TECHNIQUES TO MITIGATE TRAFFIC

OVERLOAD AND PROTOCOL INEFFICIENCIES IN

MOBILE AD HOC NETWORKS

APPROVED BY SUPERVISING COMMITTEE:

Dr. Rajendra V. Boppana, Supervising Professor

Dr. Turgay Korkmaz

Dr. Ali S. Tosun

Dr. Weining Zhang

Dr. Robbert Hiromoto ,Outside Member

Accepted:

Dean of Graduate School

Dedication

I would like to dedicate this dissertation to my late father Dr. Inter De Silva, my mother Dulice De Silva, and my brother and sisters. Without their love, support, advice, motivation, and encouragement, I could not have completed this dissertation. I am especially grateful to my Brother-in-law Ranjan De Silva for providing financial assistance for my higher education; without his initial support, I could have not accomplish what I have accomplished.

TECHNIQUES TO MITIGATE TRAFFIC OVERLOAD AND PROTOCOL INEFFICIENCIES IN MOBILE AD HOC NETWORKS

by

SAMAN AMENDRA DESILVA, B.S., M.S.

DISSERTATION
Presented to the Graduate Faculty of
The University of Texas at San Antonio
in Partial Fulfillment
of the Requirements
for the Degree of

DOCTOR OF PHILOSOPHY IN COMPUTER SCIENCE

THE UNIVERSITY OF TEXAS AT SAN ANTONIO
College of Sciences
Department of Computer Science
December 2004

Acknowledgements

I would like to express my deep gratitude to Dr. Rajendra V. Boppana for his guidance, support, suggestions, and the endless number of hours spent in editing this dissertation. I would also like to thank Dr. Turgay Korkmaz, Dr. Ali S. Tosun, Dr. Weining Zhang, and Dr. Robert Hiromoto for serving on my final dissertation committee and providing valuable comments and constructive criticism. I would like to thank Dr. Samir Das, my first advisor, for drawing my interest in ad hoc networking research, his support, and his guidance during the early stages of my doctoral studies. I would like to extend a special thanks to Dr. Robert Hiromoto for all of the encouragement, support, and advice he has provided from the begin to the end. I also would like to thank everyone who proofread my dissertation and provided me with valuable comments and corrections. A special thanks goes to my supervisors Dr. Robert Hiromoto and Dr. Larry Williams for giving me the opportunity and flexibility to attend college while working.

This research was partially supported by AFOSR grant F49260-96-1-0472 and NFS MII grant CDA-9633299.

December 2004

TECHNIQUES TO MITIGATE TRAFFIC OVERLOAD AND PROTOCOL INEFFICIENCIES IN MOBILE AD HOC NETWORKS

Saman Amendra Desilva, Ph.D.
The University of Texas at San Antonio, 2004

Supervising Professor: Dr. Rajendra V. Boppana

A mobile ad hoc network (MANET) is a collection of wireless devices moving in seemingly random directions and communicating with one another without the aid of an established infrastructure. Multi-hop communication between two distant nodes is achieved by other similar nodes acting as intermediate routers. Since there is no inherent traffic admission control, MANETs are likely to operate in a congested state, due to high traffic overload.

Owing to shared, half-duplex wireless channels and frequent node mobility, establishing and maintaining routes among communicating nodes is a challenge. Several routing protocols have been proposed specifically for ad hoc networks to handle frequent route breaks. However, extensive studies in literature and our own work indicate that the ad hoc routing protocols designed to excel, for example, in dense, low-mobility networks do not work well in sparse, high-mobility networks. Also, routing protocols may perform well at low to medium loads, prior to network saturation, but they do not sustain their performance for high loads.

This dissertation addresses these problems and provides several general techniques that can be used to augment existing routing protocols by changing the logic or algorithm used. We also provide general solutions that can correct an existing protocol's deficiencies without modifying the existing protocol. These solutions are implemented at the medium access control (MAC) sub layer and can be beneficial to many on-demand routing protocols.

We find that the IEEE 802.11 MAC protocol, as defined, is overcautious and prevents even lightly exposed nodes from communicating with their neighbors. Our modification to mitigate this weakness improves the throughputs of commonly used on-demand routing protocols by 10-20%.

As the network load increases, the packet delays increase, and routes are broken frequently due to

exposed nodes. The former increases the route repair time, and the latter increases the control overhead, especially for on-demand routing protocols that use network-wide flooding to repair routes. We illustrate this for the most commonly used routing protocol, AODV, which loses throughput beyond saturation.

To mitigate the rapid loss of throughput by AODV, we propose two solutions: (a) removing duplicate control packets waiting at the MAC layer for transmission, and (b) a dynamically adaptive route repair timer to estimate the time needed to repair a route. The latter requires protocol changes, while the former does not. Using simulations, we show that both techniques enable AODV to perform gracefully under traffic overload conditions.

Under traffic overload, the number of false route breaks — due to temporarily non-response intermediate node in routing paths — is high. While the next hop is in the communication range, route breaks caused by transmission failures are called false route breaks. These false route breaks increase the routing overhead and cause performance degradation after the point of saturation is reached. Similarly, some routing protocols are prone for stale routes. Data packets that use stale routes consume network bandwidth for a few hops before they reach a dead-end and they are dropped, creating real route breaks. To reduce the false and real route breaks, we propose next hop status prediction techniques. Using prediction prior to transmission (pre-prediction) can reduce the effects of real route breaks. When the prediction is used after a transmission failure, the number of false route breaks may be reduced. Using two on-demand routing protocols, the AODV and the DSR, we show that our prediction schemes are effective, especially in high-traffic loads.

Table of Contents

Dedication . . . . . **ii**

Acknowledgements . . . . . **iv**

Abstract . . . . . **v**

List of Tables . . . . . **x**

List of Figures . . . . . **xi**

1 Introduction **1**
1.1 Point-to-Point Data Communication 2
1.1.1 Satellites data communication 2
1.1.2 Point-to-point fixed wireless 3
1.2 Cellular Networks 3
1.2.1 Wireless local area networks (WLAN) 5
1.3 Ad Hoc Networks 6
1.3.1 MANET routing 7
1.3.2 MAC protocol 8
1.4 Challenges 10
1.5 Motivation 10
1.5.1 MAC protocol 13
1.5.2 Routing protocol 14
1.5.3 Broadcast management 14
1.5.4 Route maintenance 15
1.6 Contributions of This Dissertation 15
1.6.1 Improving 802.11 MAC protocol design 15
1.6.2 Broadcast management in MANET routing protocols 16
1.6.3 Prediction schemes for route management 17
1.6.4 MAC layer scheduling 17
1.7 Organization of the Dissertation 18

2 Background **19**
2.1 Ad Hoc Networks 21
2.2 Ad Hoc Routing Protocols 22
2.2.1 Ad Hoc On-Demand Distance Vector (AODV) 22
2.2.2 The Dynamic Source Routing Protocol for Mobile Ad Hoc Networks (DSR) 24
2.2.3 Optimized Link State Routing Protocol (OLSRP) 25
2.2.4 Topology Dissemination Based on Reverse-Path Forwarding (TBRPF) 26

2.3 MAC Protocols . . . 27
2.3.1 Carrier Sense Multiple Access (CSMA) protocol . . . 29
2.3.2 Multiple Access Collision Avoidance (MACA) protocol . . . 29
2.3.3 Flow Acquisition Multiple Access (FAMA) protocol . . . 30
2.3.4 MACAW: Media access protocol . . . 30
2.3.5 802.11 MAC Protocol . . . 30
2.4 Performance Analysis . . . 32
2.4.1 The Network Simulator . . . 33
2.4.2 Glomosim . . . 33
2.4.3 OPENT . . . 35
2.4.4 Performance metrics . . . 36

3 On the Impact of Noise Sensitivity in 802.11 MAC Protocol 37
3.1 Modeling Signal Strength . . . 38
3.2 Communication Regions . . . 42
3.3 Impact of Competing Transmissions . . . 44
3.4 Collision Distance . . . 52
3.5 Modification to the 802.11 Protocol . . . 54
3.6 Performance Analysis . . . 55
3.6.1 Four-node network . . . 55
3.6.2 Static linear chain . . . 58
3.6.3 Static grid . . . 59
3.6.4 Mobile ad hoc network . . . 59
3.7 Related Work . . . 66
3.8 Conclusions . . . 67

4 Sustaining Performance Under Traffic Overload 68
4.1 Solutions to Mitigate Underperformance . . . 69
4.2 Behavior of Mobile Ad Hoc Networks Beyond Saturation . . . 71
4.3 Reducing RREQ Explosion Using Efficient Broadcast Delivery . . . 73
4.3.1 Random Assessment Delay (RAD) . . . 73
4.3.2 Scalable Broadcasting Algorithm (SBA) . . . 74
4.3.3 Performance of broadcast management schemes . . . 75
4.4 Combining RADe with Modified MAC Protocol . . . 76
4.5 Reducing Unnecessary Control Packets . . . 77
4.5.1 Reduced broadcasts with Modified 802.11 MAC protocol . . . 78
4.6 Dynamic Hop Time (DHT) . . . 80
4.6.1 Estimating RREP time . . . 81
4.7 Comparing Broadcast Reduction Techniques . . . 84
4.7.1 Combining multiple techniques . . . 86
4.8 Conclusions . . . 86

5 Next Hop Prediction Techniques 88
5.1 Classification of Prediction Schemes . . . 90
5.1.1 Pre-transmission prediction . . . 91
5.2 Basic Prediction Criteria . . . 93
5.2.1 Time-based prediction . . . 93
5.2.2 Distance-based prediction . . . 94
5.2.3 Signal-to-Noise ratio (SNR) prediction . . . 95

5.2.4 State-based prediction . . . 96
5.2.5 Distance predictor with global knowledge (Global) . . . 96
5.3 Analysis of Basic Predictors . . . 97
5.3.1 Evaluating real and false link breaks in AODV and DSR . . . 97
5.3.2 Performance of pre-transmission prediction . . . 98
5.3.3 Performance of post-transmission prediction . . . 105
5.3.4 Cost of pre- and post-transmission predictors . . . 109
5.3.5 Fairness in pre- and post-transmission prediction . . . 110
5.4 Adaptive Prediction . . . 110
5.5 Conclusion . . . 112

6 Experimental Evaluation of Channel State Dependent Scheduling 114
6.1 Wireless Testbed . . . 115
6.2 Signal Level Characteristics . . . 117
6.3 Channel State Dependent Scheduler with Channel Sensing . . . 118
6.3.1 Channel Sensing . . . 120
6.4 Wireless Channel Emulation . . . 121
6.5 Experimental Evaluation . . . 123
6.6 Conclusions . . . 126

7 Conclusions 127
7.1 Impact of Noise Sensitivity on Overall Network Performance . . . 128
7.2 Performance Under Traffic Overload . . . 129
7.3 Interface Queue Scheduling . . . 133
7.3.1 Performance evaluation . . . 135
7.4 Future Work . . . 135

A Collision Distance Calculation with Background Noise 137

Bibliography . . . 140

Vita 145

List of Tables

2.1 Examples of protocols in various network layers. 20

3.1 Default parameters for the radio model used in Glomosim simulator. 41
3.2 Simulation Topologies. 45
3.3 Achieved CBR throughputs for the 4-node network with $d = 689$ and 688 meters. 45
3.4 Achieved CBR throughputs for the 4-node network with $d = 540$ and 539 meters. 49
3.5 Achieved CBR throughputs for the 4-node network with $d = 539$ for unequal distances for communication node pairs. 51
3.6 Example 2:Throughput of the four-node network in Figure 3.4 with the proposed modification, d=688 m, node distance is 300 m. 55
3.7 TCP throughputs for the 4-node network with $d = 688$ m. 57

5.1 Summary of pre-transmission prediction actions, benefits, and costs. 99
5.2 Post-transmission prediction model summary. 105

6.1 Various statistics related to error rates and signal levels in the five experiments. 117

List of Figures

1.1 Satellite wireless network. . . . 2
1.2 Fixed wireless network. . . . 3
1.3 Cellular network. . . . 4
1.4 Mesh network. . . . 6
1.5 MANET network. . . . 7
1.6 CBR performance of a 100-node ad hoc network with AODV routing protocol and 802.11 MAC protocol. . . . 11
1.7 TCP performance of a 100-node ad hoc network with AODV routing protocol and 802.11 MAC protocol. . . . 11
1.8 TCP and CBR performance of a 100-node ad hoc network with AODV routing protocol and 802.11 MAC protocol. . . . 12

2.1 Four layer protocols stack in mobile ad hoc network. . . . 19
2.2 DSR performance with options recommended. . . . 26
2.3 Hidden terminal. . . . 28
2.4 Exposed node. . . . 28
2.5 802.11 MAC protocol DATA transfer with RTS, CTS and ACK control packets. . . . 31
2.6 AODV performance after fixing ERROR. . . . 36

3.1 Node interference setup. . . . 40
3.2 Signal strength as a function of distance from the transmitting node. . . . 41
3.3 Different regions in the propagation of a transmission with 15 dBm transmission power and free-space signal propagation up to 226m andtwo-ray signal model for larger distances. . . . 44
3.4 A 4-node network to illustrate the impact of the competing transmissions. . . . 44
3.5 Example of successful transmission between N_1 and N_2 in topology T_1, when d = 688 m. . . . 46
3.6 Example of synchronized transmission in topology T_3 ($d = 688$ m). . . . 48
3.7 Different regions in the propagation of a transmission. . . . 48
3.8 Collision distance experiment. . . . 52
3.9 Collision Distance: In Figure 3.8, collision distance D (Y-axis) for a specified communication distance d (X-axis). . . . 54
3.10 Proposed modification to 802.11 M AC protocol logic. . . . 56
3.11 16 node linear chain simulation setup. Nodes are placed 300 m apart. . . . 57
3.12 Cumulative CBR throughput improvement for static linear chain. . . . 58
3.13 Per connection throughput in a 16-node linear chain. . . . 58
3.14 Grid topology. . . . 59
3.15 Throughput improvement in grid topology. . . . 59
3.16 Throughput achieved for CBR traffic in a MANET with the 802.11 and modified MAC protocols using AODV routing protocol. . . . 60
3.17 CBR data end-to-end delay in a MANET with AODV. . . . 60

3.18 Average hop count for data delivered with 802.11 and modified MAC protocols. 61
3.19 Fairness evaluation of the 802.11 and modified MAC protocols. 61
3.20 RREQ control packets transmitted in the MANET (AODV). 61
3.21 Total DATA (unicast) packets lost due to collisions (AODV). 61
3.22 Total DATA collisions per successfully transmitted data frame. 62
3.23 Routing overhead per data packet delivered. 62
3.24 TCP throughput achieved in a MANET with the 802.11 and modified MAC protocols without CBR background traffic. 63
3.25 TCP throughput achieved in a MANET with the 802.11 and modified MAC protocols with 100 Kbps background traffic. 63
3.26 TCP throughput achieved in a MANET with the 802.11 and modified MAC protocols with 200 Kbps background traffic. 63
3.27 TCP throughput achieved in a MANET with the 802.11 and modified MAC protocols with 300 Kbps background traffic. 63
3.28 CBR throughput using DSR routing protocol. 64
3.29 CBR throughput using LAR routing protocol. 64
3.30 CBR throughput for MANET using $1000 \times 1000m^2$ and AODV. 65
3.31 CBR throughput for MANET using $1600 \times 1600m^2$ and AODV. 65
3.32 TCP throughput for MANET using $1600 \times 1600m^2$. 65
3.33 TCP throughput for MANET using $1000 \times 1000m^2$. 65
3.34 CBR throughput for MANET with high-mobility nodes. 66
3.35 CBR throughput for MANET with low-mobility nodes. 66

4.1 Performance of a 100-node ad hoc network with AODV and 802.11 protocols. 68
4.2 Control packets transmitted on wireless links. 71
4.3 Data, RTS and CTS packets transmitted on wireless links. 71
4.4 Control packet priority queue size for node 64 when offered load is 500 Kbps. 72
4.5 Control packet priority queue size for node 64 when offered load is 600 Kbps. 72
4.6 Data packet queue size for node 64 when offered load is 500 Kbps. 73
4.7 Data packet queue size for node 64 when offered load is 600 Kbps. 73
4.8 Performance of SBA, SBAe, RAD and RADe. 75
4.9 Number of broadcasts transmitted in SBA, SBAe, RAD and RADe. 75
4.10 CBR performance using RADe with modified 802.11 protocol. 77
4.11 TCP performance using RAD and RADe broadcast technique with modified MAC. 77
4.12 Reduced broadcast control packet priority queue size for node 64 when offered load is 500 Kbps. 78
4.13 Reduced broadcast control packet priority queue size for node 64 when offered load is 600 Kbps. 78
4.14 Improved throughput with reduced broadcast technique. 79
4.15 Control and data packets transmitted on wireless links when reduced broadcast technique is used. 79
4.16 CBR throughput improvement with modified 802.11 protocol and reduced broadcast. 79
4.17 TCP throughput with modified MAC protocol and reduced broadcast. 79
4.18 Performance of MANET after increasing the hop time by 3,10 and 20 times the value proposed single hop propagation time. 81
4.19 End-to-end delay of MANET after increasing the hop time by 3,10 and 20 times the value proposed single hop propagation time. 81
4.20 CBR performance of dynamic hop time and static hop times. 82

4.21 Broadcasts transmitted in dynamic hop time and static hop times. 82
4.22 End-to-end delay of dynamic hop time and static hop times. 83
4.23 CBR average end-to-end delay for flooding, dynamic hop time with modified MAC, RADe with modified MAC and reduced broadcast with modified MAC protocols. 83
4.24 CBR throughput of flooding, dynamic hop time with modified MAC, RADe with modified MAC, and reduced broadcast with modified MAC protocols. 83
4.25 Broadcast transmitted in flooding, dynamic hop time with modified MAC, RADe with modified MAC, and reduced broadcast with modified MAC protocols. 83
4.26 CBR: Total number of packets traveled 1-hop to reach the final destination. 84
4.27 CBR: Total number of packets traveled 2-hop to reach the final destination. 84
4.28 CBR: Total number of packets traveled 3-hop to reach the final destination. 85
4.29 Jain's fairness index for the proposed modification. 85
4.30 TCP performance of flooding, dynamic hop time with modified MAC, RADe with modified MAC, and reduced broadcast with modified MAC protocols. 85
4.31 CBR throughputs with DHT with RADe, DHT with RB, RADe with and DHT, RADE along with RB. 85

5.1 Baseline MANET's performance with AODV and DSR. 90
5.2 Baseline MANET's total control packets (RREQ, RREP, and RERR) with AODV and DSR. 90
5.3 Illustration of pre-transmission prediction. 91
5.4 Illustration of post-transmission prediction. 92
5.5 State diagram for the state predictor. 96
5.6 AODV real route breaks and false route breaks in MANET. 98
5.7 DSR real route breaks and false route breaks in MANET. 98
5.8 Pre-transmission predictor flow diagram. 100
5.9 AODV pre-transmission prediction schemes in MANET. 101
5.10 AODV total RREQs transmitted in pre-transmission prediction schemes. 101
5.11 AODV pre-transmission prediction: Number of out of range packets entering the IFQ per second. 102
5.12 AODV pre-transmission prediction: Number of out of range packets exiting the IFQ per second. 102
5.13 AODV pre-transmission prediction: Number of real route breaks per second. 102
5.14 AODV pre-transmission prediction: Incorrect out of range predictions per second. 102
5.15 DSR pre-transmission prediction schemes in MANET. 103
5.16 DSR pre-transmission prediction route request per second. 103
5.17 DSR pre-transmission prediction: Out of range packets exiting the IFQ per second. (M's out of range packets in Figure 5.8.) 104
5.18 DSR pre-transmission prediction: Real route breaks per second using prediction. 104
5.19 AODV pre-transmission prediction scheme: TCP performance. 104
5.20 AODV pre-transmission prediction: 200 Kbps CBR background noise achieved throughput. . 104
5.21 DSR pre-transmission prediction: TCP performance. 104
5.22 DSR pre-transmission prediction scheme: 200 Kbps CBR background traffic achieved throughput. 104
5.23 Post-transmission prediction flow diagram with cost and benefit. 106
5.24 AODV post-transmission prediction schemes in MANET. 107
5.25 AODV false route breaks in post-transmission predictions. (Branch H in Figure 5.23.) 107
5.26 AODV post-transmission prediction RREQs transmitted in MANET. 107
5.27 DSR post-transmission prediction schemes in MANET. 107

5.28 DSR post-transmission prediction false route breaks. . . . 108
5.29 DSR post-transmission prediction: RREQs transmitted in MANET. . . . 108
5.30 AODV post-transmission prediction scheme: TCP performance. . . . 108
5.31 AODV post-transmission prediction: 200 Kbps CBR background traffic achieved throughput. 108
5.32 DSR post-transmission prediction: TCP performance. . . . 109
5.33 DSR post-transmission prediction: 200 Kbps CBR background traffic achieved throughput. . 109
5.34 Pre-transmission prediction and CBR: Total number of packets traveled 1-hop to reach the final destination. . . . 110
5.35 Pre-transmission prediction and CBR: Total number of packets traveled 2-hop to reach the final destination. . . . 110
5.36 Pre-transmission prediction and CBR: Total number of packets traveled 3-hop to reach the final destination. . . . 110
5.37 Post-transmission prediction and CBR: Total number of packets traveled 1-hop to reach the final destination. . . . 111
5.38 Post-transmission prediction and CBR: Total number of packets traveled 2-hop to reach the final destination. . . . 111
5.39 Post-transmission prediction and CBR: Total number of packets traveled 3-hop to reach the final destination. . . . 111
5.40 Jain's fairness index for pre-transmission prediction using DSR. . . . 111
5.41 Jain's fairness index for post-transmission prediction using AODV. . . . 111
5.42 Adaptive time prediction using pre-transmission prediction and using DSR as MANET routing protocol. . . . 112

6.1 Probability mass function of signal levels in the five experiments. . . . 118
6.2 Cumulative distribution of various run lengths of dropped packets. . . . 118
6.3 Channel state dependent scheduling system showing the different queues in the device driver. 119
6.4 State transition diagram of a neighbor node. . . . 120
6.5 Finite-state Markov chain model for channel emulation. . . . 122
6.6 Experiment 1 node setup. . . . 123
6.7 Experiment 2 node setup. . . . 123
6.8 Total received bandwidth vs. offered load on the weak link alone in Experiment 1. . . . 124
6.9 Total received bandwidth vs. offered load in Experiment 1. . . . 124
6.10 Bandwidth consumed by the dropped packets vs. offered load on weak link alone in Experiment 1. . . . 124
6.11 Breakdown of the offered load on the weak link in Experiment 1. . . . 124
6.12 Total received bandwidth vs. offered load on weak link in Experiment 2. . . . 125
6.13 FTP transfer time vs. UDP load on the strong link in Experiment 3. . . . 126

A.1 Node setup for collision distance calculation. . . . 137

Chapter 1

Introduction

Recent advances in wireless communication technology and portable computing devices such as notebooks and PDAs have generated a lot of interest in developing and improving protocols for networks for such devices. It is widely recognized that wireless networks have many characteristics that are not shared by their wired counterparts. The key differences include the limited bandwidth, the shared nature of the wireless medium, the highly time-varying and unreliable nature of wireless channels, and the host mobility. Also, the service needs and expectations for a mobile device are frequently different from those of a powerful desktop computer. Due to these differences, the protocols developed for wired networks often do not perform effectively on wireless networks without certain wireless-specific enhancements. In addition, to handle mobile hosts and specific mobile services, new protocols must be devised.

Since the late eighteenth century, wireless technology has been experimented; in the early stages, radio waves were used over long distances as a means of communication. During the nineteenth century, the use of wireless radio communication grew rapidly. A major milestone in wireless computer communication was achieved in the University of Hawaii's ALOHANET research project [2]. Since the first use of "Packet Radio" in ALOHANET in the 1970's, wireless data communication has gained popularity.

Wireless networks can be classified into three primary categories, based on the structure of the network. The simplest structure is the point-to-point structure. In this structure, a wireless device communicates directly with another wireless device. Typically, the nodes in a point-to-point structure are stationary. The popular cellular networks are another type of wireless network. In a cellular network, a geographical region is served and managed by a base station that provides 1-hop wireless connectivity to users' cellular phones

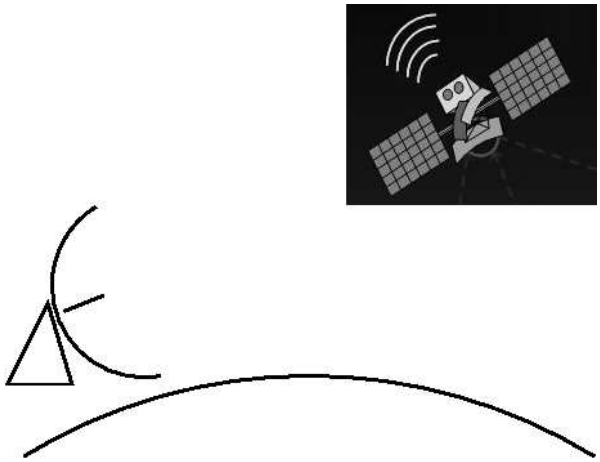

Figure 1.1: Satellite wireless network.

and other devices. The third type of data network is an ad hoc network. An ad hoc network is simply an infrastructure-less network, which is a more complex and relatively new idea in the network community. In the following sections, we will discuss the key characteristics of each infrastructure and our motivation and research interest in these infrastructures.

1.1 Point-to-Point Data Communication

In a typical point-to-point data communication network, the sender and receiver are strategically located to communicate with each other. Several of these techniques are described below.

1.1.1 Satellites data communication

Since April of 1960, when the first satellite, Sputnik 1, was launched, the satellite technology has greatly advanced. Today, satellites are primarily used for telephony, video and data communication. There are several applications of data communication using satellites. A few of them are private data networks, internet service providers to backbone, end-user data networks (small office, home office, residential), and in-flight entertainment. A typical satellite-based wireless network is shown in Figure 1.1. Satellite data communication is still struggling to enter the end-user data networks due to stiff competition from landline data network providers, such as cable and DSL companies. These companies are able to deliver high bandwidth data points at a low cost. Cable data network providers offer 10 Mbps shared service, and DSL providers

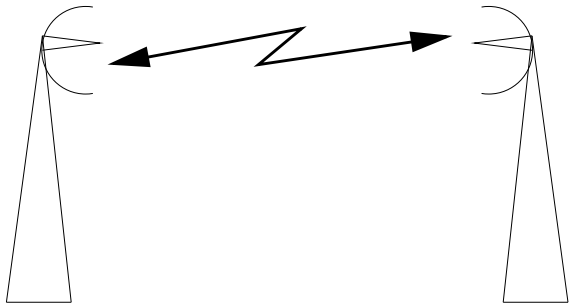

Figure 1.2: Fixed wireless network.

commonly offer a speed of up to 1.5 Mbps. On the other hand, satellites (high orbiting) can deliver speeds of up to 400 Kbps upstream and 1.5 Mbps downstream. An advantage to satellite data communication is its ability to reach remote areas for which landline communication is not feasible or is unavailable.

1.1.2 Point-to-point fixed wireless

A well-known point-to-point fixed wireless service is the Local Multipoint Distribution Service (LMDS). This service is a relatively new service in the U.S., but it is very popular in countries where there is no infrastructure to deliver high speed data. Typically, these services can offer 384 Kbps to 1.5 Mbps downstream speeds and 128 Kbps upstream speeds. This service can be delivered to an end-user or a service provider. A typical fixed, point-to-point wireless network is shown in Figure 1.2.

1.2 Cellular Networks

One key requirement for a cellular network is a fixed infrastructure (see Figure 1.3). The infrastructure consists of base stations, or access points. These base stations are connected to each other and to the internet or other larger networks. Each mobile node is connected to a base station via a wireless link. The base station can communicate to all mobile nodes in its radio range (called a cell). Nodes are free to move from one cell to another cell. When a node moves to a new cell, it leaves its current base station and joins the base station in the new cell. The most common use of the cellular network is voice service.

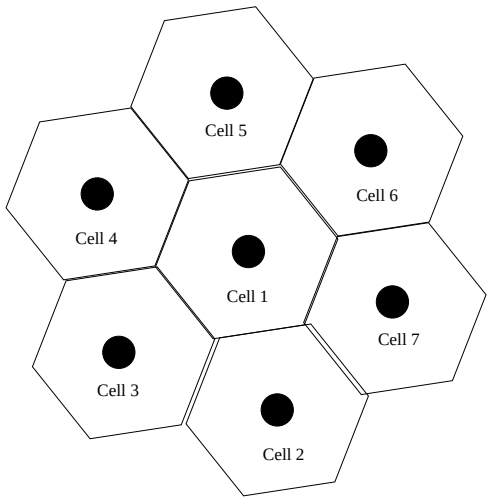

Figure 1.3: Cellular network.

Cellular wireless data

Cellular wireless data is an overlay to the existing cellular telephony infrastructure. A cellular wireless network uses various types of transport technology. The three dominant technologies are TDMA (Time Division Multiple Access), GSM (Global System for Mobile), and CDMA (Code Division Multiple Access). TDMA divides the frequency band into several channels, or time slots, which are then stacked into shorter time units. This allows the TDMA to carry several calls on a single channel. CDMA, on the other hand, encodes the wireless data using random codes. This allows CDMA a greater reuse of the frequency band. GSM is used in more than 200 countries around the world and is dominant in Europe and Asia. GSM uses a combination of Frequency Division Multiplexing (FDMA) and TDMA to transmit a wireless signal.

At present, cellular wireless data technology is being upgraded from 2^{nd} Generation (2G) to 3^{rd} Generation (3G). The bandwidth is the most significant constraint for cellular wireless data. GSM technology made significant progress in increasing data speeds from 9.6 Kbps in 2G to 384 Kbps in 3G. CDMA technology provides even greater bandwidth improvements from 2G to 3G. The bandwidth in 3G is improved to 2.4 Mbps from 144 Kbps in 2G. CDMA designers are expecting 10 Mbps within the next few years using the 3G CDMA technology.

Based on customer needs, CDMA 3G uses different technologies to deliver data. The wideband CDMA (W-CDMA) is a multi-cooperation effort that delivers data at rates of up to 2 Mbps and promises to deliver

data at rates of up to 10 Mbps by the year 2005. The CDMA2000 1xRTT is the first phase of the CDMA2000 2G technology and can deliver voice and data rates of up to 144 Kbps. The CDMA2000 1xEV-DO delivers data on a separate channel at speeds of up to 2.4 Mbps. The CMDA2000 1xEV-DV integrates voice and data on the same channel and delivers data at rate of up to 2.4 Mbps.

A drawback to such cellular wireless networks is that the provider must bear the cost of building the infrastructure, and the cost of acquiring the signal spectrum.

1.2.1 Wireless local area networks (WLAN)

A wireless local area network is somewhat like a single cell in a cellular network. To reduce the cost of implementation, free signal bands and standards-based hardware are used. The IEEE standard 802.11 [20], which specifies the physical and the medium access (MAC) layer protocols, is one such standard for wireless LANs. Currently, there are several supplements to the 802.11 MAC standard. The MAC 802.11b [21], one of the three available and the most widely-used technology, denoted by Wi-Fi, delivers up to 11 Mbps throughput using a 2.4 GHz band. The 802.11g [23] is backward compatible with the 802.11b standard and delivers up to 54 Mbps, using a 2.4GHz band. On the other hand, the 802.11a [22] standard uses a 5GHz band and delivers up to 54 Mbps of throughput. The 802.11 based wireless networks are primarily designed for home, office, or campus use. In the U.S., it uses the license-free ISM (industrial-scientific-military) bands. Commercial Wi-Fi services are available in places such as Internet cafes, coffee houses, and airports around the world.

Typically, the 802.11 networks operate using static access points, which are connected via a wired LAN. In most cases, users in a single cell will remain in that cell and are not expected to move from one cell to another cell. Even though roaming between cells is feasible, in most cases roaming is not implemented. The disadvantage in the 802.11b and the 802.11g standards is the use of a 2.4 GHz spectrum, which is crowded with other devices, such as bluetooth, microwave ovens, cordless phones, video sender devices, among others. Therefore, interference from the other devices that operate in the same frequency band results in performance degradation.

Mesh networking is the new trend in broadband wireless service. Mesh networking uses inexpensive

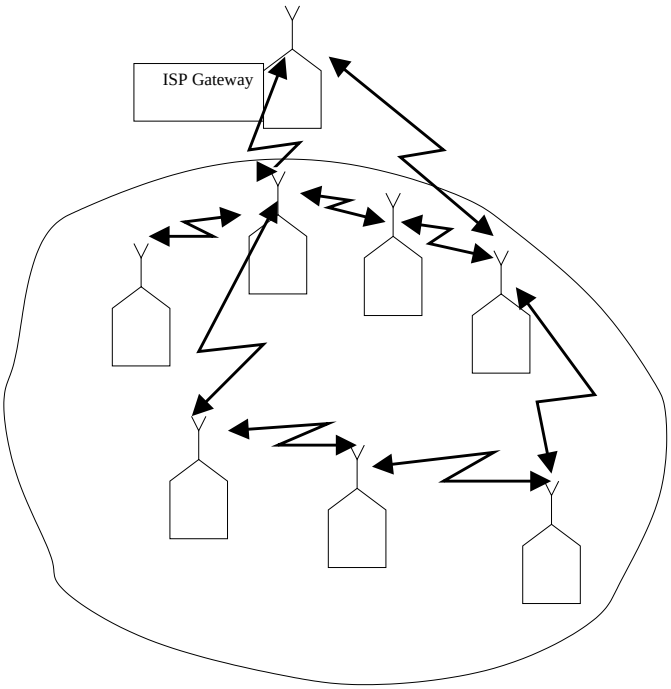

Figure 1.4: Mesh network.

802.11 MAC Wi-Fi devices. These networks can be spread over cities or small neighborhood communities. In neighborhood communities, the wireless router antenna are placed on rooftops, and these routers work as routers to the broadband wireless network and as an access point to mobile devices (see Figure 1.4). Broadband wireless networks, such as mesh networks, are gaining popularity because of the low cost of implementation and the lack of regulatory restrictions on the frequency band used.

1.3 Ad Hoc Networks

In contrast to cellular networks, ad hoc networks are infrastructure-less networks (no fixed base stations; see Figure 1.5). Each node in an ad hoc network acts as a potential router, routing packets for a pair of communicating nodes that may not be within each others' radio range. Thus, unlike in cellular networks and WLANs, communication in ad hoc networks may take place via multiple wireless hops. For example, a wireless communication from node A to node E (not in each others' communication range) in Figure 1.5 would take place via nodes B and D. There are many technical challenges that must be addressed to make such networks usable in practice. Primarily, as the nodes can be mobile and, thus, the topology may be

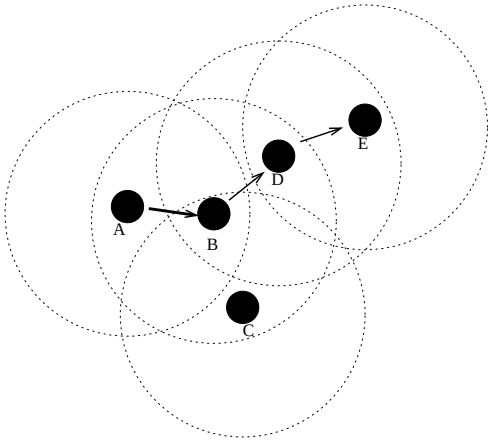

Figure 1.5: MANET network.

changing frequently, a dynamic routing protocol must be employed to maintain routes between a pair of source-destination nodes. The protocol must be efficient in terms of routing overhead because of bandwidth constraints in a wireless link; it must be adaptive to various sizes and forms of Mobile Ad Hoc Networks (MANET), and it must meet power constraints of the mobile devices.

In the recent past, there has been a strong interest on the part of the network research community to address various issues related to ad hoc networking. Due to the growing interest, the Internet Engineering Task Force (IETF) formed a new working group for mobile ad hoc networking [47] whose focus is to develop and standardize IP-based routing protocols for ad hoc networks. The MANET routing protocols may be classified as on-demand or reactive ("as needed") [58, 59, 38], and proactive [54, 17]. These are described below.

1.3.1 MANET routing

On-demand protocols use a source-initiated route discovery process to discover routes when a route is needed. When adjacent nodes in a route move away from one another, the routes break. Undetected broken routes are denoted as stale routes. The stale part of the route is erased, and a new route discovery is initiated. Depending on the protocol and the scenario, the broken route may be repaired locally as well. On-demand protocols are attractive when compared to more traditional, proactive, shortest-path based protocols (e.g., distributed Bellman-Ford [15] or distributed link-state protocols [40]), as they usually have a lower routing

overhead for common traffic scenarios [19].

On the other hand, proactive protocols maintain the routes for all possible destinations, even if many of these routes are not required. Each node maintains its own view of the network and the link cost of all of its outgoing links. To maintain an up-to-date view, each node broadcasts the link cost of all of its neighbors. These link costs are transmitted to neighboring nodes when there is a change detected in the link cost. When the node receives this information, it updates its view of the network topology and then uses the shortest-path algorithm to choose the next hop node address to a destination. Since the node keeps next hop information to all destinations, the proactive protocol demonstrates a lower packet latency than the reactive protocol does. The drawback to the proactive protocol is the routing overhead, especially when compared to reactive protocols.

Currently, there are several proactive and reactive protocols under consideration for standardization by the MANET working group in the IETF [47]. The IETF's primary objective is to standardize the routing protocol's functionality for wireless networks.

1.3.2 MAC protocol

Several MAC protocols have been proposed for wireless communication. The IEEE 802.11 [20] and the Multiple Access Collision Avoidance (MACA) [39] are the best-known MAC protocols and have been studied extensively by researchers. The Power Aware Multi-Access protocol with Signaling for ad hoc networks (PAMAS) [65] and the Floor Acquisition Multiple Access (FAMA) [30] are two additional protocols that have been proposed and studied. Among others, the IEEE 802.11 MAC protocols has gained the interest of the research community and the industry for high-speed wireless communication.

802.11 MAC protocol

The 802.11 MAC protocol is designed in a such a way that it can be used in cellular and ad hoc networks. Due to the inherent problem of collision detection in wireless networks, the 802.11 MAC protocol uses a CSMA/CA (carrier-sense multiple access with collision avoidance). This protocol persists on a busy channel and employs a random backoff after the channel switches to idle in order to avoid collision when other nodes

are waiting to transmit. The 802.11 standard optionally uses RTS/CTS (request-to-send and clear-to-send) packets to reserve the channel between a pair of source and destination nodes in order to avoid the classic hidden-terminal problem [68].

In an ongoing communication, a hidden node is a node that is within the range of the destination node but outside of the range of the source node. In such a scenario, a hidden node has the illusion of a free channel and transmits data, which results in a collision with the ongoing transmission. In contrast, an exposed node is a node that is within the communication range of the source, but it is outside of the range of the destination. In this scenario, the exposed node senses a busy medium, even though the node can successfully transmit to another node without colliding with the ongoing transmission.

To overcome the unreliability of the wireless media, the 802.11 protocol's destination uses a link layer ACKnowledgment (ACK), a short control packet sent by the receiving node in response to successful reception of the DATA packet. If the sender of a DATA packet does not receive an ACK, then it retransmits the DATA until an ACK is received or until the maximum retry limit is reached.

Transport protocols

Internet transport protocols are typically designed, tested, and fined-tuned for wired network topologies. Wireless networks have many characteristics that are not shared by their wired counterparts. Recently many researchers have looked at transport layer performance on single hop wireless networks [7, 12, 4] and found that the traditional transport protocols do not perform well in the wireless networks. In an ad hoc network, the transport level problems take an additional level of complexity, as the data must now travel through multiple wireless hops and use a dynamic routing protocol to find the route from the source to destinations. Recent studies of ad hoc networks using the 802.11 MAC protocols have shown that the traditional transport layer protocols designed for wired networks do not perform well in the wireless networks [77, 50, 46, 35], and that the interactions between the MAC layer and the upper layer may be causing unfairness among the communications.

1.4 Challenges

There are many challenges in the design of a MANET. The absence of a centralized authority forces network management to take place in a distributed fashion, where each node is designed to act in the best interest of the MANET. Therefore, nodes must collaborate among themselves to form an efficient MANET. Often, a communication in a MANET requires multiple hops and must find a *good route* and repair this route when the network topology changes. Route discovery and route repair must be efficient, fast, and scalable.

Wireless links continue to have a significantly lower capacity than wired links. Bandwidth constraints become problematic when routing protocols use part of the available bandwidth for control packets to discover and maintain routes. Multiple access and possible contention of channels, fading, noise, and interference causes a further reduction in the available capacity for data transmission. Even though the wireless link capacity has improved significantly since the standardization of the 802.11 MAC protocol, user bandwidth demands have been rising at an even higher rate. Therefore, the MANET is frequently expected to operate at or above the network capacity. Hence, MANET protocols must be designed to operate efficiently in both high and low network loads.

Wireless networks are more prone to physical security threats than the regular wired networks. The ad hoc nature of nodes joining and leaving a MANET increases the possibility of eavesdropping, spoofing, impersonating, and corrupting data while data is been sent from a source to its destination. Even though it is difficult to design protocols to eliminate such problems, they must be designed to reduce the threat to security [80, 51, 10].

Some, if not most, devices in a MANET are mobile and battery-powered. As the devices are becoming smaller, available energy is reduced due to reduced battery sizes. Therefore, energy-efficient and power-aware routing is another challenging problem [66].

1.5 Motivation

Bandwidth constraints play a major role in a MANET's usability. MANET users who are accustomed to wired network responses and services will demand a higher capacity than what is available. The MANET

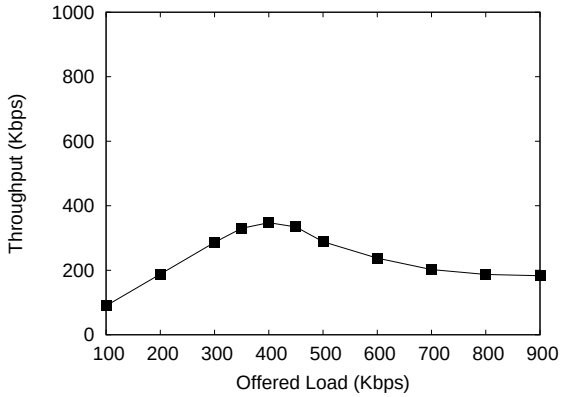

Figure 1.6: CBR performance of a 100-node ad hoc network with AODV routing protocol and 802.11 MAC protocol. There are 50 sources sending data to 50 destinations at a constant bit rate.

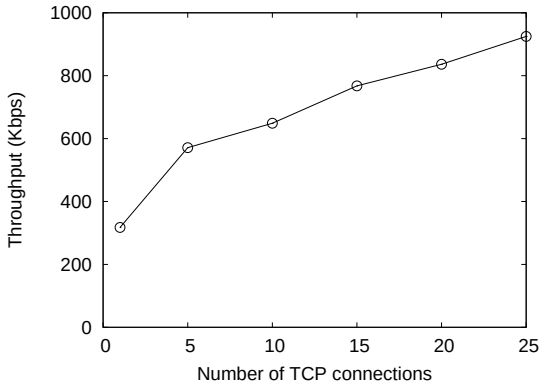

Figure 1.7: TCP performance of a 100-node ad hoc network with AODV routing protocol and 802.11 MAC protocol. There are 1-25 FTP connections using TCP.

must be designed to handle such a situation gracefully. Due to shared wireless channels, as the number of users increase, the bandwidth available per user decreases. To illustrate this issue, we simulate a 100-node MANET using Glomosim simulator. One hundred nodes in a 1200×1200 m^2 terrain were simulated. The node movement is patterned by the random waypoint model with speeds in the range of [1,19] m/sec. Fifty CBR connections with a packet size of 512 bytes were used to simulate varying network loads. Each simulation was run for 600 seconds (the first 100 seconds are used to warm-up the network, and no statistics were collected during this time). The simulations were repeated 9 times with different random seeds, and these were averaged to minimize the impact of worst-case and best-case scenarios. (Unless otherwise specified, this is the baseline simulation setup we use for performance evaluation.)

Figure 1.6 shows throughput using AODV routing protocol. One the average, a maximum throughput of 350 Kbps for CBR traffic is achieved. The maximum throughput may vary, depending on the number of nodes and the total number of connections established within the ad hoc network. Therefore, a small-scale ad hoc network could achieve only a maximum of 20% of the single wireless channel's bandwidth. If there are 50 senders, then each sender achieves an average of 0.4% of the channel's BW.

When the offered load is increased beyond a certain value (400 Kbps for our example network), the network breaks down. For example, when the example MANET is offered a 900 Kbps load, the network achieves only 180 Kbps. The network does not even sustain the peak throughput it achieved at lower load

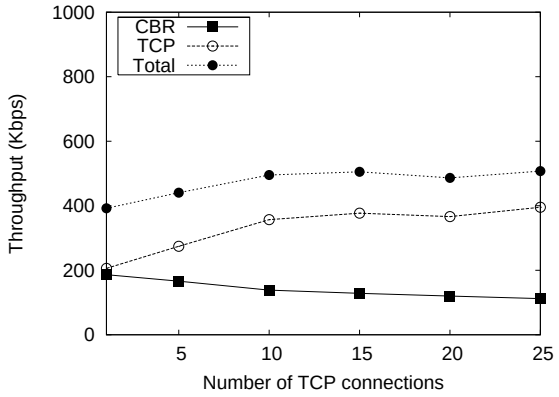

Figure 1.8: TCP and CBR performance of a 100-node ad hoc network with AODV routing protocol and 802.11 MAC protocol. There are 50 sources sending data to 50 destinations using CBR traffic at 200 Kbps and 1-25 FTP connections using TCP.

levels. Clearly, this makes the network unreliable and unusable in high traffic load scenarios.

In Figure 1.7, we show a MANET using FTP communications. In this simulation, 1-25 FTP connections (using TCP) were simulated. In contrast to CBR communications, TCP communications show a significantly high throughput, and no significant problem of performance degradation, when the number of TCP connections are increased. In fact, when there are 25 FTP communications, each communication receives an average of 38 Kbps. This is significantly higher than the average received by MANET using 50 CBR communications. The reason for the higher bandwidth is TCP's congestion control mechanism. Using congestion control, the MANET is prevented from reaching the saturation point. Dyer and Boppana [27] show that TCP communications in a MANET take advantage of shorter routes when they are available. As a result, a MANET can achieve higher throughput using TCP communications.

For a more realistic scenario, we simulated TCP and CBR communications in the example MANET. For this simulation, we used 1-25 FTP connections along with 200 Kbps CBR offered load. 50 sources and 50 destinations are used to offer CBR load. Figure 1.8 shows the performance of TCP, CBR and the total achieved throughput. There is a significant loss (up to 55%) in TCP throughput when 200 Kbps CBR load is added on the MANET. On the other hand, CBR communications lost 50% of its achieved throughput when 25 TCP connections were added on the MANET.

The use of congestion control benefits the MANET and maximizes its throughput. The absence of the same saturates the MANET and causes it to perform poorly beyond the saturation point. Mixing TCP traffic,

the rate of which changes with congestion and UDP traffic, which is oblivious to network congestion, results in a significant loss in throughput. With the recent popularity of voice over IP and video transmissions, UDP type traffic, as a fraction of total traffic, is likely to increase. Therefore, MANETs are likely to deliver low performance, which becomes worse as the load increases.

The on-demand routing protocols for MANETs typically rely on transmission failures to detect route failures. In a congested MANET, it is likely to have transmission failures due to congestion. These transmission failures may be incorrectly interpreted as route breaks. These unnecessary route breaks increase the routing overhead, which reduces the network bandwidth available for data transmissions. In addition, increase in the overhead increases the amount of congestion-oblivious traffic on the MANET. The problem is exacerbated when the control overhead traffic is given higher priority over data.

There has been a significant amount of research conducted to study the performance of MANETs prior to saturation, but there has been very little work that focuses on the performance of MANETs at and beyond the point of saturation. Therefore, analyzing MANETs under conditions of saturation and identifying their weaknesses are necessary and crucial to their widespread use. There are several areas that can be explored for potential problems when a MANET is saturated. Weaknesses in the MAC and routing protocols can cause performance losses in MANETs that have reached the point of saturation. In addition, the interaction between the transport layer and the MAC and/or routing protocols can cause performance losses. For example, frequent route breaks can lead to poor TCP performance (due to its congestion avoidance algorithm).

1.5.1 MAC protocol

When the network is saturated, interface queues (IFQs) from the network layer to the MAC layer frequently overflow. While some nodes are hidden or exposed, other nodes may be in communication range. Traditionally, the IFQs use a First-Come-First-Served (FCFS) order and process data as they arrive, irrespective of the destination status. In cellular-based stations, non-traditional queues (for example, channel state dependent queuing, round robin, most frequently used) have been shown to benefit cellular network's throughput [9]. Similar queuing techniques can be used in congested nodes in the MANET to reduce the number of attempts to reach nodes that are temporarily unreachable.

Several researchers have investigated the effectiveness of the 802.11 MAC protocol and its impact on the overall performance of the MANET [74, 73, 76, 36]. Simulation studies of ad hoc networks have shown evidence of unfairness (unfair distribution of channel access) at the MAC layer, which causes short and long-term unfairness (bandwidth distribution) in the application layer [62, 77, 36]. In addition, radio interference can cause unfairness in MANETs. Only a few researchers have studied the effects of interference on the MANET. Unfairness, typically, does not lead to a poor overall performance of the wireless network, but unfairness can lead to frequent route breaks in MANETs. Frequent route breaks can increase the route maintenance overhead and cause performance degradation.

1.5.2 Routing protocol

A routing protocol uses control packets to discover and maintain routes. These control packets do not carry data, so they are considered to be an overhead caused by the routing protocol. The routing overhead caused by most routing protocols uses a significant part of the MANET's capacity. For on-demand protocols that rely on network-wide floods to solicit routes (using route request, or RREQ, control packets), the control overhead is a significant problem, especially when the network is congested. By reducing the overhead, the overall performance of a MANET may be improved. For on-demand protocols, such as the AODV, RREQ floods in the network contribute a major part of the routing overhead. Reducing the number of RREQs can increase the throughput in a MANET. Similarly, inefficiencies in detecting route breaks and repairing them can cause an unnecessary increase in the routing overhead.

1.5.3 Broadcast management

There have been few studies that address ways to reduce the routing overhead by reducing the number of RREQ broadcasts. Most techniques in the literature attempt to reduce the routing overhead in order to reduce congestion and facilitate higher throughput prior to saturation. However, such techniques do not perform well when the network is operating beyond saturation.

An alternative approach to reduce the routing overhead is to reduce the need for route discovery. When the network is at or beyond the point of saturation, the 802.11 MAC protocol causes frequent false route

breaks. While the next hop is within the communication range, route breaks caused by transmission failures are called false route breaks. These route breaks are not due to topology changes; they are congestion-related route breaks. Avoiding such route breaks can help improve the MANET's overall performance.

1.5.4 Route maintenance

In contrast to false route breaks, real route breaks are link failures that occur when the next hop node moves out of the communication range. The number of real route breaks will increase as the network mobility increases. Also, an excessive number of real route breaks can be caused by the use of stale routes. Additionally, low-adaptive algorithms may not detect real route breaks quickly. When stale routes are used to send data, some of the network BW will be used for data transmissions that are not productive.

1.6 Contributions of This Dissertation

This dissertation contributes to the following areas of MANET design:

- Propose a modification of the 802.11 MAC protocol to improve the performance of MANETs.
- Identify the weakness of an on-demand routing protocol under traffic overload and propose solutions to mitigate the problem.
- Design and use of a prediction model to predict the link status of the next hop for the MANET.
- Design of an IP queuing mechanism for ad hoc networks to utilize the available bandwidth efficiently.

1.6.1 Improving 802.11 MAC protocol design

The IEEE 802.11 MAC layer protocol plays a crucial role in the overall throughput obtained in a mobile ad hoc network. We show that the virtual carrier sense mechanism, as designed and used in the 802.11 MAC protocol, can have a crippling effect on distant but competing transmissions. We propose a modification that changes how the physical sense is detected when a node is transmitting a CTS in response to a received RTS. Using simulations, we show that the proposed modification provides up to 80% higher UDP throughput for

static wireless networks, and from 11-25% higher throughput for mobile ad hoc networks. The proposed modification alleviates the exposed node problem for MANETs operating at or near the point of saturation.

1.6.2 Broadcast management in MANET routing protocols

We investigated the performance of wireless ad hoc networks with traffic loads beyond the point of saturation. While it is desirable to operate a network below saturation, an ad hoc network should be designed to gracefully degrade its performance under severe loads. Using the AODV (ad hoc on-demand distance vector) and the 802.11 as example routing and MAC (Medium Access Control) level protocols, we show that the throughput of an ad hoc network decreases rapidly after the point of saturation is reached. The reasons for this behavior are high route maintenance overhead and increased radio interference. We propose modifications to the routing protocols to mitigate these negative factors and provide graceful degradation of performance under heavy loads.

The reason for the high routing overhead is the inability of the routing protocol to adapt to the network conditions. In high network loads, the length of time taken to propagate RREQs to the intended destinations increases. As a result, the route repair time is increased as the network load increases. The AODV routing protocol uses a constant single hop time to calculate the total route repair time. When a node issues a RREQ and does not receive a reply within this time, it will retransmit a RREQ. This fixed time period is designed for low loads and, in congested networks, causes the routing protocol to retransmit RREQs unnecessarily.

To mitigate this problem, we propose two solutions. The first solution requires each node to analyze its interface queue and remove all duplicate RREQs generated due to premature timeout. The routing protocol is not modified. The second solution requires a modification of the routing protocol, but it is more effective in reducing the control overhead. This modification estimates average queue delay for a hop and uses this information to compute the route repair time. Using simulation, we show that this approach is very effective in stabilizing control overhead and mitigating any throughput loss.

1.6.3 Prediction schemes for route management

The routing overhead plays a critical role in the performance of a MANET. The efficient and quick discovery of a route is an important characteristic of the route discovery process in a routing protocol, while quickly and correctly identifying a broken route is an important characteristic of its route maintenance process. In this study, we show that on-demand routing protocols fail to identify route breaks correctly in high traffic conditions. Our simulation studies show that, under a high load, a large portion of route breaks discovered by some routing protocols are false; in some protocols, inefficiencies cause a large number of route breaks to go undetected for a long period of time.

We propose route status prediction schemes to mitigate this problem. These prediction schemes are similar to the branch prediction schemes used in processor pipeline design [55]. The prediction can be made at the MAC level, prior to transmission or after the transmission attempt has failed. The former is called pre-transmission prediction, and the latter post-transmission prediction. Accurate pre-transmission prediction is useful in mitigating the stale route problem. Similarly, accurate post-transmission prediction is useful in mitigating the false route break problem. In this work, we study the effectiveness of various pre- and post-transmission predictions for MANETs with high traffic loads.

1.6.4 MAC layer scheduling

Wireless links are often subjected to burst errors, leading to consecutive packet losses which could be buffered and transmitted later and giving priority to packets on stronger links. This improves the overall bandwidth utilization. A channel state dependent scheduling protocol based on the above idea is developed and implemented in a wireless LAN; a commercial wireless LAN, Lucent Technology's Wavelan, is used with Pentium-based laptops and PCs. The protocol is implemented as part of the device driver in the Linux operating system. The protocol includes a channel sensing mechanism to help determine when a wireless link emerges from the error state. An experimental performance evaluation, with UDP streams and FTP sessions, demonstrates the significant performance benefits of the channel state dependent scheduling. The performance evaluation includes a novel channel modulation technique based on the previously collected traces for the reproducibility of experiments.

1.7 Organization of the Dissertation

The rest of the dissertation is organized as follows. Chapter 2 presents the background material on MANETs and the protocols we used in this work. Chapter 3 identifies a weakness in the IEEE 802.11 MAC protocol and shows how the performance can be improved with a simple modification to it. Chapter 4 identifies the reasons for RREQ flooding in on-demand routing protocols and proposes several techniques to reduce RREQ flooding and to improve the MANET's performance. Chapter 5 presents route status prediction schemes and their effectiveness. Chapter 6 presents a fair scheduling algorithm for WaveLan to reduce the effect of the head-of-line problem. Chapter 7 concludes the dissertation and provides direction for future work.

Chapter 2

Background

In this chapter, we present the background material related to ad hoc networks and to the research presented in this dissertation. First, we briefly describe an ad hoc network, using the 4-layer (Internet model) network protocol stack [61].

The network protocol stack illustrated in Figure 2.1 shows the various layers data passes through when it is transmitted from one node to another node. Table 2.1 shows the commonly used protocols in each layer. The layer approach enables, optimizes or enhances the functionality of protocols in one layer without affecting compatibility with the protocols in the adjacent layers.

The application layer defines the basic user-level protocol. For example, a web browser uses the HTTP (Hyper Text Transfer Protocol) as the application-layer protocol to communicate with the server. For reliable communication with the server, the HTTP protocol chooses the TCP (Transmission Control Protocol) from the transport layer.

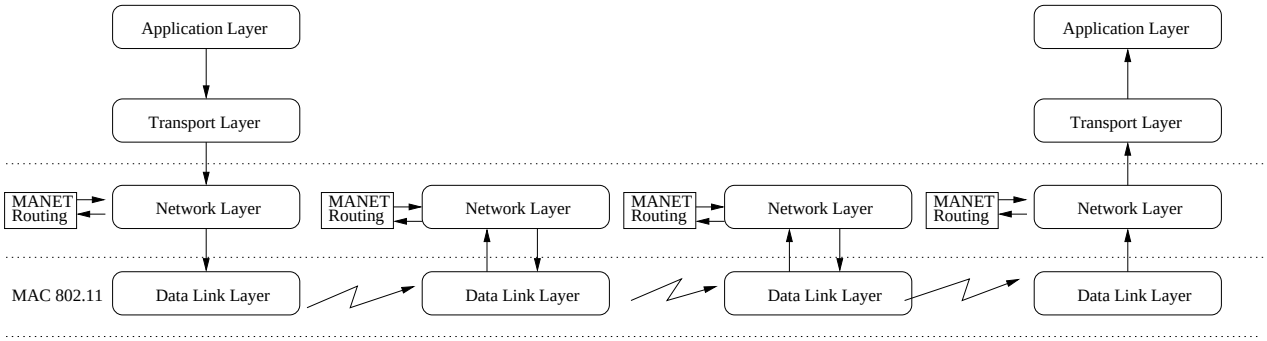

Figure 2.1: Four layer protocols stack in mobile ad hoc network.

Table 2.1: Examples of protocols in various network layers.

Layer Name	Examples
Application	FTP, SMTP, SSH, IRC, IMAP, POP
Transport	TCP, UDP, STPC
Network	IP, IPv6, RIP, OSPF, AODV, DSR, . . .
Data Link	Ethernet, 802.11, FDDI, PPP

Transport protocols deliver packets from application-to-application. Different transport protocols provide different types of services to the application layer. The most primitive protocol is the User Datagram Protocol (UDP). UDP offers a best-effort service from source to destination. Best-effort service attempts to deliver data with no guarantee of delivery or delay. In contrast, the Transmission Control Protocol (TCP) establishes a reliable and bidirectional virtual link between the source and destination nodes. The communication between these nodes is bidirectional. TCP uses the sliding window protocol to deliver data reliably, in order, and without duplication. It also has a congestion control mechanism to avoid congestion.

TCP, in turn, uses the IP (Internet Protocol) from the network layer. The IP protocol is used to communicate to any set of computers in an interconnected network in an LAN or a WAN (Wide Area Network). The IP protocol provides a connection between two nodes across the LAN or WAN. Each node in the WAN receives a unique IP address that facilitates one node's ability to identify another. IP protocols normally use dynamic routing protocols to find alternate routes whenever a link becomes unavailable.

The data-link layer is the final layer in the protocol stack. This layer takes the responsibility of transferring the data from a node to a neighbor node. If the nodes are connected to a Local Area Network (LAN) via Ethernet, it uses the 802.3 protocol, which defines transfer of data between two hosts that are connected via Ethernet. On the other hand, if the nodes use wireless channels, such as those based on Wi-Fi [5], then the MAC 802.11 protocol is used.

An important component in the network layer is the routing protocol, which discovers and maintains routes to all hosts in the network. Examples of routing protocols are RIP and OSPF for wired networks, and AODV and DSR for wireless networks. Below the MAC layer is the physical layer, which sends actual radio or electric signals on the wire or through space. In this chapter, the emphasis will be on ad hoc

routing protocols and wireless medium access control (MAC) protocols. We investigate the weaknesses in the existing routing and MAC protocols and the related literature.

2.1 Ad Hoc Networks

A mobile ad hoc network is a collection of wireless devices moving in seemingly random directions and communicating with one another without the aid of an established infrastructure. Communication protocols for MANETs are designed to work in a peer-to-peer networking mode. To extend the normal coverage of the node, neighboring nodes act as routers. Thus, data may be sent via multiple hops from a source to its destination. There are a couple of technical challenges that must be addressed to make such networks usable in practice. First, since the nodes can be mobile and, thus, the topology can be changing frequently, a highly adaptive routing protocol must be employed to maintain routes between all pairs of source-destination nodes. Second, a number of computing nodes must have fair and efficient access to shared channels. The half-duplex, broadcast nature of the wireless medium makes the design of the medium access control (MAC) protocol nontrivial.

Currently, several routing protocols are under consideration for standardization in the MANET (mobile ad hoc network) working group [47] by the IETF (Internet Engineering Task Force). Progress has also been made in designing random-access MAC protocols based on carrier-sense, multiple access (CSMA). For example, the IEEE standard for the 802.11 [20] specifies the physical and the MAC layer protocols for wireless LANs.

Figure 2.1 illustrates the MANET data transfer between two nodes, using the network protocol stack. The MANET uses the 802.11 MAC protocol on the physical layer to communicate with the next hop in the route. The ad hoc network routing protocol is implemented at the network layer to supplement the functionality of the IP protocol. The MANET uses are not restricted to any particular type of application-layer protocol; users frequently use protocols such as the FTP, HTTP, SMTP, etc. Application layer protocols choose the transport protocol that is best suited to the application.

In the next two sections, we briefly describe the routing-layer protocols and the MAC layer protocols. In the last section, we present the simulation method used for performance evaluation.

2.2 Ad Hoc Routing Protocols

Routing protocols can be divided into proactive and reactive protocols. Proactive protocols constantly maintain routes among nodes. Reactive protocols discover routes on-demand. In general, proactive protocols have a low latency and do not scale well, whereas reactive protocols tend to have a higher latency and scale better than proactive protocols. There are several different dynamic routing protocols in both the proactive and the reactive protocol categories [60, 38, 17, 54]. The advantages and disadvantages of proactive and reactive protocols have been studied extensively [19, 11, 27]. The literature we describe below features a few commonly used routing protocols for ad hoc networks.

2.2.1 Ad Hoc On-Demand Distance Vector (AODV)

The AODV protocol [59, 58] maintains a routing table consisting of <destination node, next hop, number of hops to destination> tuples for each node in the network. When a node attempts to send a data packet to a destination for which it does not already know the route (i.e., it does not have a valid routing table entry), it uses a *route discovery* process to dynamically determine a route. The route discovery works by flooding the network with route request (RREQ) packets. These RREQs are given unique IDs, based on sender and sequence number information. Each node receiving an RREQ for the first time rebroadcasts it, unless it is the destination node or it has a route to the destination that is fresh in its routing table. Such a node replies to the RREQ with a route reply (RREP) packet that is routed back to the original source. RREPs are sent as unicast packets in order to reduce the bandwidth (BW) consumed. To facilitate this, each node that received a RREQ makes a route entry for the original sender of the RREQ, with the node it heard from the next hop. These entries (called reverse path entries) are created at the time the RREQ is forwarded. The reverse path entries expire after a short interval of time that is only sufficient to allow a route reply to be propagated.

Even a single-route discovery results in flooding of the entire network and consumes precious network BW for control activity. Any BW consumed for control activities is an overhead. To reduce the overhead, the AODV uses expanding rings. In the absence of any past information about a destination, the source sends its RREQ with a TTL (Time To Live) of 1 initially. This RREQ is disseminated among nodes that are

1-hop away from the source. If a route reply is not received after the timeout period, another RREQ is sent with a higher TTL; this continues until the MAX TTL is reached (network diameter).

If any active link is broken, the intermediate node that detects it notifies the affected source node using a route error (RERR) packet. The source and any intermediate nodes that receive the RERR packet remove the indicated routes from their routing tables. The RERR propagation works in the following fashion. A set of predecessor nodes is maintained in each routing table entry. They indicate the set of neighboring nodes that send data packets, which are forwarded by this node using the entry provided. These nodes are notified with RERR packets, and each predecessor node processes and forwards the RERR to its own set of predecessors, effectively erasing all routes using the broken link.

RREQs, RREPs and RERRs are the control packets used by the AODV to discover and maintain routes. These control packets are queued in an InterFace Queue (IFQ) between the network and the MAC layers. There are at least three interface queues with different priority levels. Data packets are placed in a lower priority IFQ, and control packets are placed in higher priority IFQ. When the control queue grows in size, data packets are delayed until the control queue is empty. When the network is saturated, there is a significant amount of control activity, which increases the control queue size. There have been few studies on the behavior of ad hoc networks that exceed the point of saturation. Most of the techniques discussed in the literature attempt to reduce the routing overhead in order to reduce congestion and facilitate higher throughput prior to saturation. Castaneda et al. [13] use a query localization technique in another on-demand routing protocol, DSR, to reduce network congestion and to improve end-to-end delay. Similarly, Geral et al. [31] use passive clustering to reduce the routing overhead. Gu et al. [34] describe an embedded mobile backbone that is dynamically constructed to form a 2-level physical, heterogeneous-multihop wireless network in order to eliminate network-wide route broadcasts. Das et al. [49] use route caches to reduce the congestion.

An alternative approach to reduce the routing overhead is to reduce the need for route discovery. When the network is at or beyond the point of saturation, the 802.11 MAC protocol causes frequent *false* route breaks [24]. If the next hop does not respond, even when it is within the radio range of a transmitting node, then it makes the latter falsely conclude that the route is broken. Such route breaks are termed false route breaks. Reducing these false route breaks reduces the need for route discovery. Ray et al. [62] reduced the

number of transmission failures reducing the instances a channel is reserved but unused. On the other hand, Xu et al. [76] reduced the number of transmission failures by reducing the number of collisions, which they achieved by restricting the communication distance. Distant nodes (low received signal strength) are more likely result in a collision than the nodes that are near (strong signal strength).

2.2.2 The Dynamic Source Routing Protocol for Mobile Ad Hoc Networks (DSR)

The dynamic source routing protocol uses source routing to deliver data. Each packet in DSR should contain an ordered list of nodes through which data should pass. This allows a source node to direct the path taken by its data packets, unlike in AODV. When a node attempts to send a data packet to a destination for which it does not already know the route (i.e., it does not have a routing table entry), it uses a "route discovery" process to dynamically determine a route. This route discovery process is similar to that used by AODV.

Route discovery works by flooding the network with route request (RREQ) packets. These RREQ packets contain the source and destination address, along with the unique identity of the request. When a node receives an RREQ and if it is the destination of the RREQ, it responds with an RREP containing the accumulated route from the source to the destination. Otherwise, the node appends its own address to the RREQ header and rebroadcasts the RREQ to its neighbors. If a node that is not the destination node indicated in the RREQ receives the RREQ (request) and contains a route to the destination, it may send an RREP. Upon receiving the RREP, the source node records the route indicated in RREP in its route cache.

Route maintenance

When the MAC protocol reports a transmission failure, called link layer feedback, DSR will initiate a route failure. It removes all of the routes that use the broken link from the route cache and sends an RERR to each previous node that used this link. When a connection's source node receives this RERR, it removes the route to the destination from its route cache and uses another route from its route cache, if one is available. If there is no alternate route in the route cache, it initiates a route discovery to find a new route.

If the intermediate node that detected the broken link has an alternate route to the destination, then it uses "data salvage" (routing a data packet on an alternate route) to reduce the number of dropped packets.

If there is no alternate route in the route cache, all of the packets in the IFQ which list the broken link as the next hop are dropped.

Route cache: Each node contains a route cache. Each entry in route cache completely specifies the intermediate nodes to a destination. Frequently, this route cache is used to respond to RREQs, even if it is not the destination. The route cache is updated when it learns a new route; routes are learned from RREQ, RREP and DATA packets that are routed through the node. To enrich the route cache, the DSR recommends promiscuous listening, which helps the node learn more routes. Entries from the route cache are removed when a node receives an RERR.

Impact of using route cache and data salvage in DSR: Published studies [38, 19] indicate that DSR with optimized route cache and data salvage can provide good performance in a dense, low-speed MANET. However, highly mobile MANETs may not benefit from data salvage or the use of a route cache for route reply. To demonstrate this, we simulated 100-node MANET used in Chapter 1 with 50 CBR connections and DSR as the routing protocol.

Figure 2.2 shows the performance of the original DSR, the DSR without the use of a route cache for RREPs (DSR-no-cache), the DSR without data salvage (DSR-no-salvage), and the DSR without route cache and data salvage (DSR-no-cache-salvage). As the node mobility increases, routes are broken frequently; and route caches contain many stale (broken) routes. A MANET with the proposed DSR optimizations suffers from performance losses due to stale routes in high mobility situations.

2.2.3 Optimized Link State Routing Protocol (OLSRP)

The optimized link state routing protocol is a table-driven protocol that proactively seeks routes. Nodes periodically exchange topology information proactively. Each node in the MANET selects a relay agent from its neighbor nodes. This relay node is identified as the “multipoint relay” (MPR). Only the selected MPRs are responsible for relaying topology information. This helps the OLSR to reduce the control overhead and to maintain the routes.

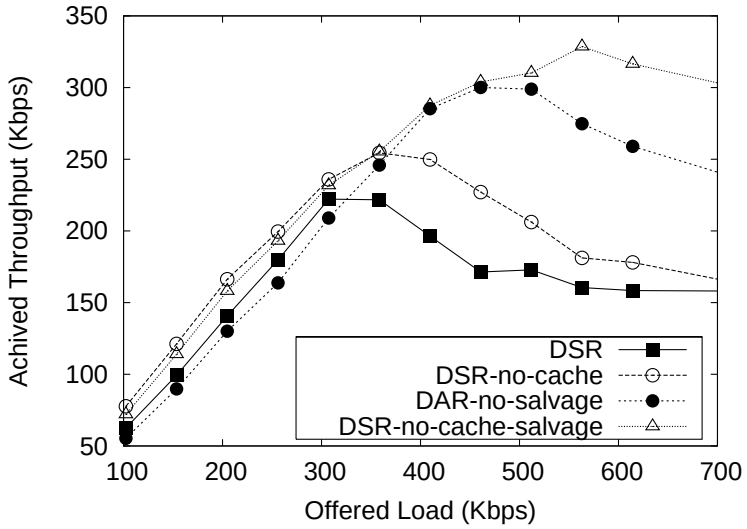

Figure 2.2: DSR performance with options recommended.

2.2.4 Topology Dissemination Based on Reverse-Path Forwarding (TBRPF)

Topology Dissemination Based on Reverse-Path Forwarding (TBRPF) [54] is a proactive link-state routing protocol. Each node in the MANET computes the source tree containing all of the shortest paths to all of the reachable nodes. Each node then broadcasts HELLO messages, which contain a portion of its source tree, to reduce routing overhead. Therefore, nodes must receive multiple HELLO messages to learn the entire tree. In a highly mobile MANET, the TBRPF allows the nodes to broadcast the entire tree in order to improve the topology's robustness. HELLO messages are broadcast periodically, along with a list of nodes that have established a 1-way communication. A node can establish a 2-way communication when the node receives a HELLO that includes its own ID or an UPDATE REQUEST message. Once the node establishes a 2-way communication, the node sends an update request to the new neighbor and awaits an UPDATE reply message. The reply message will confirm that the communication is 2-way, and this allows the nodes to exchange topology information.

When there is no HELLO response from a neighbor for a specified period of time, it is considered lost. At this time, the node may conduct a differential update of its topology. If the MAC protocol provides transmission failures, a similar procedure is performed. The failed link is marked as a lost link, and the node may send a differential update to its neighbors. In this dissertation, we do not consider proactive routing

protocols.

2.3 MAC Protocols

The wireless medium is a broadcast medium that is shared by multiple devices. In a given instance, only one device may access the medium. If multiple devices access the medium at the same time, it will result in garbled data (a collision), making communication difficult. Therefore, a wireless communication must have a set of rules that each device must follow in order to avoid collisions. These rules are set by the wireless MAC protocol, which must be efficient and fair. There are several MAC protocols that have been proposed and standardized. Later in this chapter, we will give a brief description of the popular IEEE 802.11 MAC protocol.

The wireless medium's unique properties make the design of the wireless MAC protocol very different from, and more complex than, the wired MAC protocol. For example, it is very difficult to transmit and receive signals at the same time (typically, transmitting signal power and receiving signal power differ by order of magnitude). Therefore, collision detection is not possible while transmitting. This makes the wireless MAC protocols different from the wired MAC protocols. As a result, the MAC protocols must be designed to reduce the probability of collision; such protocols are called collision avoidance (CA) protocols.

Radio signals are subject to reflection, diffraction, and scattering. The signal received by a mobile device can be time-shifted, and the received signal power can vary as a function of time. This is known as *multipath propagation*. When the change of power is greater than the specified value, the device is considered to be in a "fade". As a result, wireless communications can be "'bursty"'; in other words, a channel could be lost for short durations of time.

Another challenge in wireless networks is the *hidden node* (hidden terminal) [68]. A hidden node is a wireless device that is within the range of the intended receiver but out of the range of the source of a wireless transmission. For example, in Figure 2.3, nodes A and B are within each other's communication range. Node C is within B's communication range, but not A's. When node A is communicating with node B, node C cannot hear the transmission from node A. During node A's transmission to node B, if node C senses the media, it falsely concludes that the node's media is idle. Therefore, node C is hidden from node

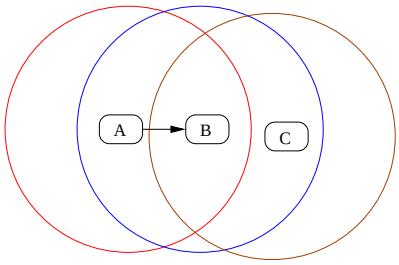

Figure 2.3: Hidden node. Node C is hidden from node A.

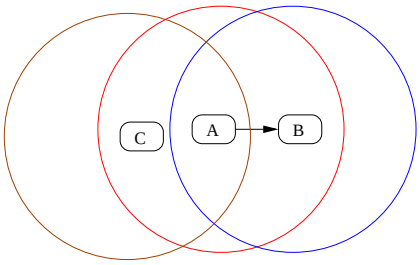

Figure 2.4: Exposed node. Node C is exposed to node A's transmission to node B.

A. If node C transmits during node A's transmission, the node B's reception of node A's transmission is disrupted. Hence, hidden nodes can cause data transmission collisions.

Another situation that wireless networks must try to avoid is *exposed nodes*. Exposed nodes are the opposite of hidden nodes. An exposed node is a node that is within the source node's range but out of the destination node's range. For Example, in Figure 2.4, nodes A and B are within each other's communication range. Node C is within the communication range of node A, but not within the range of node B. Thus, when node A is transmitting to node B, node C will sense the wireless medium is busy. However, any transmission from node C cannot reach node B. Thus, node C could have transmitted without interfering with the transmission from node A to node B. In this case, node C is an exposed node to node B. If exposed nodes are not minimized, the bandwidth is underutilized.

When a node is receiving a transmission from two different sources, one transmission is stronger than the other. Therefore, the node source with the stronger signal can capture the wireless medium. A capture of this nature may improve the performance, but it results in unfairness.

Random burst errors, fading, and capturing result in data delivery failures and ultimately result in dropping of the data. This presents several problems for transport layer protocols. A prominent example of this is the behavior of the windows-based congestion control algorithm in TCP. TCP interprets data packet drops as a sign of congestion downstream and folds-back its congestion window, leading to a poor utilization of the wireless channel. Several mechanisms are proposed in the literature to augment TCP so that it can distinguish between packet losses due to random channel failures and those due to congestion at the router [6, 46, 71]. However, such solutions are specific to one higher layer protocol (i.e., TCP). Aside from the be-

havior of a higher layer network protocol, there is a concern as to whether the traditional packet scheduling mechanisms of the network interface are efficient in wireless networks. In a traditional operating system, the device driver typically does the scheduling in the network interface on a first-come-first-serve (FCFS) basis. At this level, the packets are no longer identified by which "connection" or "flow" they belong to. If a network interface is multiplexing several flows, then all packets are queued in the same driver queue and sent to the hardware interface on an FCFS basis, as and when the interface is available. This simplistic scheduling technique works fine for wired network interfaces. However, in a shared wireless medium, the radio link characteristics are location dependent, and the "link quality" between a host and its neighbors may vary independently. Channel errors due to fading and multipath propagation may be dependent on the location of the neighbor; the hidden node problem [68] may affect one neighbor but not another, and so on. Such location-dependent errors necessitate the consideration of *channel state dependent scheduling* (CSDS) mechanisms [9] at the radio interface.

2.3.1 Carrier Sense Multiple Access (CSMA) protocol

The CSMA is the most primitive MAC protocol [41] for wireless access. In this protocol, a node that needs to transmit senses the channel for ongoing transmissions. If the node detects an ongoing transmission, then it waits for a random amount of time before it re-attempts the transmission. If the node detects no transmission, then it can begin transmission.

2.3.2 Multiple Access Collision Avoidance (MACA) protocol

The multiple access collision avoidance protocol [39] improves on the CSMA. In this protocol, the CSMA is improved by reducing the hidden nodes. MACA uses Request-to-Send (RTS) and Clear-To-Send (CTS) control packets to announce the packets to be transmitted. A node that needs to transmit, broadcasts a RTS packet containing the length of the DATA packet to be transmitted. Upon receiving an RTS, the destination node sends a CTS packet containing the length of the DATA packet to follow. All nodes receiving the RTS or the CTS refrain from transmitting for the expected duration of the transmission.

2.3.3 Flow Acquisition Multiple Access (FAMA) protocol

The flow acquisition multiple access protocol [30] improves on the MACA protocol. It uses the same RTS-CTS-DATA exchange as the MACA does. Before transmitting an RTS, a node checks its carrier to see if there is an ongoing transmission. If the channel is busy, the node calculates a random backoff time and waits for this period before it rechecks the channel sense. This random backoff helps prevent RTS collisions.

2.3.4 MACAW: Media access protocol

MACAW is an extension of the MACA protocol. The node receiving the DATA is required to send an ACKnowledgment (ACK). The protocol uses an RTS-CTS-DATA-ACK exchange to complete the data transmission.

2.3.5 802.11 MAC Protocol

The IEEE 802.11 protocol [20] provides peer-to-peer networking, using a Distributed Coordinate Function (DCF) based on a Carrier Sense, Multiple-Access with Collision Avoidance (CSMA/CA) protocol. To implement the CSMA/CA, the 802.11 MAC uses many techniques. A node that needs to transmit must first sense the medium; if the medium is busy, then the node defers, using an exponential backoff time. This backoff time is counted down only when the channel is idle. If the medium is free after the completion of backoff, then the node waits for the Distributed Inter Frame Space (DIFS- 50 μ seconds) specified by the 802.11 standard. After the DIFS time elapses, the node senses the medium one more time before transmitting. If the medium is busy, it defers; otherwise, it transmits. If a packet is received by the next node without a collision, then the receiver will transmit an ACK after differing for a Small Inter Frame Space (SIFS 10 μseconds) length of time specified by the IEEE 802.11 MAC protocol. An ACK packet is used by the receiver to confirm the successful reception of DATA from the sender.

In order to reduce collision, the 802.11 MAC protocol uses both the physical carrier sense and the virtual carrier sense. A mobile node physically senses the carrier; when the noise level is higher than a preset limit, the channel is said to be busy. To maintain the virtual carrier sense, each transmission maintains the duration of the channel usage for the current communication. A mobile node can begin to use the radio channel only

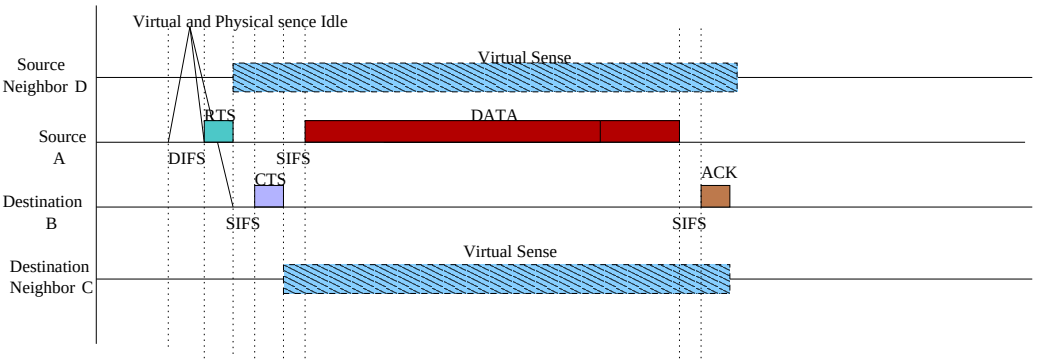

Figure 2.5: 802.11 MAC protocol DATA transfer with RTS, CTS and ACK control packets.

when both the physical sense and the virtual sense indicate that the channel is idle.

To overcome the inherent problems of the collision avoidance protocol in wireless communication, the 802.11 protocol uses optional link layer control packets, denoted as RTS (Request-to-Send) and CTS (Clear-To-Send). RTS/CTS packets are used by the sender and the receiver of a unicast communication to notify all nodes around them of the duration of the channel usage [68]. An RTS will contain the intended receiver of the pending DATA transmission and the duration of the channel usage. In response, the receiver will reply with a CTS if it determines that the channel is free, with the exception of the RTS it received. All nodes receiving an RTS or a CTS will set their virtual carrier sense in the Network Allocation Vector (NAV) for the duration specified in the RTS/CTS packet. Information in the NAV, along with the physical carrier sense, will give the status of the carrier sense of the node.

This RTS/CTS transaction reduces the probability of collisions with hidden nodes. The RTS/CTS exchange is used only for unicast packets larger than the *RTSTheshold* specified by the standard. It must be noted that RTSs/CTSs are relatively smaller than the DATA packets and have less likelihood of colliding.

Figure 2.5 illustrates the data transfer from node A to node B. Assume that node D is a node in node A's communication range and outside of node B's communication range; similarly, node C is in node B's communication range, but it is outside of node A's communication range. To initiate a unicast data transfer, the sender (node A) must send a RTS if the virtual and the physical carrier senses indicate that the channel is idle. If it is idle, then the node defers the transmission for one DIFS and initiates the transmission of the RTS. Nodes receiving the RTS (nodes D and B in this example) update their NAV to the transmission

duration (virtual sense) indicated in the RTS frame. Node B, the intended receiver of the RTS, sends a CTS frame after deferring for the Small Inter Frame Space (SIFS) time period, if the NAV (prior to the reception of the RTS) at the receiver indicates that the channel is idle and the physical sense indicates that the channel is idle. All nodes receiving the CTS (node C) update the NAV table to the transmission time specified by the CTS packet. Node A then sends the DATA packet after the SIFS time period has passed, assuming that the channel is reserved. Upon successfully receiving the DATA, node B sends an ACK to node A.

The physical sense can be viewed as the measurement of signal interference at the node. Interference affects the performance of the wireless network. There are two types of interference (noise) in wireless networks. The first type of noise is caused by external signal interference (by devices such as cordless phones and Microwave ovens, which may use the same frequency band as the 802.11 MAC devices). The second type of noise is caused by distant transmissions between other nodes in the ad hoc network. In our work, we consider only the latter type of interference, which is caused by a distant node communication. It is shown that interference causes unfairness between communications and a capture effect in the MANETs [74, 73]. A channel capture occurs when two nodes are transmitting and one node's received signal power is significantly stronger than the other's. As a result, only the node with stronger signal power is able to communicate, and the node with the weaker signal does not receive any bandwidth.

Recently, several researchers investigated the effectiveness of the 802.11 MAC protocol and its impact on the overall MANET performance [74, 73, 76, 36]. Simulation studies of ad hoc networks have shown evidence of unfairness (unfair distribution of channel access) at the MAC layer, which causes short and long-term unfairness (bandwidth distribution) in the application layer. Hu and Saadawi [36] demonstrate that the TCP connection with the strongest signal (relative to the noise level) can capture the channel and exacerbate the unfairness problem. In another study of the MAC layer [62], the authors show that the RTS/CTS handshaking protocol cannot prevent all of the interference.

2.4 Performance Analysis

The MANET testbeds are still in the primitive stages. Although there have been some experimental works, they are based on small-scale testbeds [16, 48]. To implement a larger-scale testbed with hundreds of nodes,

there must be a large resource pool that includes at least one person per testbed node. Another drawback to testbed-based evaluation is that it is difficult to repeat the experiments for comparison and validation. At the time of this writing, there are several implementations of the MANET routing protocols. The routing protocols include AODV, DSR, LOSR, TBRPF, and many others. In order to overcome the difficulties in using a physical testbed, there are several studies that use emulated testbeds. In some of these studies, emulation is used for node mobility [25, 79], and in the other studies the MANET network's ad hoc network protocols are tested in static nodes [25]. Implementations for ad hoc networking protocols are primarily for the Linux operating system, though there are a few implementations for the BSD and Windows Operating systems.

Because of the difficulties and limited configurations in real and emulated testbeds, simulators are frequently used in MANET research. Simulation allows for large-scale network simulations and provides a good platform to compare two or more routing protocols. There are several popular simulators used by the research community: ns, Glomosim, OPENT, and a few others. Comparative performance of three of the popular simulators is described in [14].

2.4.1 The Network Simulator

The network simulator (ns) from Lawrence Berkeley National Laboratory is, thus far, the most popular simulator in the MANET research community [1]. The ns simulator development began in 1989 and quickly gained the confidence of the network research community. The wireless extension to $ns-2$ (version 2 of ns simulator) simulator was provided by the Monarch Project [33]. There were several ad hoc routing protocols (for example, DSR, AODV, etc.) implemented for $ns-2$. The $ns-2$ has an extensive collection of transport protocols and MAC protocols to support various wired, as well as wireless, protocols.

2.4.2 Glomosim

Another popular simulator is the Glomosim [78], a scalable simulator for wireless simulations. The Glomosim is implemented using the PARSEC (PARallel Simulation Environment). A noticeable difference in Glomosim from the $ns2$ is that Glomosim provides a summary of results that helps the simulator run faster.

Another advantage of using the Glomosim is that it uses a more realistic modeling of a wireless channel. $ns2$ approximates a node's noise to be the noise of the highest noise source, whereas Glomosim computes it as the sum of all noise sources in the MANET.

One primary weakness in the Glomosim is the lack of various TCP implementations. QualNet, the commercial simulator of Glomosim, has a few additional TCP implementations.

Glomosim mobility model: Glomosim supports three mobility models: *Random drunken, Random waypoint*, and *trace-based* mobility. In the *Random Drunken Model*, a node selects a random position to move to from a current position. The random position is either up, down, left or right. The node will move to this random position if the next position is within the physical terrain. The random waypoint model requires parameters, minimum speed, maximum speed, and pause time [63, 8]. In this model, a node randomly selects a geographical destination in the physical terrain. The node then travels toward the destination from its current position at a speed between the specified minimum and maximum speed that is uniformly chosen. When the node reaches the destination, it pauses for the specified *pause time*. After the pause time expires, it chooses another random destination. Optionally, the user can specify the node mobility in a trace file.

Radio model: The Glomosim radio model supports two basic path-loss models: free space and two-ray. In addition, there are two fading models: Rayleigh distribution and Ricean distribution.

Routing protocol: Glomosim supports the AODV, DSR, LAR1, FISHEYE, and STATIC routing protocols. Glomosim uses the AODV draft 3, as specified by the IETF. Glomosim DSR's implementation is modeled after the $ns-2$'s DSR implementation, with the exception of the following optimizations: Preventing route reply storms, path state and flow-state mechanisms, piggybacking route discovery, and gratuitous route error. LAR implementation follows [42] closely.

Baseline MANET configuration: One hundred nodes in a 1200×1200 m^2 terrain is used as the baseline MANET in this dissertation. Unless otherwise specified, this is the baseline simulation setup is used for performance evaluation. In this baseline MANET, The node movement is patterned by the random waypoint

model with speeds in the range of [1,19] m/sec. Each simulation was run for 600 seconds (the first 100 seconds are used to warm-up the network, and no statistics were collected during this time). The simulations were repeated 9 times with different random seeds, and these were averaged to minimize the impact of worst-case and best-case scenarios. Radio transmit power is set at 15 dBm and radio receive threshold, radio receiver sensitivity, radio propagation limit and signal-to-noise ratio is set at -81 dBm, -91 dBm, -111 dBm and 10 dB, respectively.

Correction to AODV protocol implementation in Glomosim

We identified an instance of incorrect implementation, which leads to severe performance degradation under high loads. Upon receiving an RERR (route error), AODV removes the a specified routes to the destination if the node that transmitted the RERR is the next hop. In the AODV version included in the Glomosim 2.03, we found that the node, upon receiving an RERR, removes its route to the destination, even when it hears from a node that is not in its next hop. This can lead to the removal of routes that are valid. For example, let's assume that node A has a route to node D via node B and C, and that node E also has a route to node D via node F, which is independent of the route from node A to node D. If the link fails between nodes C and D, then node C sends an RERR to remove the route to node D via node C. Based on the Glomosim simulator, node B and node E will both remove the route to node D. According to the AODV specification, only node B's route to node D must be removed. In our work, we corrected this mistake. Figure 2.6 shows the performance of the AODV as included in Glomosim (AODV-error) and the corrected version (AODV).

2.4.3 OPENT

OPENT simulator is one of the few popular, commercially available simulators extensively used in the research industry. The simulator is very robust; it has an extensive collection of libraries of networking protocols, and various types of MAC, routing, and transport protocols.

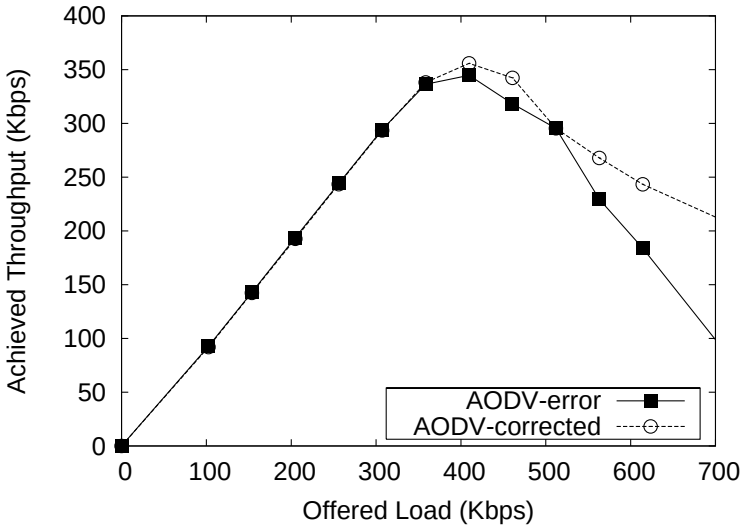

Figure 2.6: AODV performance after fixing ERROR.

2.4.4 Performance metrics

Ad hoc networks are complex networks with several factors that influence network performance. For example, factors such as the number of nodes (node density), terrain size, node mobility (patterns and speed), the routing protocol used, MAC protocol used, and traffic patterns (application and different transport protocols) are commonly used parameters to simulate different types of MANETs. To evaluate the performance of these varying MANETs, it is common to use throughput, delivery rate and end-to-end delay. To evaluate the effectiveness of the routing protocols, researchers use routing packets per data packets delivered, route repair time and end-to-end delay. In our work, we primarily used throughput and end-to-end delay as the performance metrics. In addition, we show the number of hops that packets travel to show that our proposed modifications do not favor shorter over longer routes to achieve higher throughputs. In addition, we also compute Jain's fairness index [37] provided below, to show that our proposed modifications do not increase unfairness.

$$\text{Fairness Index} = \frac{\left(\sum_{i=1}^{N} \gamma_i\right)^2}{N \sum_{i=1}^{N} \gamma_i^2}$$

Where γ_i is the number of packets received for each communication during a fixed simulation time.

Chapter 3

On the Impact of Noise Sensitivity in 802.11 MAC Protocol

Interference with transmissions affect the performance of wireless networks significantly. There are two types of interference (noise) in wireless networks. The first type of noise is caused by external signal interference (for example, devices such as cordless phones and Microwave ovens, which may use the same frequency band as the 802.11 device). The second type of noise is caused by distant transmissions in the ad hoc network. Using simple network configurations, we show that the 802.11 MAC performs poorly when the noise level around the intended receiver of a transmission is higher than a predefined threshold value because the protocol prevents it from responding to its sender's transmission. This increase in noise could be due to distant wireless transmissions or some other random source. In this chapter, we consider only the interference caused by distant wireless transmissions. This type of interference causes unfairness and a capture effect in MANETs [74, 73].

We show that the physical carrier sense mechanism, as designed and used in the 802.11 MAC, could have a crippling effect on distant but competing transmissions. We propose a modification to mitigate this situation and show, using simulations, that the proposed modification provides up to 50% higher UDP throughput for static wireless networks, and up to 33% and 44% higher throughput for UDP and TCP on mobile ad hoc networks. We further demonstrate that the proposed modification is beneficial to MANETs, regardless of node density and routing protocol.

3.1 Modeling Signal Strength

To explain the propagation characteristics of the radio channel used for 802.11 MAC based networks, we use the analytical model of two-ray path loss for signals [29] and an implementation of the 802.11 MAC model in Glomosim [78]. Several other studies used the currently available hardware and came up with similar parameter values [25, 32].

The signal strength at distance d, denoted by $P_r(d)$, can be computed using the two-ray signal model:

$$P_r(d) = \frac{P_t G_t G_r h_t^2 h_r^2}{d^4 L} \tag{3.1}$$

Here, $P_r(d)$ is the received power at distance d meters, P_t is the transmitting power, G_t and G_r are the antenna gains in transmitter and receiver respectively, h_r and h_t are heights of the receive and the transmit antennas, and L is the system loss.

The two-ray model does not give an accurate result for short distances, due to oscillations caused by the constructive and destructive combination of the two-rays. For short distances, the free-space path model is commonly used [29]. In the free space path model, λ is wavelength in meters. The signal strength is given by (3.2).

$$P_r(d) = \frac{P_t G_t G_r \lambda^2}{(4\pi)^2 d^2 L} \tag{3.2}$$

The cross-over distance d_c is set as distance d, where the free-space model and the two-ray model result in the same receive power P_r. Therefore, for d $< d_c$, free-space model is used, and for d $\geq d_c$, two-ray model is used [29]. With this information, we could compute the cross-over distance as:

$$P_r(d_c) = \frac{(P_t h_t^2 h_r^2)}{d_c^4 L} = \frac{P_t \lambda^2}{(4\pi)^2 d_c^2}$$

$$d_c = \frac{4\pi h_t h_r}{\lambda}$$

The following receive/transmit antenna heights and wavelength are used in our simulations:

h_r = h_t = 1.5 m, and λ = 0.125 m for 2.4 GHz frequency. Therefore,

$$d_c = \frac{4\pi(1.5 \times 1.5)}{0.125} = 226.28\ m$$

For distances $\leq$ 226 m, the free-space model is used, and for other distances ($\geq$ 227 m) the two-ray model is used [29].

It is common to use G_t = G_r = 1 and L = 1 [29]. For $d < d_c$,

$$\begin{aligned} P_r(d) &= \frac{P_t G_t G_r \lambda^2}{(4\pi)^2 d^2 L} \\ &= \frac{P_t(1)(1)\lambda^2}{(4\pi)^2 d^2 (1)} \\ &= P_t \times \left(\frac{\lambda}{4\pi d}\right)^2 \\ &= P_t \times \left(\frac{0.125}{4\pi d}\right)^2 \text{ mW} \end{aligned} \tag{3.3}$$

For $d > d_c$,

$$\begin{aligned} P_r(d) &= \frac{P_t G_t G_r h_t^2 h_r^2}{d^4 L} \\ &= \frac{P_t(1)(1) h_t^2 h_r^2}{d^4 (1)} \\ &= P_t \times \left(\frac{h_t h_r}{d^2}\right)^2 \\ &= P_t \times \left(\frac{1.5 \times 1.5}{d^2}\right)^2 \\ &= P_t \times \left(\frac{2.25}{d^2}\right)^2 \text{ mW} \end{aligned} \tag{3.4}$$

The transmission power in milliWatts (mW) can be converted to dBm using (3.5) and from dBm to mW using (3.6).

$$\text{Power in dBm} = 10 \times log_{10}(\text{Power in milliWatts}) \tag{3.5}$$

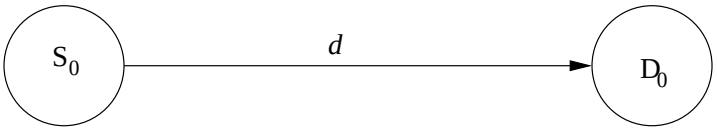

Figure 3.1: Node interference setup.

$$\text{Power in milliWatts} \quad = \quad 10^{(\text{Power in dBm}/10)} \tag{3.6}$$

Using (3.5), the signal strength given by (3.3) and (3.4) can be converted to dBm.

$$\begin{aligned} P_r(d) &= 10 \times log_{10}\left(P_t \times \left(\frac{0.125}{4\pi d}\right)^2\right) \text{ dBm}, \quad d \leq 226m \\ &= 10 \times log_{10}(P_t) + 10 \times log_{10}\left(\frac{0.125}{4\pi d}\right)^2 \text{ dBm} \\ &= 10 \times log_{10}(P_t) + 20 \times log_{10}\left(\frac{0.125}{4\pi d}\right) \text{ dBm} \end{aligned} \tag{3.7}$$

$$\begin{aligned} P_r(d) &= 10 \times log_{10}\left(P_t \times \left(\frac{2.25}{d^2}\right)^2\right) \text{ dBm}, \quad d \geq 227m \\ &= 10 \times log_{10}(P_t) + 10log_{10}\left(\frac{2.25}{d^2}\right)^2 \text{ dBm} \\ &= 10 \times log_{10}(P_t) + 20log_{10}\left(\frac{2.25}{d^2}\right) \text{ dBm} \end{aligned} \tag{3.8}$$

To calibrate the received signal strength in the Glomosim simulator, a simple 2-node network shown in Figure 3.1 is simulated. The distance between the two nodes (d) is varied from 1 m to 2200 m. Various parameters used in modeling the transmission, propagation and reception of the radio signal are indicated in Table 3.1. The transmitting power of node S_0 is set at 15 dBm and node D_0's receiving power is measured using the simulation. We used (3.7) and (3.8) to calculate this value and verified that the simulator model matches the analytical model. Figure 3.2 shows the signal strength at various distances from the transmitter.

Table 3.1: Default parameters for the radio model used in Glomosim simulator.

Parameter	Value	Comments
RADIO-TYPE	RADIO- ACCNOISE	Do not ignore noise in commutations
RADIO-RX-TYPE	SNR-BOUNDED	Signal-to-noise ratio error model is used; SNR $\geq$ 10 dB for correct reception
RADIO-FREQUENCY	2.4 GHz	Transmission Frequency
RADIO-BANDWIDTH	2.0 Mb/s	Nominal channel BW
RADIO-TX-POWER	15.0 dBm	Transmitting power
RADIO-RX-SENSITIVITY	-91.0 dBm	Physical carrier sense: Physical Carrier sense is busy if the noise level ≥ -91.0 dBm
RADIO-RX-THRESHOLD	-81.0 dBm	Minimum signal strength required to receive a transmission
PROPAGATION-LIMIT	-111.0 dBm	Power at which simulated effects of transmission ceases

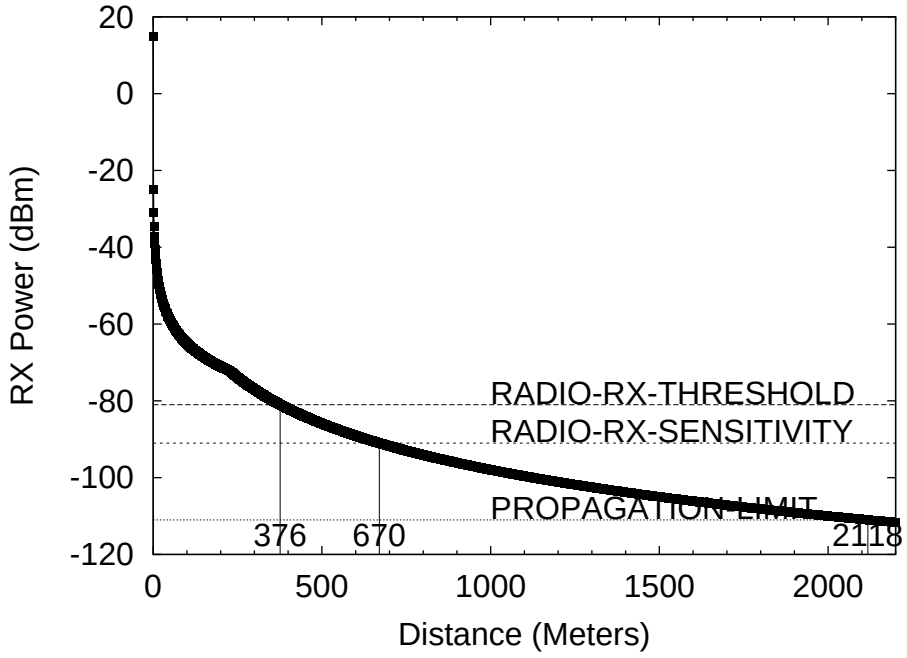

Figure 3.2: Signal strength as a function of distance from the transmitting node.

3.2 Communication Regions

With respect to a transmission, nodes can be in one of four regions, depending on their distance from the sender; they are the Communication, Sensing, Noise, and Non-interfering regions. The four regions are separated by three levels of receiving signal power [20]. A node in the Communication region can receive a transmission with a signal power greater than the RADIO-RX-THRESHOLD, which is -81 dBm in the Glomosim implementation. Any signal power level received below the RADIO-RX-THRESHOLD is accumulated to the node's noise level, and the signal cannot be received. In the 2-node setup in Figure 3.1, signal strength reduces to -81 dBm at a distance of 376 m from S_0. Therefore, D_0 can receive the data transmitted by S_0 at distances up to 376 m. In this simulation setup, there is only one transmitting node; therefore, there is no additional noise. If there are nodes communicating beyond node D_0's communication range, the destination node's SNR threshold of 10 dB must be satisfied in order to receive the data. Therefore, in the absence of any other communication or external interference, 376 m is the effective communication range of the wireless network. A node 377 m away from a transmitting node simply senses increased noise levels.

The Sensing region refers to instances where the signal strength is between the RADIO-RX-THRESHOLD and the RADIO-RX-SENSITIVITY level. In this case, the RADIO-RX-SENSITIVITY is used to determine the physical sense of the medium. If the total noise level is above the RADIO-RX-SENSITIVITY level, then a node senses a busy physical medium and goes into a sensing state. The typical value of the RADIO-RX-SENSITIVITY is -91 dBm [18, 70], which is also used in the Glomosim simulator. The signal strength reduces to -91 dBm at a distance of 670 m from the transmitter. In Figure 3.2, for d between 376 m and 670 m, D_0 receives a signal power between the RADIO-RX-SENSITIVITY and the RADIO-RX-THRESHOLD.

In addition to the noise generated by distant transmissions, some low level thermal and ambient noise is present and sensed by mobile nodes. These noises are considered to be background noise in the simulation model and added as a constant to calculate the noise level of a node. By probing the simulator, we measured the background noise used in the Glomosim simulator to be -100.97 dBm. Therefore, in the two node setup, node D_0's total noise is the sum of the background noise and the noise from node S_0's transmission. In the absence of any competing transmissions, a transmission can be successfully received at up to 376 m away

from the source. However, the range over which the node is caused to go into a sensing state is increased by the background noise.

For D_0 to sense a busy physical medium,

$$S_{0,\text{ Total noise}} > \text{RADIO-RX-SENSITIVITY}$$

$$S_{0,\text{Background noise}} + S_{0,noise} > \text{RADIO-RX-SENSITIVITY}$$

Substituting -100.97 dBm for background noise and -91 dBm for RADIO-RX-SENSITIVITY, we get the following:

$$-100.97dBm + S_{0,noise} > -91dBm$$

This can be rewritten in milliWatts, using (3.6), for easier manipulation:

$$\begin{aligned} S_{0,noise} &> 7.94 * 10^{-10}mw - 7.99 * 10^{-11}\ mw \\ &= 7.14 * 10^{-10}\ mw \end{aligned}$$

Next, milliWatts can be converted to dBm using (3.5):

$$S_{0,noise} = -91.47\ dBm \tag{3.9}$$

Since background noise is added to the total noise, node D_0 can now tolerate a noise level of -91.47 dBm (see Equation 3.9) from node S_0. D_0 receives a -91.47 dBm noise level when d = 688 m. At distances 377 m to 688 m from S_0's transmission, D_0 emits a noise level of -91 dBm or greater. Therefore, the sensing range is increased to 377 to 688 m, from 377 to 670 m.

At power levels below the PROPAGATION-LIMIT, the signal is diminished and will not cause any interference to the nodes. The range between the RADIO-RX-SENSITIVITY and the PROPAGATION-LIMIT is considered to be the Noise region. In this region, S_0's transmission noise alone is insufficient for D_0 to sense a busy medium. The noise from S_0 is added to the total noise of D_0. However, if the sum of noises at D_0 exceeds the RADIO-RX-SENSITIVITY because of multiple distance transmissions, then D_0 senses a busy medium.

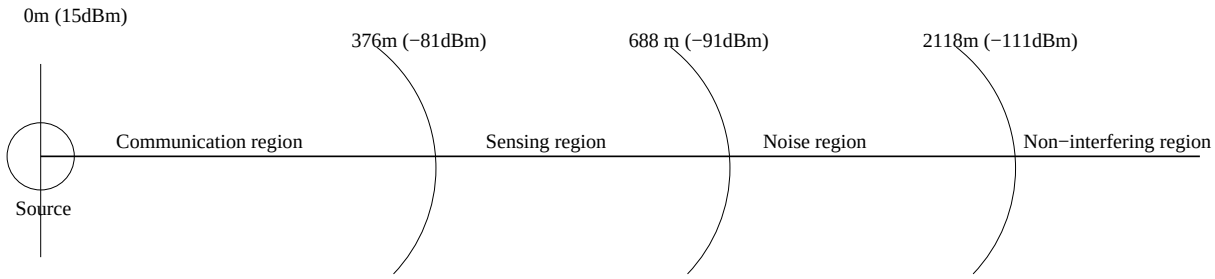

Figure 3.3: Different regions in the propagation of a transmission with 15 dBm transmission power and free-space signal propagation up to 226m andtwo-ray signal model for larger distances.

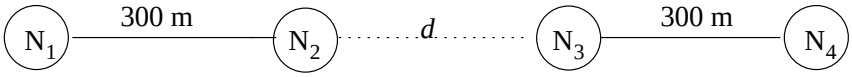

Figure 3.4: A four-node network to illustrate the impact of the competing transmissions. There are two connections: $C_{1,2}$ denotes the communication between N_1 and N_2, and $C_{3,4}$ between N_3 and N_4. The distance d between N_2 and N_3 is varied.

The PROPAGATION-LIMIT defines the signal level at which a node is unable to influence a distant node. Often, -111 dBm is used as the PROPAGATION-LIMIT. A signal power of -111 dBm is received at a distance of 2118 m away from the communicating node. Therefore, a transmission has no impact on a node beyond 2118 m. Figure 3.3 illustrates the four regions, the distance from the source to each region, and the power level.

3.3 Impact of Competing Transmissions

Using a simple 4-node network, we show the impact of a competing transmission on the throughput achieved for a communication of interest. The network, shown in Figure 3.4, has 4 static nodes N_1 through N_4 arranged in a straight line. The distance between N_2 and N_3 is d. The other two distances are fixed at 300 m each. There are two constant bit rate (CBR) transmissions: $C_{1,2}$, which denotes the communication between N_1 and N_2, and $C_{3,4}$, which denotes the communication between N_3 and N_4. Depending on the direction of the data transmissions, 4 different simulation topologies are possible (see Table 3.2). Since T_1 and T_4 topologies are symmetrical, T_4 topology is not further considered here.

At each source, the maximum sustainable load of 1.62 Mb/s is offered with CBR (though the nominal channel BW is 2 Mb/s, the actual throughput achieved is 1.62 Mb/s, due to the overhead of various pro-

Table 3.2: Simulation Topologies.

Topology	$C_{1,2}$	$C_{3,4}$
T_1	$1 \Longrightarrow 2$	$3 \Longrightarrow 4$
T_2	$1 \Longrightarrow 2$	$3 \Longleftarrow 4$
T_3	$1 \Longleftarrow 2$	$3 \Longrightarrow 4$
T_4	$1 \Longleftarrow 2$	$3 \Longleftarrow 4$

Table 3.3: Achieved CBR throughputs for the 4-node network with $d = 689$ and 688 meters. The numbers in bold indicate significant reduction in throughputs when N_2 and N_3 are within the sensing range of each other.

Topology	Achieved throughput (Kbps)			
	$d = 689$ m Noise region		$d = 688$ m Sensing region	
	$C_{1,2}$	$C_{3,4}$	$C_{1,2}$	$C_{3,4}$
$T_1 \Longrightarrow \Longrightarrow$	1547	1542	**444**	1492
$T_2 \Longrightarrow \Longleftarrow$	1547	1542	1527	1527
$T_3 \Longleftarrow \Longrightarrow$	1547	1542	**1123**	**1123**

tocols). The Maximum Transfer Unit (MTU) packet size (1460 bytes) was used to send CBR traffic. We varied d and the directions of $C_{1,2}$ and $C_{3,4}$. By varying the distance d, achieved throughput is measured for each combination of communication directions. The routes between the nodes were established using the AODV routing protocol at the start of the simulation, and these were preserved for the rest of the simulation to avoid any impact of AODV's behavior on these simulations.

In the first set of simulations, the distance d is set at 689 m, such that N_2 and N_3 are outside the sensing range of each other's transmissions. As expected, both connections achieve about 1550 Kbps, nearly the maximum throughput possible, for all possible directions of transmissions.

Throughput in the Sensing region: In the second set of simulations, the distance d is reduced by 1 m, to 688 m. Now, N_2's transmission can cause N_3 to go into the sensing state, and vice versa. Results of these simulations are shown in Table 3.3.

Nodes in the Communication region will have both physical sense and virtual sense, indicating a busy channel. In the Sensing region, nodes detect the physical carrier sense but not the virtual carrier sense since the affected nodes are beyond the communication distance. As a result, nodes in the Sensing region detect a

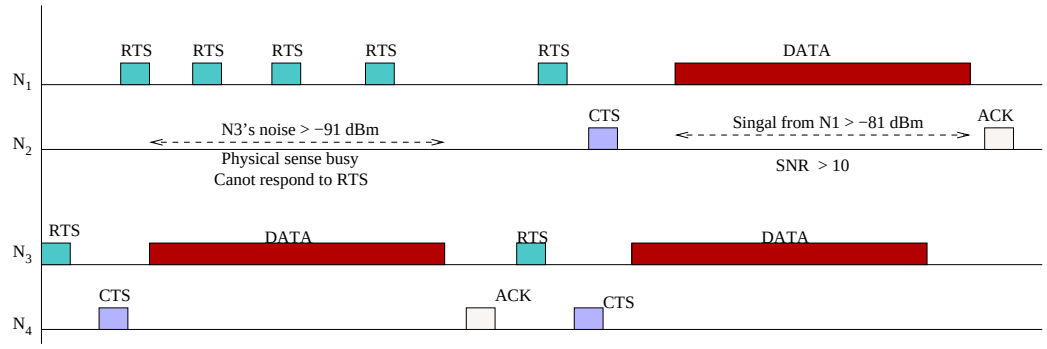

Figure 3.5: Example of successful transmission between N_1 and N_2 in topology T_1, when d = 688 m. N_1 transmits RTSs until an RTS falls between the transmission gap of N_3. After a successful transmission of data, the two communications remain synchronized for few additional data transfer on $C_{1,2}$.

busy medium and do not know when they can initiate new transmissions.

Consider topology T_1: $C_{1,2}$ achieves only 29% of the throughput it achieved when $d = 689$ m, but $C_{3,4}$ is nearly unaffected. The reason for the divergence in behavior is as follows: N_2 is the receiver of the data in $C_{1,2}$, and N_3 is the source of the data in $C_{3,4}$. Since N_2 and N_3 are within each other's sensing range, when N_3 is transmitting, N_2 will be in the sensing state and is prevented from sending an RTS or a CTS. In response to an RTS from the node N_1, N_2 can transmit a CTS only when N_3 is idle or receiving a frame (an ACK or a CTS from node N_4). Since N_3 is transmitting data to N_4, it spends a significant portion of the time in the transmitting mode and a very small amount of the time in the receiving mode. There is no idling since the CBR communication has an infinite data backlog. Furthermore, a CTS that cannot be sent immediately upon receiving an RTS is dropped, and it is not queued for later transmission. The protocol is designed so that the source N_1 will time out, backoff, and retransmit an RTS. As a consequence, there is a significant loss of throughput for the $C_{1,2}$ connection. On the other hand, N_2 spends only a small proportion of time transmitting (a CTS and an ACK) and most of the time receiving DATA. Therefore, N_3 will spend only a small portion of its time in the sensing state due to transmissions by N_2. As a result, it is very unlikely that $C_{1,2}$'s transmissions could adversely affect those of $C_{3,4}$.

For a $C_{1,2}$ transmission to succeed, N_1 must transmit an RTS when N_3 is not transmitting. A possible instance of successful transmission between N_1 and N_2 is shown in Figure 3.5. Since N_1 is not aware of the channel state of N_2, N_1 will continue to send RTSs back to back (with exponential backoff). When an RTS

transmission falls between a gap in N_3's transmissions, N_2 transmits a CTS. When N_1 receives the CTS from N_2, it assumes that the channel is reserved and transmits DATA. While receiving DATA, if N_2's SNR is maintained above the SNR-THRESHOLD, DATA will be received successfully. If the DATA is received, then N_2 will transmit the ACK without checking the state of the physical sense, since the channel is assumed to be reserved. Further evaluation of simulation data shows that successful transmission of a data segment in $C_{1,2}$ is followed by one to four additional successful transmissions, since the two communications remain synchronized for a while. When the synchronization is broken (due to random backoffs etc.), N_1 will have to probe N_2 with RTSs until another transmission gap is discovered.

In the T_2 topology, node N_2 and node N_3 are both receivers, so they interfere with each other only when sending ACK or CTS frames. The time spent in transmitting a CTS or an ACK is significantly small when compared to time spent in transmitting a DATA frame. Therefore, N_2 and N_3 spend very little time in the sensing mode. Since N_2 and N_3 are able to transmit CTSs and ACKs without the interference, both communications are able to achieve the full bandwidth

In the T_3 topology, N_2 and N_3 spend most of their time transmitting DATA and RTS frames since they are the sources. To be successful, N_2 and N_3 must initiate the transmission of RTSs when the other node is not transmitting frames. Since both have an equal probability of competing for the channel, each communication is expected to achieve 50% of the offered load, but the simulation results show that each communication was able to achieve 75% of the offered load. The higher throughput is a result of the synchronized transmission between nodes N_2 and N_3. Figure 3.6 gives an illustration of how a transmission is synchronized between the two sets of nodes to achieve 1.12 Mb/s of throughput for each connection. In Figure 3.6, node N_3 is able to transmit RTSs between the transmission gaps of N_2. When a transmission is complete, a few additional transmissions will follow. During this period, both $C_{1,2}$ and $C_{3,4}$ are able to receive the full bandwidth. We found that such a synchronized transmission continues for four to six data deliveries, but eventually one of the communications will fail. Both communications have an equal chance of failing, and the failing node needs to find an appropriate gap in the other node's transmissions. During this period, the winning node is able to achieve the full throughput, and the failing node does not receive any throughput.

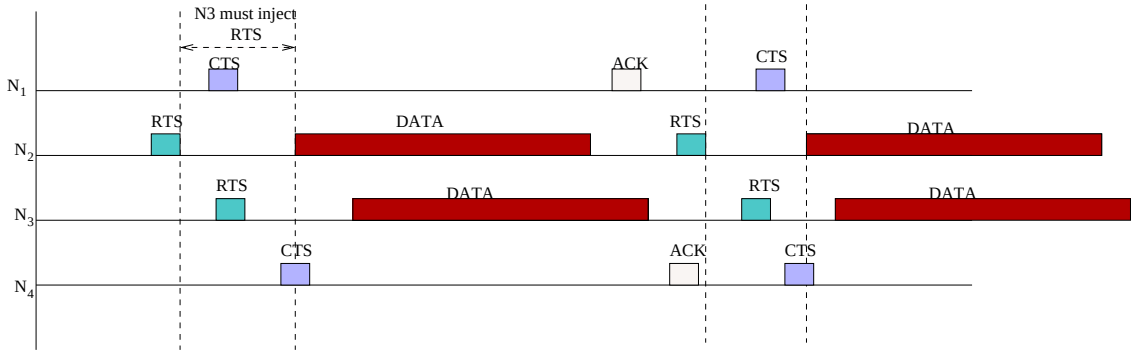

Figure 3.6: Example of synchronized transmission in topology T_3 ($d = 688$ m).

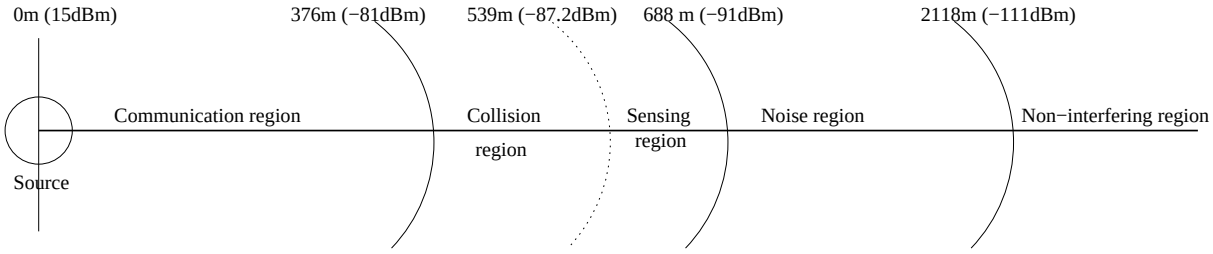

Figure 3.7: Different regions in the propagation of a transmission. Collision region is reached when distance between the communicating nodes is 300 m. Collision distance varies with communicating node's distance.

On average, both nodes are able to synchronize and send 5 back-to-back DATA frames. Once the synchronization is broken, it takes the failing node an average of 2.5 DATA transmissions to find a new gap in the transmission and synchronize with the other transmission. Therefore, a node, on average, sends 7.5 DATA frames and searches for the transmission gap for the transmission time of 2.5 DATA frames, approximately. As a result, T_3 achieves 75% of the offered load, and 25% of the loss in throughput is due to the search for transmission gaps. Reducing the distance from $d = 688$ m to $d = 540$ m does not change the throughput.

Throughput in the Collision region: A significant drop in throughput is seen at $d = 539$ m for all topologies. Throughputs for $d = 540$ m and $d = 539$ m are shown in Table 3.4. Throughput losses are primarily due to the collision of received signals at N_2 and N_3. We identify 377 to 539 m as the Collision region with respect to a single interfering distant communication. This range is dependent on transmission power, ambient noise level, the distance between the communicating nodes, and the distance between the communications. Figure 3.7 shows an updated version of the Communication and Noise regions given in

Table 3.4: Achieved CBR throughputs for the 4-node network with $d = 540$ and 539 meters. The numbers in bold indicate significant reduction in throughputs when N_1 and N_3 are within collision distance.

Topology	Achieved throughput (Kbps)			
	$d = 540$ m Collision/Sensing region		$d = 539$ m Sensing region	
	$C_{1,2}$	$C_{3,4}$	$C_{1,2}$	$C_{3,4}$
$T_1 \Longrightarrow \Longrightarrow$	442	1537	**0.003**	1500
$T_2 \Longrightarrow \Longleftarrow$	1538	1542	**329**	**329**
$T_3 \Longleftarrow \Longrightarrow$	1122	1123	**738**	**753**

Figure 3.3. At this distance, N_2's receive signal cannot maintain the necessary SNR (the SNR falls below 10 dB, due to increased noise from N_3's transmission) to receive the DATA frames. Derivation of 539 m as the upper-end of Collision region is shown in Equation 3.10.

In topology T_1, N_2's reception of RTSs and DATA collides with the noise of node N_3's transmission of DATA and RTSs. Similarly, N_3's reception of an ACK and a CTS collides with the noise from N_2's transmission of a CTS and an ACK. Since the DATA transmission duration is longer than that of the CTS or the ACK, the throughput achieved by $C_{1,2}$ is nearly zero, while $C_{3,4}$ is able to achieve 96% of the throughput it achieved at $d = 540$ m.

Consider the T_2 topology. At $d = 539$ m, N_1's transmission can affect N_3's reception of the RTS and DATA by making the SNR at N_3 fall below RADIO-RX-SNR-TRESHOLD. To be successful, N_1's RTS and DATA transmissions to N_2 must be uninterrupted by N_4's RTS or DATA transmissions. Thus, each communication achieves only 20% of the offered load.

Collisions in topology T_3 are similar to those in topology T_2. In topology T_3, the CTSs and ACKs are lost due to collision. Since both $C_{1,2}$ and $C_{3,4}$ have an equal chance of colliding, both communications receive equal throughput. The probability and the cost of DATA collisions is higher than those of CTS or ACK collisions. Therefore, T_3 receives higher throughput than T_2.

The maximum collision distance can be calculated using the RADIO-RX-SNR-THRESHOLD and the background noise (-100.97 dBm in Glomosim).

Using the two-ray path model, we can compute the receiving signal power when the source and the destination are 300 m apart. As shown in Figure 3.2, the signal strength received from a transmitter that is

300 m away is -77.041 dBm.

$$\frac{\text{Receive Signal Power}}{\text{Total Noise}} \quad < \quad RADIO-RX-SNR-THRESHOLD$$

For node N_2:

$$\frac{N_{2,}\text{Receive Signal Power}}{N_{2,}\text{Total Noise}} \quad < \quad RADIO-RX-SNR-THRESHOLD$$

$$\frac{N_{2,}\text{Receive Signal Power}}{\text{Background Noise} + N_{2,}\text{Noise}} \quad < \quad RADIO-RX-SNR-THRESHOLD$$

Substituting the known values, we get

$$\frac{-77.041dBm}{(-100.97dBm + N_{2,noise})} \quad < \quad 10dB.$$

Using (3.6), we determine the maximum tolerable $N_{2,noise}$ level from other communications.

$$\begin{aligned}
\frac{1.976 \times 10^{-8}\ mw}{(7.99 \times 10^{-11}\ mw + N_{2,noise})} &< 10 \\
1.976 \times 10^{-8}\ mw &< 10 \times (7.99 \times 10^{-11} + N_{2,noise})\ mw \\
(1.976 \times 10^{-8} - 7.99 \times 10^{-10})\ mw &< 10 \times N_{2,noise} \\
\frac{(1.897 \times 10^{-8})}{10}\ mw &< N_{2,noise} \\
N_{2,noise} &> 1.897 \times 10^{-9}\ mw
\end{aligned}$$

Using (3.5), this can be converted back to dBm.

$$N_{2,noise} \quad > \quad -87.22\ dBm \tag{3.10}$$

In our experiment, N_3 is the only noise source on N_2. Therefore, in Figure 3.2, we determine that N_3's transmission will create a noise level of -87.2 dBm, or higher, for distances of 539 m or less.

Table 3.5: Achieved CBR throughputs for the 4-node network with $d = 539$ for unequal distances for communication node pairs. N_1 and N_2 are place 300 m apart, and N_2 and N_4 are placed 290 m apart. Numbers in bold indicate significant change in throughput.

Topology	Achieved throughput (Kbps)			
	N_1N_2,N_3N_4=300		N_1N_2=300, N_3N_4=290	
	$C_{1,2}$	$C_{3,4}$	$C_{1,2}$	$C_{3,4}$
$T_1 \Longrightarrow \Longrightarrow$	0.003	1500	0.5	1572
$T_2 \Longrightarrow \Longleftarrow$	329	329	**2**	**1572**
$T_3 \Longleftarrow \Longrightarrow$	738	753	**7**	**1560**
$T_4 \Longleftarrow \Longleftarrow$	1500	0.003	**1278**	**477**

It must be noted that the results provided in Table 3.4 can change if the distance between N_1-N_2 and N_3-N_4 is changed. If the communicating nodes are moved closer, for example, N_1 and N_2, and N_3 and N_4 are placed 290 m apart, then the collision distance for N_2 is reduced to 520 m. With the reduced distance, N_2 receives a stronger signal from N_1. With the increased received signal power, N_2 is now able to withstand a higher noise level from N_3's transmissions. Therefore, N_3 must be closer in order to cause N_2's reception to fail due to a low SNR. A similar observation can be made on N_3's SNR.

If the two pairs are placed at unequal distances, throughputs may vary significantly. For example, if N_3,N_4 are placed 290 m apart and N_1,N_2 are placed 300 m apart, and d = 539 m, $C_{3,4}$ will outperform $C_{1,2}$ in all cases, with the exception of T_4. Since N_3 and N_4 are placed closer than N_1 and N_2, N_3's SNR is higher than N_2's SNR. We have shown above that N_2's SNR falls below 10 dB when the communicating nodes are 300 m apart and the distance between the communications is 539 m. As a result, N_2 suffers collisions due to N_3's transmission noise. Since N_3 is able to receive at a slightly higher SNR than N_2, node N_3 is able to maintain a safe SNR value. Therefore, N_3's receptions are not lost due to N_2's transmissions. As a result, the $C_{1,2}$ communication receives almost zero throughput, and $C_{3,4}$ receives the full offered load. $C_{1,2}$ performs better than $C_{3,4}$ in T_4 because N_2 is able to put N_3 in sensing mode and prevent it from responding to N_4's RTSs with CTSs. Therefore, N_3 transmits infrequently and does not impact N_2 as much as N_2 impacts N_3.

Figure 3.8: Collision distance experiment. N_1 is the source and N_2 is the destination of the communication. N_3 is a distant node whose transmission increases the noise level at N_2.

3.4 Collision Distance

We define collision distance as the minimum distance between the receiving node and the distant transmission source, such that the distant communication does not collide with the receiving signal. To illustrate the collision distance, the 3-node setup shown in Figure 3.8 is used. N_1 and N_2 are the communication nodes; N_1 is the source, and N_2 is the destination. N_3 is a distant node whose transmission increases the noise level at N_2.

For node N_2 to receive node N_1's transmission while node N_3 is transmitting, the signal-to-noise ratio at node N_2 must be greater than the SNR-THRESHOLD (10 dB is used in most simulations and off-the-shelf hardware).

For N_2 to successfully receive,

$$\begin{aligned} SNR &> \frac{\text{Incoming Signal from } N_1}{\text{Total } N_2 \text{ Noise}} \\ &= \frac{N_1 \text{ Signal strength}}{N_3 \text{ Signal strength+Background noise}}, \text{ since } N_2 \text{ has no other noise sources.} \end{aligned} \tag{3.11}$$

If node's background noise is ignored (Appendix A shows the calculation of collision distance with the node's background noise), we can rewritte (3.11) as follows.

$$SNR > \frac{N_1 \text{ Signal strength}}{N_3 \text{ Signal strength}} \tag{3.12}$$

since N_3 is outside of N_2's communication range, N_3's signal strength is $P_{t_{N_3}} \left(\frac{2.25}{D^2}\right)^2$. When d is between 0 m and 226 m, N_1's signal at N_2 is $P_{t_{N_1}} \left(\frac{0.125}{4\pi d}\right)^2$.

Therefore, (3.12) can be rewritten as follows.

$$10 > \frac{P_{t_{N_1}} \left(\frac{0.125}{4\pi d}\right)^2}{P_{t_{N_3}} \left(\frac{2.25}{D^2}\right)^2}$$

Assuming that N_1 and N_3 transmit at same power level, we have the following

$$\begin{aligned} 10 &> \frac{\left(\frac{0.125}{4\pi d}\right)^2}{\left(\frac{2.25}{D^2}\right)^2} \\ &= \left(\frac{\frac{0.125}{4\pi d}}{\frac{2.25}{D^2}}\right)^2 \\ \sqrt{10} &> \frac{0.125 D^2}{2.25 \times 4\pi d} \\ 715d &> D^2 \\ D &< \sqrt{715d} \quad d \leq 226m \end{aligned} \tag{3.13}$$

When d is between 227 m and 376 m path loss is determined using (3.3), $P_{t_{N_1}} \left(\frac{2.25}{d^2}\right)^2$.

For $d >$ 226 m (3.12) can be manipulated as follows, assuming $P_{t_{N_1}} = P_{t_{N_3}}$.

$$\begin{aligned} 10 &> \frac{P_{t_{N_1}} \left(\frac{2.25}{d^2}\right)^2}{P_{t_{N_3}} \left(\frac{2.25}{D^2}\right)^2} \\ &= \left(\frac{\frac{2.25}{d^2}}{\frac{2.25}{D^2}}\right)^2 \\ &= \left(\frac{D}{d}\right)^4 \\ \sqrt[4]{10} &> \frac{D}{d} \\ D &< \sqrt[4]{10}\, d, \quad d \geq 227m \end{aligned} \tag{3.14}$$

Distance D is computed using 3.13 and 3.14 for all possible values of d and plotted in Figure 3.9. The X-axis represents the distance between two communicating nodes. The Y-axis represents the minimum

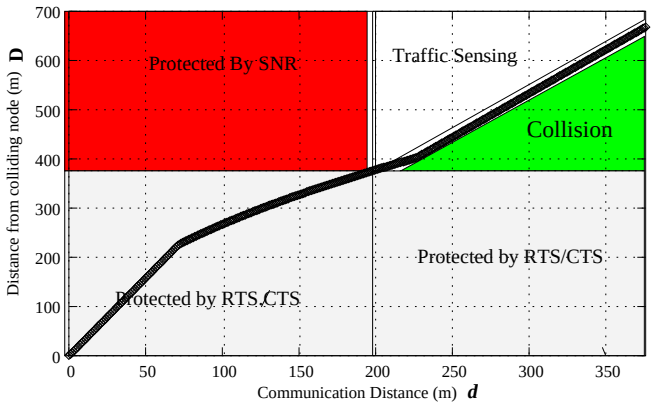

Figure 3.9: Collision Distance: In Figure 3.8, collision distance D (Y-axis) for a specified communication distance d (X-axis).

distance between a receiving node and an interfering node for the transmission to be received successfully. In Figure 3.9, for a D $\leq$ 376 m, a communication receives RTS/CTS protection (the distance communication is within the RTS/CTS range of the destination node). Therefore, in the absence of other communications, collision distances less than 376 m do not exist. The corresponding value of 198 m for the communication represents a safe distance for two nodes to communicate without the possibility of collision (from a single noise source).

3.5 Modification to the 802.11 Protocol

The throughput drops off in topology T_1 as d is reduced from 689 m to 688 m because node N_2 is close enough to be put in the sensing mode by the transmissions of node N_3 to node N_4. Since a node can transmit an RTS or a CTS only when it is not in a sensing state (as specified in the IEEE 802.11 standard [20]), N_2 is prevented from sending a CTS response to N_1's RTS transmission. The simulation data verifies that N_2 receives RTS transmissions from N_1 reliably, but N_2 is unable to send CTS due to the increased noise level caused by the competing transmissions from N_3.

Our observation is that, if a node can receive an RTS, then it is able to receive the DATA from the same source. Furthermore, if the overall noise level is low (≤ -81 dBm in the Glomosim implementation), then sending a CTS is not likely to disrupt any other transmissions. Therefore, N_2 not responding to a CTS from N_1 reduces the network throughput without any notable benefit.

Table 3.6: Example 2:Throughput of the four-node network in Figure 3.4 with the with the proposed modification, d=688 m, node distance is 300 m. Numbers in bold represent significant change in throughput.

Topology	Achieved TCP throughput (Kbps)			
	802.11 MAC		Modified MAC	
	$C_{1,2}$	$C_{3,4}$	$C_{1,2}$	$C_{3,4}$
$T_1 \Longrightarrow \Longrightarrow$	444	1492	**1584**	1576
$T_2 \Longrightarrow \Longleftarrow$	1527	1527	1584	1579
$T_3 \Longleftarrow \Longrightarrow$	1123	1123	1150	1150

To eliminate this unnecessary loss of throughput, we modified the 802.11 MAC protocol so that when a node receives an RTS, it will respond with a CTS packet transmission if the virtual sense is idle and the noise level is within the sensing range (-91 dBm to -81 dBm in Glomosim).

The code fragment in Figure 3.10 shows the changes made to the Mac802_11ReceivePacketFromRadio function in the Glomosim 802_11.pc file. Changes made are coded with a #define flag PROPOSED_802_11.

3.6 Performance Analysis

We evaluated the performance of the proposed modification for the 4-node static network previously described, a longer static chain, a static grid, and a 100-node MANET. Simulations include the CBR traffic for static networks and the CBR and the TCP traffic for MANETs with different node densities and mobilities.

3.6.1 Four-node network

First, we reran the 4-node (Figure 3.4) simulations with $d = 688$ m. Instead of the standard 802.11 MAC protocol, we used the modified MAC protocol, which uses a higher noise level to determine when to send CTS packets from a node that has received RTS packets. All of the other aspects of the standard 802.11 MAC protocol are unchanged. With this modification, both connections now achieve about 1575 Kbit/s throughput for the T_1 topology. This is to be expected, however, since the modification is designed to work well for the situations indicated by T_1. The two communications in the T_1 topology show a significant benefit using the modified MAC protocol.

The modified MAC protocol enhances the performance of topology T_1 when competing nodes are in

```
void Mac802_11ReceivePacketFromRadio(
   GlomoNode* node,
   GlomoMac802_11* M802,
   Message* msg)
{
   M802_11ShortControlFrame* hdr = (M802_11ShortControlFrame*)msg->packet;
   M802->IsInExtendedIfsMode = FALSE;
. . .

      assert(!Mac802_11IsTransmittingState(M802->state));
   if (hdr->destAddr == node->nodeAddr)
   {
      switch (hdr->frameType) {
      case M802_11_CTS:
              . . .

      case M802_11_RTS:
```

```
#ifndef PROPOSED_802_11
            if (Mac802_11IsWaitingForResponseState(M802->state)){
            } else if ((!Mac802_11WaitForNAV(node, M802)) &&
                  (RadioStatus(node, M802) == RADIO_IDLE))


#else
            if (Mac802_11IsWaitingForResponseState(M802->state)){
            } else if ((!Mac802_11WaitForNAV(node, M802)) &&
                  (CurrentNoiseLevel < -81))
#endif
```

```
         {
            // Transmit CTS only if NAV (software carrier sense)
            // and the radio says the channel is idle.
              . . .
             Mac802_11TransmitCTSFrame(node, M802, msg);
         } else {
            if (RadioStatus(node, M802) != RADIO_IDLE) {
               M802->rtsPacketsIgnoredDueToBusyChannel++;
            } else {
               assert(Mac802_11WaitForNAV(node, M802));
               M802->rtsPacketsIgnoredDueToNAV++;
            }//if//
         }//if//
         GLOMO_MsgFree(node, msg);
         break;
          . . .
   }//Mac802_11ReceivePacketFromRadio//
```

Figure 3.10: Proposed modification to 802.11 MAC protocol logic. Changes made to Glomosim implementation in 802_11.pc file is indicated by a rectangular box.

Table 3.7: TCP throughputs for the 4-node network with $d = 688$ m.

Topology	Achieved TCP throughput (Kbps)			
	802.11 MAC		Modified MAC	
	$C_{1,2}$	$C_{3,4}$	$C_{1,2}$	$C_{3,4}$
$T_1 \Longrightarrow \Longrightarrow$	36	1259	**1447**	1234
$T_2 \Longrightarrow \Longleftarrow$	1224	1223	1253	1258
$T_3 \Longleftarrow \Longrightarrow$	678	669	**947**	**945**

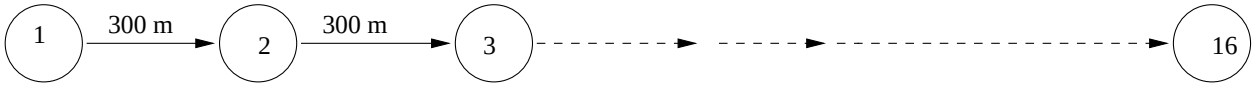

Figure 3.11: 16 node linear chain simulation setup. Nodes are placed 300 m apart.

the sensing range of one another. As indicated in Section 3.3, the sensing range in these experiments is 540 m to 688 m. We compared the modified protocol with the standard 802.11 MAC protocol for d= 689, 688, 540, and 539 meters. In all cases, the modified MAC protocol performed as well as or better than the original 802.11 MAC protocol. In some instances (for example, in topology T_1 with d=540 to 688 m), the modified MAC's performances showed a significant improvement. These simulations were run with the AODV routing protocol turned off, to eliminate its impact. To determine the impact of the routing protocol, we ran additional simulations with the AODV routing protocol for d = 539, 540, 688, and 689 m. The results for both versions of the MAC protocols do not show a significant performance gain or a significant performance loss when using AODV as the routing protocol. Only for the T_1 topology, $C_{1,2}$ communication lost 18% of throughput when using the AODV protocol. In all other cases, the performance loss due to AODV is less than 3%.

To evaluate the performance of the modification for TCP traffic, we replaced the CBR connections with FTP connections. Again, there are four different topologies, with T_1 and T_4 being symmetrical. Since TCP communications are bi-directional, each node is a source and a destination, regardless of the communication direction. If a node is a source of TCP data, then it is also a destination of TCP ACKs. The TCP throughputs for the original and the modified MAC protocols are shown in Table 3.7 (with the AODV routing protocol) for $d = 688$ m. As in the case of CBR traffic, the propsed modification has no effect on performance for $d \leq 539$m.

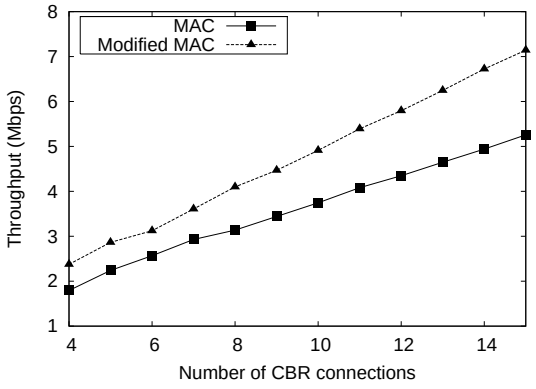

Figure 3.12: Cumulative CBR throughput improvement for static linear chain.

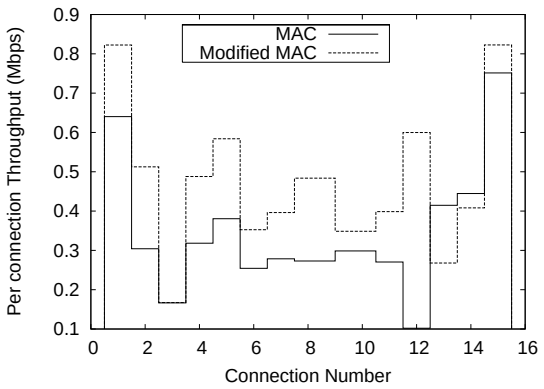

Figure 3.13: Per connection throughput in a 16-node linear chain.

3.6.2 Static linear chain

In this section, we analyze the benefits of the proposed MAC protocol modification using a linear chain of 16 nodes. The distance between the adjacent nodes is 300 m (see Figure 3.11). The nodes are stationary. Each node, except the right-most node, has a communication with the neighbor to its right. The routing protocol is turned off to minimize the impact of the routing algorithm on the throughput. Each CBR connection generates 0.8 Mbps (one half of the full bandwidth) using 1460 byte packets. Chain lengths varying from 5 to 16 nodes were simulated. The cumulative throughput of these CBR communications is plotted in Figure 3.12. The proposed modification of the 802.11 MAC protocol improves overall throughput by 35%. Figure 3.13 shows the achieved throughput for individual connections in the 16-node chain (15 CBR connections) simulation.

Per link throughput for the 15 CBR communications indicates that the first and the last nodes achieve higher throughput than the center CBR connections, since they have less noise and less competition for bandwidth than the center nodes do. In the 13^{th} and 14^{th} CBR connections, the proposed modification performs poorly in comparison to the original MAC protocol, but in all other connections, the proposed modification improves the performance.

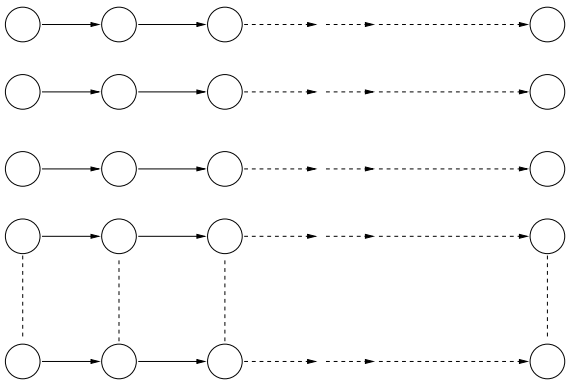

Figure 3.14: Grid topology.

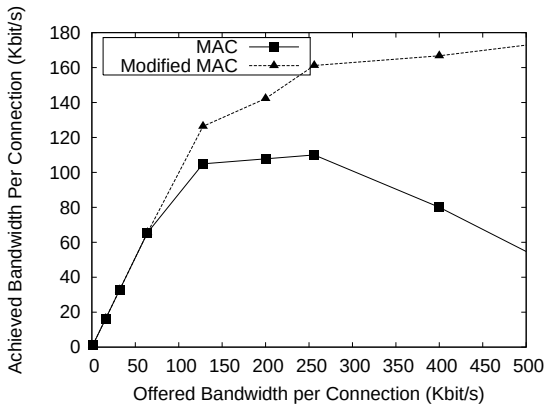

Figure 3.15: Throughput improvement in grid topology.

3.6.3 Static grid

Next we considered a 10×10 grid of nodes, with 300 m between adjacent nodes (see Figure 3.14). The nodes are stationary, and each node has a connection to send data to its right neighbor, if it exists. There are a total of 90 connections. The nodes in the left column are senders only, and the nodes in the right column are receivers only. Once again, we used AODV for the initial setup of routes and turned it off to minimize its impact on performance. Since the network is static, there is no need for the routing algorithm after the routes are determined once. The proposed modification improves the throughput by 50% at the load of 250 Kbps per connection (see Figure 3.15). More importantly, the proposed modification sustains the throughput as the network is overloaded, while the original 802.11 MAC protocol degrades the performance significantly.

3.6.4 Mobile ad hoc network

Next, we conducted simulations of MANETs. The original version of the 802.11 MAC protocol was compared to the modified version. The baseline MANET is used with Fifty CBR connections. The achieved throughput is illustrated in Figure 3.16. The proposed modification increased the peak throughput by 15%, and resulted in performance gains of up to 33% for different network loads. In addition, the proposed modification reduces the end-to-end delay by 50% (see Figure 3.17), and uses fewer hops to deliver data (see Figure 3.18). Also, we show in Figure 3.19 that our proposed modification improves the fairness of the network for loads beyond saturation.

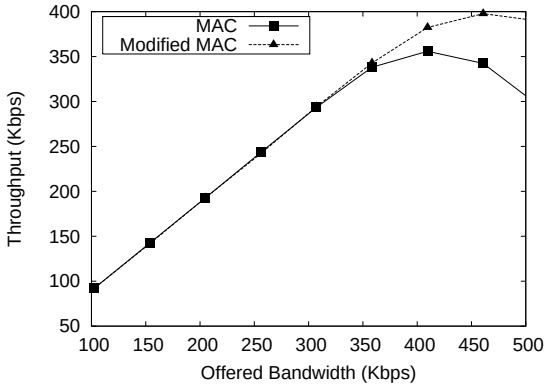

Figure 3.16: Throughput achieved for CBR traffic in a MANET with the 802.11 and modified MAC protocols using AODV routing protocol.

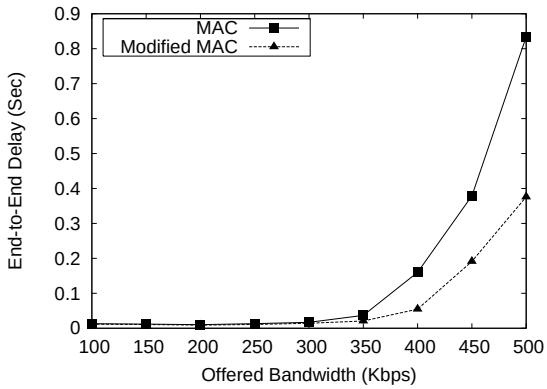

Figure 3.17: CBR data end-to-end delay in a MANET with AODV.

The proposed modification increases the reliability of the link by reducing the false route breaks. A false route break is a route break that is caused by false link failures. False link failures are link failures that are caused by the nodes that are within the communication range but do not respond. False link failures could be due to a NAV indicating that the channel is busy, or they could be due to an increased noise level at the destination. The modified MAC protocol addresses the latter type of false route breaks. This reduces the need for the routing protocol to unnecessarily find routes, which reduces the control overhead. We found that the number of control packets was reduced by as much as 37% for high loads with the proposed modification (see Figure 3.20).

Frequent route breaks increase the end-to-end delay, as data need to be queued until the route is repaired. By reducing the number of route breaks, the proposed modification is able to achieve lower end-to-end delays. Since the AODV routing protocol gives higher priority to routing packets (RREQ, RREP and RERR) than to data packets, reduced routing overhead in the modified MAC protocol results in shorter control queues. Shorter control queues result in shorter data queues. This, in turn, allows the modified MAC protocol's data to achieve a lower end-to-end delay. Improved reliability in the MAC protocols increases the lifetime of the route. In saturation, AODV routing protocols initially find a non-optimal route, and later update that route with a shorter route; therefore, the more route breaks occur, the more data take non-optimal routes. Therefore, the proposed MAC protocol is able to achieve a lower average of hop counts than the 802.11 MAC protocol (Figure 3.18).

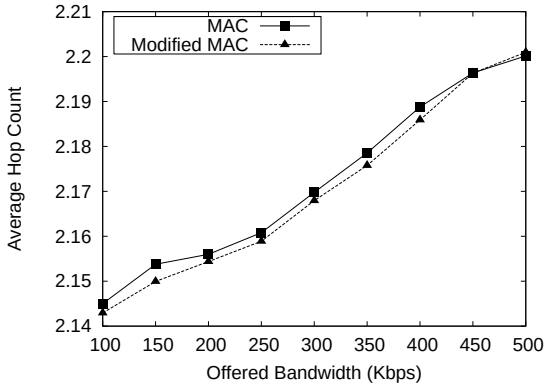

Figure 3.18: Average hop count for data delivered with 802.11 and modified MAC protocols.

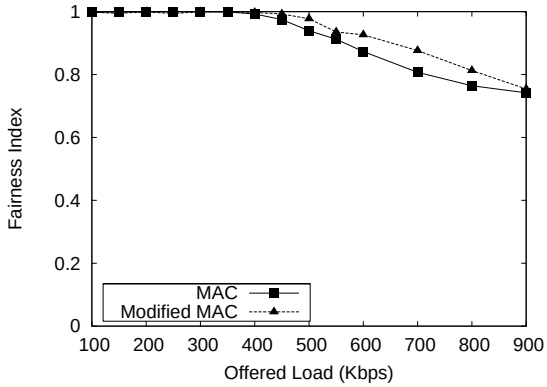

Figure 3.19: Fairness evaluation of the 802.11 and modified MAC protocols.

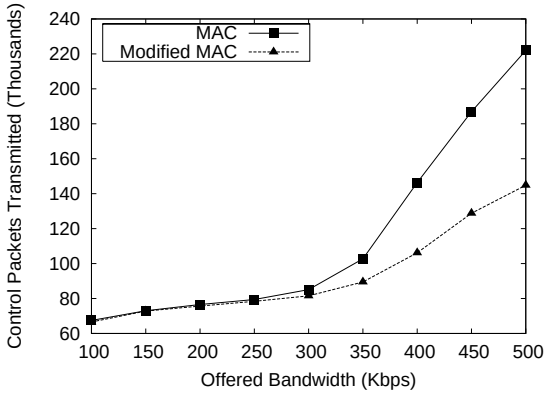

Figure 3.20: RREQ control packets transmitted in the MANET (AODV).

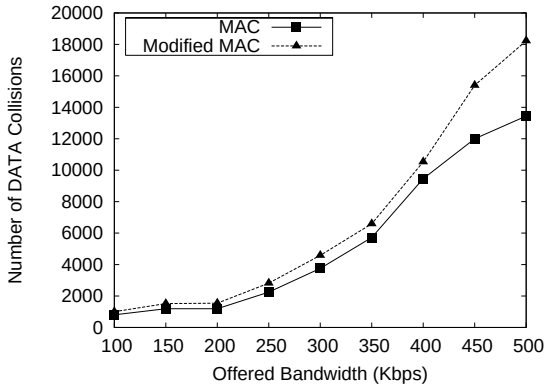

Figure 3.21: Total DATA (unicast) packets lost due to collisions (AODV).

The proposed modification sends CTSs more aggressively than the 802.11 MAC protocol does. Sending additional CTSs increases the network utilization in the MANET; this could increase the interference as well. We analyzed the simulation data to see if the modified MAC protocol would increase the total collisions in the network. At the radio layer, the number of collisions was reduced by 24%. However, at the routing layer, the number of collisions increased. In the radio layer, each colliding packet is counted as a collision, irrespective of the destination. For example, if node A is receiving a frame from node B that is destined for node C and the frame collides at node A, it is counted as a collision at node A, even though the frame is not of any use to node A. On the other hand, the routing layer only counts the collisions of the frames that are destined for the node. These frames can be either unicast or broadcast messages.

The aggressive CTS transmissions contribute to the increase in collisions at the routing layer. Figure

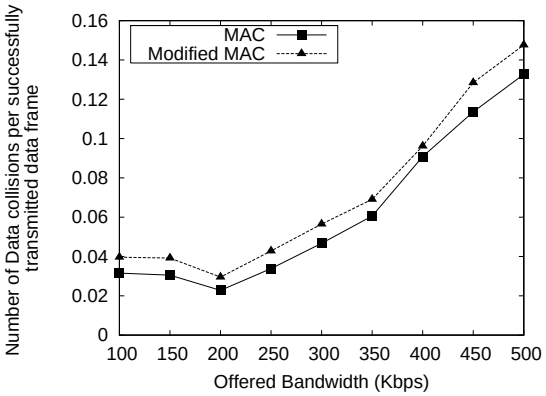

Figure 3.22: Total DATA collisions per successfully transmitted data frame.

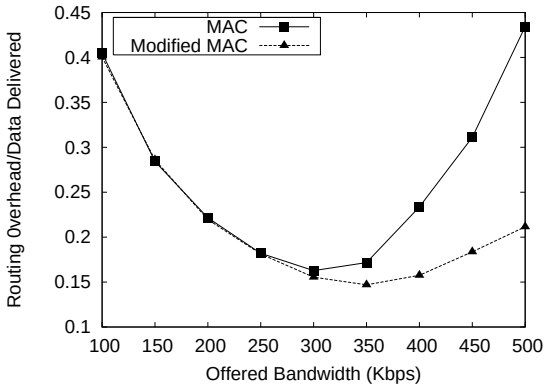

Figure 3.23: Routing overhead per data packet delivered.

3.21 shows a 29% increase in DATA collisions. Despite the sharp increase in DATA collisions, Figure 3.22 shows that the DATA collisions per data frame successfully transmitted has not increased significantly for the proposed MAC modification; there is only a 2% to 3% increase in collision per DATA frame successfully transmitted. The modification significantly reduces the routing overhead in the MANET. Figure 3.23 shows that the modification significantly lowered the amount of control overhead per data packet delivered.

TCP performance

To evaluate the proposed modification using a more realistic network load, the TCP traffic (using FTP) along with the CBR background traffic was simulated. The MANET parameters and other simulation parameters are identical to those used in the CBR simulations. Up to 25 FTP connections were used, along with 0, 100, 200 and 300 Kbps of CBR background traffic. 50 sources and 50 destinations were used to inject the background traffic into the network. The numbers of TCP connections are increased from one through twenty five to evaluate the performance of the proposed modification. In Figures 3.24 to Figure 3.27, we present the achieved TCP throughputs for various CBR background traffic loads. As the CBR background traffic is increased, the proposed modification gives better performance. For 0, 100, 200 and 300 Kbps background traffic, throughput is improved by 8%, 10%, 21% and 44%, respectively.

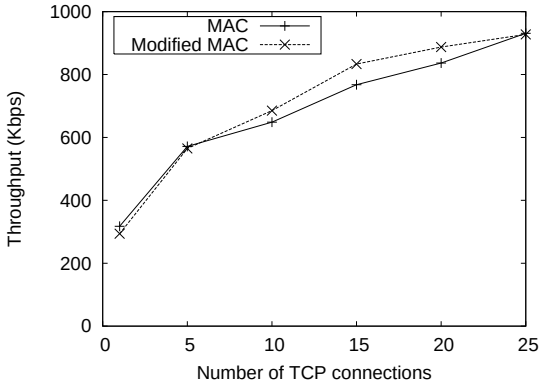

Figure 3.24: TCP throughput achieved in a MANET with the 802.11 and modified MAC protocols without CBR background traffic.

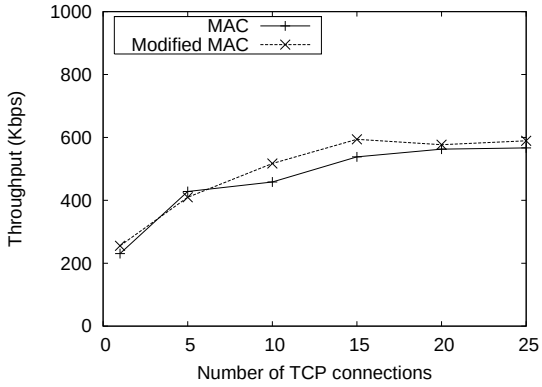

Figure 3.25: TCP throughput achieved in a MANET with the 802.11 and modified MAC protocols with 100 Kbps background traffic.

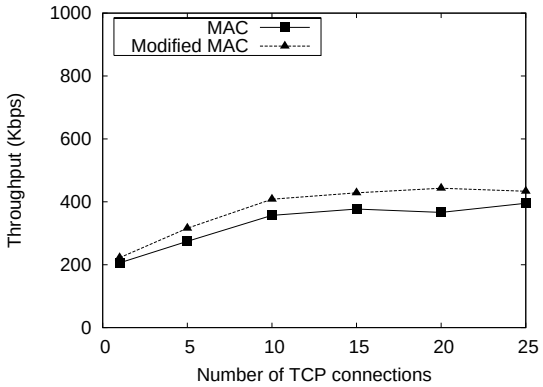

Figure 3.26: TCP throughput achieved in a MANET with the 802.11 and modified MAC protocols with 200 Kbps background traffic.

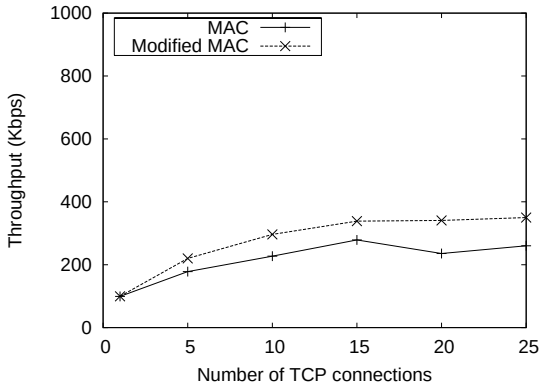

Figure 3.27: TCP throughput achieved in a MANET with the 802.11 and modified MAC protocols with 300 Kbps background traffic.

Other routing protocols

In this section, we show that the proposed modification improves the MANET throughput when DSR and LAR routing protocols are used instead of the AODV routing protocol. All simulation and network parameters are the same as previously described. Figure 3.28 shows the CBR performance using the DSR routing protocol, and Figure 3.29 shows the same for the LAR routing protocol. The performance improvements are comparable to those achieved in the AODV routing protocol.

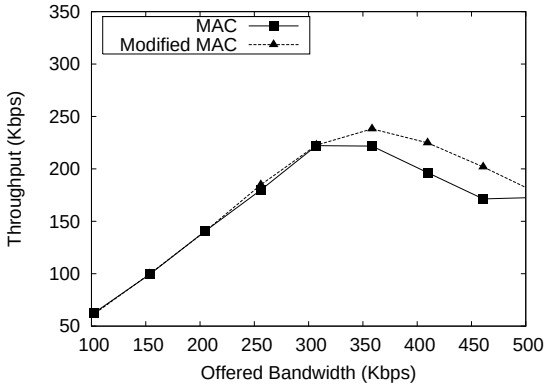

Figure 3.28: CBR throughput using DSR routing protocol.

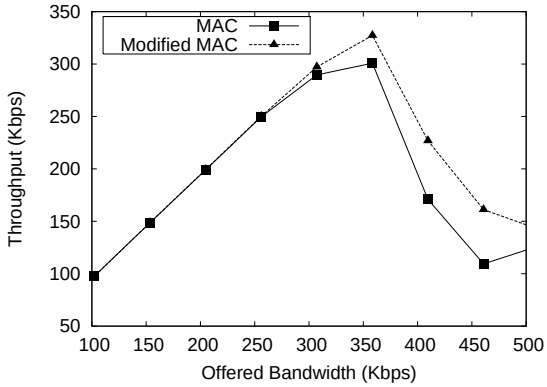

Figure 3.29: CBR throughput using LAR routing protocol.

High and Low density MANET simulation

In this section, we evaluate whether the proposed modification works better in dense or sparse networks. In these simulations, terrain area is changed to increase or decrease the density of the nodes in the MANET; all other parameters are the same as in the previous MANET simulations. The AODV routing protocol is used for a high-density terrain (a terrain size of 1000 $\times$ 1000 m^2). For a low-density MANET, a terrain size of $1600 \times 1600m^2$ is selected. In the previous 1200×1200 terrain simulations, there were 30.8 nodes per node communication area. In the dense and sparse simulations, there are 44 and 17.3 nodes per communication area, respectively.

In the first set of simulations, 50 CBR communications were setup as they were in the previous CBR simulations. Figure 3.30 shows the performance of dense simulation (high-density network), and Figure 3.31 shows the performance of sparse (low-density) networks, using the 802.11 MAC and modified MAC protocols. In both high and low-density MANETs using CBR communications, the proposed modification outperforms the original 802.11 MAC protocol. In the low-density simulations, the peak throughput is improved by 9%, and up to 21% higher throughput is achieved. Similarly, high-density simulation shows a 17% increase in peak throughput, and performance gains of up to 17%.

To evaluate the possible drawbacks and benefits of the proposed modification using TCP, simulations were repeated for a terrain size of $1600 \times 1600m^2$ (sparse simulation; see Figure 3.32) and a terrain size of $1000 \times 1000m^2$ (dense simulation; see Figure 3.33). In these simulations, 200 Kbps of background traffic

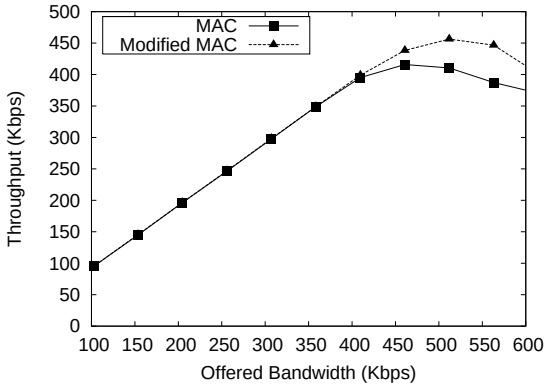

Figure 3.30: CBR throughput for MANET using $1000 \times 1000m^2$ and AODV.

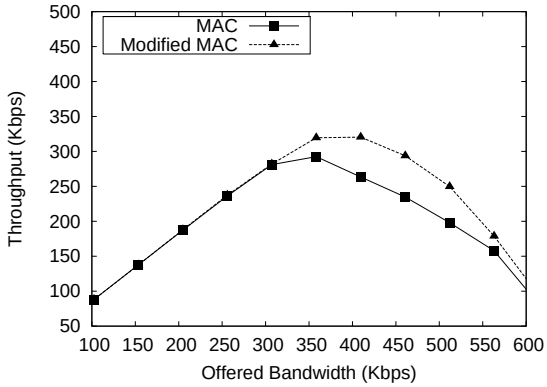

Figure 3.31: CBR throughput for MANET using $1600 \times 1600m^2$ and AODV.

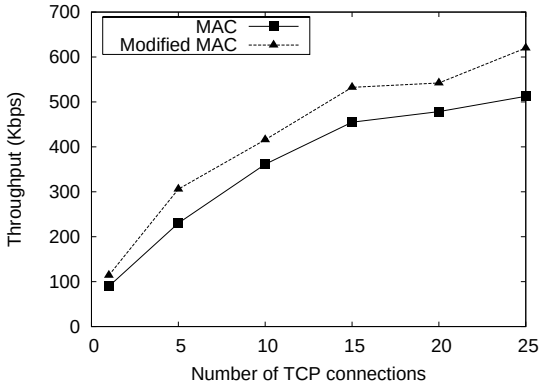

Figure 3.32: TCP throughput for MANET using $1600 \times 1600m^2$.

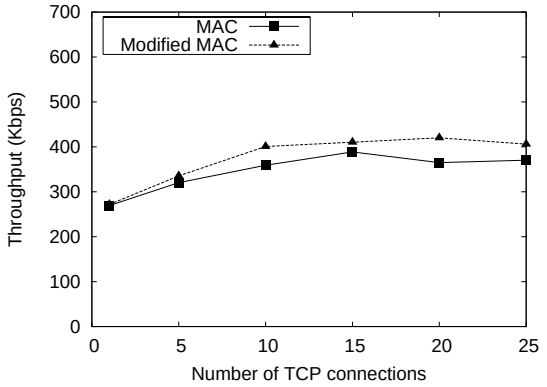

Figure 3.33: TCP throughput for MANET using $1000 \times 1000m^2$.

is used to simulate a more realistic MANET. The results indicate that the proposed modification achieves higher throughputs in both dense and sparse MANETs. Figure 3.32 and Figure 3.33 show that the sparse MANET realized greater performance gains than the dense MANETs did.

High and low mobility MANET simulations

Next, we evaluated the effect of node mobility on the proposed modification; node speed is increased by varying the average node speed between [10,19] m/sec (the earlier simulations used a [1,19] m/sec average speed) and a 0 pause time. In order to evaluate the low-mobility MANET simulation, the average speed of the node was held between 1 and 10 m/sec, and the node pause time was set to 15 seconds. CBR traffic was used in this simulation. Figure 3.34 and Figure 3.35 show the performance of high and low-mobility MANETs, respectively. The modified MAC protocol continues to achieve a performance gain for both high

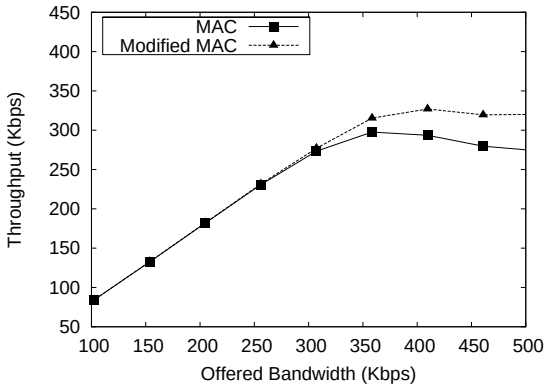

Figure 3.34: CBR throughput for MANET with high-mobility nodes (nodes move [10,19] m/sec).

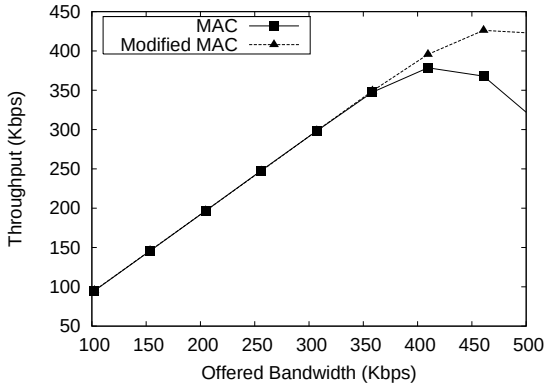

Figure 3.35: CBR throughput for MANET with low-mobility nodes (nodes move [1,10] m/sec with 15 sec. pause time).

and low-mobility MANETs. The low mobility MANET shows a 12% higher peak throughput and up to 15% increases in throughput. Similarly, the high-mobility MANET shows a 9% higher peak throughput and up to 14% increases in performance.

3.7 Related Work

In the literature, there are several proposed solutions to avoid exposed nodes in MANETs [3, 64]. In these studies, authors considered exposed nodes within the communication range, and devised techniques to synchronize transmissions among exposed nodes.

Acharya et al. [3] proposed a MACA-P protocol to address the exposed node problem. They used control gaps between the RTS/CTS to synchronize a transmission that is exposed. Their modification requires significant and complex changes to the 802.11 MAC protocol. Acharya et al. show that they achieve significant performance improvement for static ring topology. Shukla et al. [64] took a similar but simpler approach to mitigate exposed nodes within the communication region. In this study, DATA and ACK are synchronized to achieve parallel transmission between two communicating nodes that are exposed. The authors compared their work with the MACA-P protocol for static networks and showed that their protocol performed better than the MACA-P protocol. In addition to static ring topology, the authors have shown that the proposed modification can improve randomly placed nodes. On the other hand, in our work, we identify exposed nodes beyond the Communication region (exposed nodes in sensing range), and propose a simple

solution to avoid such exposed nodes.

3.8 Conclusions

We have investigated the 802.11 MAC protocol's behavior in the presence of competing but distant communications. We have shown, using a simple 4-node static network, that a connection could be dominated by another, even though neither can interfere with the other's ability to successfully receive radio signals. The problem is due to the overly cautious use of noise levels, in the absence of a virtual carrier sense, to infer the transmission activity of other nodes. We have shown that, by slightly relaxing the constraints, we can improve the overall performance of the network significantly. Our proposed modification allows a node to respond in more instances with a CTS when it receives an RTS from a potential sender. This improves the performance significantly for pathological situations where the sender of a communication causes the receiver of another transmission to go into sensing range without a virtual carrier sense. While these situations are temporary in a mobile ad hoc network, they do occur frequently. For example, a MANET with 100 nodes and 50 CBR connections using the AODV routing protocol, we have shown, using simulations, that the proposed modification improves CBR throughput by up to 33%, while reducing routing packets by nearly a half. In addition, the proposed modification improves the end-to-end delay and hop count. By simulating different MANET node densities and different routing protocols, we show that the proposed modification's benefits extend to various MANETs and routing protocols.

Chapter 4

Sustaining Performance Under Traffic Overload

In this chapter, we investigate the performance of wireless ad hoc networks with traffic loads beyond saturation. Figure 4.1 shows the network throughput achieved by the baseline MANET as the load is increased from 100 Kbps to 900 Kbps. It can be seen that throughput increases almost linearly for offered loads of up to 400 Kbps. For offered loads beyond 450 Kbps, throughput decreases rapidly. At a traffic load of 900 Kbps, the achieved throughput is approximately 180 Kbps, about one half of the peak throughput. Even with the proposed MAC protocol modification, the throughput is not sustained under high traffic loads.

While it is desirable to operate a network at traffic loads below saturation, an ad hoc network should be designed to gracefully degrade its performance under severe loads. As the number of nodes increases, the bandwidth available per user decreases proportionately. The maximum throughput achieved in the above

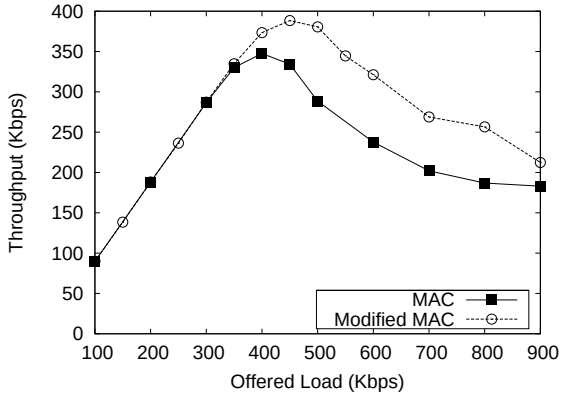

Figure 4.1: Performance of a 100-node ad hoc network with AODV and 802.11 protocols.

example network is 350 Kbps, less than 20% of a single wireless channel bandwidth. Since, there are 25 senders, each sender achieves less than 1% of the channel's bandwidth (BW) on the average. Due to such low throughput per connection, the ad hoc network is likely to . The lack of flow be subject to relatively high traffic loads by the user applications. control at transport layer for UDP data makes it impossible for the network to operate below point of saturation as the load is increased. While the 802.11 MAC protocol [20] is designed to give a node and its neighbors a fair share of the bandwidth, it generally does not work well when the network is saturated. Therefore, the network layer and the routing protocol must ensure that throughput degrades gracefully when the network is overloaded.

4.1 Solutions to Mitigate Underperformance

There have been few studies on the behavior of ad hoc networks operating beyond the point of saturation. Most techniques discussed in the literature attempt to reduce routing overhead in order to reduce congestion and facilitate higher throughput prior to saturation [13, 31, 34, 49]. On-demand routing protocols frequently flood the network with RREQs to repair broken routes. Reducing the number of RREQs flooding the network can significantly reduce the routing overhead in the MANET

An alternative approach to reduce the routing overhead is to reduce the need for route discovery. When the network has reached saturation, the 802.11 MAC protocol causes frequent *false route breaks*. Reducing these false route breaks reduces the need for route discovery. For example, in [62], the number of packet drops are reduced by using RTS validation. Upon receiving an RTS control packet, a MANET node that uses validation assesses the state of the channel when the DATA packet is expected to begin transmission. If the channel is expected to be busy, the node will defer all transmissions; otherwise, the node will respond to an RTS with a CTS. Xu et al. [76] use a different technique to reduce false route breaks. CTS responses are restricted to shorter distances than the normal communication range in order to minimize the collisions caused by hidden nodes. However, we have not come across any specific studies that address ways to make a MANET operate gracefully when traffic loads exceed the point of saturation.

Another approach to reducing the routing overhead is to improve the efficiency of delivery of RREQs (broadcasts). Several efficient broadcast techniques proposed in the literature may be used for this purpose.

These techniques are designed to minimize the number of retransmissions, while attempting to ensure that a broadcast packet is delivered to each node in the network. We can adapt these techniques to reduce RREQs, which are disseminated using network-wide broadcasts. These techniques can be divided into several categories: hierarchical, cluster-based, probability, and network knowledge-based. For example, the Location Aided Routing (LAR) proposed by Ko et al. [42] uses a cluster-based broadcast to reduce the number of RREQs used in route discovery. In other works, probability-based [69], distance-based [56, 45], and network knowledge-based [57, 43] broadcast protocols have been shown to reduce the number of broadcasts in a MANET when using an on-demand routing protocol. Williams et al. [75] provide a detailed comparison of several such protocols. We are particularly interested in probability-based techniques that use neighborhood information. In this chapter, we evaluate the suitability of two such techniques, SBA (Scalable Broadcasting Algorithms) and RAD (Random Assessment Delay), to reduce the control overhead in congested networks.

In this chapter, we analyze the performance of MANETs beyond the saturation point. Using the AODV routing protocol and the 802.11 MAC protocol, we investigate the reasons for the sharp decline in throughput for traffic loads beyond saturation. We show that the route discovery mechanism, used in protocols like AODV, is responsible for bandwidth losses beyond saturation. We investigate several approaches to reduce the RREQ overhead. We examined two broadcast management techniques, RAD and SBA, and found that they provide adequate improvement only when implemented at the MAC layer with a significant MAC to IP interaction. We also propose two different techniques to reduce the number of route requests. The first technique is called “reduced broadcast”, which manages the MAC InterFace Queue (IFQ) to reduce route requests in the MANET. The second technique modifies the frequency at which the RREQs are generated. Using simulations, we show that proposed modifications, when used in conjunction with our earlier modifications to the 802.11 MAC protocol to mitigate false route breaks [24], will enable a MANET to perform gracefully under traffic overloads.

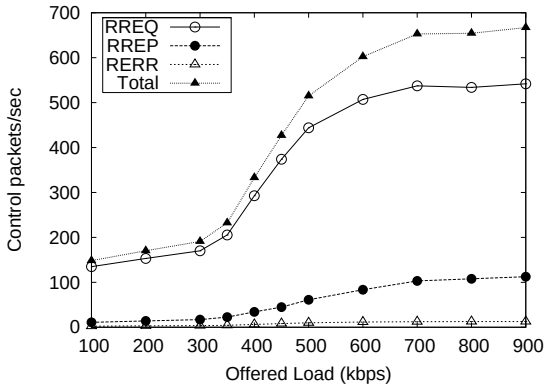

Figure 4.2: Control packets transmitted on wireless links.

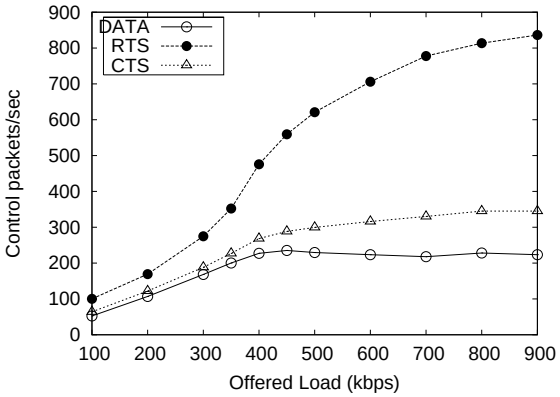

Figure 4.3: Data, RTS and CTS packets transmitted on wireless links.

4.2 Behavior of Mobile Ad Hoc Networks Beyond Saturation

To understand the reasons for the rapid loss of throughput in the example MANET (Figure 4.1), we examined the packets transmitted at the MAC level. We consider only the IEEE 802.11 MAC with the AODV routing. A similar analysis can be made for the proposed MAC. AODV generates three types of control packets: RREQs, RREPs, and RERRs. RREQs and RERRs are broadcasts and each packet sent from the routing layer results in a broadcast at the MAC level. Data packets (denoted DATA) and RREP control packets are sent as unicast packets preceded by the RTS/CTS exchange. Figure 4.2 gives the number of RREQ, RREP, RERR, and the sum of the three transmitted during the simulation for each traffic load. Figure 4.3 gives total DATA, RTS, and CTS packets transmitted during the simulation.

Since the node mobility pattern is unchanged, the increase in load should not cause an excessive increase in the routing protocol overhead. Figure 4.2 indicates that the total routing overhead (the sum of RREQ, RREP and RERR packet transmissions) remains stable up to 300-350 Kbps. Beyond 350 Kbps, however, the routing overhead (mainly RREQs) increases rapidly. Under a high traffic load, the number of instances a node is exposed increases significantly. This, in turn, causes an exposed node not to respond to an RTS with a CTS, which causes the sender of the RTS to falsely conclude, after several retries, that the route is broken. AODV responds to these false route breaks by initiating route discoveries. These extra route discoveries increase the RREQs used to maintain routes in the MANET.

The number of DATA packets transmitted linearly increases up to 400 Kbps and decreases slightly but

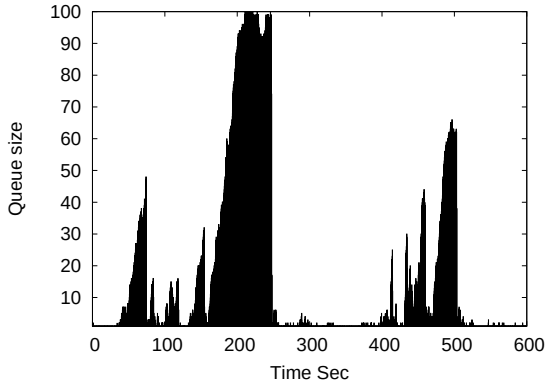

Figure 4.4: Control packet priority queue size for node 64 when offered load is 500 Kbps.

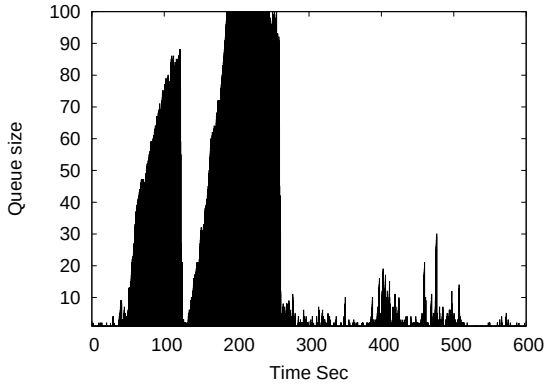

Figure 4.5: Control packet priority queue size for node 64 when offered load is 600 Kbps.

steadily for higher loads, due to busy channels. The drop in throughput for high loads is sharper, due to false route breaks, which increases the probability that data packets will take few hops and be dropped by intermediate nodes. To understand the reasons, we looked at the RTS and CTS packets transmitted by MAC. The number of RTS packets increases sharply with high loads, but CTS packets remain the same. To elicit a CTS for an RTS, the receiving node must receive the RTS (it must overcome the existing noise levels) and must have an idle channel (if the receiving node is exposed to other transmissions, it cannot send a CTS). In fact, the rapid increase in RREQ packets indicates that wireless channels are being clogged by the control packet broadcasts, which increases the ambient noise level, makes the channel busy, and causes distant nodes to go into the sensing mode. In Chapter 3, we have shown that by modifying the behavior of CTS transmissions, such false route breaks can be reduced significantly at and slightly beyond saturation. However, it does not mitigate sharp drops in throughput under high traffic overloads.

The IFQ between the network and MAC layers typically maintains priority queues. In the Glomosim implementation AODV uses two priority queues: a higher priority queue for control packets, and a lower priority queue for data packets.

We have examined the control queue lengths for one of the congested nodes (in one scenario) at loads of 400 Kbps, 500 Kbps, and 600 Kbps. At 400 Kbps, the control queue size rarely grew beyond 1. However, for loads of 500 Kbps (Figure 4.4) and 600 Kbps (Figure 4.5), the control queue's size often grew to large values. As a result, data packets were held at the low priority IFQ until the high priority control queue was

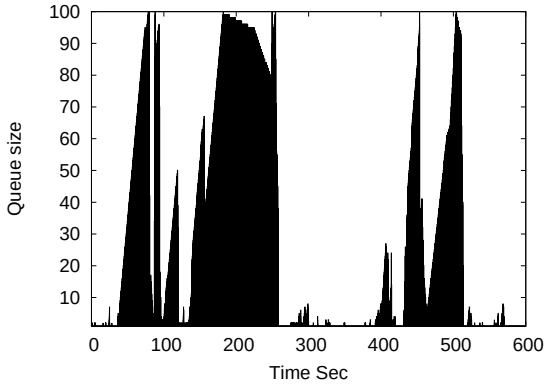

Figure 4.6: Data packet queue size for node 64 when offered load is 500 Kbps.

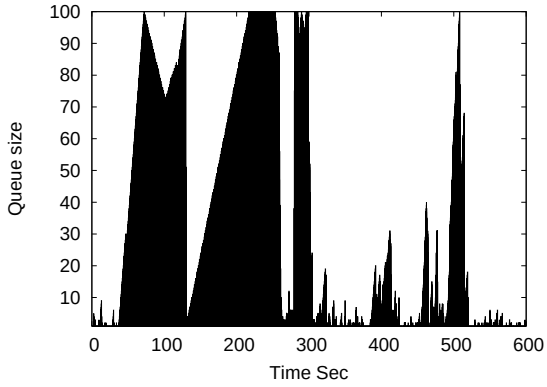

Figure 4.7: Data packet queue size for node 64 when offered load is 600 Kbps.

empty. Figure 4.6 (500 Kbps) and Figure 4.7 (600 Kbps) show that the data queue sizes remain very high for significant portions of the simulation time. In fact, our results in Figure 4.2 and Figure 4.3 indicate that the control packets dominate the data packets when the MANET is in saturation.

4.3 Reducing RREQ Explosion Using Efficient Broadcast Delivery

In this section, we briefly discuss and evaluate the efficient broadcasting techniques previously proposed and shown to reduce the number of RREQs in a MANET [75]. We implemented a probability-based broadcasting scheme that uses counters (Random Assessment Delay or RAD) [57] and a scheme that uses Neighbor-knowledge broadcasting (Scalable Broadcasting Algorithms or SBA) [56]. These two techniques are briefly discussed below, followed by a performance analysis of each technique.

4.3.1 Random Assessment Delay (RAD)

In the RAD technique, RREQs are rebroadcasted with a predetermined probability. When a node, say, x, receives a previously unseen RREQ, it starts a counter with a value of zero and sets a random delay between 0 to T_{max} seconds. During this selected random delay, x increments the count by one for each broadcast of this RREQ it hears. If the counter value is less than the preset value at the end of the random delay, then x rebroadcast the RREQ. Otherwise, it drops the RREQ. In our simulation, T_{max} is selected as 0.2 seconds, and the threshold value is set to be six; previous research shows that values above six are not effective

[52]. We ran the initial simulations with various values for T_{max} and selected 0.2 second because this gave consistently good performance.

4.3.2 Scalable Broadcasting Algorithm (SBA)

SBA requires that all nodes have knowledge of their neighbors that are within a two hop radius. To ensure this, each node periodically (for example, once every second) transmits a HELLO message (this includes lists of one hop neighbors) to its neighbors. Much like RAD, SBA selects a random time period to delay the relay of the RREQ, and the random delay is weighted by the relative neighbor degree. The neighbor degree is calculated as:

$$\frac{d_{N_{max}}}{d_{me}}$$

In this equation, d_{me} is the node's current number of neighbors, and $d_{N_{max}}$ is the maximum neighbor count of the node's neighbors. The random delay is computed as Uniform$[0, T_{max}] * \frac{d_{N_{max}}}{d_{me}}$.

Therefore, the node with the most neighbors usually broadcasts before the others do. After the random delay, the node processes the list of nodes covered by the redundant broadcast packets, using the 2-hop neighborhood table. If all of the node's neighbors are reached by the redundant broadcasts, then the RREQ is dropped; otherwise, the RREQ is transmitted.

In both RAD and SBA, the randomized delay accomplishes two things. First, it allows nodes to monitor redundant broadcast packets and assesses whether they need to rebroadcast the RREQ packet. As a result, the number of broadcast packets transmitted can be reduced. Second, the retransmission jitter is increased, which prevents broadcast collisions. In the first case, if the T_{max} value is high, a large number of broadcasts will be dropped, but the route discovery will take a long time. As an alternate method, Williams et al. [75] proposed that the broadcast be sent to the IFQ after a short random delay. In this case, the packet remains in the IFQ until the channel is ready to send. While the packet is in the IFQ, redundant packets are examined and assessed to determine whether the rebroadcast is still needed. If the routing layer determines that the rebroadcast is not necessary, then the broadcast packet is removed from the IFQ by the network or the MAC layer (depending on the implementation). A drawback to this approach is that it requires a complex IFQ

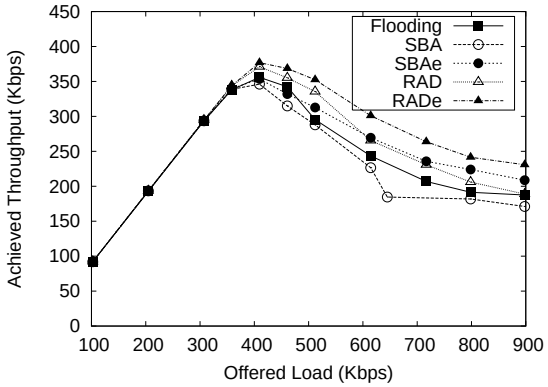

Figure 4.8: Performance of SBA, SBAe, RAD and RADe.

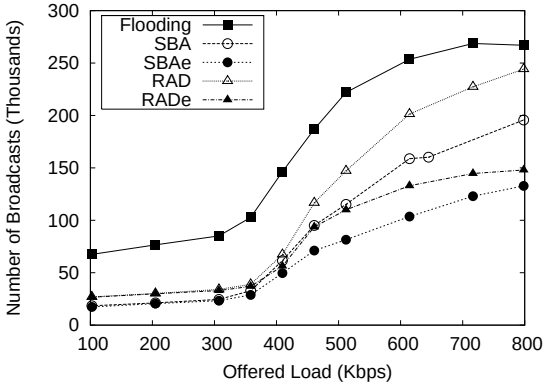

Figure 4.9: Number of broadcasts transmitted in SBA, SBAe, RAD and RADe.

management; for example, in this case the routing layer needs to assess redundant broadcasts and send a drop notification to the IFQ in order to drop the packet that is scheduled to be transmitted. The enhanced versions are identified as SBAe and RADe, respectively.

4.3.3 Performance of broadcast management schemes

We reran the traffic overload simulations with RADe and SBAe techniques incorporated. Figure 4.8 shows the 100-node simulation results for the AODV (Flooding) routing protocol, and for the modified versions of AODV, using SBA, RAD, SBAe and RADe. Unlike the RAD, SBA, RADe and SBAe, the AODV uses a simple flooding scheme to propagate RREQs; therefore, we label the AODV's performance as "Flooding" in the performance figures. Figure 4.9 shows the number of broadcasts for each of these techniques.

For loads below the saturation point (100-400 Kbps), the broadcast management schemes performed as well as the simple flooding scheme. Figure 4.9 shows that, for offered loads of 100-400 Kbps, the broadcast management schemes reduce the number of broadcasts by approximately 66%. In this range, the MANET has excess network capacity and is able to maintain the same level of throughput using simple flooding. Even though there is a significant reduction in the number of broadcasts in SBA, it is not able to achieve a higher throughput than simple flooding. As the network reaches the saturation point, SBA underperforms. The poor performance is primarily due to the cost associated with the HELLO messages. The benefit of the reduced number of broadcasts is mitigated by the cost of the HELLO messages. The number of broadcast packets transmitted in SBA and SBAe is nearly equal for loads of up to 400 Kbps. For loads beyond 400

Kbps, SBA increases the number of RREQs transmitted at a higher rate than SBAe does.

A similar trend is seen in the number of RREQs transmitted by RAD and RADe. Figure 4.8 shows a 7% increase in peak throughput by RADe, and 40% higher throughput than simple flooding at 900 Kbps loads. RAD, on the other hand, barely outperforms the simple flooding used by AODV. In fact, at very high loads RAD and SBA are less effective than RADe and SBAe in reducing the number of RREQs transmitted. The ability to drop a RREQ after it has spent time in the IFQ effectively stretches t_{max}, based on the queue size, for RADe and SBAe. Therefore, the enhanced implementation is used for the rest of this chapter.

SBA is able to reduce more redundant broadcasts than RAD, but due to the overhead associated with the maintenance of the 2-hop neighbor table, SBA fails to perform as well as RAD. In addition to the network overhead, SBA requires additional memory to maintain the 2-hop neighbor table, and it requires additional CPU cycles to determine whether a broadcast must be relayed or not. On the other hand, RAD is simple and adds only a small amount of overhead to the route management in the AODV protocol. Therefore, we selected RAD protocol as the better protocol of the two, and we select this technique to compare to our proposed RREQ reduction techniques.

4.4 Combining RADe with Modified MAC Protocol

We reran the simulations with the modified 802.11 MAC protocol and RADe. The throughputs for these cases are provided in Figure 4.10. Modifying the 802.11 MAC protocol to reduce false route breaks improves the peak bandwidth. This modification also helps the RADe broadcast management technique to improve the peak throughput. There is a gain of approximately 14% in the peak throughput when RADe is used along with our modified MAC protocol. At a load of 600 Kbps, RADe with the modified MAC gives 45% more throughput than simple flooding with the standard 802.11 MAC protocol. To evaluate a more realistic traffic scenario, we simulated TCP traffic on the example MANET. The traffic load consisted of up to 25 TCP connections with 200 Kbps of CBR background traffic. Figure 4.11 shows the aggregated TCP throughput achieved.

The results indicate that RADe in conjunction with the modified MAC protocol increase TCP throughputs up to 31%. Due to the congestion control mechanism with dynamic backoff, used by the TCP protocol,

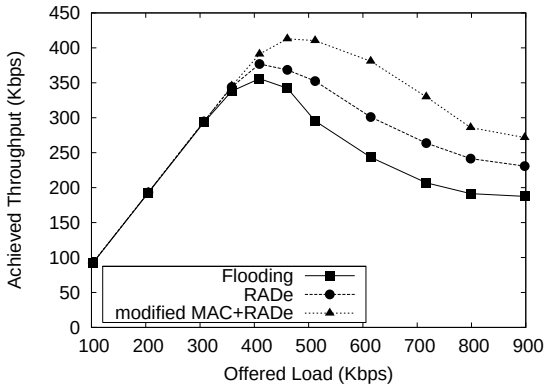

Figure 4.10: CBR performance using RADe with modified 802.11 protocol.

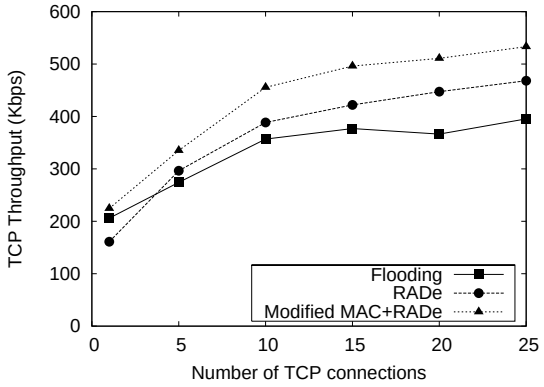

Figure 4.11: TCP performance using RAD and RADe broadcast techniques with modified MAC. A 200 Kbps CBR background traffic is used.

the MANET will operate near the point of saturation (the background CBR traffic is not enough to cause saturation). Therefore, reduction of RREQs in broadcast management techniques help RADe to increase its throughput over the simple flooding.

4.5 Reducing Unnecessary Control Packets

Our analysis of the IFQs (see Figure 4.4 and Figure 4.5) revealed that multiple RREQs with the same source and destination were queued in this node (due to repeated RREQ retries). Based on the design of the AODV, if an RREP is not received within the set duration of time, which is likely to occur in a severely congested network, then it sends another RREQ to that destination. Thus, a congested network causes more RREQ packets to be generated by the route discovery process. More importantly, multiple RREQs with increasing TTLs (due to the expanding rings mechanism used by AODV) sent by a node targeted to a destination could be sitting in a congested node's IFQ. In fact, this was observed in the queues of congested nodes, including the previously discussed node 64.

In this section, we propose a simple modification to examine the IFQ and eliminate such RREQs. We compare our modification to the original AODV RREQ propagation mechanism, denoted Flooding.

Prior to placing an RREQ in the IFQ, the RREQ is examined to see if an earlier RREQ with the same source and destination and the smaller TTL is present. If so, the earlier route request is replaced by the

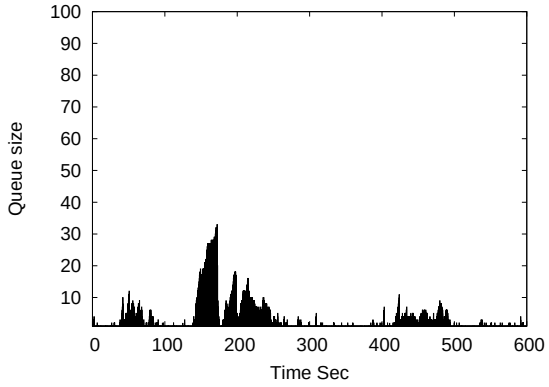

Figure 4.12: Reduced broadcast control packet priority queue size for node 64 when offered load is 500 Kbps.

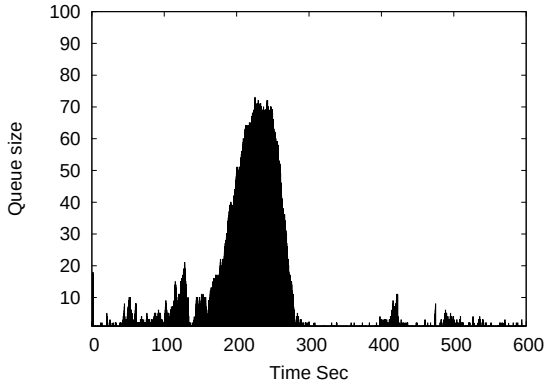

Figure 4.13: Reduced broadcast control packet priority queue size for node 64 when offered load is 600 Kbps.

newer RREQ. Otherwise, the RREQ is queued at the end, as is normally done. We call this reduced broadcast (RB). The proposed RB technique increases the queue management overhead in proportion to the length of the queue. Based on our simulations, the control queue size is rarely larger than two, for offered loads of less than 450 Kbps. Therefore, the proposed modification is not burdensome prior to network saturation. For high loads, the RB technique promises to reduce the control queue size significantly. Therefore, the extra overhead needed to implement this technique is not excessive. To evaluate the benefits of the proposed modification, we repeated the simulations and examined the control queue sizes for offered loads of 500 Kbps (Figure 4.12) and 600 Kbps (Figure 4.13). In comparison to Flooding (Figures 4.4 and 4.5), the proposed modification reduces the queue size significantly. In Figure 4.14, we plot the throughput achieved using Flooding and RB. Figure 4.14 shows that, when compared to Flooding, the reduced broadcast method gives slightly higher peak throughput and, more importantly, degrades more gracefully under traffic overload. For a traffic load of 700 Kbps, the reduced broadcast method sustains 71% of its peak throughput, while the original 802.11 protocol can only sustain 50% of its peak throughput.

4.5.1 Reduced broadcasts with Modified 802.11 MAC protocol

We reran the simulations using the previously proposed modified MAC and RB for the CBR traffic load. The throughputs for the four cases — the original 802.11 MAC protocol with AODV (Flooding), the 802.11 MAC protocol with AODV modified to reduce broadcasts (RB), the modified 802.11 MAC protocol with

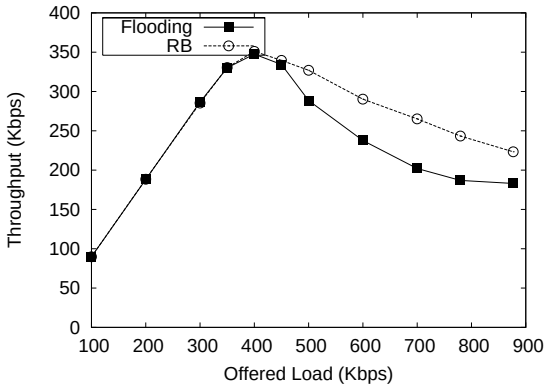

Figure 4.14: Improved throughput with reduced broadcast technique.

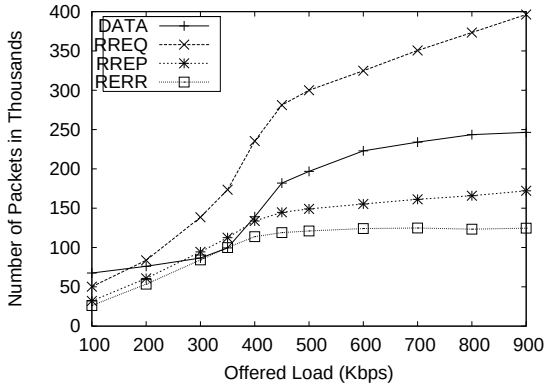

Figure 4.15: Control and data packets transmitted on wireless links when reduced broadcast technique is used.

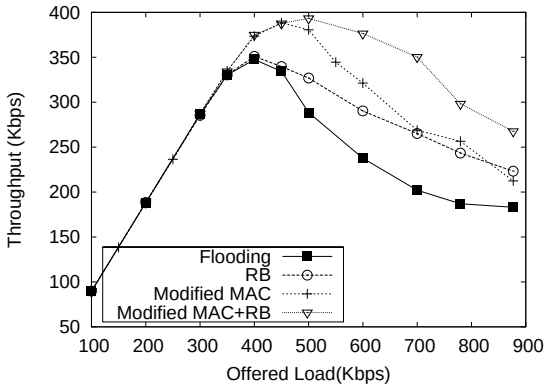

Figure 4.16: CBR throughput improvement with modified 802.11 protocol and reduced broadcast.

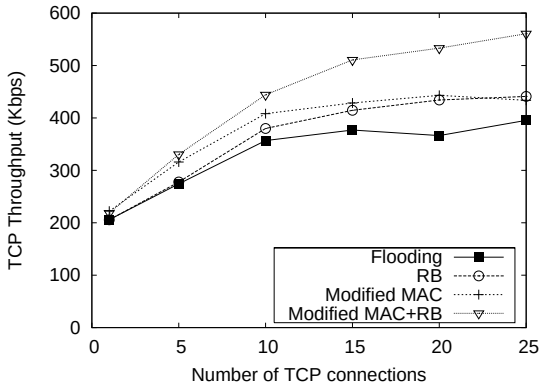

Figure 4.17: TCP throughput with modified MAC protocol and reduced broadcast. A 200 Kbps CBR background traffic is used.

AODV, and the modified 802.11 MAC protocol with AODV with reduced broadcasts — are provided in Figure 4.16.

The modified MAC protocol improves the peak throughput at the point of saturation but does not mitigate the sharp drop in throughput beyond saturation. However, with the RB and the MAC modification the throughput reduces gracefully (even under heavy loads). At 900 Kbps, the throughput is about 70% of its peak throughput, and it is 50% higher than with Flooding.

We also simulated TCP traffic on the example MANET. Figure 4.17 shows the aggregated TCP throughput achieved. For 25 connections, with 200 Kbps of CBR background traffic, the RB or the modified MAC increases the throughput by 6%; when both modified MAC and RB are used, the TCP throughput is in-

creased by 50%.

4.6 Dynamic Hop Time (DHT)

In section 4.2, we have shown that high network loads increase the control packet queue size. The primary reason is the duplicate RREQs sent by source nodes. In Section 4.5, we have proposed a relatively simple technique to remove duplicate RREQs from IFQs. In this section, we explore the possibility of reducing duplicate RREQ generation.

First we review the AODV route discovery process. After broadcasting a new RREQ, a node waits for an RREP. If an RREP is not received within the estimated RREP time, another RREQ is transmitted with a new estimated RREP time. There is a limit on the number of such retries. For each retry, the RREP time is doubled. The RREP time is calculated as:

$$2 \times \text{NODE\_TRAVERSAL\_TIME} \times (\text{TTL\_VALUE} + \text{TIMEOUT\_BUFFER}) \tag{4.1}$$

Here, the node NODE_TRAVERSAL_TIME (NTT) is the time it takes for a RREQ or RREP to propagate from one node to another. TTL_VALUE is the maximum number of hops the control packet may take, and the TIMEOUT_BUFFER is a static configurable parameter to increase the timeout period. The NTT is a predetermined value and is not adaptive to the network conditions. Using a large value for the NTT will result in slow discovery of routes if the first RREQ is lost at the source due to collision (broadcasts do not use an RTS/CTS). On the other hand, using a small value for the NTT will result in control packet flooding at high network loads (see Figure 4.2).

Therefore, we analyzed the effect of increasing the NTT. The example 100-node simulation is repeated in order to compare the performances of different NTT values. The default value of the NTT (40 milliseconds [59]) is compared to 3, 10, and 20 times the default NTT. Figure 4.18 shows the corresponding throughputs. By increasing the NTT by three times the default value, the peak bandwidth is increased slightly. More significantly, the MANET is better able to sustain the throughput than it is with the larger NTT. The disadvantage of using a very high NTT is that routes are repaired slowly and hence, packet dellays increase (see Figure 4.19), especially with expanding rings. Since the network conditions cannot be predetermined, using

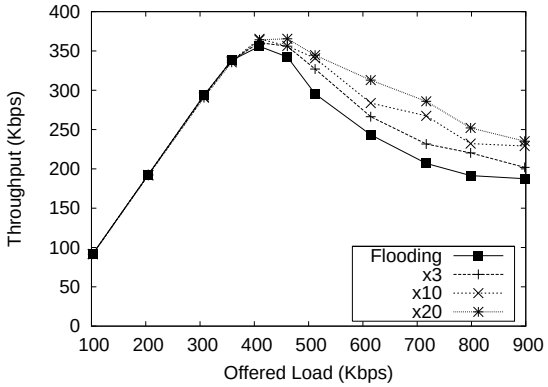

Figure 4.18: Performance of MANET after increasing the hop time by 3,10 and 20 times the value proposed single hop propagation time.

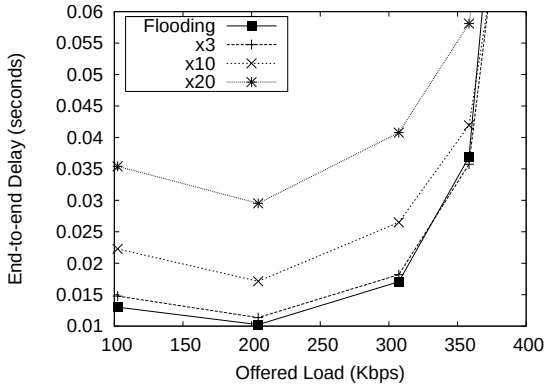

Figure 4.19: End-to-end delay of MANET after increasing the hop time by 3,10 and 20 times the value proposed single hop propagation time.

a static value for the NTT leads to unsatisfactory performance.

To mitigate the NTT problem, we propose the use of a dynamic NTT. Using the proposed dynamic NTT, we show that the MANET's performance can be improved significantly. We call it the Dynamic Hop Time (DHT) technique.

4.6.1 Estimating RREP time

As the network load is increased, the actual time taken per hop increases due to the increase in packet queuing time in the IFQ, and the increase in channel access time. The queuing time is significantly more than the channel access time and the propagation delay.

Proposed design

We introduce the concept of estimated travel time (ETT). When the MANET is operating under high loads, this traversal time is dominated by the queue delays. Therefore, the queue delay at a node can be approximated for a hop time for that node. When the traffic load is low, RREQs do not spend a significant amount of time in the queue; therefore, for low loads, a static NTT can be used.

Each node maintains a table for the ETT, with an entry for *each* hop count. RREQs are used to compute the ETT or the queue delay. The RREQ packet size is increased to record the Total Queue Delay (TQD) experienced by the packet. Each time the RREQ is transmitted, the time it spends waiting in the IFQ is

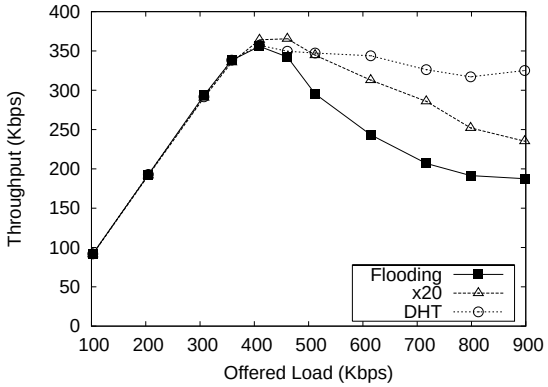

Figure 4.20: CBR performance of dynamic hop time and static hop times.

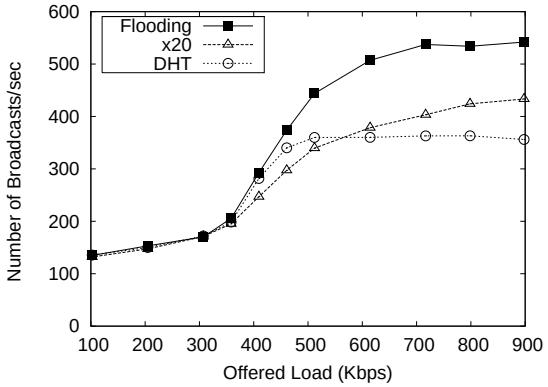

Figure 4.21: Broadcasts transmitted in dynamic hop time and static hop times.

calculated. This calculated value is added to the TQD field in the RREQ before the RREQ is transmitted.

When a node receives an RREQ, the node reads the TQD and the number of hops taken from the original destination. Based on the number of hops taken, the node updates the ETT for that hop count using the following smoothed-average function:

$$ETT_{new} = \frac{3}{4} ETT_{current} + \frac{1}{4} TQD \tag{4.2}$$

Here, the $ETT_{current}$ is the current value of the ETT, and the ETT_{new} is the new value of the ETT. When a node initiates a transmission with a specified TTL, the corresponding ETT is looked up on the ETT table and used to estimate the NTT value. If the there is no estimation for the required TTL, or if the estimated NTT value is less than the value of the static NTT, than the static value is used to compute the route repair time. We call this the dynamic hop time (DHT) technique to manage RREQ explosion.

Performance

To evaluate the performance of DHT, we reran the 100-node simulation setup that was previously simulated. Figure 4.20 includes the original protocol, using static NTT (denoted Flooding), 20 times NTT (denoted x20), and the proposed DHT. Neither mechanism has a noticeable effect at low network loads. While x20 has a marginally higher peak throughput, DHT is able to retain the throughput better than the x20.

It is noteworthy that DHT is able to sustain 95% of the peak throughput as the load is increased. Figure

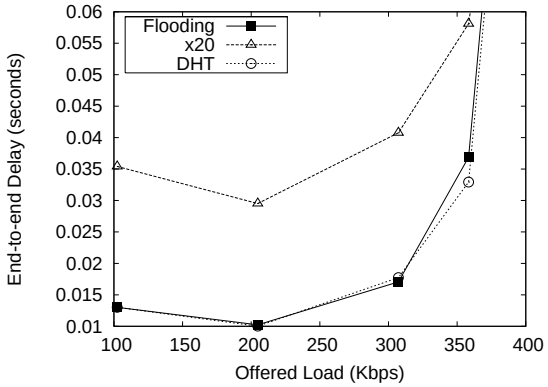

Figure 4.22: End-to-end delay of dynamic hop time and static hop times.

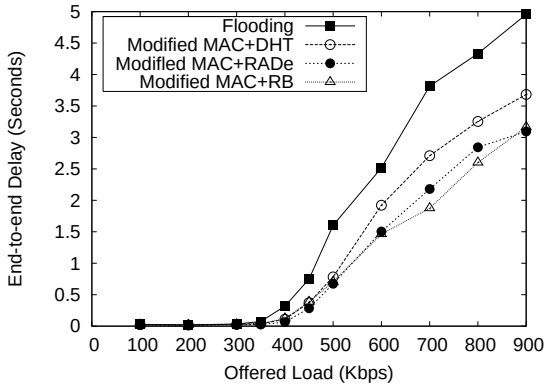

Figure 4.23: CBR average end-to-end delay for flooding, dynamic hop time with modified MAC, RADe with modified MAC and reduced broadcast with modified MAC protocols.

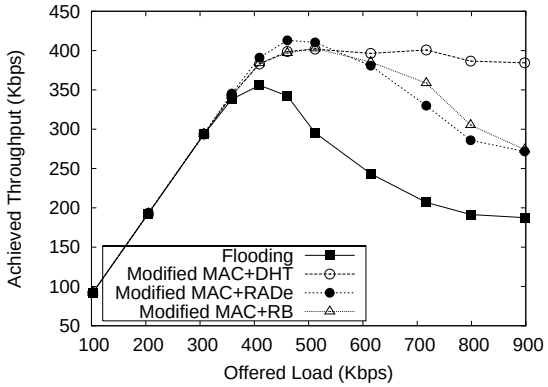

Figure 4.24: CBR throughput of flooding, dynamic hop time with modified MAC, RADe with modified MAC, and reduced broadcast with modified MAC protocols.

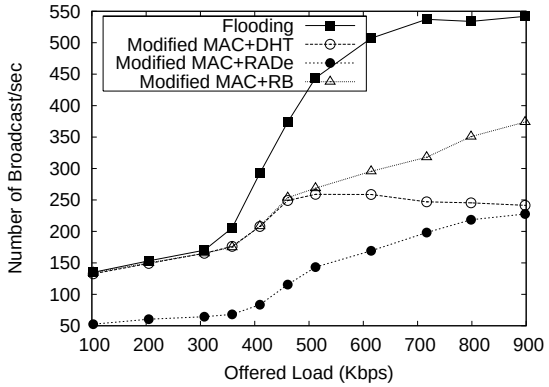

Figure 4.25: Broadcast transmitted in flooding, dynamic hop time with modified MAC, RADe with modified MAC, and reduced broadcast with modified MAC protocols.

4.21 shows the total broadcasts (RREQs) transmitted. In comparison to flooding, x20 and the DHT are able to limit the number of RREQs as the load increases, with DHT being more adaptive than x20. Reducing the number of RREQs is a direct result of higher throughput in x20 and DHT. Figure 4.22 shows end-to-end delays for flooding, x20, and the DHT for unsaturated network loads. The x20 increases the end-to-end delay by a factor of 2 for low loads, whereas DHT is able to maintain low end-to-end delay comparable to that by Flooding.

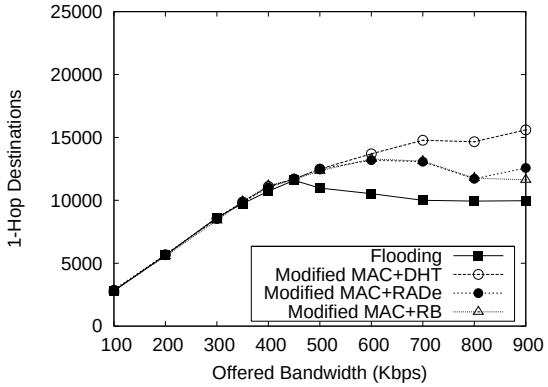

Figure 4.26: CBR: Total number of packets traveled 1-hop to reach the final destination.

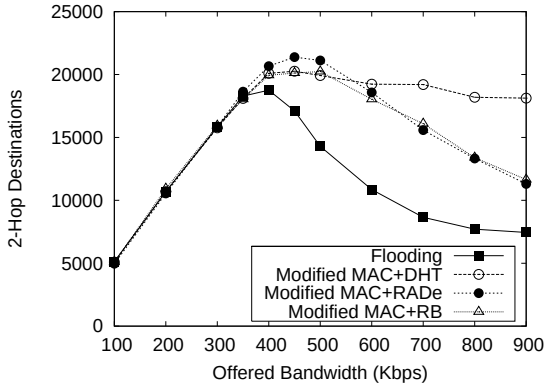

Figure 4.27: CBR: Total number of packets traveled 2-hop to reach the final destination.

4.7 Comparing Broadcast Reduction Techniques

In this section, we compare RADe, RB, and DHT techniques with the proposed MAC to the default AODV and standard 802.11 MAC protocol combination (denoted Flooding). Figure 4.23 shows the average end-to-end delay for delivered packets for the CBR traffic. Compared to Flooding, the three protocols (RB, RADe, DHT) reduce end-to-end delay by a significant amount for loads beyond saturation (400 Kbs). The reduced broadcast has the lowest end-to-end delays of the three techniques. The DHT has higher end-to-end delays since it delivers more long-hop data packets than the other two techniques. Figure 4.24 indicates that RADe and RB have comparable throughputs, while DHT is significantly better. Figure 4.25 gives the RREQs transmitted with each technique.

To evaluate whether the proposed modifications favor shorter routes over longer routes, we examined the number of packets that traveled 1, 2 and 3 hops to reach the final destination. Figures 4.26, 4.27, and 4.28 show the total number of packets that traveled 1, 2 and 3 hops to reach the final destination, respectively. It is clear from these figures that each proposed modification sustains more long route deliveries than the default Flooding case, indicating that the proposed modifications do not favor shorter-route connections to achieve higher throughput. We have further examined the number of 4-hop and higher hop count data packet deliveries, not shown here, and they did not show any evidence that RB, DHT, and RADe favor shorter routes. In fact, using Jain's fairness index, we show in Figure 4.29 that our proposed modifications improve the fairness of BW allocated to CBR connections for network loads beyond saturation.

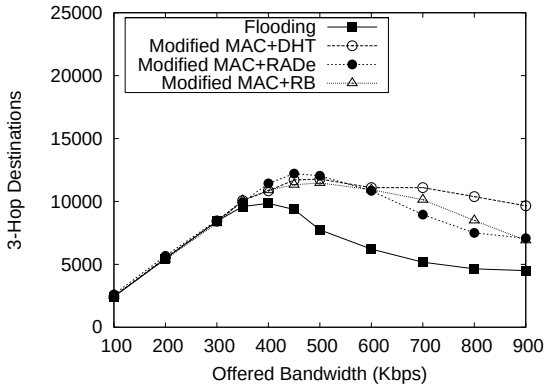

Figure 4.28: CBR: Total number of packets traveled 3-hop to reach the final destination.

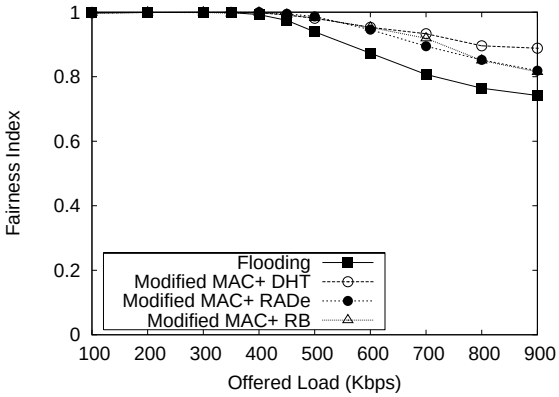

Figure 4.29: Jain's fairness index for the proposed modification.

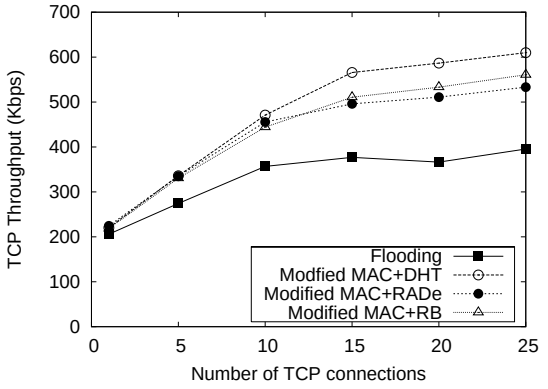

Figure 4.30: TCP performance of flooding, dynamic hop time with modified MAC, RADe with modified MAC, and reduced broadcast with modified MAC protocols.

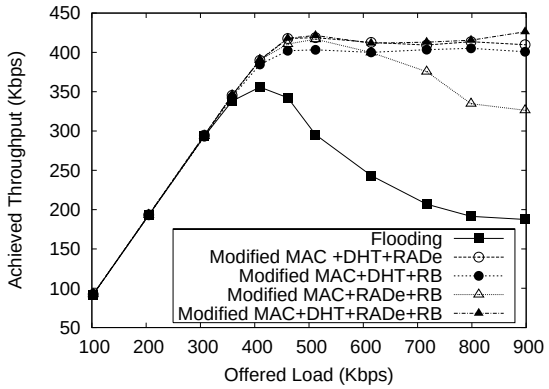

Figure 4.31: CBR performance using DHT with RADe, DHT with RB, RADe with and DHT, RADe with RB. All combined techniques use modified MAC.

To evaluate the proposed modifications using a more realistic network load, TCP traffic and a fixed CBR background traffic were used. The MANET parameters and simulation parameters are identical to those in the CBR simulation. One through twenty-five FTP connections were simulated along with 200 Kbps of CBR background traffic, using 50 connections. Figure 4.30 presents the achieved TCP throughput. The proposed modifications provide a significant improvement in throughput. DHT, RB, and RADe were able to achieve higher throughputs, up to 60%, 45%, and 39%, respectively.

4.7.1 Combining multiple techniques

The DHT, RADe, and the RB techniques reduce the number of broadcasts transmitted on the MANET in different ways. In DHT, sources adaptively generate fewer broadcasts, such that unnecessary RREQs are reduced. RADe, on the other hand, will deliver an RREQ to the destination with fewer broadcasts. The RB examines the IP to MAC interface queue and removes duplicate RREQs within a node. In this section, we explore the benefit of combining these techniques.

Figure 4.31 shows the performance for CBR traffic when DHT is used with RAD, DHT with RB, and RAD with RB. Every combination, with the exception of the RADe and RB combination, performed well and retained 95% of its peak throughput. RADe with RB is the only combination that does not contain the DHT. Therefore, to retain the high peak bandwidth in a saturated network, DHT is highly recommended.

4.8 Conclusions

In this chapter, we have examined the throughput of MANETs under heavy traffic loads that exceed the point of saturation. We believe that MANETs should be designed to handle high-traffic loads and exhibit graceful degradation of performance in such situations. Using the AODV as the routing protocol and the 802.11 MAC protocol, we have investigated the throughput behavior of a 100-node MANET for CBR traffic ranging from 100 Kbps to 900 Kbps, and TCP traffic with up to 25 connections and 200 Kbps of CBR background traffic. The MANET saturates at a load level of 400 Kbps when a CBR load is offered, but it retains only a half of its peak throughput when the load is increased to 900 Kbps. The primary reasons are that (a) an overactive route discovery process causes too many route control packets to fill the interface queue and dominate the MAC-level packet transmissions, and (b) exposed nodes cause RTS timeouts and false route breaks, which, in turn, makes the route discovery process send more control packets. This seems to cause the network to go into a tail spin, and throughput drops sharply.

We have proposed and evaluated two different techniques to reduce the number of RREQ transmissions in the MANET. The first technique reduces duplicate RREQs at the IP layer, and the second technique adaptively adjusts the route repair time. In addition, the benefits of the modified MAC protocol proposed in

Chapter 3 are evaluated, along with the proposed techniques. We also evaluated a generic broadcast management technique, denoted RADe, that reduce the number of RREQs by reducing the number of broadcast relays. Using extensive simulations, we showed that oune of our proposed techniques, DHT, retains nealy all of the peak throughput even under heavy traffic loads.

Chapter 5

Next Hop Prediction Techniques

Several proactive and reactive dynamic routing protocols have been proposed and analyzed in the literature [60, 38, 17, 54, 19, 11, 27]. Proactive protocols constantly maintain routes by exchanging routing information periodically or when the network topology changes significantly. This results in a nearly constant overhead for route discovery and route maintenance when proactive protocols are used. On the other hand, reactive protocols seek routes on-demand and maintain routes as needed. This results in a varying routing overhead as the network condition changes. A reactive protocol's routing overhead increases as the node mobility, offered load, or number of connections increase since it does not refresh them on a regular basis. When a node becomes more mobile, the lifetime of a route decreases and the route breaks more frequently.

A reactive protocol must repair the broken routes. When the route is broken, the protocol must (a) detect the broken link that caused the route to break, (b) invalidate all routes using the broken link, (c) discover or replace it with an alternate route. Reactive routing protocols using the 802.11 MAC protocol rely on the link layer feedback to detect failed transmissions. A transmission failure is due to a non-responsive next hop. These transmission failures are treated as broken links by on-demand routing protocols such as AODV and DSR.

There are several reasons for a non-responsive next hop. The most obvious reason is the next hop moving out of the communication range. In addition, the next hop will not respond to a transmitting node's RTS with a CTS if it is exposed to another transmission. The 802.11 MAC protocol cannot distinguish between these two types of transmission failures, so the 802.11 MAC protocol reports both transmission failures to the routing layer. The latter type of failure is temporary and should not be considered a link failure; the former

type of failure is a permanent failure that requires route repair. By incorrectly treating an exposed next hop node as an out of range node, a routing protocol unnecessarily invalidates one or more routes established through this next hop. If the 802.11 MAC protocol can distinguish between the reasons for a non-responsive next hop, unnecessary link failures can be avoided.

Similarly, DATA transmission failures due to collision also increase false link failures. Here, the next hop is within range and response with a CTS and starts receiving DATA, but it does not receive the entire packet due to a burst of noise in the MANET. This scenario is also temporary, since the node noise can reduce after a while to permit successful reception.

When the MANET is in saturation, false link failures such as the one described above become frequent. These false route breaks increase the control overhead incurred to maintain routes. Increase in control overhead can adversely affect a MANET's performance. Therefore, false route breaks have a "double whammy" effect since traffic overload causes false route breaks, which, in turn, increase the control overhead and further saturate the network.

With the definition of false route breaks given above, all route breaks can be divided into false route breaks and real route breaks. Real route breaks are route breaks generated when the next hop node moves out of the communication range of a transmitting node in the path. Typically, a MANET has a few real route breaks, but an excessively large number of real route breaks can adversely affect the MANET's performance. Stale routes cause such excessively large numbers of real route breaks.

When the MANET is below saturation, increase in routing overhead does not cause significant loss in the MANET's performance as long as the increased overhead is less than the MANET's unused capacity. Therefore, increased false route breaks do not affect the MANET when it is below saturation. Real route breaks can affect the packet delivery rate and lower the MANET's performance, even when it is not congested. On the other hand, false and real route breaks can cause a significant problem when the MANET is saturated. Figures 5.1 and Figure 5.2 show the baseline MANET's performance using AODV and DSR routing protocols and their control overhead, respectively.

In this study, we investigate prediction schemes to aid the MAC layer in determining when the next hop is really in communication range or moved out of range. If the prediction schemes are able to aid the

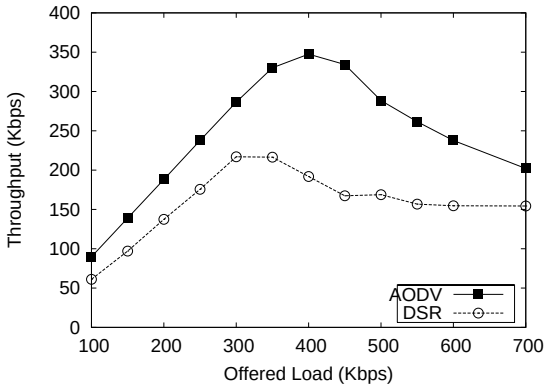

Figure 5.1: Baseline MANET's performance with AODV and DSR.

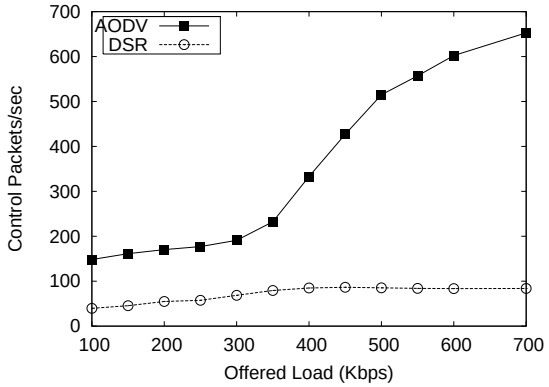

Figure 5.2: Baseline MANET's total control packets (RREQ, RREP, and RERR) with AODV and DSR.

MAC layer, then the number of false route breaks can be reduced and the impact of real route breaks can be mitigated to some extent. We propose various techniques to predict a next hop status (in range or out of range). Our schemes are designed to be implemented at the MAC sub-layer level, so they could benefit many types of routing protocols. Using simulations, we show that some of the prediction schemes do provide a significant performance gain when the MANET is in saturation.

5.1 Classification of Prediction Schemes

The event of interest is the node's transmission of a data packet to the next hop specified in the destination route. As discussed before, the unicast data packet is preceded by an RTS/CTS exchange between the two nodes. Therefore, two types of prediction strategies are feasible: *pre-transmission prediction*, in which the sending node predicts whether the next hop node is in range or out of range even before attempting to transmit; and *post-transmission prediction*, in which the sending node predicts, after a transmission attempt fails, whether the non-responsive next hop is still in range or has moved out of range of communication. Accordingly, link failure notification is sent to the routing layer. Prediction data is updated after each transmission is received from a neighbor and also after a transmission failure. Both types of predictions can be implemented by modifying the MAC layer and require no changes to the routing protocol.

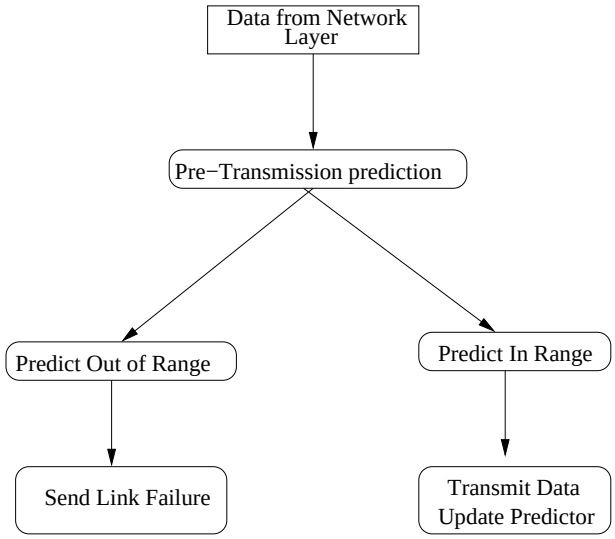

Figure 5.3: Illustration of pre-transmission prediction.

5.1.1 Pre-transmission prediction

Pre-transmission prediction is used prior to transmitting data to determine whether the next hop node is within range or out of range. If the next hop is predicted to be out of range, then the data is not transmitted; instead, a link failure notification is sent to the routing layer. If the out of range prediction is correct, then the node saves an unsuccessful attempt to send data. In the 802.11 MAC protocol, a node will attempt a *Short Retry count*, number of RTS (simulation default 7) transmissions, or a *Long Retry count* (simulation default 4) of data transmissions before transmission failure is generated. Therefore, if the out of range link prediction is correct, the node will reduce the number of unsuccessful RTS attempts. However, an incorrect out of range prediction for a next hop that is still in range costs dearly since the routing protocol invalidates an existing route and initiates route maintenance, which causes unnecessary control overhead.

If the next hop is predicted to be in range, then the data is transmitted. In this case, the prediction scheme neither provides benefit nor causes harm compared to the original 802.11 MAC protocol. In fact, the 802.11 MAC protocol can be considered as using a pre-transmission prediction that always predicts the next hop to be in range. Figure 5.3 shows the flow diagram of the pre-transmission prediction.

A node collects prediction data from the transmissions it receives from its neighboring nodes. In pre-transmission prediction, if the next hop is predicted to be in range, then the node will send an RTS. If the

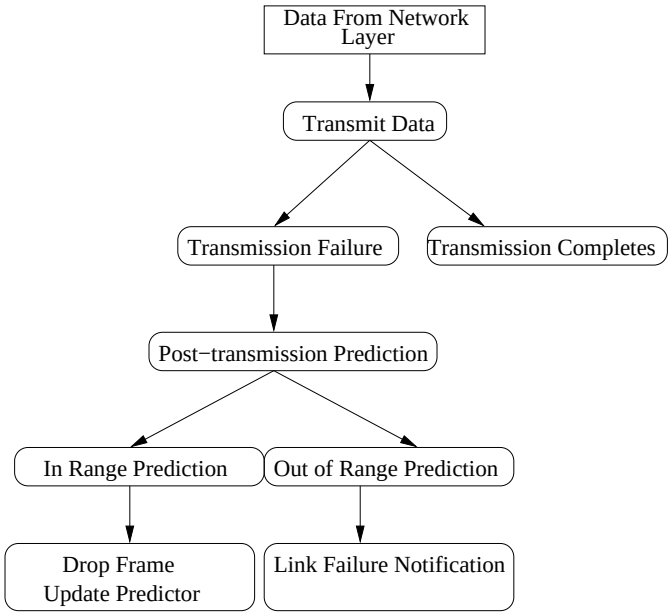

Figure 5.4: Illustration of post-transmission prediction.

next hop node responds with a CTS, then the node can update its prediction status. Similarly, if there is a transmission failure or non response from the next hop neighbor, this information may be used to update prediction data.

If the pre-transmission prediction is performed accurately, nodes can reduce the impact of real route breaks. By predicting an out of range next hop (real route break) correctly, nodes can eliminate transmission attempts in broken routes. This can help low-adaptive routing protocols that are slow to repair routes or that do not purge stale routes in a timely manner. The pre-transmission prediction has little or no impact on reducing false route breaks. In fact, incorrect out of range prediction of the next hop causes an increase in false route breaks.

Post-transmission prediction

Post-transmission prediction is used after a data transmission attempt fails. When a data transmission failure occurs, the predictor is used to determine whether the non-responsive next hop node is in or out of range. If the prediction is out of range, a link failure notification is generated and passed on to the routing layer, which is the default behavior for the 802.11 MAC protocol. Otherwise, the packet is *silently* dropped, and no link failure is sent to the routing layer. An out of range prediction neither provides benefit nor causes harm

when compared to the original 802.11 MAC protocol. Whenever a non-responsive next hop is correctly predicted as in range, however, substantial benefit can occur by avoiding unnecessary route invalidations and route discovery overhead since false route breaks are reduced. Figure 5.4 illustrates the application of post-transmission prediction.

Post-transmission prediction reduces false route breaks but does not reduce the real route breaks. In fact, an incorrect out of range prediction delays detection of real link breaks, which results in the use of a route that is no longer valid (stale route). Therefore, incorrect out of range prediction increases the number of stale routes.

5.2 Basic Prediction Criteria

For both pre- and post-prediction methods, a variety of criteria may be used as the basis for prediction. We propose four basic prediction criteria. These basic criteria may be used individually to obtain simple prediction schemes, or they may be combined to obtain more complex prediction schemes.

5.2.1 Time-based prediction

In the time-based prediction scheme, each node records the last time it received a communication from each of its neighbors. This information is gathered at the MAC layer and recorded for any communication heard from neighboring nodes. A neighbor may be heard when it transmits a MAC level RTS, CTS, DATA, or ACK frame. Prediction is then based on the elapsed time between the last heard time and the time at which the prediction is made. If the elapsed time is greater than a preset threshold, then the next hop is predicted to be out of range; otherwise, it is predicted to be in range.

After experimenting with several values, we found that a five second threshold value works well in the MANETs we simulated. A shorter threshold leads to premature ending of valid routes and increases false route breaks, while a longer threshold leads to slow discovery of real route breaks. The nodes we simulated move at a speed in the range of [1,19] meters/second. The relative speed of a pair of nodes could vary from -38 mps to +38 mps. In the worst case scenario, a node could be moving away at a speed of 38 mps. Within a five second period, nodes can move 190 m, or 50% of the communication range.

5.2.2 Distance-based prediction

The distance-based predictor attempts to predict the distance between the node and the next hop. If the predicted distance is greater than the communication range (376 m), then the next hop is predicted to be out of range; otherwise, it is predicted to be in range. The next hop distance is predicted using the last known relative speed of the two nodes, the last known distance, and the last heard time of the next hop node. The last heard time, last estimated distance, and last estimated relative speeds are calculated and recorded for each neighboring node when a transmission from that node is heard. To estimate the relative speed of a neighbor, it must be heard at least two times. If a neighbor is heard at times t_1 and t_2, then the time difference δ_t is computed. If δ_t is very small (70 milliseconds) or very large (5 seconds), the relative speed is not calculated or updated; otherwise, the relative speed is calculated as follows.

We compute the distance between the nodes using the signal strength of the transmissions at times t_1 and t_2. Signal strength information captured by the radio layer is available to the MAC layer, and the distance between the nodes can be calculated using 3.1 and 3.2 (given in Chapter 3). Let these distances be d_1 and d_2, respectively. If the distance can be estimated for two consecutive time periods, then relative speed v is computed using the following:

$$\delta_t = t_2 - t_1 \ ; \quad \delta_d = d_2 - d_1 \ ; \quad v = \frac{\delta_d}{\delta_t}$$

A negative relative speed indicates that the nodes are moving toward each other. A positive relative speed indicates that the nodes are moving away from each other. When a prediction is to be made at time t_3, estimated speed v at time t_2 is used to predict the change in distance, $\delta_{d_{(t_2,t_3)}}$ since time t_2 using $\delta_{d_{(t_2,t_3)}} = v_{t_2} * (t_3 - t_2)$.

Now, the estimated distance to the next hop is calculated as $d = d_2 + \delta_{d_{(t_2,t_3)}}$. If the distance d is greater than 376 m, then the distance predictor predicts the next hop to be out of range; otherwise, it is predicted to be in range.

5.2.3 Signal-to-Noise ratio (SNR) prediction

The SNR predictor is similar to the distance predictor. The SNR predictor estimates the signal strength of a transmission and the noise level at the destination based on recent history. The SNR predictor maintains signal strength and the noise level for all neighboring nodes. Owing to the complexity of obtaining the noise level at its neighbors', the predicting node uses its own noise level as an approximation of the next hop's noise level. To predict the signal strength of its transmission at the next hop, the node uses the previous transmission's signal level and the rate at which the signal is changing. Thus, the SNR predictor predicts the possible SNR at the next hop. If this estimated SNR is greater than 10 dB, then the next hop is predicted as reachable; otherwise, the next hop is predicted as unreachable.

To predict the signal strength, the last received signal strength and the rate at which the signal strength has been changing are needed. If the node receives transmissions from the destination node at time t_1 and time t_2, $t_2 > t_1$ with signal power S_1 and S_2, then the signal rate of change (SR) is computed as follows:

$$SR_{(t_2,t_1)} = \frac{S_2 - S_1}{t_2 - t_1}$$

To predict the signal strength at time $t_3 > t_2$, the node uses the following:

$$\text{Predicted}S_{t_3} = S_2 + SR_{t_2,t_1} * (t_3 - t_2)$$

Each time a node receives a transmission, its average noise level is updated using the following smoothing function:

$$\delta_n = CurrentNoise - AvgNoise$$

$$AvgNoise = AvgNoise + \frac{1}{4}\delta_n$$

Finally, the predicted SNR, can be determined:

$$\text{Predicted SNR} = \frac{\text{Predicted}S_{t_3}}{AvgNoise}$$

The SNR predictor is very similar to the distance predictor. The SNR predictor requires the average noise level information in addition to signal strength information for prediction. The SNR predictor is a

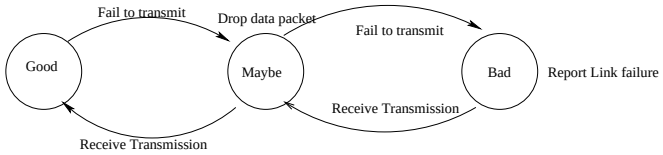

Figure 5.5: State diagram for the state predictor.

better prediction scheme for node reachability than a distance prediction is. For example, let's assume that the next hop is at a distance of 375 m. If the distance predictor predicts this correctly, the next hop is considered reachable. However, in a MANET with high traffic, the probability of successful communication is low, since, at this distance, a small amount of noise in the MANET is sufficient for next hop data to collide. On the other hand, if the SNR predictor is used, noise will be factored into the prediction; therefore, it is likely that the SNR predictor will predict the next hop to be unreachable.

5.2.4 State-based prediction

A state predictor uses a technique that is similar to the two-bit branch prediction used in the pipelined data path of a high-performance CPU [55]. Instead of using four states, this predictor uses three states: $Good$ (node is in range), $maybe$ (node may not be in range), and bad (node is out of range) states. The state diagram for the predictor is shown in Figure 5.5. When a node attempts to transmit a MAC frame to its next hop, but the transmission is not successful, the next hop's state is changed to *maybe* if the next hop's current state is *good*, to *bad* if the current state is *maybe*, or kept in *bad* otherwise. Whenever the node hears its next hop's transmission (possibly to another node), it changes the neighbor's state to *maybe* if the current state is *bad*, or to *good* if the current state is *maybe*, or leaves it in the *good* state otherwise.

Compared to distance and SNR predictors, the state predictor requires less memory, and there are fewer computations to collect prediction information and to make a prediction.

5.2.5 Distance predictor with global knowledge (Global)

In addition to the four predictors used earlier, a fifth predictor is used for comparison purposes. This predictor assumes that it has the exact distance between the node and its next hop at the time of the transmission. In the global predictor, the real distance between a transmitting node and its next hop is calculated using

the position coordinate information that is available in simulation. If the distance between the node and its next hop is greater than 376 m, the next hop is predicted to be out of range; otherwise, it is predicted to be in range. This predictor is easy to implement in a simulation environment, but is unrealistic to use in an actual MANET. (If it were to be implemented in MANET, the nodes would need to use a Global Positioning Satellite (GPS) along with an elaborate protocol to propagate the node position information to all nodes in the MANET.) We use the global predictor as an upper bound on the maximum performance improvement feasible with next hop prediction at the MAC level.

5.3 Analysis of Basic Predictors

To analyze the performance of the four basic prediction schemes for pre- and post-transmission predictions, we use the AODV and DSR proactive routing protocols, described in Chapter 2.

5.3.1 Evaluating real and false link breaks in AODV and DSR

To evaluate the relative frequencies of real and false link breaks, we used the baseline MANET simulation with CBR traffic load varied from 100 Kbps to 700 Kbps. Figures 5.6 and 5.7 show the average false and real link breaks, for AODV and DSR respectively, as a function of network load. To determine whether a failed transmission is due to a false or real link break, we examined the simulation data. If the distance between a transmitting node and its next hop is less than the communication range (376 m), then it is a false link break; otherwise, it is a real link break. Each false link break results in a false route break, which triggers a route discovery process by the source of the connection. Route discovery floods a portion of, or the entire, network with RREQ packets.

Since mobility of the nodes was not changed for different loads, a good routing protocol is expected to maintain a constant number of real link breaks with little increase in false link breaks. In both AODV and DSR, this is indeed the situation until the point of saturation. The number of false link breaks is very low prior to saturation, but it climbs rapidly after saturation is reached (350+ Kbps). This is due to more and more nodes being exposed to frequent transmissions within their sensing range and their inability to respond to their senders' RTS transmissions. In AODV, the number of real link breaks remains steady until

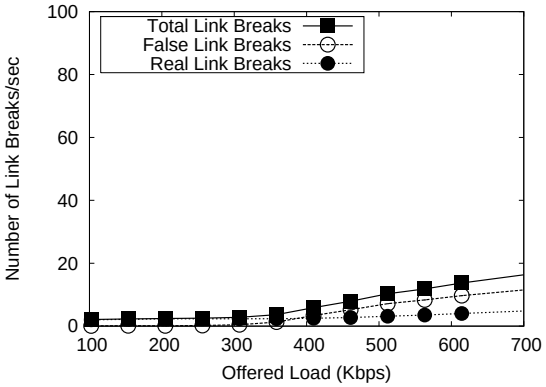

Figure 5.6: AODV real route breaks and false route breaks in MANET.

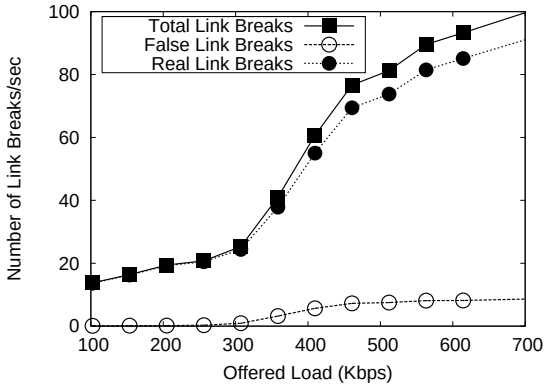

Figure 5.7: DSR real route breaks and false route breaks in MANET.

the load is high (500+ Kbps), at which point RERRs are delayed due to channel contention and then dropped by intermediate nodes due to congestion. This causes source nodes to continue to use stale routes, which increases the number of real link breaks at high loads.

Unlike AODV, DSR has a significantly high number of real route breaks for all loads. The number of real route breaks increases rapidly as the network reaches saturation (300 Kbps). DSR has a large number of real route breaks because of the stale routes problem described in Chapter 2. As the number of real route breaks increases, DSR attempts to salvage more data packets using alternate routes in intermediate nodes' route caches. If these alternate routes are stale, then data packet salvaging will result in more real route breaks at a downstream node in the salvaged path or at the node that attempted data salvaging. An increase in real route breaks results in data packet drops at intermediate nodes after the data packets consume the network bandwidth. In addition, false route breaks also increase with traffic load, as in the case of AODV. Since control traffic is slow, even at high loads, there is no significant impact of false route breaks on DSR. Therefore, DSR perfors poorly at all loads due to the high number of real route breaks.

5.3.2 Performance of pre-transmission prediction

The four basic prediction schemes are simulated using the 100-node MANET described in the previous section. The prediction models are used in conjunction with the pre-transmission prediction. Figure 5.8 is an expanded version of Figure 5.3, with the cost and benefit details of correct and incorrect predictions (see also Table 5.1). In this work, we do not consider the cost of implementing a prediction scheme. Our analysis

Table 5.1: Summary of pre-transmission prediction actions, benefits, and costs.

	Predict next hop in range	Predict next hop out of range
Correct prediction	Transmit data Benefit and cost of are the same as in the standard 802.11 MAC.	Do not transmit data MAC: Send link failure to routing protocol. Routing: use an alternate route, send RERR to the source of data packet, or initiate route discovery. Benefit: A transmission attempt is saved.
Incorrect prediction	Transmit data Benefit and cost are the same as in the standard 802.11 MAC.	Do not transmit data MAC: Send link failure to routing protocol. Routing: use an alternate route, send RERR to the source of data packet, or initiate route discovery. Cost: A false route break is created.

considers only network BW wasted or saved by the prediction scheme. Predicting the next hop to be always in range is the same as the normal 802.11 MAC protocol operation. In this case, all data will travel through branch B (predict in range) in Figure 5.8, so there is no additional benefit or cost for in range prediction. Therefore, the performance changes due to predictions depend on the number of packets taking branch A in Figure 5.8. Each packet taking branch C in the diagram (correct prediction of next hop out of range) saves an unnecessary transmission attempt, which is several RTS attempts. Therefore, the benefit of taking branch C is very small. On the other hand, each data packet that takes branch D (incorrect out of range prediction) results in a false route break. A false route break will result in an unnecessary route discovery that increases the control overhead, which can be significant. Comparing the benefits of correct prediction and costs of incorrect prediction, we note that several correct predictions must be made to overcome the loss of a single incorrect prediction. Therefore, for the pre-transmission to be effective, the $\frac{C}{D}$ ratio must be very high. Since the benefit from correct prediction is not significant, for a noticeable improvement, A needs to be very large.

Figure 5.7 shows that DSR has a large number of real route breaks, so DSR's performance may be improved by an appropriate pre-transmission prediction. On the other hand, Figure 5.6 shows that AODV

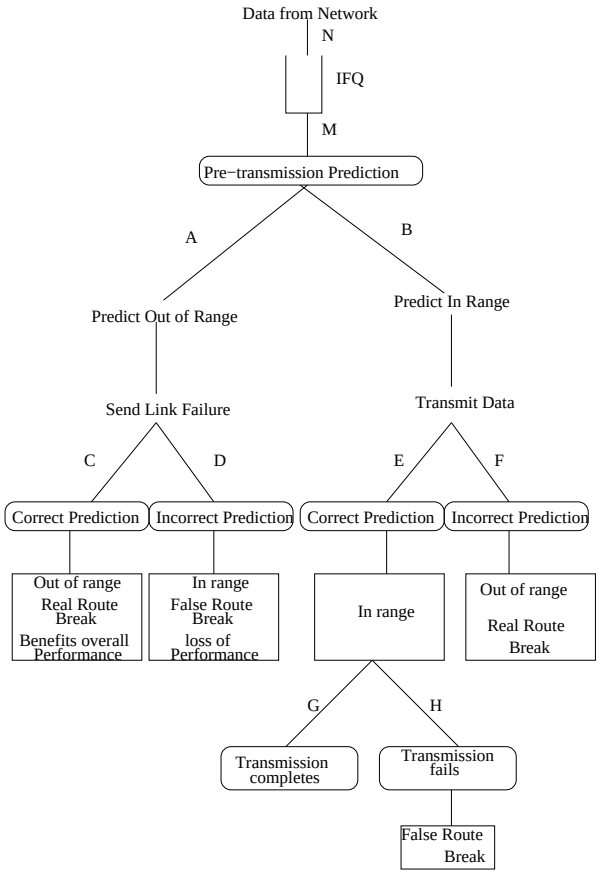

Figure 5.8: Pre-transmission predictor flow diagram. The letter next to a branch indicates the number of packets taking that branch.

contains only a few real route breaks in comparison to false route breaks, so we do not expect AODV's performance to be improved by using pre-transmission prediction. We give a detailed performance comparison of AODV and DSR with pre-transmission prediction below.

AODV: Figure 5.9 shows AODV performance with pre-transmission prediction. With the exception of SNR prediction, other pre-transmission prediction techniques do not show any significant gain or loss of performance in comparison to the original MAC protocol. SNR prediction did not perform well with loads below saturation or beyond saturation. Figure 5.10 shows the number of RREQs transmitted for all prediction schemes. The state predictor's number of RREQs remains close to that of the original MAC protocol (denoted AODV-MAC). For further analysis, we chose the state prediction technique and the worst performing (SNR) prediction technique. For comparison, we also selected the global predictor and the original

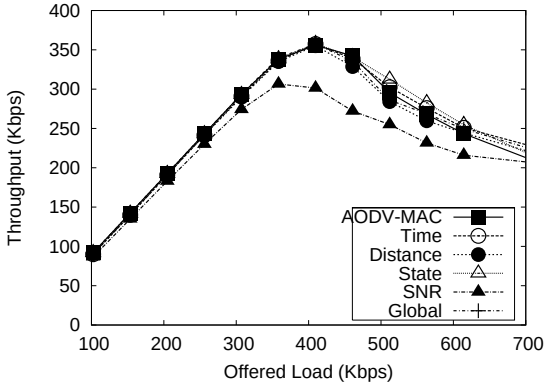

Figure 5.9: AODV pre-transmission prediction schemes in MANET.

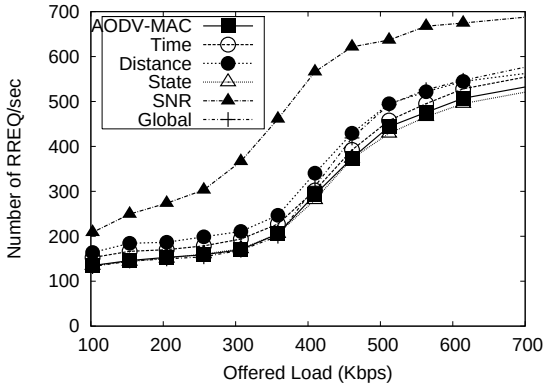

Figure 5.10: AODV total RREQs transmitted in pre-transmission prediction schemes.

MAC protocol.

Figure 5.11 shows the number of out of range packets arriving at the network-to-MAC IFQ. Higher values indicate more stale routes. SNR and Global predictors reduce the number of stale routes by breaking routes frequently. When the link layer indicates the loss of a next hop, AODV drops all the packets that are in the IFQ which use the same next hop. Due to this, the actual number of out of range packets that are seen by the MAC is nearly the same for all four cases (see Figure 5.12). Figure 5.13 shows the number of out of range packets that are transmitted after prediction. Higher values indicate greater incorrect pre-transmission predictions. The state predictor shows that it is ineffective in predicting out of range next hops since the number of real route breaks are very close to the number of out of range data packets exiting the IFQ. On the other hand, the SNR predictor was able to detect most or nearly all out of range packets using pre-transmission prediction.

The SNR predictor performed poorly because it tends to predict the next hop as out of range more frequently than the state predictor. Furthermore, a large percentage (60-80%) of out of range predictions made by the SNR predictor are incorrect.

Figure 5.14 gives the rate of incorrect out of range predictions, which indicates the rate of false route breaks caused by the prediction scheme since the cost of each incorrect prediction is a false route break. Overall, the SNR predictor looses throughput due to inaccurate out of range predictions.

In summary, we conclude that since AODV does not have too many false or real route breaks prior to

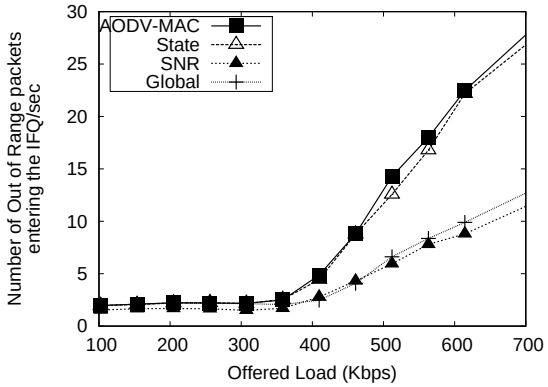

Figure 5.11: AODV pre-transmission prediction: Number of out of range packets entering the IFQ per second (N's out of range packets in Figure 5.8).

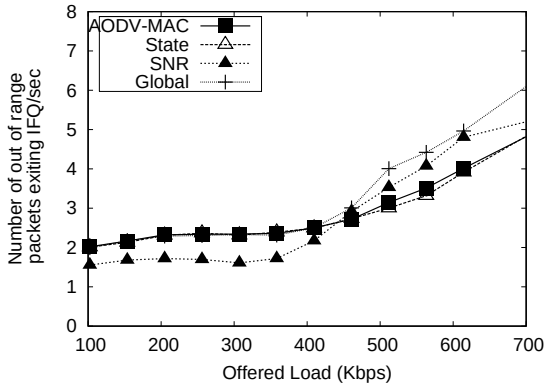

Figure 5.12: AODV pre-transmission prediction: Number of out of range packets exiting the IFQ per second (M's out of range packets in Figure 5.8).

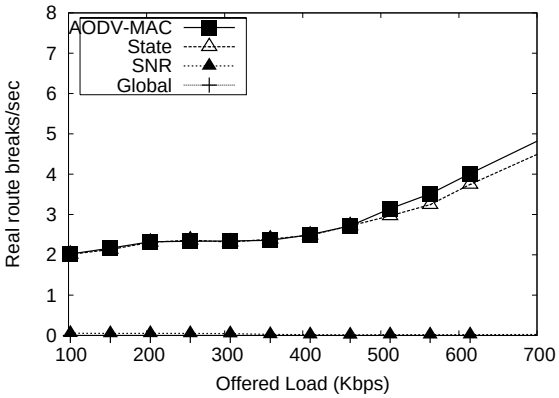

Figure 5.13: AODV pre-transmission prediction: Number of real route breaks per second (out of range packets transmitted per second) (F branch in Figure 5.8).

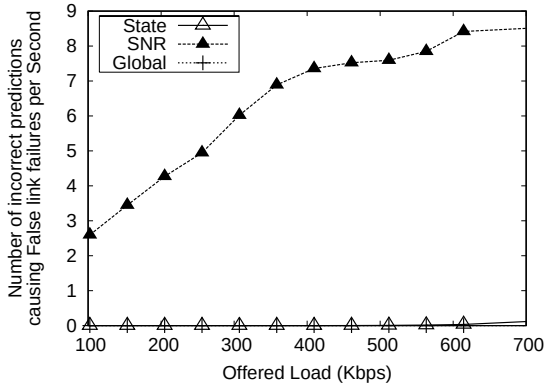

Figure 5.14: AODV pre-transmission prediction: Incorrect out of range predictions per second (false route breaks created by pre-transmission predictions per second). D branch in Figure 5.8.

saturation, a pre-transmission prediction, offers very little benefit over the 802.11 MAC protocol. Any improvement beyond saturation is solely due to the net reduction of false route breaks, which may be achieved by post-transmission prediction schemes. The global prediction that is depicted here confirms that pre-transmission prediction does not improve the throughput, even with global knowledge of the nodes' locations.

DSR: Figure 5.15 shows DSR's performance with and without pre-transmission prediction (Figure 5.16 shows RREQs rate). With the exception of the distance predictor, all other predictors performed well beyond

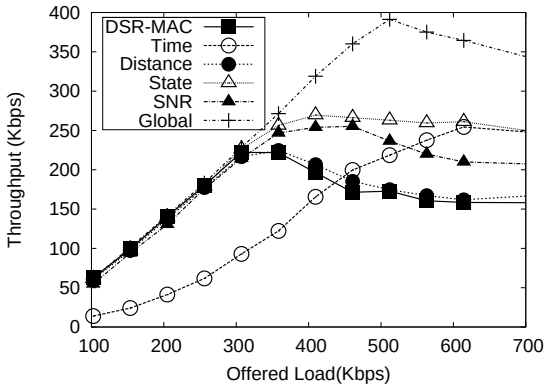

Figure 5.15: DSR pre-transmission prediction schemes in MANET.

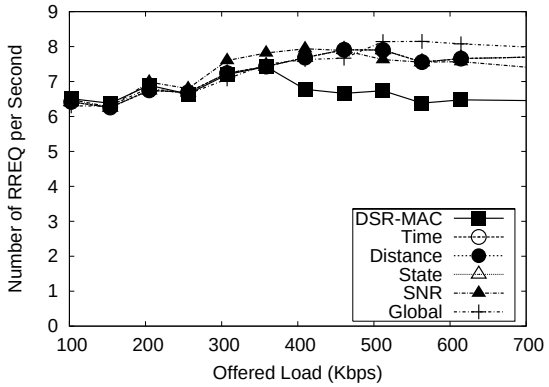

Figure 5.16: DSR pre-transmission prediction route request per second.

saturation. The time predictor does not work well for low loads because it exacerbates the stale route problem by predicting lost-next-hops as in range. Otherwise, the predictors do not have an impact prior to saturation. Global predictors show a 77% higher peak throughput than the original 802.11 MAC protocol, denoted by DSR-MAC. This is an indication that an accurate pre-transmission predictor can significantly improve DSR's performance. The presence of a large number of out of range transmissions (see real route breaks in Figure 5.7) is the primary reason for DSR's performance improvement with pre-transmission prediction. When there is a large number of real route breaks, there are a lot of packets that could take branch A in Figure 5.8. State, SNR, and global predictors are selected for further analysis. Figure 5.17 shows the rate of out of range packets exiting the IFQ. This is the number of transmission failures if transmitted without prediction. Figure 5.18 shows the number of transmission attempts made to out of range next hops after predictions.

For high loads, the state predictor is able to reduce the number of real route breaks by 75% and the SNR predictor by about 45%. This explains the performance difference between the SNR and the state predictors for high loads. The reduction in real route breaks enables pre-transmission predictors to perform well beyond saturation. In particular, the state predictor improves DSR's performance by 50% at 700 Kbps, and retains nearly all of its peak throughput, as opposed to only 70% retained by the 802.11 MAC protocol.

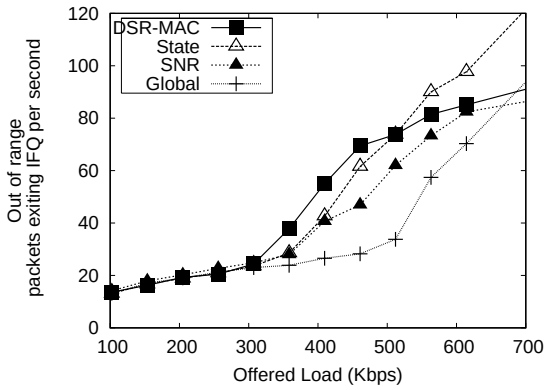

Figure 5.17: DSR pre-transmission prediction: Out of range packets exiting the IFQ per second. (M's out of range packets in Figure 5.8.)

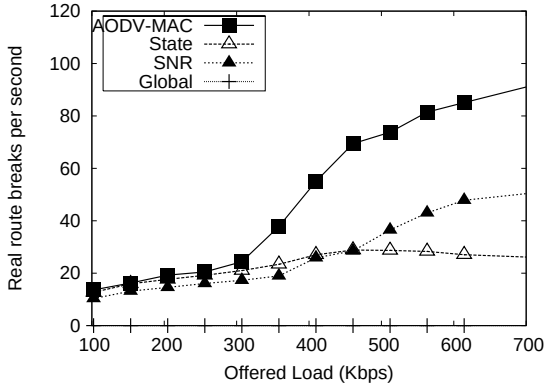

Figure 5.18: DSR pre-transmission prediction: Real route breaks per second using prediction (out of range transmissions attempted per second). (Branch F in Figure 5.8.)

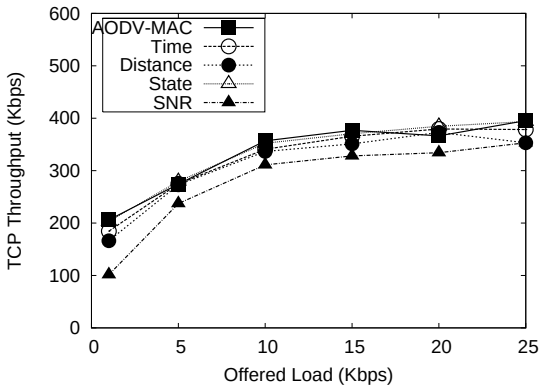

Figure 5.19: AODV pre-transmission prediction scheme: TCP performance.

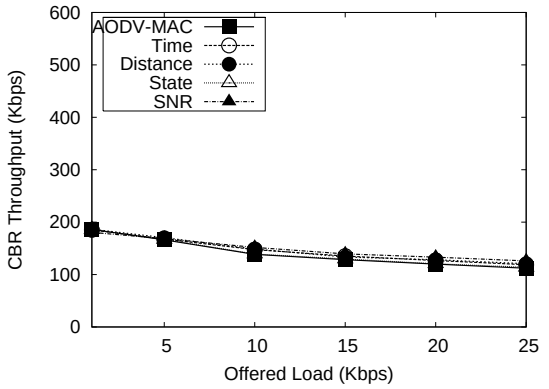

Figure 5.20: AODV pre-transmission prediction: 200 Kbps CBR background noise achieved throughput.

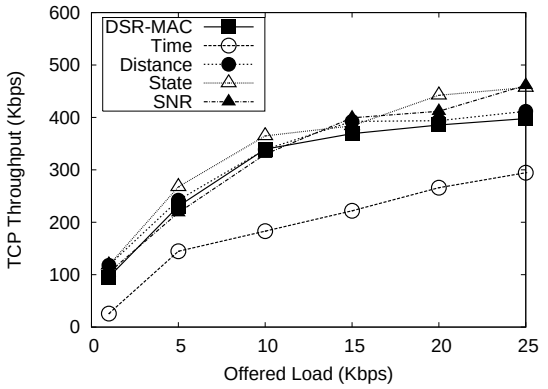

Figure 5.21: DSR pre-transmission prediction: TCP performance.

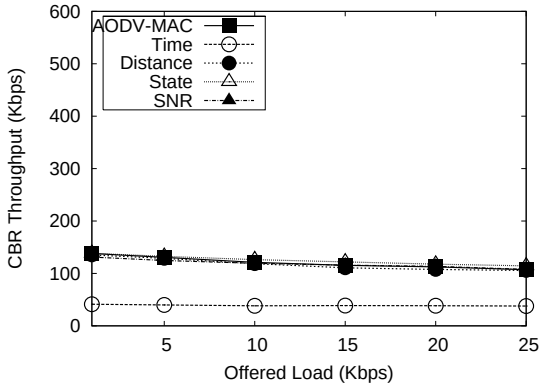

Figure 5.22: DSR pre-transmission prediction scheme: 200 Kbps CBR background traffic achieved throughput.

Table 5.2: Post-transmission prediction model summary.

	Predict next hop in range	Predict next hop out of range
Correct prediction	Silently drop data Benefit: reduced false route breaks	Send false route break (same as 802.11 MAC
Incorrect prediction	Silently drop Data Cost: Real route detection is delayed; some BW is wasted by the data packets that continue to use the broken link	Send false route break (same as 802.11 MAC)

TCP performance using pre-transmission prediction

To evaluate the pre-transmission prediction using a more realistic network load, TCP traffic (using FTP) along with CBR background traffic was simulated. Figures 5.19 and 5.20 show the achieved TCP throughput and the background CBR throughput for AODV and Figure 5.21 and Figure 5.22 give the same information for DSR.

AODV using pre-transmission does not show any significant loss or gain using TCP. For DSR, the state predictor performed well (up to 20% higher throughput). However, the time predictor performed poorly due to the TCP congestion back-off mechanism. The time predictor would take 5 seconds to detect a route break; a wait period of this length can cause a TCP sender to timeout several times and go into a long back-off period. The problem is worse in DSR since it attempts to salvage packets, which exacerbates the timeout period problem.

5.3.3 Performance of post-transmission prediction

To evaluate the impact of post-transmission prediction, the four basic prediction schemes are simulated using the baseline MANET with CBR traffic. A summary of post-transmission prediction costs and benefits is provided in Table 5.2. Figure 5.23 is an expanded version of Figure 5.4, indicating the cost and the benefit of post-transmission prediction.

Post-transmission prediction improves performance by maximizing the instances in which branch E

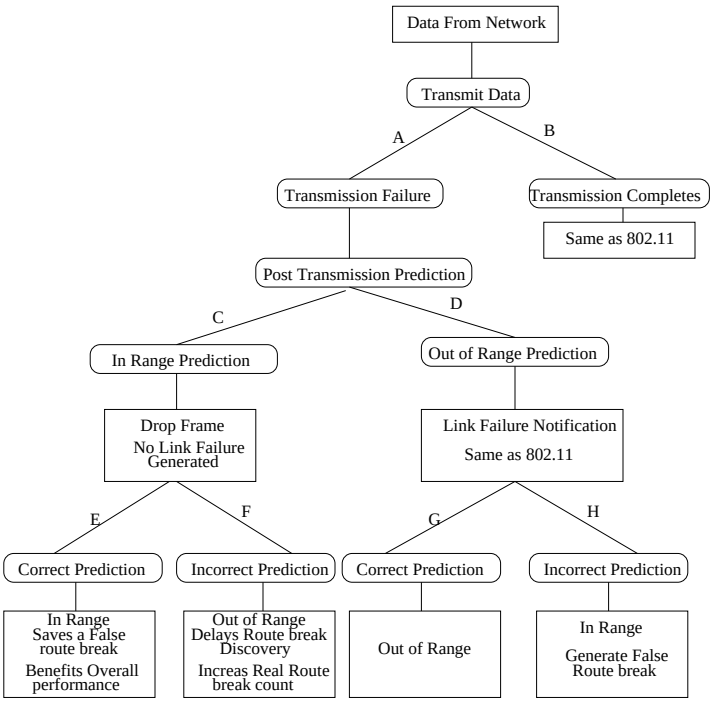

Figure 5.23: Post-transmission prediction flow diagram with cost and benefit.

is traversed and minimizing the instances in which branch F is traversed. Branch E represents correctly predicted false route breaks, and branch F represents delays in route break discoveries. Frames taking branch D have no significant effect on the performance relative to the 802.11 MAC protocol. In fact, without prediction, all failed transmissions require MAC to take branch D. We will now analyze the impact of post-transmission prediction on AODV's and DSR's performance.

AODV: Figure 5.24 shows CBR throughput for the MANET simulation for the four prediction schemes using post-transmission prediction in conjunction with the AODV routing protocol. Using AODV, all four prediction schemes sustain peak throughput beyond the saturation point. Compared to the original 802.11 MAC protocol, all four prediction schemes retain approximately 95% of their peak throughputs, whereas the original 802.11 MAC protocol retains only 62% of its peak throughput. With the exception of the time predictor, there is a small gain in peak throughput (approximately 7%) for all prediction techniques when compared to the 802.11 MAC protocol.

Performance gains from post-prediction are achieved by reducing false route breaks. Figure 5.25 shows the number of false route breaks sent per second in the 802.11 MAC protocol and in the prediction schemes

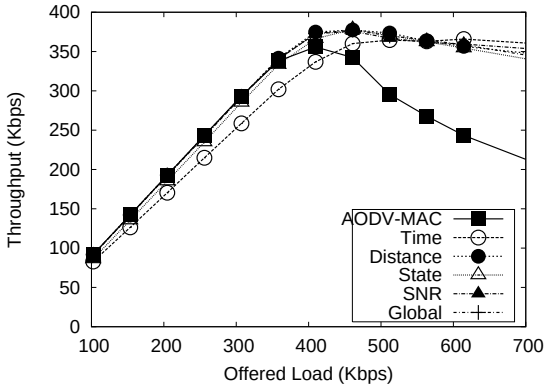

Figure 5.24: AODV post-transmission prediction schemes in MANET.

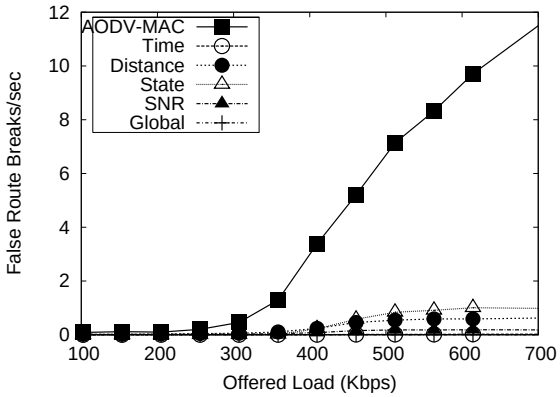

Figure 5.25: AODV false route breaks in post-transmission predictions. (Branch H in Figure 5.23.)

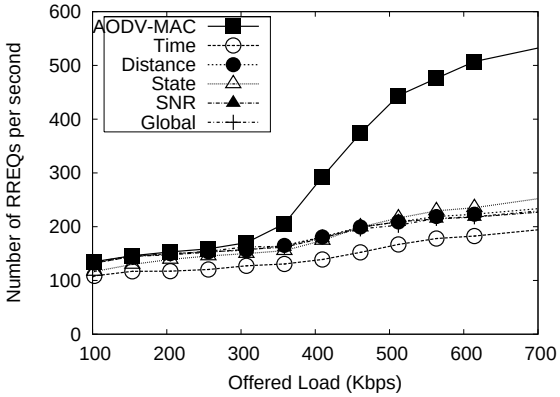

Figure 5.26: AODV post-transmission prediction RREQs transmitted in MANET.

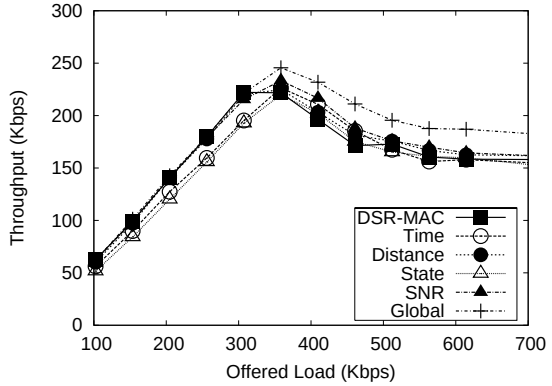

Figure 5.27: DSR post-transmission prediction schemes in MANET.

used. Time prediction is able to reduce the number of false route breaks most significantly, but it loses performance at low loads because it selects most out of range packets as in range. As a result, there is an increase in the number of stale routes, which results in loss of throughput in low loads. We also verify it by examining the number of RREQs reduced by the prediction schemes. This data is plotted in Figure 5.26.

DSR: Figure 5.27 shows DSR's post-transmission prediction performance. None of the prediction schemes show significant improvement. Global prediction shows a 10-15% improvement for high loads; this is an indication that if the prediction scheme is very accurate, post-transmission prediction can slightly improve DSR's performance. Figure 5.28 shows the number of false route breaks in the 802.11 MAC protocol and the prediction schemes simulated. The prediction schemes are able to reduce the number of false route

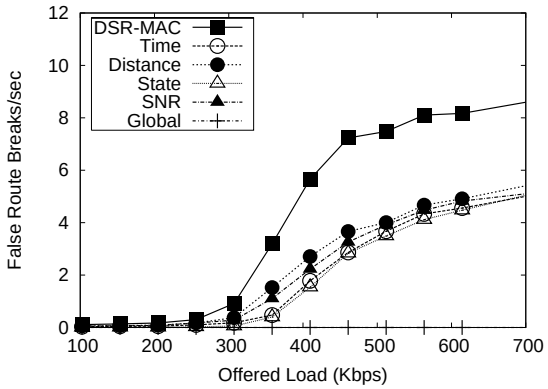

Figure 5.28: DSR post-transmission prediction false route breaks.

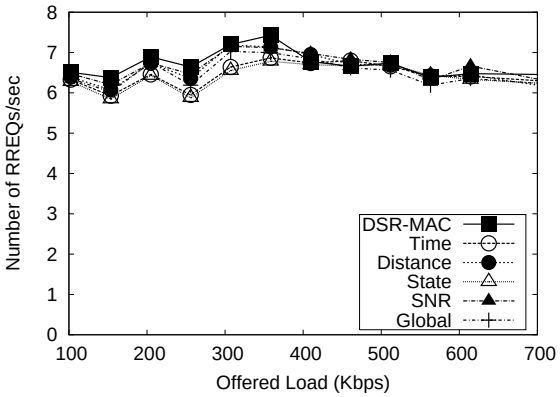

Figure 5.29: DSR post-transmission prediction: RREQs transmitted in MANET.

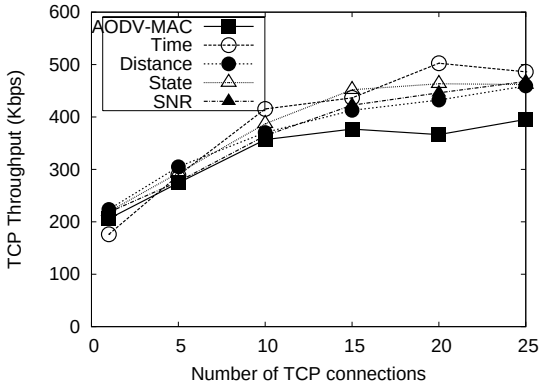

Figure 5.30: AODV post-transmission prediction scheme: TCP performance.

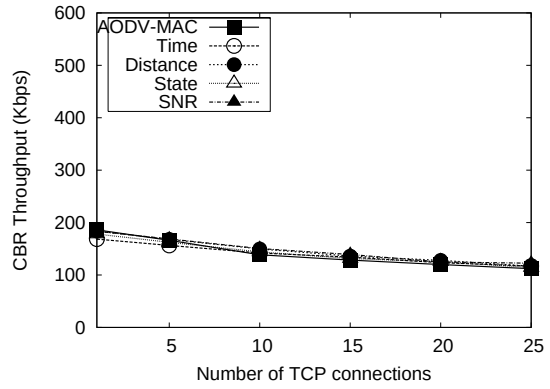

Figure 5.31: AODV post-transmission prediction: 200 Kbps CBR background traffic achieved throughput.

breaks by approximately 50% for high loads, but there is no performance benefit because false route breaks do not cause any significant problems for DSR, which is able to repair broken routes with a lower number of RREQs. Since, post-prediction schemes do not reduce RREQs in DSR significantly (see Figure 5.29), post-transmission prediction is of little benefit when DSR is used as the routing protocol.

TCP performance

We evaluated the post-transmission predictor for TCP traffic as well. The simulation setup is the same as the setup for pre-transmission prediction of the TCP evaluation. Figure 5.30 shows the TCP performance (Figure 5.31 shows the background traffic performance) for post-transmission prediction in AODV. All of the predictors performed better than the 802.11 MAC protocol, with the time predictor improving the

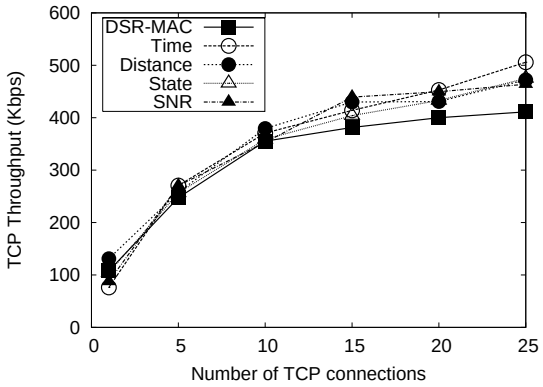

Figure 5.32: DSR post-transmission prediction: TCP performance.

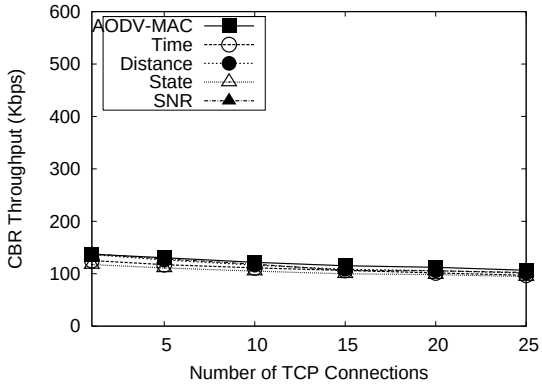

Figure 5.33: DSR post-transmission prediction: 200 Kbps CBR background traffic achieved throughput.

performance by up to 33%.

Figure 5.32 shows the DSR's TCP performance using post-transmission prediction (Figure 5.33 shows background traffic). Even though DSR's CBR simulations did not show a significant performance increase, TCP communication shows a gain of up to 25% in achieved throughput.

5.3.4 Cost of pre- and post-transmission predictors

In pre-transmission prediction, a prediction is done for each packet placed in the IFQ, so the pre-transmission predictor is used extensively. The distance and SNR predictors will require a considerable amount of CPU cycles to predict the next hop's status. They also require more data to be tracked and compared than the time and the state predictors do. On the other hand, the time and state predictors require only a few CPU cycles, and the required data is easily maintained. Therefore, for pre-transmission prediction, we believe the time and state predictors should be considered first. Post-transmission prediction needs to make a prediction only for failed transmissions. Therefore, the cost of the post-transmission prediction is low, even for high maintenance predictors such as the distance and SNR predictors.

Overall, the state and time predictors are the simplest predictors to implement and have a lower computational overhead when compared to the other two predictors, distance and SNR.

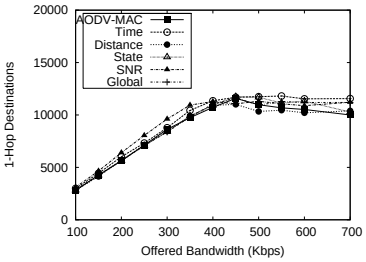

Figure 5.34: Pre-transmission prediction and CBR: Total number of packets traveled 1-hop to reach the final destination.

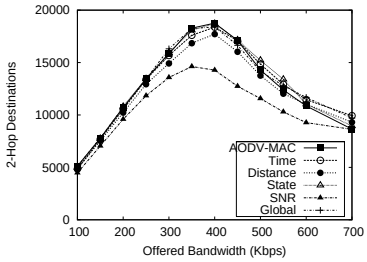

Figure 5.35: Pre-transmission prediction and CBR: Total number of packets traveled 2-hop to reach the final destination.

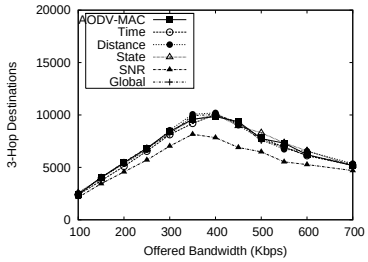

Figure 5.36: Pre-transmission prediction and CBR: Total number of packets traveled 3-hop to reach the final destination.

5.3.5 Fairness in pre- and post-transmission prediction

To evaluate whether pre- and post-transmission predictions favor shorter routes over longer routes, we examine the number of packets that traveled 1, 2 and 3 hops to reach their destinations for the AODV routing protocol. Figures 5.34, 5.35, and 5.36 show the total number of packets that traveled 1, 2 and 3 hops to reach the final destination using pre-transmission prediction, respectively. These figures do not show a significant change in the number of packets delivered compared to the original MAC protocol. Therefore, we can conclude that pre-transmission prediction does not alter fairness for the AODV protocol. Figures 5.37, 5.38, and 5.39 show the total number of packets that traveled 1, 2 and 3 hops, respectively, to reach the final destination using post-transmission prediction for AODV. These figures indicate that all prediction schemes deliver more packets under network saturation than the original 802.11 MAC protocol, indicating that there is no visible sign of prediction schemes favoring shorter routes to achieve higher throughput. Further, we calculated Jain's fairness index for pre- and post-prediction schemes. Figure 5.40 and Figure 5.41 show the fairness of pre-transmission prediction using DSR and post-transmission prediction using AODV, respectively. The figures show, with the exception of the time predictor in pre-transmission prediction, that there is no significant loss of fairness by using prediction schemes.

5.4 Adaptive Prediction

Our analysis indicates that prediction can help when the MANET is in saturation, but some predictors degrade the performance when the MANET is not saturated. In this section, we design a simple rule to

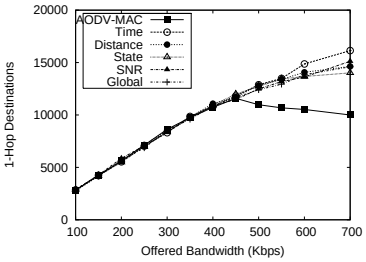

Figure 5.37: Post-transmission prediction and CBR: Total number of packets traveled 1-hop to reach the final destination.

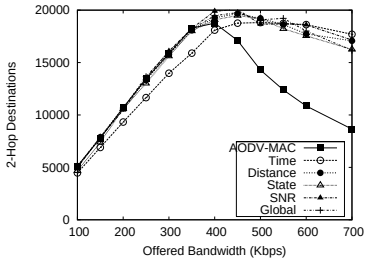

Figure 5.38: Post-transmission prediction and CBR: Total number of packets traveled 2-hop to reach the final destination.

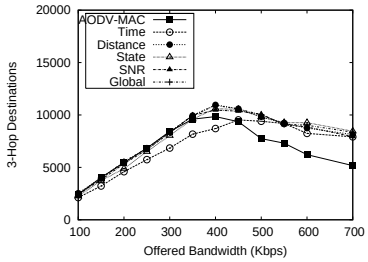

Figure 5.39: Post-transmission prediction and CBR: Total number of packets traveled 3-hop to reach the final destination.

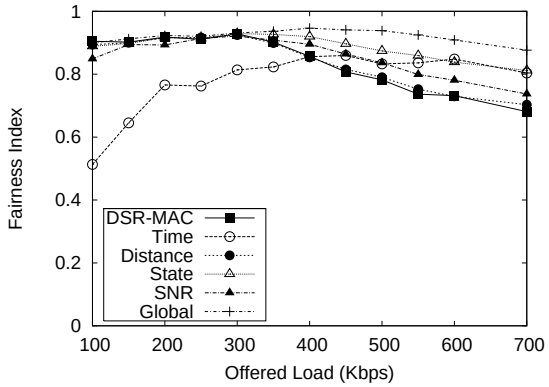

Figure 5.40: Jain's fairness index for pre-transmission prediction using DSR.

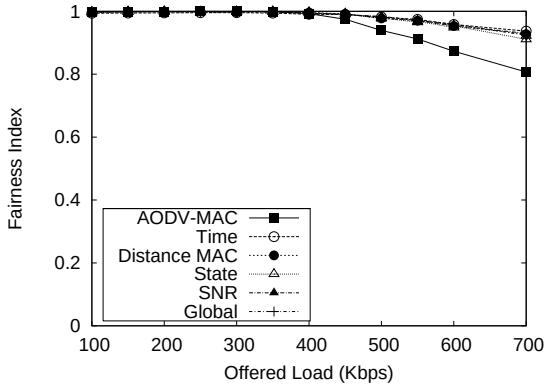

Figure 5.41: Jain's fairness index for post-transmission prediction using AODV.

recognize whether the network is in saturation or not. If the network is in saturation, we apply the chosen prediction technique; otherwise, no prediction is applied.

The channel status captured by the radio layer can be used to find a node's radio channel idle period. The idle period can be calculated as a fraction of the total sample time. If the idle time fraction is less than a preset value, then the network is near saturation; otherwise, it is not congested. To avoid a node moving in and out of prediction and non-prediction mode, the percentage of times prediction is used is computed as:

$$\frac{\text{Total number times prediction is applied}}{\text{Total number of times checked for prediction}} \times 100$$

If this prediction percentage is greater than the preset value denoted by $P_{percentage}$, then the prediction is applied to all data packets sent by that node. Preliminary experiments indicate that when the radio layer is 25% or less idle, prediction schemes can retain the full bandwidth in unsaturated loads and sustain the

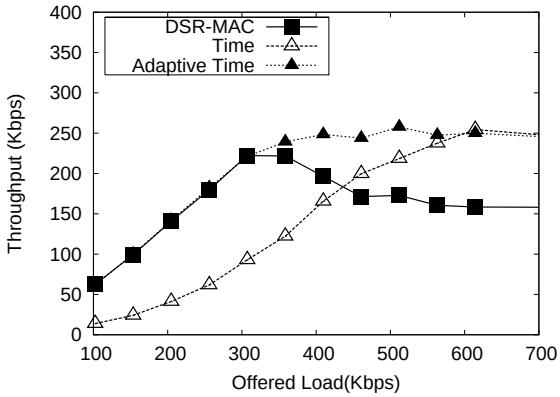

Figure 5.42: Adaptive time prediction using pre-transmission prediction and using DSR as MANET routing protocol.

throughput beyond saturation.

To demonstrate that adaptive prediction improves the performance of a congested network without impacting its low-load performance, we use the time predictor. As an example, we chose the time-based along with pre-transmission predictor for DSR. Figure 5.42 shows the performance of the adaptive time predictor. It performs as well as the original MAC prior to saturation, and sustains the peak throughput beyond saturation.

5.5 Conclusion

Transmitting to a nonexistent next hop or falsely concluding a non-responsive next hop as out of range degrades the MANET's performance. Therefore, we investigated next hop status prediction schemes to mitigate these problems. We proposed two types of prediction strategies: pre- and post-transmission prediction. The former reduces the real route breaks, and the latter reduces the false route breaks. We proposed four basic prediction mechanisms: time, state, distance and signal-to-noise ratio (SNR) predictors. Pre-transmission prediction reduces the impact of real route breaks and, thus, improves the achieved throughput. Post-transmission prediction reduces false route breaks, which reduces the routing overhead. This reduction in routing overhead results in higher throughput.

The performance of post- and pre-transmission predictors for CBR traffic shows that routing protocols with a large number of real route breaks can benefit from pre-transmission prediction and the routing

protocols with a large number of false route breaks can benefit from post-transmission prediction. TCP simulations show that the MANETs can generally benefit from both pre- and post-transmission predictors. We observed this with both DSR and AODV. In the presence of TCP communications, a MANET operates near saturation. Therefore, both pre- and post-transmission predictors show a significant benefit.

Some of these predictors performed well in high loads, but performed poorly for low-loads. To overcome the underperformance, we introduced adaptive prediction schemes so that a predictor is used only when the network is congested. Finally, we also showed that prediction schemes improve network fairness.

Chapter 6

Experimental Evaluation of Channel State Dependent Scheduling

Acknowledgement-based wireless MAC protocols (such as the 802.11 MAC protocol) may cause head of the line blocking if the packet at the head of the IFQ is targeted to a host with a weak link (a link in the error state) [9]. This is because several retransmission attempts may be necessary for the packet to be successfully transmitted, if it is transmitted at all. On the other hand, the packets waiting in the outgoing queue may be destined to nodes with a strong link quality, and, thus, would have a much higher probability of successful transmission on the first attempt. Therefore, packet latency and channel utilization can be improved if packets with strong link quality to the destination are given priority, instead of following a strict First-Come-First-Served (FCFS) discipline. Several scheduling schemes were investigated and performance was evaluated via discrete event simulations. Improvements were significant. In a recent work, scheduling issues at the radio interface are formalized and a new fairness model for wireless packet networks is proposed to handle location-dependent error bursts [40]. In contrast, our approach is purely experimental. The goal is to implement and evaluate a simple channel state dependent scheduling (CSDS) mechanism on a wireless LAN testbed in an ad hoc setting. The commercially available wireless interfaces did not use the acknowledgment-based link-layer protocol to its fullest at the time of this work. Acknowledgements are normally implemented purely in hardware, and it would not be practical for it to interact with the scheduling scheme in an efficient fashion. Thus we take a slightly different approach in our protocol, which is based on a channel sensing mechanism. Also, unlike the simulation study in [9], our focus is on the reliability and

the performance of such scheduling mechanisms in wireless networking hardware and operating systems.

The remainder of the chapter is organized as follows: In Section 6.1, we describe the hardware and software details of the wireless LAN testbed used in the experiment. In Section 6.2, we evaluate the error characteristics of the wireless LAN that form the basis of the protocol. Section 6.3 develops the channel state dependent protocol. In Section 6.4, we discuss a trace-based channel modulation technique used in the experiments. Section 6.5 presents and analyzes all experimental data, followed by the conclusions in Section 6.6.

6.1 Wireless Testbed

The wireless testbed consists of several 200 MHz, 32 MB Pentium Pro desktop PCs (Dell XPS series) and 120 MHz, 24 MB Pentium laptops, running the Linux operating system version 2.0.31. The wireless interfaces used are Lucent Technology's Wavelan ISA (for desktop) and PCMCIA (for laptop) card [18]. Each card contains a LAN controller, modem control unit and a radio transceiver and is attached to a small external unit that houses the antenna [18]. Our Wavelan system operates in the 2.419 - 2.445 GHz ISM (industrial-scientific-military) license-free band at a nominal bit rate of 2 Mbits/sec. Wavelan employs Direct Sequence Spread Spectrum (DSSS) technology for its superior resistance to multipath fading and interference from other transmissions in the same frequency band. It also uses antenna diversity for improved tolerance against multipath fading. Wavelan employs a low-power radio (transmit power about 500 milliwatts) and is primarily targeted for in-building wireless extension of an existing Ethernet LAN. Wavelan-to-Ethernet bridges are available for a seamless extension. We, however, use Wavelan as a pure wireless LAN, instead of just an extension to an Ethernet LAN.

As it is technically difficult to detect collisions in a radio environment [70], Wavelan employs a CSMA/CA (carrier-sense multiple access, with collision avoidance) MAC protocol, which is a precursor of the recent IEEE standard 802.11 [20]. More specifically, Wavelan does not use the RTS/CTS handshake or ACK frames to acknowledge the successful reception of DATA frames by receiving nodes. Briefly, CSMA/CA tries to reduce the possibility of collision by treating a busy medium as a collision. If a station has an outstanding packet when the medium is busy, it will delay transmission for a random period after the medium

is sensed as continuously idle for an IFS (inter-frame spacing) period. This technique does not eliminate the collisions, but merely avoids them. Collisions are still possible, if, in spite of the randomness in transmission scheduling, two stations happen to transmit at the same time. Hidden terminal scenarios may also cause collisions. The sending station is never aware of any collision. The receiving node may be able to capture [44] at most one of the colliding packets depending on the signal strengths of the colliding packets. If packets are lost due to collisions, it is assumed that the higher layer protocols will kick in to perform any necessary retransmissions . Wavelan does not employ the power control, frequency, or code diversity common in many CDMA based cellular systems. It is, thus, suitable for use in an ad hoc or infrastructure-less environment, where there is no stationary base station with specific routing responsibilities.

Wavelan's modem control unit reports certain quantities related to the channel state upon arrival of each packet. One of them is a 6-bit quantity called signal level, and it is derived from the receiver's automatic gain control (AGC) setting just after the beginning of the packet. This quantity can be read after the reception of each packet from the card's parameter storage area (PSA) by a software probe inserted in the device driver. In measurements reported in the literature [28], the signal level was found to be a significant determinant of whether or not a packet could be received reliably in a Wavelan system. This observation is also supported by our independent measurements reported in the next section. The signal level drops with distance as well as when the signal passes through obstacles such as walls. In addition to signal level, the modem control unit reports two other quantities *signal quality* and *noise level*, which are not useful in our experiments.

On the software side, the IP queue hands over packets to the Wavelan device driver if the packets are for the wireless interface. The device driver checks whether or not the interface is busy. If it is not busy, it copies the packet to the interface hardware, sets up a timer to expire after a preset interval, and marks the device to be busy. The interface interrupts the driver after it is successful in sending the packet on the air. We here determined that for the smallest sized packets (60 bytes), it takes about 1.5 ms and 0.6 ms for the PCMCIA and ISA card drivers, respectively, to get this interrupt for a successful send. The driver then resets the timer and the busy flag. If the timer expires before the interrupt comes from the interface, the device driver tries to diagnose the cause of the transmission delay/failure by checking certain status flags on the interface and then resets the hardware as well as the busy flag accordingly.

Table 6.1: Various statistics related to error rates and signal levels in the five experiments.

Experiment Number	1	2	3	4	5
Mean signal level	12.14	10.84	8.01	7.92	7.68
Standard deviation of signal level	0.73	0.89	0.73	0.72	0.65
Error rate (%)	0.06	0.33	7.18	23.40	44.21
Mean run-length of packets correctly received	1677	586	19	8.9	4.9
Mean run-length of packets dropped	1.07	1.94	1.47	2.63	3.89

6.2 Signal Level Characteristics

In Wavelan, the signal level is a number between [0,63], but in actual measurements this number is rarely found to be outside the range of [4,40]. A sequence of experiments was done using two laptops to study the time variation of the signal level in a typical office/laboratory environment. This forms a background for our implementation of channel state dependent scheduling. More than one hundred thousand back-to-back UDP packets (1 KB long) are sent from one host to the other and the signal level for each received packet (complete or damaged) is recorded. Care is exercised, such that only path loss behavior between the two hosts influences the signal level and there are no extraneous sources of packet loss, such as buffer overflows or any significant interference from other radio signals, including from other Wavelan modems. In the five experiments reported here, the distance between the hosts is gradually increased and/or more obstacles are placed in between them to increase signal path loss. Experiments were performed in a typical university laboratory environment with people moving about and a large number of computers and monitors around. Thus, there were sources of both radiation and multipath fading.

Mean statistical data related to the signal level and error rate from the five experiments is shown in Table 6.1. Histograms of the signal level are presented in Figure 6.1. The experiments show, not surprisingly, that there is a very strong correlation between the signal level and the error rate. A signal level of roughly 10 is found to be sufficient for the receiver to receive packets with very low error rates. When the signal level falls below 8, the error rate becomes significantly higher. This experiment is similar to the experiments reported in [28], though they use a different frequency band. Another observation is that the run of dropped packets often lasts over several packets even for a large packet size considered for the experiments (see Figure 6.2).

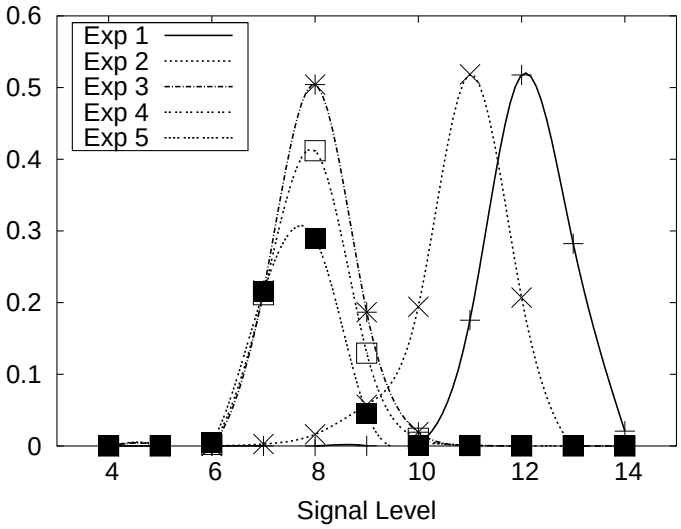

Figure 6.1: Probability mass function of signal levels in the five experiments.

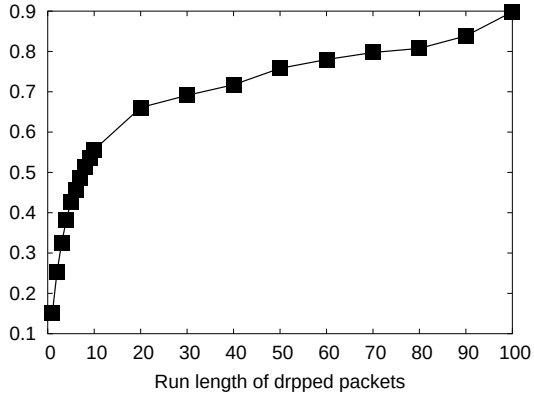

Figure 6.2: Cumulative distribution of various run lengths of dropped packets.

Thus, the error is often bursty.

6.3 Channel State Dependent Scheduler with Channel Sensing

The simple idea behind the scheduler is that when the link quality between two hosts is known to be poor, data packets otherwise ready to be transmitted are buffered at the sender until the link quality improves. This technique has several benefits over the typical FCFS scheduling. First, the channel utilization is better, as packets transmitted over a weak link have a high probability of being dropped and will probably have to be retransmitted by the higher layer protocols. In any case, such packets occupy the channel without

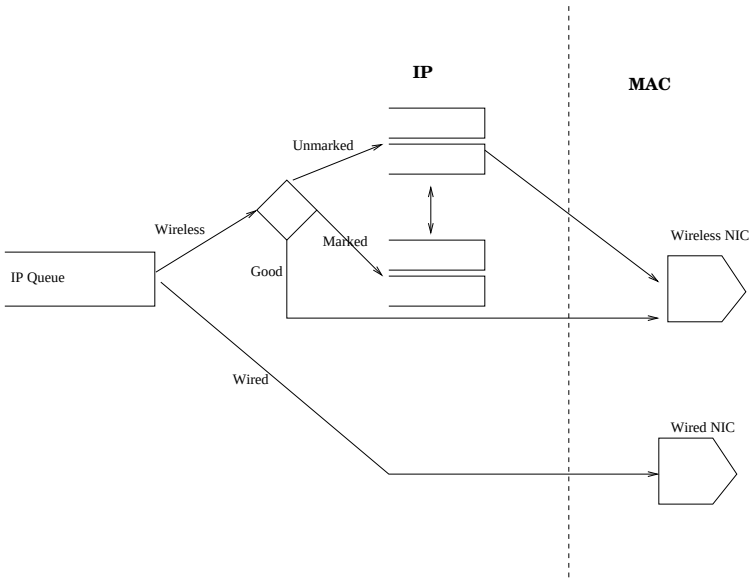

Figure 6.3: Channel state dependent scheduling system showing the different queues in the device driver.

contributing to the received throughput. Secondly, the error rate is better than that of the FCFS scheduling as the buffered packets can be transmitted later, when the link quality improves. Lastly, packets on the strong links, if any, can be transmitted sooner, thus utilizing channel bandwidth, which would otherwise be wasted. However, one problem with such a scheme is that the receiver alone can learn about the link quality by measuring the signal level of a received (correctly or incorrectly) packet, while the sender has to make all scheduling decisions. Thus the link quality information must be appropriately fed back to the sender. Also, a channel sensing mechanism for the weak links (on which transmissions have been stopped by the scheduler) must be implemented to avoid starvation. Our channel state dependent scheduler includes both the feedback and channel sensing mechanisms. The scheduler is implemented as part of the Wavelan device driver (see Figure 6.3). The link to each neighboring node can be in one of three states (also see Figure 6.4):

1. *Good*: The link is known to be in good state. Data packets will be forwarded over this link.

2. *Marked*: The link is weak and packets may not reach the destination reliably. Packets are queued until the link quality improves. If and when this queue grows large enough to occupy all available memory, either packets are dropped from the front of the queue, or the application is blocked from generating more packets.

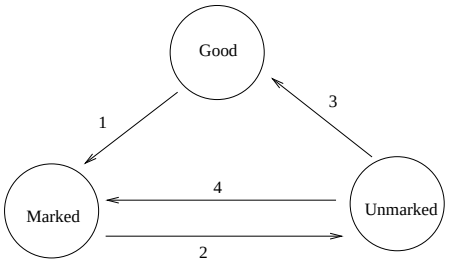

Figure 6.4: State transition diagram of a neighbor node.

3. *Unmarked*: A marked link becomes unmarked when the link quality improves. The queued data packets are immediately sent out. Once the queue is empty, the state is changed to "good". If the link quality degrades while the queue is in an unmarked state, transmission on this link stops and the state is changed back to marked again. The quality of the link is determined via measurements of the signal level. Three thresholds are useful for describing the state machine of the scheduler, the *drop threshold* (the level below which there is a very high probability of the packet being dropped, usually level 8), *high water mark*, and *low water mark*. The high water mark is a little above the drop threshold (usually 1 or 2 levels). The low water mark is a level lower than the drop threshold, below which signal level information is unavailable from the Wavelan interface (usually level 4). Assuming that the state of the link is "good", if the level reaches the drop threshold, the state is changed to "marked" (transition 1 in Figure 6). If subsequently it goes above the high water mark, the state is changed to "unmarked" (transition 2). As mentioned before, the link goes back to a "good state", once the unmarked queue becomes empty (transition 3). It is possible for an unmarked queue to move to marked if the signal level falls below the drop threshold (transition 4).

6.3.1 Channel Sensing

Once a link is in a marked state, the scheduler must be able to detect when the link quality improves. However, the signal level is ordinarily unknown in the absence of any communication on that link. A channel sensing mechanism is developed to avoid starvation. The device driver sends a specially formatted *link quality request* packet to the neighbor on the other end of a marked link. Any host receiving such a packet replies with a *link quality response* packet using the measured signal level of the request packet. These packets are the smallest possible size for Wavelan (60 bytes). These link quality request packets

are sent periodically (about 80 packets/second in our experiments) until the link quality is improved. This amounts to a maximum sensing overhead of 77 Kbits/sec for each weak link, which is a small fraction (less than 4%) of the nominal channel bandwidth of 2 Mbits/sec. Link quality response packets are also sent when any receiving node detects a signal level falling below the high water mark. Note, however, that when there is a 2-way communication, many of these link quality responses can be piggybacked on the data packets.

The protocol is efficient. No additional data copying is necessary. The only overhead is the additional communication for channel sensing. Queuing is implemented in a similar fashion as in the IP queue, using the sk_buff structures in Linux and the related kernel routines to manipulate the queue [26]. Active links are identified via a hashing mechanism on the hardware (MAC) address in a received Wavelan frame. This scheduling protocol is sensitive to a sudden complete loss of signal. If the receiver does not receive any signal from the sender to assess the poor link quality, the sender will be unaware of the broken link and continue sending packets (assuming the link was not already in the "marked" state) until the upper layer protocol detects the problem. A link layer acknowledgement can solve the problem, a solution also adopted in the IEEE 802.11 standard [20]. However, in the absence of such acknowledgments in Wavelan, acknowledgments must be generated by software (in the device driver, for example). A software generated acknowledgement for every data packet was not deemed to be efficient in our testbed and was not adopted.

6.4 Wireless Channel Emulation

Since comparison with the traditional FCFS scheduling is an important part of the evaluation of our scheduling protocol, reproducible behavior on the part of the wireless channel is crucial. However, the idea of reproducibility is contrary to the very nature of wireless channels, as the channel condition can change unpredictably, especially when one or more hosts are mobile. We solve this problem of reproducibility by *emulating* the wireless channel using traces of signal levels collected using independent experiments. Identical emulated channel models are used to drive the experiments for comparative performance evaluations. Such emulation also permitted us to tune various protocol specific parameters, such as the high water mark, periodicity of channel sensing etc. The emulation technique is similar in philosophy to the techniques described in [53], though we are more concerned about modeling "channel" parameters rather than "network"

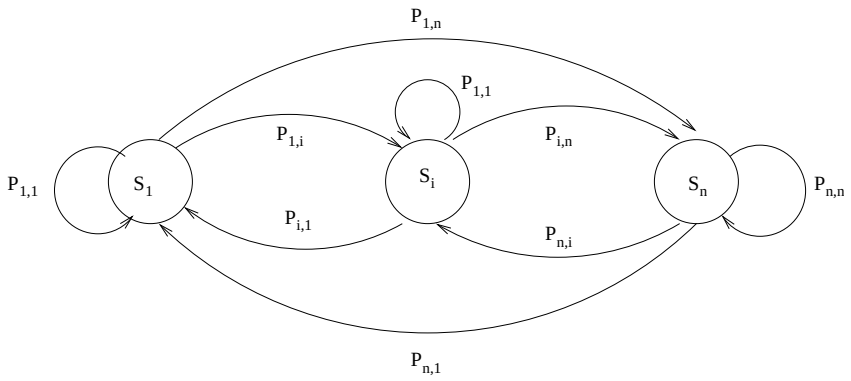

Figure 6.5: Finite-state Markov chain model for channel emulation.

level parameters.

We model the wireless channel between a pair of hosts as a *Finite-state Markov Chain* (FSMC) [72] with n states, S_1 through S_n (see Figure 6.5). Each state corresponds to a small range of signal levels, after partitioning the total range of signal levels obtained into n equal partitions. In our experiments, n=16. In the data collection phase, two laptops are used with unidirectional streams of small UDP packets at a constant rate of about 80 packets/sec. The signal level of each packet received (either reliably or damaged) is recorded along with the previous signal level. At the end of the experiment, the frequency count of every transition of signal levels between each subsequent pair of packets is used to compute a probability of each possible state transition, P_{ij}, $i, j = 1, ..., n$. Note that any state can move to any other state in the chain, including itself. Thus, an $n \times n$ matrix of state transition probabilities $[P_{ij}]$ is obtained that characterizes the wireless channel behavior. A similar model was used in [72] to model a *Rayleigh* faded wireless channel, except that only state transitions between neighboring states or to the same state were used and signal-to-noise ratio (SNR) was used instead of signal levels. This channel model is reproduced in the channel modulation phase that runs concurrently with the actual scheduling experiments. In this phase, the device driver uses an interrupt-driven process to change the state of the wireless channel at the same rate as in the data collection phase with the matrix of state transition probabilities. Any packet transmitted in the experiment piggybacks this state information, which is interpreted as the signal level at the receiver (instead of the actual signal level sensed). All packets received with signal level less than or equal to the drop threshold are dropped at

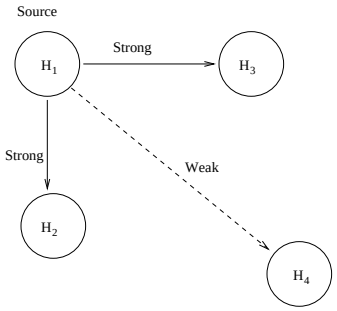

Figure 6.6: Experiment 1 node setup.

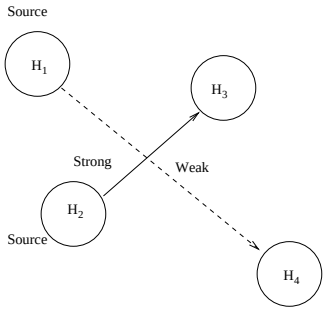

Figure 6.7: Experiment 2 node setup.

the receiver device driver. All other packets are accepted.[1]

6.5 Experimental Evaluation

Experiments are conducted with modulated wireless channels as described above. Using traces collected in a mobile environment where one host is stationary and another is moving (at the speed of a slow walk) around it, often going to the fringe areas where link quality may be poor. However, this data collection phase was carefully monitored so that the link was never completely broken, as then no signal level information can be recorded. During the real experiments, modulated wireless channels are used based on this trace.

Two sets of experiments were designed to demonstrate various benefits of the channel state dependent scheduling. In the first set, one host (source) communicates (unidirectionally) with three other hosts via wireless links (see Figure 6.6). Two of these links are assumed to be strong and are not modulated. The third link is modulated. The source generates UDP packets at a steady rate and sends them to a randomly selected destination. Receivers count the number of packets correctly received to determine the received bandwidth. Experiments were repeated with increasing loads on the network. The results indicate that the weak, modulated link can achieve a larger received bandwidth in the CSDS scheme than in the FCFS

[1]Thus, the Markov Chain has two types of states, the error state and error-free state. In our case, an error (error-free) state has the probability of error = 1 (0). Thus, the error is a deterministic function of state, rather than stochastic. Though, in principle, the chain could be described in terms of only two states, the multiple-state models (sometimes called Hidden Markov Models) such as the one we use here have recently been advocated for a better characterization of the observable error sequence in a physical channel [67].

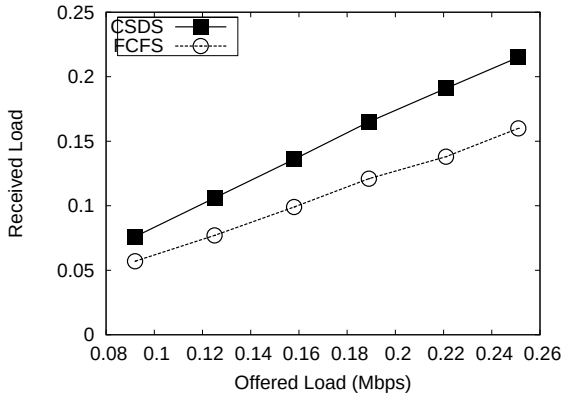

Figure 6.8: Total received bandwidth vs. offered load on the weak link alone in Experiment 1.

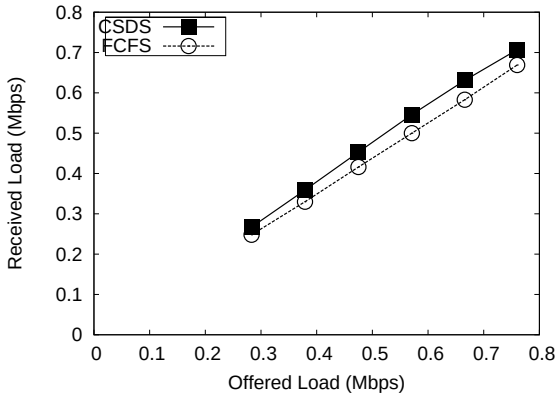

Figure 6.9: Total received bandwidth vs. offered load in Experiment 1.

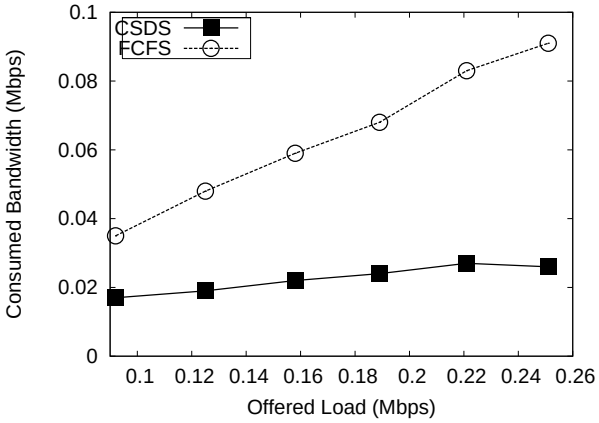

Figure 6.10: Bandwidth consumed by the dropped packets vs. offered load on weak link alone in Experiment 1.

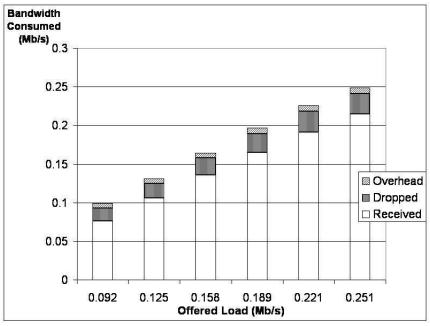

Figure 6.11: Breakdown of the offered load on the weak link in Experiment 1.

(Figure 6.8) and this contributes to a better overall bandwidth usage (Figure 6.9). Focusing on the number of dropped packets and bandwidth consumed by them alone, it is seen that CSDS can save 50-70% of the bandwidth wasted in dropped packets (Figure 6.10).

Figure 6.11 shows the breakdown of the offered load on the weak link. Note that a small portion is wasted in overheads (channel sensing packets) and another portion on dropped packets. The rest is obtained as received bandwidth. Note that the overhead is a very small fraction of the received bandwidth and does not increase with increasing load. This is because the overhead of channel sensing packets is dependent on the state of the channel alone.

The next experiment is set up specifically to demonstrate that the bandwidth savings from CSDS can

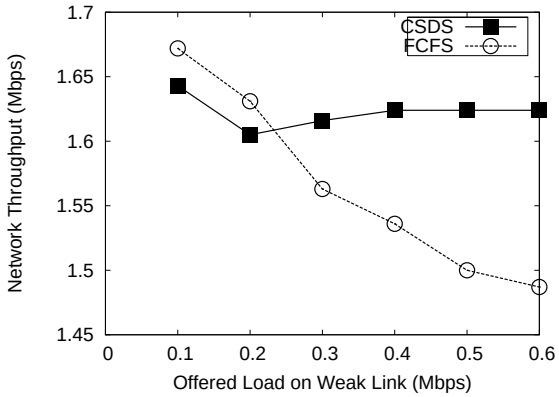

Figure 6.12: Total received bandwidth vs. offered load on weak link in Experiment 2.

be useful to other, independent hosts in a loaded network (see Figure 6.7). In this experiment, we load the wireless channel (strong link quality, not modulated) with a pair of hosts that are communicating at the rate of 1.7 Mbits/sec. A second pair is added to the network that communicates on a weak, modulated wireless channel. All communication is unidirectional UDP. The load on the modulated link is slowly. increased to observe performance. Note that CSDS achieves greater bandwidth utilization and always does better except at a very low load (see Figure 6.12). Poor performance at a very low load is due to channel sensing packets, which consume more bandwidth when compared to bandwidth savings obtained by the CSDS scheme. For higher loads, savings are higher. The savings are primarily due to reduced channel contention caused by packets that are transmitted but cannot be received due to poor link quality.

A third experiment is set up to study the behavior of TCP in CSDS. This experiment is similar to the first experiment except that the FTP transfer of a large file is used as a benchmark and the completion time of the FTP is used as a performance metric. The two strong links are loaded with UDP streams as before. The FTP transfer is set up on the weak link. A different trace was used with a lower drop rate (18% instead of the 37% in the previous trace used in experiments 1 and 2). This is because, with a very high drop rate, TCP was found to hang occasionally for long periods of time, and the experimental data was not stable enough for a meaningful evaluation. In addition, a necessary change is made in the CSDS scheme to reduce packet drops: state changes from a good state to a marked state one signal level above the drop threshold (instead of at the drop threshold). Note that this makes the CSDS scheme very conservative. Figure 6.13 shows the

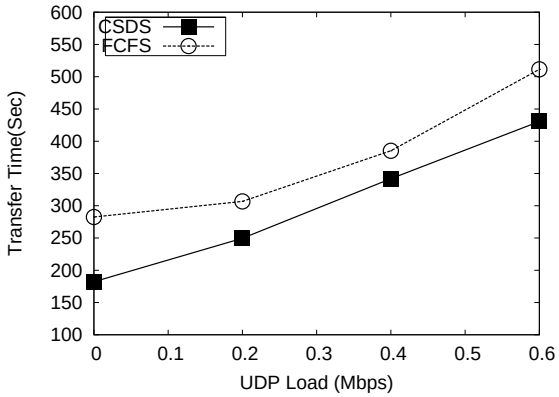

Figure 6.13: FTP transfer time vs. UDP load on the strong link in Experiment 3.

average FTP transfer time for a large (12 Mbytes) text file for an increasing load on the good links. CSDS is observed to be 20% to 50% faster than FCFS.

6.6 Conclusions

The error behavior of a wireless LAN is often unpredictable and bursty. A channel state dependent scheduler that includes a mechanism to monitor the channel quality and accordingly buffer or schedule packets on a link can provide significant benefits. We experimentally demonstrate the benefits, both in terms of lower packet drop rates and higher bandwidth utilization, using an in-building wireless LAN. Unlike prior work in this direction [9], our scheme includes a channel sensing mechanism, where a weak link is periodically sampled to determine when it comes out of the error state. In this work, we also develop a trace-based channel modulation technique to achieve reproducibility of experiments.

Chapter 7

Conclusions

An ad hoc network is a collection of mobile nodes moving in seemingly random patterns and communicating with each other using wireless links without the aid of an infrastructure. For a node to communicate with its neighboring node, there must be an efficient and reliable medium access control protocol. For a node to communicate with another node that is out of its communication range, there must be a dynamic routing protocol that finds a route from the source node to the destination node must be through the other nodes in the MANET. This routing protocol must adapt to various network conditions, have low overhead, and should provide routes promptly.

The IEEE 802.11 protocol is a commonly used wireless MAC protocol for communication between two nodes sharing a wireless medium. The 802.11 MAC protocol is designed to work in infrastructure (wireless LAN) and in infrastructure less (ad hoc) wireless networks. In an infrastructure wireless network, there is a centralized authority to manage the network, therefore, it rarely increases the network load beyond the available capacity. On the other hand, ad hoc networks will frequently attempt to work in loads that are beyond the MANET's capacity. There are two primary reasons for this: (a) there is no centralized node to control the load injected by nodes; (b) the MANET's available bandwidth is only a small fraction of the point-to-point bandwidth. Therefore, users, accustomed to wired-network services, expect a higher bandwidth than what the MANET can offer. Despite the fact that MANETs are prone to be in saturation, there is only a handful of research that studies the MANET under high loads.

In this dissertation, we have studied in detail the IEEE 802.11 MAC protocol and several on-demand routing protocols under high loads. We have identified several weaknesses in the 802.11 MAC and ad

hoc routing protocols. We have proposed several modifications to MAC and routing protocols to mitigate these weaknesses. Using simulations, we have shown that our proposed modifications can improve the peak throughput and sustain it beyond saturation. The contributions of this dissertation are summarized below.

7.1 Impact of Noise Sensitivity on Overall Network Performance

The IEEE 802.11 Medium Access Control (MAC) layer plays a crucial role in the overall performance of an ad hoc network. In Chapter 3, we have shown that the physical carrier sense (noise for distance communication) mechanism as designed and used in 802.11 has a crippling effect on distant but competing transmissions. We proposed a simple modification to the 802.11 MAC protocol to mitigate this problem and showed that this modification benefits ad hoc wireless networks with stationary and mobile nodes.

Communication regions

Using simulations and analytical models, we show that, with respect to a transmission, a receiving node can be in one of the four regions: Communication, Sensing, Noise and Non-interfering regions. A node in the Communication region of a sending node can hear the latter's transmission and can receive the same successfully. A node in the Sensing region cannot hear the transmission, but the noise level is high enough that it senses a busy medium and is prevented from using the channel. In the Noise region, the noise generated by a distant transmission is not strong enough for the node to detect a busy medium. In the Non-interfering region, transmission noise has completely dissipated. The sensing region contains a collision region, in which a node's trasmission can collide with a distant trasmission. The distance of the Collision region varies based on the competing nodes' transmission distances, while the Communication, Sensing, and Non-interfering regions depend only on the distance from the transmitting node.

Reducing the impact of competing transmissions

Using two pairs of communicating nodes, we have shown that when the communicating nodes are as far apart as 688 m (nearly 2 times the communication range), one communication can have a crippling effect on the other. The reason for poor performance is, a node in the Sensing region becomes an exposed node

and cannot respond to its partner's RTS, even though it can safely do so, since it is not going to impact the other node's transmission. Our observation is that if a node can receive a RTS from a node that intends to send data, then it must be able to receive a larger data frame after the RTS/CTS exchange.

Therefore, we have proposed the following modification to the 802.11 MAC protocol: If a node receives a RTS, the virtual sense is idle and the noise level is less than that of a receiving signal level threshold, it can send a CTS in response to the received RTS. This modification improves the link stability in two possible ways: (a) false link breaks are reduced, which reduces the control overhead, especially in on-demand protocols which use network-wide floods to repair broken routes, and (b) links are used more productively.

Performance

We analyzed the performance benefits of the proposed modification using the previously mentioned 4-node setup, a grid of stationary wireless nodes, and a mobile ad hoc network. In the 4-node setup, both sets of nodes are able to communicate without interfering with each other as long as they are not in each other's collision region. This is a considerable gain in performance over the standard 802.11 MAC protocol. The proposed modification also improved the performance of static chain and static grid networks by 35% and 50%, respectively. For a 100-node MANET with 50 CBR communications and AODV routing protocol, the proposed modification improved the CBR throughput by up to 30%. The proposed modification also improved the end-to-end packet delay, hop count, and delivery rate. Using extensive simulations, we showed that the proposed modification can be beneficial for various routing protocols and for both UDP and TCP traffic.

7.2 Performance Under Traffi c Overload

Most studies on routing protocols focus only on the performance prior to network saturation, but MANETs are likely to operate under high traffic loads since there is no centralized admission control. Therefore, we have examined the throughputs of MANETs under heavy traffic beyond saturation. Using 100 nodes with 50 CBR connections generating UDP traffic, we evaluated the performance of the popular AODV routing

protocol, with loads varying from 100 to 900 Kbps. Our simulations indicated that, at high offered loads, the network is only able to retain one-half of its peak throughput. An analysis of the control overheads revealed that the sharp drop-off in the throughput is due to the AODV routing protocol generating far too many route discoveries in response to false route breaks caused by exposed nodes.

Behavior of MANETs beyond saturation

To understand the rapid loss of throughput for loads beyond saturation, we examined and found that there is a sharp increase in the number of RREQs transmitted and in the number of RTS attempts made in the MANET beyond saturation. The increase in RREQs is unexpected, as the mobility remains the same for all offered loads. An examination of the network-to-MAC interface queue (IFQ) of a congested node indicated that the queue sizes grew to large values at high loads. Further analysis revealed that the IFQ had too many duplicate RREQs from the same source to the same destination. This indicates that the routing protocol is over-reactive and that it is initiating too many network-wide floods to repair broken routes. We also showed that generic broadcast management techniques are not very effective under traffic overload.

Reducing unnecessary control packets

We proposed two different techniques to reduce unnecessary control packets. First, we evaluated a simple approach to remove the duplicate RREQs from IFQs. In this technique, prior to inserting an RREQ into an IFQ, the queue is checked for a duplicate RREQ. If there is duplicate RREQ, the newer RREQ replaces the older RREQ. We call this technique the reduced broadcasts (RB) technique. Using simulations, we showed that RB reduces the number of RREQs transmitted.

The second technique is to estimate the node queue delay adaptively. When an RREQ is transmitted, estimated queue delays are used in computing the estimated route reply time, instead of a static formula used by AODV. Since our estimation is adaptive and increases with the load, it is close to the real route reply time; therefore, there are fewer duplicate RREQs transmitted by the nodes to repair routes. This technique is called the DHT (Dynamic Hop Time).

Performance evaluation

The RB and DHT techniques are different but effective in reducing the numbers of RREQs in the MANET. We compared the performance of these techniques to the original AODV routing protocol. The results showed that there is a significant improvement in peak throughput and that the modification allows the network's performance to degrade gracefully with traffic overload. In particular, the DHT technique retains nearly all of the peak throughput, even under heavy traffic loads.

Predicting Next Hop Status

On-demand ad hoc routing protocols rely on MAC transmission failure notification (also called link layer feedback) to detect unreachable next hops, and hence broken routes. Each transmission failure reported to the routing layer may result in a route error being transmitted to multiple previous nodes in the paths affected by the broken link. When the network is in a state of saturation, transmission failures can occur for several reasons. All transmission failures reported by the MAC layer are interpreted as next hop out of range due to node movement, which is not always true. Transmission failures can also be caused by hidden nodes, exposed nodes, and collisions. Such transmission failures should not be considered as route failures, as they are temporary. Treating all transmission failures as permanent route failures increases the total number of route breaks perceived by the routing protocol. As a result, the control overhead increases. Therefore, reducing these unnecessary route breaks can reduce the routing overhead and improve network performance under high loads. To remedy this, we proposed link status prediction techniques.

A prediction technique can be used to predict whether the next hop is in range or out of range. Use of prediction reduces node dependency on link layer feedback and helps distinguish real route breaks from false route breaks. Eliminating unnecessary route errors and route discoveries due to false route breaks reduces the control overhead and provides more bandwidth for data transmissions.

The idea of link status prediction is similar to the branch prediction techniques used in the pipelined processor [55]. These prediction techniques can be employed at the MAC layer and, thus, can benefit a variety of routing protocols.

Classification of prediction schemes

Prediction of in range next hop can be done prior to a transmission attempt (pre-transmission prediction) or after a transmission failure (post-transmission prediction). If the pre-transmission prediction predict that the next hop is in range, then normal transmission is done; otherwise, no transmission is attempted and a route error is initiated. Pre-transmission prediction can save bandwidth by predicting out of range correctly, since fruitless transmission attempts are reduced. However, incorrect out of range predictions can cost dearly, since they increase false route breaks, which increase the routing overhead.

Post-transmission prediction is used after a transmission failure to predict whether the failed next hop is actually out of range. If the next hop is predicted to be out of range, normal link layer feedback is provided to the routing protocol; otherwise, the packet is dropped silently without any link layer feedback. If the next hop is correctly predicted as in range, an expensive false route break is avoided. If the in range prediction is incorrect (an out of range next hop is predicted as in range), then the discovery of a route break is delayed, which can decrease the throuhput in networks that are not congested.

Prediction criteria

We proposed four different prediction criteria: time, distance, signal-to-noise ratio and state. The time-based predictor is the simplest of the four, in which the next hop is predicted to be out of range if the elapsed time the next hop was last heard exceeded a preset limit. By varying this, we found that, for the 100-node MANET we simulated, a limit of five seconds gives the best accuracy.

The distance-based predictor computes the distance between two nodes using the previously received signal strengths and the relative speed of the next hop. If the predicted distance is less than 376 m, then the next hop is predicted to be in range; otherwise, it is predicted to be out of range.

The signal-to-noise ratio (SNR) predictor is similar to the distance predictor, but instead of calculating the distance to the next hop, the signal-to-noise ratio at the next hop is predicted.

The state predictor uses three states, good, maybe, and bad, to keep track of the next hops status. Each transmission failure causes the state of next hop to be degraded and each transmission heard from it causes its status to be upgraded. If the next hop status is bad, a link layer feedback is generated.

Performance evaluation

We evaluated the impact of pre- and post-transmission predictions on two on-demand routing protocols, AODV and DSR. Both routing protocols perform poorly under high loads without any modification. In particular, AODV increases false route breaks rapidly when the MANET reaches saturation, and DSR increases the number of real route breaks as the load increases. Using the proposed predictors, we show that these false and real route breaks can be reduced.

Using simulations, we show that DSR, the routing protocol with large numbers of real route breaks, can benefit from pre-transmission prediction. Here, the prediction saves unnecessary transmission attempts to nodes that are known to be out of range. As a result, no bandwidth is wasted in attempting to transmit data that is undeliverable. As result, there is more bandwidth available for data transmission. Our simulation results show that the DSR is able to double the throughput when the prediction techniques are used. The cost of incorrect prediction is a false route break, which increases routing overhead. If the cost of incorrect prediction is higher than the benefit of correct prediction, then the MANET will perform poorly using pre- transmission prediction. Under low loads, this is the case when using time-based pre-transmission prediction. We have further shown that these losses can be reduced using adaptive prediction, which applies prediction only in an overloaded network.

For AODV, we showed that post-transmission prediction can help reduce the number of false route breaks. Our results indicate that AODV is able to achieve slightly higher peak throughput and able to sustain this throughput beyond saturation.

7.3 Interface Queue Scheduling

Traditionally, the IFQs use First-Come-First-Served (FCFS) discipline. In this work, we show that FCFS queuing can be inefficient in wireless networks, where nodes can be unreachable for bursts of time. Unreachable nodes create head-of-the-line problems, where repated attempts are made to deliver a packets in the head of the line while the other data packets with reachable next hops sit in the queue. These attempts to tranmit to unreachable next hop nodes will increase the wireless network noise while delaying the reset of

the data in queue.

We proposed the channel state based scheduling (CSDS) to reduce the head-of-the-line phenomenon by queuing data based on the channel or next hop state (reachable or not reachable). Data to reachable next hops will be transmitted while the data to unreachable next hops (with a poor quality channel) will be held in the queue until the channel state improves. To implement CSDS, a couple of technical challenges must be met. First, the node must determine when the channel reaches an unreachable state and then it must determine when the node moves back into a reachable state.

Channel sensing

To detect the channel state, a channel sensing mechanism is essential. First, we evaluate and conclude that there is a strong correlation between a signal strength and packet transmission errors. A three state transmission diagram is proposed to establish the next hop channel state. If the neighbor node is reachable, the node is in a *good state*; if the neighbor node is unreachable, the node is in a *marked state*. When the node is in the *marked state*, the transmission of data is delayed until the channel is moved to an *unmarked state*. When a node in a *marked state* improves the channel, the node is moved to an *unmarked state*. In the *unmarked state*, data is transmitted with a higher priority. In the unmarked state, data is transmitted with a higher priority.

Wireless channel emulation

In a wireless testbed, it is very difficult to repeat an experiment to compare one or more alternative protocols. To overcome this difficulty, we performed a wireless channel emulation to study the comparative performance of CSDS and FCFS scheduling schemes. To emulate a wireless channel, we first obtained the signal strength traces for several independent experiments. Using this data, we computed the probability of signal strength changes. For example, if the current signal strength is n, we determine the probability of the next signal strength $n-2, n-1, n+1, n+2$ and so on. Using these probabilities, we model the wireless channel between a pair of wireless hosts as a *Finite-state Markov Chain* with n number of states, where n is a range of modeled signal strengths.

7.3.1 Performance evaluation

We setup two simple experiments to show that the CSDS uses the wireless channel more efficiently than the FCFS scheduling. In the first set of experiments, one host (source) is communicating (unidirectionally) with three other hosts via wireless links. Two of these links are assumed to be strong and are not modulated. The third link is modulated. The source generates UDP packets at a steady rate and sends them to a randomly selected destination. Receivers counted the number of packets correctly received to determine the received bandwidth. In this experiment, we show that CSDS can achieve up to 25% higher throughput in the emulated weak channel.

In the second set of experiments, four nodes are setup with two independent communication channels. One set of nodes is assumed to be strong and is not modulated; in the other set of nodes, the wach communication channel is emulated. This communication users the CSDS queuing. We show that the total network throughput is improved for medium and high loads on the weak links in this setup. For low loads, the channel sensing mechanism has consumed more bandwidth than what CSDS scheduling can save. As a result, there is a small loss in achieved throughput.

To evaluate File Transfer Protocols (FTP) performance, we use a setup similar to the first experiment. Here, we use different signal strenght trace than in the first two experiments and fiynd that the file transfer on weak channel using CSDS is 20-50% faster than FCFS scheduling.

7.4 Future Work

In a MANET, the bandwidth is the biggest constraint that must be addressed. In some cases, end-to-end delay, fairness, or power conservation can be more critical than the bandwidth constraint. For future work, we plan to investigate in detail the impact of our proposed modifications on end-to-end delay, fairness and power consumption.

In some scenarios, end-to-end delay can be the most critical factor in MANET performance. As the MANET enters saturation end-to-end delay increases rapidly. There are several reasons for high end-to-end delay: (a) frequent route breaks (2) slow discovery of routes (c) slow propagation of data across the MANET

(d) choosing long routes. In Chapter 3 we have shown that by reducing the number of route breaks that we are able to reduce end-to-end delay by a significant amount. In the future, we plan to investigate the other reasons for high end-to-end delay, and propose protocol modification to mitigate high end-to-end delay.

In the literature, there are several studies that have questioned TCP fairness in MANETs. In Chapter 3, we have shown a simple node setup, where distant but competing nodes' communication can be unfair in the throughput that they would receive. Due the nature of MANET communication, it is difficult to define a suitable fairness model. For future work, we plan to devise a fairness model for MANETs, and to evaluate the fairness of the MANETs beyond saturation. If the MANET increases its unfairness beyond saturation, we plan to investigate the reason and evaluate techniques to reduce such unfairness.

As the network reaches saturation, simulations show that the 802.11 MAC protocol rapidly increases the number of collisions and RTS transmissions. The increased number of collisions wastes the available network bandwidth and increases the power consumption by MANET's nodes. We plan to evaluate the energy-effrciencies of our proposed modifications.

Appendix A

Collision Distance Calculation with Background Noise

For node N_2 to receive node N_1's transmission while node N_3 is transmitting, the signal-to-noise ratio at node N_2 must be greater than the SNR-THRESHOLD (10 dB is used in most simulations and off-the-shelf hardware):

For N_2 to successfully receive,

$$\begin{aligned} SNR &> \frac{\text{Incoming Signal from } N_1}{\text{Total } N_2 \text{ Noise}} \\ &= \frac{N_1 \text{ Signal strength}}{N_3 \text{ Signal strength + Background noise}}, \text{ Since } N_2 \text{ has no other noise sources} \end{aligned} \quad \text{(A.1)}$$

In our simulations we use a background noise of -100.97dBm. This can be converted to mW using (3.6) in Chapter 3.

$$\text{Background noise } = -100.97dBm = 7.99 \times 10^{-11} mW$$

Since N_3 is outside of N_2's communication range, N_3's signal strength is

$$P_{t_{N_3}} \left(\frac{2.25}{D^2}\right)^2.$$

Figure A.1: Collision distance calculation. N_1 is the source and N_2 is the destination of the communication. N_3 is distant node whose transmission increases the noise level at N_2.

When d is between 0 m and 226 m, N_1's signal at N_2 obtain using (3.4) as

$$P_{t_{N_1}} \left(\frac{0.125}{4\pi d}\right)^2.$$

Therefore, (A.1) can be rewritten as follows.

$$10 \quad > \quad \frac{P_{t_{N_1}} \left(\frac{0.125}{4\pi d}\right)^2}{P_{t_{N_3}} \left(\frac{2.25}{D^2}\right)^2 + 7.99 \times 10^{-11}}$$

Both N_2 and N_3 are transmitting with a 15dBm power. Therefore,

$$P_{t_{N_1}} = P_{t_{N_3}} = 15\ dBm = 31.62\ mW$$

Now we can rewrite,

$$\begin{aligned}
10 \quad &> \quad \frac{31.62 \left(\frac{9.9\times10^{-5}}{d^2}\right)}{31.62 \left(\frac{5.0625}{D^4}\right) + 7.99 \times 10^{-11}} \\
&= \quad \frac{\left(\frac{3.13\times10^{-4}}{d^2}\right)}{\left(\frac{160.08+7.99\times10^{-11}D^4}{D^4}\right)} \\
&= \quad \frac{3.13 \times 10^{-4} D^4}{\left(160.08 + 7.99 \times 10^{-11} D^4\right) d^2} \\
\left(160.08 + 7.99 \times 10^{-11} D^4\right) 10d^2 \quad &> \quad 3.13 \times 10^{-4} D^4 \\
160.08d^2 + 7.99 \times 10^{-11} d^2 D^4 \quad &> \quad 3.13 \times 10^{-5} D^4 \\
160.08d^2 \quad &> \quad \left(3.13 \times 10^{-5} - 7.99 \times 10^{-11} d^2\right) D^4 \\
\frac{160.08d^2}{3.13 \times 10^{-5} - 7.99 \times 10^{-11} d^2} \quad &> \quad D^4 \\
D \quad &< \quad \sqrt[4]{\frac{160.08d^2}{3.13 \times 10^{-5} - 7.99 \times 10^{-11} d^2}}
\end{aligned} \tag{A.2}$$

When d is between 227 m and 376 m path loss is determined using (3.3) as 3,

$$P_{t_{N_1}} \left(\frac{2.25}{d^2}\right)^2.$$

Now, (**??**) can be manipulated as follows.

$$
\begin{aligned}
10 &> \frac{P_{t_{N_1}}\left(\frac{2.25}{d^2}\right)^2}{P_{t_{N_3}}\left(\frac{2.25}{D^2}\right)^2 + 7.99\times10^{-11}} \\
10 &> \frac{31.62\left(\frac{2.25}{d^2}\right)^2}{31.62\left(\frac{2.25}{D^2}\right)^2 + 7.99\times10^{-11}} \\
10 &> \frac{31.62\left(\frac{5.0625}{d^4}\right)}{31.62\left(\frac{5.0625}{D^4}\right) + 7.99\times10^{-11}} \\
&= \frac{160.08D^4}{\left(160.08 + 7.99\times10^{-11}\right)D^4} \\
d^4\left(160.08 + 7.99\times10^{-11}D^4\right) &> 16.0D^4 \\
160.08d^4 &> \left(16.0 - 7.99\times10^{-11}\right)D^4 \\
\frac{160.08d^4}{16.0 - 7.99\times10^{-11}} &> D^4 \\
D &< \sqrt[4]{\frac{160.08}{16.0 - 7.99\times10^{-11}}}\ d
\end{aligned}
\tag{A.3}
$$

Bibliography

[1] The network simulator - ns-2. http://www.isi.edu/nsnam/ns/.

[2] N. Abramson. Development of the alohanet. *IEEE Trans. on Information Theoy*, 31:119–123, 1985.

[3] Arup Acharya, Archan Misra, and Sorav Bansal. MACA-P: A MAC for concurrent transmissions in multi-hop wireless networks. In *Proceedings of the First IEEE International Conference on Pervasive Computing and Communications*, page 505. IEEE Computer Society, 2003.

[4] Ajay Bakre and B. R. Badrinath. I-TCP: Indirect TCP for mobile hosts. *15th International Conference on Distributed Computing Systems*, 1995.

[5] Anand Balachandran, Geoffrey M. Voelker, and Paramvir Bahl. Wireless hotspots: current challenges and future directions. In *Proceedings of the 1st ACM international workshop on Wireless mobile applications and services on WLAN hotspots*, pages 1–9. ACM Press, 2003.

[6] H. Balakrishnan, S. Seshan, E. Amir, and R. Katz. Improving TCP over wireless networks. *Proceedings of ACM Mobicom'95 Conferance*, November 1995.

[7] Hari Balakrishnan, Venkata N. Padmanabhan, Srinivasan Seshan, and Randy H. Katz. A comparison of mechanisms for improving TCP performance over wireless links. *IEEE/ACM Transactions on Networking*, 5(6):756–769, 1997.

[8] Christian Bettstetter, Giovanni Resta, and Paolo Santi. The node distribution of the random waypoint mobility model for wireless ad hoc networks. *IEEE Transactions on Mobile Computing*, 2(3):257–269, 2003.

[9] P. Bhagwat, P. Bhattacharya, A. Krishna, and S. Tripathi. Using channel state dependent packet scheduling to improve TCP throughput over wireless lans. *Wireless Networks*, 3:91–102, 1997.

[10] D.P. Bhargava, S. Agrawal. Security enhancements in AODV protocol for wireless ad hoc networks. In *Vehicular Technology Conference*, 7-11 Oct. 2001.

[11] J. Broch, D. A. Maltz, D. B. Johnson, Y.C. Hu, and J. Jetcheva. A performance comparison of multi-hop wireless ad hoc network routing protocols. In *Proc. 4th Annual ACM/IEEE International Conf. on Mobile Computing and Networking (ACM MobiCom '98)*, pages 85–97, Oct. 1998.

[12] Ramon Caceres and Liviu Iftode. Improving the performance of reliable transport protocols in mobile computing environments. *IEEE Journal of Selected Areas in Communications*, 13(5):850–857, 1995.

[13] R. Castenada, S. R. Das, and M. K. Marina. Query localization techniques for on-demand routing protocols for mobile ad hoc networks. *ACM/Kluwer Wireless Networks (WINET)*, 8(2):137–151, March 2002.

[14] David Cavin, Yoav Sasson, and Andre Schiper. On the accuracy of MANET simulators. In *Proc. ACM POMC'02*, pages 38–43, 2002.

[15] C. Cheng, R. Riely, and S.P.R. Kumar. A loop-free extended bellman-ford routing protocol without bouncing effect. In *In Proc. ACM SIGCOMM 89*, pages 224–236, 1989.

[16] Kwan-Wu Chin, John Judge, Aidan Williams, and Roger Kermode. Implementation experience with manet routing protocols. *SIGCOMM Comput. Commun. Rev.*, 32(5):49–59, 2002.

[17] T. Clausen and P. Jacquet. Optimized link state routing protocol (OLSR). In *IETF, Internet Draft, IEEE RFC 3626, http://www.ietf.org/rfc/rfc3626.txt*, October 2003.

[18] Lucent Technology Inc. Wireless communication and Networking Division. Wavelan air interface data manual. In *Document 407-0024785, Issue A*, April 1997.

[19] S. R. Das, R. Castaneda, J. Yan, and R. Sengupta. Comparative performance evaluation of routing protocols for mobile, ad hoc networks. In *Proceedings of the 7th Int. Conf. on Computer Communications and Networks (ICCCN)*, pages 153–161, Lafayette, LA, October, 1998.

[20] IEEE Standards Department. IEEE 802.11 standard for wireless LAN, medium access control (MAC) and physical layer (PHY) specifications, 1997.

[21] IEEE Standards Department. Supplement to IEEE standard for information technology- telecommunications and information exchange between systems- local and metropolitan area networks- specific requirements- part 11: Wireless lan medium access control (MAC) and physical layer (PHY) specifications: Higher-speed physical layer extension in the 2.4 ghz band, 1999.

[22] IEEE Standards Department. Supplement to IEEE standard for information technology telecommunications and information exchange between systems - local and metropolitan area networks - specific requirements. part 11: wireless lan medium access control (mac) and physical layer (phy) specifications: high-speed physical layer in the 5 ghz band, 1999.

[23] IEEE Standards Department. IEEE std 802.11g-2003 (amendment to IEEE Std 802.11, 1999 edn. (reaff 2003) as amended by IEEE stds 802.11a-1999, 802.11b-1999, 802.11b-1999/cor 1-2001, and 802.11d-2001), 2003.

[24] S. Desilva and R. Boppana. On the impact of noise sensitivity on performance in 802.11 based ad hoc networks. In *Proceeding of International Conference on Communications (ICC04) (http://www.cs.utsa.edu/ sdesilva/publications/icc04.pdf)*, Paris, France, June, 2004.

[25] S. Desilva and S. Das. Experimental evaluation of a wireless ad hoc network. In *Proceedings of the 9th Int. Conf. on Computer Communications and Networks (ICCCN)*, Las Vegas, October, 2000.

[26] The Linux documentation project(LDP). Lpd homepage.

[27] T. Dyer and R.V. Boppana. A comparison of TCP performance over three routing protocols for mobile ad hoc networks. In *Proceedings of ACM Mobihoc*, October 2001.

[28] D. Eckhardt and P. Steenkiste. Measurement and analysis of error characteristics of an in-building wireless network. In *Proc. ACM SIGCOMM'96 Conf.*, pages 243–254, Aug. 1996.

[29] K. Fall and K. Varadhan. The ns manual, December 13, 2002.

[30] Chane L. Fullmer and J. J. Garcia-Luna-Aceves. Floor acquisition multiple access (FAMA) for packet-radio networks. In *SIGCOMM*, pages 262–273, 1995.

[31] M. Gerla, K. Taek, and G. Pei. On-demand routing in large ad hoc wireless networks with passive clustering. In *Wireless Communications and Networking Conference, 2000. WCNC*, 2002.

[32] M. Gerla, K. Tang, and R. Bagrodia. TCP performance in wireless multi-hop networks. In *Proceedings of IEEE WMCSA'99*, February 1999.

[33] CMU Monarch Group. CMU monarch extensions to ns. http://www.monarch.cs.cmu.edu/.

[34] D. Gu, G. Pei, H. Ly, M. Gerla, B., Zhang, and X. Hong. UAV aided intelligent routing for ad-hoc wireless network in single-area theater. In *Wireless Communications and Networking Conference, 2000 WCNC*, 2000.

[35] Gavin Holland and Nitin H. Vaidya. Analysis of TCP performance over mobile ad hoc networks. In *Mobile Computing and Networking*, pages 219–230, 1999.

[36] S. Hu and T. Saadawi. Revealing TCP unfairness behavior in 802.11 based wireless multi-hop networks. *Personal, Indoor and Mobile Radio Communications 2001 12th IEEE International Symposium on*, pages E–83 –E–87, Sep/Oct 2001.

[37] R. Jain, D. Chiu, and W. Hawe. A quantitative measure of fairness and discrimination for resource allocation in shared computer systems.

[38] David B Johnson and David A Maltz. Dynamic source routing in ad hoc wireless networks. In Imielinski and Korth, editors, *Mobile Computing*, volume 353. Kluwer Academic Publishers, 1996.

[39] P. Karn. MACA - a new channel access protocol for packet radio. pages 134–140, 1990.

[40] Srinivasan Keshav. ATM networks, the internet, and the telephone netowrk. In *An Engineering Approach to Computer Networking*. Addison-Wesley, 1997.

[41] L. Kleinrock and F. Tobagi. Packet switching in radio channels: Part i—carrier sense multiple-access modes and their throughput-delay characteristics. *IEEE Transactions on Communications COM-23*, pages 272–288, December 1975.

[42] Young-Bae Ko and Nitin H. Vaidya. Location-aided routing (lar) in mobile ad hoc networks. *Wirel. Netw.*, 6(4):307–321, 2000.

[43] A. Laouiti, A. Qayyum, and L. Viennot. Multipoint relaying: An efficient technique for flooding in mobile wireless networks. In *35th Annual Hawaii International Conference on System Sciences (HICSS'2001)*. IEEE Computer Society, 2001.

[44] C. T. Lau and C. Leung. Capture models for mobile packet radio networks. *IEEE Trans. on Communications*, 40(5):917–925, May 1992.

[45] S. Lee and C. Kim. Multicast tree construction and flooding in wirelss ad hoc netwroks. In *In proceedings of the ACM International Workshop on ¡odeling, Analysis, and Simulation of Wireless and Mobile Systems (MSWIM)*, 2000.

[46] J. Liu and S. Singh. ATCP: TCP for mobile ad hoc networks. *IEEE J–SAC*, 19(7):1300–1315, 2001.

[47] J. Macker and S. Corson. Mobile ad hoc networks (MANET).

[48] David A. Maltz, Josh Broch, and David B. Johnson. Experiences designing and building a multi-hop wireless ad hoc network testbed. http://www.monarch.cs.cmu.edu/.

[49] M. K. Marina and S. R. Das. Performance of route caching strategies in dynamic source routing. In *Proceedings of the 2nd Wireless Networking and Mobile Computing (WNMC), In conjunction with the Int'l Conference on Distributed Computing Systems (ICDCS) 2001*, Phoenix, April 2001.

[50] Ramesh Neelamani and Amit Saxena. TCP over wireless.

[51] Keng Seng Ng and W.K.G. Seah. Routing security and data confidentiality for mobile ad hoc networks. In *Vehicular Technology Conference*, 22-25 April 2003.

[52] Ze-Yao Ni, Yu-Chee Tseng, Yuh-Shyan, and Jang-Ping Sheu. The broadcast storm problem in a mobile ad hoc network. *Wireless Networks*, 8, March 2003.

[53] B. Noble, M. Satyanarayanan, G. T. Nguyen, and R. H. Katzd. Trace-based mobile network emulation. In *Proc. of 1997 ACM SIGCOMM Conf.*, pages 51–61, Sept 1997.

[54] R. Ogier, F. Templin, and M. Lewis. Topology dissemination based on reverse-path forwarding (TBRPF). In *IETF, Internet Draft, http://www.ietf.org/internet-drafts/draft-ietf-manet-tbrpf-11.txt*, April 13, 2004.

[55] David A. Patterson and John L. Hennessy. *Computer Organization and Design: the Hardware/Software Interface*. Morgan Kaufmann, San Manteo, CA.

[56] Wei Peng and Xi-Cheng Lu. On the reduction of broadcast redundancy in mobile ad hoc networks. In *Proceedings of the 1st ACM international symposium on Mobile ad hoc networking & computing*, pages 129–130. IEEE Press, 2000.

[57] Wei Peng and Xi-Cheng Lu. On the reduction of broadcast redundancy in mobile ad hoc networks. In *Proceedings of the 1st ACM international symposium on Mobile ad hoc networking & computing*, pages 129–130. IEEE Press, 2000.

[58] C. Perkins. Ad-hoc on-demand distance vector routing. In *MILCOM '97 panel on Ad Hoc Networks*, Nov. 1997.

[59] C. E. Perkins, E. M. Moyer, and S. R. Das. Ad hoc on demand distance vector (AODV) routing. In *IETF, Internet Draft, IEEE RFC 3561, http://www.ietf.org/rfc/rfc3561.txt?number=3561*, July 2003.

[60] C. E. Perkins and E. M. Royer. Ad-hoc on-demand distance vector routing. In *Proceeding of the IEEE workshop on Mobile Computing Systems and Applications*, pages 90–100, February 1999.

[61] L. Peterson and B. Davie. *Computer Networks*. Morgan Kaufmann, 2000.

[62] S. Ray, J. Carruthers, and D. Starobinski. RTS/CTS-induced congestion in ad hoc wireless LAN. In *In Proceedings of IEEE WCNC 2003*, 2003.

[63] Giovanni Resta and Paolo Santi. An analysis of the node spatial distribution of the random waypoint mobility model for ad hoc networks. In *Proceedings of the second ACM international workshop on Principles of mobile computing*, pages 44–50. ACM Press, 2002.

[64] Deepanshu Shukla, Leena Chandran-Wadia, and Sridhar Iyerr. Mitigating the exposed node problem in IEEE 80211 ad hoc network. In *Computer Communications and Networks, 2003. ICCCN 2003. Proceedings*, pages 157 – 162, 2003.

[65] S. Singh and C. Raghavendra. Pamas: Power aware multi-access protocol with signalling for ad hoc networks. *ACM ComputerCommunications Review*, 1999.

[66] Suresh Singh, Mike Woo, and C. S. Raghavendra. Power-aware routing in mobile ad hoc networks. In *Mobile Computing and Networking*, pages 181–190, 1998.

[67] S. Sivaprakasam and K. S. Shanmugan. An equivalent markov model for burst errors in digital channels. *IEEE Trans. on Communications*, 43(2/3/4):1347–1355, Feb/March/April 1995.

[68] F. A. Tobagi and L. Klerinrock. Packet switching in radio channels: Part ii the hidden terminal problem in carrier sense multiple-access and the busy-tone. *IEEE Trans. on Communications*, pages 1417–1422, December 1975.

[69] Yu-Chee Tseng, Sze-Yao Ni, Yuh-Shyan Chen, and Jang-Ping Sheu. The broadcast storm problem in a mobile ad hoc network. *Wirel. Netw.*, 8(2/3):153–167, 2002.

[70] B. Tuch. Development of wavelan, an ism band wireless lan. *AT&T Technical Journal*, pages 27–37, July/Aug 1993.

[71] Feng Wang and Yongguang Zhang. Improving TCP performance over mobile ad-hoc networks with out-of-order detection and response. In *Proceedings of the 3rd ACM international symposium on Mobile ad hoc networking & computing*, pages 217–225. ACM Press, 2002.

[72] H. S. Wang and N. Moayeri. Finte-state markov channel– a useful model for radio communication channel. *IEEE Trans. on Vehicular Technology*, 44(1):163–171, Feb. 1995.

[73] C. Ware, J. Judge, J. Chicharo, and E. Dutkieewicz. Unfairness and capture behavior in 802.11 ad hoc network. In *IEEE International Conference on Communications (ICC 2000)*, 2000.

[74] C. Ware, T. Wysocki, and J. Chicharo. Simulation capture behavior in 802.11 radio modens. *Telecommunication and information Technology*, 2(2):46–54, 2001.

[75] Brad Williams and Tracy Camp. Comparison of broadcasting techniques for mobile ad hoc networks. In *Proceedings of the 3rd ACM international symposium on Mobile ad hoc networking & computing*, pages 194–205. ACM Press, 2002.

[76] K. Xu, M. Gerla, and S. Bae. Effectiveness of RTS/CTS handshake in IEEE 802.11 based adhoc networks. *Ad Hoc Network Journal*, 1(1).

[77] S. Xu and T. Saadawi. Revealing the problems with 802.11 medium access control protocol in multihop wireless ad hoc network. *Journal of Computer Networks*, 38(4):531–548, March 2002.

[78] Xiang Zeng, Rajive Bagrodia, and Mario Gerla. Glomosim: A library for parallel simulation of large-scale wireless networks. In *Workshop on Parallel and Distributed Simulation*, pages 154–161, 1998.

[79] Yongguang Zhang and Wei Li. An integrated environment for testing mobile ad-hoc networks. In *Proceedings of the 3rd ACM international symposium on Mobile ad hoc networking & computing*, pages 104–111. ACM Press, 2002.

[80] L. Zhou and Z.J. Haas. Securing ad hoc networks. *IEEE Network*, 13(6):24–30, 1999.

Vita

Saman Desilva was born in Colombo, Sri Lanka, on February 12, 1967. He is the son of Inter De silva and Dulcie De silva. After completing his studies at the D. S. Senanayake High School in Colombo, Sri Lanka, in 1986, he pursued a career in accounting at The Institute of Chartered Accountants in Colombo, Sri Lanka. Later, he attended The University of Arizona in Tempe, Arizona, from 1990 to 1992. After transferring to The University of Texas in 1992, he obtained a Bachelor of Science degree in Computer Science in May of 1994. He continued his graduate studies at The University of Texas at San Antonio, where he earned a Master of Science degree in Computer Science in May of 1996.

Publications in Computer Science

S. Desilva and S.R. Das, Experimental Evaluation of a Wireless Ad Hoc Network. *Proceedings of the 9th Int. Conf. on Computer Communications and Networks (IC3N)*, Las Vegas, October 2000.

S. Desilva and S.R. Das, Experimental Evaluation of Channel State Dependent Scheduling in an In-building Wireless LAN. In *Proceedings of the 7th Int. Conf. on Computer Communications and Networks (IC3N)*, Lafayette, Louisina, October, 1998, pages 414-421.

S. Desilva and R. Hiromoto, A Compatible TCP Protocol for Ad Hoc Wireless Network, In *Proceedings of 12th IEEE Workshop on Local and Metropolitan Area Networks*, August 11-14 2002, Stockholm, Sweden.

S. Desilva and R.V. Boppana, On the Impact of Noise Sensitivity on Performance in 802.11 Based Ad hoc Networks, In *Proceeding of The International Conference on Communications (ICC04)*, June 2004, Paris, France.

S. Desilva and R.V. Boppana, Sustaining Performance Under Traffic Overload, In *Proceeding of International 2004 International Workshop on Mobile and Wireless Ad Hoc Networking (MWAN04) in conjunction with The 2004 International Conference on Wireless Networks*, Las Vegas, Nevada, USA.

S. Desilva and R.V. Boppana, Mitigating Malicious Control Packet Floods in Ad Hoc Networks, To appear in the *Proceeding of IEEE Wireless Communication and Networking Conference (WCNC2005)*, March 2005, New Orleans, Louisiana, USA.