networks. Moreover for wireless, the clustered network simplifies the computation of available resources: instead of choosing routes across links we consider routes through different clusters. So instead of checking the constraints on sets of individual links, we check constraints within each cluster. Checking a cluster's constraints makes sense since wireless constraints affect areas of the network rather than individual links.

This chapter describes a simple QoS routing scheme using OSPF over a clustered network. The purpose is to evaluate the impact of clustering on QoS routing. We evaluate

the increase in path length caused by clustering. We compare the performance of QoS routing algorithms over clustered and flat networks via simulations.

4.2 QoS Routing with Clustering

In this section, we describe a simple scheme that uses shortest path routing for intra-cluster routing and OSPF (Open Shortest Path First) routing for inter-cluster routing.

Ideally clustering forms non-overlapping sets of connected stations. The stations are geographically close and interfere so they share a common bottleneck constraint. This bottleneck constraint is less affected by interference from stations outside the cluster. Then clustering decouples the constraints across clusters, which simplifies resource accounting. For simplicity, we assume clusters are formed over a grid. That is, we super-impose a grid with intervals I over our network and each square in the grid is a cluster. Alternatively, we could have used one of many clustering schemes ([16], [27], [32], [2], [23]).

Since stations within a cluster share a common capacity constraint, routing through the cluster should minimize the access of the shared medium. We suggest using shortest path routing with weights equal to the reciprocal of the link speed to minimize the time of transmissions. This approach is confirmed later by simulations that show that within a small area shortest path routing gives approximately the same performance (see Section 4.3) as more sophisticated algorithms that require more overhead.

For the inter-cluster routing, consider an algorithm that works as follows. If the destination lies within the same cluster as the source, then the algorithm selects the shortest path to the destination. Otherwise, the inter-cluster routing selects a cluster-level path via OSPF to the cluster containing the destination station. From the current station, the algorithm chooses the shortest path to the next cluster in the cluster-level path. The algorithm then selects the shortest path connecting the current station to any station belonging to the next cluster. The process continues until reaching the cluster containing the destination station. In the final cluster, we use the shortest path to connect the current station the destination.

Any QoS routing scheme requires estimates of the available capacity which is the available channel utilization in wireless. We estimate a cluster's used capacity by estimating the channel utilization detected by a representative *leader station* that is centrally located in the cluster. Over some period (e.g. 5 seconds), the leader station measures the fraction

of time the channel is busy. The assumption is that the channel utilization in the current period is a good estimate for the channel utilization of the next period. The leader measures the fraction of time it transmits over the channel and the fraction of time it detects a carrier sense signal indicating that the channel is busy. For each cluster C let R_C be the estimate of the channel utilization from the last period. The clusters exchange their R_C-values with all other clusters via a link state protocol. In our simulations we use weights $W_C = R_C + \epsilon$ where ϵ is a constant factor so that shortest path routing is used when all measured R_C are zero. This W_C scales path segments based on the current channel utilization per cluster.

4.2.1 Overhead of Clustering

Cluster-level routing leads to improvement in scalability in terms of control bandwidth (fewer link state messages), storage, and computation. However, requiring a coarse level of granularity results in a less efficient use of network resources because of the following effects. First, the routes over a clustered network are longer than routes not constrained by the clustering. Second, the routes over a clustered network result in more self-interference. Third the cluster-based routing algorithm wastes available capacity by restricting the choice of the paths: instead of considering all possible paths, we limit the choices to cluster-level paths. Here we look at the first and second effects, the third is considered later via simulations.

We compare routing performed over a flat network against that performed at a cluster level with intracluster routing using shortest path. Compare the path P_F returned by OSPF operating over a flat network with the path P_C returned by OSPF operating over the cluster level topology. Consider an idle network in which no flow has yet been routed. Within such a network, OSPF is just shortest path routing. We analyze the impact of clustering.

Consider the following continuous network model. Take a rectangle $[0, X] \times [0, Y]$. Any point in this rectangle is a station. Thus any continuous path is a viable route between two points. Recall that we cluster the network into a grid of squares. For simplicity, assume that the grid intervals are of interference radius r and X and Y are multiples of r. This continuous model approximates a network with a large station density.

In an idle network, the path P_F is the straight line connecting the source $s = (x_s, y_s)$ and the destination $d = (x_d, y_d)$. We determine path P_C by first finding the short-

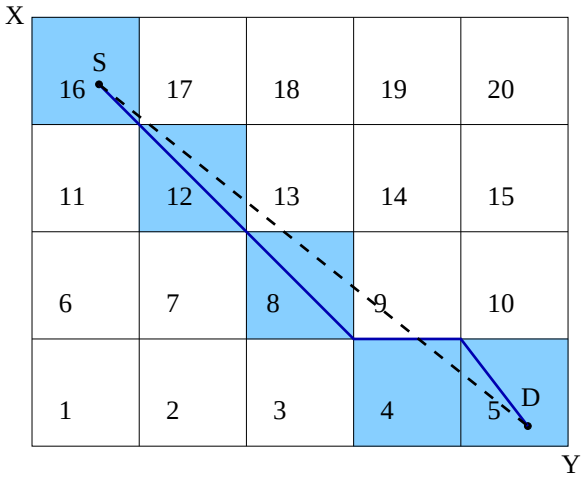

Figure 4.1: **Path Length Comparison** The dotted line shows P_F. The cluster level path is 16-12-8-4-5. We form P_C by using shortest path segments to connected the cluster path. The solid line shows P_C.

est cluster-level path and then connecting that path using shortest path segments. If s and d belong to the same grid square, then P_C is a straight line giving the same result as P_F. Otherwise, s belongs to some grid square G_s and d to some grid square $G_d \neq G_s$. The shortest cluster-level path is the set of connected squares from G_s to G_d (say $G_{i_1}, G_{i_2}, \ldots, G_{i_n}$ where $i_1 = s$ and $i_n = d$). We connect the clusters to form the station-level path. Initially $p = s$. We use the shortest path from $p = s$ to the closest point $q \in G_{i_2}$. Then update $p = q$. While $p \in G_{i_j} \neq G_d$, use the shortest path from p to $q \in G_{i_{j+1}}$ and update $p = q$. Finally use the shortest path from p to d. See the example in Figure 4.1.

Typically the intracluster connectivity consists of horizontal, vertical, and diagonal edges connecting the corners of the grids. Let $x = |x_s - x_d|$ and $y = |y_s - y_d|$. Then shortest path at the cluster level results in a bound $P_C \leq \sqrt{2}\min\{x, y\} + |x - y|$ since we can use diagonal transitions with either strictly vertical or strictly horizontal transitions. Taking $x = P_F \sin\theta$ and $y = P_F \cos\theta$ have that $P_C \leq 1.0825 P_F$.

Unlike P_F, P_C may not be straight. Hence P_C uses more capacity along its route because there is more self-interference. Assume the transmission of station at point (a, b) forms a unit disk interference region of radius r. In shortest path, the points within the middle of P_F experience the most interference. These points require the most capacity when establishing the route. Take the station at point (a, b) on P_F. If the amount of interference

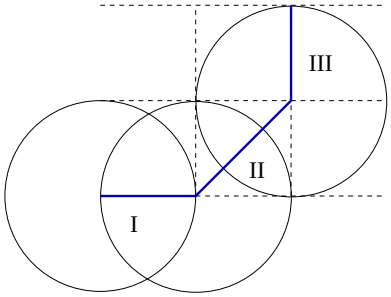

Figure 4.2: Capacity Comparison: Worst Case 1

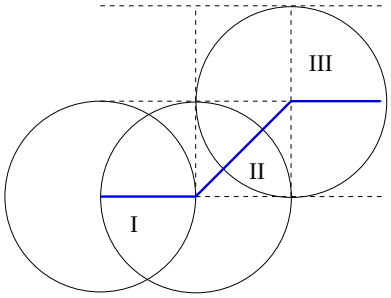

Figure 4.3: Capacity Comparison: Worst Case 2

experienced at (a, b) is proportional to the number of stations whose interference region includes (a, b), then the maximum the amount of capacity used by P_F is proportional to $2r$.

In the clustered case, certain stations experience more interference from the path itself. A vertex station requires capacity proportional to $2r$. The worst case is as in Figures 4.2 and 4.3 at the midpoint of II. The midpoint experience interference proportional to $\sqrt{2}r$ from II, and $\sqrt{r^2 - (\frac{r}{2})^2}$ from I and from III. Thus the maximum capacity required by P_C is proportional to $(\sqrt{3} + 1)r$.

4.3 Simulations

We validate our cluster-level routing scheme via flow level simulations using Matlab [40]. At the flow level, we must approximate the behavior for the admission control and the measurements. Given a set of flows, [19] uses clique constraints and global information to check for the existence of a feasible schedule that achieves the rates of the flows. We

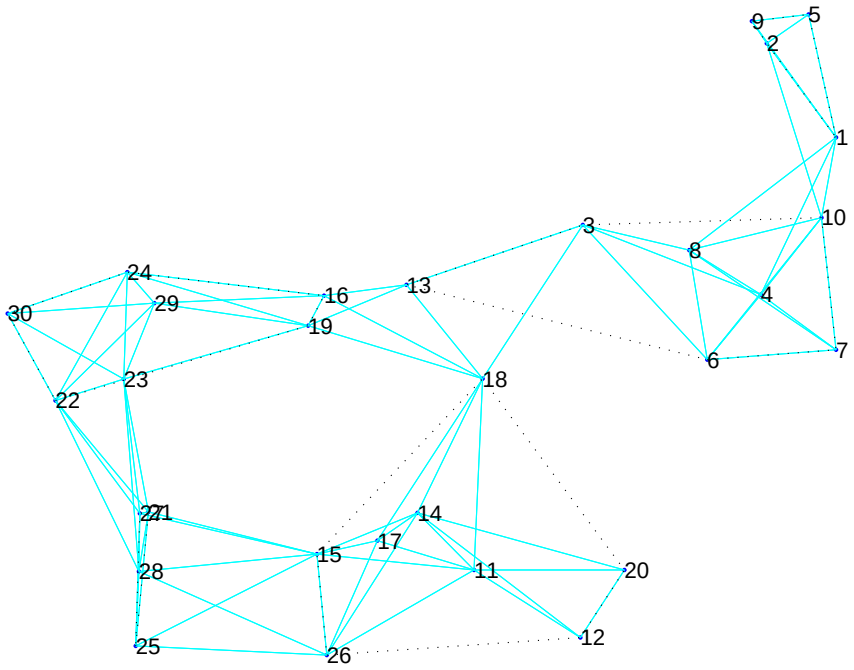

Figure 4.4: Example Network

assume the network admits a flow if there exists such a feasible schedule. First we check the performance of shortest path for intracluster routing over small areas. Then we compare OSPF over the clustered network with OSPF over a flat network.

4.3.1 Small Area

First consider a small dense network as in Figure 4.4. These stations span a square whose side is twice the interference radius. For this small area, we study the effects of routing when all routes exist over a shared interference area. We run traffic scenarios with the following parameters.

- *S-D (Source-Destination) Pairs:* We use models with demands that are uniform across all s-d pairs and demands that are uniform over fixeds sets of s-d pairs. Limiting the demand to a fixed number of s-d pairs checks the routing algorithm's ability to discover and use longer paths with less penality to future flows.

- *Flow Size:* We use models with flows of size 10kbps (from voice) or 100kbps (from video). The scenarios with small flows show how the algorithms find a route since the impact of the flow is small. However with large flows, few routes exist. The scenarios with large flows focus on how the algorithms choose a route since the impact of the flow is large.

- *Network Load* We generate flows with exponential interarrivals and exponential durations. We manipulate the mean interarrival and duration times to generate loads of $2, 4$, and 8Mbps on the network.

We compare the following routing strategies: Shortest Path (SP), OSPF, Integer Linear Program (ILP). SP returns the path with the minimum number of hops. Recall that a link's utilization is actually the channel utilization and includes the usage of interfering transmissions. We use this channel utilization as the link weights for OSPF and the constraints of the ILP. We compute the capacity constraints using global information for the OSPF weights and the ILP constraints. Figure 4.5 compares these algorithms under various traffic models. For a given traffic model, the admission ratio of SP gives the x-coordinate; the y-coordinate is the total improvement in the admission ratio of OSPF or ILP over SP. The difference with SP is typically 10%. The difference may be even more minimal since we do not model the link state exchanges needed for OSPF and ILP. These simulations verify that within a small area of the network SP gives approximately the same performance. The improvements over SP occur when the network can load balance on multiple paths. However in small areas, interference effects result in few independent paths. These results validate our use of shortest path routing at the intracluster level.

4.3.2 Large Area

To see the effects of clustered routing we consider topologies over a larger area of the network. These topologies range over multiple independent interference areas. We consider topologies of a hundred stations with different station densities.

- *Grid3* is a random distribution of stations over a 3×3 grid.

- *Grid4* is a random distribution of stations over a 4×4 grid.

- *Mesh* is a mesh pattern topology.

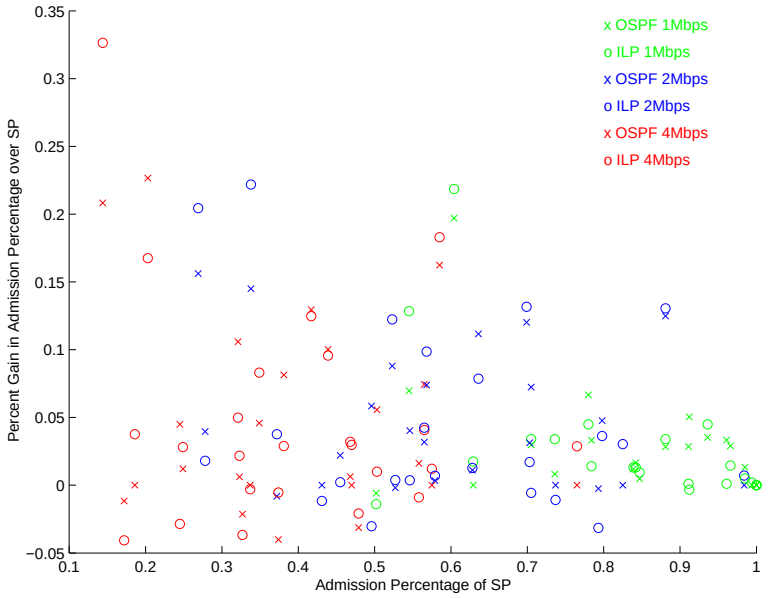

Figure 4.5: Percentage of improvement in admittance percentage over SP under various loads

The transmission radius 0.5, and the interference radius is $r = 1$. For all topologies, we cluster by squares of size $r = 1$. Figures 4.6 and 4.9 show the Grid3 and Mesh topologies respectively with the clusters.

Over these larger topologies, propagation of link state information becomes a factor. Previously, we assumed all computations were performed using global information. Now we also model the propagation time to disseminate updated information. We use an event driven simulator where each station has a current local view of the network. A station's local view updates the new event according to the number of hops the station is from the event. We compare the performance of OSPF for the following.

- *G-OSPF*: OSPF computed over a flat network using global information
- *E-OSPF*: OSPF computed over a flat network using local information under the event driven simulator
- *CL-OSPF*: OSPF computed at the cluster-level using local information under the event driven simulator. We use SP routes at the intracluster level.

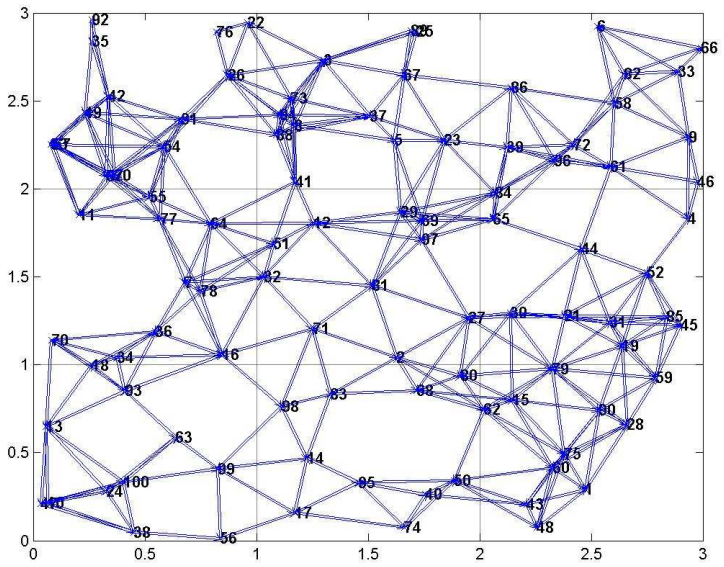

Figure 4.6: Grid3 Topology

Random Traffic Pattern

First we present results for the random traffic patterns used in 4.3.1. Figure 4.7 shows comparisons of average path lengths. We plot the average path length of E-OSPF and CL-OSPF against the average path length of G-OSPF. Resulting average path lengths are very close confirming our analysis. Clustered routing is comparable to flat routing in terms of path length.

Figure 4.8 shows comparisons of admission ratios. We plot the admission ratios of E-OSPF and CL-OSPF as a function of the admission ratio of G-OSPF. Typically the admission ratio from flat routing is a total improvement of 10% over the admission ratio from clustered routing. The margin seems approximately the same for all topologies. One imagines that as network area increases the number of clusters increase. Then the number of choices over the clustered network is close to the number over the flat network. However the effect of increasing the area is not immediately evident here. Perhaps a more limiting effect is that clustering decreases path diversity. We try to understand the limits in path diversity in a more detailed study for an individual s-d pair below.

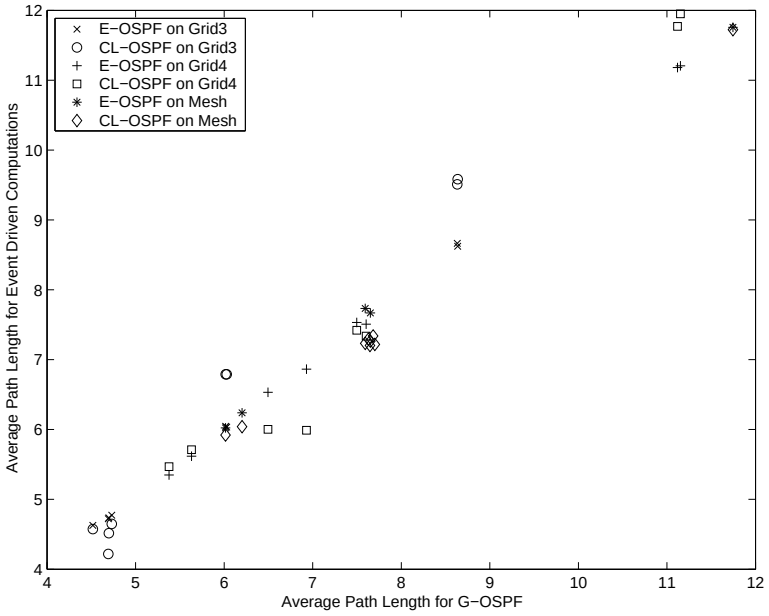

Figure 4.7: Average Path Length

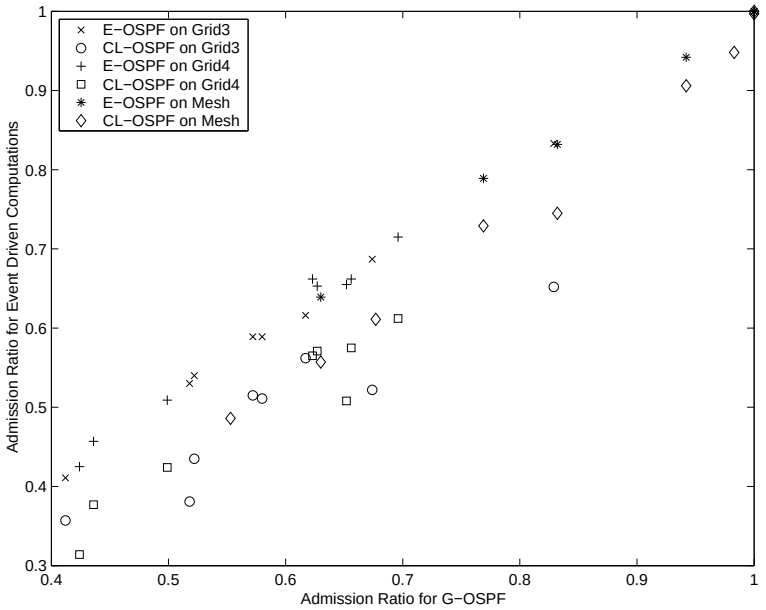

Figure 4.8: Admittance Ratio

Source-Destination Pair Traffic

Here we focus on the total capacity of the network for a given s-d pair. These simulations compare the ability of the clustered computation to meet the capacity attained by flat computation. These simulations start with an idle network and route as many flows for a given s-d as possible. We compute a route and then add if there is enough capacity and continue; if there is not enough capacity along the computed route, then we stop. This behavior should be possible with an admission control algorithm similar to that suggested in Section 2.6. An admitted flow remains for the duration of the simulation. Below we compare the results from all s-d pairs in terms of the average path length, the number of unique paths, and the number of admitted flows.

We begin with an example showing the routes discovered by CL-OSPF. Figure 4.9 shows the set of paths found by CL-OSPF connecting stations 10 and 71 on the mesh topology. Notice that the links across clusters do not interfere. It seems that the 35 different paths used by CL-OSPF covers most of the independent interference regions in the network. However, CL-OSPF admits 94 flows as compared with E-OSPF which admits 129. We credit CL-OSPF's loss of capacity to the missing transmissions through 74 and 79. Here the cost of clustering lies in CL-OSPF's failure to discover all paths. This failure occurs since clustering forces the routing to conform to the shortest path segment through a cluster.

Figures 4.10 and 4.11 compare the differences in the average number of different paths found for s-d pairs of some hops away. Typically a large number of different paths occur in the flat computation, but these paths do not represent the actual number of independent paths. Figure 4.14 shows the average path lengths.

Figures 4.12 and 4.13 show the average number of admitted flows for s-d pairs of some hops away. The best performance of CL-OSPF occurs when s-d are nearby or far. When s-d are close, few path choices exist. When s-d are far, the cluster level has enough path diversity to achieve close to the flat result. With enough path diversity, missing a few links as in the example for 10-71 is not as costly.

We find comparable use of network resources in terms of path length between clustered and flat routing. The inefficiency in clustered routing lies in discovering available paths since there is less diversity of available paths. However, using less diversity requires less update information.

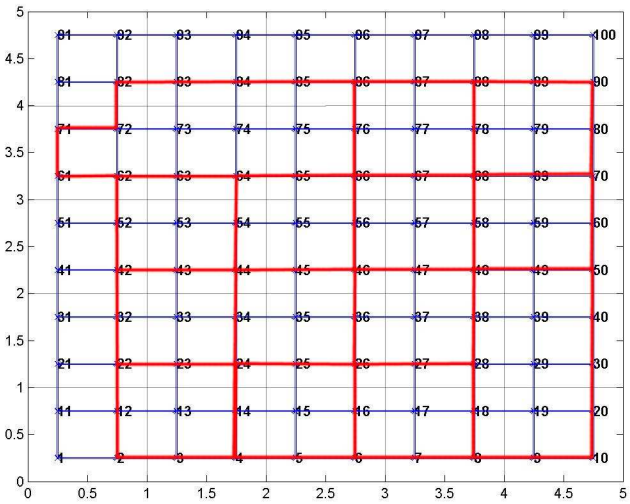

Figure 4.9: Routes from CL-OSPF for 10-71 on Mesh Topology

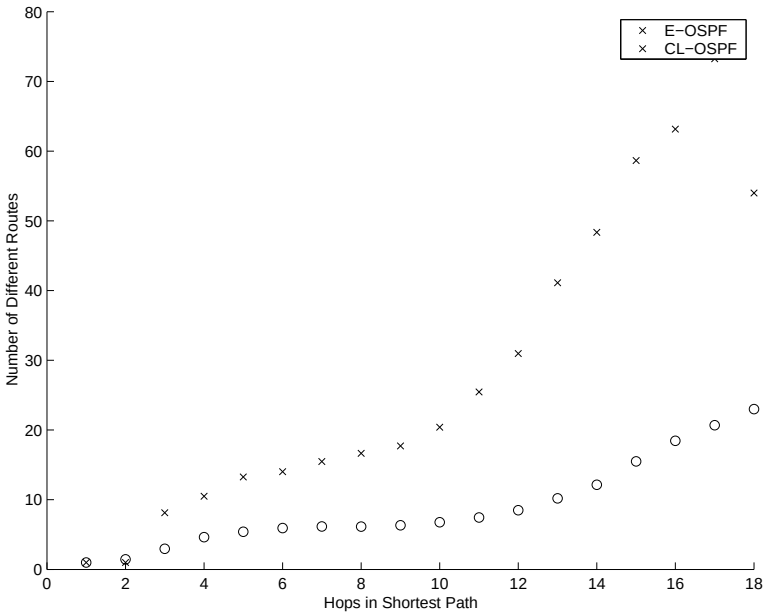

Figure 4.10: Number of Unique Paths for Mesh Topology

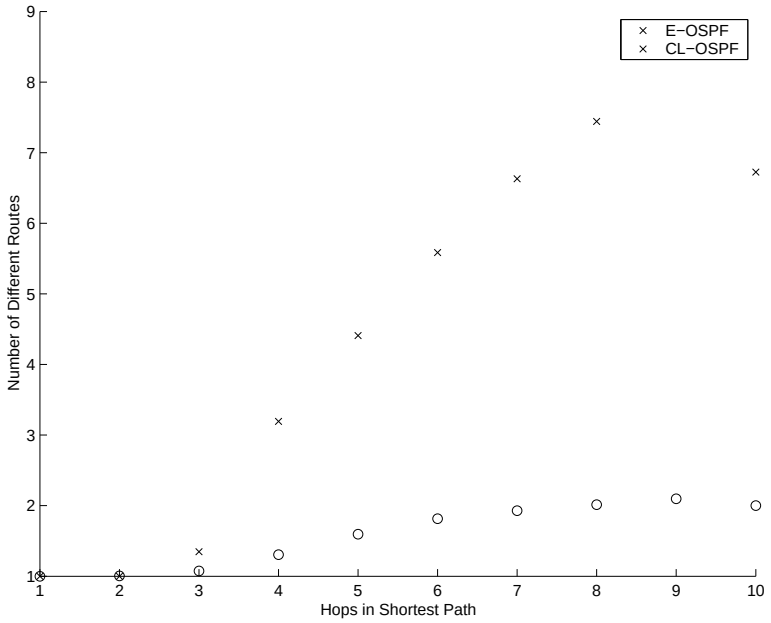

Figure 4.11: Number of Unique Paths for Grid3 Topology

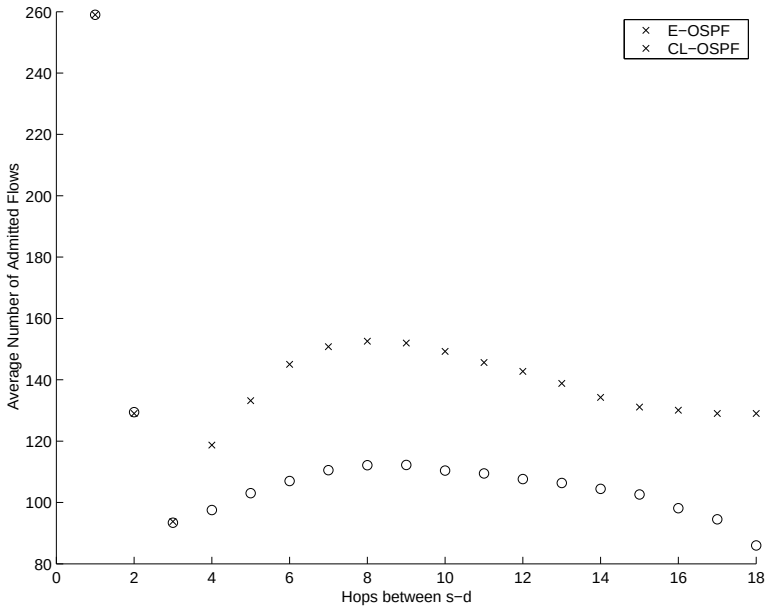

Figure 4.12: Average Number of Admitted Flows for Mesh Topology

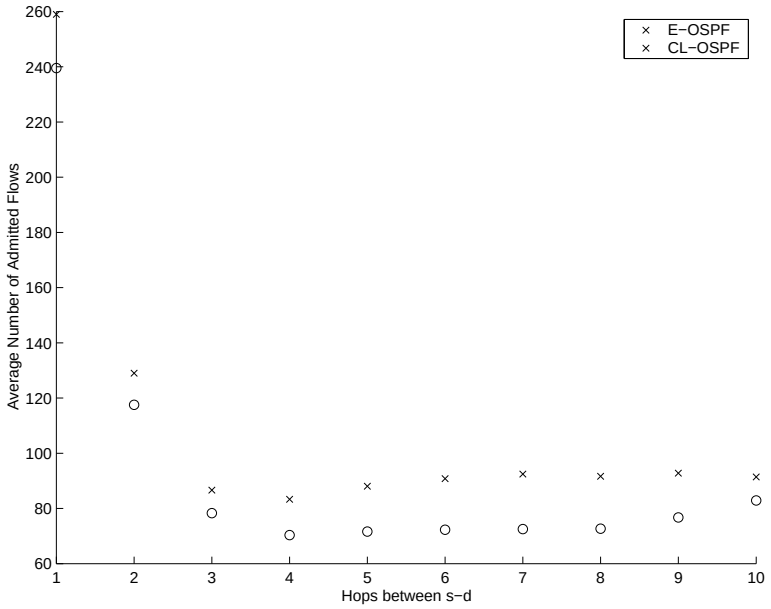

Figure 4.13: Average Number of Admitted Flows for Grid3 Topology

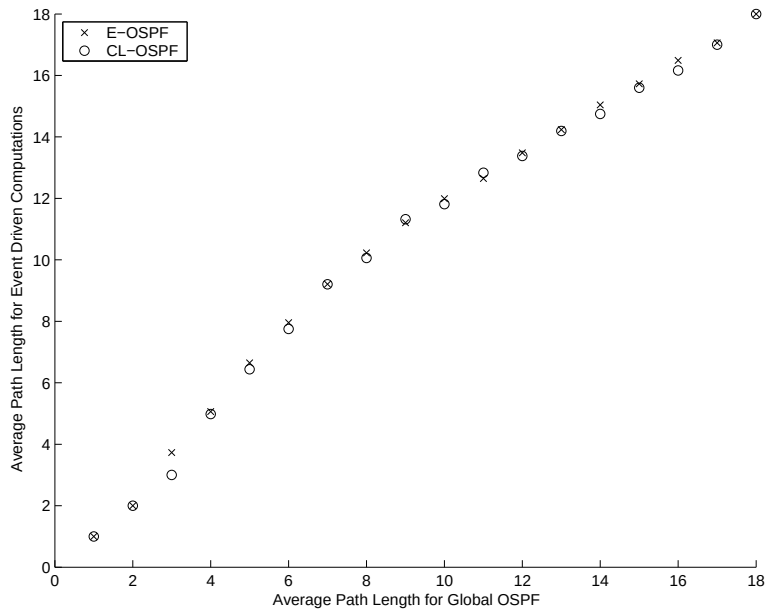

Figure 4.14: Average Path Length for Mesh Topology

4.4 Conclusion

We studied when QoS routing using OSPF operates over a clustered network. This scheme is based on the observation that interfering routes share the same capacity constraints offering little difference in performance. Instead, the choice exists at a coarser level of granularity where it makes sense to consider the network as clusters with decoupled interference domains. We use a continuous network model to bound the costs of clustering in terms of longer path length and self-interference. We use simulations to compare OSPF over a flat network with that over a clustered network. Routing over a clustered network gains in scalability, but loses capacity due to increases in path length, increases in path interference, and loss of path diversity. However using a clustered network allows for simple strategies for routing and for estimating available capacity.

Bibliography

[1] F. Anjum, M. Elaoud, D. Famolari, A. Ghosh, R. Vaidyanathan, A. Dutta, P. Agrawal, T. Kodama, and Y. Katsube, "Voice performance in WLAN networks - an experimental study," in *Proc. IEEE Globecom*, 2003.

[2] S. Basagni, "Distributed Clustering for Ad Hoc Networks," *Proceedings of the IEEE International Symposium on Parallel Architectures, Algorithms, and Networks (I-SPAN)*, Perth, Western Australia, June 1999, pp. 310-315.

[3] G. Bianchi, "Performance analysis of the IEEE 802.11 distributed coordination function," *IEEE J. Select Areas Communications*, vol. 18, no. 3, pp. 535-547, 2000.

[4] L. Bononi, M. Conti, and E. Gregori, "Runtime optimization of IEEE 802.11 wireless LANs performance," *IEEE Transactions on Parallel and Distributed Systems*, vol. 1, pp. 66-80, 2004.

[5] J. Brumbaugh-Smith and D. Shier, "Minimax Models for Diverse Routing." INFORMS Journal on Computing, 14, 2002.

[6] S. Chen and K. Nahrstedt, "Distributed quality-of-service routing in ad-hoc networks," *IEEE Journal Selected Areas in Communication*, vol. 17 no. 8, pp. 1488?1505, Aug 1999.

[7] C. Chuah and R. Katz, "Network Provisioning and Resource Management for IP Telephony," Report No. UCB/CSD-99-1061, August 1999.

[8] P. Clifford, K. Duffy, D.J. Leith and D. Malone. "On improving voice capacity in 802.11 infrastructure networks," *IEEE WirelessCom 2005*, 214 - 219, June 13-16, Maui, Hawaii, USA.

[9] T. Cormen, C. Leiserson, and R. Rivest, /it Introduction to Algorithms, The MIT Press, 1990.

[10] A. El Gamal, J. Mammen, B. Prabhakar, and D. Shah, "Throughput-Delay Trade-off in Wireless Networks," *Proceedings IEEE INFOCOM*, Hong Kong, March 2004. http://fleece.ucsd.edu/ massimo/Journal/IEEE-TIT-Capacity-Submission.pdf

[11] M. Elaoud and P. Agrawal, "Voice capacity in IEEE 802.11 networks," in *Proc. IEEE PIMRC*, 2004.

[12] M. Ergen and P. Varaiya, "Understanding the Markov Model Analysis for Throughput in IEEE 802.11 DCF Mode," submitted to *IEEE Trans. On Networking.*

[13] S. Garg and M. Kappes, "Can I add a VoIP call?," in *Proc. IEEE ICC*, 2003, pp. 779.783.

[14] S. Garg and M. Kappes, "An Experimental Study of Throughput for UDP and VoIP Traffic in IEEE 802.11b Networks," in *Proc. IEEE Wireless Communications and Networking Conference*, 2003.

[15] R. J. Gibbens and F. P. Kelly, "Distributed connection acceptance control for a connectionless network," http://www.statslab.cam.ac.uk/ frank/dcac.html

[16] M. Grossglauser and D. Tse, "Mobility Increases the Capacity of Ad-hoc Wireless Networks," *IEEE/ACM Transactions on Networking* vol. 10, no. 4, pp. 477-486, August 2002.

[17] P. Gupta, and P. R. Kumar, "The Capacity of Wireless Networks," *IEEE Transactions on Information Theory,* vol. 34, no. 5, pp. 910-917, 2000.

[18] P. Gupta and P. R. Kumar, "A System and Traffic Dependent Adaptive Routing Algorithm for Ad Hoc Networks," *Proceedings of the 36th IEEE Conference on Decision and Control*, 2375-2380, San Diego, December 1997.

[19] R. Gupta, J. Musacchio, and J. Walrand, "Sufficient Rate Constraints for QoS Flows in Ad-Hoc Networks", *Submitted to INFOCOM 2005. Available at http://www.eecs.berkeley.edu/∼guptar/RGpublications.html.*

[20] R. Gupta and J. Walrand, "Approximating Maximal Cliques in Ad-Hoc Networks", *Proc. PIMRC 2004*, Barcelona, Spain, September 2004.

[21] N. Hedge, A. Proutiere, and J. Roberts, "Evaulating the voice capacity of 802.11 WLAN under distributed control," in *Proc. LANMAN*, 2005.

[22] K. Jain, J. Padhye, V. N. Padmanabhan, and L. Qiu, "Impact of Interference on Multi-hop Wireless Network Performance," *Proceedings ACM Mobicom 2003*, San Diego, CA, USA, September 2003.

[23] P. Krishna, N. H. Vaidya, M. Chatterjee, and D. K. Pradhan. "A cluster-based approach for routing in dynamic networks." *ACM SIGCOMM Computer Communication Review*, pages 49–65, April 1997.

[24] R. Krishnan and J. Silvester, "An Approach to Path-Splitting in Multipath Networks," *Proceedings of the ICC*, 13583-1357, 1993.

[25] P. R. Kumar and P. Varaiya, *Stochastic Systems: Estimation, Identification, and Adaptive Control*, Prentice Hall Inc., 1986.

[26] J. Li, C. Blake, D. S. J. De Couto, H. I. Lee, and R. Morris, "Capacity of Ad Hoc Wireless Networks," *Proceedings ACM Mobicom 2001*, Rome, Italy, July 2001.

[27] C. R. Lin, M. Gerla, "Adaptive Clustering for Mobile Wireless Networks," *IEEE Jour. Selected Areas in Communications*, pp. 1265-1275, Sept. 1997.

[28] H. Luo, S. Lu, and V. Bhargavan, "A New Model for Packet Scheduling in Multihop Wireless Networks," *ACM Journal of Mobile Networks and Applications (MONET)* vol. 9, no. 3, June 2004.

[29] D. Malone, K. Duffy and D.J. Leith, "Modeling the 802.11 distributed coordination function in non-saturated heterogeneous conditions." To appear in *IEEE/ACM Transactions on Networking (2007).*

[30] J. W. Robinson and T. S. Randhawa, "Saturation throughput analysis of IEEE 802.11e enhanced distributed coordination function," in *IEEE J. Select. Areas Communications,* vol. 22, no. 5, pp. 917-928, June 2004.

[31] E. M. Royer, C. Perkins, and S. R. Das, "Quality of Service for Ad-Hoc On-Demand Distance Vector Routing," *Internet Draft draft-ietf-manet-aodvqos-00.txt*, July 2000.

[32] R.Sivakumar, B.Das, and V.Bharghavan, "Spine Routing in Ad Hoc Networks", *ACM/Baltzer Cluster Computing Journal*, 1998.

[33] T. Tung and J. Walrand. "Providing QoS for Real-Time Applications." *Proceedings of IASTED CIIT*, Scottsdale, AZ, Nov. 2003.

[34] A. Veres, A. T. Campbell, M. Barry, and L.-H. Sun, "Supporting service differentiation in wireless packet networks using distributed control," *IEEE Journal on Selected Areas of Communication*, vol. 19, no. 10, pp. 2081-2093, Oct. 2001.

[35] Jean Walrand and Pravin Varaiya, *High-Performance Communication Networks, Second Edition*, Morgan Kaufmann Publishers, 2000.

[36] Y. Xiao, "Performance Analysis of Priority Schemes for IEEE 802.11 and IEEE 802.11e Wireless LANs," *IEEE Transactions on Wireless Communications*, Vol. 4, No. 4, Jul. 2005. pp.1506- 1515.

[37] http://abilene.internet2.edu/

[38] IEEE Std 802.11, .IEEE Standard for Wireless LAN Medium Access Protocol and Physical Layer Specifications,. Aug. 1999.

[39] Madwifi. http://sourceforge.net/projects/madwifi

[40] www.mathworks.com