

**USING TERRAIN AND LOCATION INFORMATION TO IMPROVE
ROUTING IN AD HOC NETWORKS**

A Dissertation
Presented to
The Academic Faculty

by

Brian Rivera

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
School of Electrical and Computer Engineering

Georgia Institute of Technology

May 2007

**USING TERRAIN AND LOCATION INFORMATION TO IMPROVE
ROUTING IN AD HOC NETWORKS**

Approved by:

Dr. Henry Owen, Advisor
School of Electrical and Computer
Engineering
Georgia Institute of Technology

Dr. Yorai Wardi
School of Electrical and Computer
Engineering
Georgia Institute of Technology

Dr. John Copeland
School of Electrical and Computer
Engineering
Georgia Institute of Technology

Dr. Randal Abler
School of Electrical and Computer
Engineering
Georgia Institute of Technology

Dr. Jun Xu
College of Computing
Georgia Institute of Technology

Date Approved: March 26, 2007 □

To my wife Keira and my daughter Kaitlyn

ACKNOWLEDGEMENTS

First and foremost, I wish to thank my advisor Henry Owen for supporting me as I slowly completed this research. His patience and assistance helped me finally get this done. I'd also like to thank my committee members for their patience.

I'd like to thank my fellow PhD students, Ben Koehler, Ozgur Gurbuz and Myong Choi. Ben made the several grad courses we took together interesting and provided support to keep me going. Ozgur provided good advice on being a graduate student. Myong helped me get settled in Henry's lab and introduced me to Quake Arena.

I'd like to also thank my some of my co workers, Dr. Reginald Hobbs, Dr. Adrienne Raglin, Dr Ananthram Swami, Greg Cirincione, Natalie Ivanic and Richard Gopaul for their support. Ananthram spent many an afternoon answering various questions about the physical layer as well as the finer points of probability and statistics. I'd also like to Dr John Gowens for convincing me to begin the PhD program. I don't think he thought it would take me this long. I'd also like to thank my current immediate management chain: Glenn Racine & Dr Barbara Broome for supporting me and helping me carve out time to get this research done.

I'd also like to thank my parents for their support and encouragement. Last and most importantly, I would like to thank my wife Keira and daughter Kaitlyn for supporting me in my long journey toward completing this dissertation. Their patience and support as I devoted many days and nights to completing this research is greatly appreciated.

TABLE OF CONTENTS

	Page
ACKNOWLEDGEMENTS.....	iv
LIST OF TABLES.....	viii
LIST OF FIGURES.....	ix
LIST OF ABBREVIATIONS.....	xix
SUMMARY.....	xx
CHAPTER 1 INTRODUCTION.....	1
1.1 Applications of Ad hoc Network.....	1
1.2 Differences Between Wired and Wireless Networks.....	2
1.3 Research Objectives.....	2
1.4 Thesis Outline.....	3
CHAPTER 2 BACKGROUND AND RELATED WORK.....	4
2.1 Classification of MANET Protocols.....	4
2.2 On Demand Algorithms.....	4
2.3 Proactive Algorithms.....	5
2.4 Link State versus Distance Vector.....	5
2.5 Hybrid Algorithms.....	6
CHAPTER 3 MANET ROUTING PROTOCOLS.....	7
3.1 Ad hoc On Demand Distance Vector (AODV) Routing Protocol.....	7
3.2 Dynamic Source Routing (DSR) Protocol.....	8
CHAPTER 4 GEOGRAPHIC & LOCATION BASED ROUTING PROTOCOLS.....	10
4.1 Distance Routing Effect Algorithm for Mobility (DREAM).....	11

4.2 Greedy Perimeter Stateless Routing (GPSR).....	16
4.3 Scalable Location Update-based Routing Protocol (SLURP).....	19
4.4 Location Aided Routing (LAR).....	21
CHAPTER 5 TERRAIN BASED ROUTING PROTOCOL.....	25
5.1 Algorithm Overview.....	25
CHAPTER 6 SIMULATION EXPERIMENTS.....	35
6.1 Simulation Overview.....	35
6.2 Performance Metrics.....	36
6.3 Experiment 1: Effects of Path Loss Prediction Error on TBR Performance.....	38
6.4 Experiment 2: Comparison of TBR to DSR, DREAM, and LAR.....	40
CHAPTER 7 RESULTS.....	43
7.1 Experiment 1: Effects of Path Loss Prediction Error on TBR Performance.....	44
7.2 Experiment 2: Comparison of TBR to DSR, DREAM, and LAR.....	54
CHAPTER 8 CONCLUSIONS AND CONTRIBUTIONS.....	83
8.1 Summary of Experiment 1: Effects of Path Loss Prediction Error on TBR Performance.....	83
8.2 Summary of Experiment 2: Comparison of DREAM, DSR, & LAR to TBR...	84
8.3 Contributions.....	85
CHAPTER 9 FUTURE WORK.....	87
APPENDIX A DETAILED RESULTS.....	89
9.1 Graphs for Experiment 1: Effects of Prediction Error in Flat Terrain.....	89
9.2 Graphs for Experiment 1: Effects of Prediction Error in Hilly Terrain.....	93
9.3 Graphs for Experiment 2 Performance Comparison in Flat Terrain.....	98

9.4 Graphs for Experiment 2 Performance Comparison in Hilly Terrain.....	141
REFERENCES.....	184

LIST OF TABLES

	Page
Table 1: MANET Protocol Classification	4
Table 2: Performance Metrics.....	38
Table 3: Path Loss Prediction Error Simulation Parameters	39
Table 4: Scenario Parameters.....	41
Table 5: Performance Metrics.....	44

LIST OF FIGURES

	Page
Figure 1: DREAM Routing Cone	13
Figure 2: DREAM Forwarding Restrictions based on Distance.....	15
Figure 3: DREAM Cone Degeneration	16
Figure 4: GPSR Greedy Forwarding Failure	17
Figure 5: LAR Scheme 2 Failure	24
Figure 6: Routing Table Entry	27
Figure 7: TBR Route Request Packet Format	30
Figure 8: TBR Route Reply Packet Format.....	31
Figure 9: TBR Data Packet Format.....	32
Figure 10: Failed Link Table Entry.....	32
Figure 11: TBR Route Maintenance Packet Format	33
Figure 12: Route Maintenance.....	34
Figure 13: Throughput in Flat Terrain	45
Figure 14: Packet Delivery Ratio in Flat Terrain.....	46
Figure 15: Average End to End Delay in Flat Terrain	47
Figure 16: Average Path Length in Flat Terrain.....	48
Figure 17: Number of Overhead Packets in Flat Terrain	49
Figure 18: Packet Efficiency in Flat Terrain	50
Figure 19: Throughput in Hilly Terrain.....	51
Figure 20: Packet Delivery Ratio in Hilly Terrain.....	51
Figure 21: Average End to End Delay in Hilly Terrain	52

Figure 22: Average Path Length in Hilly Terrain	53
Figure 23: Packet Transmission Efficiency in Hilly Terrain.....	54
Figure 24: Throughput in Flat Terrain at 1 Packet/Second	57
Figure 25: Packet Delivery Ratio in Flat Terrain at 1 Packet/Second	59
Figure 26: Average End to End Delay in Flat Terrain at 1 Packet/Second.....	60
Figure 27: Packet Delivery Efficiency in Flat Terrain, 1 Packet/Second.....	62
Figure 28: Throughput in Flat Terrain at 2 Packets/Second.....	63
Figure 29: Packet Delivery Ratio in Flat Terrain at 3 Packets/Second.....	65
Figure 30: Throughput in Flat Terrain at 4 Packets/Second.....	68
Figure 31: Number of Overhead Packets in Flat Terrain at 4 Packets/Second.....	70
Figure 32: Average End to End Delay in Hilly Terrain at 1 Packet/Second	72
Figure 33: Packet Delivery Ratio in Hilly Terrain at 1 Packet/Second.....	73
Figure 34: Packet Delivery Efficiency in Hilly Terrain at 2 Packets/Second	75
Figure 35: Throughput in Hilly Terrain at 3 Packets/Second.....	77
Figure 36: Packet Delivery Efficiency in Hilly Terrain at 3 Packets/Second	78
Figure 37: Packet Delivery Efficiency in Hilly Terrain at 4 Packets/Second	80
Figure 38: Number of Overhead Packets in Hilly Terrain at 4 Packets/Second.....	81
Figure 39: Exp 1: Throughput in Flat Terrain	89
Figure 40: Exp 1: Packet Delivery Ratio in Flat Terrain.....	90
Figure 41: Exp 1: Average End to End Delay in Flat Terrain	90
Figure 42: Exp 1: Average Path Length in Flat Terrain.....	91
Figure 43: Exp 1: Number of Overhead Packets Transmitted in Flat Terrain.....	91
Figure 44: Exp 1: Total Number of Overhead Bits in Flat Terrain.....	92

Figure 45: Exp 1: Total Number of Packets Transmitted in Flat Terrain	92
Figure 46: Exp 1: Packet Delivery Efficiency in Flat Terrain.....	93
Figure 47: Exp 1: Throughput in Hilly Terrain	93
Figure 48: Exp 1: Packet Delivery Ratio in Hilly Terrain.....	94
Figure 49: Exp 1: Average End to End Delay in Hilly Terrain	94
Figure 50: Exp 1: Average Path Length in Hilly Terrain	95
Figure 51: Exp 1: Total Number of Overhead Packets in Hilly terrain	95
Figure 52: Exp 1: Total Number of Overhead Bits in Hilly Terrain.....	96
Figure 53: Exp 1: Total Number of Packets Transmitted in Hilly Terrain.....	96
Figure 54: Exp 1: Packet Delivery Efficiency in Hilly Terrain.....	97
Figure 55: Throughput (Flat, Low Mobility, 1 pkt/sec)	98
Figure 56: Throughput (Flat, Medium Mobility, 1 pkt/sec)	99
Figure 57: Throughput (Flat, High Mobility, 1 pkt/sec)	99
Figure 58: Packet Delivery Ratio (Flat, Low Mobility, 1 pkt/sec)	100
Figure 59: Packet Delivery Ratio (Flat, Medium Mobility, 1 pkt/sec)	100
Figure 60: Packet Delivery Ratio (Flat, High Mobility, 1 pkt/sec).....	101
Figure 61: Average End to End Delay (Flat, Low Mobility, 1 pkt/sec).....	101
Figure 62: Average End to End Delay (Flat, Medium Mobility, 1 pkt/sec).....	102
Figure 63: Average End to End Delay (Flat, High Mobility, 1 pkt/sec)	102
Figure 64: Average Path Length (Flat, Low Mobility, 1 pkt/sec).....	103
Figure 65: Average Path Length (Flat, Medium Mobility, 1 pkt/sec).....	103
Figure 66: Average Path Length (Flat, High Mobility, 1 pkt/sec).....	104
Figure 67: Number of Packets Transmitted (Flat, Low Mobility, 1 pkt/sec)	104

Figure 68: Number of Packets Transmitted (Flat, Medium Mobility, 1 pkt/sec)	105
Figure 69: Number of Packets Transmitted (Flat, High Mobility, 1 pkt/sec).....	105
Figure 70: Number of Overhead Packets (Flat, Low Mobility, 1 pkt/sec).....	106
Figure 71: Number of Overhead Packets (Flat, Medium Mobility, 1 pkt/sec).....	106
Figure 72: Number of Overhead Packets (Flat, High Mobility, 1 pkt/sec)	107
Figure 73: Packet Delivery Efficiency (Flat, Low Mobility, 1 pkt/sec).....	107
Figure 74: Packet Delivery Efficiency (Flat, Medium Mobility, 1 pkt/sec).....	108
Figure 75: Packet Delivery Efficiency (Flat, High Mobility, 1 pkt/sec).....	108
Figure 76: Throughput (Flat, Low Mobility, 2 pkts/sec).....	109
Figure 77: Throughput (Flat, Medium Mobility, 2 pkts/sec).....	109
Figure 78: Throughput (Flat, High Mobility, 2 pkts/sec).....	110
Figure 79: Packet Delivery Ratio (Flat, Low Mobility, 2 pkts/sec).....	110
Figure 80: Packet Delivery Ratio (Flat, Medium Mobility, 2 pkts/sec).....	111
Figure 81: Packet Delivery Ratio (Flat, High Mobility, 2 pkts/sec).....	111
Figure 82: Average End to End Delay (Flat, Low Mobility, 2 pkts/sec)	112
Figure 83: Average End to End Delay (Flat, Medium Mobility, 2 pkts/sec)	112
Figure 84: Average End to End Delay (Flat, High Mobility, 2 pkts/sec).....	113
Figure 85: Average Path Length (Flat, Low Mobility, 2 pkts/sec).....	113
Figure 86: Average Path Length (Flat, Medium Mobility, 2 pkts/sec).....	114
Figure 87: Average Path Length (Flat, High Mobility, 2 pkts/sec)	114
Figure 88: Number of Packets Transmitted (Flat, Low Mobility, 2 pkts/sec).....	115
Figure 89: Number of Packets Transmitted (Flat, Medium Mobility, 2 pkts/sec).....	115
Figure 90: Number of Packets Transmitted (Flat, High Mobility, 2 pkts/sec).....	116

Figure 91: Number of Overhead Packets (Flat, Low Mobility, 2 pkts/sec)	116
Figure 92: Number of Overhead Packets (Flat, Medium Mobility, 2 pkts/sec)	117
Figure 93: Number of Overhead Packets (Flat, High Mobility, 2 pkts/sec).....	117
Figure 94: Packet Delivery Efficiency (Flat, Low Mobility, 2 pkts/sec)	118
Figure 95: Packet Delivery Efficiency (Flat, Medium Mobility, 2 pkts/sec)	118
Figure 96: Packet Delivery Efficiency (Flat, High Mobility, 2 pkts/sec)	119
Figure 97: Throughput (Flat, Low Mobility, 3 pkts/sec).....	119
Figure 98: Throughput (Flat, Medium Mobility, 3 pkts/sec).....	120
Figure 99: Throughput (Flat, High Mobility, 3 pkts/sec).....	120
Figure 100: Packet Delivery Ratio (Flat, Low Mobility, 3 pkts/sec).....	121
Figure 101: Packet Delivery Ratio (Flat, Medium Mobility, 3 pkts/sec).....	121
Figure 102: Packet Delivery Ratio (Flat, High Mobility, 3 pkts/sec)	122
Figure 103: Average End to End Delay (Low Mobility, 3 Packets/Second).....	122
Figure 104: Average End to End Delay (Medium Mobility, 3 Packets/Second).....	123
Figure 105: Average End to End Delay (High Mobility, 3 Packets/Second).....	123
Figure 106: Average Path Length (Flat, Low Mobility, 3 pkts/sec)	124
Figure 107: Average Path Length (Flat, Medium Mobility, 3 pkts/sec)	124
Figure 108: Average Path Length (Flat, High Mobility, 3 pkts/sec)	125
Figure 109: Number of Packets Transmitted (Flat, Low Mobility, 3 pkts/sec).....	125
Figure 110: Number of Packets Transmitted (Flat, Medium Mobility, 3 pkts/sec).....	126
Figure 111: Number of Packets Transmitted (Flat, High Mobility, 3 pkts/sec)	126
Figure 112: Number of Overhead Packets (Flat, Low Mobility, 3 pkts/sec)	127
Figure 113: Number of Overhead Packets (Flat, Medium Mobility, 3 pkts/sec)	127

Figure 114: Number of Overhead Packets (Flat, High Mobility, 3 pkts/sec).....	128
Figure 115: Packet Delivery Efficiency (Flat, Low Mobility, 3 pkts/sec)	128
Figure 116: Packet Deliver Efficiency (Flat, Medium Mobility, 3 pkts/sec)	129
Figure 117: Packet Delivery Efficiency (Flat, High Mobility, 3 pkts/sec)	129
Figure 118: Throughput (Flat, Low Mobility, 4 pkts/sec).....	130
Figure 119: Throughput (Flat, Medium Mobility, 4 pkts/sec).....	130
Figure 120: Throughput (Flat, High Mobility, 4 pkts/sec)	131
Figure 121: Packet Delivery Ratio (Flat, Low Mobility, 4 pkts/sec).....	131
Figure 122: Packet Delivery Ratio (Flat, Medium Mobility, 4 pkts/sec).....	132
Figure 123: Packet Delivery Ratio (Flat, High Mobility, 4 pkts/sec)	132
Figure 124: Average End to End Delay (Flat, Low Mobility, 4 pkts/sec)	133
Figure 125: Average End to End Delay (Flat, Medium Mobility, 4 pkts/sec)	133
Figure 126: Average End to End Delay (Flat, High Mobility, 4 pkts/sec).....	134
Figure 127: Average Path Length (Flat, Low Mobility, 4 pkts/sec).....	134
Figure 128: Average Path Length (Flat, Medium Mobility, 4 pkts/sec)	135
Figure 129: Average Path Length (Flat, High Mobility, 4 pkts/sec)	135
Figure 130: Number of Packets Transmitted (Flat, Low Mobility, 4 pkts/sec).....	136
Figure 131: Number of Packets Transmitted (Flat, Medium Mobility, 4 pkts/sec).....	136
Figure 132: Total Number Packets Transmitted (Flat, High Mobility, 4 pkts/sec)	137
Figure 133: Number of Overhead Packets (Flat, Low Mobility, 4 pkts/sec)	137
Figure 134: Number of Overhead Packets (Flat, Medium Mobility, 4 pkts/sec)	138
Figure 135: Number of Overhead Packets (Flat, High Mobility, 4 pkts/sec).....	138
Figure 136: Packet Delivery Efficiency (Flat, Low Mobility, 4 pkts/sec)	139

Figure 137: Packet Delivery Efficiency (Flat, Medium Mobility, 4 pkts/sec)	139
Figure 138: Packet Delivery Efficiency (Flat, High Mobility, 4 pkts/sec)	140
Figure 139: Throughput (Hilly, Low Mobility, 1 pkt/sec)	141
Figure 140: Throughput (Hilly, Medium Mobility, 1 pkt/sec)	142
Figure 141: Throughput (Hilly, High Mobility, 1 pkt/sec).....	142
Figure 142: Packet Delivery Ratio (Hilly, Low Mobility, 1 pkt/sec)	143
Figure 143: Packet Delivery Ratio (Hilly, Medium Mobility, 1 pkt/sec)	143
Figure 144: Packet Delivery Ratio (Hilly, High Mobility, 1 pkt/sec).....	144
Figure 145: Average End to End Delay (Hilly, Low Mobility, 1 pkt/sec).....	144
Figure 146: Average End to End Delay (Hilly, Medium Mobility, 1 pkt/sec).....	145
Figure 147: Average End to End Delay (Hilly, High Mobility, 1 pkt/sec)	145
Figure 148: Average Path Length (Hilly, Low Mobility, 1 pkt/sec).....	146
Figure 149: Average Path Length (Hilly, Medium Mobility, 1 pkt/sec).....	146
Figure 150: Average Path Length (Hilly, High Mobility, 1 pkt/sec)	147
Figure 151: Number of Packets Transmitted (Hilly, Low Mobility, 1 pkt/sec)	147
Figure 152: Number of Packets Transmitted (Hilly, Medium Mobility, 1 pkt/sec)	148
Figure 153: Number of Packets Transmitted (Hilly, High Mobility, 1 pkt/sec).....	148
Figure 154: Total Number of Overhead Packets (Hilly, Low Mobility, 1 pkt/sec).....	149
Figure 155: Total Number of Overhead Packets (Hilly, Medium Mobility, 1 pkt/sec)..	149
Figure 156: Total Number of Overhead Packets (Hilly, High Mobility, 1 pkt/sec)	150
Figure 157: Packet Delivery Efficiency (Hilly, Low Mobility, 1 pkt/sec).....	150
Figure 158: Packet Delivery Efficiency (Hilly, Medium Mobility, 1 pkt/sec).....	151
Figure 159: Packet Delivery Efficiency (Hilly, High Mobility, 1 pkt/sec).....	151

Figure 160: Throughput (Hilly, Low Mobility, 2 pkts/sec).....	152
Figure 161: Throughput (Hilly, Medium Mobility, 2 pkts/sec).....	152
Figure 162: Throughput (Hilly High Mobility, 2 pkts/sec).....	153
Figure 163: Packet Delivery Ratio (Hilly, Low Mobility, 2 pkts/sec).....	153
Figure 164: Packet Delivery Ratio (Hilly, Medium Mobility, 2 pkts/sec).....	154
Figure 165: Packet Delivery Ratio (Hilly, High Mobility 2 pkts/sec).....	154
Figure 166: Average End to End Delay (Hilly, Low Mobility, 2 pkts/sec).....	155
Figure 167: Average End to End Delay (Hilly, Medium Mobility, 2 pkts/sec).....	155
Figure 168: Average End to End Delay (Hilly, High Mobility, 2 pkts/sec).....	156
Figure 169: Average Path Length (Hilly, Low Mobility, 2 pkts/sec).....	156
Figure 170: Average Path Length (Hilly, Medium Mobility, 2 pkts/sec).....	157
Figure 171: Average Path Length (Hilly, High Mobility, 2 pkts/sec).....	157
Figure 172: Number of Packets Transmitted (Hilly, Low Mobility, 2 pkts/sec).....	158
Figure 173: Number of Packets Transmitted (Hilly, Medium Mobility, 2 pkts/sec).....	158
Figure 174: Number of Packets Transmitted (Hilly, High Mobility, 2 pkts/sec).....	159
Figure 175: Number of Overhead Packets (Hilly, Low Mobility, 2 pkts/sec).....	159
Figure 176: Number of Overhead Packets (Hilly, Medium Mobility, 2 pkts/sec).....	160
Figure 177: Number of Overhead Packets (Hilly, High Mobility, 2 pkts/sec).....	160
Figure 178: Packet Delivery Efficiency (Hilly, Low Mobility, 2 pkts/sec).....	161
Figure 179: Packet Delivery Efficiency (Hilly, Medium Mobility, 2 pkts/sec).....	161
Figure 180: Packet Delivery Efficiency (Hilly, High Mobility, 2 pkts/sec).....	162
Figure 181: Throughput (Hilly, Low Mobility, 3 pkts/sec).....	162
Figure 182: Throughput (Hilly, Medium Mobility, 3 pkts/sec).....	163

Figure 183: Throughput (Hilly, Medium Mobility, 3 pkts/sec).....	163
Figure 184: Packet Delivery Ratio (Hilly, Low Mobility, 3 pkts/sec).....	164
Figure 185: Packet Delivery Ratio (Hilly, Medium Mobility, 3 pkts/sec).....	164
Figure 186: Packet Delivery Ratio (Hilly, High Mobility, 3 pkts/sec)	165
Figure 187: Average End to End Delay (Hilly, Low Mobility, 3 pkts/sec)	165
Figure 188: Average End to End Delay (Hilly, Medium Mobility, 3 pkts/sec)	166
Figure 189: Average End to End Delay (Hilly, High Mobility, 3 pkts/sec).....	166
Figure 190: Average Path Length (Hilly, Low Mobility, 3 pkts/sec)	167
Figure 191: Average Path Length (Hilly, Medium Mobility, 3 pkts/sec)	167
Figure 192: Average Path Length (Hilly, High Mobility, 3 pkts/sec).....	168
Figure 193: Number of Packets Transmitted (Hilly, Low Mobility, 3 pkts/sec).....	168
Figure 194: Number of Packets Transmitted (Hilly, Medium Mobility, 3 pkts/sec).....	169
Figure 195: Number of Packets Transmitted (Hilly, High Mobility, 3 pkts/sec)	169
Figure 196: Number of Overhead Packets (Hilly, Low Mobility, 3 pkts/sec)	170
Figure 197: Number of Overhead Packets (Hilly, Medium Mobility, 3 pkts/sec).....	170
Figure 198: Number of Overhead Packets (Hilly, High Mobility, 3 pkts/sec).....	171
Figure 199: Packet Delivery Efficiency (Hilly, Low Mobility, 3 pkts/sec)	171
Figure 200: Packet Delivery Efficiency (Hilly, Medium Mobility, 3 pkts/sec)	172
Figure 201: Packet Delivery Efficiency (Hilly, High Mobility, 3 pkts/sec).....	172
Figure 202: Throughput (Hilly, Low Mobility, 4 pkts/sec).....	173
Figure 203: Throughput (Hilly, Medium Mobility, 4 pkts/sec).....	173
Figure 204: Throughput (Hilly, High Mobility, 4 pkts/sec)	174
Figure 205: Packet Delivery Ratio (Hilly, Low Mobility, 4 pkts/sec).....	174

Figure 206: Packet Delivery Ratio (Hilly, Medium Mobility, 4 pkts/sec).....	175
Figure 207: Packet Delivery Ratio (Hilly, High Mobility, 4 pkts/sec)	175
Figure 208: Average End to End Delay (Hilly, Low Mobility, 4 pkts/sec)	176
Figure 209: Average End to End Delay (Hilly, Medium Mobility, 4 pkts/sec)	176
Figure 210: Average End to End Delay (Hilly, High Mobility, 4 pkts/sec).....	177
Figure 211: Average Path Length (Hilly, Low Mobility, 4 pkts/sec)	177
Figure 212: Average Path Length (Hilly, Medium Mobility, 4 pkts/sec)	178
Figure 213: Average Path Length (Hilly, High Mobility, 4 pkts/sec).....	178
Figure 214: Number of Packets Transmitted (Hilly, Low Mobility, 4 pkts/sec).....	179
Figure 215: Number of Packets Transmitted (Hilly, Medium Mobility, 4 pkts/sec).....	179
Figure 216: Number of Packets Transmitted (Hilly, High Mobility, 4 pkts/sec)	180
Figure 217: Number of Overhead Packets (Hilly, Low Mobility, 4 pkts/sec)	180
Figure 218: Number of Overhead Packets (Hilly, Medium Mobility, 4 pkts/sec).....	181
Figure 219: Number of Overhead Packets (Hilly, High Mobility, 4 pkts/sec).....	181
Figure 220: Packet Delivery Efficiency (Hilly, Low Mobility, 4 pkts/sec)	182
Figure 221: Packet Delivery Efficiency (Hilly, Medium Mobility, 4 pkts/sec)	182
Figure 222: Packet Delivery Efficiency (Hilly, High Mobility, 4 pkts/sec).....	183

LIST OF ABBREVIATIONS

AODV	Ad hoc On Demand Distance Vector
ABR	Associativity Based Routing
CBR	Constant Bit Rate
DREAM	Distance Routing Effect Algorithm for Mobility
DSR	Dynamic Source Routing
DV	Distance Vector
GPSR	Greedy Perimeter Stateless Routing
LAR	Location Aided Routing
LS	Link State
MANET	Mobile Ad hoc Network
OLSR	Optimized Link State Routing
SLURP	Scalable Location Update-based Routing Protocol
STAR	Source Tree Adaptive Routing
TORA	Temporally Ordered Routing Algorithm
TBR	Terrain Based Routing
TBRPF	Topology Based Reverse Path Forwarding
ZRP	Zone Based Routing Protocol

SUMMARY

In recent years, mobile computing has become an integral part of society. As the cost of laptops and wireless networking hardware has declined, society has become increasingly “connected”. In the 1990’s, this “connectedness” was due to cell phones and pagers. In the new millennium, ubiquitous access to the Internet is providing this. High speed wireless internet access is increasingly becoming part of our daily lives. As a result of this dependence on instant access to information, there is a growing need to create these wireless networks without having access to a fixed networking infrastructure. Instead, these mobile nodes can join together to create an ad hoc network to facilitate information sharing. The ad hoc nature of the networks requires different protocols than traditional networks. These mobile ad hoc networks provide a variety of interesting research problems.

The objective of this research is to demonstrate the feasibility of using terrain and location information to improve routing in mobile ad hoc networks through the development of a distributed routing algorithm that uses location and digital terrain information to efficiently deliver packets in a mobile ad hoc network. Through a comprehensive set of simulations, we will show that the new algorithm performs better than current MANET protocols in terms of standard metrics: delay, throughput, packet loss, and efficiency.

This research is motivated by the observation that radio communications are greatly affected by the physical environment. In hilly or urban environments, the range, and indirectly the performance of a wireless network, is much lower than in large open areas. However, MANET protocols typically consider the physical environment only when it causes a change in connectivity. We examine whether the network can estimate

the physical environment and predict its impact on the network, rather than waiting to react to the physical environment.

CHAPTER 1

INTRODUCTION

Mobile ad hoc networks have their roots in the DoD funded packet radio network research (PRNET) from the early 1970's. For the most part, wireless networks were relegated to military systems or to amateur radio operators. The cost of computers and radios was high and the form factors limited people primarily to fixed wireless.

With the phenomenal growth of computing power and reduction of packaging size during the 1990's and 2000's, wireless networks moved out of the research labs and into commercial use. Advances in packaging and miniaturization made laptop computers with WiFi (802.11a/b/g) capability affordable. With the increased availability came increased interest in developing algorithms tailored to the wireless environment.

This has given rise to a new type of network known as a mobile ad hoc network (MANET). Unlike cellular and PCS networks, they require no fixed infrastructure but instead rely on the relaying of packets from one node to another in order to reach a destination. Typically, these nodes have short-ranged radios with limited battery power. The result is a distributed multi-hop network with a dynamic network topology.

1.1 Applications of Ad hoc Network

The primary application for ad hoc networks has been for military operations. As the military has become more dependent on computer networks, it has sought to develop a resilient, mobile networking architecture to support network-centric warfare [4]. Ad hoc networks have also found use in sensor networks, both civilian and military. In the case of sensor networks, the nodes in the networks are typically fixed rather than mobile. In the civilian sector, mobile ad hoc networks have been proposed for use in

collaborative/distributed computing and emergency/disaster recovery operations, as well as in mesh networks and in hybrid wireless network architectures [5].

1.2 Differences Between Wired and Wireless Networks

The nature of the wireless environment (error prone, intermittent connectivity, mobility) is fundamentally different from the traditional wired environment (see [1], [2], [3] & [4] for an overview). Because of this, MANET networks differ from traditional wired networks in several respects. First, MANETs are mobile with links changing at a much different time scale than is typically seen in wired networks. Second, MANETs lack the centralized infrastructure found in fixed wired networks. Rather than having host computers attached to a high capacity network backbone, computers in a MANET are connected to one another in a peer to peer manner. This unique operating environment requires routing protocols specifically tailored to meet these types of challenges.

1.3 Research Objectives

The objective of this research was to demonstrate the feasibility of using terrain and location information to improve routing in mobile ad hoc networks. This was accomplished by developing a distributed routing algorithm that used location and digital terrain information to efficiently deliver packets in a mobile ad hoc network. The new algorithm performs better than current MANET protocols in terms of standard metrics: delay, throughput, packet loss, and efficiency.

This research is motivated by the observation that radio communications are greatly affected by the physical environment. In hilly or urban environments, the range, and indirectly the performance of a wireless network, is much lower than in large open areas. However, MANET protocols typically consider the physical environment only when it causes a change in connectivity. We examine whether the network can estimate

the physical environment and predict its impact on the network, rather than waiting to react to the physical environment.

1.4 Thesis Outline

In CHAPTER 2 we examine how MANET algorithms can be classified. We discuss the different classes of MANET protocols. In CHAPTER 3, we examine in detail several traditional MANET protocols and discuss their strengths and limitations. Details of several geographic/location based routing protocols are discussed in CHAPTER 4. Our Terrain Based Routing Algorithm is described in CHAPTER 5. CHAPTER 6 & CHAPTER 7 discuss the simulation experiments used to compare TBR to DSR, DREAM, & LAR and the results of the experiments. Next, we draw some conclusions and discuss contributions we make to the body of knowledge in CHAPTER 8. Finally, we present several areas of future work.

CHAPTER 2

BACKGROUND AND RELATED WORK

In this chapter we discuss why MANET routing protocols are different from traditional wired routing protocols. After providing a motivation for specialized MANET routing protocols, we discuss how MANET protocols are classified.

2.1 Classification of MANET Protocols

MANET routing algorithms are divided into three broad classes based on how routes are discovered. The three classes are on demand, proactive and hybrid. Table 1 lists some popular routing algorithms and how they are classified.

Table 1: MANET Protocol Classification

Protocol		Classification
Ad hoc On-demand Distance Vector	AODV	On Demand
Associativity-Based Routing	ABR	On Demand
Dynamic Source Routing	DSR	On Demand
Optimized Link State Routing	OLSR	Proactive
Source Tree Adaptive Routing	STAR	Proactive
Temporally Ordered Routing Algorithm	TORA	On Demand
Topology-based Reverse Path Forwarding	TBRPF	Proactive
Zone Based Routing Protocol	ZRP	Hybrid

2.2 On Demand Algorithms

On demand routing algorithms, also known as reactive or source initiated algorithms, discover routes only when nodes need to communicate with one another. This reduces the amount of routing overhead carried by the network as routes are created or maintained only when needed. This is especially important in wireless networks where

the routes are likely to change several times due to node mobility. The tradeoff is a delay while the routing protocol discovers a route to the destination. In addition, the flooding techniques needed to discover routes can consume a significant portion of the available bandwidth. Popular on demand protocols include ABR [6], AODV [7], DSR [8], and TORA [9].

2.3 Proactive Algorithms

Unlike on demand algorithms, proactive algorithms actively maintain routes to every other node in the network. Proactive protocols send periodic updates regarding the status of their outgoing links or when a link's status changes. These updates are typically sent as tables of information, giving rise to the term table driven protocols. In mobile networks where network topology may change frequently, sending out link status updates for unused links consumes limited bandwidth. Because routes are updated so often, much of this information is never used, making the route updates a waste of bandwidth. The benefit to a proactive approach is that routes are always available. There is no delay while a route is established. Thus if network traffic is well distributed or the traffic is delay sensitive, this technique may be appropriate. Popular proactive protocols include OLSR [10], STAR [11] and TBRPF [12].

2.4 Link State versus Distance Vector

Proactive protocols can be further divided into either Link State (LS) or Distance Vector (DV). Link state protocols maintain state information for the entire network by flooding topology information across the network. This creates a large amount of overhead. Each node knows the complete topology of the network, allowing it to compute a route to every node in the network using a shortest path algorithm. Traditionally, this overhead has made link state protocols too inefficient to be used in wireless networks.

However protocols such as OLSR, STAR and TBRPF use different techniques to minimize this overhead.

In contrast, distance vector (DV) protocols only send connectivity information to adjacent nodes. These protocols typically operate using a version of the Distributed Bellman-Ford algorithm. Each node maintains a table (vector) listing each destination, the best-known cost for reaching it, and the next hop to use to get there. When a DV algorithm receives an update from one of its neighbors, it checks to see if the update gives a better path to a destination. If so, it changes the entry in its table and broadcasts its table to its neighbors. In [13], Tenenbaum discusses problems such as convergence time and counting to infinity that affect DV algorithms as well as techniques to mitigate them.

2.5 Hybrid Algorithms

As its name implies, the hybrid class of routing protocols is a combination of reactive and proactive protocols. For nearby nodes, the routing algorithm proactively maintains routes. For distant nodes, the algorithm only determines routes as needed. This combines the best aspects of the two previous classes by minimizing the amount of information broadcast to the network while decreasing the initial route discovery time for many cases. An example of a hybrid protocol is the Zone Routing Protocol [14][15].

CHAPTER 3

MANET ROUTING PROTOCOLS

As discussed previously, researchers have developed a variety of routing protocols specifically tailored to meet the challenges of mobile ad hoc networks. Since the 1990's, many routing protocols have been submitted to the IETF MANET Working Group for consideration for standardization. The literature has several overviews of the proposed protocols [41], [42] & [43]. Two of the most popular are described below.

3.1 Ad hoc On Demand Distance Vector (AODV) Routing Protocol

The Ad-hoc On-Demand Routing Distance Vector (AODV) protocol [7] is a purely on-demand routing protocol supporting both unicast and multicast traffic. AODV borrows some of the broadcast route discovery aspects of the Dynamic Source Routing (DSR) protocol [16] as well as the idea of destination sequence numbers from the Destination Sequenced Distance Vector (DSDV) routing protocol [17]. Unlike traditional distance vector algorithms, however, AODV does not use periodic updates. Instead routes are discovered as needed and expire if unused.

AODV discovers routes using a route request / route reply cycle. When a source node needs a route to a destination for which it does not have an entry in its routing table, it broadcasts a route request (RREQ) packet to its neighbors. These nodes forward the packet on unless they have a "fresh enough" route to the destination. AODV uses destination sequence numbers to determine the "freshness" of routing information. An intermediate node has a valid route if the sequence number for the destination in the intermediate node's routing table is greater than or equal to the destination sequence number in the RREQ packet. As soon as the source receives a RREP, it can start sending data. If the source later receives a RREP containing a greater sequence number for the

destination or the same sequence number with a smaller hop count, the source will update its routing table and use the new route.

Each entry in the routing table has a timer associated with it. Each time a node forwards a packet using an entry, it resets the timer. Thus as long as packets traveling from the source to the destination along a route, the route will be maintained. Once the source stops communicating with the destination, the intermediate nodes will eventually remove the source and destination from their routing tables as the timers expire. Nodes that forwarded RREQ but never received a RREP will timeout and remove the entry in their table for the source.

If a node on the route becomes unreachable while the route is still active, the node upstream of the break sends a route error (RERR) message to the source node to notify it that the destination is unreachable. The RERR message contains a sequence number for the destination that is one greater than the previous sequence number. If the source still needs still needs to communicate with the destination, it increments the previous destination sequence number and broadcasts a new RREQ. Incrementing the destination sequence number eliminates any route replies that might use the old, now invalid, route.

3.2 Dynamic Source Routing (DSR) Protocol

The Dynamic Source Routing (DSR) protocol [8], [16] is an on-demand source routing protocol. In source routing, each data packet contains the full path used to reach the destination. DSR is composed of two processes: route discovery and route maintenance. DSR's route discovery process is a typical query response cycle. If the source's route cache does not contain a path to the destination, the source creates a Route Request (RR) packet and first broadcasts it only to its immediate neighbors. If the source node times out before receiving a response from its immediate neighbors, it again broadcasts the RR, but this time to the entire network.

The Route Request packet identifies the destination the source is trying to reach, the source's identity, a unique "request id", and a record of all nodes the packet has traversed through. When a node receives a RR packet, it first checks to see if it has already seen the packet. If so, the node simply discards the packet and does nothing. Otherwise, if the node is the destination listed in the RR, the node generates a route reply packet. If the node is not the destination, it checks its route cache to determine if it knows a route to the destination. If there is a route in the cache, the node generates a route reply message. If the node does not know how to reach the destination, it adds it itself to the route record in the packet and rebroadcasts the packet. This process continues until the packet reaches the destination or a node with a route to the destination.

Whenever a node transmits a packet, it verifies that the next hop received it. This can be done via passive acknowledgement, explicit acknowledgement or lower layer feedback. If a node cannot verify the packet was correctly received, it sends a Route Error to the packet's source. The Route Error packet contains the address of the node as well as the address of the unreachable next hop. When a node receives a Route Error, it removes any cached routes that contain the link listed in the error message. Since DSR supports multiple routes to a destination, this search could be non-trivial.

DSR supports several techniques that can enhance its performance in both the Route Discovery and Route Maintenance phases by reducing the overhead of DSR by eliminating RREQ floods as often as possible [18]. These extensions include gratuitous route replies, salvaging or rerouting at an intermediate node, and gratuitous errors.

CHAPTER 4

GEOGRAPHIC & LOCATION BASED ROUTING PROTOCOLS

One of the problems with the protocols discussed in the previous chapter is that they do not consider the impact of location information on routing. Instead, information is simply flooded throughout the network (DSR & AODV). In recent years, the availability of cheap Global Positioning System (GPS) devices that can accurately measure location has grown dramatically. As the price and size of GPS devices has decreased, their utility has increased. GPS devices are now found in everything from automobiles to cell phones. Thus it is not unreasonable to assume that future mobile devices will have GPS capability built in. Along with location information, GPS provides accurate time information that can also be exploited.

The increased availability has led researchers to examine the possible benefits of using location information to make routing decisions. The idea of using GPS information for routing purposes was first broached by Dommetty and Jain in a 1996 technical report surveying the potential uses for GPS [19]. They did not develop the idea other than to mention it as a possible use for GPS data.

One of the first uses of location information was for the geocasting class of protocols. The geocasting protocols deliver data to nodes based on their geographical location [20], [21]. The geocasting protocols are more like multicast protocols where group membership is based on a node's location and are designed for applications such as traffic advisories or location specific advertising. One of the first and certainly the most well known is the georouting protocol suite developed by Rutgers University's DATAMAN research group for DARPA's GloMo project. However, since these protocols are not directly relevant to the proposed research, they will not be further considered.

Since the development of the geocasting, several geographic or location based protocols have been developed to move beyond the geocasting paradigm and actually take advantage of the location information to make unicast routing decisions. Several recent papers [22], [23], [24] provide an overview of the current state of the research in location based routing protocols. Much of the work thus far has been focused on selecting the next hop based upon distance information and how to recover from local distance minima (see Figure 5) and not much work has been done on novel algorithmic development.

The majority of the research is based on variations of the four protocols we will examine next. In the following subsections, we examine the four of the most well known location based routing algorithms: DREAM, GPSR, LAR & SLURP. We discuss in detail their operation, their advantages and their limitations.

4.1 Distance Routing Effect Algorithm for Mobility (DREAM)

The DREAM protocol [25], developed by Basagni et al is based on the observation that the farther apart two nodes are, the slower they appear to be moving with respect to each other. DREAM's authors label this observation the distance effect. This observation combined with the basing location updates on mobility rate forms the foundation of the protocol.

Each node running the DREAM protocol maintains a routing table containing all nodes in the network. Periodically, based its mobility rate, each node floods the network with a control packet containing its location information. The control packet also contains a distance "lifetime." Most of the time, the lifetime is short so the location information does not propagate very far. Infrequently, a longer lifetime packet is sent that will reach every node in the network. The protocol designers only used 2 values for the lifetime, but any number could theoretically be used.

When a node receives a control packet, it computes the distance between it and the packet's source. If the distance is less than the lifetime, the node rebroadcasts the packet. Otherwise, it discards the packet. In both cases, the intermediate node updates the source's location in its routing table. Although the designers don't mention maintaining a list of recently processed control packets, each node must maintain a list of nodes within the lifetime distance of the source would continually rebroadcast these packets.

Since the nodes using DREAM periodically broadcast their locations, each node in the network receives this update, albeit some more frequently than others. Therefore, by our discussion in Section 2.1 DREAM is proactive. When node A wishes to send information to B, it uses the location information to calculate a direction to B. A transmits the message to all neighbors in the direction of B.

4.1.1 Protocol operation:

As mentioned previously, each node periodically broadcasts a control packet containing its own location information. To take advantage of the distance effect, each packet is assigned a lifetime based on the geographical distance the packet should travel. The majority of the packets are set to have a short lifetime. This means they will be propagated only a short distance. Infrequently, longer-lived control packets are sent across the network.

To send a packet to a destination, the source determines which of its one-hop neighbors lies in the direction of the destination. This is done computing which neighbors lie in the cone originating at the source and covering the probable location of the destination, as shown in Figure 1. If the maximum speed of the destination is known, it is easy to compute the maximum distance the node could have traveled. Otherwise, for some given probability of finding the destination, the size of the cone can be determined.

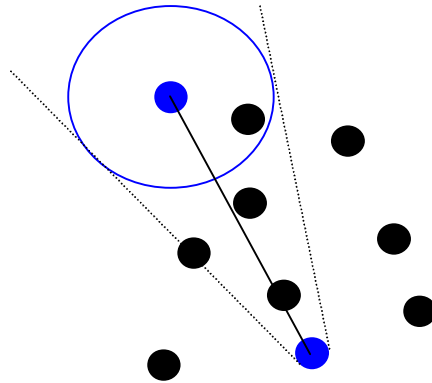


Figure 1: DREAM Routing Cone

When a node receives a packet, it first checks to see if it is the destination. If it is and an acknowledgement is requested, the node sends an acknowledgement back to source. Otherwise, it forwards the packet on to its neighbors that are "in the direction of the destination." Since each neighbor in the direction of the destination forwards packets, there is a good chance multiple packets will be received at the destination. An acknowledgement, if requested, is sent for each duplicate. While this helps ensure that at least one acknowledgement reaches the source, it is not very efficient. This is especially true when forwarding data packets. Each node blindly forwards packets even if it has just completed sending a duplicate. This wastes both energy and bandwidth. A more sensible approach would be to drop duplicate packets.

4.1.2 Properties:

DREAM has several properties that make it attractive. From an overhead perspective, it is bandwidth and energy efficient. Control messages carry only the coordinates and identity of the nodes. The control messages are limited to nodes that need to know exactly where a node is. Nodes farther away do not need to know exactly where a node is located. Also, the rate control messages generated is proportional to

the mobility of the node. If a node doesn't move, it doesn't need to send out location updates.

The protocol is inherently loop-free- messages since messages must propagate away from the source. However, an unspecified recovery procedure must also be loop free to insure loop freedom at all times.

DREAM also is robust. Every data packet is forwarded provided there is a node that has a neighbor in the direction of the destination. This results in multiple copies of the message traversing different routes. This improves reliability but reduces the overall throughput of the network and drains battery power faster.

4.1.3 Limitations:

The DREAM protocol has several problems in its current form. First, the protocol does not specify a recovery procedure when no nodes are in the direction of the destination. One solution to this problem is simply to flood the packet to the destination and this is the approach used in the protocol developers' simulations. There is a second problem related to the uncertainty of a destination's location and the distance between the source and the destination. This problem doesn't manifest itself until the physical (geographical) size of the network becomes large.

If the distance between the source and destination is large relative to the uncertainty in the destination's location, the intermediate nodes must lie along a line between the source and destination. Instead of a cone shaped area, it degenerates into a line. Figure 2 shows how simply doubling the distance between 2 nodes can result in no nodes being "in the direction of the destination." Figure 3 demonstrates how the cone rapidly degenerates into a line as the source-destination distance grows. Therefore, unless a node lies on the line to the destination, the source will be forced to use the recovery procedure.

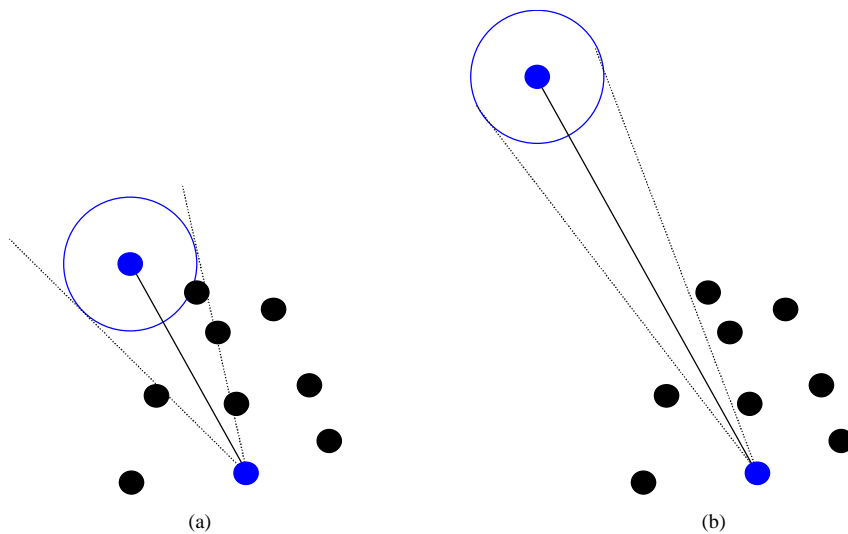


Figure 2: DREAM Forwarding Restrictions based on Distance

The protocol designers did not find this problem because the networks in their simulations were relatively small. The physical size of their network was a 100 x 100 unit square. Each node had a transmission radius of 40 units. Each node sent a packet a distance of 40 units (i.e. typically 2 hops from the source) every 125 clock tics. Every tenth update is flooded throughout the network. This means at the minimum speed of 0.02 units/clock tic, the uncertainty radius will be 25 units at long distances. This is $\frac{1}{4}$ of the maximum distance between nodes, making the cone angle approximately 28 degrees. Compare this with the 2.4 degree cone angle for the longest source-destination distance shown in Figure 3. The cone angle can vary from 28 to 61.5 degrees. Because of the choice in simulation variables values, the cone degeneration did not occur. In practice, it is quite possible that degeneration could occur. In fact, the authors of DREAM acknowledged these problems in discussions with the authors of [44] and recommended that the minimum angle be set to 30 degrees and that traditional flooding be used as the recovery mechanism.

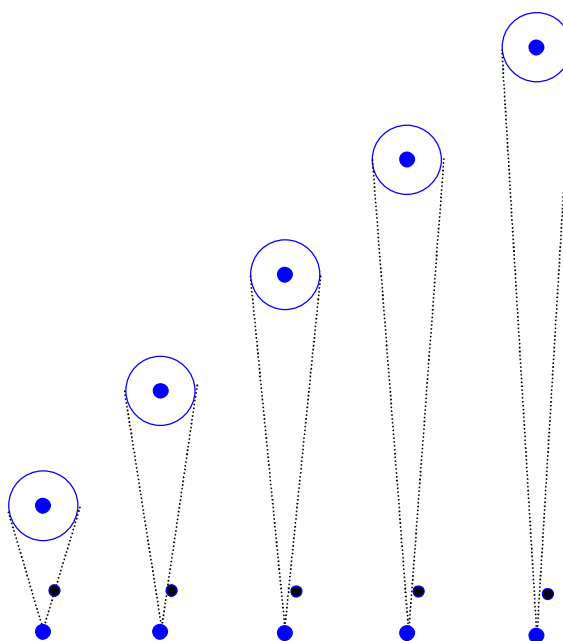


Figure 3: DREAM Cone Degeneration

4.2 Greedy Perimeter Stateless Routing (GPSR)

The Greedy Perimeter Stateless Routing (GPSR) protocol was developed by Karp and Kung. GPSR uses the location of the nodes to make routing decisions. It is a proactive protocol that uses beacons to discover neighboring nodes. The authors claim that the protocol has minimal overhead since only one hop neighbor information is required [27]. However, they assume the existence of a location service that can provide the location information needed for the destination and do not consider the overhead required to provide this service in their analysis.

As mentioned above, each node periodically broadcasts a beacon packet that contains the node's location information. Each node maintains a list of its neighbors and the last time it received a beacon. If a node does not receive a beacon within a timeout interval, it removes the node from its neighbor table. In addition to the beacons, GPSR includes the local node's location information on every packet it forwards. In their

implementation, Karp & Kung optimize the protocol by having nodes transmit a beacon only if they have not forwarded a packet within the beacon interval.

GPSR operates by forwarding packets in one of two ways: greedy forwarding and perimeter forwarding. Perimeter forwarding is only used when the greedy method fails.

When a node has a packet to send, it looks up the destination's location and determines which neighbor is closest to the destination. The node then sends the packet to that node. This process continues until the packet reaches the destination. This greedy forwarding process relies only on local neighbor location information.

However, due to the use only local information, there can be cases which require a packet to temporarily travel farther away from the destination [27][28]. A simple example is shown in Figure 4 where node B is closer to the destination (D) than any of B's neighbors. However, B cannot make anymore forward progress. In this case, greedy forwarding fails.

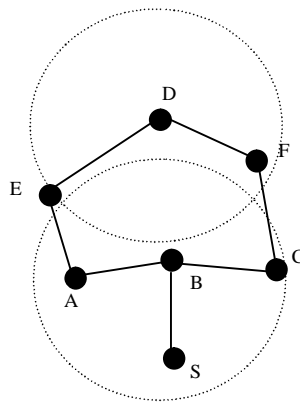


Figure 4: GPSR Greedy Forwarding Failure

To address this case, Karp proposes to route around this local minimum by using the right-hand rule for traversing a graph. As noted in [26] the right-hand rule states that “when arriving at node x from node y, the next edge traversed is the next one sequentially

counterclockwise about x from edge (x,y) .” This causes the void to be traversed in a clockwise edge order. In order to ensure that this approach works in all cases, Karp must planarize the graph. A graph is planar if no two edges cross. Karp uses an algorithm for removing edges from the graph that are not part of the Relative Neighbor Graph (a type of planar graph). A packet remains in perimeter mode until it reaches a node physically closer to the destination. Once this occurs, the packet resumes greedy forwarding.

4.2.1 Properties

GPSR has several properties which make it an attractive solution. It requires minimal state information at each node and control packets are not flooded across the network. Both of these features make the protocol efficient.

4.2.2 Limitations

As described, GPSR has several limitations that reduce its attractiveness. First and foremost, GPSR assumes that an underlying location service exists to provide location information on each node in the network. The periodic beacons only provide location information for a node’s one hop neighborhood. GPSR’s authors do not consider the impact of providing the location service and assume that determining the location of a destination will be a one time cost. However, if the nodes are mobile, the location service will consume a fair amount of bandwidth to maintain accurate location information.

Secondly, there appears to be a discrepancy in how GPSR performs in perimeter node. In [thesis], Karp discusses greedy perimeter probing and states that each node periodically broadcasts a “perimeter probe packet” to recover from greedy forwarding failures. However, this does not seem to be considered in the results later discussed in [26] or in [27]. In both papers, Karp only mentions that the graph is re-planarized when a

node gains or loses a neighbor and that the local neighbor table and information in the data packet together contain all the information required to route.

4.3 Scalable Location Update-based Routing Protocol (SLURP)

Woo and Sing developed an ad hoc routing protocol based on a location management strategy that minimizes routing overhead. SLURP [29] consists of two parts, a location management algorithm and a geographic routing algorithm. Together these two algorithms allow SLURP to maintain approximate location information and find relatively good routes to any destination. The location information is proactively maintained however routes are only discovered as needed.

SLURP assumes all nodes in the network are equipped with a GPS system that provides them with location information. Woo and Sing also assume that the network can be broken up into rectangular regions known as home regions. Further more, they assume a static mapping f that maps a node's ID into a specific home region [26]. Each node is assumed to know the location of each home region. Using the mapping function, each node is assigned a home region that will maintain the node's location information. This is discussed below.

4.3.1 Location Tracking Algorithm

Most of the complexity of SLURP lies in the location management part of the protocol. As mentioned previously, each node is assigned a home region. Whenever a node moves into a new region, it must notify the nodes inside its home region that it has moved by sending a message to its home region. Once this message reaches the home region, it is broadcast to all of the nodes located in the region. Thus every node in the home region knows what region the node is in.

When a node S wants to send a message to another node D , S sends a query to node D 's home region using a location discovery packet to find D 's current location. The

source forwards the packet to the node closest to the destination's home region, using Most Forward with fixed Radius (MFR) [29] without backwards progression. This packet contains the source's ID, current location, a sequence number and the destination's ID. Once the packet reaches the destination's home region, the first node to receive it sends the destination's current location (region) back to the source.

Since all nodes in the network are mobile, there may be times when no node is located in the destination's home region. In order to address the case, all nodes keep a list of the other nodes located in the same region as them. As nodes move in and out of regions, they notify the other nodes in the region. If the last node in a region leaves, it notifies the eight surrounding regions that the region is now empty and passes the location information for all nodes using that region as their home region. When a node enters an empty region, it queries one of the eight surrounding regions to get an updated list.

In order to route packets, each node maintains a location table that contains, for each node in the network, the node's id, location, current region ID and best neighbor for routing to the node.

4.3.2 Approximate Geographic Routing Algorithm

Once the source knows the region the destination is located in, it uses the Most Forward with fixed Radius (MFR) [29] without backwards progression to route the packet to the destination. Since the source only knows the region the destination lies in and not its actual location, the source uses the center of the region as the location of the destination. Once the packet reaches the destination's current region, the packet is source routed to the destination. The first node the packet reaches may or may not know a route to the destination. If it doesn't, it uses a location discovery packet to find a route to the destination. This location discovery packet is limited to the destination's current region. Thus, the first packet uses MFR to reach the destinations region and once inside the

region, source routing is used to route the packet to the destination. SLURP's designers assume that the regions will be relatively small and something like DSR will be appropriate.

4.3.3 Properties:

SLURP's design helps minimize the overhead of route discovery by eliminating a network wide flooding of the discovery message. Additionally, only updating the home region when a node enters a new region reduces overhead.

4.3.4 Limitations:

SLURP has several limitations that make it unattractive for use in actual implementation. First, a mapping function must be developed to convert a node's address to a home region. Next, the size of the home regions must be selected so that there is a good chance at least one node is likely to be in the home region. If the regions are made small relative to the mobility, a significant amount of bandwidth will be wasted with frequent location updates as nodes move into new regions. The location dissemination functionality creates extra overhead. Another problem with SLURP is that it relies on the presence of an additional protocol to route inside the regions. These shortcomings make it somewhat impractical for use in real networks.

4.4 Location Aided Routing (LAR)

Location Aided Routing (LAR) [31],[32],[33], proposed by Ko and Vaidya, attempts to reduce routing overhead by limiting packet flooding based on location. LAR assumes each node has accurate location information provided by a GPS system. Ko and Vaidya propose two slightly differing routing schemes: LAR Scheme 1 and LAR Scheme 2. Both schemes use the classical flooding routing approach, but restrict the flood to specific a geographic area. It is important to note that LAR is really only a route

discovery technique. Ko and Vaidya consider LAR to be a replacement for the route discovery phase of a MANET protocol such as DSR [31].

Before we discuss the two schemes, we will examine their common features. LAR works by estimating a circle, known as the expected zone, which contains the destination. The radius of the expected zone is determined by estimating the distance the destination traveled since the source last learned the destination's location. Thus the expected zone is a circle with radius $v(t_1-t_0)$ centered at the last known location of the destination. With t_1 equal to the current time, t_0 equal to the time of the last known location and v the average speed of the destination.

Once the source had determined the expected zone of the destination, it must flood the packet to every node in the expected zone with the hope that the destination lies in that zone. In order to get the packet to the expected zone, the source computes a request zone. The request zone covers nodes that lie between the source and the destination. Any node that receives a packet and lies within the request zone described in the packet forwards the packet again. Any node outside the expected zone merely drops the packet and does nothing. This limits the flooding to the region network where the destination is likely to be located.

4.4.1 LAR Scheme 1:

In LAR 1, the request zone is rectangular in shape [32]. The rectangle is sized to contain both the source and the expected zone of the destination. The source broadcasts a data packet and includes the coordinates for each of the four corners of the request zone. When an intermediate node receives the packet, it checks to see if it is located inside the request zone and, if so, rebroadcasts the packet. If the node is outside of the zone, it simply discards the packet. Like traditional flooding approaches, duplicate packets are discarded (via unique packet identifiers). When the destination receives the route discovery packet, it sends a route reply to the source and includes its current location

along with a timestamp. Thus the expected zone and request zone can be kept relatively small while the source and destination are communicating regularly.

4.4.2 LAR Scheme 2:

LAR Scheme 2 expands upon LAR Scheme 1 by attempting to further restrict the number of nodes inside the request zone that transmit the packet. In addition to the location of the four corners of the request zone, each discovery packet also contains the location of the destination as well as the distance from the source to the destination.

When an intermediate node receives the packet, it computes its distance to the destination using the location information in the packet. It compares this to the distance listed in the packet. If the node's distance is less than the distance in the packet plus a delta value (i.e. $DIST_{int_node} < DIST_{pkt} + d$), the node will put its distance to the destination in the packet and retransmit it. This means that only packets traveling toward the destination will continue to propagate. There is a problem with this technique. If there is no route that follows the shortest distance, route discovery will fail. Figure 5 demonstrates two cases where the technique fails. In the first case, when node B receives a packet (for D) from A, it simply discards it since its distance from D is greater than A's. Figure 5b shows a slightly different case where node C is farther away from D than B is.

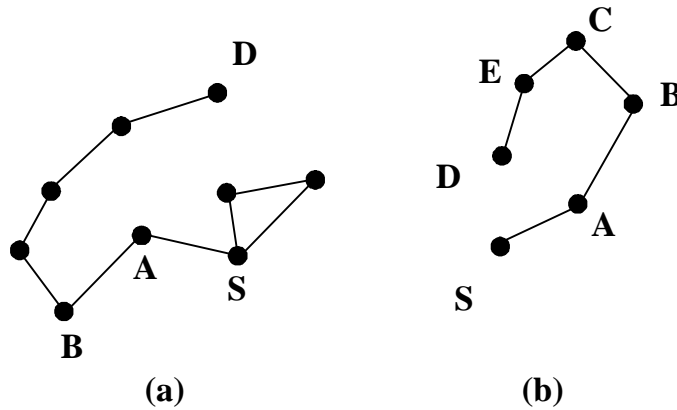


Figure 5: LAR Scheme 2 Failure

4.4.3 Properties:

LAR's design helps minimize the overhead of route discovery by eliminating a network wide flooding of the discovery message. Instead, discovery messages are limited to the area the destination is likely to be located. Unlike, SLURP, LAR does not require an active location management service.

4.4.4 Limitations:

LAR has a few limitations. Its primary limitation is the fact that it is a route discovery process rather than a routing algorithm. It is really a replacement for DSR's discovery process. Also, as mentioned in 4.4.2, some topology configurations can cause LAR's discovery process to fail. While the failure is only temporary, and can be fixed with a suitable recovery process, it can cause additional overhead. Also, in networks with low mobility, putting location information in every packet can cause significant amount of unnecessary overhead.

CHAPTER 5

TERRAIN BASED ROUTING PROTOCOL

In this chapter, we discuss a mobile ad hoc network routing algorithm that uses both terrain and location information to make routing decisions. As mentioned earlier, the objective of the research was to reduce the routing overhead by predicting link connectivity through estimation of the path losses between nodes. From this estimation, we can determine the optimal route. Through simulation, we determined the effectiveness of predictive routing in ad hoc networks using terrain and location information, determined bounds on the acceptable propagation loss prediction error, and developed a new geographic routing protocol.

This research is particularly suitable for bandwidth-constrained networks or for networks where channel access is expensive. Unlike other ad hoc routing protocols, we do not require periodic beacons or messages. The overhead incurred by sending location information is not truly wasted if the nodes could use that information to improve their situational awareness, particularly a military communications network where this information could be leveraged for use by the both the application and network layers in a cross layer design.

In the following sections, we first provide an overview of the proposed algorithm and follow that with details of key areas of the algorithm.

5.1 Algorithm Overview

None of the protocols discussed in CHAPTER 4 make use of terrain information. We propose an algorithm that uses terrain and location information to improve routing performance of ad hoc networks. We demonstrate that terrain knowledge and location

information can be used to develop a scalable efficient routing algorithm for ad hoc networks.

Our proposed TBR algorithm is an on demand algorithm. TBR does not attempt to actively maintain routes to all nodes in the network. Instead, it only maintains routes to nodes when required. It does however opportunistically learn routes and maintain location information in order to reduce or eliminate the initial route acquisition delay associated with on demand protocols. The algorithm is divided into three phases of operation: *route discovery*, *packet forwarding*, and *route maintenance*.

When the routing layer receives a packet for a destination, it checks to see if it knows the next hop the packet should take to reach the destination. If there is one, it simply sends the packet to the next hop toward the destination.

When a node can no longer reach the next hop toward the destination (i.e. a link failure) or the node determines that the next hop is the node it just received the packet from (i.e. a loop has formed), the algorithm enters the route maintenance phase. In the route maintenance phase, the algorithm first recomputes the route to the destination to ensure that the most updated location information is used in the route computation process. If the algorithm still cannot find a route or if the loop exists, it sends a route maintenance packet back toward the originator of the packet.

5.1.1 Routing Table:

The routing table maintained by the TBR algorithm contains more information than is typically found in a routing table. Figure 6 shows the fields in a TBR routing table entry. Several fields in the table are not typically found in a routing table. The *latitude & longitude* used to track the location of the nodes. We use the *heading* and *velocity* fields to store the estimated heading and velocity of the node. This is later used to predict the location of a node when computing the routing table. The *Update Time*

field stores the time this entry was last updated and the *Expire Time* field stores the time when the entry should no longer be considered valid.

Destination	Next Hop	Cost	Latitude	Longitude	Heading	Velocity	Update Time	Expire Time
-------------	----------	------	----------	-----------	---------	----------	-------------	-------------

Figure 6: Routing Table Entry

5.1.2 Route Computation:

Periodically and in response to route maintenance messages, the algorithm recomputes routes to all nodes found in the routing table using dijkstra’s algorithm [35]. Before dijkstra’s algorithm can be used however, we must first determine connectivity or link state information. Traditionally, this has been done through network wide broadcasts that are flooded through the network. In our approach, we use the terrain and location information to predict the connectivity/link state information thereby reducing the overhead.

Rather than just use the last known latitude and longitude for each node, we attempt to predict each node’s mobility. Whenever a node receives new location information for any node, determines the velocity and heading of the node using the location information currently stored in the table and the new location information. Given a pair of latitudes and longitudes and the time taken to travel between the two points, we compute the velocity and heading that the node is traveling. Whenever we compute the routing table, we use the current location information and the velocity and heading to estimate each node’s current location. A simple linear prediction model is used. Using the speed the node is traveling and the amount of time since the location was updated, we compute the distance as:

$$\text{distance} = v_{\text{estimate}} * (t_{\text{current}} - t_{\text{update}}) \quad \text{eq 1}$$

Using the heading information and basic trigonometry, we can translate the distance traveled into a change in latitude and longitude. We then add the estimated change in latitude and longitude to our last known position to determine the node's predicted location.

After predicting the current locations for all of the nodes, the TBR algorithm creates a connectivity matrix by computing the path loss between all nodes in its routing table using the predicted locations and a radio propagation model. Any path loss value above a threshold value is set to infinity to signify that no connection is present. As will be discussed in Section 5.1.7, the matrix is then modified to remove links that the route maintenance process has determined have failed.

Dijkstra's shortest path algorithm is then run using the updated connectivity matrix. The shortest route is the route with the least total path loss. This is not always the shortest route in terms of the number of hops required to reach the destination. Our approach favors shorter links that are more likely to have a successful reception. Borrowing from ABR [6] and SSA [34], we try to choose links are also more likely to remain stable.

5.1.3 Propagation Channel Model:

As mentioned previously, instead of transmitting neighbor information, we send only location information (latitude & longitude) and a timestamp as part of each packet. Using digital terrain information and a radio propagation model, we can compute connectivity information. This yields a connectivity matrix from which a routing table can be generated. Any channel propagation model can be used. The more accurate the model the more efficient the routing becomes. For this research, we used the Terrain-Integrated Rough-Earth Model (TIREM) produced by the United State Department of Defense's Joint Spectrum Center [36][37]. TIREM is the DoD's standard propagation

model. However, the results of this research serve to illustrate the viability of the approach and are not limited to a particular propagation model implementation.

5.1.4 Medium Access Control Layer:

The TBR algorithm assumes that the underlying MAC protocol provides a reliable delivery service (e.g. 802.11b). If the MAC is unable to deliver the packet, it must notify the routing layer. Other than the requirement for reliable delivery, TBR does not care about the underlying datalink/MAC. Although not considered in this research, TBR could be used in a cross-layer design [39][40] where it gets access to the received signal strength of the packet. TBR could then be modified to compare its predictions to what is happening in the real system and react appropriately.

5.1.5 Route Discovery:

When the routing layer receives a packet for a destination from the upper layers, it checks the routing table for a valid entry and sends the packet on to the next hop listed in the table if there is one. Included in the packet is the location information of the node (latitude, & longitude) and a time stamp. If there is no valid entry, the routing layer queues the packet and broadcasts a route discovery message to the network.

In the route discovery message, the node lists its location, the current time and the destination address (Figure 7). As this packet propagates out, each node adds its location information to the packet and uses the location information contained in the packet to update its routing table before rebroadcasting the modified packet. This process continues until the packet reaches the destination. The destination extracts the location information in the discovery packet creates a route reply packet

Initially, we allowed intermediate nodes to reply if they had a route to the destination. However, since the nodes returned the last location information they received rather than the predicted location information which the intermediate nodes

based their routing decisions on, a lot of overhead was generated with little improvement in performance. Sending the predicted location meant that nodes would be mixing actual and predicted location data with no way of discriminating between the two. This would also impact the mobility prediction process. As will be shown, the major impact of this design decision is a slight increase in average delay.

A side effect of allowing only the destination to respond is that the discovery message will propagate throughout the network. While this increases the amount of overhead, it also distributes up to date location information which nodes can use later. This will hopefully lead to a future reduction in the number of discovery messages needed.

Type	TTL	Destination Address		Number of Address Entries (n)
Source Address	Source Latitude	Source Longitude	Source Time	
Address 1	Address 1 Latitude	Address 1 Longitude	Offset Time 1	
⋮	⋮	⋮	⋮	
Address n	Address n Latitude	Address n Longitude	Offset Time n	

Figure 7: TBR Route Request Packet Format

When the route discovery packet reaches the destination, the destination node unicasts a Route Reply packet back to the source. The route reply message contains the location information for each node in the path from the source to destination. The route in the reply packet is the reverse of the route in the discovery packet. As the route reply packet propagates back to the source along the path listed in the packet, the intermediate nodes extract the location information from the packet and update their routing tables with the location information. Since the route reply contains the full route and location

information from the source to the destination we can ensure that the source, as well as all of the nodes along the route, will be able to compute at least one path to the destination after processing the route reply.

Figure 8 shows the fields of the Route Reply Packet. The packet contains the addresses, location and a timestamp for each node along the path. The packet also contains field for the number of entries in the list as well as a field for the current position in the list which nodes increment as the packet moves toward the destination (i.e. the originator of the discovery packet).

Type	TTL	Source Address	Destination Address
Number of Address Entries		Position in list	
Source Address	Source Latitude	Source Longitude	Source Time
Address 1	Address 1 Latitude	Address 1 Longitude	Offset Time 1
:	:	:	:
Destination Address	Destination Latitude	Destination Longitude	Destination Offset Time

Figure 8: TBR Route Reply Packet Format

5.1.6 Packet Forwarding:

When a node receives a packet, it looks up the destination listed in the packet in the node's routing table and determines the next hop. The node then forwards the packet to the next hop. If the routing table entry for the destination is invalid, the node attempts a modified route discovery process, which is described in the Route Maintenance section below. At the same time, the node updates the packet source's location in its routing table using the information found in the packet. The format of a data packet is shown in Figure 9.

Type	TTL	Size	Destination Address	Source Address
Source Latitude		Source Longitude		Source Timestamp
Data				

Figure 9: TBR Data Packet Format

5.1.7 Route Maintenance:

Since the algorithm routes by predicting link connectivity, there will be times when the prediction is incorrect or a node has suddenly dropped out of the network. When a node is unable to forward a packet along a link that should exist, it first marks the link as down by adding the link (identified by the addresses of the end nodes) to a table of failed links (Figure 10). The node then recomputes the connectivity matrix and ensures any links in the failed link list are removed from the connectivity matrix. The node then attempts to recompute a new route to the destination.

Upstream Address	Downstream Address	Time of Failure	Expire Time
------------------	--------------------	-----------------	-------------

Figure 10: Failed Link Table Entry

If an alternate path is found and it does not require the packet to backtrack toward the source of the packet (i.e. the new next hop is NOT the same node that transmitted the packet to the current node), the node simply forwards the packet on to the new next hop.

If the algorithm determines that the next hop must backtrack, the node generates a Route Maintenance (RM) packet. The RM packet is addressed to the previous hop, and, as seen in Figure 11, it contains the list of failed links (upstream address & downstream address) that the node is aware of and optionally the undeliverable packet. The RM packet is sent to the new next hop toward the destination. Since the packet is

backtracking, TBR encapsulates the data packet and uses the RM packet to signal the new next hop that it must recompute its routing table based on updated information to prevent a loop from occurring.

Type	TTL	Size	Destination Address	Source Address
Source Latitude		Source Longitude		Packet Flag
Failed Link Upstream Address ₁		Failed Link Downstream Address ₁		
⋮		⋮		
Failed Link Upstream Address _n		Failed Link Downstream Address _n		
Data Packet				

Figure 11: TBR Route Maintenance Packet Format

If the algorithm fails to find a new route to the destination, it generates a Route Maintenance (RM) packet that is addressed to the originator node listed in the source field of the data packet. As in the previous case, the RM packet contains the list of failed links (upstream address & downstream address) that the node is aware of and optionally the undeliverable packet. The RM packet is then forwarded on toward the originator of the undeliverable packet.

As the RM packet makes its way back to its destination, the intermediate nodes process the list of link failures and update their local failed links table. Each node attempts to find an alternate route in its table for the encapsulated data packet. If it finds a new path that does not involve backtracking, the node extracts the data packet from the RM packet and forwards the data packet on and discards the RM packet. If the RM packet is backtracking toward the source of the data packet, it forwards the RM packet on.

An example of route maintenance process is illustrated in Figure 12. In the figure, the heavier line shows the path based on predicted connectivity. When the packet reaches node A, the predicted link between A & D is not actually present. Node A updates its failed link table and computes an alternate route. Since the new route involves backtracking, Node A generates a RM message, listing the A-D link as failed and includes the data packet. This packet is forwarded back to node A's upstream neighbor. When the upstream neighbor receives the packet, it recomputes the route and forwards the original data packet along the new path.

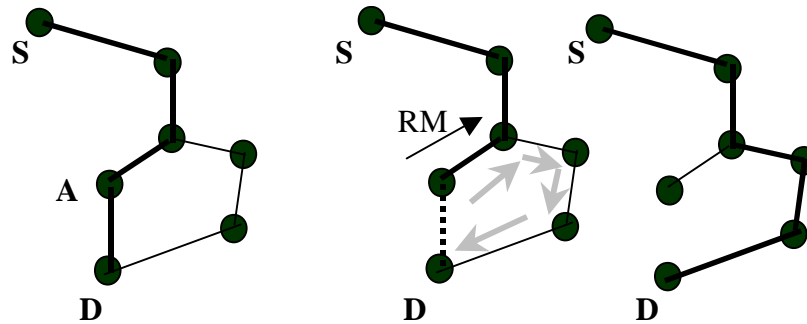


Figure 12: Route Maintenance

CHAPTER 6

SIMULATION EXPERIMENTS

Two sets of simulation experiments were developed to evaluate TBR's performance. The first set of simulations characterized the performance of TBR in the presence of prediction errors. The second set of simulations compared TBR's performance, with and without prediction errors, to the DSR, DREAM and LAR protocols. The simulations were performed using the OPNET simulation tool version 10.5 with TIREM v 3.15 as the propagation model. Details of the simulation experiments are discussed below. Performance measures include routing overhead in packets, average end to end delay, packet delivery ratio, packet delivery efficiency, and average path length for delivered packets.

6.1 Simulation Overview

All of the experiments were made using two different types of terrain: flat and hilly. As its name implies, the flat terrain was perfectly flat. This is the typical type of terrain found in most MANET simulations. The hilly terrain contained a mixed of flat and rolling hills. For both terrain types, the simulation area was set to a 5.6 km by 4.6 km rectangle. This simulation field is significantly larger than the simulation areas typically used in MANET simulations. The large area is needed to ensure that the terrain will impact the connectivity of network. The simulation time for each run was 600 seconds.

For both terrain types, we used the random waypoint mobility model [48] to model the movement of the nodes in the network. In this model, each node chooses a random destination location and speed to travel. Once the node reaches the destination, it pauses for a random amount of time and then chooses a new destination and speed. In all

scenarios, the pause time was randomly selected between 0 to 60 seconds. Three different mobility scenarios were run using different maximum speeds: low (5 m/s), medium (15 m/s) and high (25 m/s). The speed was randomly chosen over the interval $[1, \textit{maximum speed}]$ meters per second based on the mobility rate for the scenario.

In both sets of experiments, the MAC layer used was a .1 persistent CSMA with a simple ACK based reliable datalink on top to provide the reliable delivery service required by DSR, LAR and TBR. In addition, the MAC layer was set to promiscuous mode so all network traffic was sent to the routing layer. This enabled, DSR, LAR and TBR to gather information from packets not addressed to them. The transmit power for each radio was set to 125 milliwatts providing a nominal transmission range of 800 meters. The radio on each node had a channel rate of 4 Mbps with a transmit frequency of 2.4 Ghz.

6.2 Performance Metrics

Table 2 summarizes the performance metrics used to evaluate TBR, DREAM, DSR, and LAR. The metrics are defined as:

Packet Delivery Ratio: The total number of application layer data packets received (excluding duplicates) divided by the total number of application layer data packets offered.

Throughput: The total number of application layer data bits received (excluding duplicate packets) divided by the simulation length (600 seconds). Since the packet sizes are fixed, the throughput can be calculated by taking the total number of data packets received, multiplying by 2048 bits (256 bytes * 8 bits/byte) and dividing by the length of the simulation (600 seconds).

Average End to End Delay: The sum of the delays experienced by each application data packet delivered divided by the number of application layer packets delivered.

Control Packet Overhead: The total number of control packets the routing protocols sent to the lower layers.

Control Bit Overhead: The total number of control bits transmitted including overhead bits in the network layer data packets and in the control packets.

Number of Data Packets Delivered: The total number of application layer data packets successfully delivered to their destination, excluding duplicates.

Number of Data Bits Delivered: The total number of application layer data bits successfully delivered their destination, excluding bits in duplicate packets. Since the application layer data packets are fixed at 256 bytes, the number of bits delivered is equal to *Number of Packets Delivered* * 2048.

Average Path Length: The average number of hops that the application data packets took to go from a source to a destination. It is computed by adding up the number of hops that each delivered data packet took and dividing by the total number of data packets that were delivered.

Packet Delivery Efficiency: The total number of application layer data packets delivered divided by the total number of packets transmitted. The total number of packets transmitted is the sum of the number of data and control packets sent to the lower layers.

Bit Delivery Efficiency: The total number of application layer data bits delivered divided by the total number of bits transmitted. The total number of bits transmitted is the sum of the number of data and control bits sent to the lower layers.

Table 2: Performance Metrics

Metric	Unit of Measure
Packet Delivery Ratio	%
Throughput	Bits per Second
Average End to End Delay	Seconds
Control Packet Overhead	Packets
Control Bit Overhead	Bits
Number of Data Packets Delivered	Packets
Number of Data Bits Delivered	Bits
Average Path length	Hops
Packet Delivery Efficiency	--
Bit Delivery Efficiency	--

6.3 Experiment 1: Effects of Path Loss Prediction Error on TBR Performance

The first set of simulations was designed to determine how accurate the propagation model must be for the terrain-based routing algorithm to function effectively. Since both the TBR algorithm and the OPNET simulation environment use the same propagation model, the propagation prediction made by TBR would always match the channel conditions in the simulation. This is obviously not a realistic assumption in an actual system. Therefore, we needed to introduce random errors into TBR's computed path loss prediction in order to get a realistic assessment of TBR's performance in an actual implementation. These errors helped us determine the algorithm's sensitivity to prediction errors.

Each time the algorithm computed the path loss between two nodes, it modified the result it received from the propagation model as given below in equation 2. In the equation, X is the variance of a normal distributed random variable with zero mean for the simulation run and the path loss is in decibels.

$$PathLoss = PathLoss + N(0, X) \quad \text{eq 2}$$

Using the above, we examined how errors in path loss prediction degrade the performance of the routing by varying variance over a variety of different values. We

started with no error and increased the variance in 1 dB increments through 7dB and then in 5 dB increments from 10 dB to 20 dB. The network size was set to 60 nodes, the mid point between (20 and 100 nodes). Thirty source-destination pairs were randomly assigned and each source set to transmit 2.5 packets per second. The packet size was fixed at 256 bytes and the CBR flows were configured as described in Section 6.4.

The experiment was performed first using flat terrain and then again using hilly terrain. For both terrain types, each node moved using the random waypoint mobility model with a maximum speed of 5, 15, and 25 meters per second. For each data point, 10 different runs were performed and the average of the runs is presented. For each run, the nodes are given randomly distributed initial locations. The simulation parameters are summarized in Table 3.

Table 3: Path Loss Prediction Error Simulation Parameters

Parameter	Value
Network Size	60 nodes
Number of Nodes Transmitting	30 nodes
Packet Rate (per node)	2.5 packets / second
Packet Size	256 bytes
Terrain Type	Flat Hilly
Mobility Model	Random Waypoint Mobility
Maximum Speed	5 m/s 15 m/s 25 m/s
Number of random seeds	10

6.4 Experiment 2: Comparison of TBR to DSR, DREAM, and LAR

While the experiment in section 6.3 provided some performance data, it does not demonstrate that TBR offers superior performance to current algorithms. In order to show this, we compared the performance of TBR to other MANET protocols. We chose the DSR protocol because of its popularity and the availability of an OPNET model [45] & [46]. A careful review of the literature from [41], [42], and [43] shows that whether AODV or DSR is superior depends on the choice of simulation parameters. In general, the performance is comparable if the simulation scenarios are not biased toward one or the other. Thus, DSR provides a sufficient comparison baseline. LAR and DREAM were chosen since both are location based routing algorithms.

6.4.1 Protocol Implementations

We used the DSR implementation developed by NIST and further modified Major Rusty Baldwin at the Air Force Institute of Technology [45]. Since LAR is basically a route discovery optimization rather than a full routing protocol, we implemented LAR by modifying the NIST DSR implementation to include location information each packet. The DSR implementation's route discovery functions were modified to limit the flooding of the discovery packets using LAR Scheme 1. The route discover packet was modified to include the request zone information [33]. We implemented the DREAM protocol based on the details given in original paper [25] and set the minimum angle to 30 degrees. The TBR protocol was implemented as described in CHAPTER 5.

Table 4 lists the different scenario parameters that were varied during Experiment 2. These are similar to the parameters used in Experiment 1 with the addition of varying the traffic rate and network size. The comparison experiments were designed to simulate a wide variety of realistic operating environments. For TBR, we ran the algorithm with an error variance of 6 dB. This was the lower bound on acceptable performance seen in the results of Experiment 1.

Table 4: Scenario Parameters

Parameter	Value
Network Size	20, 30, 40, 50, 60, 70, 80, 90, 100 nodes
Number of Nodes Transmitting	Network Size / 2
Protocols	TBR-0, TBR-6, DREAM, DSR, LAR
Packet Rate (per node)	1, 2, 3, 4 packets / second
Packet Size	256 bytes
Terrain Type	Flat, Hilly
Mobility Model	Random Waypoint Mobility
Maximum Speed	5 m/s, 15 m/s, 25 m/s
Number of random seeds	10

Taking all possible combinations of the network size, terrain type, speed and rate parameters listed in Table 4 yields 216 different scenarios. For each scenario, we compared the performance of the five MANET routing algorithms (DSR, LAR, DREAM, TBR-0 and TBR-6) in each of the 216 scenarios. For each scenario, we gathered the metrics listed above in Table 2.

The traffic loading started with 10 source-destination (S-D) pairs and increased as the network grew, with an additional 5 S-D pairs per 10 node increase in network size. This translates into half the nodes in the network transmitting. During these experiments the traffic sources were modeled as constant bit rate (CBR) traffic with a packet size of 256 bytes. To prevent the CBR traffic flows from being synchronized, the start times were offset by a random amount over the interval 0 to 1 second.

The goal of this experiment was to compare the performance to current approaches and confirm that TBR provides improved performance. These experiments

also show the scalability of TBR to large numbers of nodes. In order to gather statically significant results, each scenario was run with 10 different random number seeds. The results we present in CHAPTER 7 are the average of these 10 different runs.

CHAPTER 7

RESULTS

This chapter presents the results for the simulation discussed in the previous chapter. A total of 11,460 individual simulation runs were performed. Experiment 1 had 660 individual simulation runs and Experiment 2 had the remaining 10,800 runs. The simulation runs times varied tremendously from approximately 1 minute per simulation run for the 20 node low traffic rate simulations to over 5 hours per simulation run for the high traffic rate, high mobility, 100 node scenarios. For both sets of experiments, we focus our analysis on the performance metrics listed in Table 5. While the initial plan was to examine several metrics in terms of bits, the results did not provide any additional insight. Since the packets are fixed size, the bit and packet metrics had almost identical graphs with one exception.

In Experiment 2, DSR in many cases transmitted less overhead in terms of bits. This was primarily due to the fact that LAR and TBR both include location information in every packet. However, DSR typically transmitted more overhead packets. As the network and traffic grew, DSR lost its advantage in bits. Because of this, we elected to restrict our overhead discussion to packet overhead.

The rest of the chapter is organized as follows. Section 7.1 provides a summary of the results of Experiment 1: Effects of Path Loss Prediction Error on TBR Performance. Section 7.2 discusses the results of Experiment 2: Comparison of TBR to DREAM, DSR, and LAR.

Table 5: Performance Metrics

Metric	Unit of Measure
Packet Delivery Ratio	%
Throughput	Bits per Second
Average End to End Delay	Seconds
Control Packet Overhead	Packets
Average Path length	Hops
Packet Delivery Efficiency	ratio

7.1 Experiment 1: Effects of Path Loss Prediction Error on TBR Performance

As mention in the previous chapter, the first set of simulations is designed to determine how accurate the propagation models must be for the terrain-based routing algorithm to function effectively. We examined two types of terrain (flat and hilly) and three mobility speeds: low (5 m/s), medium (15 m/s) and high (25 m/s). The packet rate was fixed at 2.5 packets per second. The network size was fixed at 60 nodes, 30 nodes of which were sending CBR traffic to randomly selected destinations. For each of these scenarios, we examined the impact on performance as the amount of prediction error was increased from 0 dB to 20dB. In the following sections, we discuss the effects of path loss prediction error on TBR performance in flat and hilly terrain.

7.1.1 Effects of Prediction Error in Flat Terrain

As expected, the performance of TBR degrades as the prediction error increases. The throughput and packet delivery ratio metrics show a relatively linear decrease in performance. Figure 13 shows that the low and medium mobility cases follow a relatively constant decline. The high mobility case differs slightly, having a much less linear performance drop. The high mobility case appears to alternate between periods of gradual decline (3-5 dB, 15-20 dB) and sharper decline (0-1 db, 5-6 dB, 10-15 dB). Given the confidence interval bounds, the statistical significance of these observations is

minor. For all three mobility cases, the decrease in performance from perfect prediction to 20 dB error is about the same.

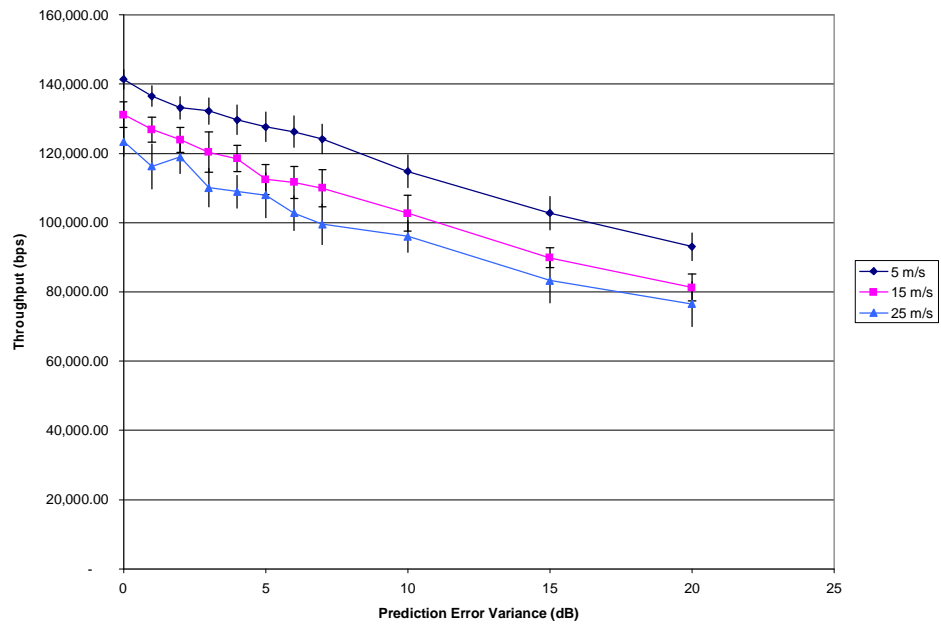


Figure 13: Throughput in Flat Terrain

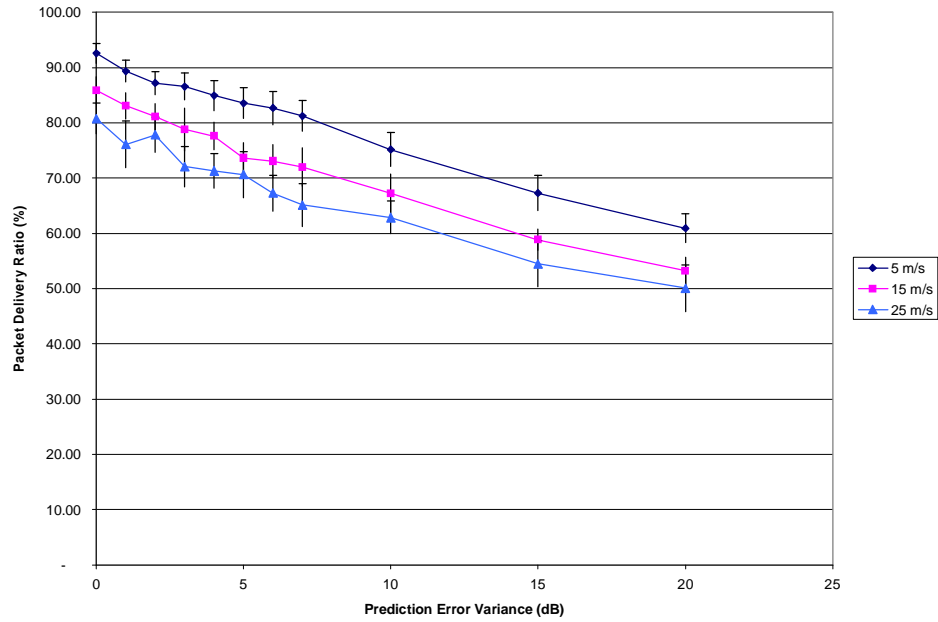


Figure 14: Packet Delivery Ratio in Flat Terrain

Figure 15 shows the average end to end delay as a function of the prediction error variance for the three mobility cases. The low mobility case shows gradual growth in average end to end delay as the error increases. This is due to the gradually increasing average path length. In contrast, the medium and high mobility cases show a relatively little increase in delay until the error variance reaches 7 dB. After 7 dB, all three mobility cases show the large increase in delay. The relative insensitivity to small prediction errors is due to the interactions between the mobility prediction algorithm and the error variance. At medium to high mobility rates, the errors in location prediction cancel out the effects of the prediction error. The low mobility rate's lower delay results from having more stable links. In the medium and high mobility cases, the mobility creates frequent breaks in the routes causing additional delay as routes are reestablished.

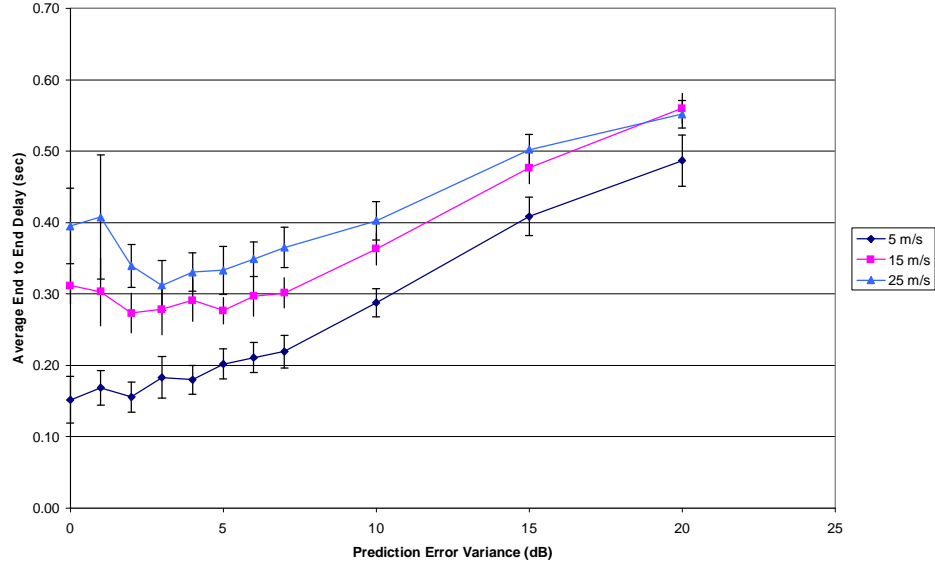


Figure 15: Average End to End Delay in Flat Terrain

Figure 16 shows the average path length as a function of the prediction error. As mentioned previously, at low mobility, the average path length gradually increases until the prediction error reaches 7dB. After that, the average path length starts to decrease. As the error increases, TBR is unable to support longer path lengths. Between, 0 and 7dB, the error causes less than optimal routes resulting in an increased average path length.

In the medium and high mobility cases, the average hop count remains relatively flat until the prediction error reaches 7dB. At 7 dB, all three mobility cases share the same average hop count and all show the same decrease as the error is increased to 20 dB.

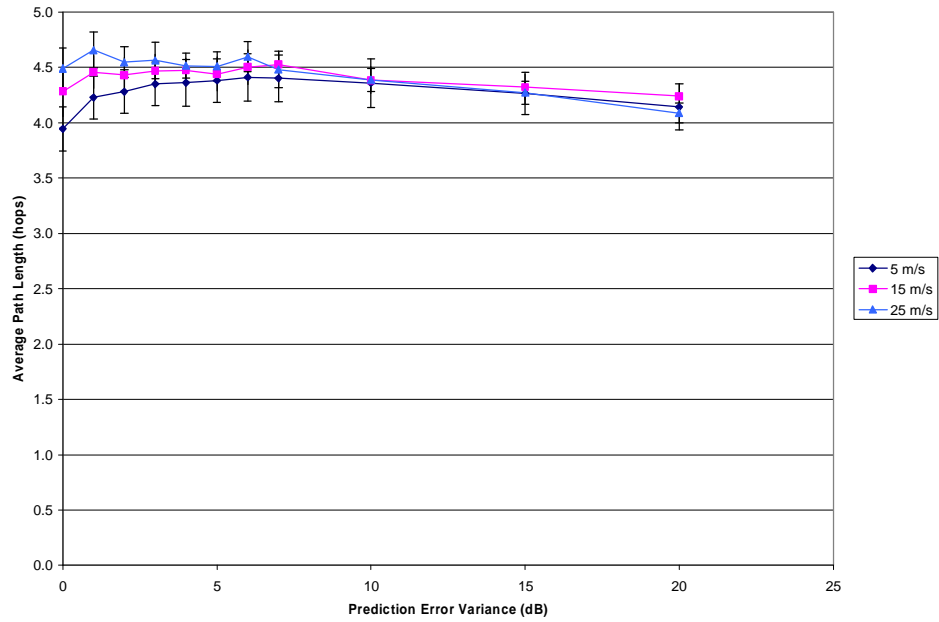


Figure 16: Average Path Length in Flat Terrain

Figure 17 shows the effects of prediction error on the number of overhead packets generated by TBR. While the difference in overhead between the low mobility case and the medium and high mobility case is significant, the difference between the medium and high mobility cases is much smaller. This is due in part to the fact that the drop in the average number of neighbors is greater going from low to medium mobility than when going from medium to high. Also, the prediction error has a greater impact on the overhead than the increase in mobility.

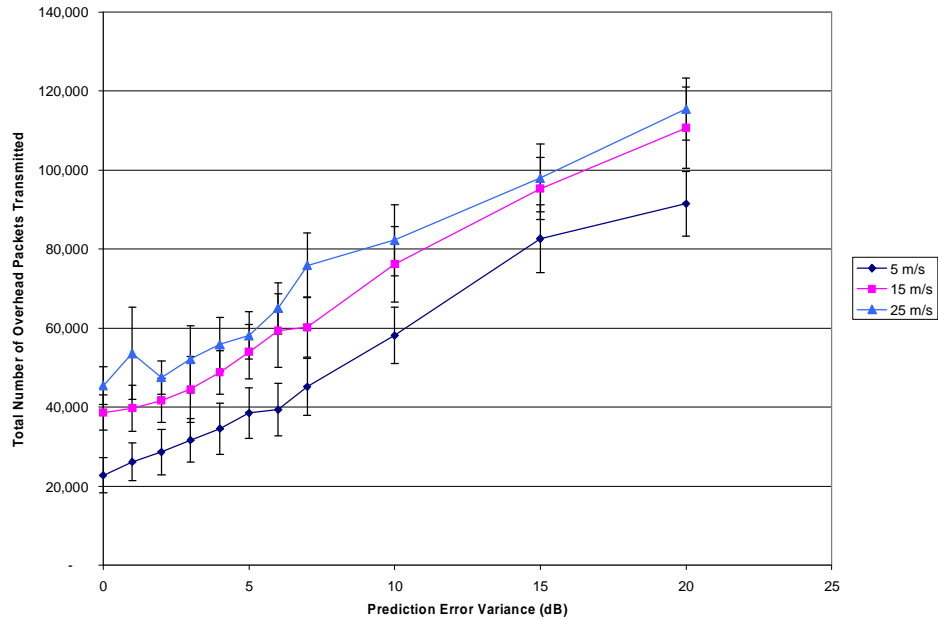


Figure 17: Number of Overhead Packets in Flat Terrain

Figure 18 shows the packet delivery efficiency of TBR as a function of the prediction error. We define the packet delivery efficiency as the ratio of the number of packets delivered to the total number of packets transmitted. The figure shows a sharp drop from 0 dB error to 1 dB error. After that, all three mobility cases exhibit similar trends with the medium and high mobility cases slowly converging. This convergence shows that the path loss prediction errors are beginning to dominate the mobility prediction errors. As the prediction error variance grows, more links are more likely to be incorrectly predicted resulting causing the route maintenance and discovery functions to be invoked more often. As the error grows above 10 dB, even the low mobility case begins to converge toward the medium and high mobility packet delivery efficiency.

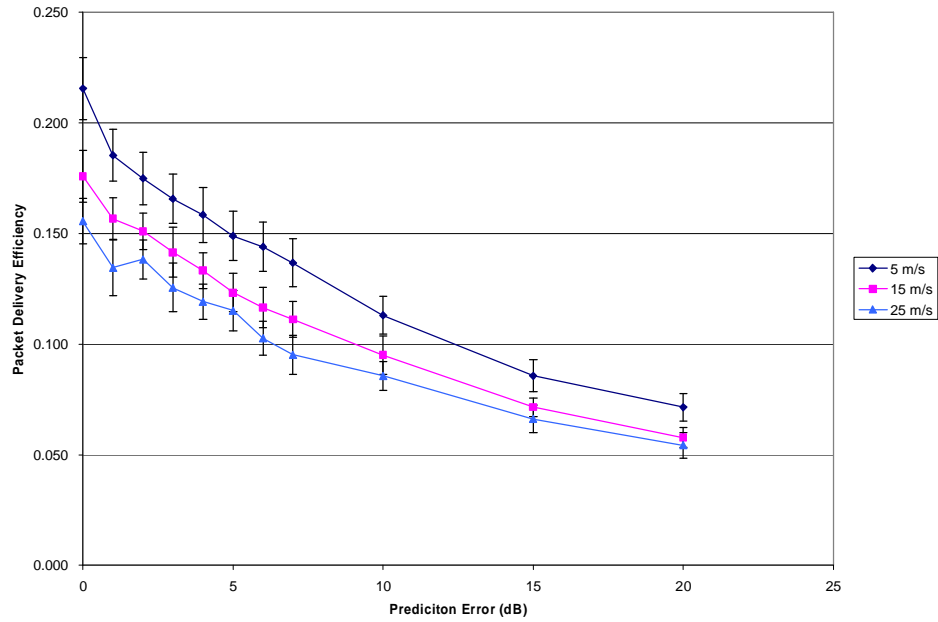


Figure 18: Packet Efficiency in Flat Terrain

7.1.2 Effects of Prediction Error in Hilly Terrain

After completing the simulations using flat terrain, we repeated them using hilly terrain to determine if errors in path loss prediction degrade TBR's performance differently. Figure 19 & Figure 20 show the throughput and packet delivery ratios in hilly terrain. In both figures, the performance degrades as the prediction error increases. All three mobility cases show a sharp drop in performance from perfect prediction to 1 dB error which then becomes more gradual as the prediction error variance is increased to 20dB. Although both the medium and high mobility rates exhibit a spike (negative for high mobility and positive for medium mobility) at 5 dB, a line drawn from the 4 dB to the 6dB point still falls within the confidence interval. From 0dB to 20 dB, the throughput decreases approximately 40% for both the low and medium mobility cases and approximately 44% for the high mobility cases.

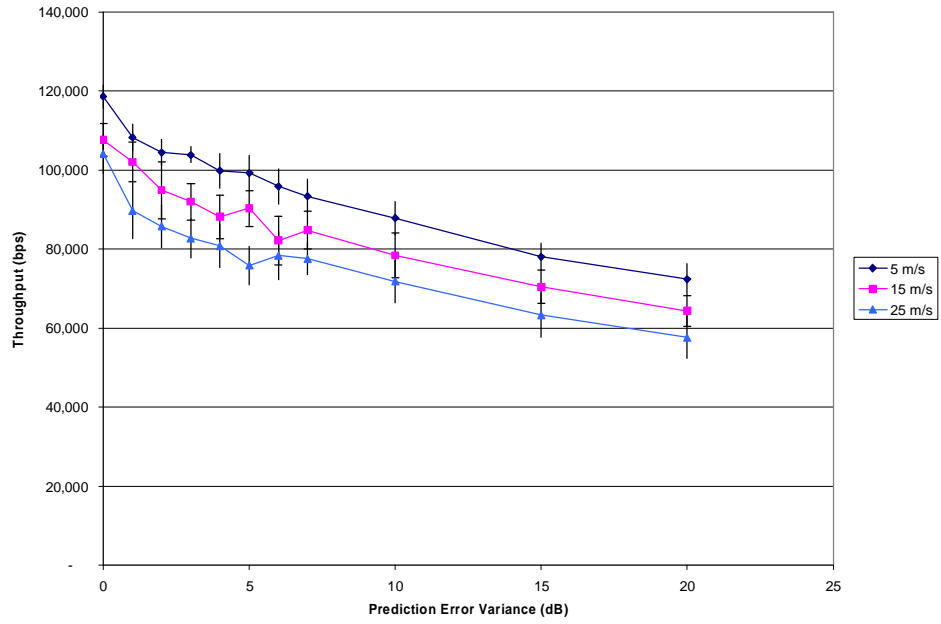


Figure 19: Throughput in Hilly Terrain

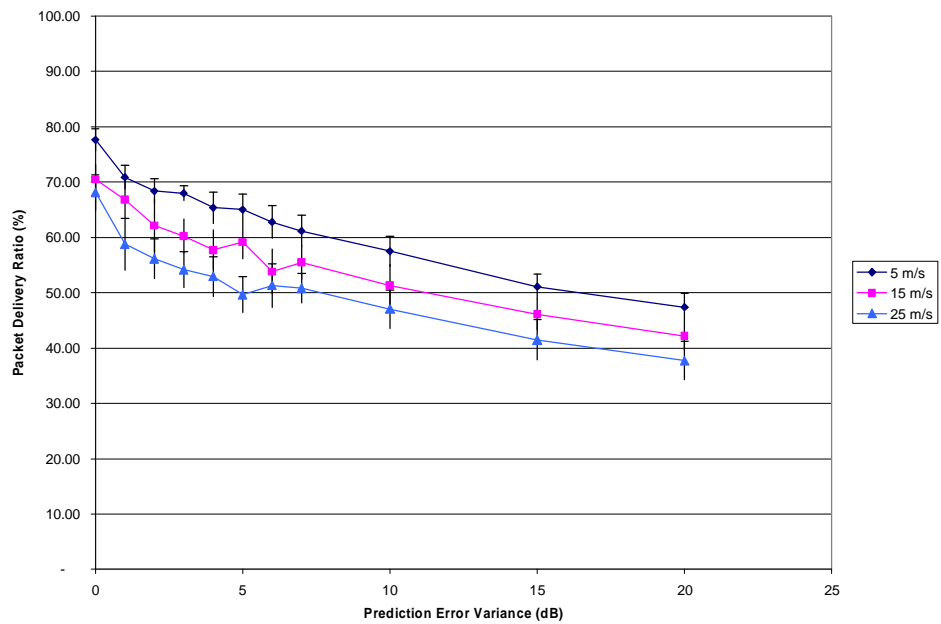


Figure 20: Packet Delivery Ratio in Hilly Terrain

The average end to end delay in the hilly terrain is significantly different than in flat terrain. In the low mobility case, delay spikes slightly once prediction errors occur, however, there is a slight downward trend until about 6dB when the delay begins to increase. The net change from 0dB to 20 dB is a 52% increase in the average delay. Below 6dB, the reduction in the percentage of delivered packets results from the inability to maintain long links due to the prediction errors. The prediction errors cause a slight increase in the average path length due to the increasing inaccuracy in the routing tables.

In the medium mobility case, the average delay drops slightly and remains flat until the prediction error reaches 15 dB. Overall, the medium mobility case exhibits slight increase of 5.4% as the error grows from 0 dB to 20dB. In the high mobility case, the average delay decreases almost 7% as the error increases. Here the delay is dominated by the time required to discover new routes rather than incorrectly predicting links.

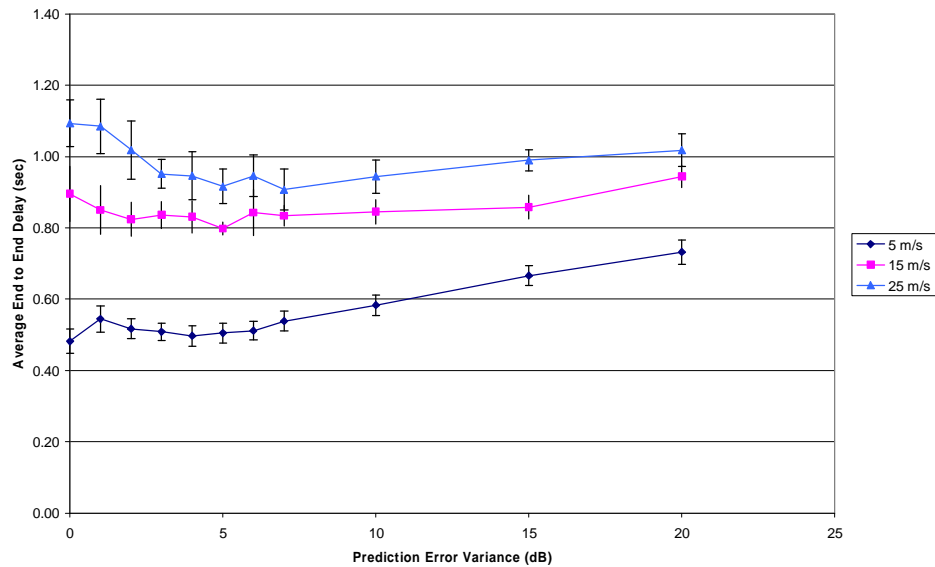


Figure 21: Average End to End Delay in Hilly Terrain

As shown in Figure 22, the mobility rate has little impact on the average path length. For all three rates, the path length remains relatively constant until the error reaches 10 dB. After this, all three mobility rates show a definite drop in the average path length resulting from the decrease in percentage of packets delivered. As the error gets larger, the average path length increases because TBR may not find the shortest path. However, as the error goes up, the ability to deliver packets on longer routes goes down. These two tendencies cancel one another out until the error exceeds 7dB. Once this occurs, even the shorter routes begin to be affected and the route maintenance and discovery functions are invoked more often, resulting in a 58% average increase in overhead.

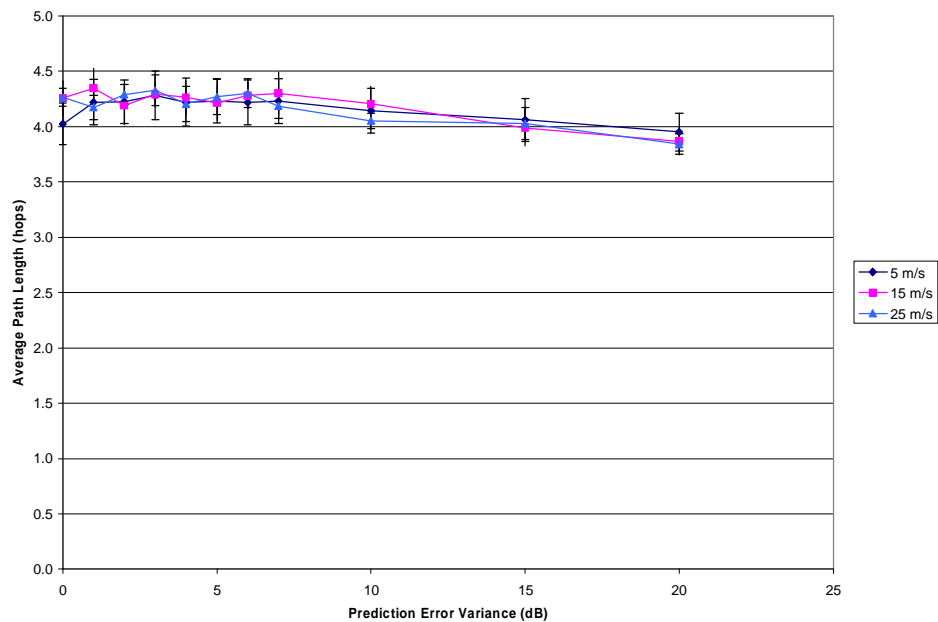


Figure 22: Average Path Length in Hilly Terrain

The packet delivery efficiency in hilly terrain follows the same trend seen in the flat terrain simulations. Figure 23 is very similar to Figure 18 except that the y axis has been shifted down (i.e. initial values are lower). As expected, the terrain induced link

breakages cause a significant growth in overhead. However, the growth in overhead is somewhat constrained by the reduced connectivity cause by the terrain.

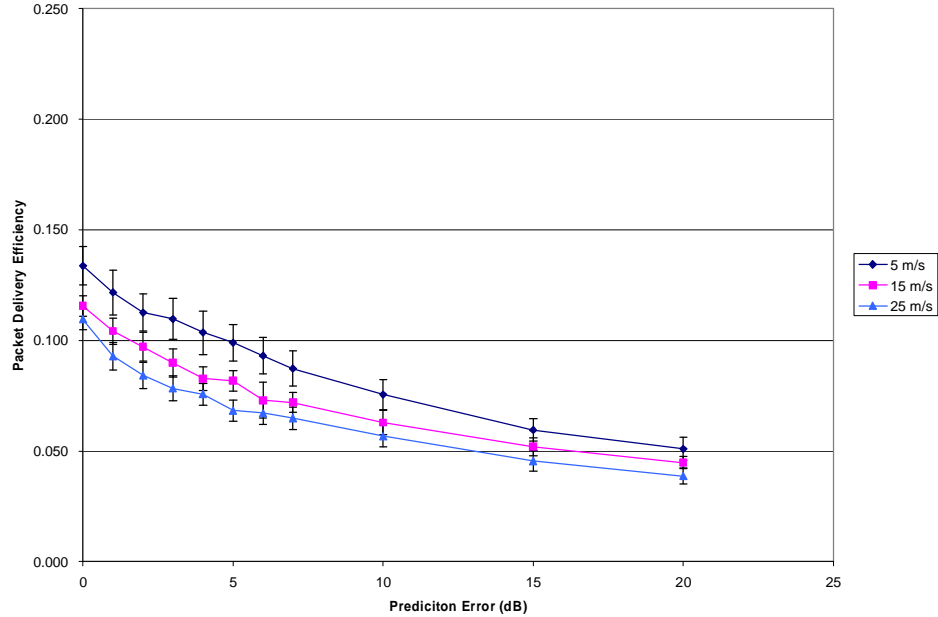


Figure 23: Packet Transmission Efficiency in Hilly Terrain

7.2 Experiment 2: Comparison of TBR to DSR, DREAM, and LAR

In the second set of experiments, we compared the performance of TBR to other MANET routing algorithms to determine how much improvement can be realized using TBR. As discussed in Section 6.4, we compared the performance of TBR with and without prediction errors to DREAM, DSR, and LAR. From the results of Experiment 1, we determined that the maximum acceptable prediction error was 6dB. When the error variance is greater than 6dB, the performance of TBR was unacceptable. Table 4 lists the different scenario parameters that were varied during the performance evaluation.

Like Experiment 1, we performed the experiment in flat and hilly terrain to determine the impact the terrain had on the performance of the protocols. In addition, we

varied the size of the network and the amount of traffic. For clarity of presentation, we do not show all of the charts generated from the simulations. Instead, we highlight the graphs displaying interesting results and provide the full set of full sized graphs in Appendix A. In the following sections, we first compare TBR to DSR, DREAM and LAR in flat terrain. Next we examine the results of the hilly terrain simulations. We then conclude with a section summarizing and comparing the flat and hilly results.

7.2.1 Flat Terrain Comparison Results

7.2.1.1. Overview

In general, TBR-0's performance exceeded the performance of all of the other protocols. The average end to end delay is the only metric that TBR-0 was not the clear leader on. This is a side affect of how TBR operates. TBR relies on gathering location information from packets moving across the network. If the amount of traffic is high, the protocol has very up to date information and can quickly route packets along optimal routes. If TBR does not know the location of a large number of nodes, it spends more time discovering routes via the route discovery process. This increases the delay as packets are queued while the discovery process is conducted. In addition, the delay is increased due to the additional overhead traffic generated by the discovery process. As mentioned in CHAPTER 5, TBR sees an increase in end to end delay because the discovery process takes longer. This is because only the destination responds to the route discovery message. If the paths are stable and the discovery process is not called often, then the initial increase in delay is averaged out over a large number of packets, minimizing the net affect of the initial discovery delay.

One common theme across the metrics was the extremely poor performance of all of the protocols in small networks. This is intuitive if one considers the fact that the scenario area is fixed for all network sizes. Therefore the network density varies

significantly from the 20 node case to the 100 node case. The smaller networks are frequently partitioned and nodes are often unable to establish a route to a destination. Because of this, the delivery ratio is low and the delay is extremely high. At small network sizes, the network is sparsely connected and traditional ad hoc networking protocols are not as effective in routing packets. As the nodes move around, they are able to eventually deliver packets to the destination. At low mobility rates, once a route is established, it is likely to last a while; conversely, a network partition will last longer as well. At higher mobility rates, the nodes have a higher average number of neighbors but the neighbors come and go more frequently causing additional overhead as links fail. In addition, the impact of the mobility is much greater in smaller networks where there are fewer links. In the 20 node high mobility scenarios, each node had an average of 2.3 neighbors compared to 3.4 neighbors in the low mobility case.

During low mobility, the 30 node network exhibited extremely poor performance for all of the protocols tested. A detailed examination of the data indicates that this was due to a smaller average number of neighbors which resulted in more partitions. Because of this, the 30 node results varied significantly between simulation seeds as mobility had a greater impact than in other scenarios. Rather than the metrics changing for the 30 node medium and high mobility scenarios, the values remained relatively constant while the rest of the network sizes decreased in performance. From a visual perspective, this made it look like the results improved. A similar issue cropped up in the 70 node scenarios at low mobility where the average number of neighbors was smaller than the 60 node network. As in the 30 node scenario, this anomaly disappeared when the mobility increased.

7.2.1.2. Packet Rate: 1 packet per second

The throughput graphs show an interesting trend, at low mobility, the all of the nodes except for TBR-6 perform similarly. As the speed increases, the performance of

LAR and DSR decreases as the mobility increases. The throughput for DREAM and the two TBR versions remain relatively constant. DREAM's performance remains constant primarily because it relies on directed flooding. As long as any set of links connects the source to the destination and the network can support the traffic, DREAM's performance remains relatively constant. TBR-0's performance is also only slightly affected. DSR and LAR both are about equally affected by mobility. Since LAR is based on DSR, this is not surprising. Since LAR and DSR rely on caching of routes, as mobility increases, the routes break more often and the cache becomes less effective, resulting in decreased throughput.

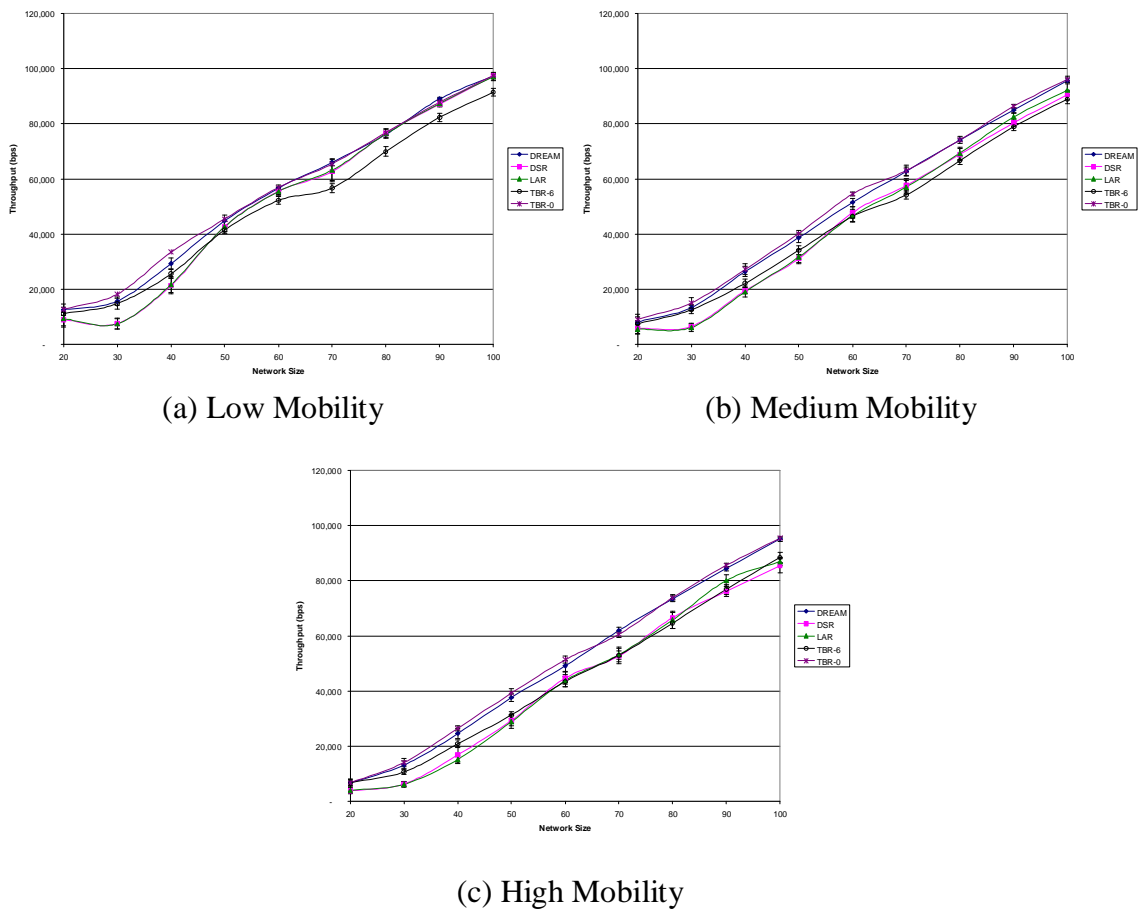
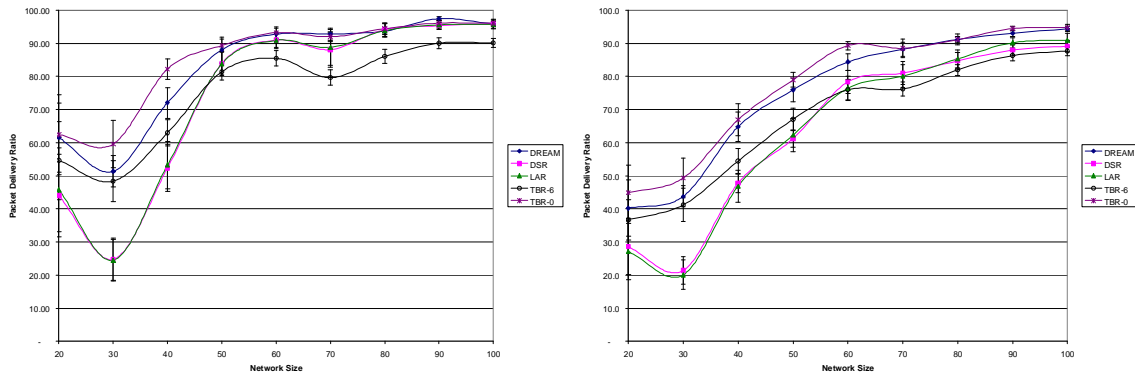


Figure 24: Throughput in Flat Terrain at 1 Packet/Second

In smaller networks, lower speeds create more stable links resulting in an increased percentage of packets being delivered. As the mobility increases, the number of network partitions increases and the delivery ratio drops. This is not as pronounced in large networks where the likelihood of a network partition is much smaller. Figure 25a shows a dip in performance for TBR-6 and DSR and LAR for the 70 node network. The drop in the percentage of packets delivered is related to the mobility random pattern. As the mobility increases, the effect disappears.

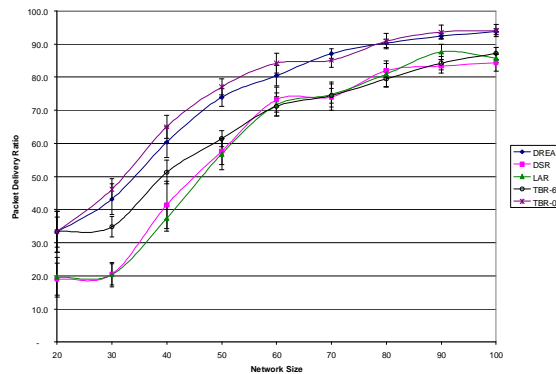
At low mobility, all of the protocols, other than TBR-6 achieve a greater than 95% delivery ratio. As was demonstrated in experiment 1, TBR-6 is expected to perform about 10% worse than TBR-0. As the network size increased, this performance hit dropped to approximately 6%. In the case of the packet delivery, TBR-6 achieves a 90% delivery ratio at low mobility. As mobility increases, TBR-6 maintains its performance better than DSR & LAR and delivers more packets than them in the high mobility 100 node scenario. This is due to the mobility prediction algorithm which is able to maintain routes better as the mobility increases.

When TBR suffers a link failure, it recomputes the route to the destination using its updated location predictions. This enables it to proactively remove links that should no longer exist. In contrast, DSR and LAR only remove links that are explicitly listed in their error messages. While this enables them to remove some stale cache entries, links that the protocol has not been explicitly notified of are not removed. This difference enables both TBR versions to maintain their performance as mobility increases.



(a) Low Mobility

(b) Medium Mobility



(c) High Mobility

Figure 25: Packet Delivery Ratio in Flat Terrain at 1 Packet/Second

The end to end delay graphs are shown in Figure 26. In networks smaller than 50 nodes, DREAM offers lower end to end delay because it does not attempt to establish routes. As mentioned previously, it floods the packet in the direction of the destination. This works well in small networks but causes congestion in larger networks and does not scale. The flooding also minimizes the effect of mobility. DREAMS scoped flooding also means that if a path to the destination exists, DREAM will find it and immediately use it. The remaining protocols all attempt to discover a route to the destination. In the smaller networks, DSR, LAR and TBR do not have good enough connectivity or enough

traffic to accurately maintain routes. Instead, they are forced to frequently use their route discovery to find a route. Since the network is sparsely connected, TBR is not able to gather much information about all the nodes in the network. However, as the network size and number of nodes transmitting grows, TBR is able to better predict connectivity and the discovery process is used less often. Likewise, DSR and LAR are able to gather and cache more routes as the network size increases. As the number of nodes in the network increases, the density increases resulting in the average distance between nodes decreasing, increasing the average link quality. This in turn helps TBR-6 as the increased link quality reduces the impact of the prediction error, resulting in increased performance and reduced delay.

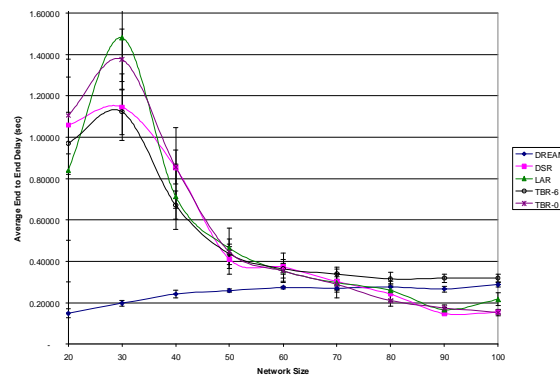
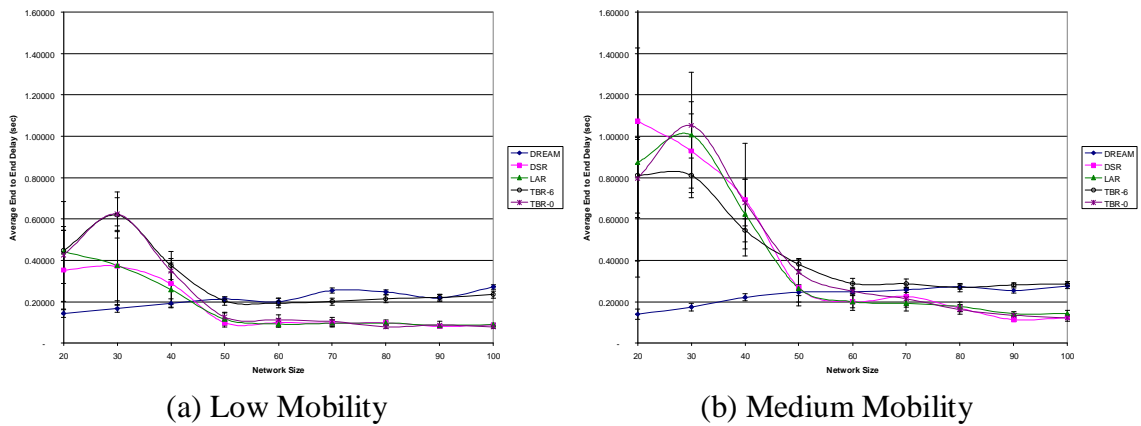
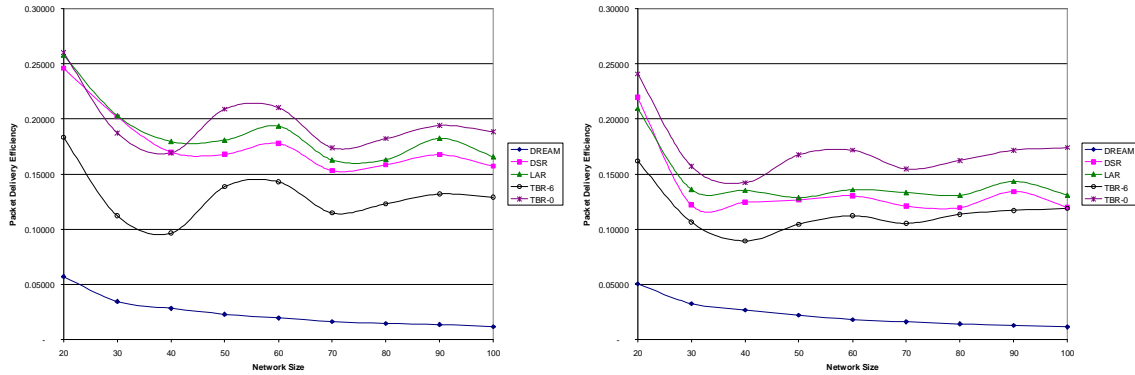


Figure 26: Average End to End Delay in Flat Terrain at 1 Packet/Second

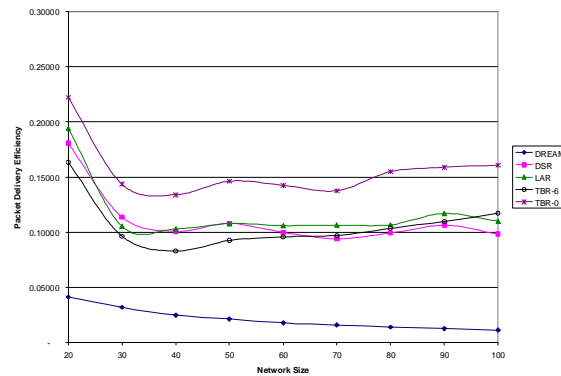
The packet delivery efficiency of DSR is significantly lower than the other protocols. LAR's limiting of control packets to the request zone helps improve its efficiency relative to DSR. TBR-0's efficiency remains higher than LAR, DSR and TBR-6. As the mobility rate increases, the efficiency of both DSR & LAR decreases more rapidly than either of the two versions of TBR. The damped oscillatory nature of the efficiency curves for DSR, LAR, TBR-0 and TBR-6 in Figure 27 results from a large increase in number of delivered packets versus the overhead created by adding new nodes. The greater the number of nodes, the more overhead that broadcast packets (i.e. route discovery requests) cause. At low mobility, the increased number of nodes allows more paths to exist and reduces partitions. At some point, the connectivity is too good and the overhead increases as more nodes receive and relay the broadcast packets. As the network increases and the density increases, the overhead grows faster than the improvement in delivery, flattening the curve. Since mobility creates additional overhead, as the mobility increases, the oscillatory behavior is further damped.

The 20 node scenario has a much higher efficiency due its small size and the number of partitions that result from this. There are simply too few nodes to relay a broadcast message, greatly reducing the number of overhead packets being transmitted. However, as the network size grows, more nodes are likely to be able to relay the broadcast.



(a) Low Mobility

(b) Medium Mobility



(c) High Mobility

Figure 27: Packet Delivery Efficiency in Flat Terrain, 1 Packet/Second

Overall, the performance of all of the protocols except for DREAM improves significantly as network size increases. DREAM’s packet delivery ratio increases but the end to end delay also increases. DREAM’s packet delivery efficiency also drops continuously while the efficiency of the remaining protocols remains relatively flat.

7.2.1.3. Packet Rate: 2 packets per second

In general, the results for the scenarios with nodes transmitting at 2 packets/second shows the same trends as in the 1 packet/second scenarios. At 2 packets/second, the additional overhead of DREAM begins to affect its performance as network size grows. As the network grows, DREAM is unable to deliver as many

packets due to the increased amount of contention and congestion which increasingly affects DREAM's unreliable hop by hop delivery mechanism. This congestion also shows up in the increased average path length for DREAM. In the larger networks, the average increase is about 10%. As the congestion grows, DREAM's flooding mechanism forces packets to travel around the congestion, increasing the average path length even as the percentage of data packets delivered is decreasing.

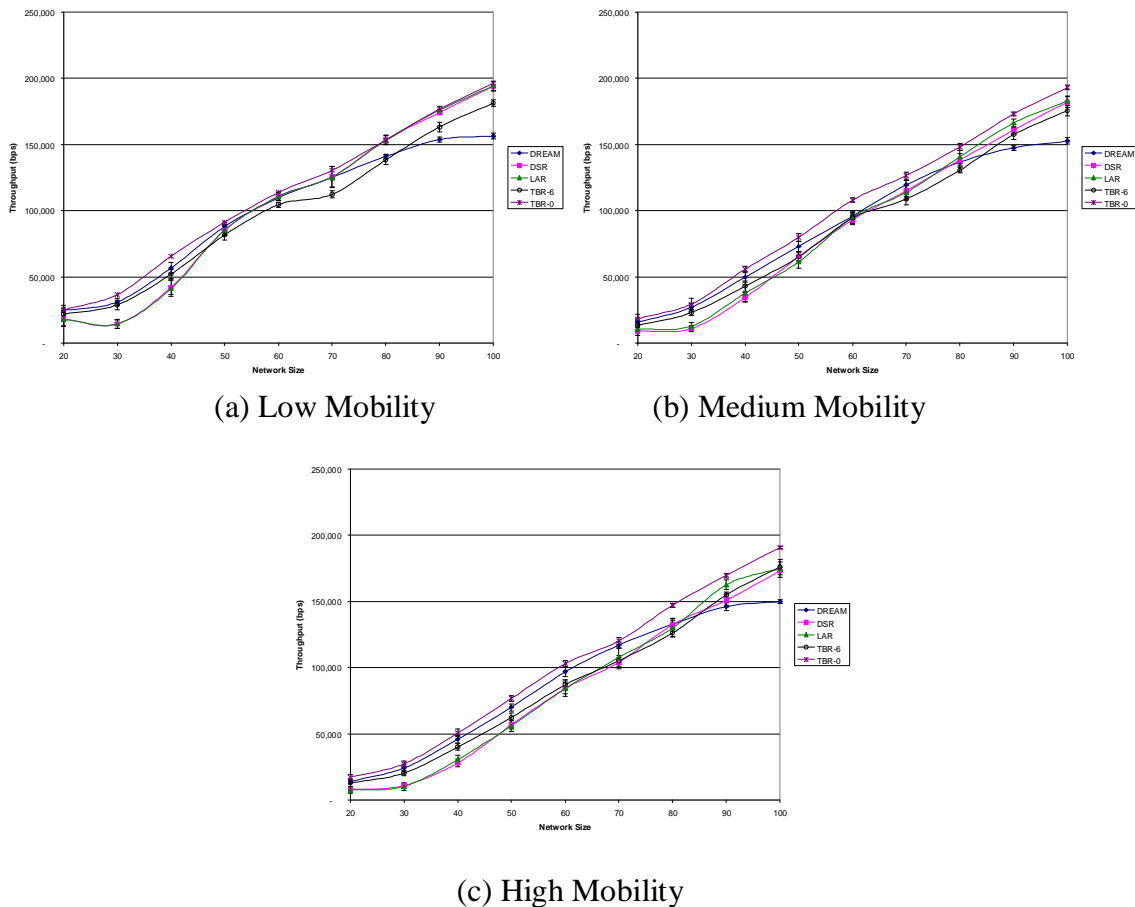


Figure 28: Throughput in Flat Terrain at 2 Packets/Second

The throughput of the 2 packet/second scenarios shown Figure 28 exhibits a similar trend to the 1 packet per second scenarios but with about twice the throughput. Above 80 nodes, DREAM becomes the worst performer of the five protocols. As would

be expected, the increased traffic flow causes an increase in the average end to end delay for all of the protocols. The increase is much more pronounced in DREAM than in the other protocols. The increased traffic flow also increases the delivery efficiency of all of the protocols except for DREAM. Graphs for all of these metrics can be found in Appendix A.

7.2.1.4. Packet Rate: 3 packets per second

As the traffic generation rate is increased to three packets per second for each of the transmitting nodes, we continue to see DREAM's performance degrade. DREAM's scoped flooding is simply too inefficient to support that packet rate when the network is larger than 60 nodes. Initially, DREAM is able to receive the end to end acknowledgements. However, as the network size grows and the network density increases, the acknowledgements increasingly fail to be received at the source resulting in the data packet being flooded throughout the network, further increasing the congestion.

Like DREAM, TBR-6 begins to have problems supporting the offered load when the network reaches 100 nodes. The number of overhead packets transmitted grows by nearly 170% and the average delay increases almost 45% compared to the delay for the 90 node network. As the traffic increases, the impact of the prediction error becomes more pronounced. This is due, in part, to the technique employed to add error to TBR's propagation model. Each time TBR-6 computes a path loss, it adds a random amount of error (normal distribution with zero mean and 6dB variance). A side effect of this approach is that the propagation results are not always the same given the same inputs. While this provides variance to the simulation engine's propagation model outputs, it makes it more likely that any two nodes may have a different view of network connectivity, given the same data. This makes it much more likely that temporary routing loops are created and that the route maintenance function is invoked. Due to the way the route maintenance was implemented, if TBR detects that 2 nodes have

conflicting views of the network (i.e. how to route the maintenance packet back to the source when a packet is undeliverable) and no path to the destination exists, it drops the packet. Since the inconsistent view only lasts until the next route recomputation, it usually is not a problem. However, if the inconsistent view occurs along a high traffic link, many packets can be lost before the problem is resolved. Looking at all of the graphs, we observe that the confidence intervals for all of TBR-6's metrics except for average path length are significantly larger for the 100 node scenario.

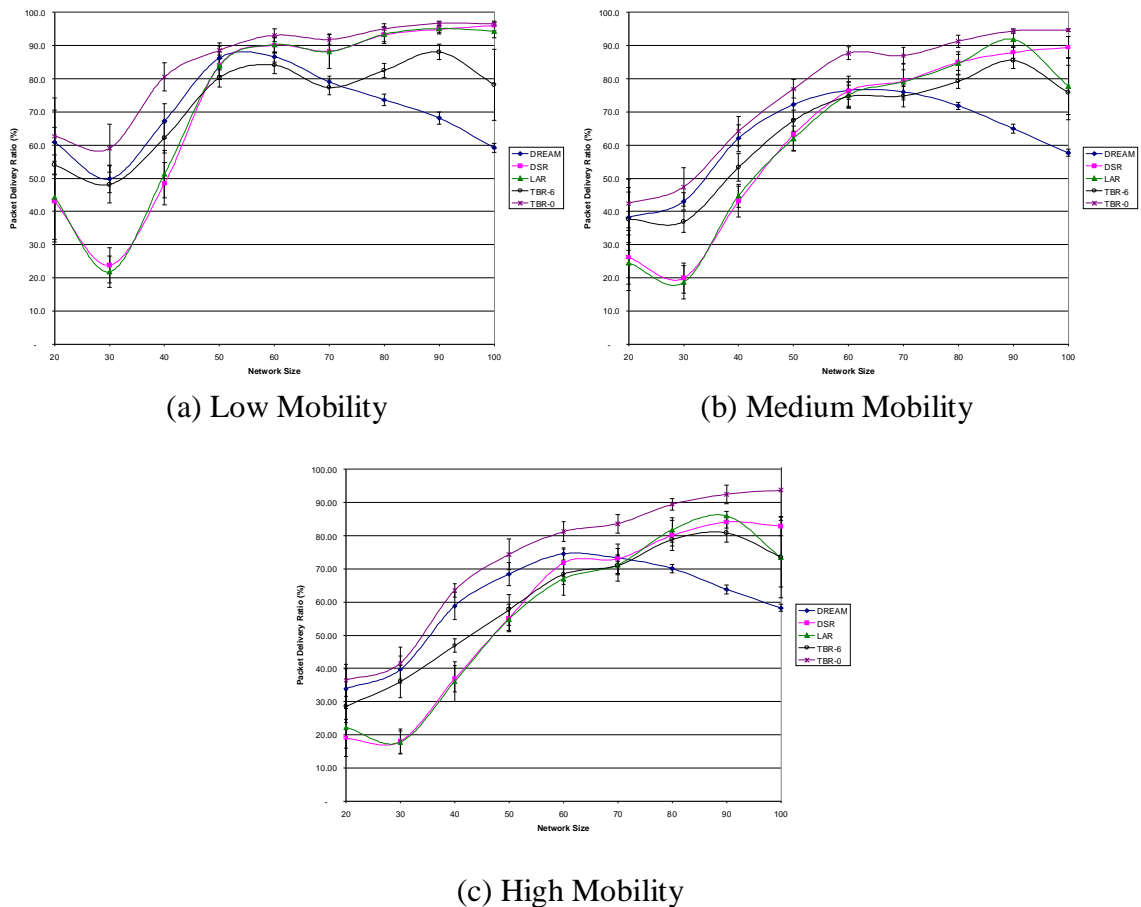


Figure 29: Packet Delivery Ratio in Flat Terrain at 3 Packets/Second

In the larger networks, at low mobility rates, TBR-0, DSR and LAR perform very similarly. At medium and high mobility rates, LAR begins to exhibit problems when the network reaches 100 nodes. Like TBR-6, LAR has a dramatic increase (253% for

medium mobility & 160% for high mobility) in the number of overhead packets transmitted. Additionally, the average delay increases by about 500%. The percentage of packet delivered decreased 14%. DSR, which is virtually identical to LAR, does not have any of these problems. Up until 100 nodes, LAR had similar performance to DSR with significantly lower overhead. Again the confidence interval is extremely large.

The problem lies in two of the runs where a large number of packets were lost due to queue overflows in the datalink layer. The localized congestion caused a cascading failure in LAR. The problem lies more in an inefficient MAC layer than in a problem with LAR. However, LAR contributes to the problem since it provides gratuitous route replies which compete with the data traffic. As the network density increases, more and more nodes are competing to get access to the channel. Eventually, the MAC can no longer support the traffic and the network collapses. The average number of neighbors in the 100 node network is 9. However, several nodes have a large number of neighbors (~17) which causes severe channel contention issues if traffic is transiting through these nodes. DSR did not suffer the same problem since does not have an aggressive policy of scoping flooding of discovery packets. This extra overhead may have helped provide DSR with the additional routes needed to prevent the localized congestion or allowed nodes outside of the request zone to reply to the route request.

At high mobility rates, DSR and TBR-0 both experience a decreased amount of throughput. The high mobility had a greater impact on DSR since its cache was much more likely to contain stale routes. TBR avoids this through its use of mobility prediction. If TBR is unable to forward a packet on to the next hop due to a link failure, it recomputes the routing table using the predicted location of the nodes in the network.

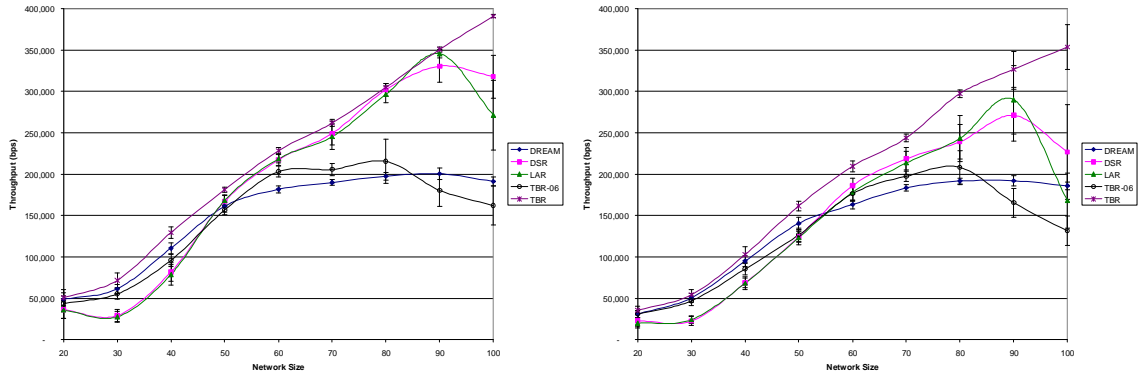
The packet delivery efficiency of the protocols at three packets per second is very similar to the two packets per second scenarios. The higher offered load increases the efficiency slightly. Like the packet delivery efficiencies at the lower offered loads, all of the protocols exhibit the damped oscillatory behavior that was previously discussed. The

only change is in the medium and high mobility 100 node networks where both TBR-6 and LAR suffer a significant drop in efficiency due to the large number of overhead packets that are were transmitted.

7.2.1.5. Packet Rate: 4 packets per second

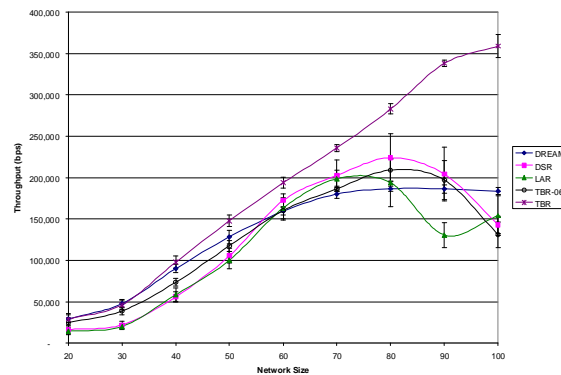
Once the traffic generation rate for transmitting nodes is increased to four packets per second, the datalink queue overflow problems that affected LAR earlier begin to affect DSR also. A low mobility rates, both LAR and DSR perform very similar to TBR-0. Figure 30 shows TBR continues to maintain the throughput as mobility increases. TBR-6 and DREAM both suffer from congestion issues once the network gets above 60 nodes. The fact that the TBR protocol does not allow gratuitous replies prevents the datalink queues from overflowing for TBR-6. However, both TBR-6 and DREAM suffer from the transmission physical layer collisions due to the hidden terminal and exposed terminal problems [54][55].

As the mobility increases, the network experiences more link failures. For LAR and DSR, the increasing mobility reduces the effectiveness of their route caching. This in turn leads to more overhead as the route discovery process is invoked. Nodes responding to these route requests using their stale cache information consume more bandwidth. In addition to the extra congestion that this traffic causes, data traffic faces increased delays as it is forced to wait for new routes to be discovered.



(a) Low Mobility

(b) Medium Mobility



(c) High Mobility

Figure 30: Throughput in Flat Terrain at 4 Packets/Second

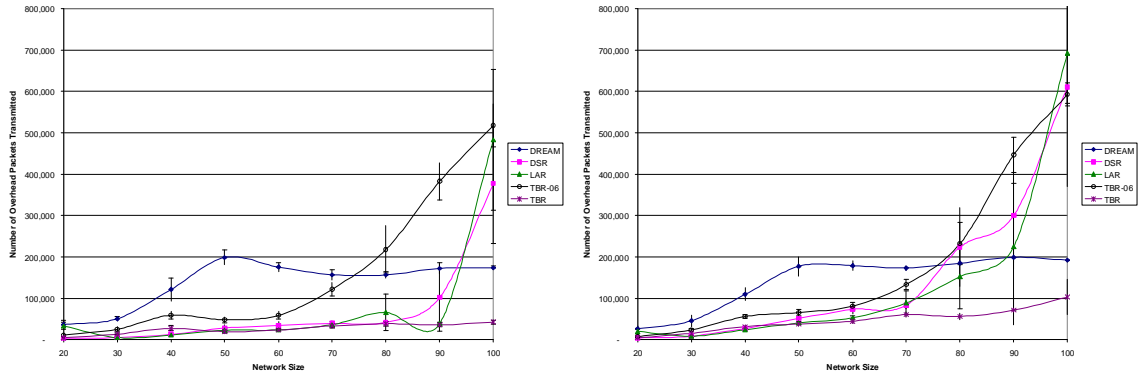
The datalink layer is unable to handle the volume of traffic being generated by the overhead, particularly if the traffic is unicast (for route replies and data packets) and thus requires reliable delivery by the datalink layer. One of the contributing factors to the congestion at the datalink/MAC layer is the power the nodes are transmitting at. The transmit power was the same for all network sizes. In the small networks, each node has an average of 2-3 neighbors depending on the mobility. Once the network reaches 100 nodes, that average has increased 9.2 neighbors. More importantly, once the network reaches 90-100 nodes, some nodes in the network can, at times, have up to 23 neighbors creating an exposed terminal problem at that node. Having that many neighbors can

significantly reduce the amount of bandwidth the node is able get and in turn reduce the traffic rate through the node that can be supported.

Unlike LAR & DSR, TBR does not have many packets dropped because of queue overflows. As mentioned earlier, in TBR, only the destination responds to a route discovery request. While this causes an increase in the time required to establish a route, it greatly reduces the overhead generated. While the overhead of the discovery messages is higher than it would be if intermediate nodes could respond to request rather than forwarding it on, the overhead is distributed across the network rather than localized at the node seeking the destination. Also, since the discovery is a broadcast message, if it is lost due to a collision, no extra overhead is created, unlike a route reply which requires reliable delivery. If the route reply is not received the datalink will resend it, adding more traffic. The discovery packets also help distribute location information that nodes across the network may be able to use.

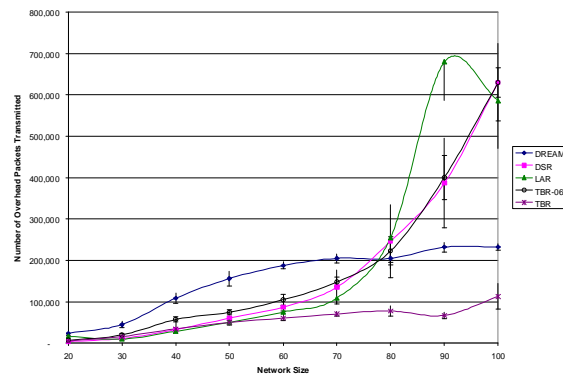
As a test, we changed the TBR protocol to enable intermediate nodes to respond to route discovery requests. We found that TBR suffered problems with queues overflowing once the intermediate nodes started replying.

In addition to reducing the packet delivery ratio and throughput for LAR and DSR, the congestion reduces the average path length of the packets that are delivered and increases the delay the packets experience. This is due to both channel access delay as well as the additional time spent waiting for a route to be discovered. As congestion increases, only shorter paths can reliably support traffic since the longer the path, the more likely the packet is to be lost due to a queue overflow. The datalink problems cause the performance metrics to vary greatly across runs causing the large confidence interval. This is due to the fact that a slight change the sequence of random numbers can have a significant impact if the network is operating at the edge of stability. As shown in Figure 31, a slight change can cause the network to go unstable, as was the case for LAR and DSR in the 90 & 100 node networks.



(a) Low Mobility

(b) Medium Mobility



(c) High Mobility

Figure 31: Number of Overhead Packets in Flat Terrain at 4 Packets/Second

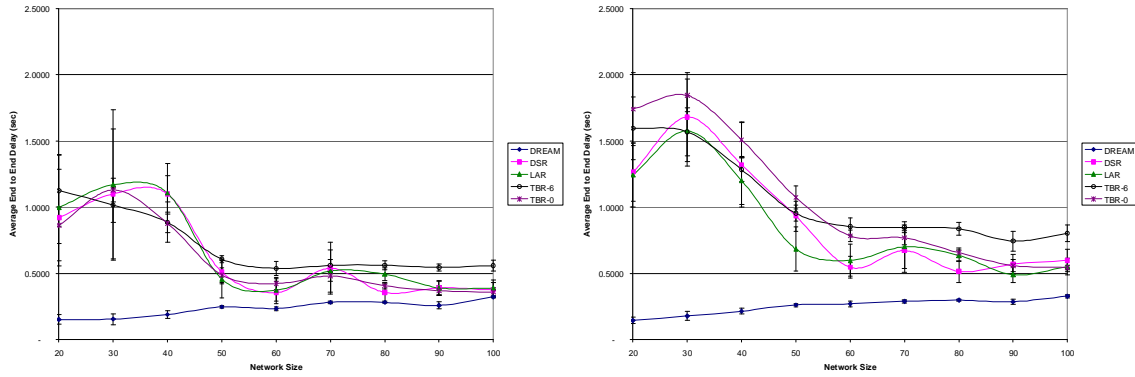
7.2.2 Hilly Terrain Comparison Results

In the second set of simulations in experiment 2, we examined how TBR’s performance compared to DREAM, DSR & LAR when the network operated in hilly terrain. As expected, the hilly terrain had a significant impact on network performance across all of the protocols. In the following four sections we summarize the results as the traffic rate is increased from one packet per second to four packets per second.

7.2.2.1. Packet Rate: 1 packet per second

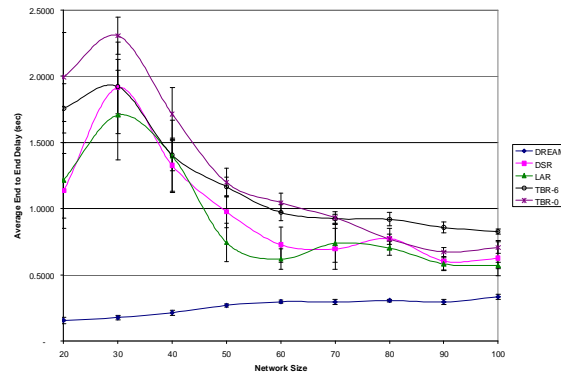
At low traffic rates, DREAM performs well. Since it doesn't have the delay associated with discovering routes, its average delay is significantly lower than the other protocols. This is particularly acute in smaller networks where partitions are likely and connectivity is intermittent. In small networks, the route discovery processes used by DSR, LAR and TBR are much more likely to fail and time out. For DSR & LAR, the timeout period is random (0.5 seconds to 10 seconds) but averages out to about 5 seconds. TBR has a fixed timeout interval of 5 seconds. In small networks with limited connectivity, the lifetime of the paths discovered by DSR, LAR and both TBRs do not last long enough to amortize the initial route discovery delay. TBR in particular is hindered by its reliance on the destination to reply to discovery requests. DSR and LAR are able to leverage the routes cached routes to provide a route sooner than TBR. However, TBR's mobility prediction provides an edge in maintaining the routes once they are established. As network size increases and connectivity improves, the routes last longer and the route discovery delay has less of an impact. There is also an increased chance that a node will have multiple routes to a destination as the connectivity improves. Because of the path loss prediction errors, TBR-6 has a higher average delay than TBR. This is due to the increased length of the paths the packets travel that result from choosing less optimal paths.

As the mobility increases, the probability of a link breakage increases. This effect is more prominent in the hilly terrain than in the flat terrain. The link breakages cause an increase in delay as new routes are discovered. Since DREAM uses directional flooding, its average delay is not as affected by the increased mobility as the other protocols. Figure 32 shows the average delay for each of the protocols.



(a) Low Mobility

(b) Medium Mobility

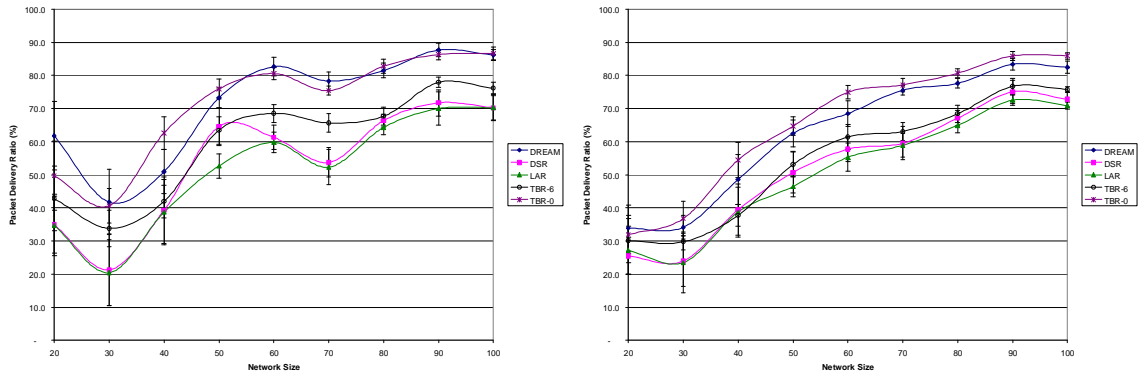


(c) High Mobility

Figure 32: Average End to End Delay in Hilly Terrain at 1 Packet/Second

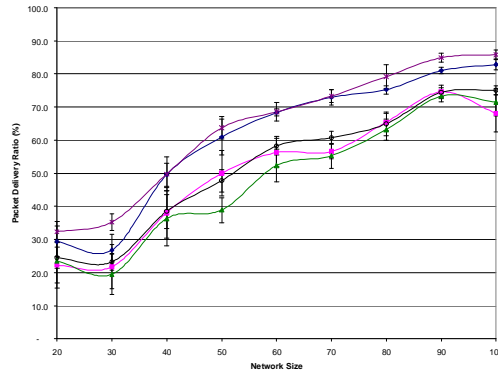
In small networks, the packet delivery ratio, shown in Figure 33, decreases as the mobility increases. This results from a decrease in the average number of neighbors, which reduces connectivity. As the network size increases, mobility provides some improvement to DSR and LAR. In the case of the 70 node network this particularly large. The improvement is due to an increase in the average number of neighbors. This increase provided just enough increase in available routes to improve their performance without increasing overhead. Once the network reaches 100 nodes, the mobility does not provide an increase in the average number of nodes but does increase the maximum number of neighbors that some nodes have. This increase leads to localized congestion

due to datalink inefficiencies and begins to affect performance resulting in a slight decrease in the percentage of packets delivered.



(a) Low Mobility

(b) Medium Mobility



(c) High Mobility

Figure 33: Packet Delivery Ratio in Hilly Terrain at 1 Packet/Second

Unlike DSR and LAR, DREAM's percentage of delivered packets decreases as mobility and network size increases. As the average number of neighbors increases, the probability of a packet being lost in a collision at the lower layer increases. In larger networks, the performance TBR-0 & TBR-6 is relatively stable in the face of mobility. Although there is a slight decrease in the average percentage of packets delivered, the upper and lower bounds of the confidence interval remain almost constant as the mobility changes.

7.2.2.2. Packet Rate: 2 packets per second

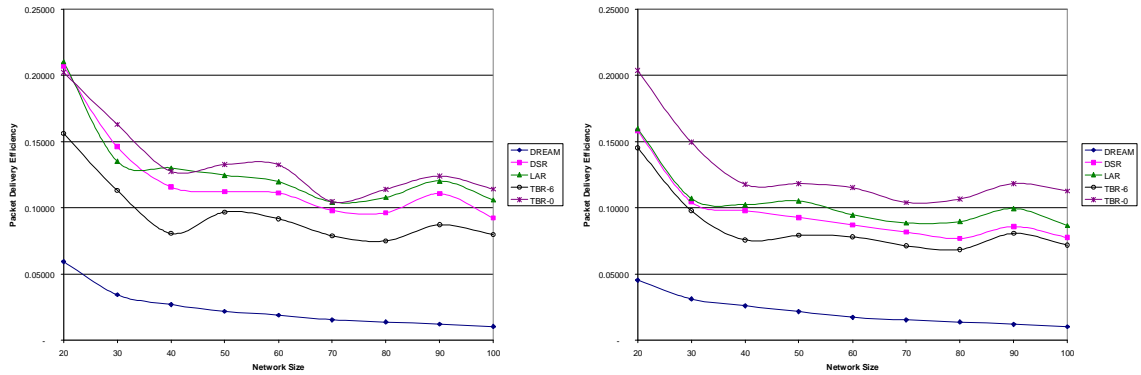
As the traffic rate was increased to two packets per second, the performance of DSR, LAR and the two versions of TBR remain about the same. In the 100 node low mobility scenario, DSR, TBR, and TBR-6 were able to reduce their end to end delay compared to the lower traffic rate because the routes create by the discovery process lasted long enough to amortize the cost of discovery. In the other scenarios, the average delay increased due to increased traffic flow for all nodes except for TBR-6 where the average delay decreased when compared to the delay at the lower traffic rate. This improvement likely resulted from the use of slightly less optimal routes. Since the choice of paths is not always optimal and not always consistent, the traffic flow was more distributed than in the case of TBR-0, DSR and LAR.

As with the lower traffic rate, TBR-0 and TBR-6 both out performed DSR and LAR in the number of packets delivered. With TBR-6 the difference was within the margin of error. Although TBR-6's delay was similar to LAR and DSR, the average path length was 10% longer.

The greatest difference in performance occurred with DREAM. The number of packets delivered by DREAM dropped dramatically as network size increased. At 100 nodes, the percentage of delivered packets dropped from an average of 83% down to an average of 68%. At the same time, the average path length increased by 0.5 hops. This increase in average path length contributed to an increase in average delay.

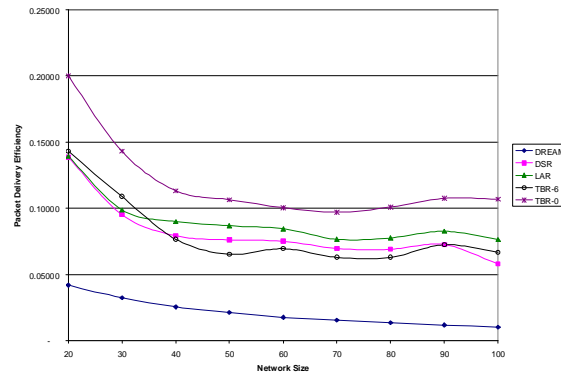
The packet delivery efficiency increased compared to the one packet per second scenarios. Although packet delivery efficiency decreased with increased mobility for all of the protocols, DSR and LAR experienced a greater impact. As Figure 34 illustrates, DSR and LAR start just below TBR-0. LAR has a higher efficiency due to limiting of the discovery packets to inside the expect zone of the destination. This helps control the number of overhead packets LAR transmits. As mobility increases, the efficiency

decreases as links break more often and LAR and DSR are forced rediscover routes. The mobility also reduces the effectiveness of the caches that LAR & DSR maintain.



(a) Low Mobility

(b) Medium Mobility



(c) High Mobility

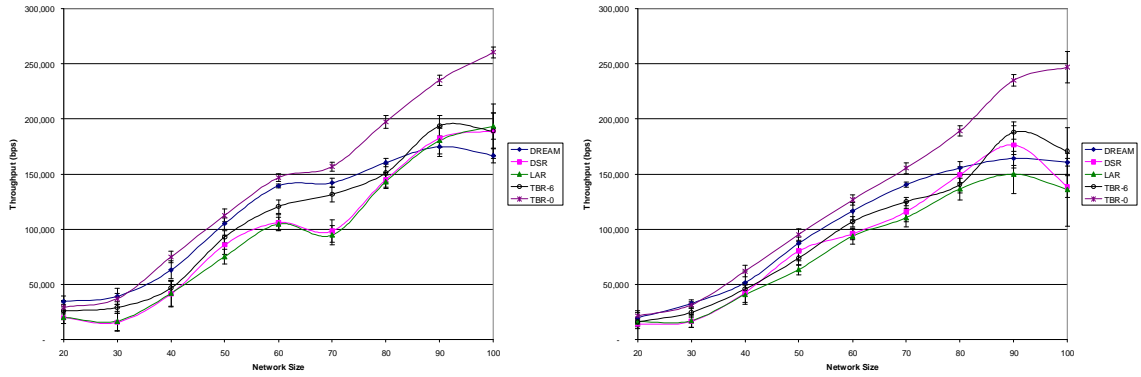
Figure 34: Packet Delivery Efficiency in Hilly Terrain at 2 Packets/Second

7.2.2.3. Packet Rate: 3 packets per second

At three packets per second, the datalink layer queues begin to overflow regularly as the network size grows and begin to significantly impact DSR and LAR. DREAM and TBR-6 also see a dramatic reduction in the number of packets delivered although it is not due to queue overflows. DREAM performance degrades as the volume of traffic increases to the point that the network can not deliver packets along the long routes useable at lower traffic rates. As the traffic increases, the probability of a node “in the

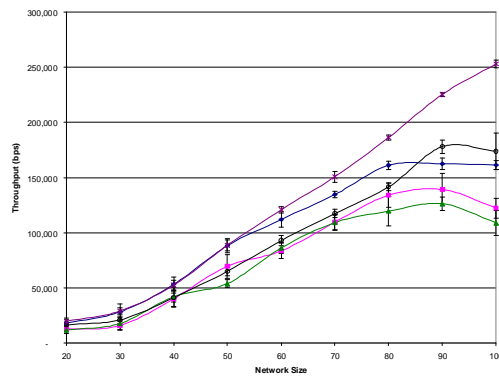
direction of the destination” receiving the packet goes down. TBR-6 sees a reduction in its percentage of delivered packets as more packets are misrouted and lost due to prediction errors (as previously discussed in the flat terrain results). As in the flat terrain case at this traffic rate, nodes face increasing amounts of channel contention at the datalink layer. In large networks, the density of nodes is high enough that the congestion causes the network becomes unstable. This instability is caused mainly by some nodes suddenly having a large number of neighbors rather than the average connectivity being too high. This congestion manifests itself in the highly variable average delay and poor packet delivery ratio. The only protocol to avoid this is TBR-0. The number of overhead packets TBR-0 generates is not enough to make the network go unstable, however, at three packets/second, TBR-0 sees a doubling in the number of forwarding failures causing the route maintenance function to be invoked much more frequently. These failures are most likely due to congestion. And, although TBR-0 is able to recover, the process adds a significant amount of delay. As mobility increases, nodes are forced to recover from more frequent link breakages further contributing to the delay.

Examining Figure 35 we see the impact of the above on the throughput of the network. As in previous traffic rates, in both flat and hilly terrain, the throughput exhibits the same dip in performance at 30 and 70 nodes in low mobility, this is due to a reduction in the average number of neighbors. As the mobility increases, the average number of neighbors in these two networks increases resulting in an improved performance that offsets the decrease in performance that the other network sizes see as the mobility increases.



(a) Low Mobility

(b) Medium Mobility

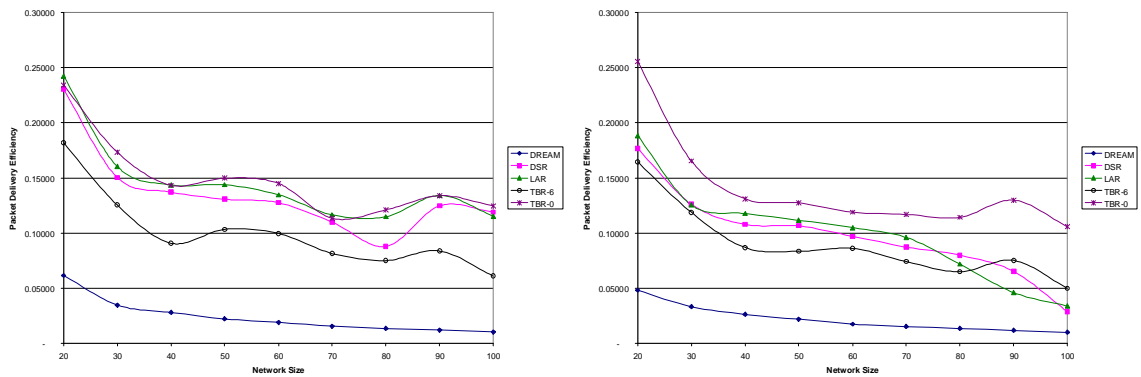


(c) High Mobility

Figure 35: Throughput in Hilly Terrain at 3 Packets/Second

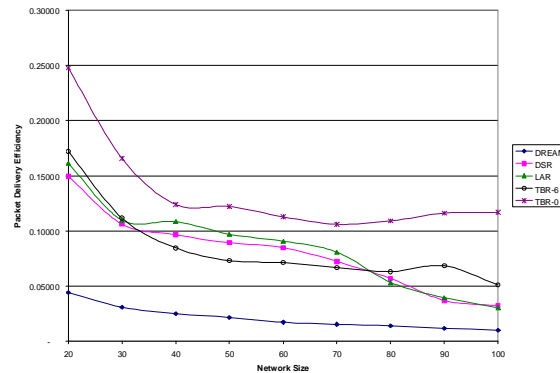
The reduction in throughput and increase in overhead traffic is caused by DSR, LAR & TBR-6 reacting to the datalink layer congestion and reflects in the packet delivery efficiency shown in Figure 36. As in previous graphs of packet delivery efficiency, we see some evidence of the damped oscillatory behavior for all of the protocols except for DREAM, which never displayed that behavior. Due to the increasing congestion, the effect is less pronounced than in some of the previous cases. As before, the increased overhead traffic caused by the mobility causes the efficiency to flatten. As expected, LAR's efficiency is slightly higher than DSR's along most of the curve. LAR's discovery overhead control technique loses its effectiveness as mobility

increases. Rather than doing a single flood across the network, like DSR, LAR limits the flooding to the request zone. If no reply is received, LAR rebroadcasts to the entire network. This becomes less effective in larger networks particularly with high mobility and lots of MAC congestion. The failure is likely due to localized congestion that DSR is able to bypass since the entire network is flooded with the first discovery request rather than just the expected region. Since nodes outside the expect zone don't reply to LAR's discovery packet, a timeout occurs and a second attempt at flooding is made. This second attempt further congests the network. The discovery timeout also causes increased delay.



(a) Low Mobility

(b) Medium Mobility



(c) High Mobility

Figure 36: Packet Delivery Efficiency in Hilly Terrain at 3 Packets/Second

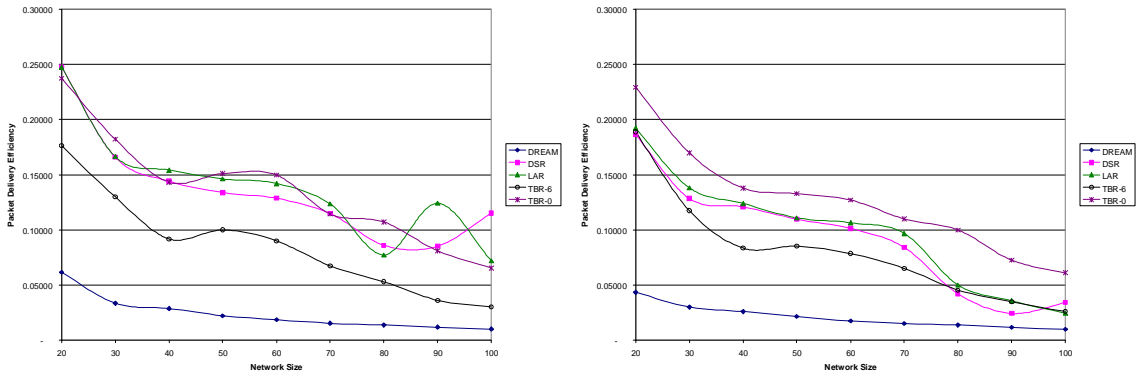
7.2.2.4. Packet Rate: 4 packets per second

Once the traffic rate is increased to four packets per second, all of the protocols show a significant drop in performance. At low mobility rates, the increased traffic demands increase the average delay of the packets that are delivered. One exception is DSR in the low mobility 100 node scenario. In this instance, DSR packets had a lower average delay than DREAM. Looking at the number of overhead packet transmitted, DSR actually transmitted 29% fewer overhead packets than in the 90 node network and 36% fewer overhead packets than TBR-0 did in the 100 node scenario. TBR-0 delivered more packets but doing so resulted in more overhead and a significant increase in the average delay. LAR was not able to duplicate DSR's low average end to end delay in the 100 node scenario. However, it was able to deliver 2.3% more packets but at a cost of 70% more overhead packets.

Like the other protocols, both TBR-0 and TBR-6 saw their performance drop significantly at the higher traffic rate. Both versions of TBR experienced a larger than expected increase in the number of times the route maintenance process was invoked due to forwarding failures. This generated a large amount of overhead traffic. During high mobility the overhead of TBR and TBR-6 exceeded that of DSR and LAR. Despite generating more overhead, both protocols were able to deliver more packets than DSR and LAR. TBR-0 was able to deliver more packets than DREAM while TBR-6 was not. TBR-0 was able to deliver packet over longer paths than DSR, LAR and TBR-6. While the average path length for DREAM was larger than for TBR-0, TBR delivered 16-22% more packets depending on the mobility rate.

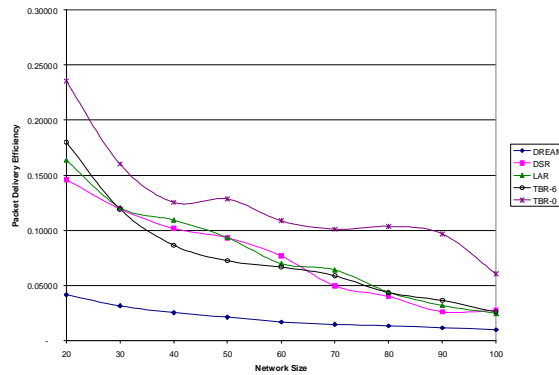
As shown in Figure 37, TBR-0, DREAM and TBR-6 show a general decline in packet delivery efficiency at a traffic rate of four packets per second. This results from the congestion caused by the increase in traffic. As the networks get larger, several nodes

have a large number of neighbors, causing congestion at the lower layers. There is also a problem with some nodes being partitioned from the network.



(a) Low Mobility

(b) Medium Mobility

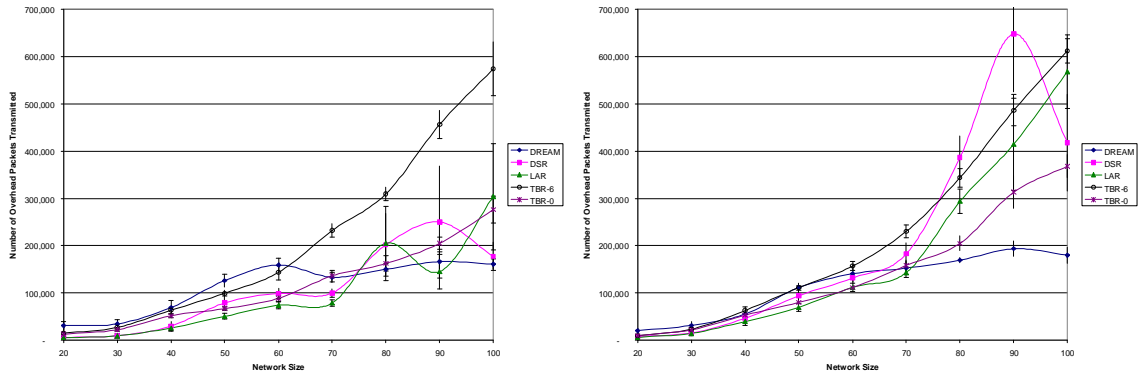


(c) High Mobility

Figure 37: Packet Delivery Efficiency in Hilly Terrain at 4 Packets/Second

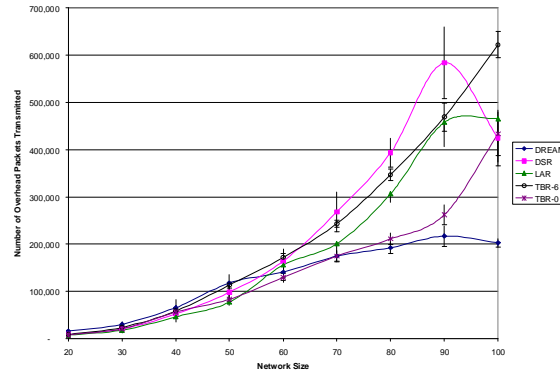
In the 80 node low mobility scenario, DSR and LAR show a large drop in delivery efficiency due to a doubling of the overhead from the 70 node scenario. In both cases, the individual simulation runs had a large variation in the amount of overhead. Large overhead in a few of the runs caused the average to be skewed higher than the 90 and 100 node averages causing the dip. As seen in Figure 38, a large variation occurred in the 80, 90 and 100 node networks at medium and high mobility but the average

amount of overhead remained roughly the same across all three networks so the dip does not occur in the medium and high mobility scenarios.



(a) Low Mobility

(b) Medium Mobility



(c) High Mobility

Figure 38: Number of Overhead Packets in Hilly Terrain at 4 Packets/Second

7.2.3 Comparing Flat and Hilly Terrain Performance

In experiment 2, we examined how TBR performed relative to DREAM, DSR, and LAR. The terrain had a significant impact on the ability of all protocols to deliver packets. The hilly terrain caused frequent link failures which greatly increased the average delay compared to the flat terrain. Both TBR-0 and TBR-6 provided significantly better performance than DSR and LAR in hilly terrain. Some of this is due amount of overhead DSR & LAR generated as the MAC layer became congested.

However, even at low traffic rates with no congestion, TBR-6 outperformed DSR & LAR. TBR-6 also benefited from the sudden link failures. The hilly terrain eliminated many of the low quality links that were present in the flat terrain. As a result, the path loss prediction errors in TBR-6 had much less impact in hilly terrain.

In general, both TBR versions were able to handle the mobility better than the other protocols in both flat and hilly terrain due to their mobility prediction algorithms which helped identify route failures without generating overhead packets. The cost was an increase in the average delay, due in large part to the slightly longer paths the protocols supported and the longer time required for discovering routes. In flat terrain, TBR-0, DSR and LAR performed very similarly with TBR-0 winning out in the packet delivery efficiency due to the low amount of overhead packets resulting from the route discovery reply limitations discussed previously. As was demonstrated in Experiment 1, TBR-6 performed significantly worse than TBR-0 and the remaining protocols. However, as the mobility increased, TBR-6's performance did not degrade as rapidly as DSR and LAR and at highly mobility TBR-6 closely matched DSR and LAR in performance. In fact in the flat terrain case, for larger networks, TBR-6's performance was better than what was seen in Experiment 1.

CHAPTER 8

CONCLUSIONS AND CONTRIBUTIONS

8.1 Summary of Experiment 1: Effects of Path Loss Prediction Error on TBR Performance

In general, the prediction error has less impact on performance in hilly terrain. Looking at the percentage change from perfect prediction to 20 dB of error, the relative impact of the prediction error is greater in flat terrain than in hilly terrain (66.3% decrease in packet delivery efficiency versus a 62.6% decrease). This primarily results from the nature of the links. In flat terrain, link failures only occur when the distance between two nodes is greater than the transmission range. The decrease in link quality occurs gradually, and the prediction errors can cause TBR to incorrectly determine that the status of these marginal links has changed when in actuality it hasn't. In contrast, the links in hilly terrain are broken rather suddenly as the terrain affects the link quality. Typically, links are either up with good quality or they are down. Because of this, the prediction errors have less of an impact. If the link is down, the prediction error is unlikely to change the predicted path loss to make the link appear up. Conversely, if a link exists, the link probably has a high quality link and an error in the prediction is unlikely to change whether the link exists or not.

The reduced connectivity in the hilly terrain also helps to reduce the number of nodes receiving broadcasts. This hinders packet delivery but also serves to limit the amount of overhead packets that are broadcast. In the flat terrain, as the prediction error grows from 0 to 20 dB, the average number of packets transmitted grows 88% while the number of overhead packets transmitted grows 213.7%. In hilly terrain, the number of

overhead packets transmitted and total number of packets transmitted both grow about 58%.

In the hilly terrain, the end to end delay is dominated by the time required to discover a route and rediscover the route when a link breaks rather than to delays resulting from the prediction errors. In contrast, with flat terrain, the prediction errors cause non optimal routes to be selected resulting in increased delay. The higher initial throughput is due to its ability to support longer paths in flat terrain, particularly in the medium and high mobility cases. As the prediction error grows, TBR is unable to maintain those longer paths as the prediction errors increasingly reduce the efficiency of the protocol.

8.2 Summary of Experiment 2: Comparison of DREAM, DSR, & LAR to TBR

The results of the Experiment 2 simulations provided valuable insight into the performance of the TBR algorithm. The simulation results support our initial claims that TBR would operate well in networks with bandwidth or channel access constraints. We also showed that while the TBR worked extremely well with no prediction errors, its performance decreases substantially when it operated in the presence of prediction errors. However, in hilly terrain, TBR-6's performance was acceptable, and in many cases superior to DSR and LAR.

Because of the level transmitted power required to establish long links, there was a large variability in the number of neighbors. In large networks in particular, the connectivity was much higher than desired. This led to congestion at the MAC layer causing the results to be skewed. The MAC layer congestion had a much larger impact on LAR and DSR than on the remaining protocols.

The research results provide a good characterization on the expected performance of TBR in a variety of scenarios. TBR performs well when the traffic load provides enough location information in the data packets for TBR to maintain fairly complete view

of the network. Therefore, TBR would perform much better in networks where the traffic is well distributed with many sets of source and destinations, rather than in a network where traffic travels between small groups of nodes. In the scenarios tested, the traffic flows were one way. The source transmitted packets and the destination provided no feedback. Without the two way flow of traffic, the location information for the destination gets stale over time, reducing the effectiveness of TBR.

TBR also functions better in the hilly terrain where link failures occur rapidly. When the network has large numbers of low quality links, the prediction errors have a much greater impact on TBR's performance.

Because of the amount of congestion, the simulation results for large networks and high traffic rates cannot be considered conclusive. Although TBR's performance is superior, much of its gain is due to the reduced overhead of limiting who can respond to the route discovery. When TBR was modified to allow intermediate nodes to respond, it suffered similar performance reductions to LAR and DSR. Therefore, in order to conclusively and fairly compare the protocols, further experiments with reduced connectivity and congestion are needed. However, the simulations performed do provide strong evidence that TBR would perform better since at lower traffic rates, TBR-0 offers superior performance, primarily in the packet delivery efficiency due to reduced overhead. Unfortunately, in this range, TBR-6 proved inferior to DSR and LAR in flat terrain and similar performance in hilly terrain.

8.3 Contributions

Our research makes several contributions. We developed a novel routing framework using location & terrain data combined with mobility prediction and network feedback on link failures. Using this routing framework, we demonstrated the feasibility of using terrain and location information to improve routing in ad hoc networks. We in turn characterized impact of path loss prediction errors in the proposed algorithm.

Through simulation, we demonstrated that although TBR is superior to current MANET algorithms, prediction errors significantly reduce the effectiveness of the algorithm.

CHAPTER 9

FUTURE WORK

While the TBR protocol has been shown to outperform the protocols it was compared with, there is room for improvement and future work. While TBR-0 (i.e. no prediction error) outperformed DREAM, DSR and LAR, TBR-6 only occasionally outperformed them. As noted in the results section, when TBR operated with prediction errors, the protocol did not perform satisfactorily in large networks with high offered load. Future work should focus on refining the TBR process and examining the impact of changing the timeout values for the link failure matrix entries as well as the timeout values for expiring old location data. Currently, the location data is expired after a fixed amount of time. Changing the timeout value to be a function of the speed that the node is moving may improve performance.

As noted in CHAPTER 7, TBR the route maintenance function could be improved to better handle special cases where the routing tables on two different nodes are inconsistent and create a temporary routing loop. This problem manifests only in route maintenance when two neighbors both believe that the other is on the shortest path back to the source. In such a case, the route maintenance packet could end up in a loop. The protocol currently just discards the packet to ensure the looping does not occur. This works well when packets are not arriving too quickly since the inconsistency is typically short lived. However, as the packet arrival rate increases, a significant number of packets may be dropped.

As mentioned in CHAPTER 7, the mobility prediction algorithm can be an important factor in performance, particularly at high mobility rates. The current mobility model uses a simple linear prediction algorithm to predict the future locations. There have been a number of techniques proposed and developed for predicting mobility.

Mobility prediction models using Kalman filtering [49],[51] have been shown to provide good prediction accuracy. Additionally, there is a significant amount of mobility prediction research in the literature for cellular and ad hoc networks [51][52].

Another area for further research is determining the optimal rate for distributing location data. Currently, each node attaches its location data on every data packet. This results in a significant amount of unnecessary overhead if the nodes are moving slowly. Dynamically adjusting the frequency that the location information is included with the data packets could significantly decrease the amount of overhead in terms of bits. The potential savings depends greatly on the packet generation rate and the average size of the data packet as well as the speed at which the node is traveling. This technique would have little if any impact on the packet delivery efficiency since the location information is sent in the data packets.

A fifth area of possible future work is to examine the impact of different mobility models. The random waypoint models used in this research does not reflect mobility patterns seen in the real world but is a popular model for MANET research simulations. There is a growing body of different mobility models [48][53] that could be used to further test TBR's performance.

A further area of research is the impact the choice of propagation model has on protocol's performance. In the current work, the prediction algorithm is the same algorithm as the underlying simulation engine. We introduced random errors into each call to the propagation model. As noted previously, this may impact the performance of the protocol since the node will produce different results using the same input data (location information). A final area of future work is to examine the impact of having a different algorithm for TBR's propagation model and the simulation engine's propagation model.

APPENDIX A

DETAILED RESULTS

9.1 Graphs for Experiment 1: Effects of Prediction Error in Flat Terrain

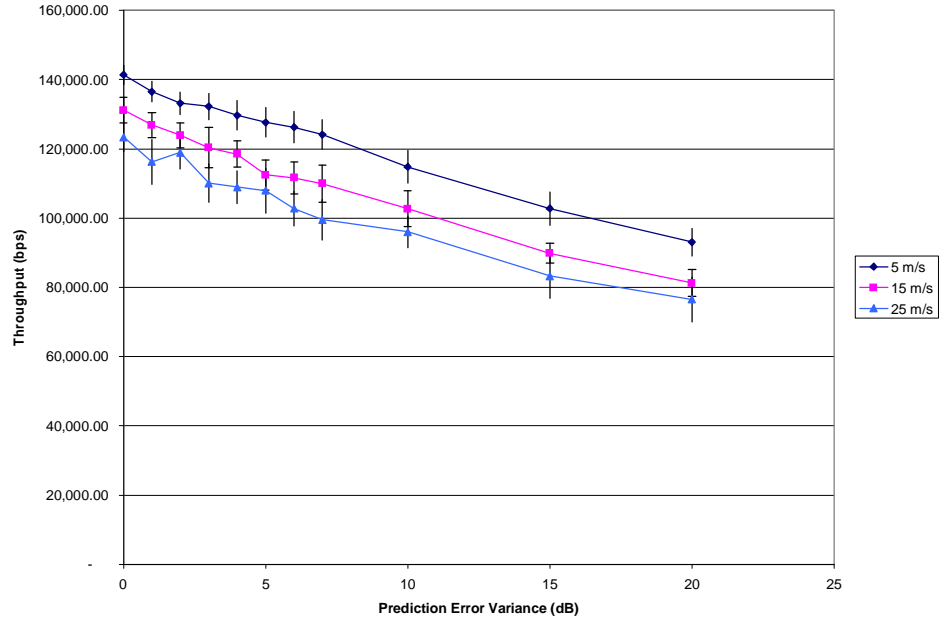


Figure 39: Exp 1: Throughput in Flat Terrain

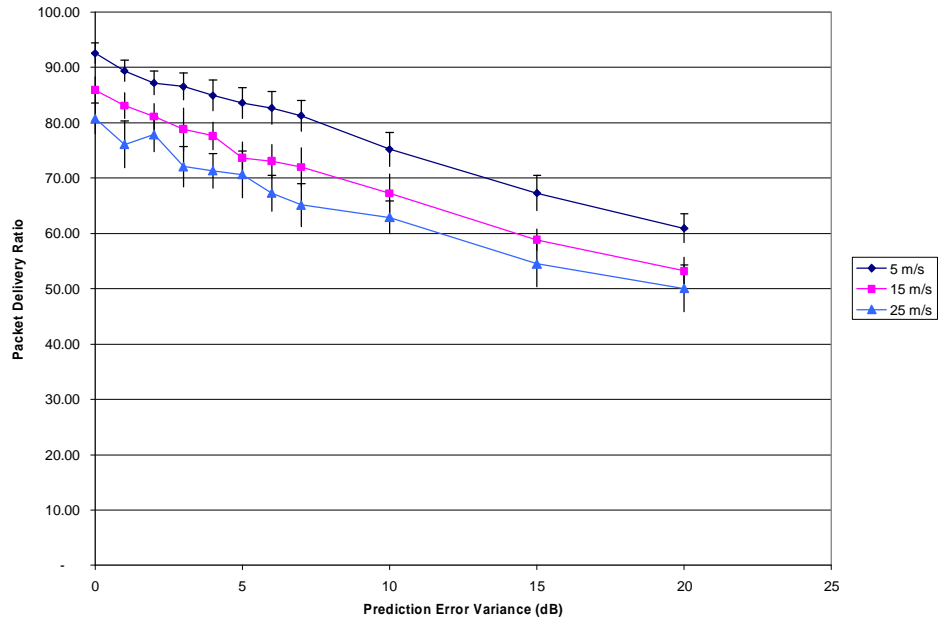


Figure 40: Exp 1: Packet Delivery Ratio in Flat Terrain

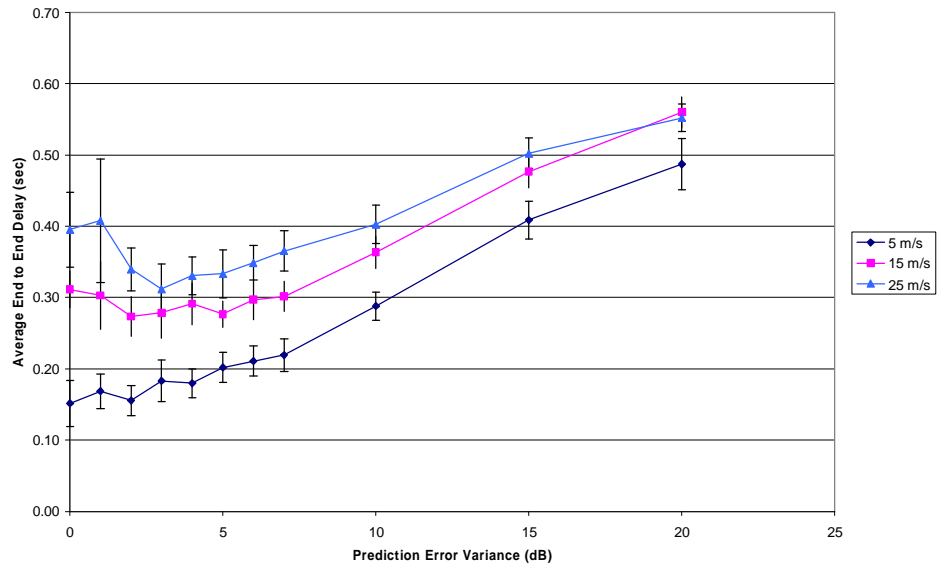


Figure 41: Exp 1: Average End to End Delay in Flat Terrain

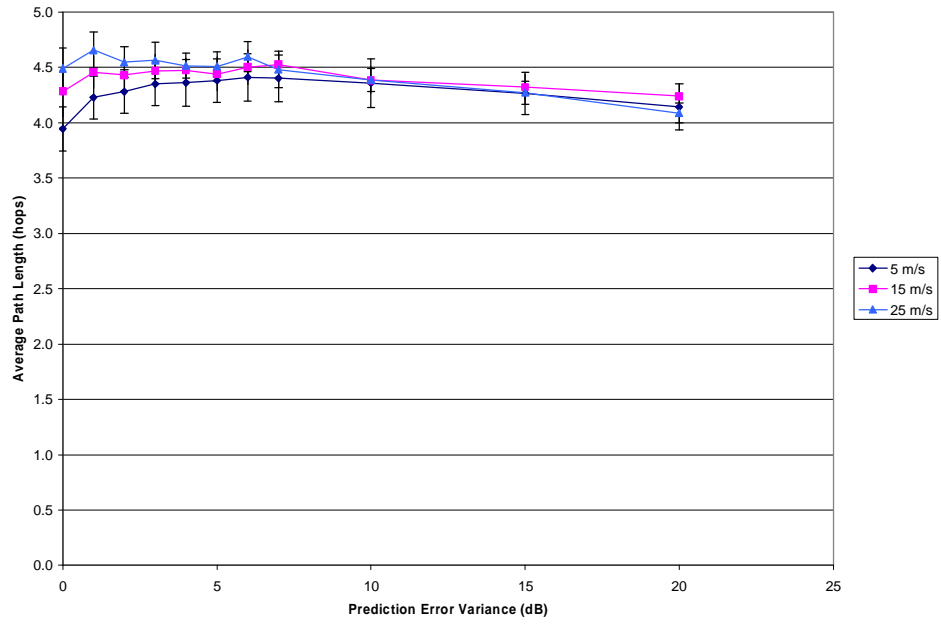


Figure 42: Exp 1: Average Path Length in Flat Terrain

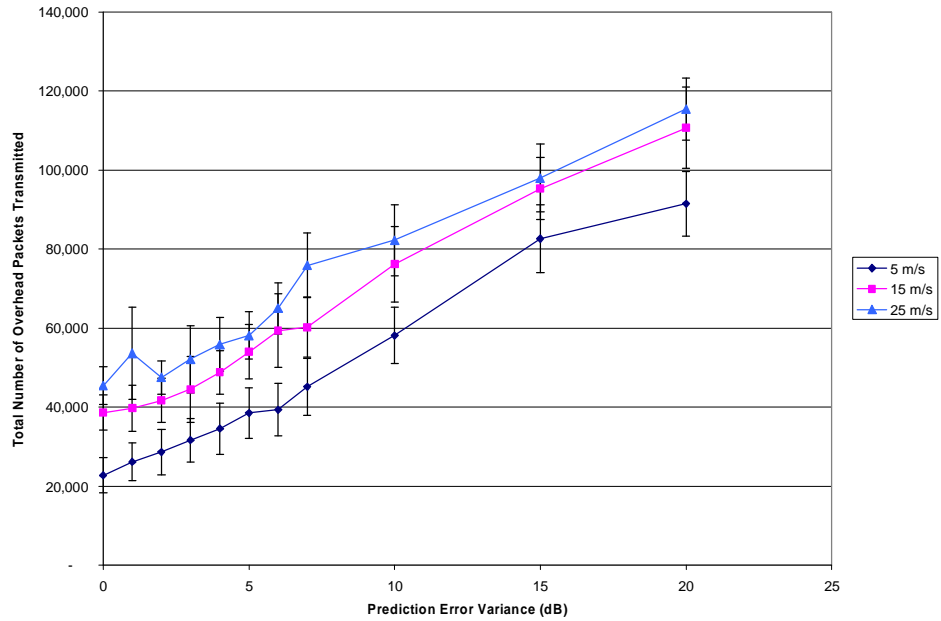


Figure 43: Exp 1: Number of Overhead Packets Transmitted in Flat Terrain

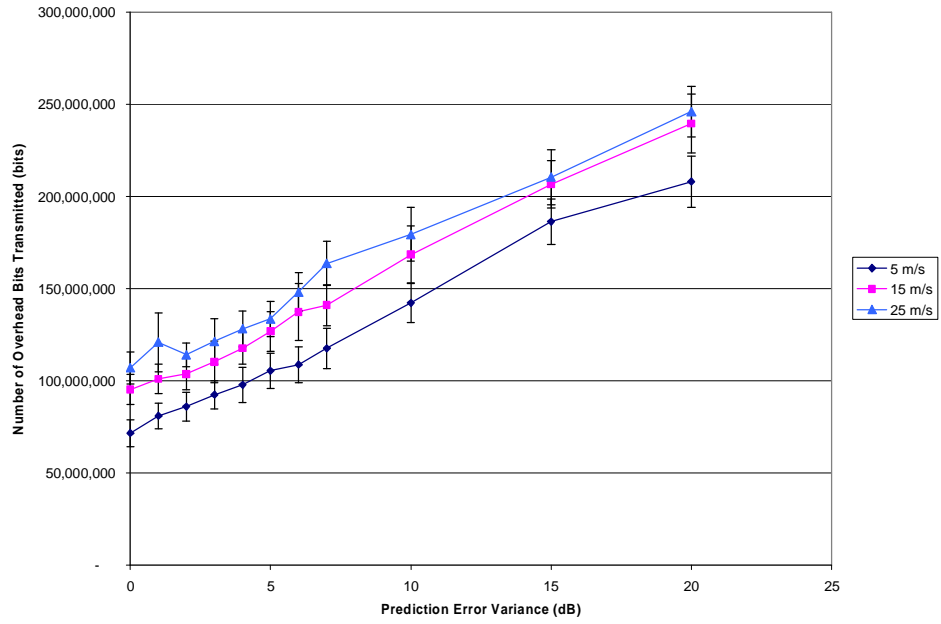


Figure 44: Exp 1: Total Number of Overhead Bits in Flat Terrain

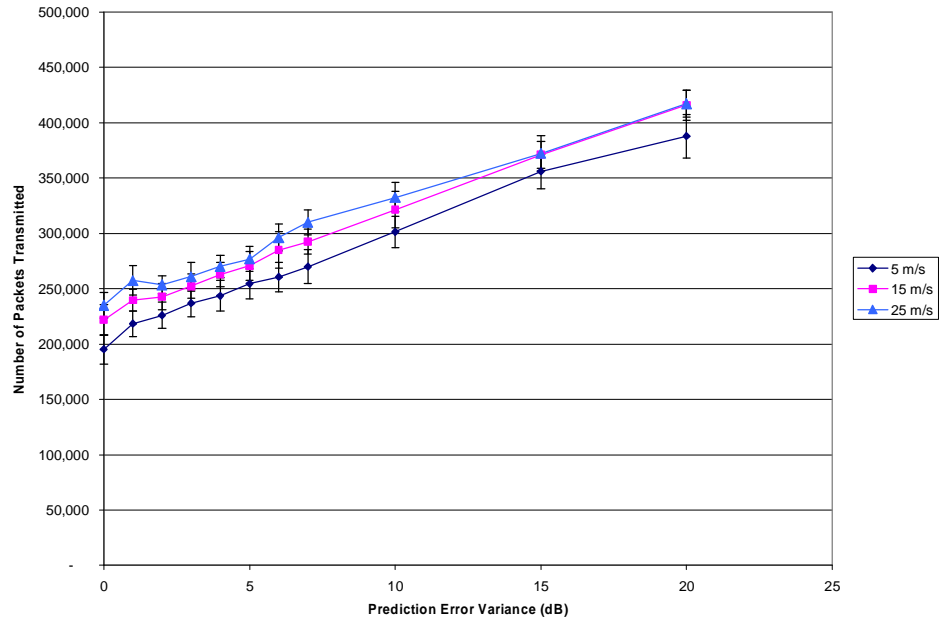


Figure 45: Exp 1: Total Number of Packets Transmitted in Flat Terrain

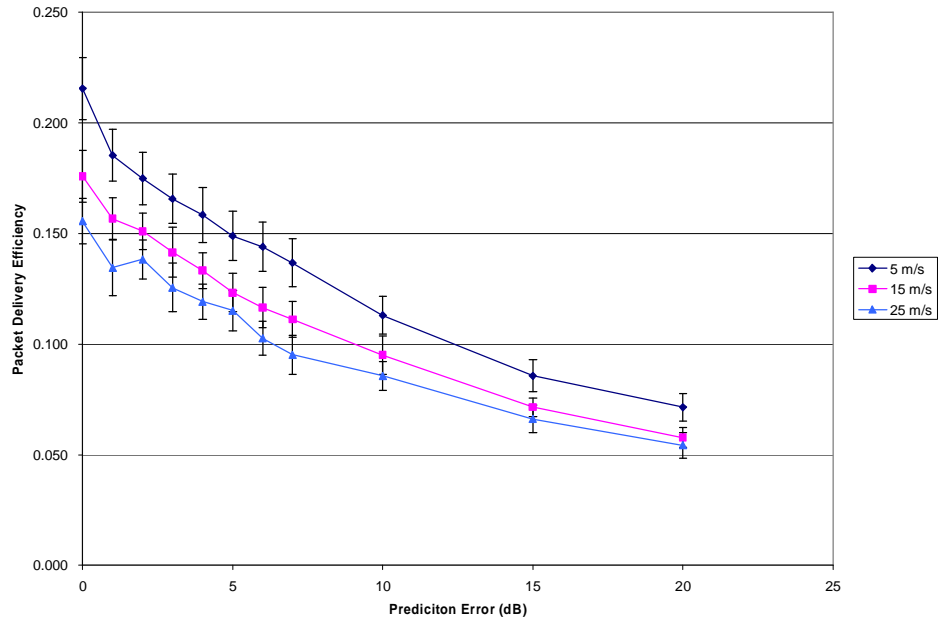


Figure 46: Exp 1: Packet Delivery Efficiency in Flat Terrain

9.2 Graphs for Experiment 1: Effects of Prediction Error in Hilly Terrain

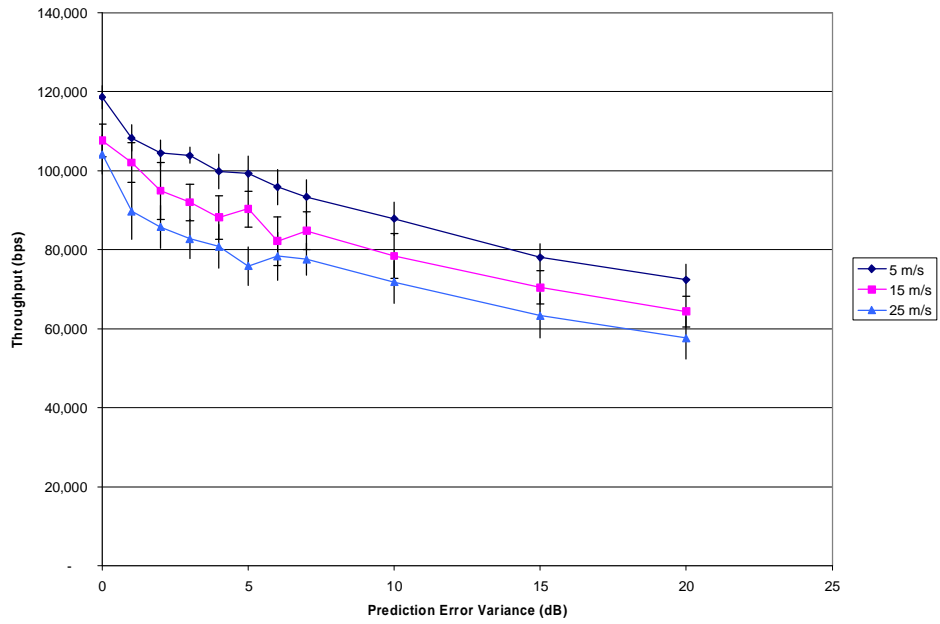


Figure 47: Exp 1: Throughput in Hilly Terrain

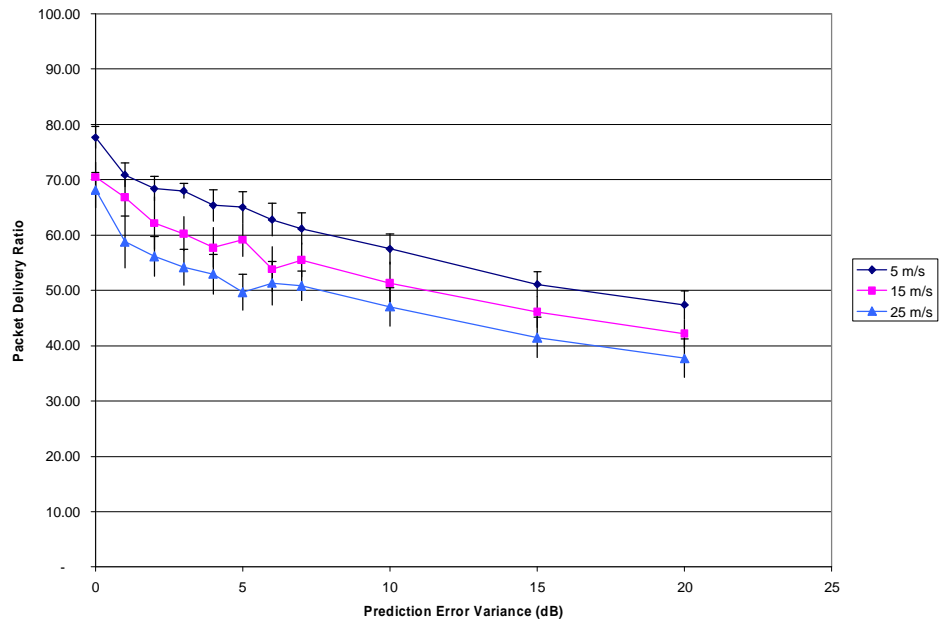


Figure 48: Exp 1: Packet Delivery Ratio in Hilly Terrain

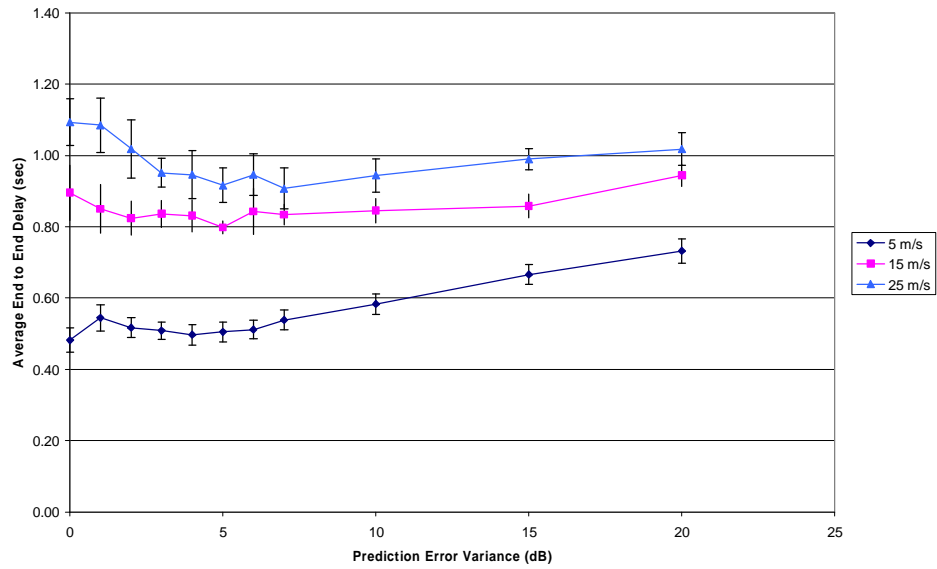


Figure 49: Exp 1: Average End to End Delay in Hilly Terrain

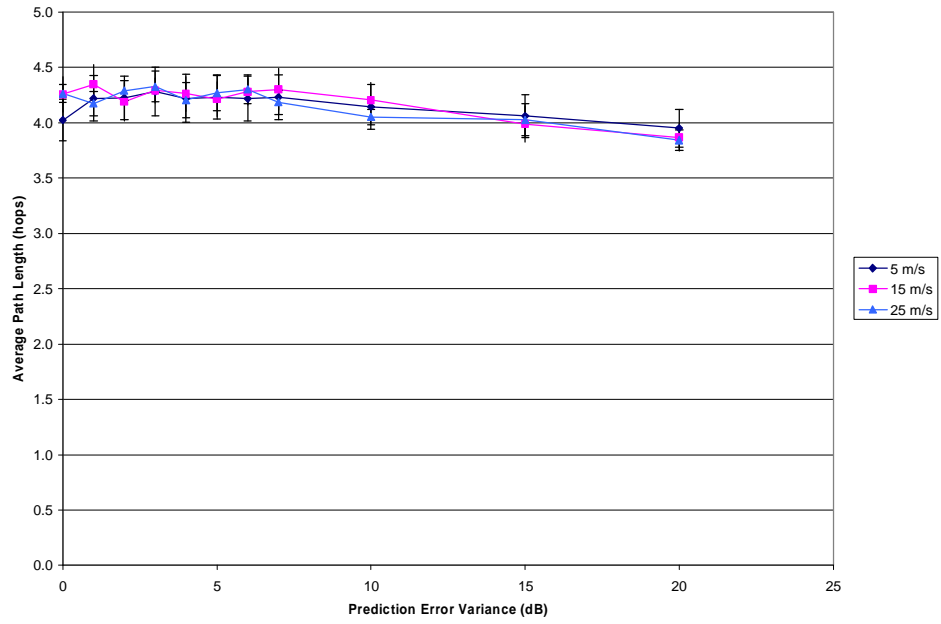


Figure 50: Exp 1: Average Path Length in Hilly Terrain

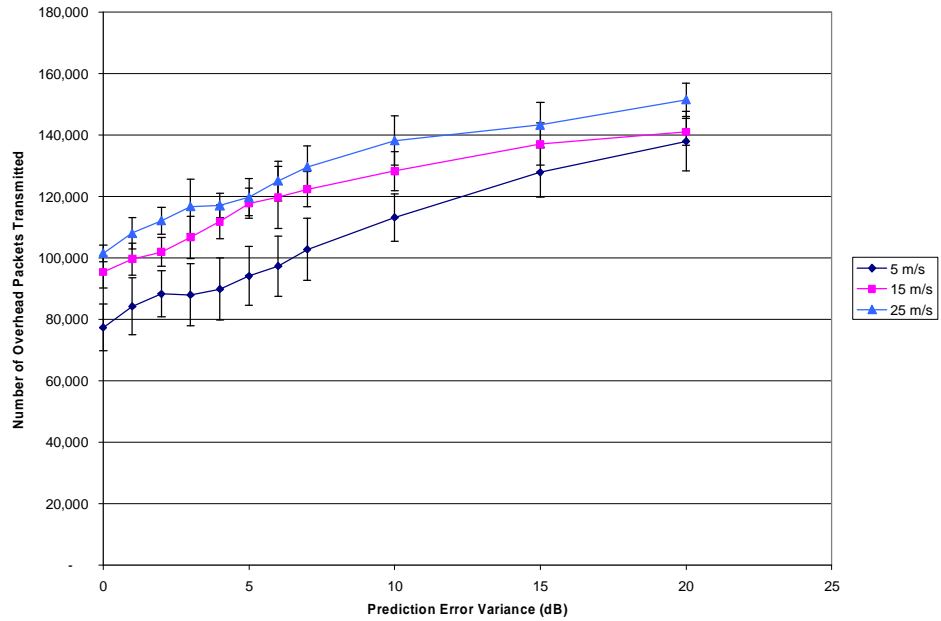


Figure 51: Exp 1: Total Number of Overhead Packets in Hilly terrain

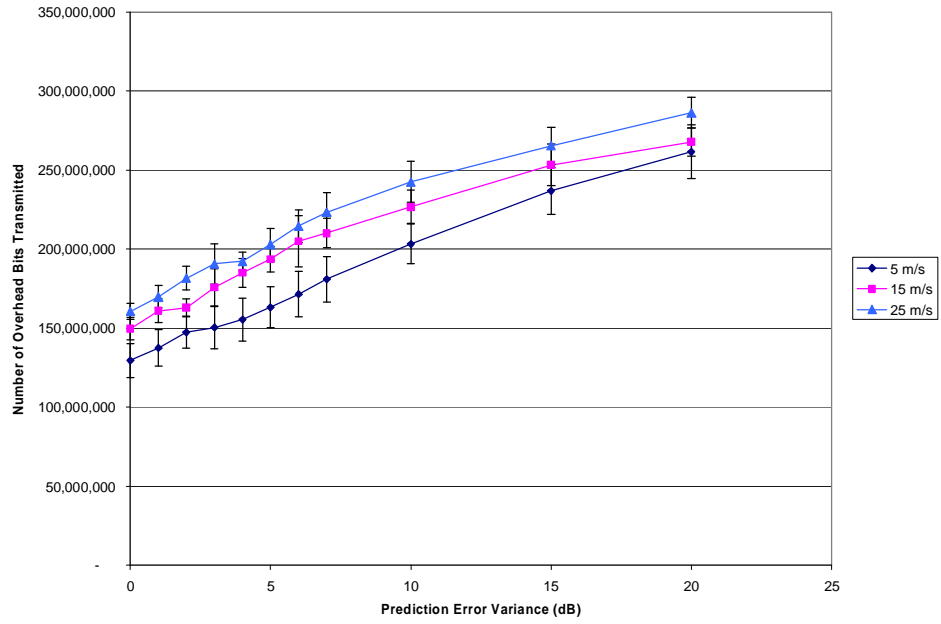


Figure 52: Exp 1: Total Number of Overhead Bits in Hilly Terrain

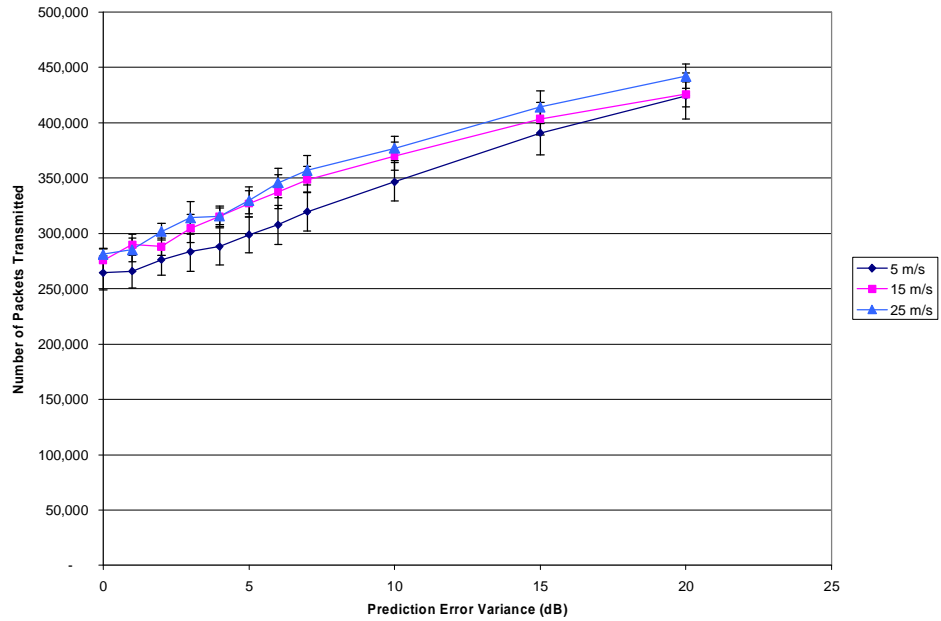


Figure 53: Exp 1: Total Number of Packets Transmitted in Hilly Terrain

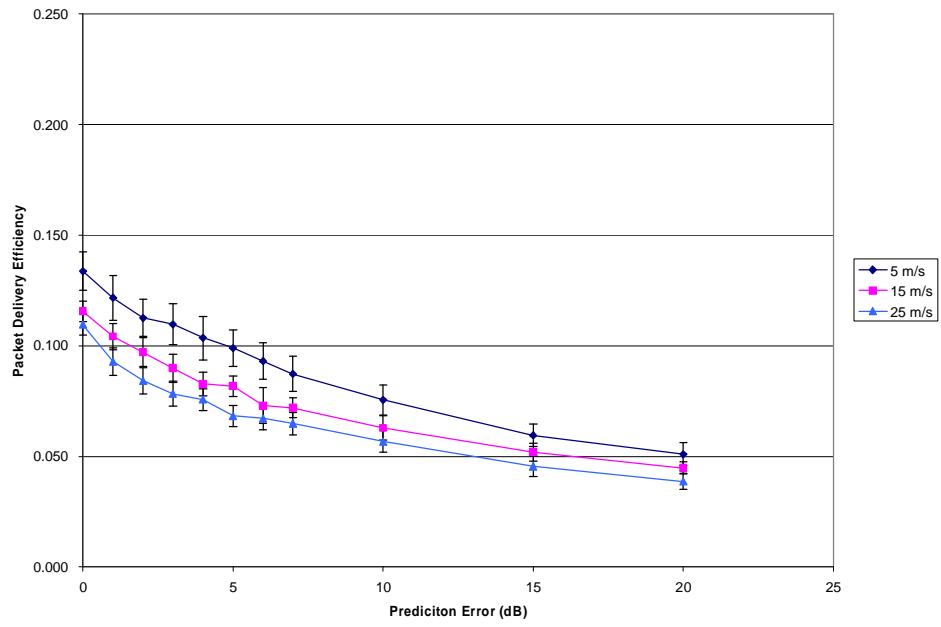


Figure 54: Exp 1: Packet Delivery Efficiency in Hilly Terrain

9.3 Graphs for Experiment 2 Performance Comparison in Flat Terrain

9.3.1 Flat Terrain, 1 Packet/Second

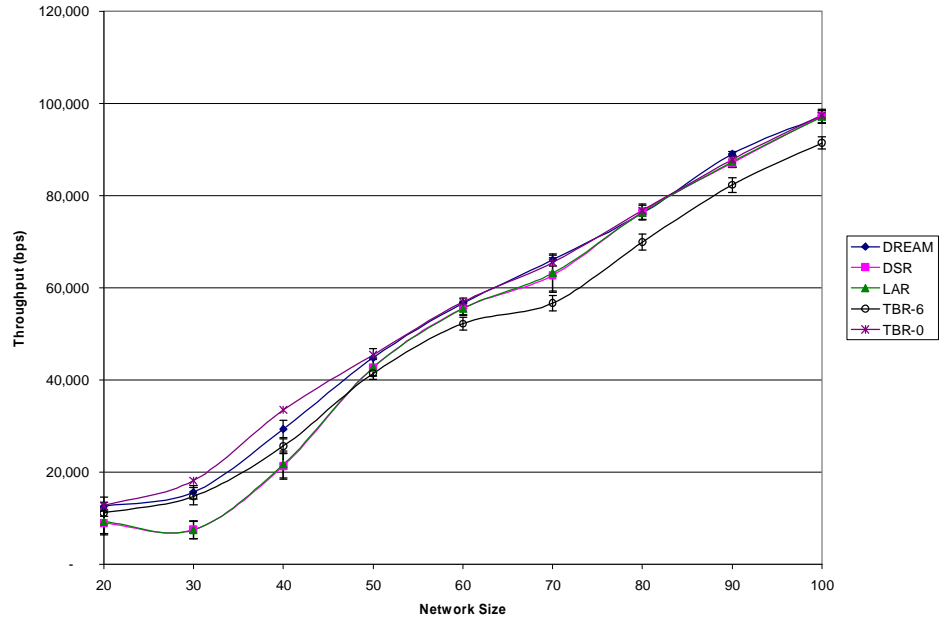


Figure 55: Throughput (Flat, Low Mobility, 1 pkt/sec)

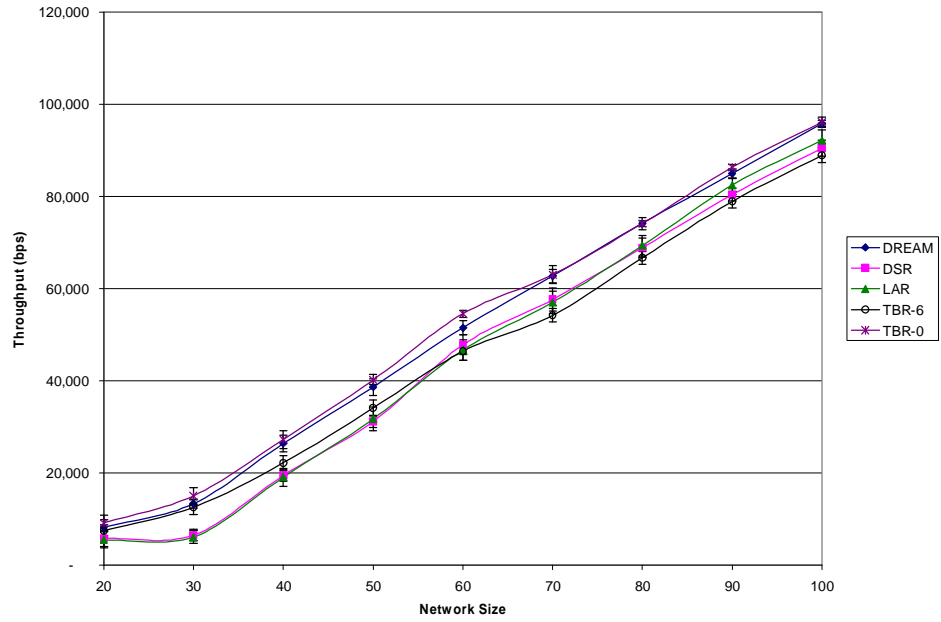


Figure 56: Throughput (Flat, Medium Mobility, 1 pkt/sec)

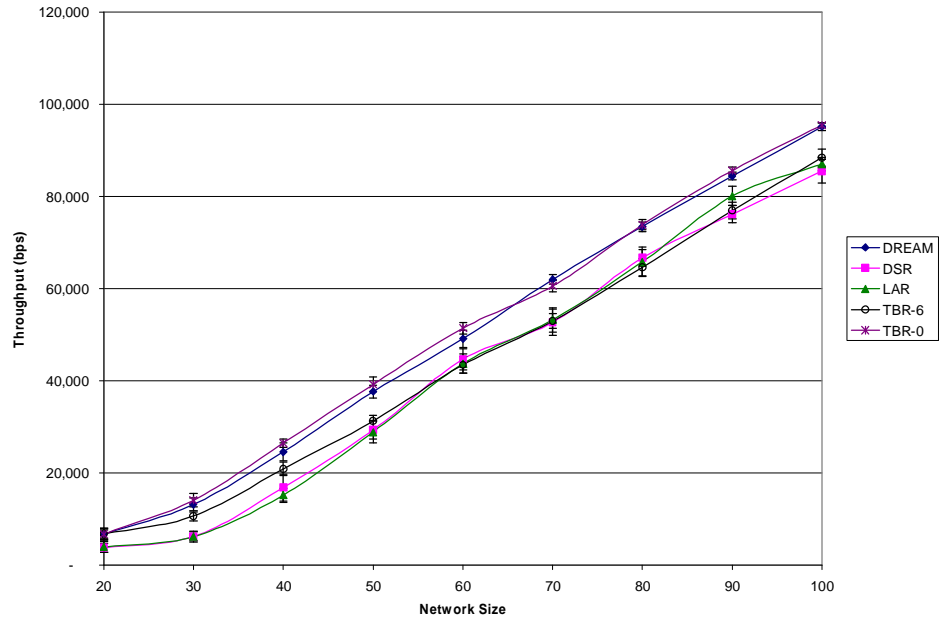


Figure 57: Throughput (Flat, High Mobility, 1 pkt/sec)

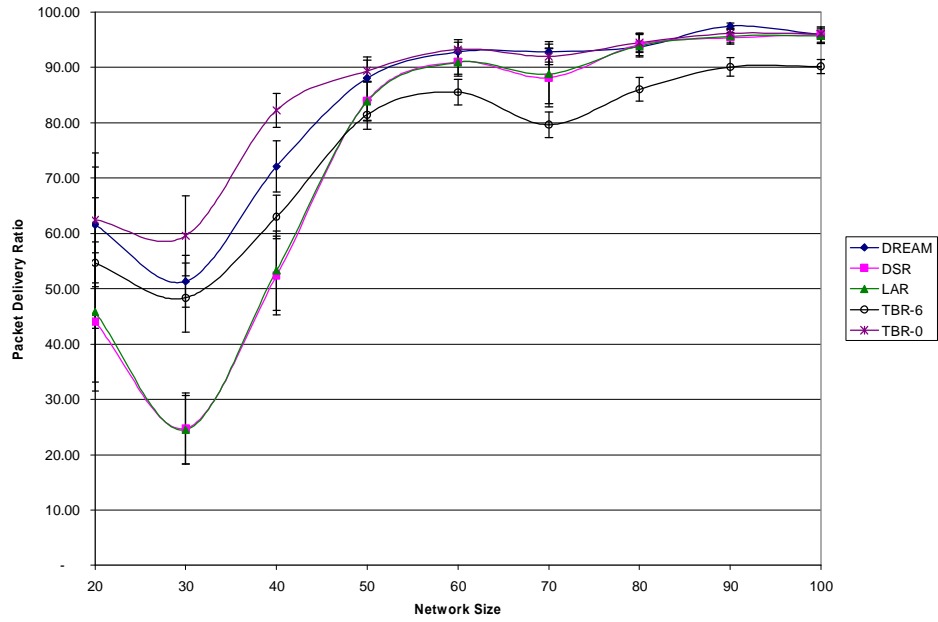


Figure 58: Packet Delivery Ratio (Flat, Low Mobility, 1 pkt/sec)

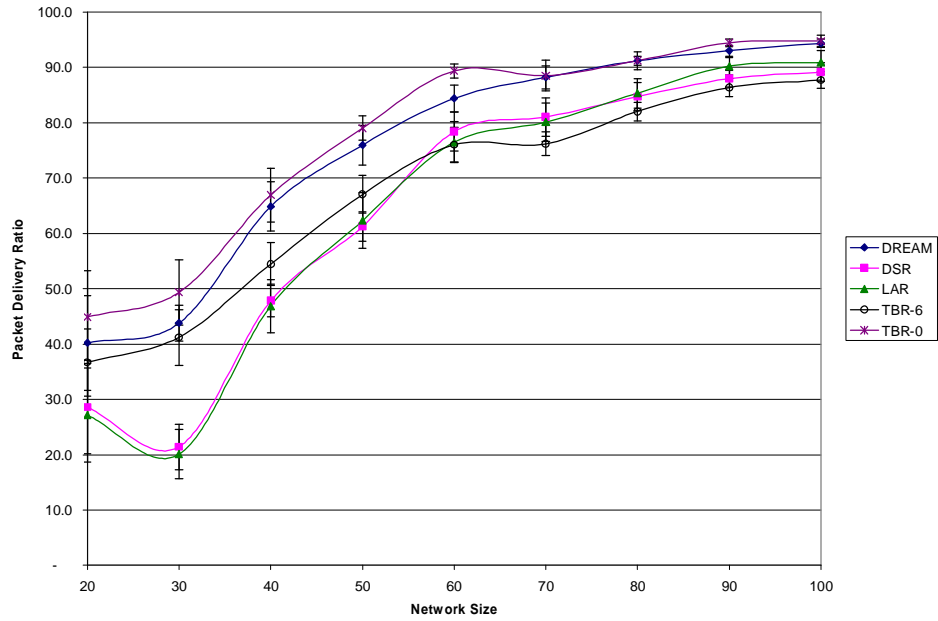


Figure 59: Packet Delivery Ratio (Flat, Medium Mobility, 1 pkt/sec)

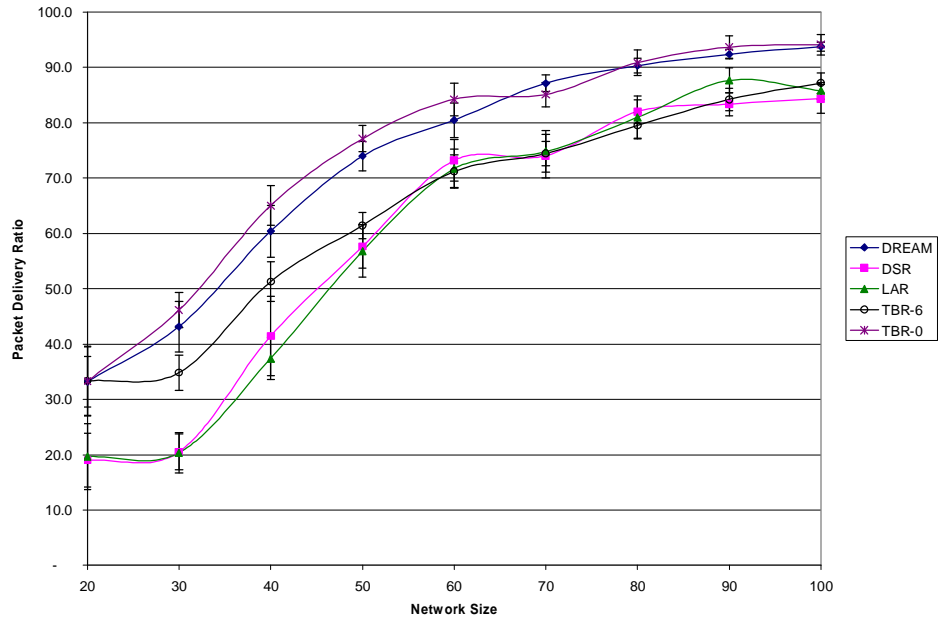


Figure 60: Packet Delivery Ratio (Flat, High Mobility, 1 pkt/sec)

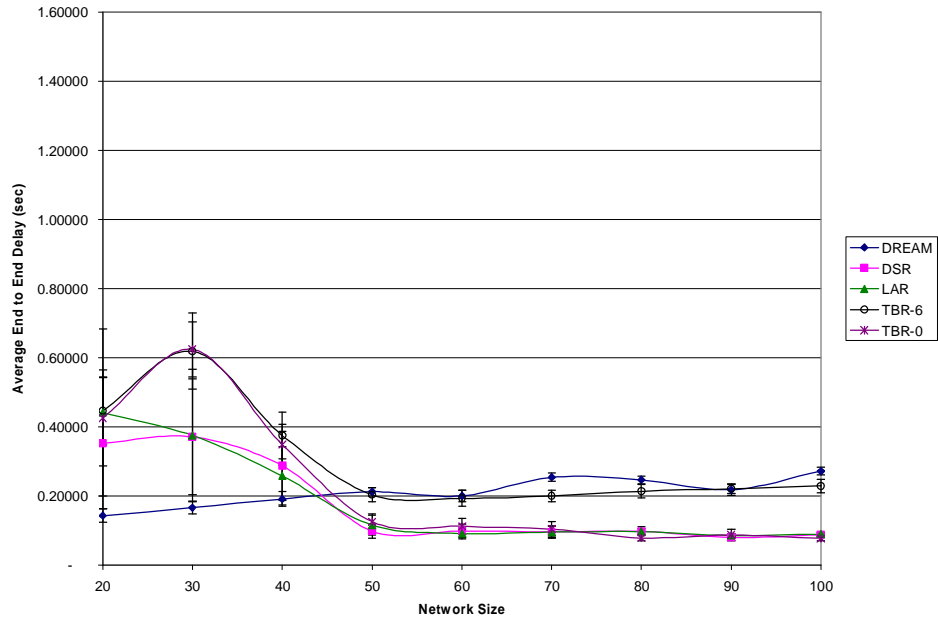


Figure 61: Average End to End Delay (Flat, Low Mobility, 1 pkt/sec)

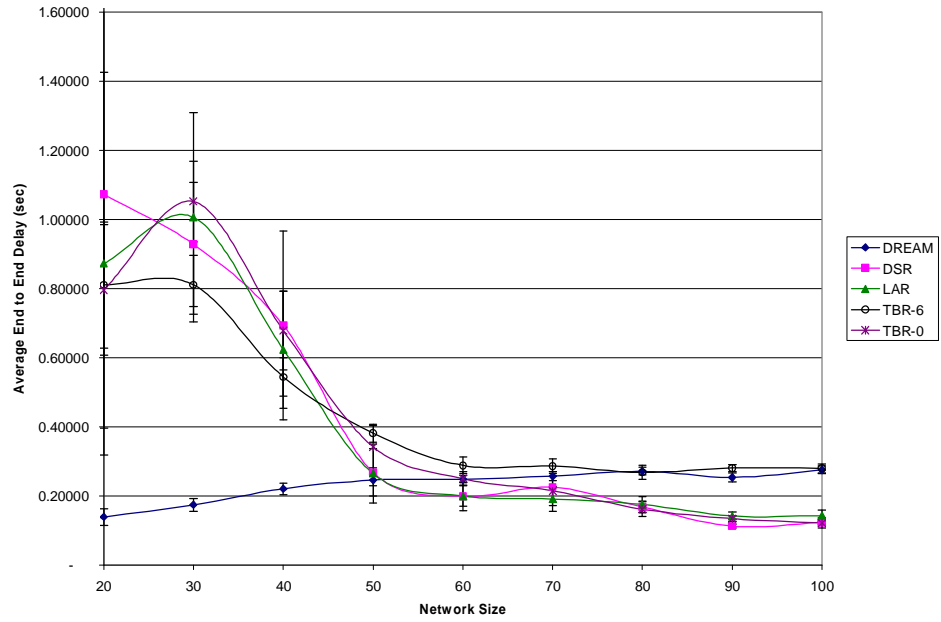


Figure 62: Average End to End Delay (Flat, Medium Mobility, 1 pkt/sec)

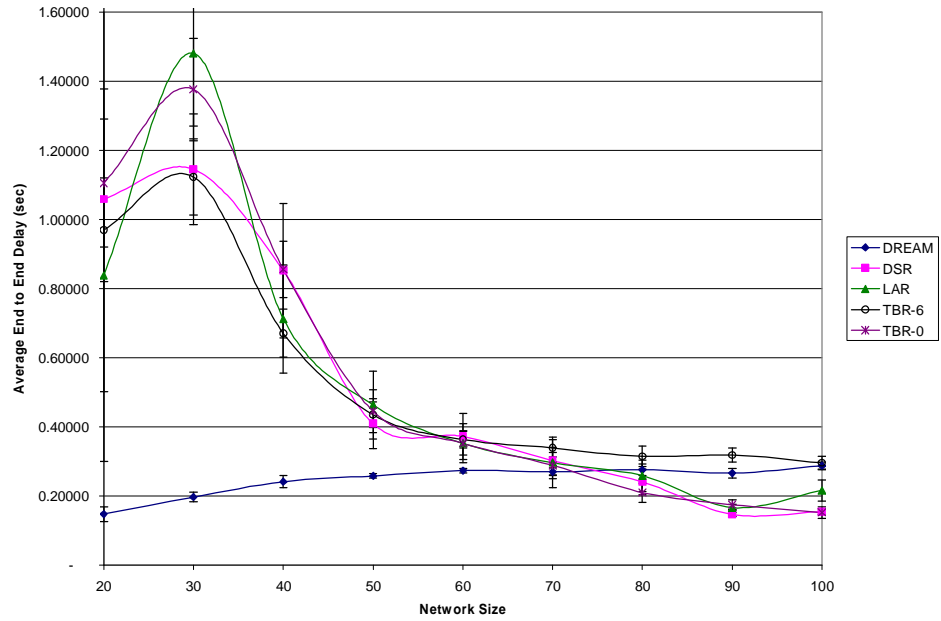


Figure 63: Average End to End Delay (Flat, High Mobility, 1 pkt/sec)

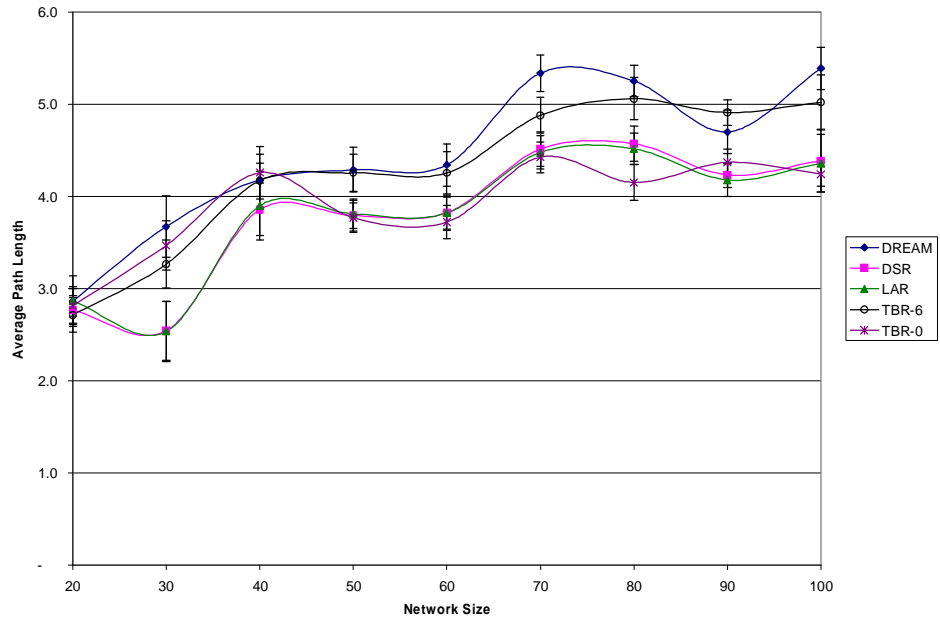


Figure 64: Average Path Length (Flat, Low Mobility, 1 pkt/sec)

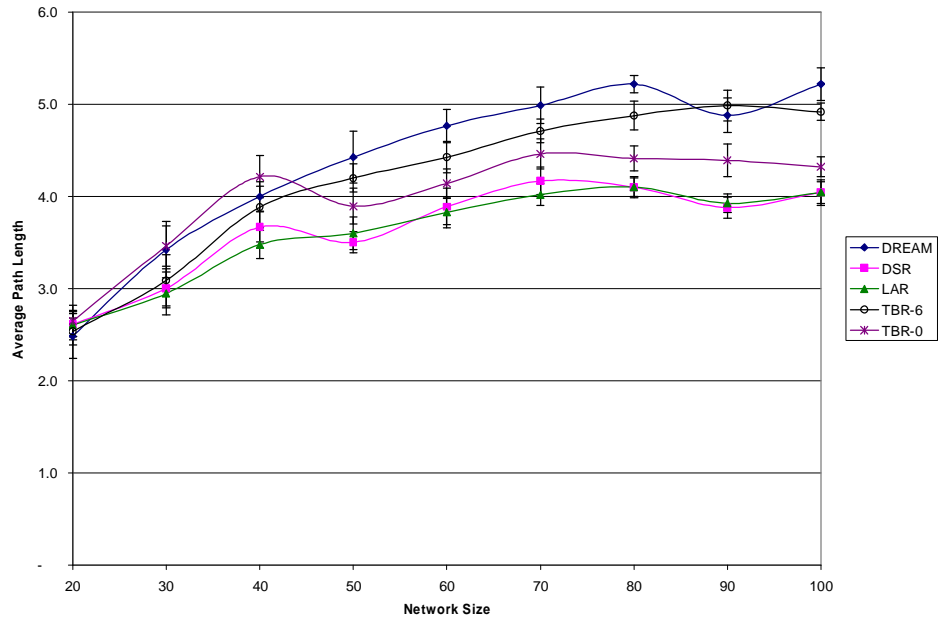


Figure 65: Average Path Length (Flat, Medium Mobility, 1 pkt/sec)

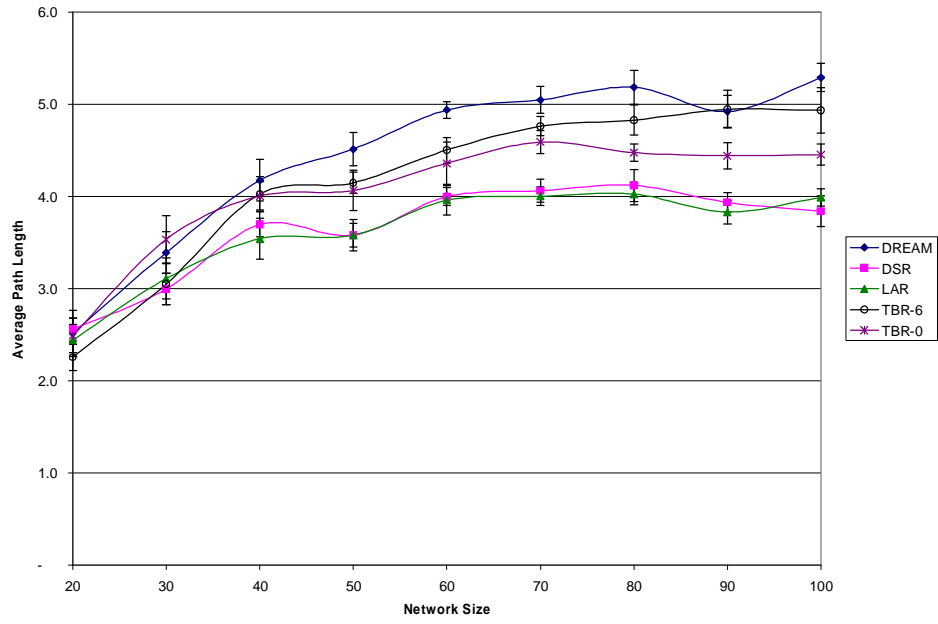


Figure 66: Average Path Length (Flat, High Mobility, 1 pkt/sec)

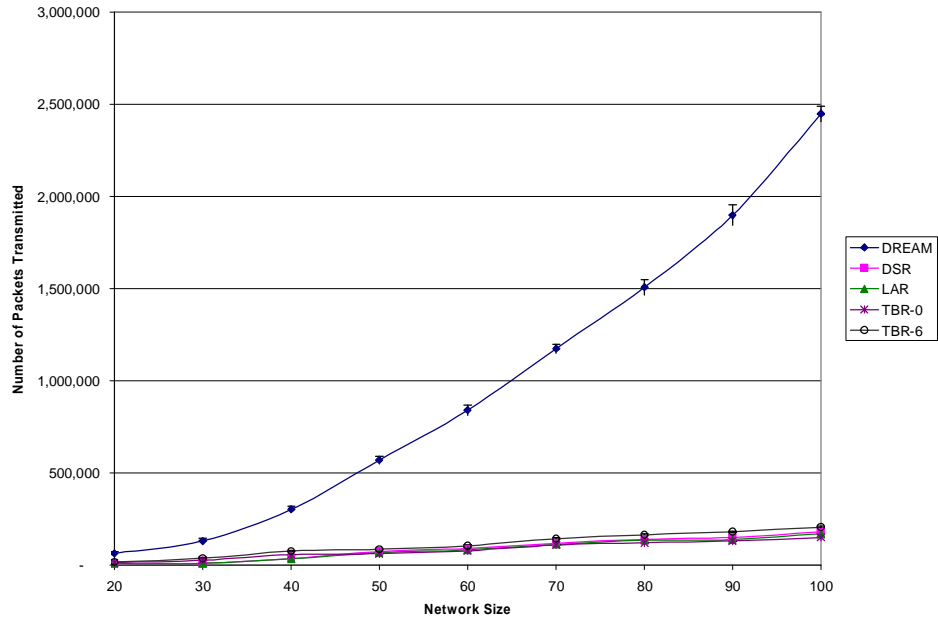


Figure 67: Number of Packets Transmitted (Flat, Low Mobility, 1 pkt/sec)

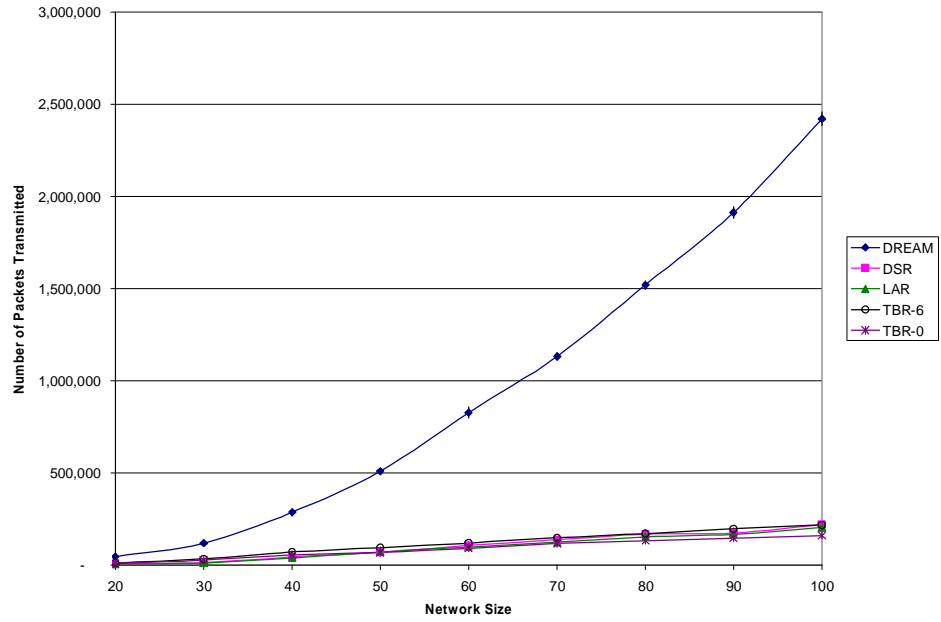


Figure 68: Number of Packets Transmitted (Flat, Medium Mobility, 1 pkt/sec)

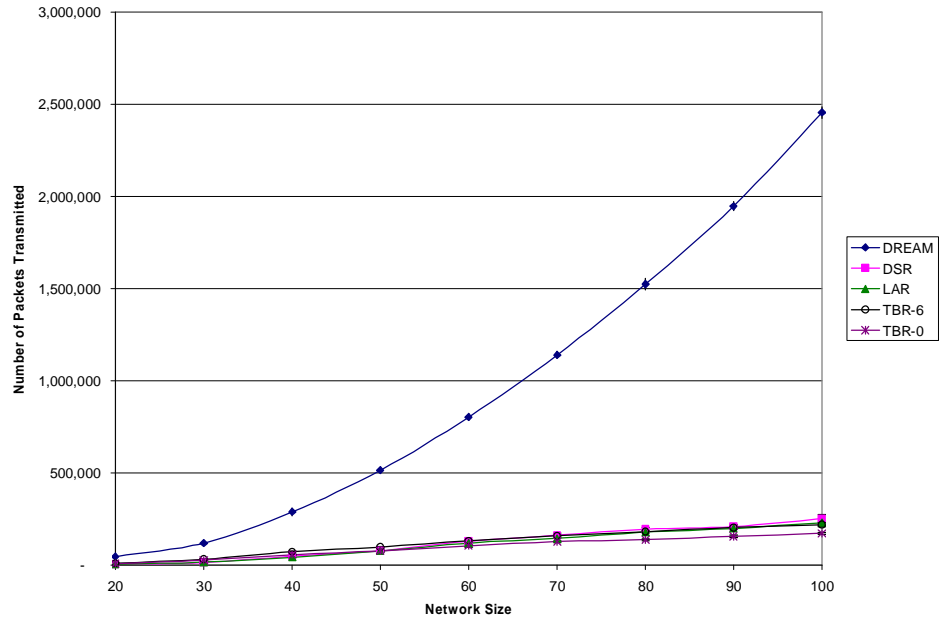


Figure 69: Number of Packets Transmitted (Flat, High Mobility, 1 pkt/sec)

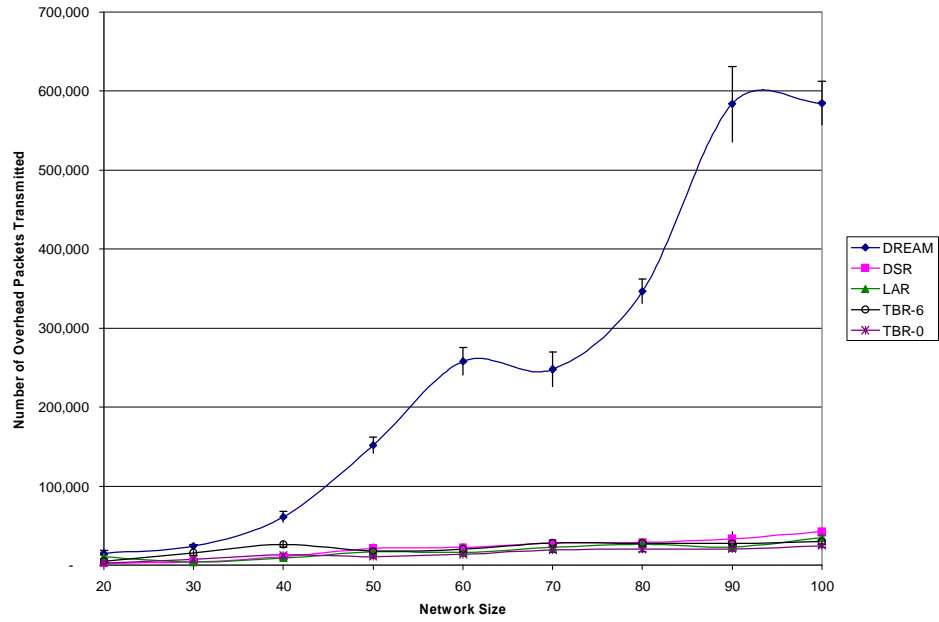


Figure 70: Number of Overhead Packets (Flat, Low Mobility, 1 pkt/sec)

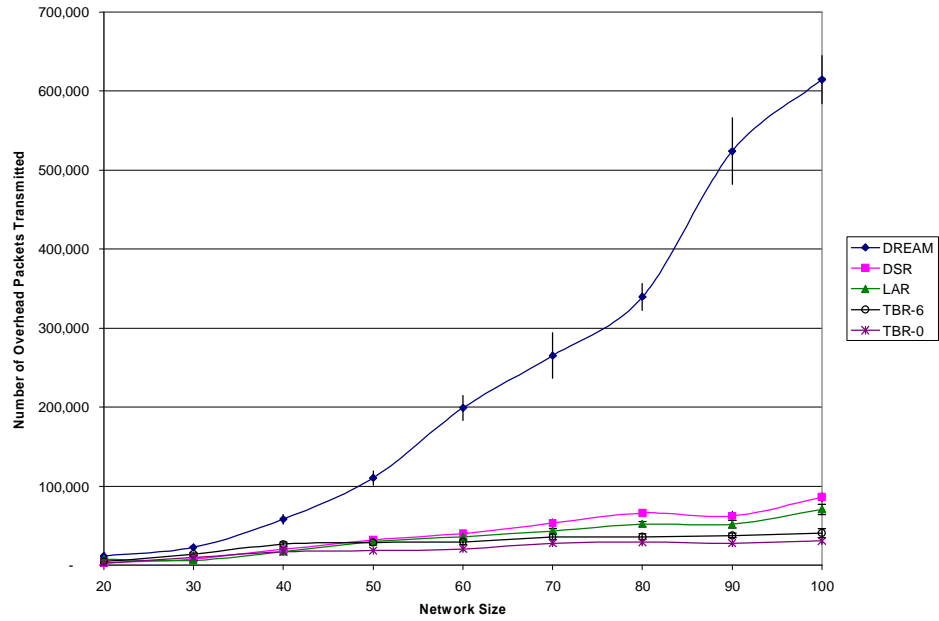


Figure 71: Number of Overhead Packets (Flat, Medium Mobility, 1 pkt/sec)

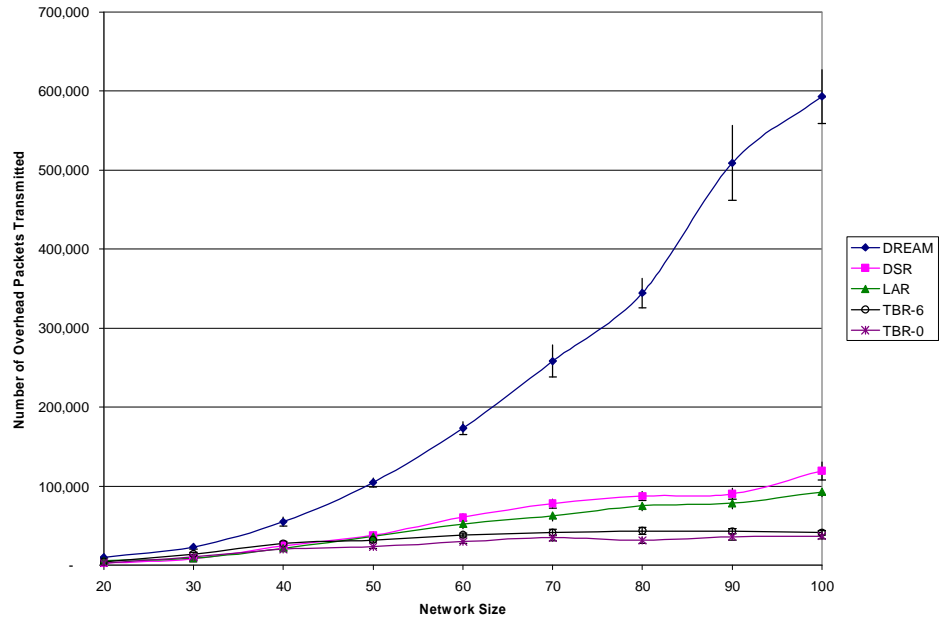


Figure 72: Number of Overhead Packets (Flat, High Mobility, 1 pkt/sec)

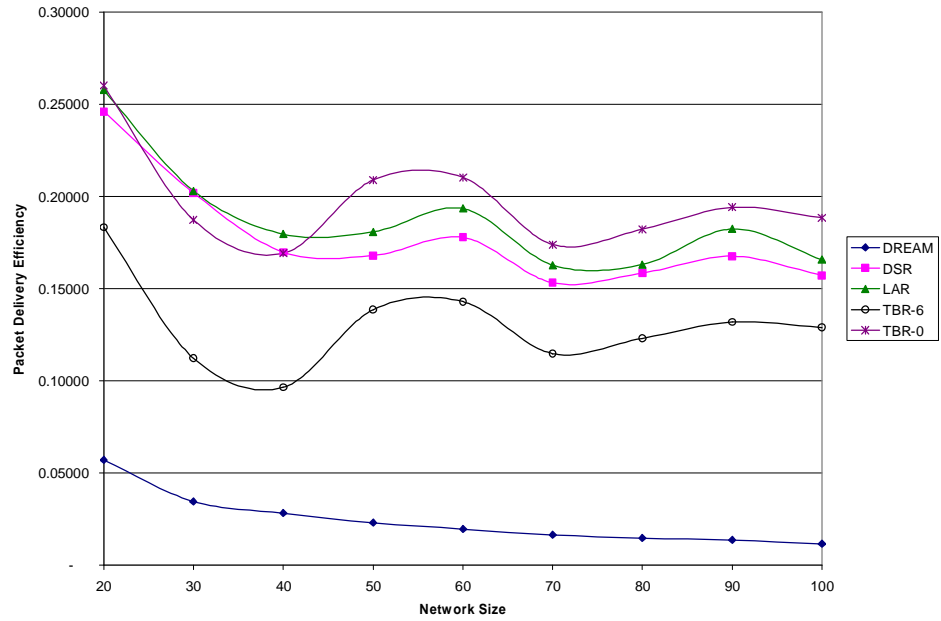


Figure 73: Packet Delivery Efficiency (Flat, Low Mobility, 1 pkt/sec)

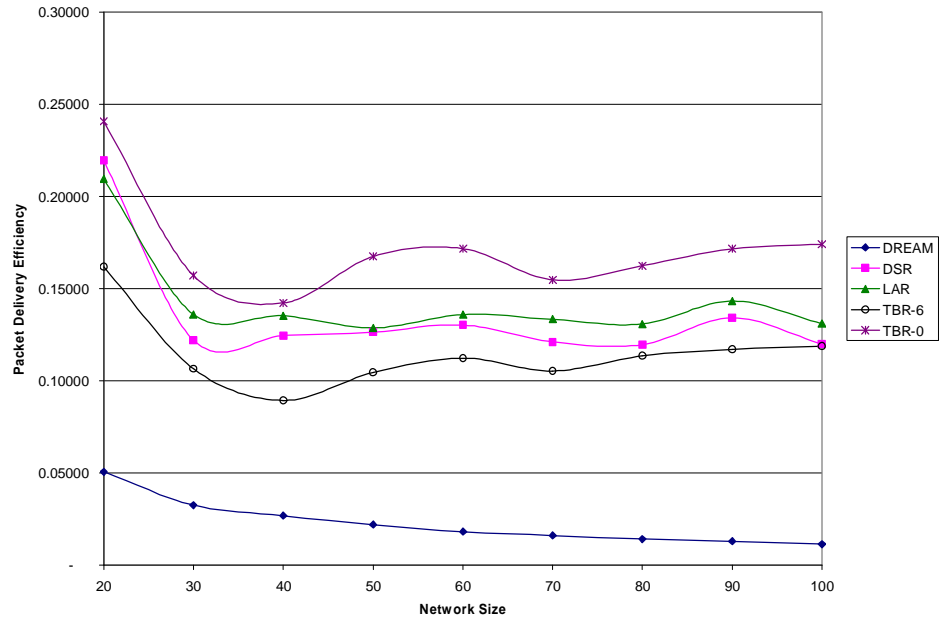


Figure 74: Packet Delivery Efficiency (Flat, Medium Mobility, 1 pkt/sec)

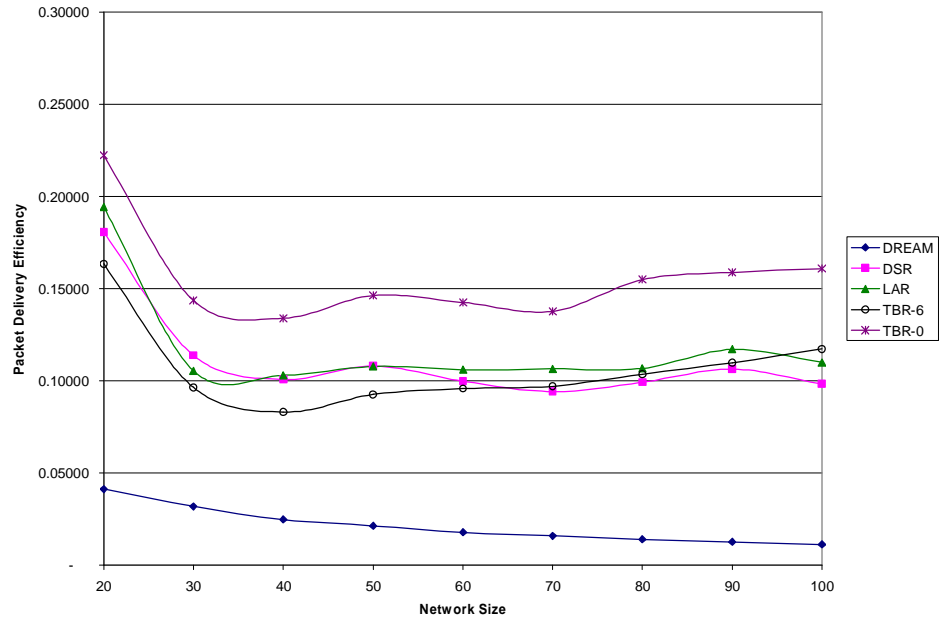


Figure 75: Packet Delivery Efficiency (Flat, High Mobility, 1 pkt/sec)

9.3.2 Flat Terrain, 2 Packets/Second

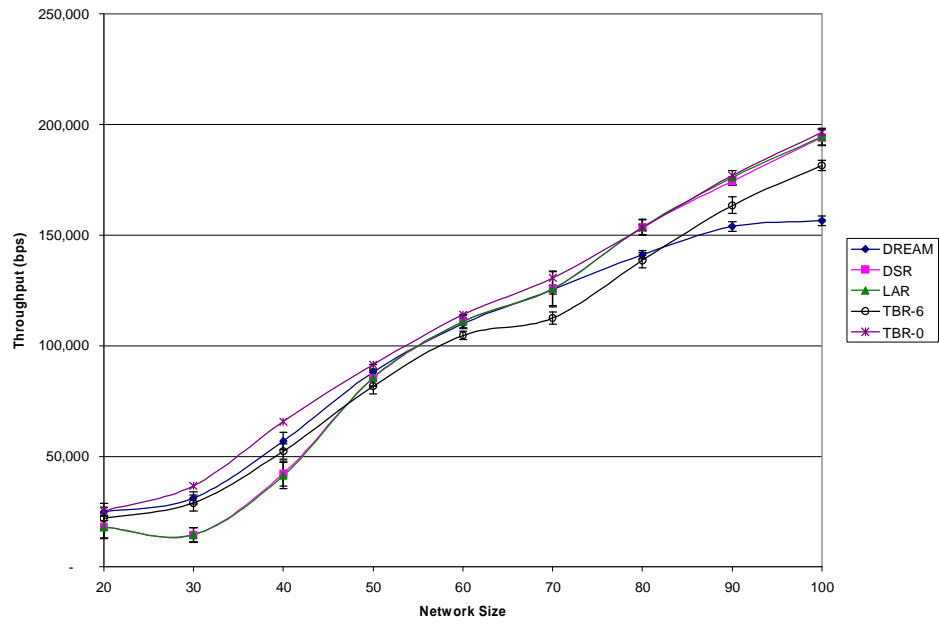


Figure 76: Throughput (Flat, Low Mobility, 2 pkts/sec)

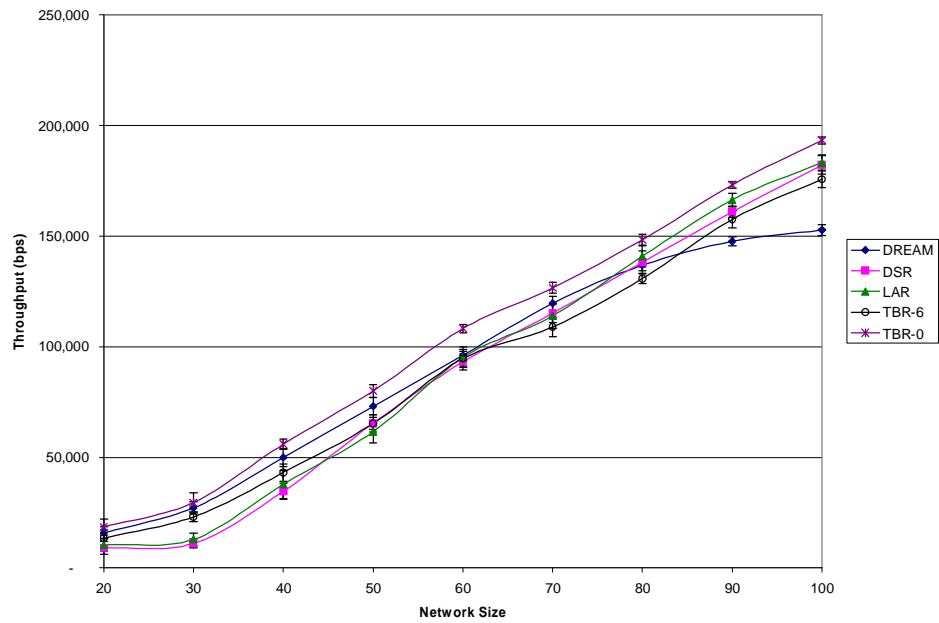


Figure 77: Throughput (Flat, Medium Mobility, 2 pkts/sec)

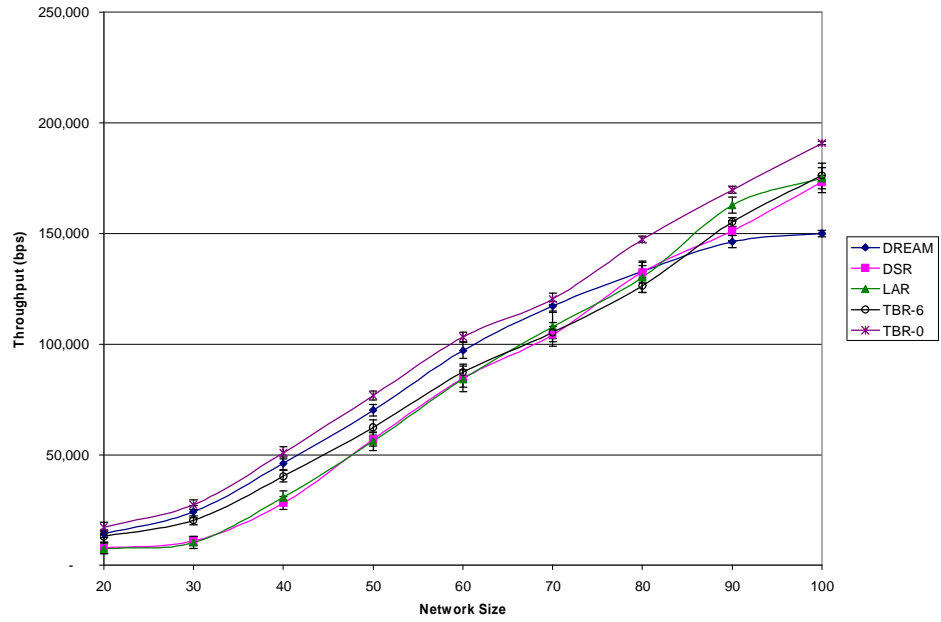


Figure 78: Throughput (Flat, High Mobility, 2 pkts/sec)

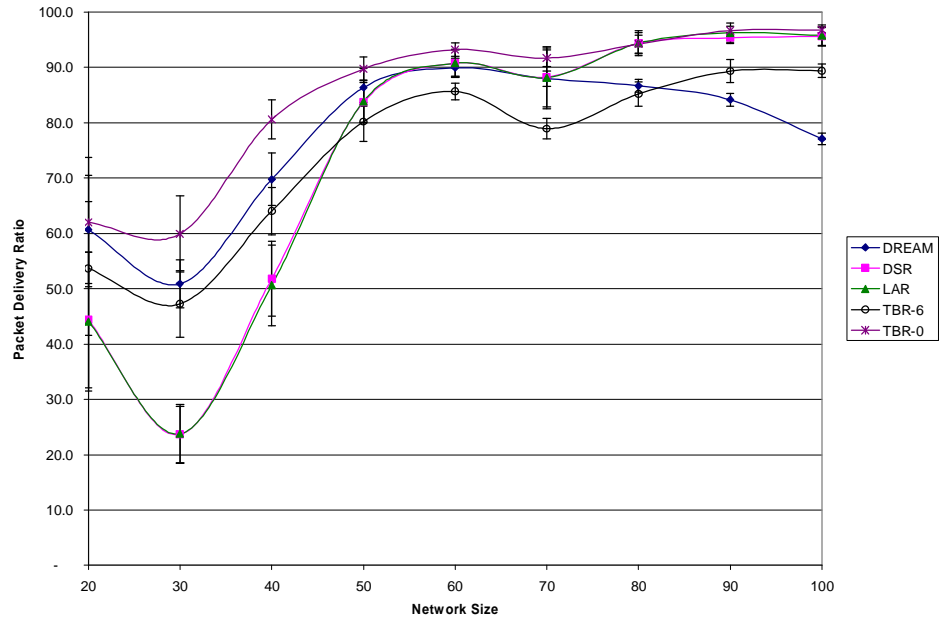


Figure 79: Packet Delivery Ratio (Flat, Low Mobility, 2 pkts/sec)

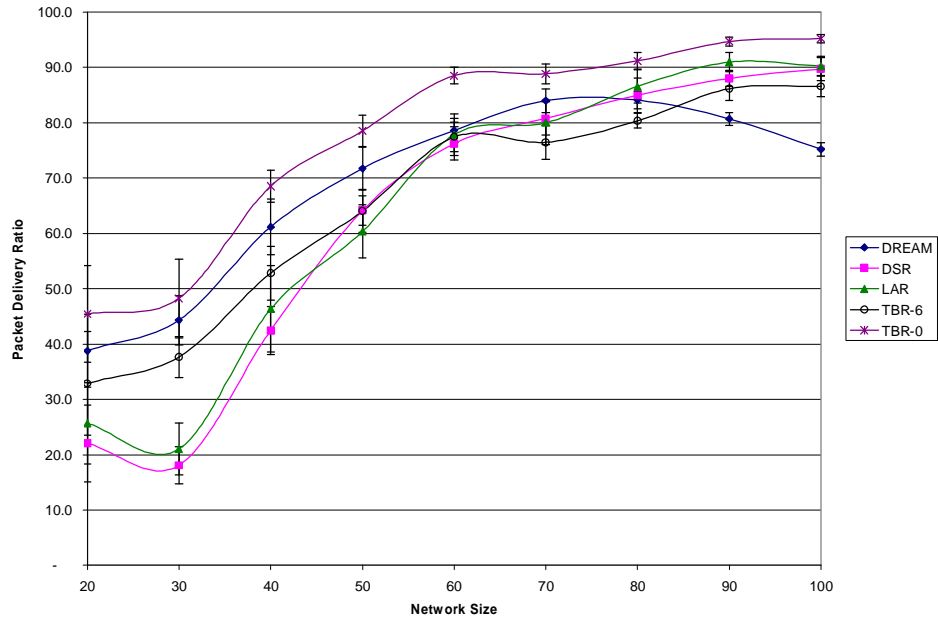


Figure 80: Packet Delivery Ratio (Flat, Medium Mobility, 2 pkts/sec)

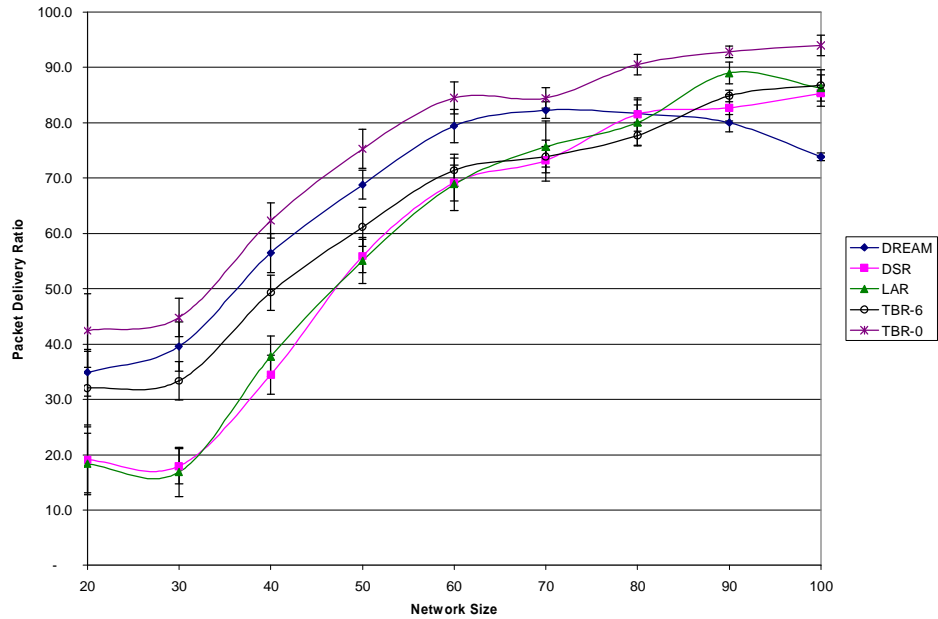


Figure 81: Packet Delivery Ratio (Flat, High Mobility, 2 pkts/sec)

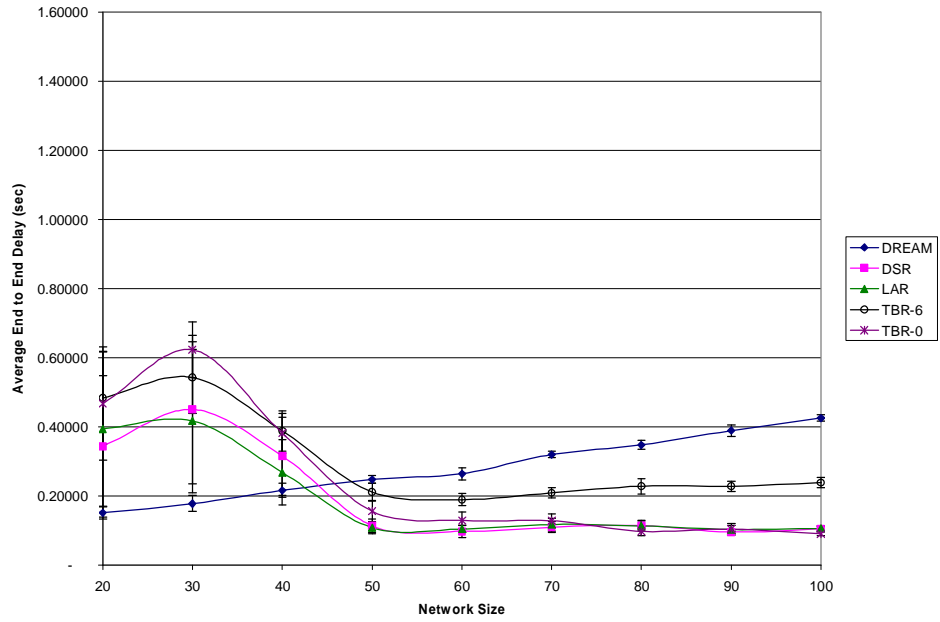


Figure 82: Average End to End Delay (Flat, Low Mobility, 2 pkts/sec)

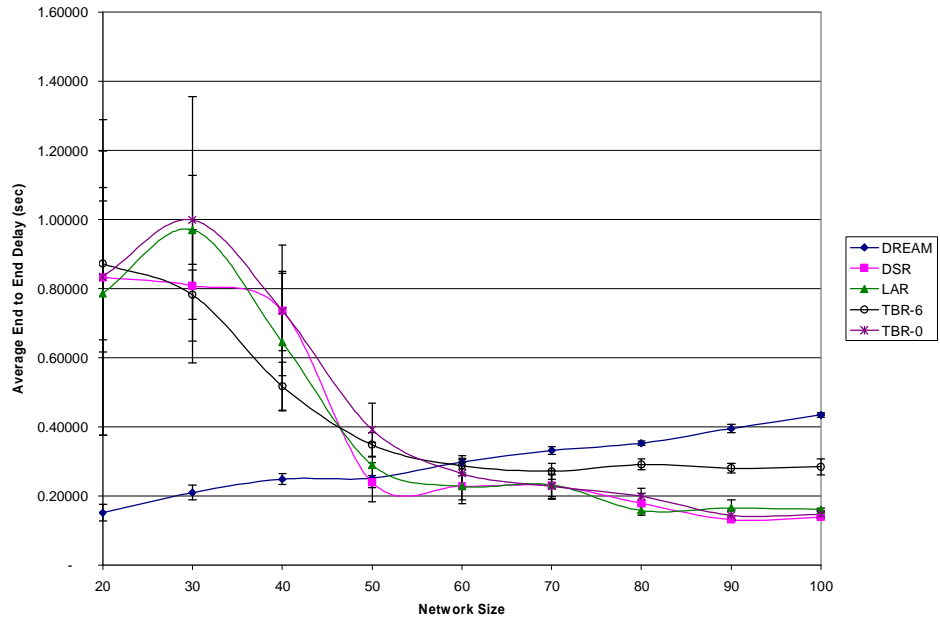


Figure 83: Average End to End Delay (Flat, Medium Mobility, 2 pkts/sec)

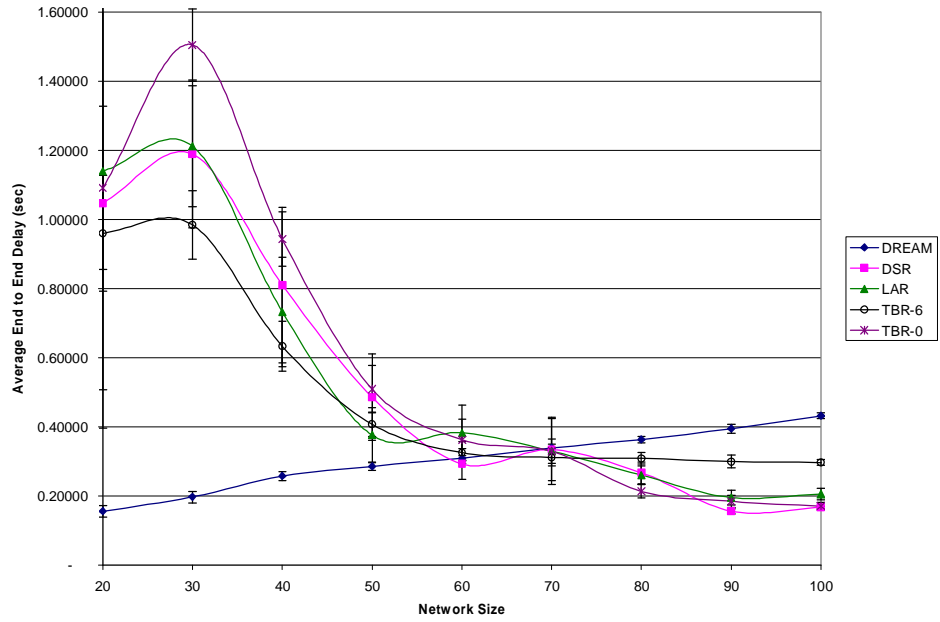


Figure 84: Average End to End Delay (Flat, High Mobility, 2 pkts/sec)

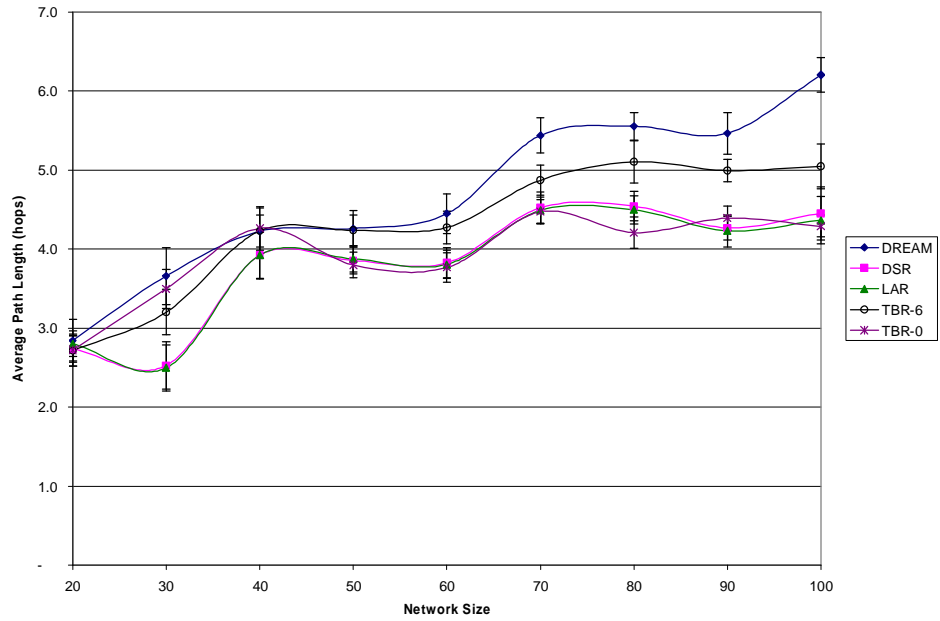


Figure 85: Average Path Length (Flat, Low Mobility, 2 pkts/sec)

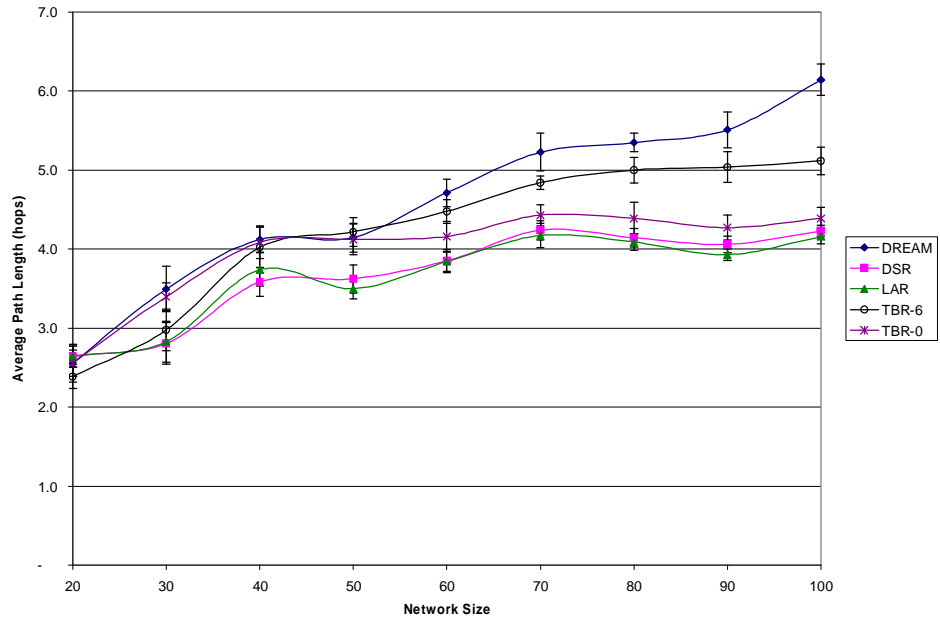


Figure 86: Average Path Length (Flat, Medium Mobility, 2 pkts/sec)

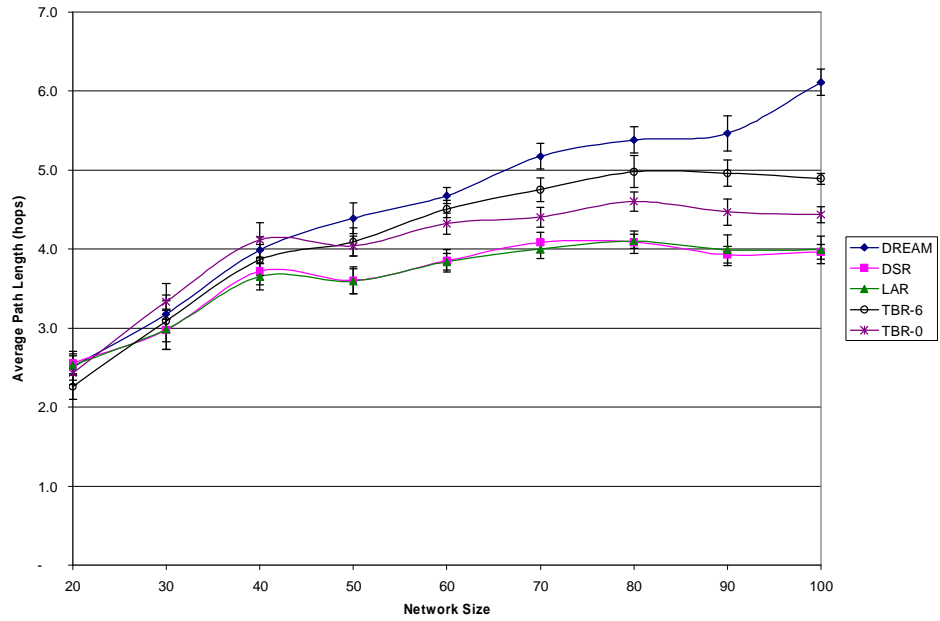


Figure 87: Average Path Length (Flat, High Mobility, 2 pkts/sec)

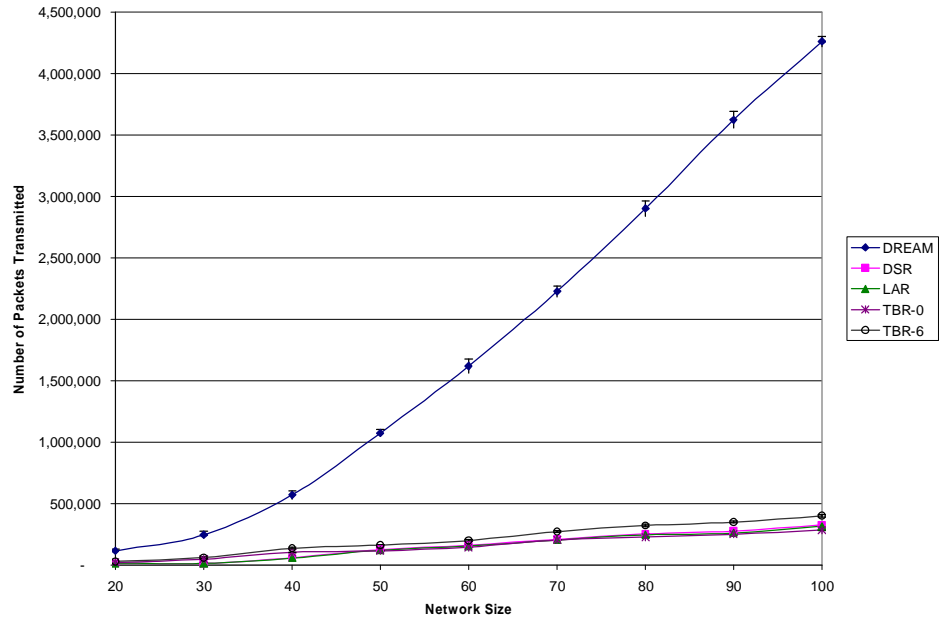


Figure 88: Number of Packets Transmitted (Flat, Low Mobility, 2 pkts/sec)

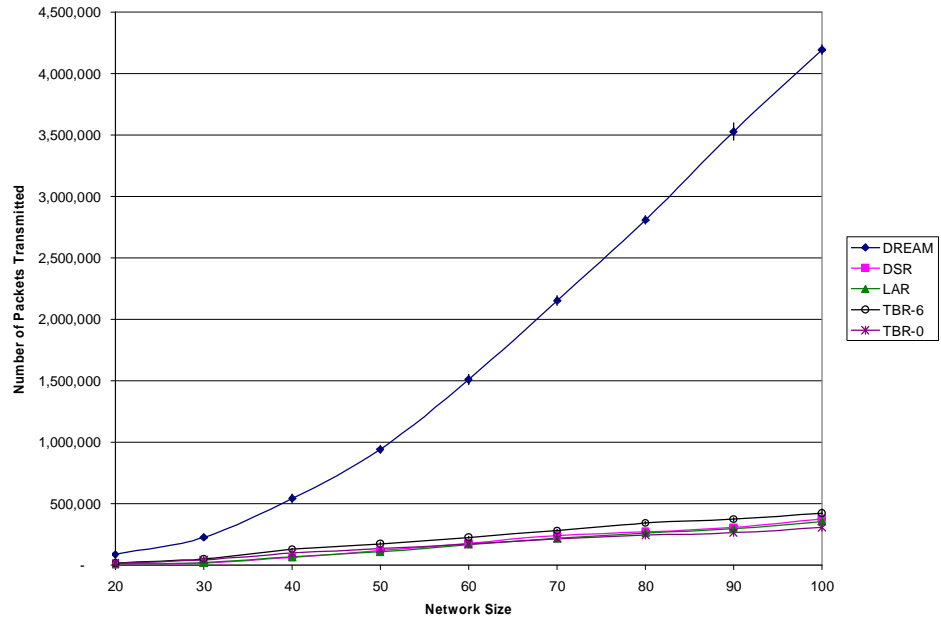


Figure 89: Number of Packets Transmitted (Flat, Medium Mobility, 2 pkts/sec)

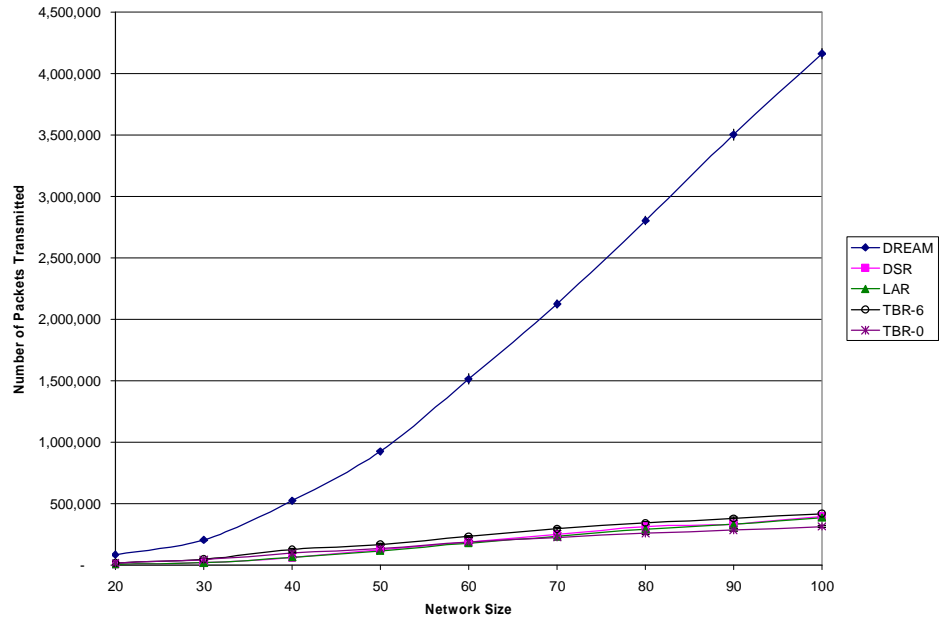


Figure 90: Number of Packets Transmitted (Flat, High Mobility, 2 pkts/sec)

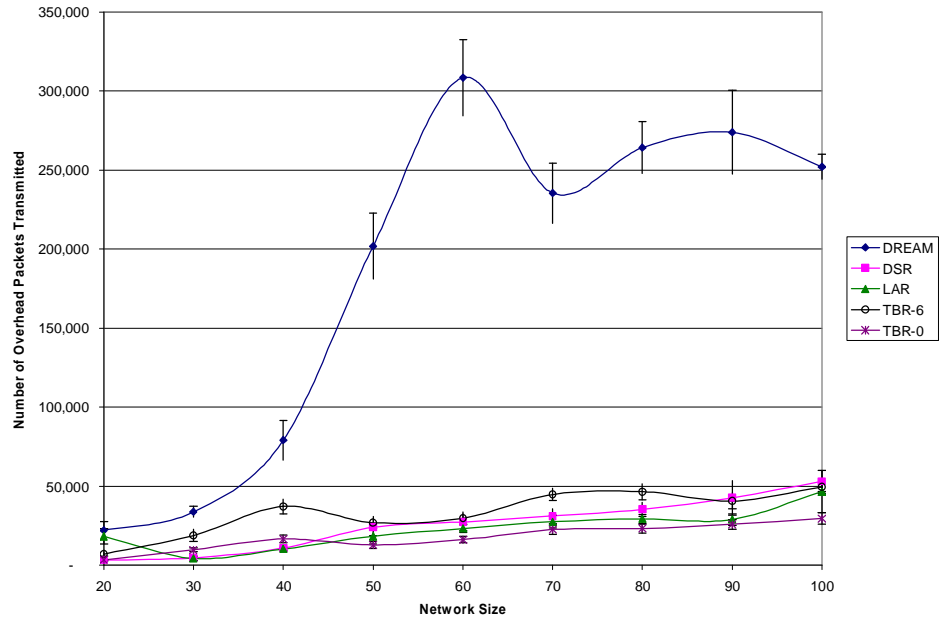


Figure 91: Number of Overhead Packets (Flat, Low Mobility, 2 pkts/sec)

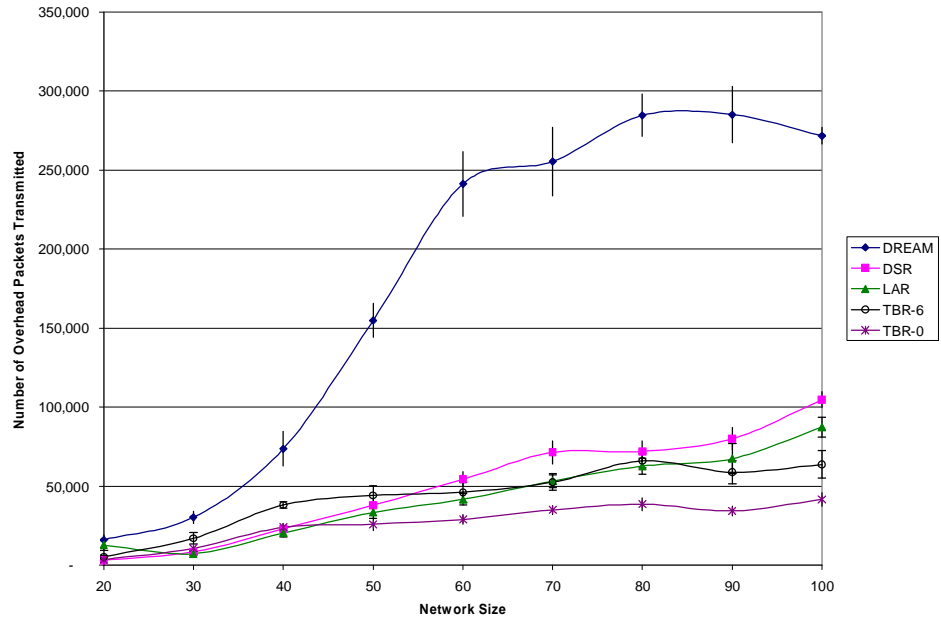


Figure 92: Number of Overhead Packets (Flat, Medium Mobility, 2 pkts/sec)

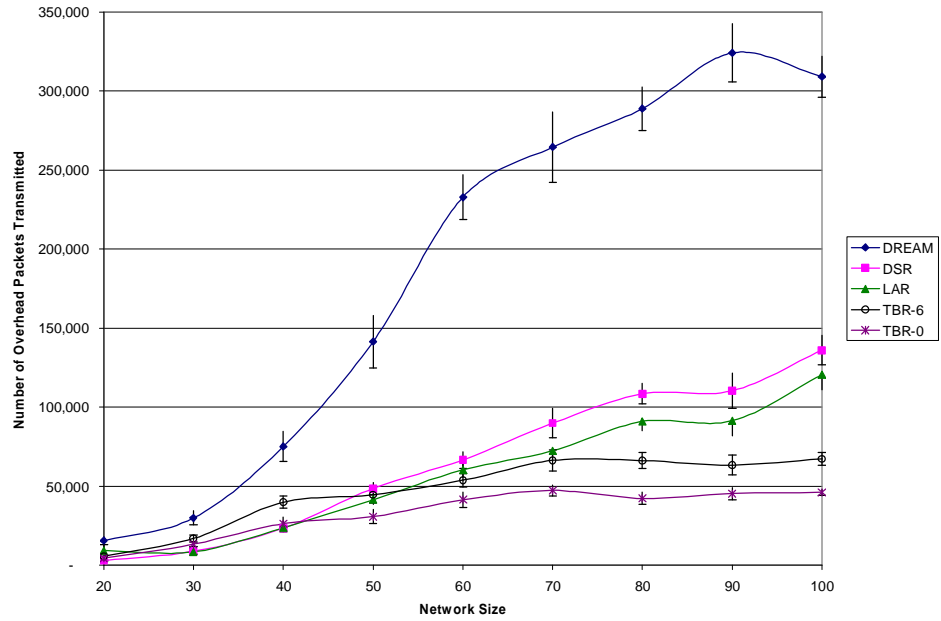


Figure 93: Number of Overhead Packets (Flat, High Mobility, 2 pkts/sec)

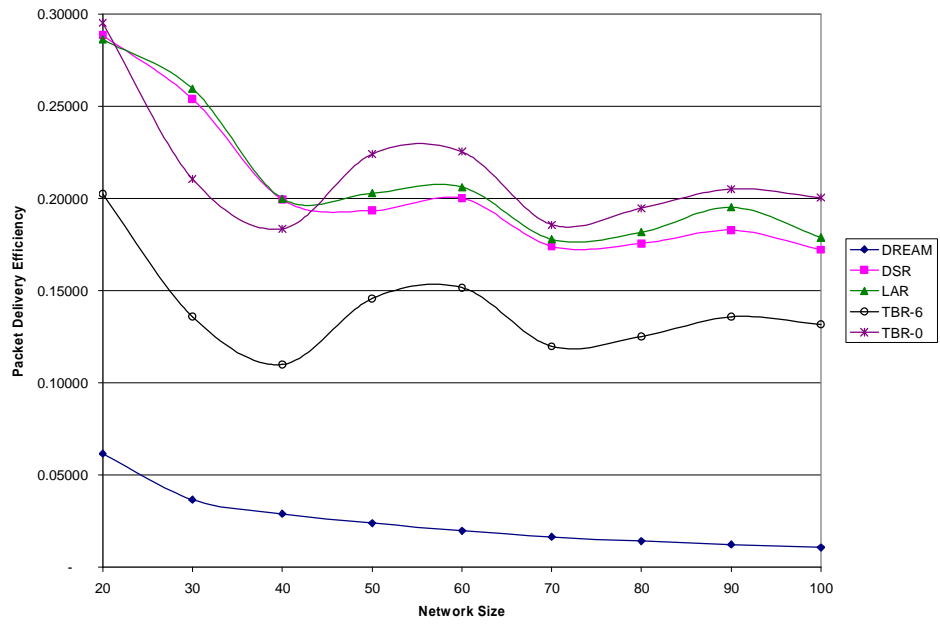


Figure 94: Packet Delivery Efficiency (Flat, Low Mobility, 2 pkts/sec)

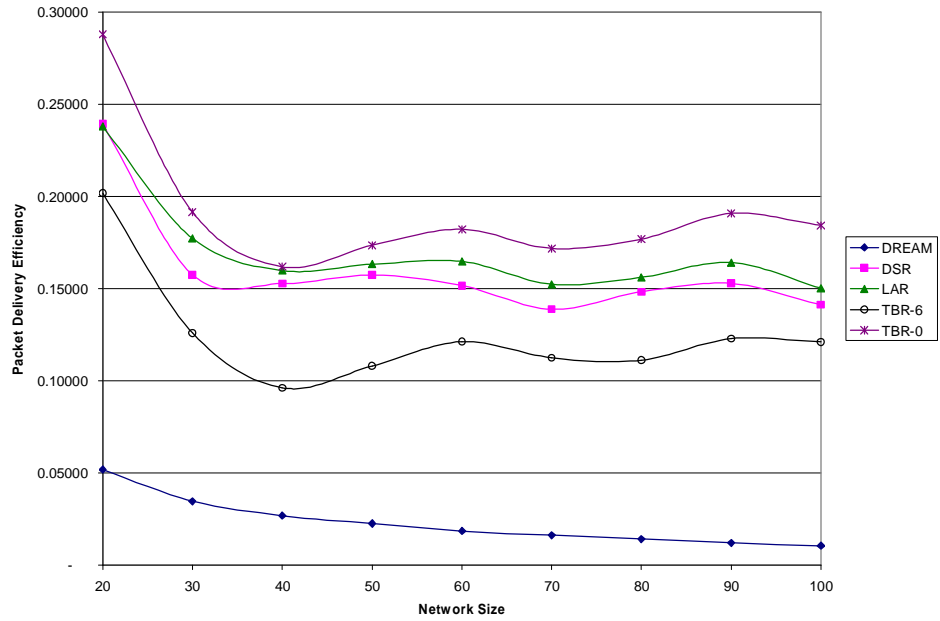


Figure 95: Packet Delivery Efficiency (Flat, Medium Mobility, 2 pkts/sec)

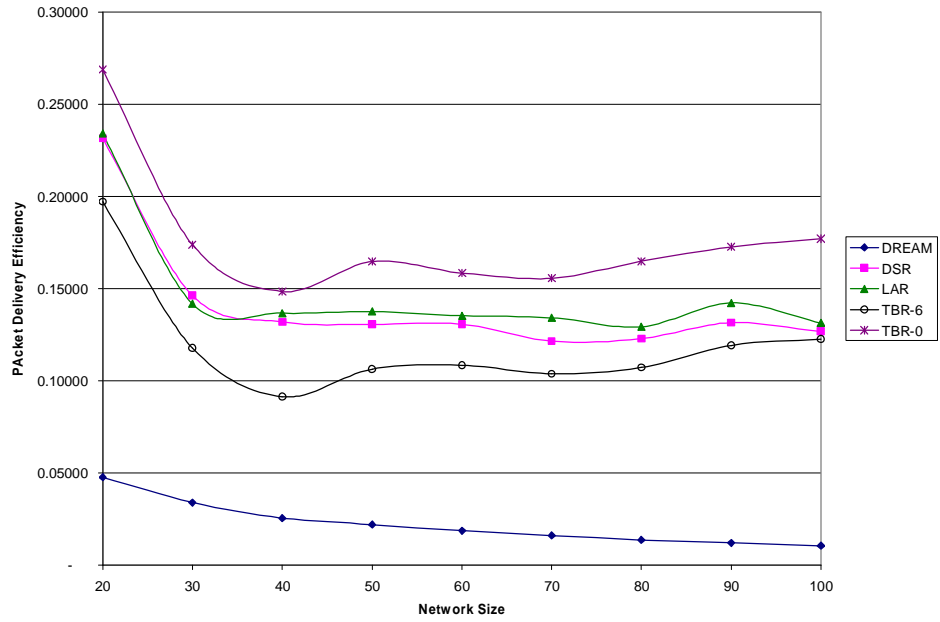


Figure 96: Packet Delivery Efficiency (Flat, High Mobility, 2 pkts/sec)

9.3.3 Flat Terrain, 3 Packets/Second

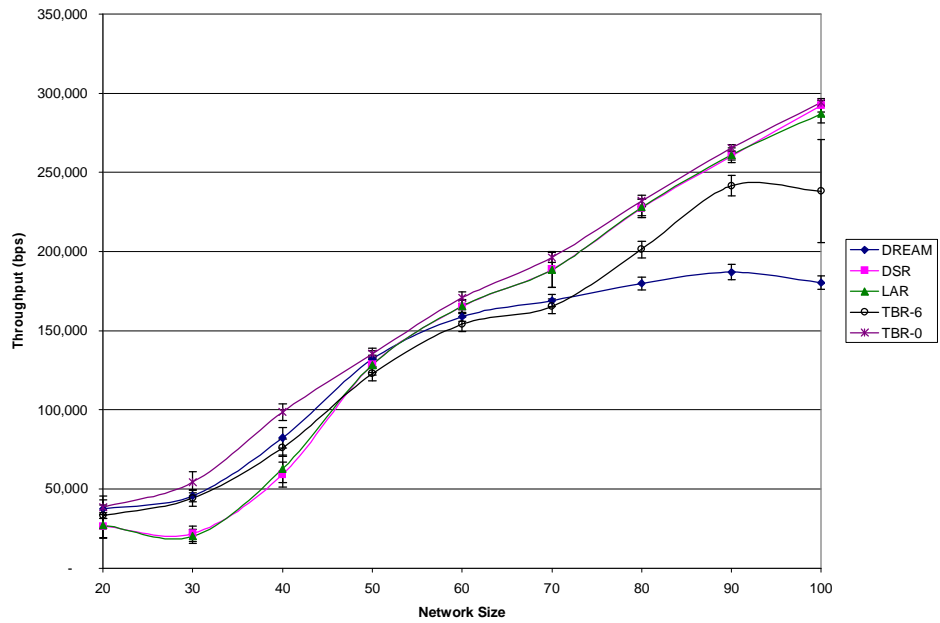


Figure 97: Throughput (Flat, Low Mobility, 3 pkts/sec)

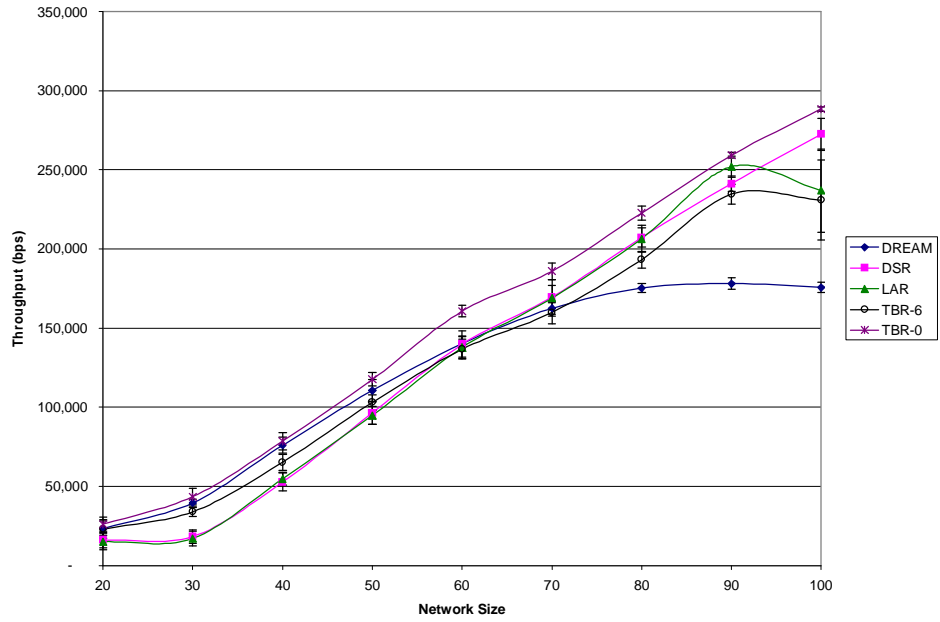


Figure 98: Throughput (Flat, Medium Mobility, 3 pkts/sec)

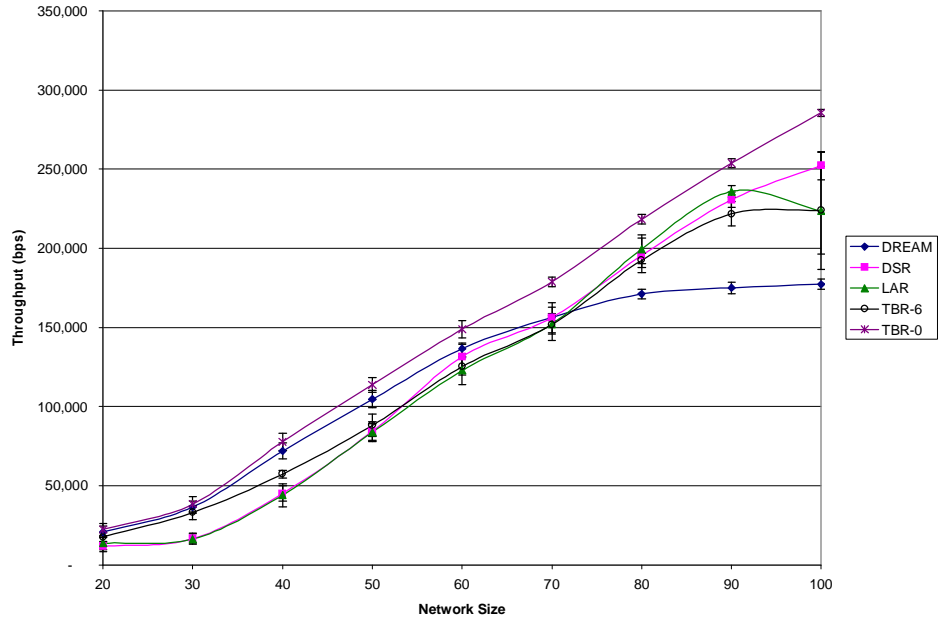


Figure 99: Throughput (Flat, High Mobility, 3 pkts/sec)

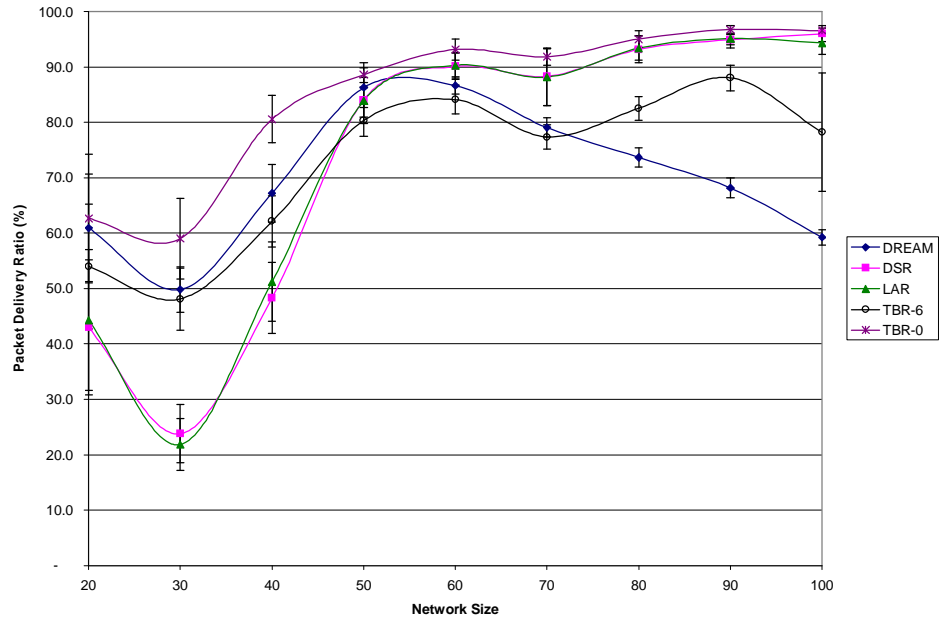


Figure 100: Packet Delivery Ratio (Flat, Low Mobility, 3 pkts/sec)

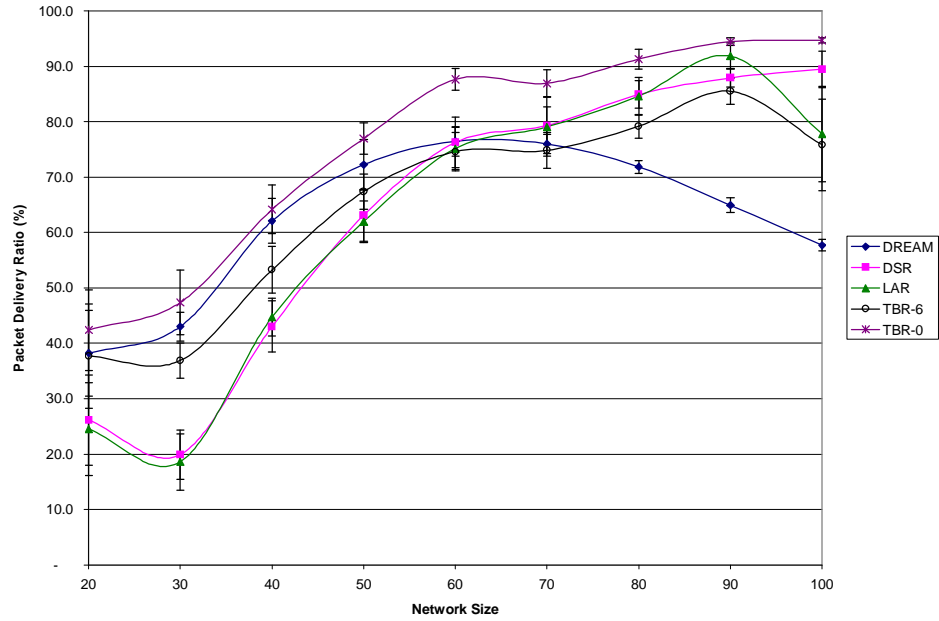


Figure 101: Packet Delivery Ratio (Flat, Medium Mobility, 3 pkts/sec)

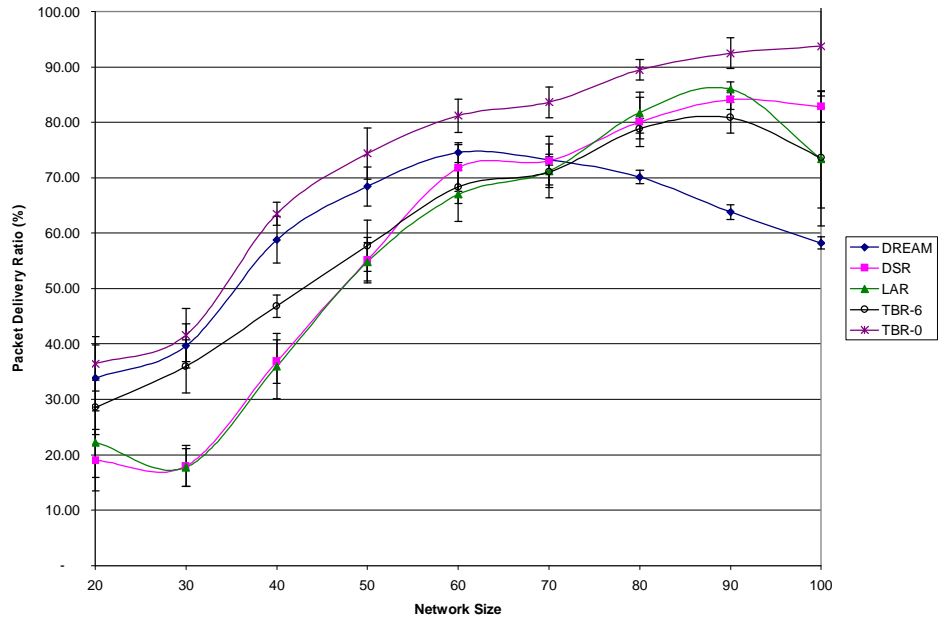


Figure 102: Packet Delivery Ratio (Flat, High Mobility, 3 pkts/sec)

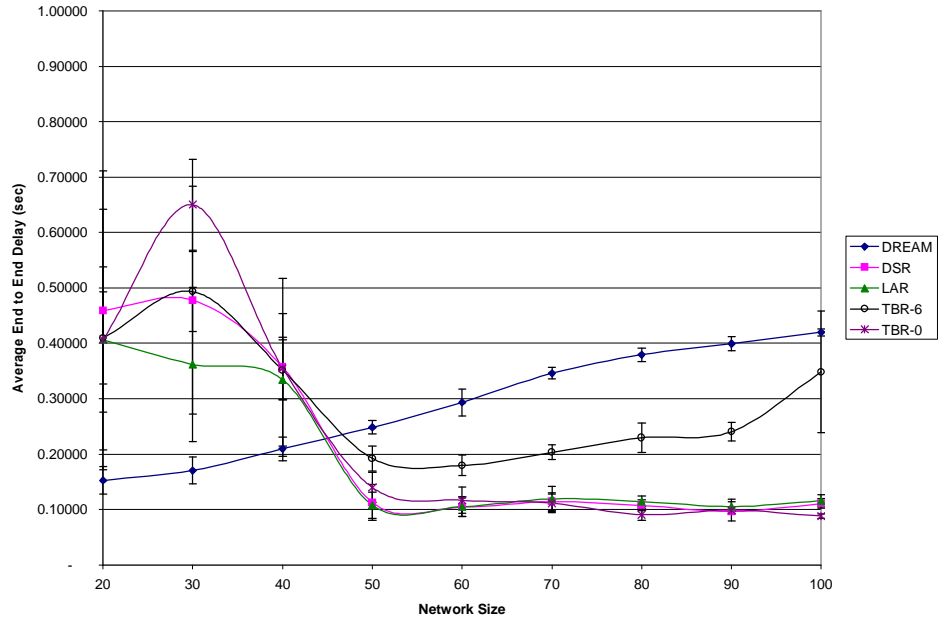


Figure 103: Average End to End Delay (Low Mobility, 3 Packets/Second)

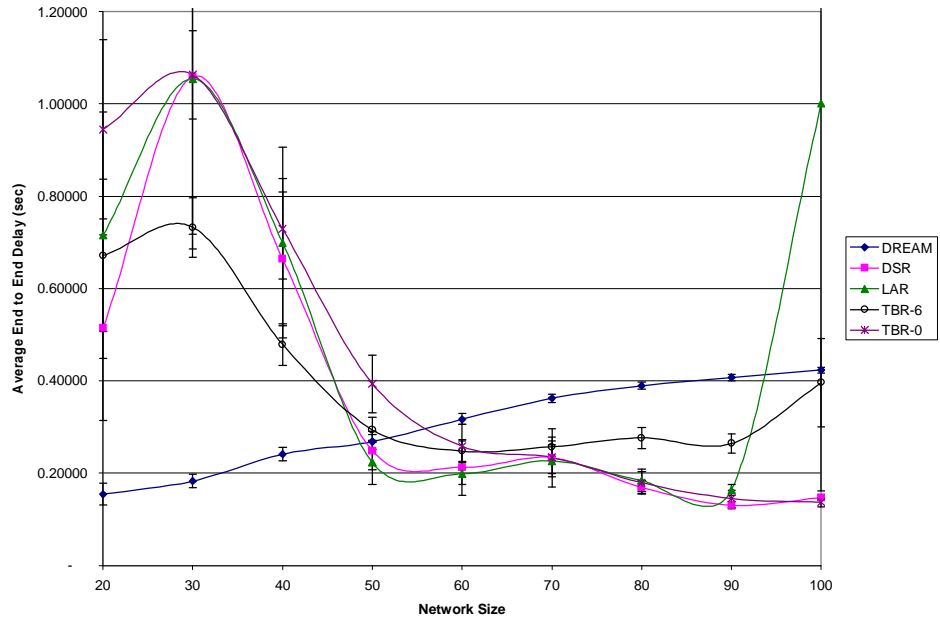


Figure 104: Average End to End Delay (Medium Mobility, 3 Packets/Second)

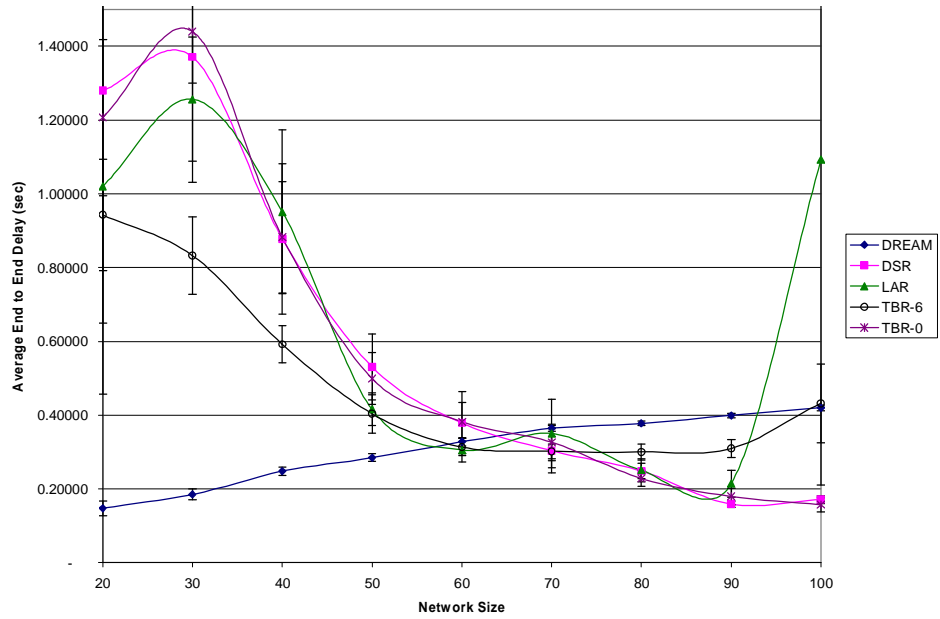


Figure 105: Average End to End Delay (High Mobility, 3 Packets/Second)

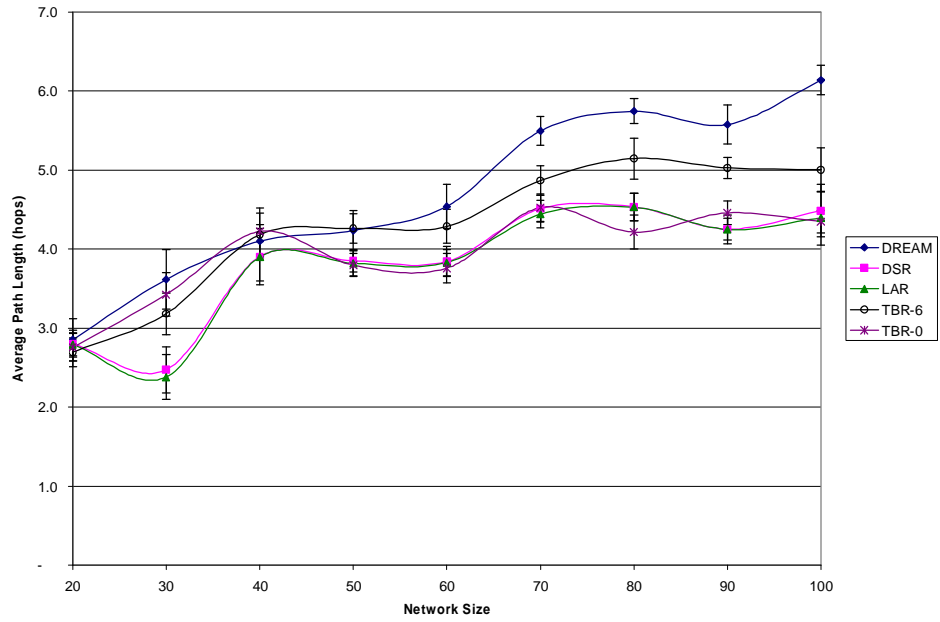


Figure 106: Average Path Length (Flat, Low Mobility, 3 pkts/sec)

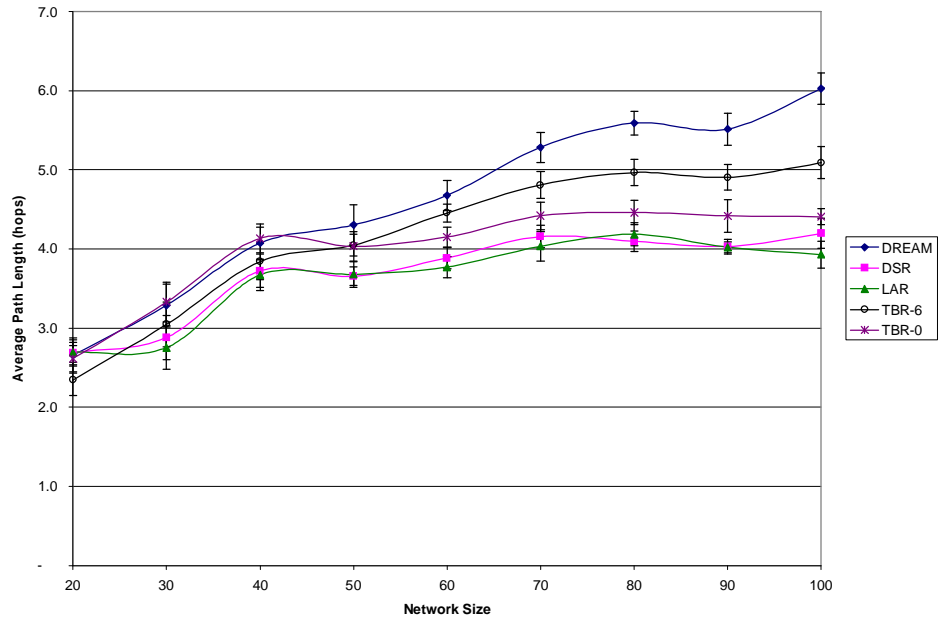


Figure 107: Average Path Length (Flat, Medium Mobility, 3 pkts/sec)

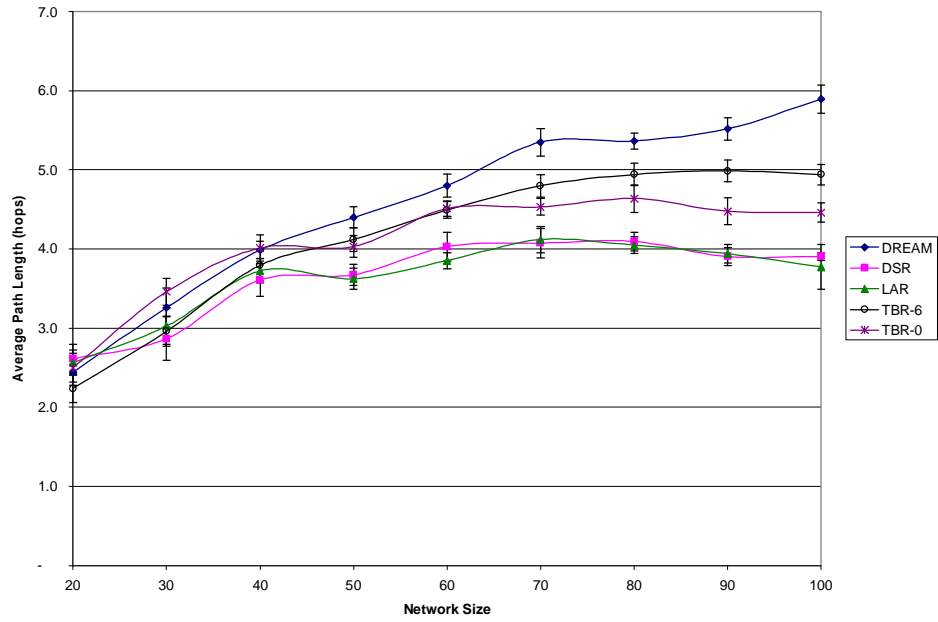


Figure 108: Average Path Length (Flat, High Mobility, 3 pkts/sec)

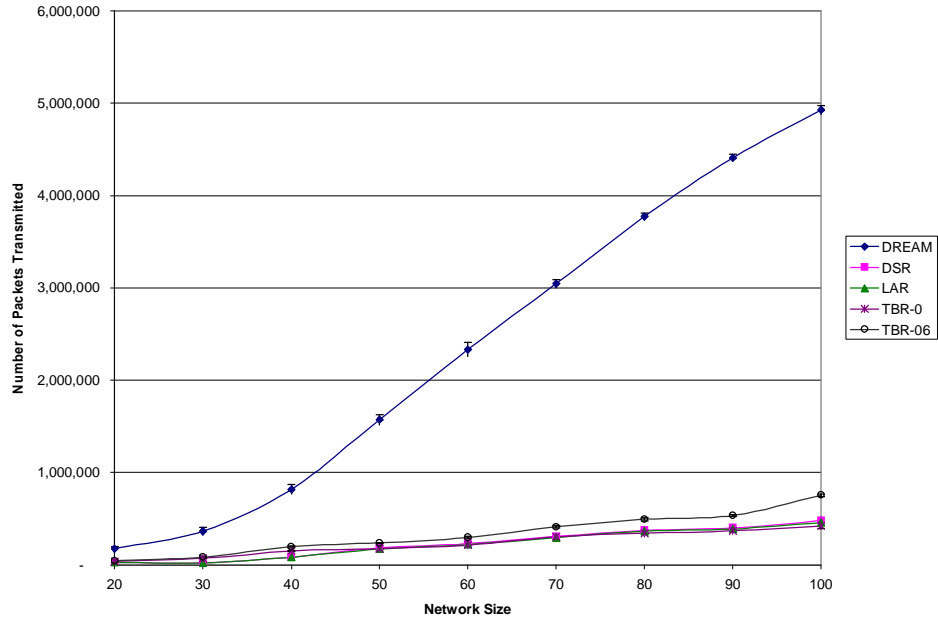


Figure 109: Number of Packets Transmitted (Flat, Low Mobility, 3 pkts/sec)

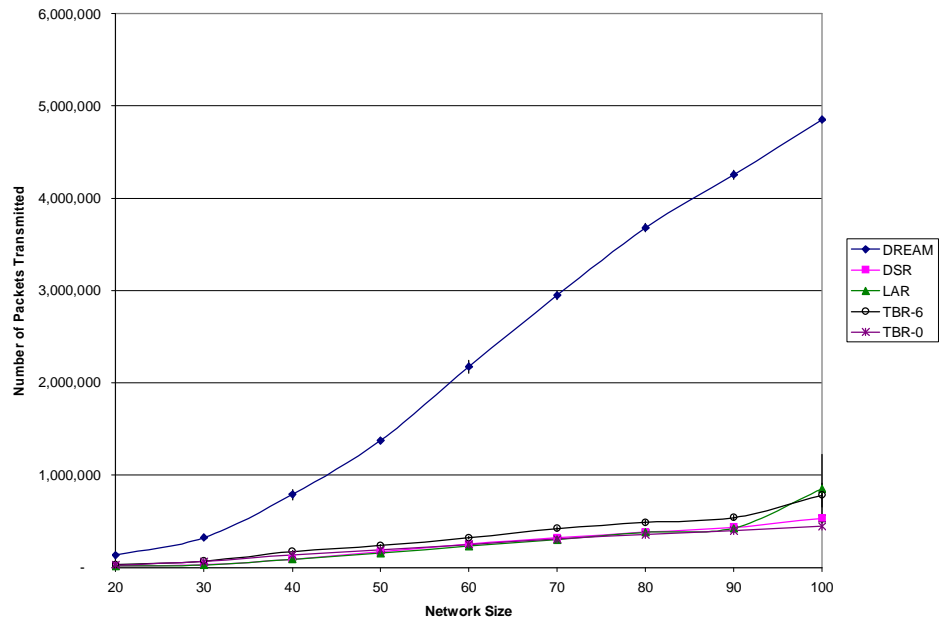


Figure 110: Number of Packets Transmitted (Flat, Medium Mobility, 3 pkts/sec)

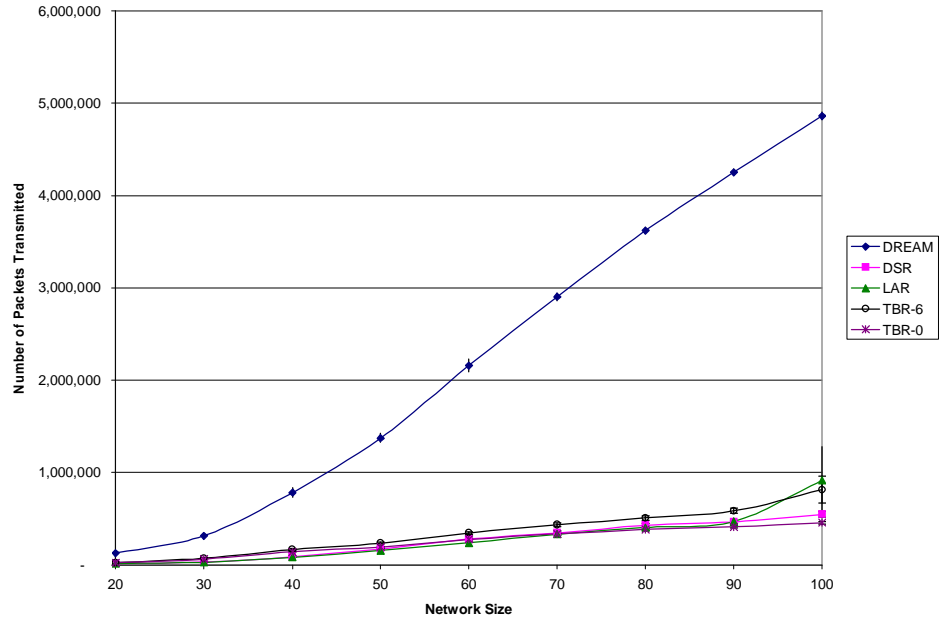


Figure 111: Number of Packets Transmitted (Flat, High Mobility, 3 pkts/sec)

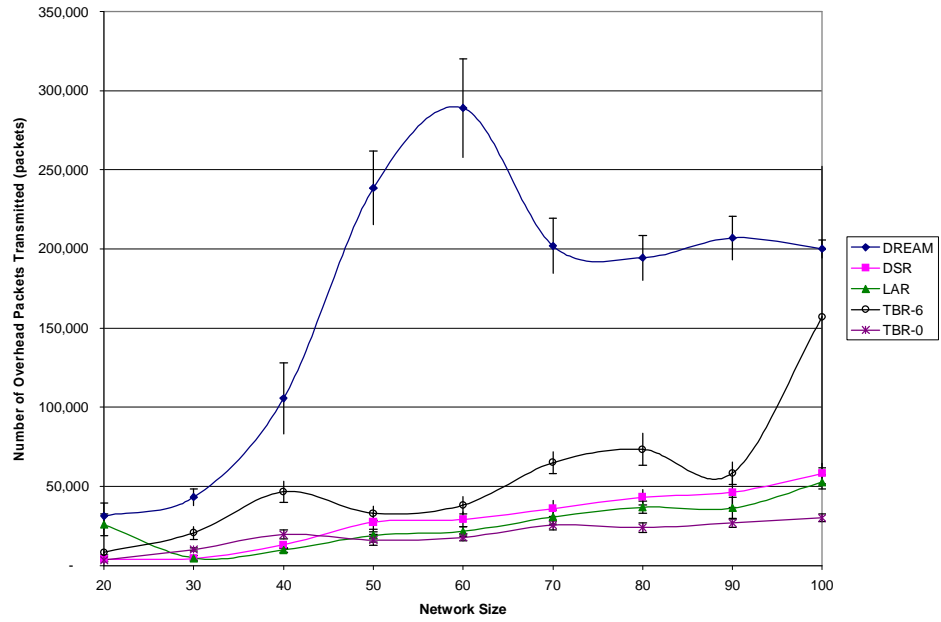


Figure 112: Number of Overhead Packets (Flat, Low Mobility, 3 pkts/sec)

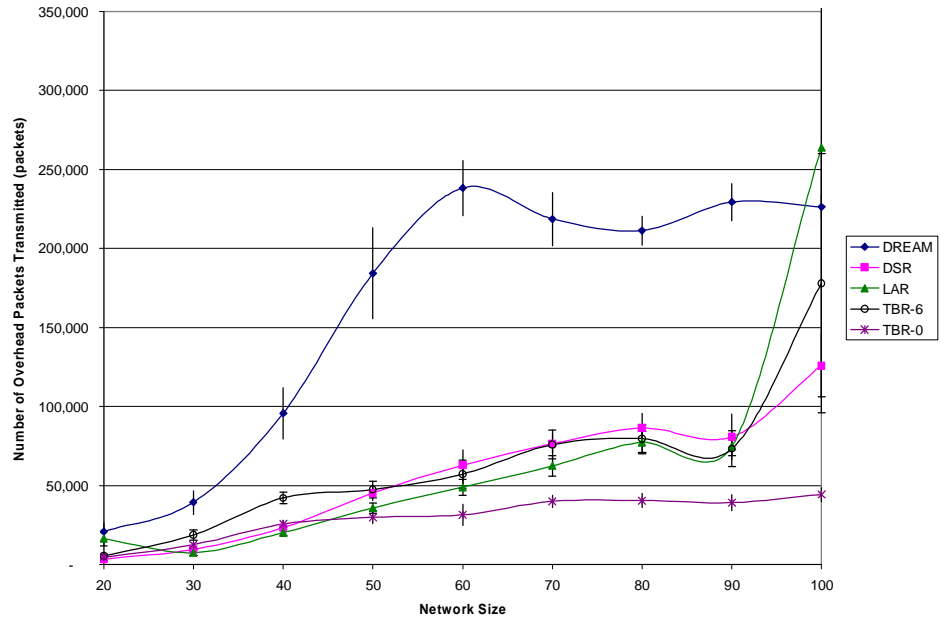


Figure 113: Number of Overhead Packets (Flat, Medium Mobility, 3 pkts/sec)

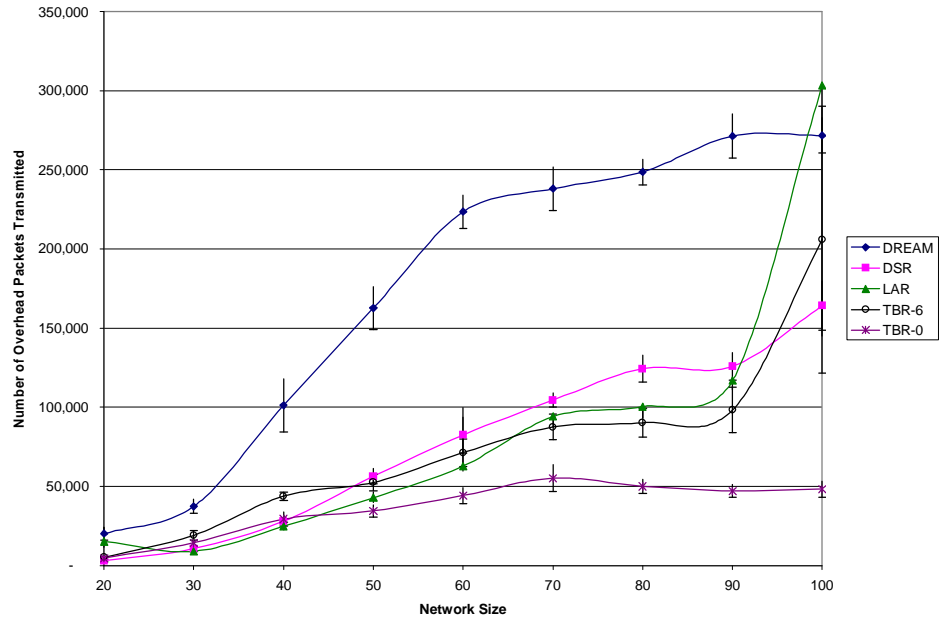


Figure 114: Number of Overhead Packets (Flat, High Mobility, 3 pkts/sec)

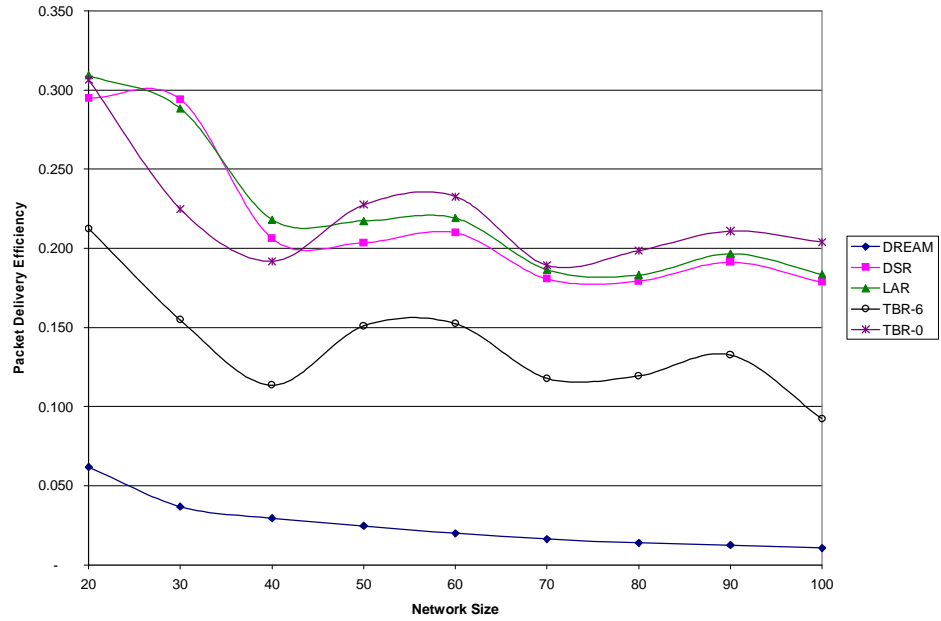


Figure 115: Packet Delivery Efficiency (Flat, Low Mobility, 3 pkts/sec)

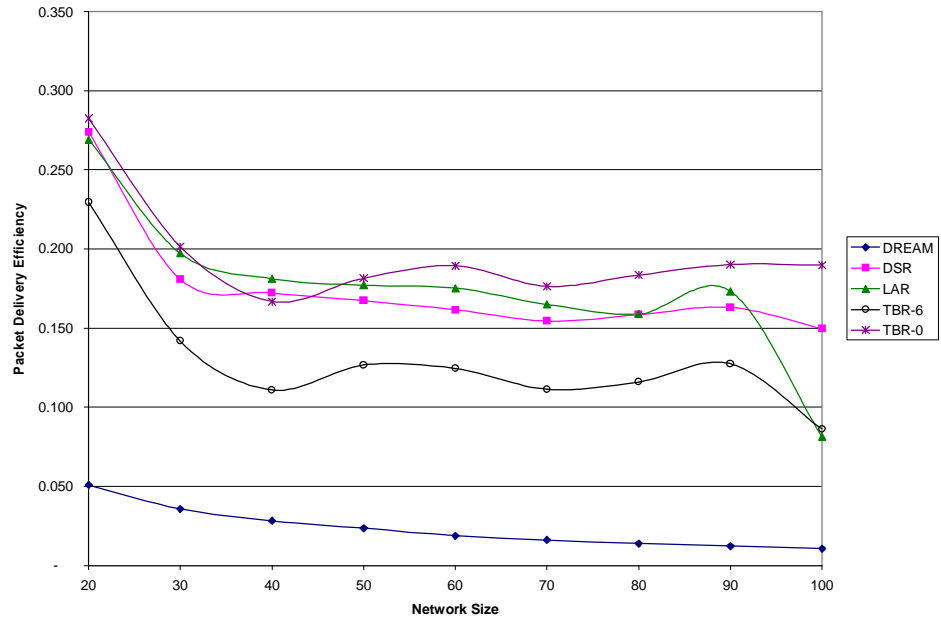


Figure 116: Packet Deliver Efficiency (Flat, Medium Mobility, 3 pkts/sec)

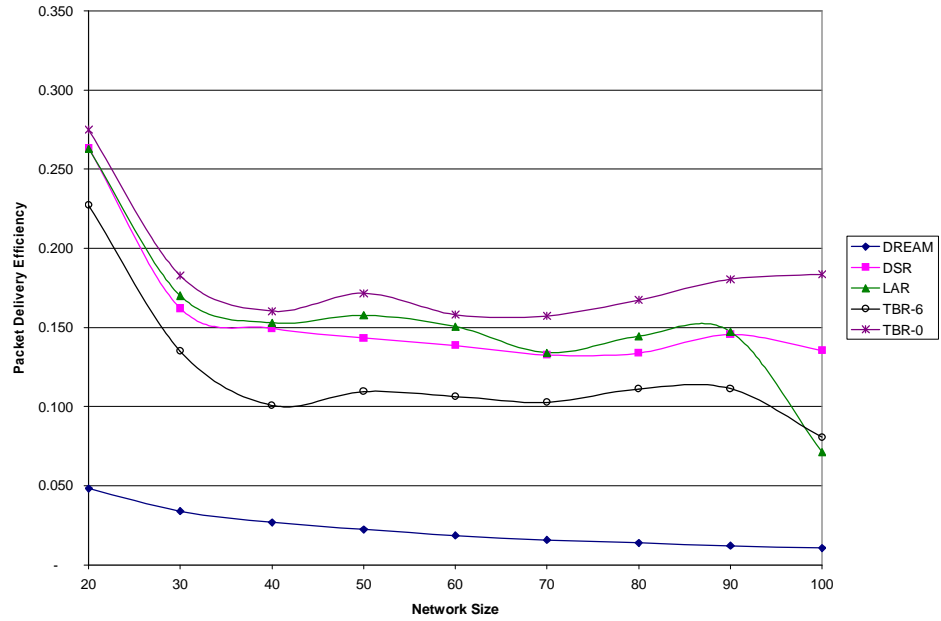


Figure 117: Packet Delivery Efficiency (Flat, High Mobility, 3 pkts/sec)

9.3.4 Flat Terrain, 4 Packets/Second

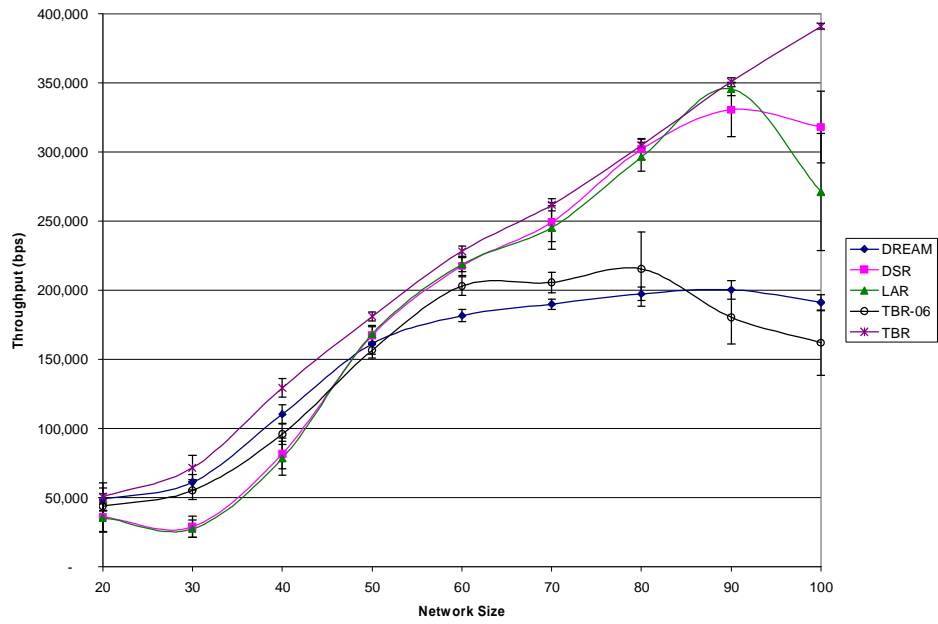


Figure 118: Throughput (Flat, Low Mobility, 4 pkts/sec)

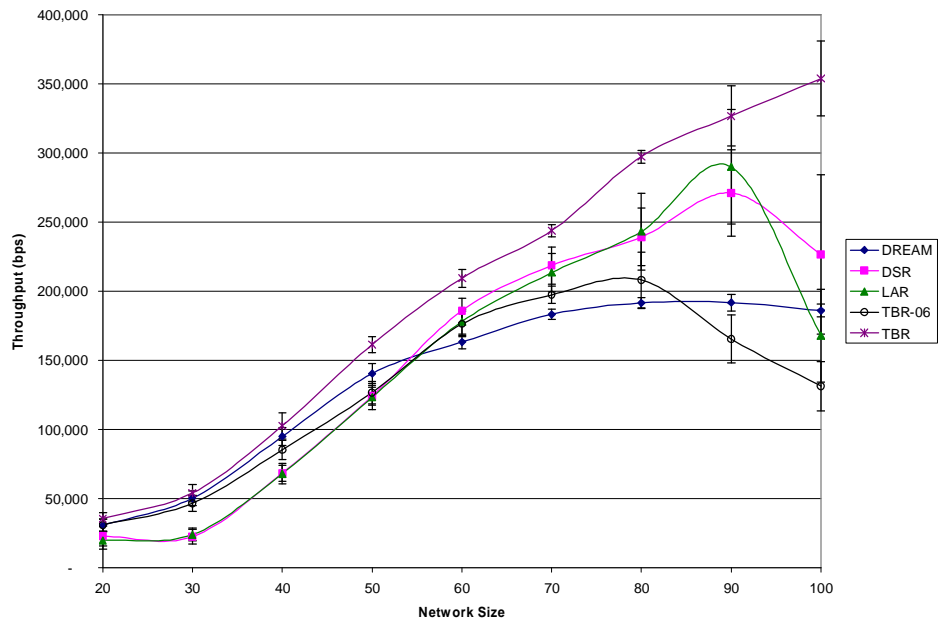


Figure 119: Throughput (Flat, Medium Mobility, 4 pkts/sec)

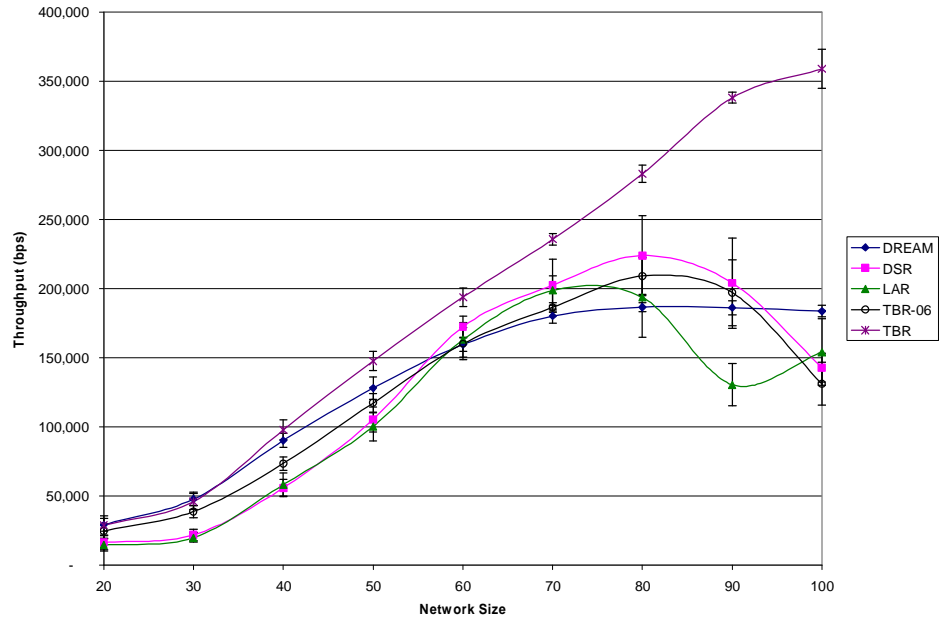


Figure 120: Throughput (Flat, High Mobility, 4 pkts/sec)

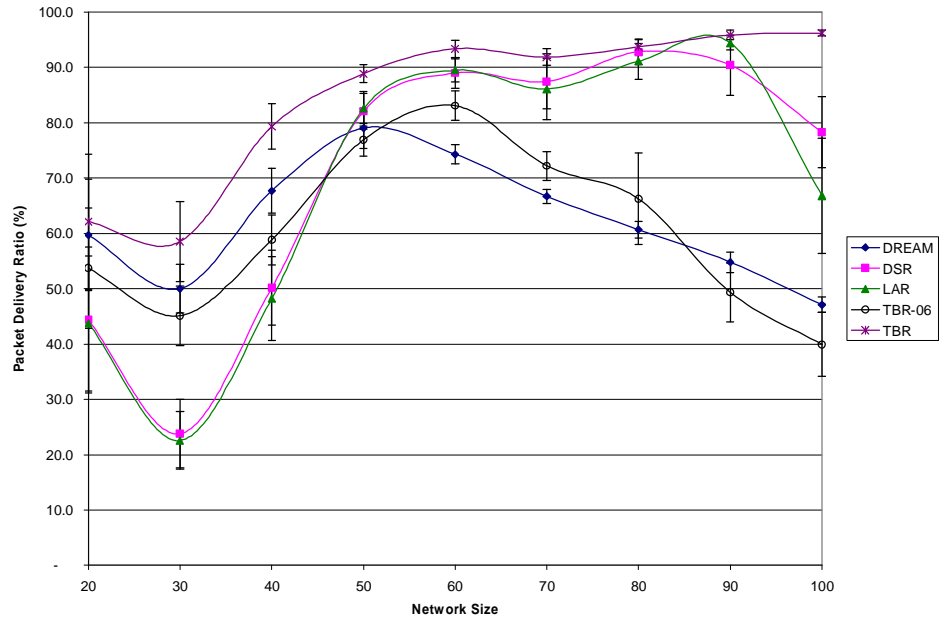


Figure 121: Packet Delivery Ratio (Flat, Low Mobility, 4 pkts/sec)

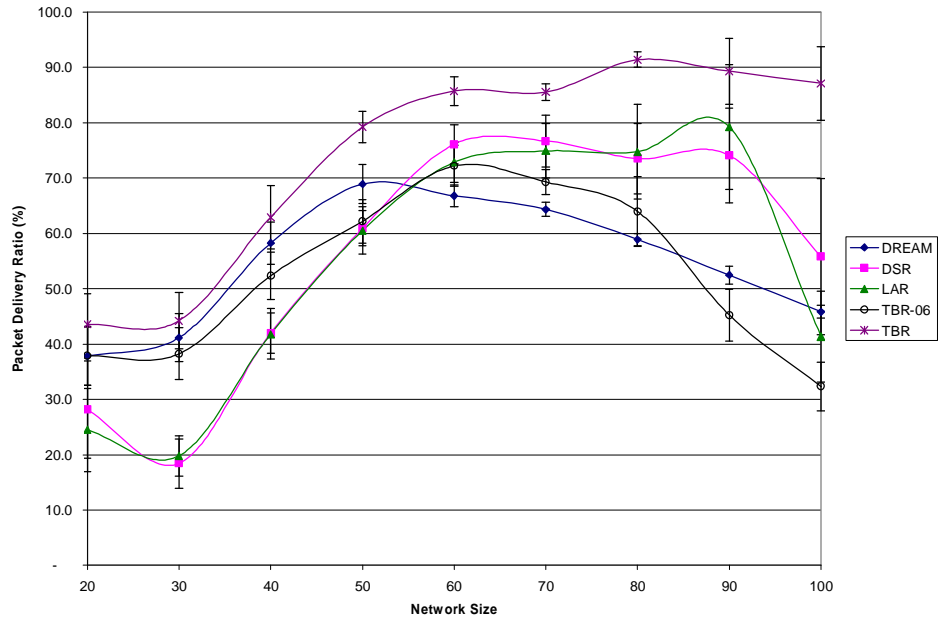


Figure 122: Packet Delivery Ratio (Flat, Medium Mobility, 4 pkts/sec)

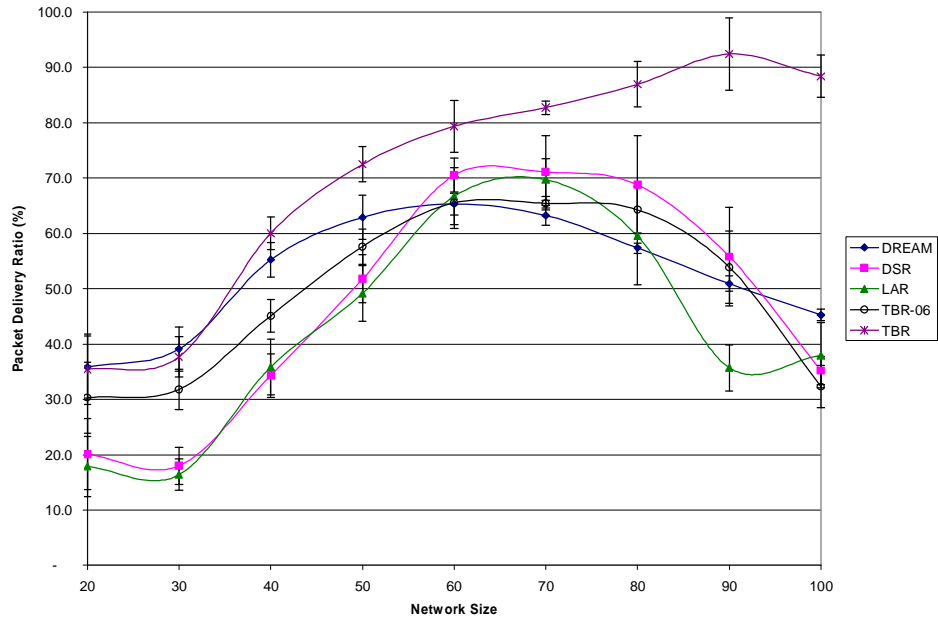


Figure 123: Packet Delivery Ratio (Flat, High Mobility, 4 pkts/sec)

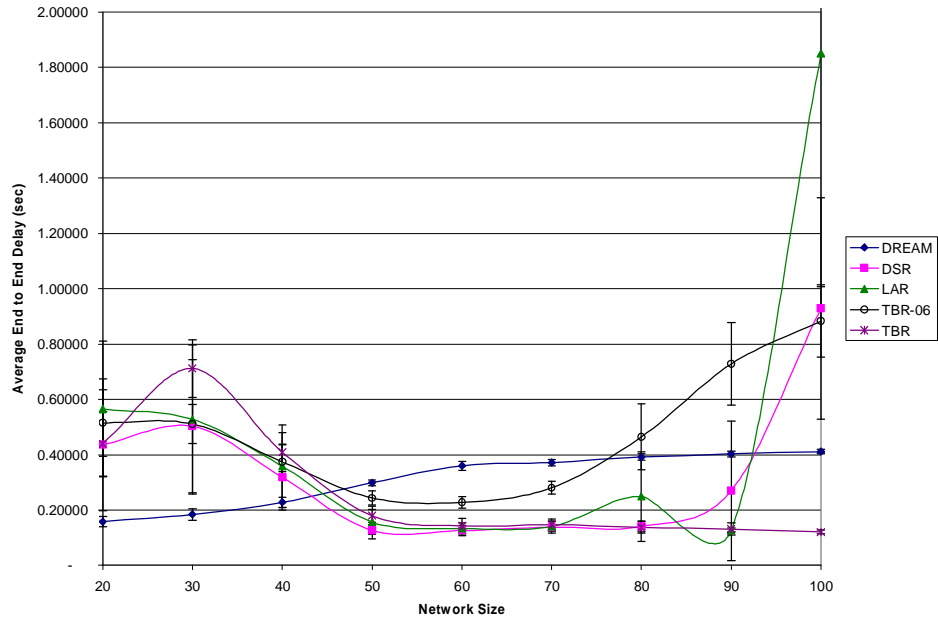


Figure 124: Average End to End Delay (Flat, Low Mobility, 4 pkts/sec)

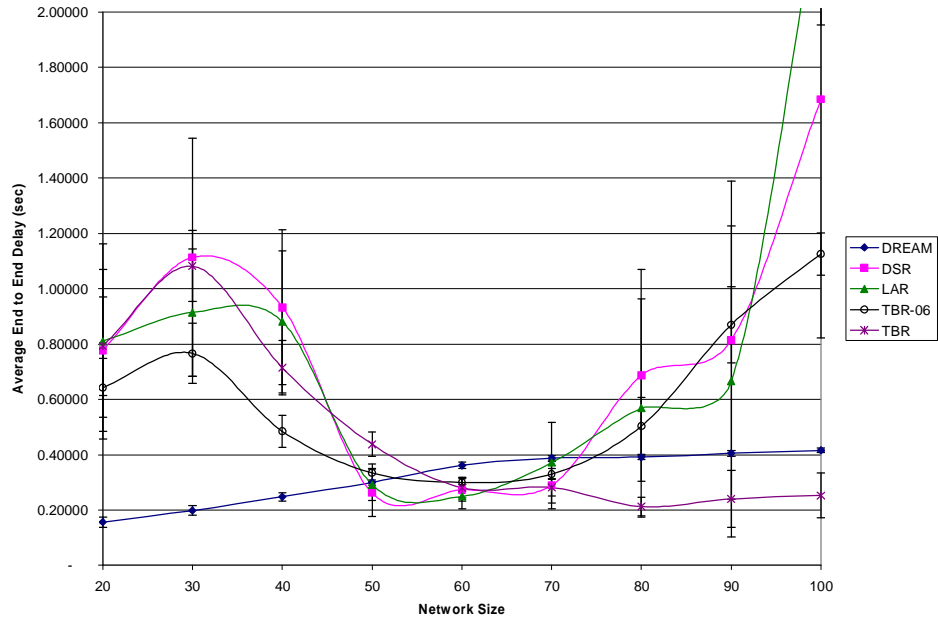


Figure 125: Average End to End Delay (Flat, Medium Mobility, 4 pkts/sec)

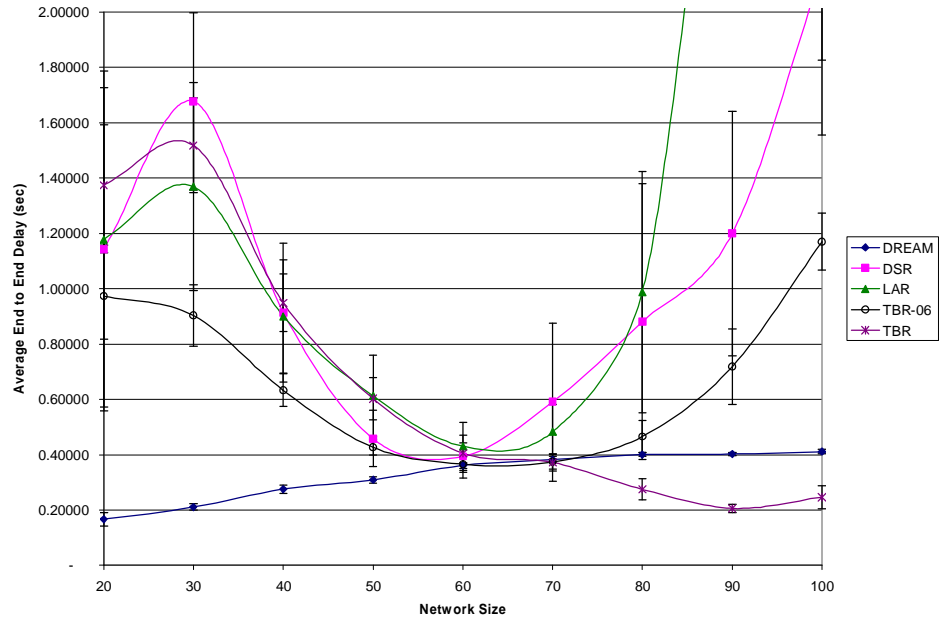


Figure 126: Average End to End Delay (Flat, High Mobility, 4 pkts/sec)

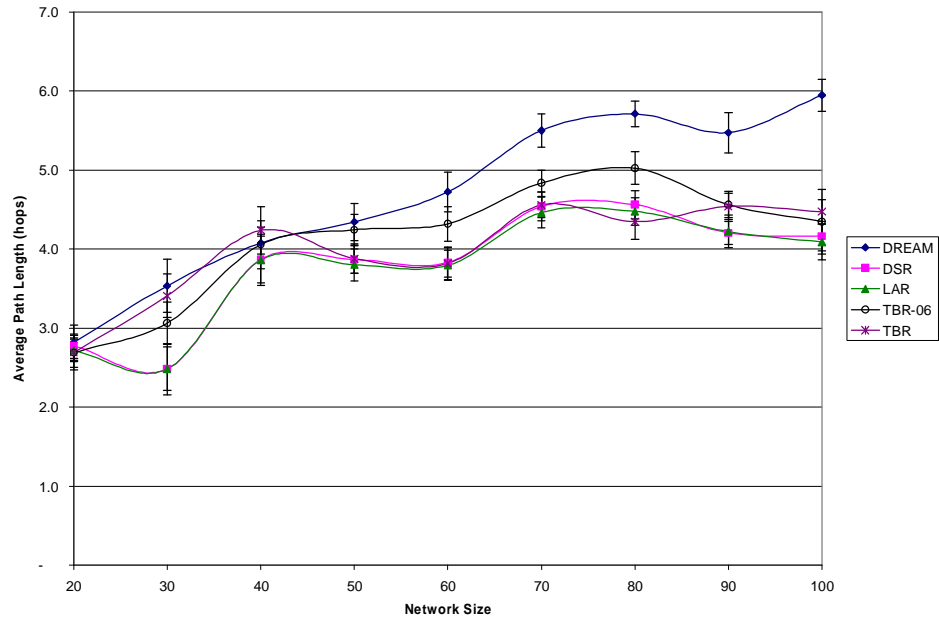


Figure 127: Average Path Length (Flat, Low Mobility, 4 pkts/sec)

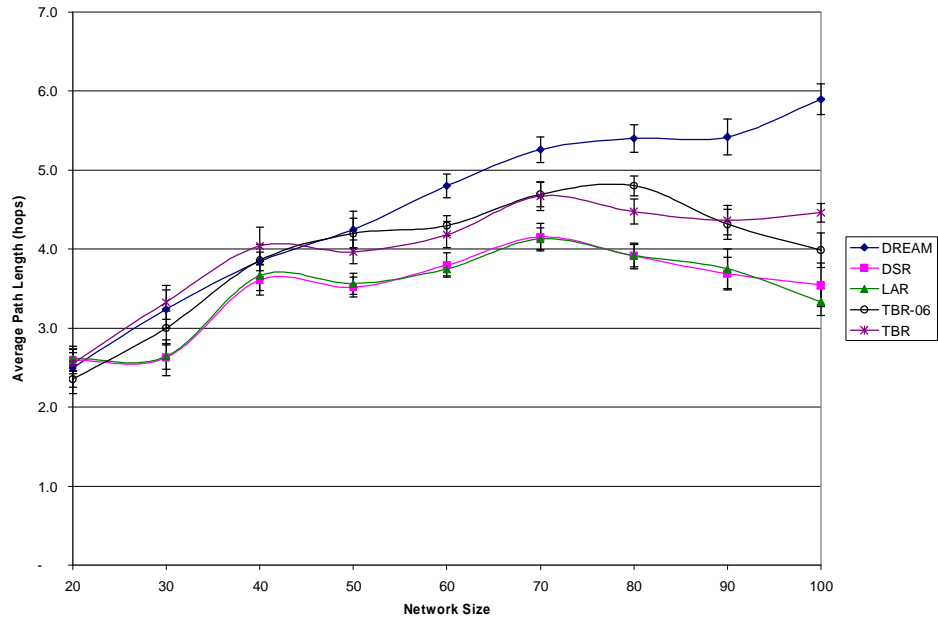


Figure 128: Average Path Length (Flat, Medium Mobility, 4 pkts/sec)

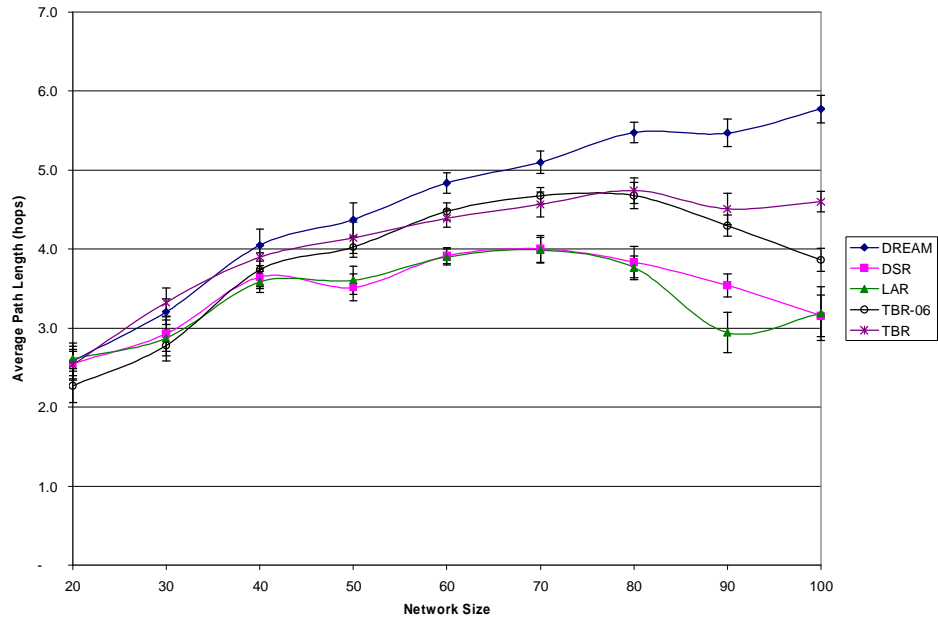


Figure 129: Average Path Length (Flat, High Mobility, 4 pkts/sec)

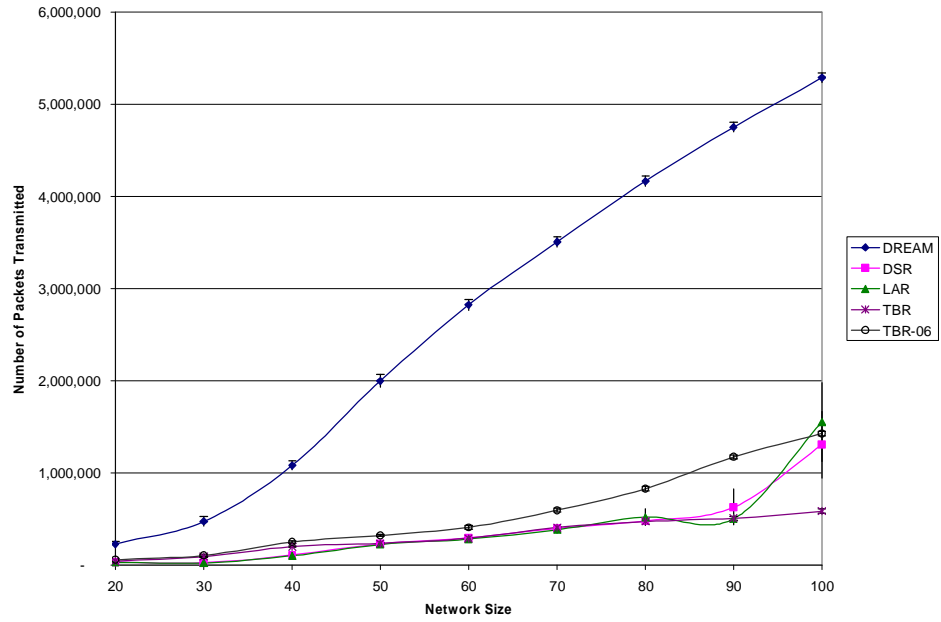


Figure 130: Number of Packets Transmitted (Flat, Low Mobility, 4 pkts/sec)

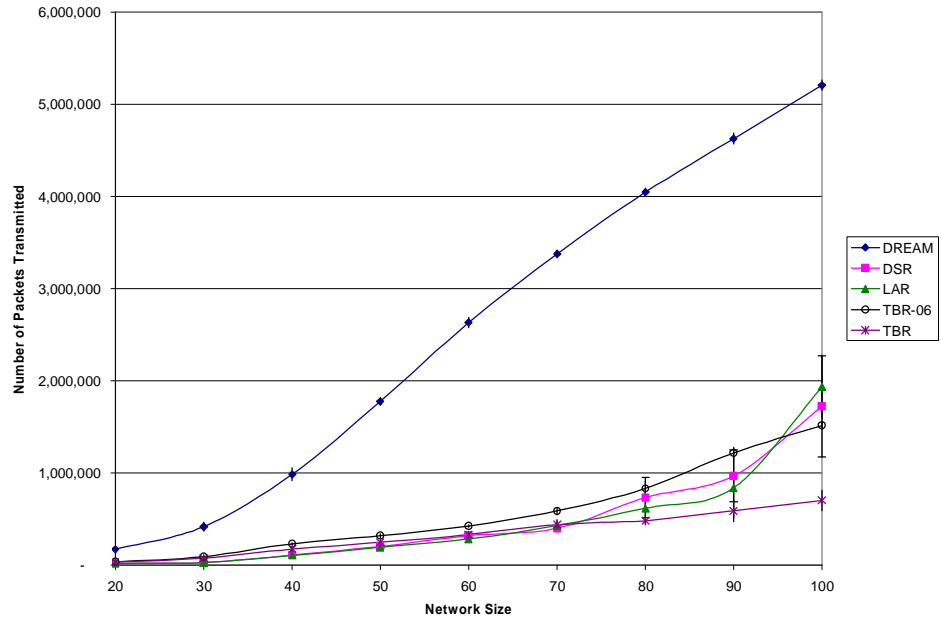


Figure 131: Number of Packets Transmitted (Flat, Medium Mobility, 4 pkts/sec)

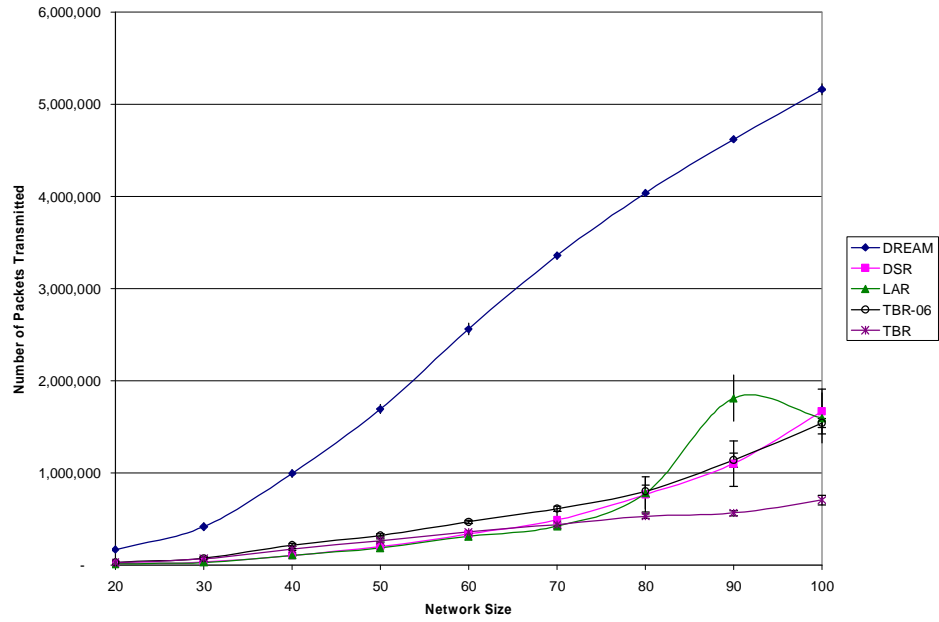


Figure 132: Total Number Packets Transmitted (Flat, High Mobility, 4 pkts/sec)

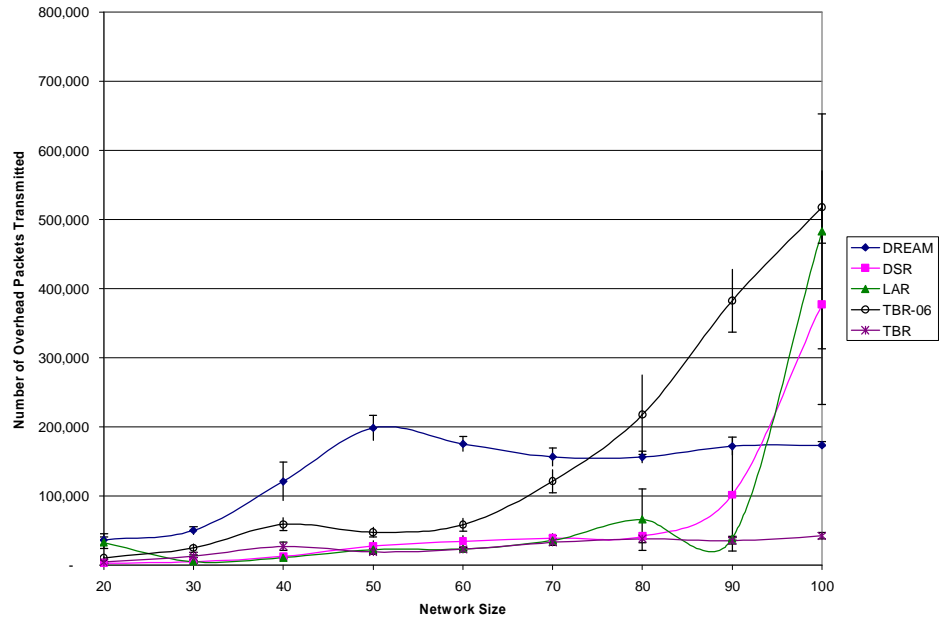


Figure 133: Number of Overhead Packets (Flat, Low Mobility, 4 pkts/sec)

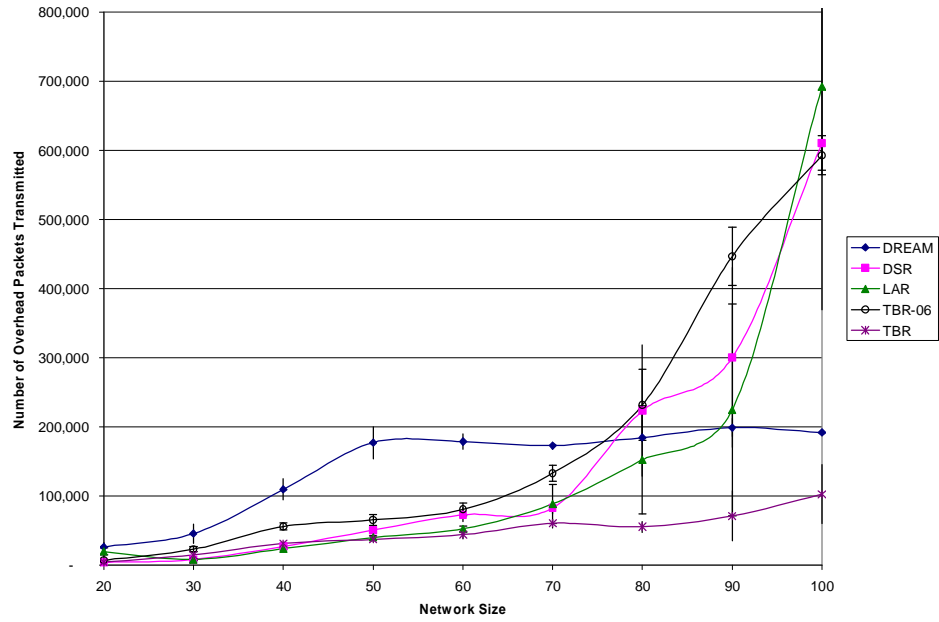


Figure 134: Number of Overhead Packets (Flat, Medium Mobility, 4 pkts/sec)

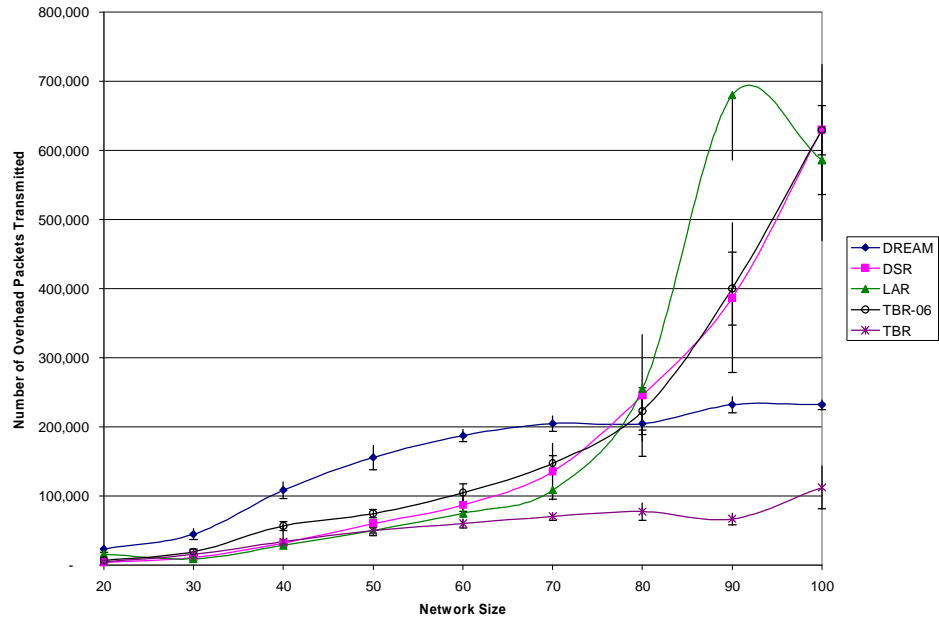


Figure 135: Number of Overhead Packets (Flat, High Mobility, 4 pkts/sec)

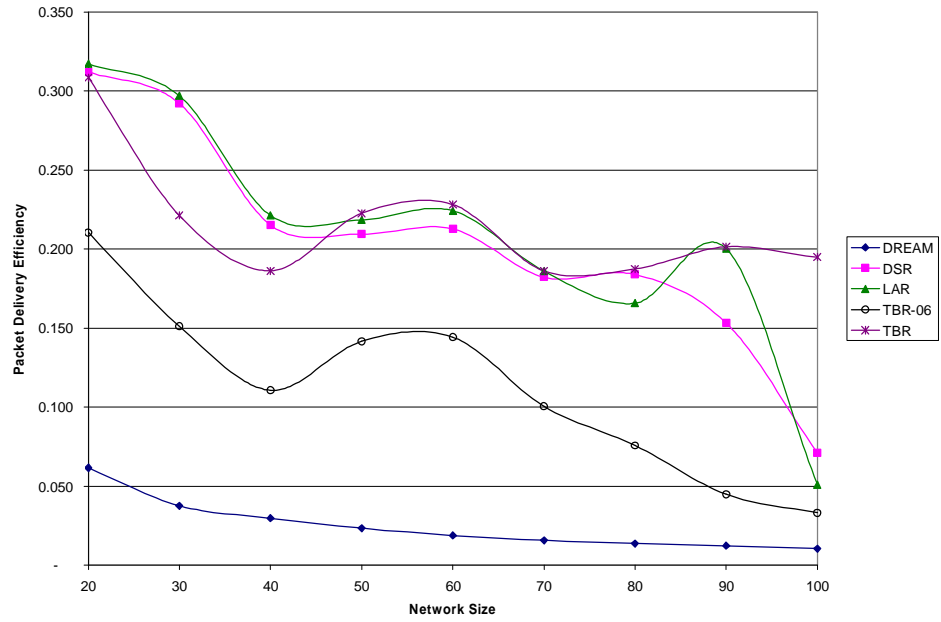


Figure 136: Packet Delivery Efficiency (Flat, Low Mobility, 4 pkts/sec)

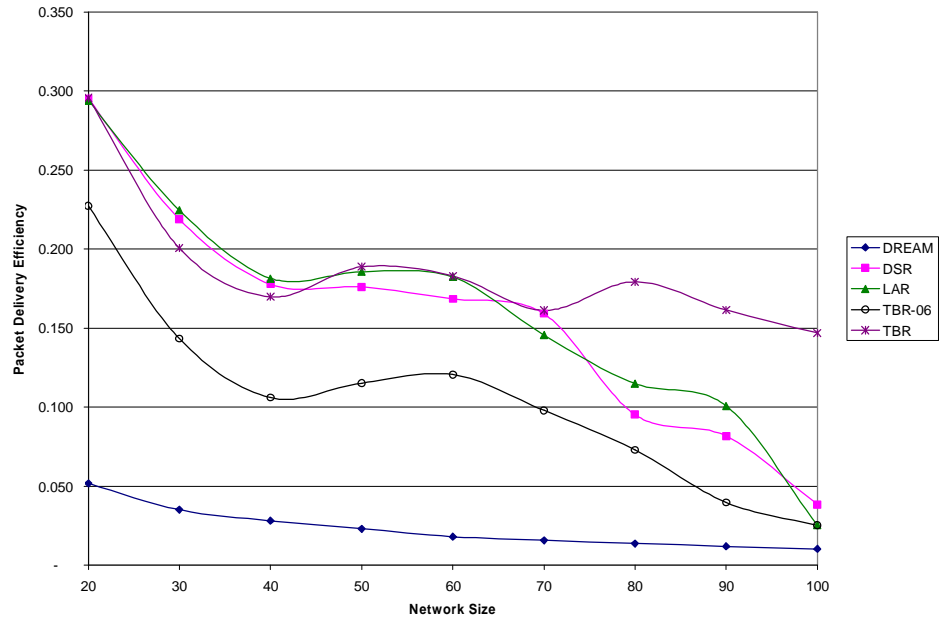


Figure 137: Packet Delivery Efficiency (Flat, Medium Mobility, 4 pkts/sec)

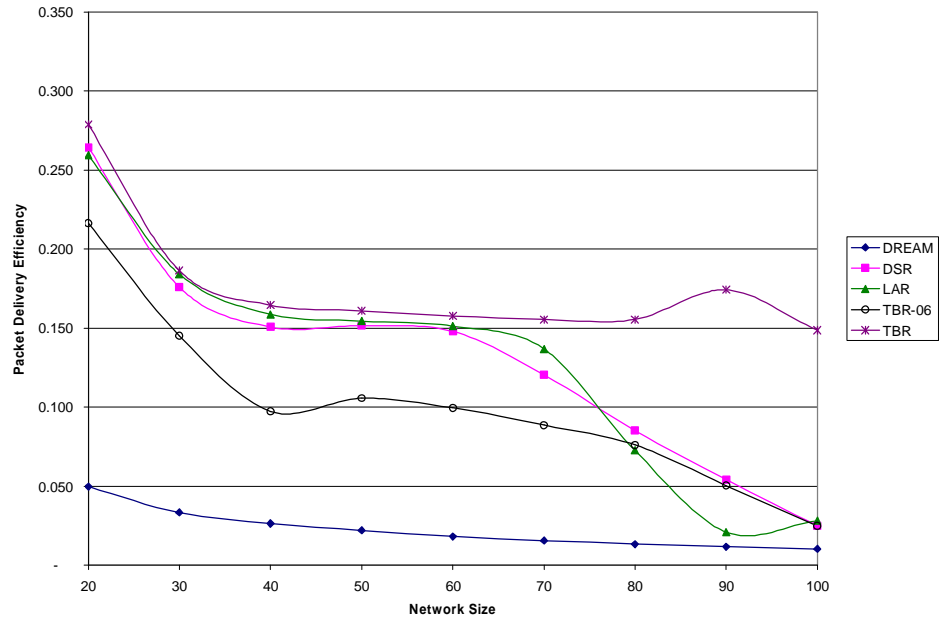


Figure 138: Packet Delivery Efficiency (Flat, High Mobility, 4 pkts/sec)

9.4 Graphs for Experiment 2 Performance Comparison in Hilly Terrain

9.4.1 Hilly Terrain, 1 Packet/Second

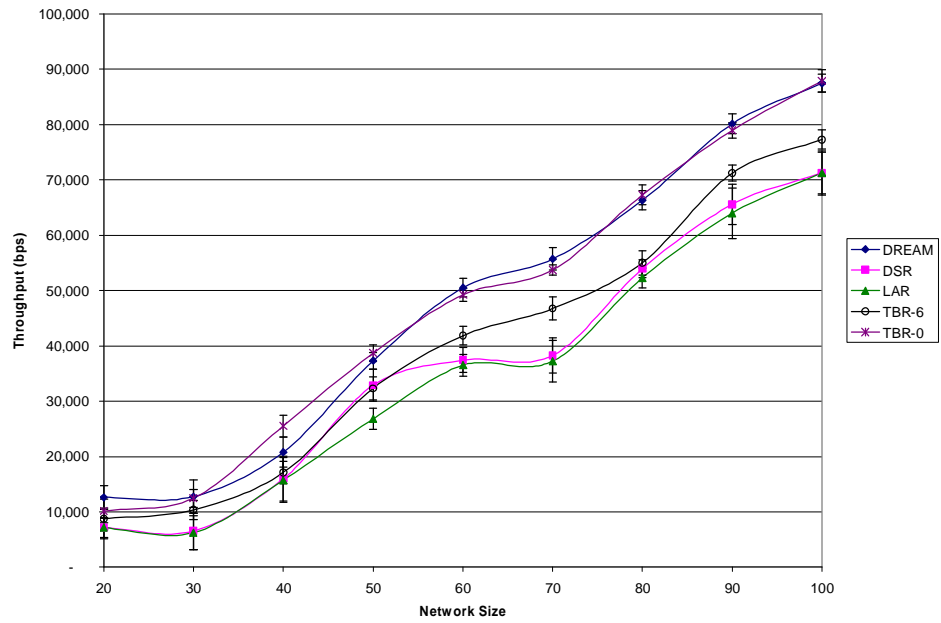


Figure 139: Throughput (Hilly, Low Mobility, 1 pkt/sec)

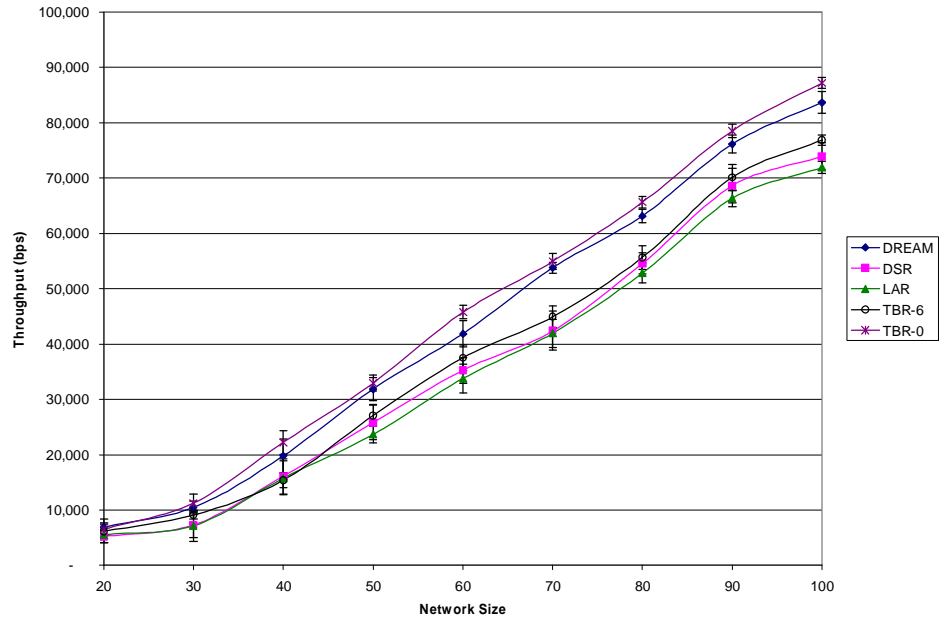


Figure 140: Throughput (Hilly, Medium Mobility, 1 pkt/sec)

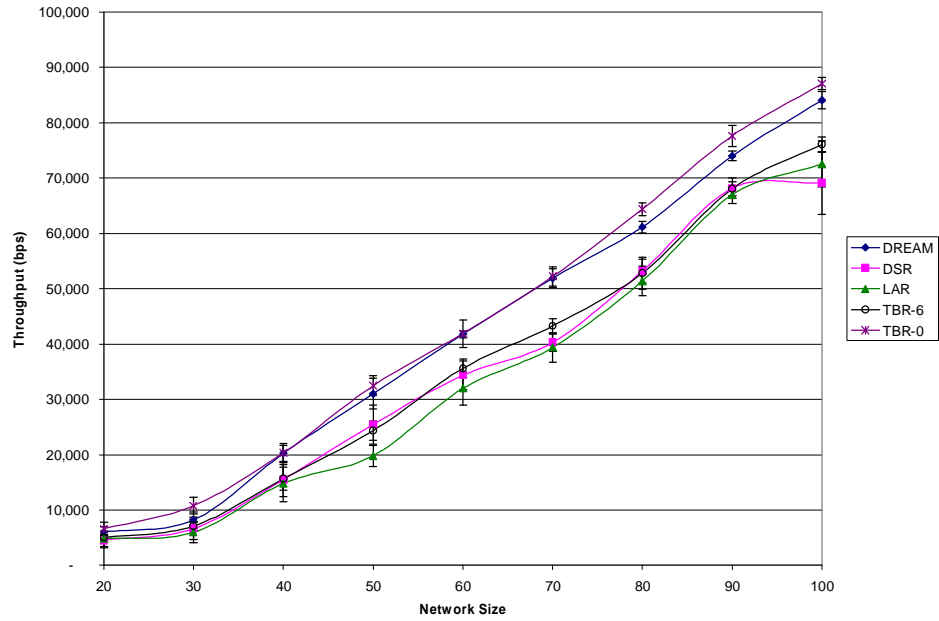


Figure 141: Throughput (Hilly, High Mobility, 1 pkt/sec)

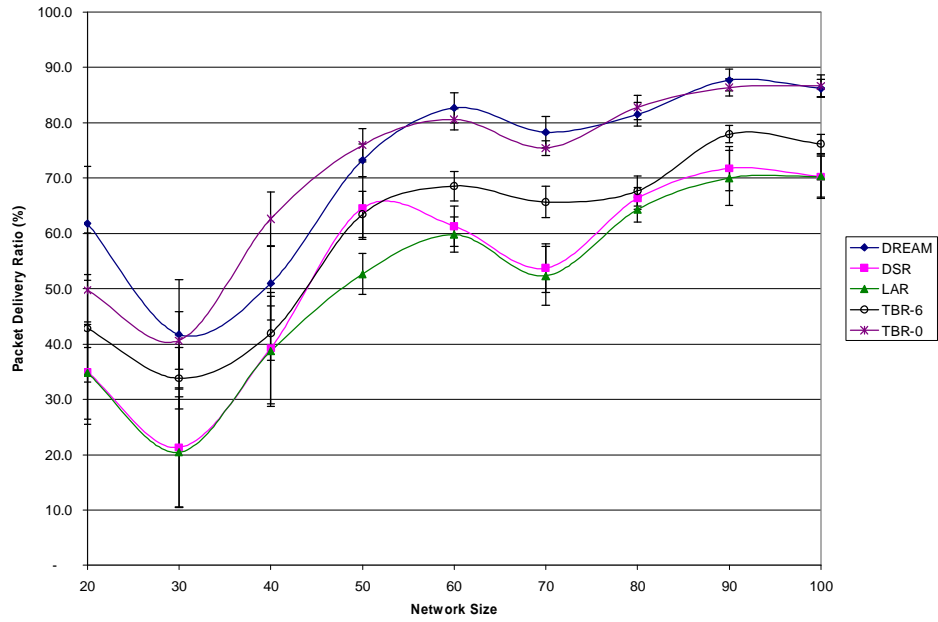


Figure 142: Packet Delivery Ratio (Hilly, Low Mobility, 1 pkt/sec)

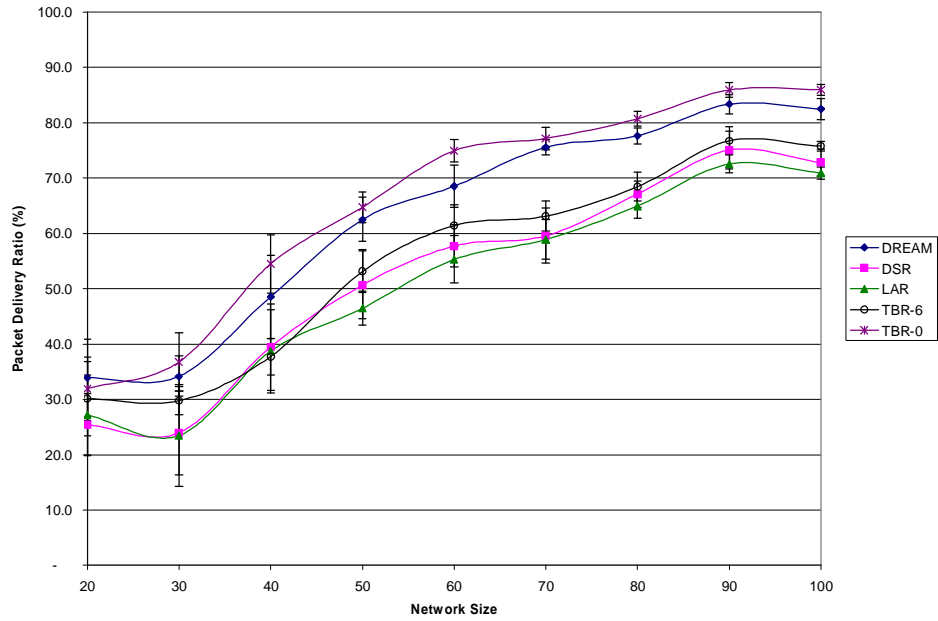


Figure 143: Packet Delivery Ratio (Hilly, Medium Mobility, 1 pkt/sec)

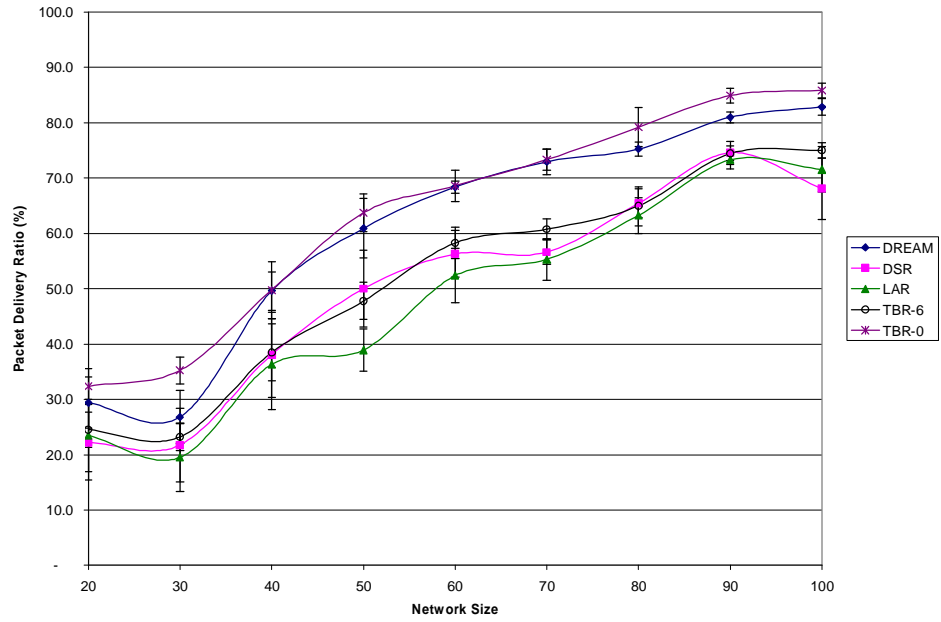


Figure 144: Packet Delivery Ratio (Hilly, High Mobility, 1 pkt/sec)

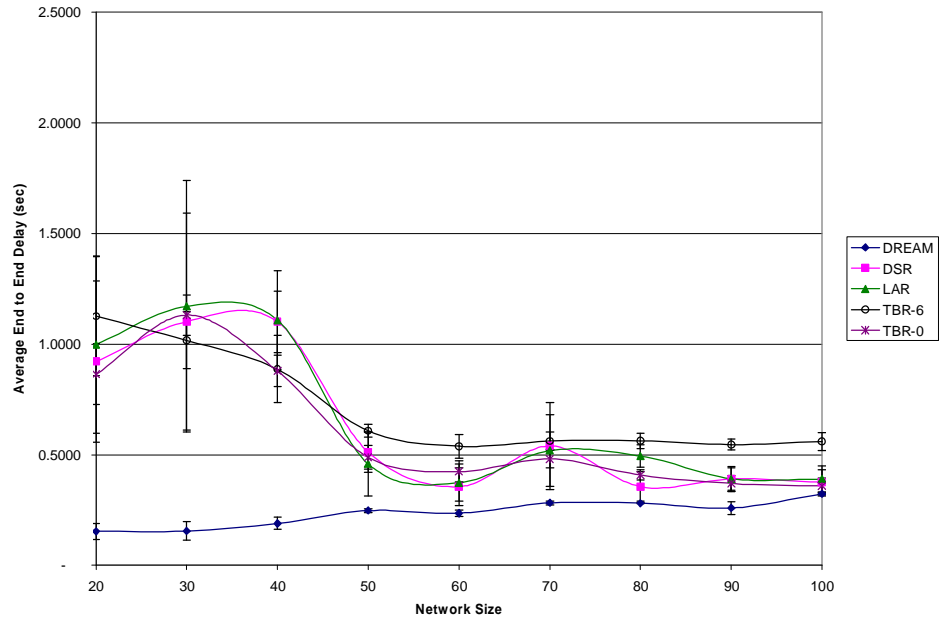


Figure 145: Average End to End Delay (Hilly, Low Mobility, 1 pkt/sec)

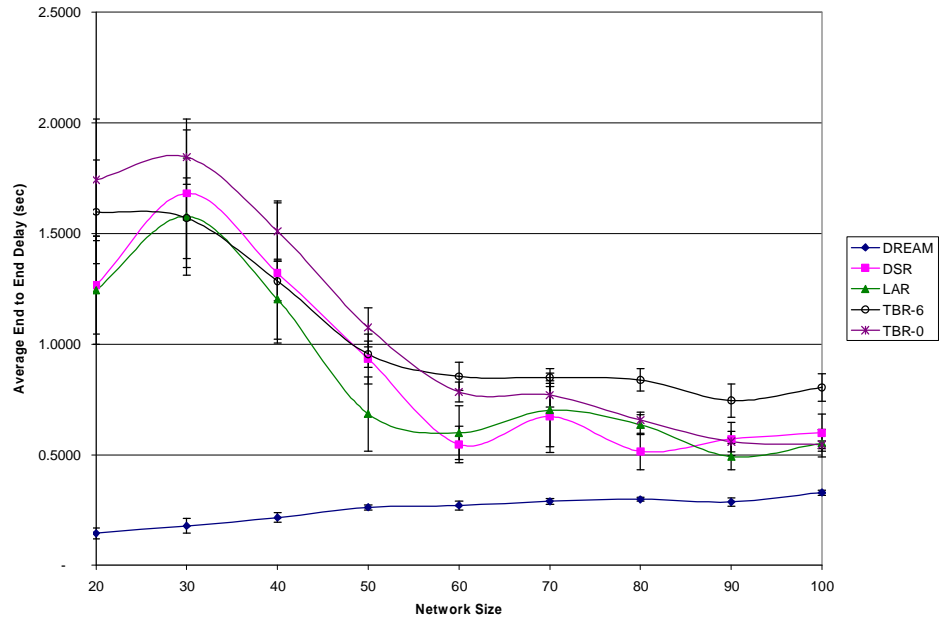


Figure 146: Average End to End Delay (Hilly, Medium Mobility, 1 pkt/sec)

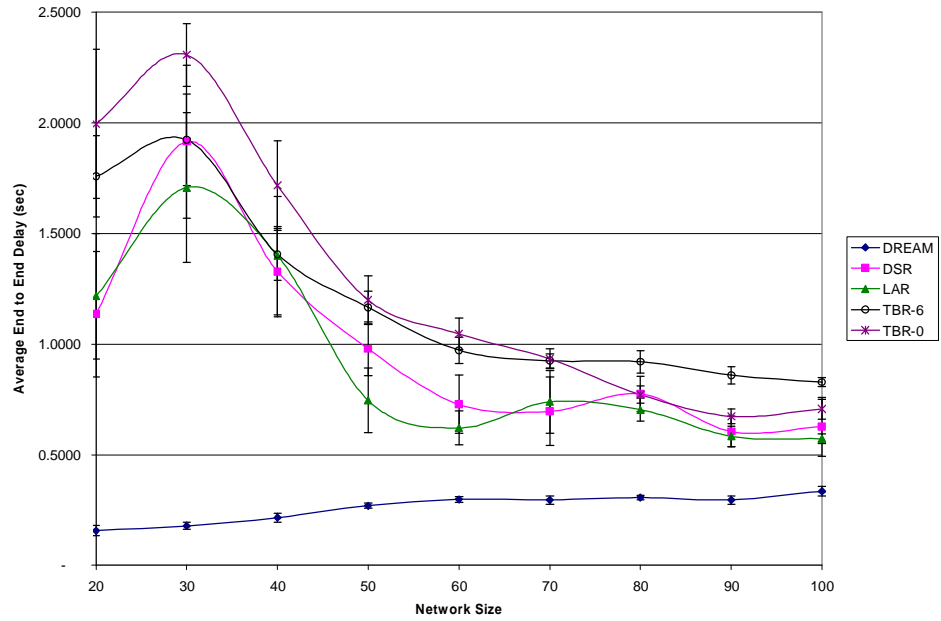


Figure 147: Average End to End Delay (Hilly, High Mobility, 1 pkt/sec)

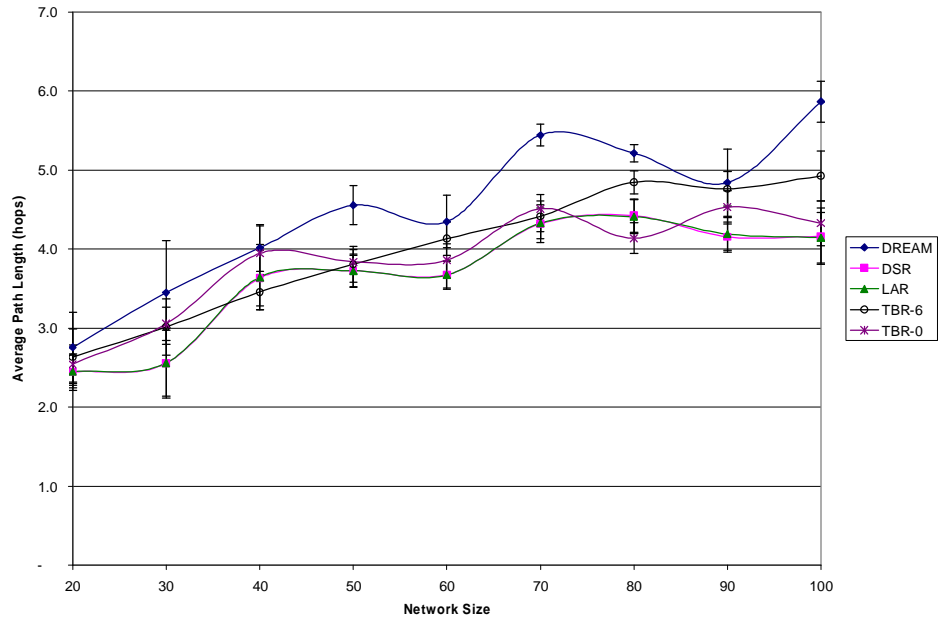


Figure 148: Average Path Length (Hilly, Low Mobility, 1 pkt/sec)

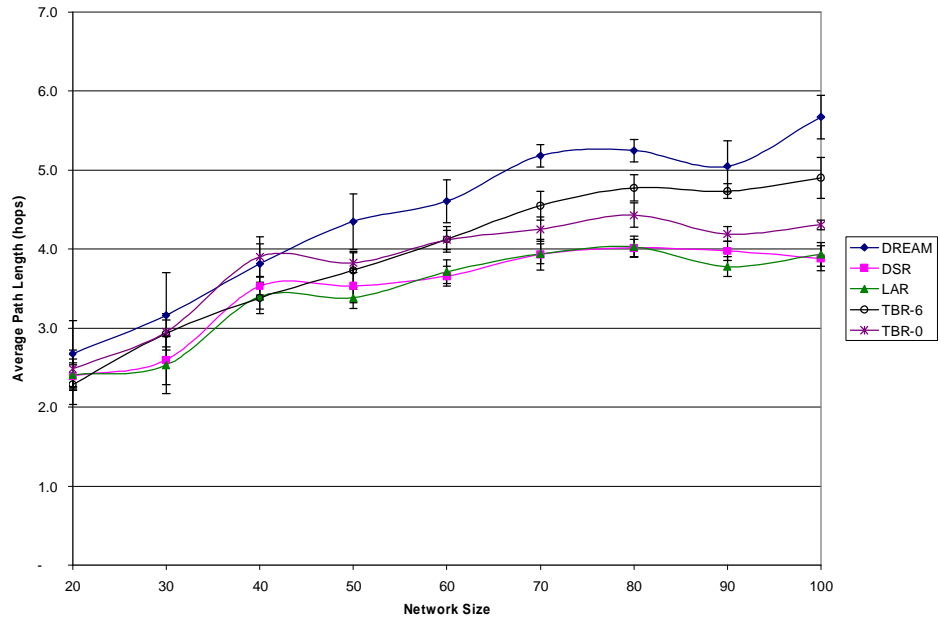


Figure 149: Average Path Length (Hilly, Medium Mobility, 1 pkt/sec)

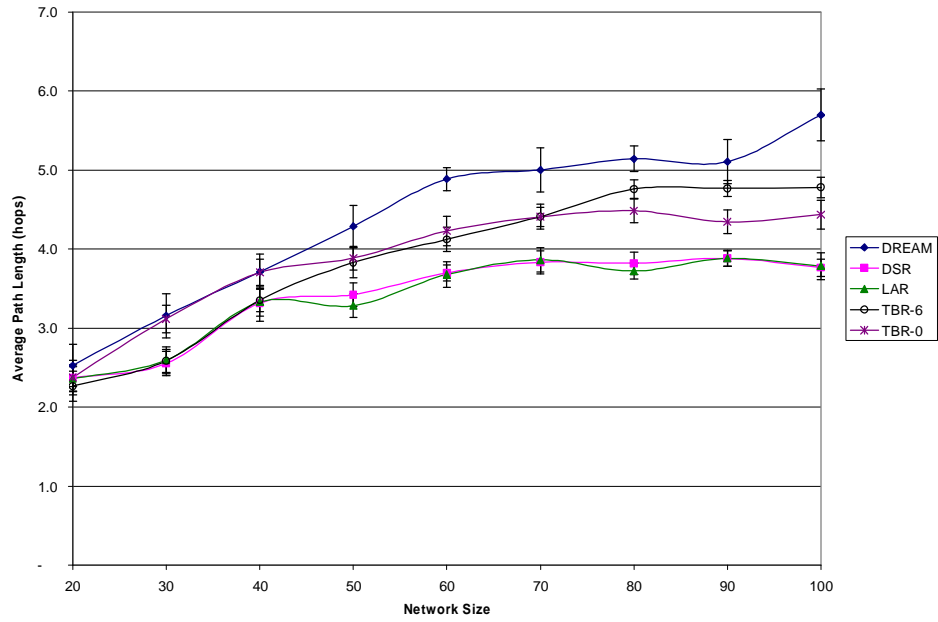


Figure 150: Average Path Length (Hilly, High Mobility, 1 pkt/sec)

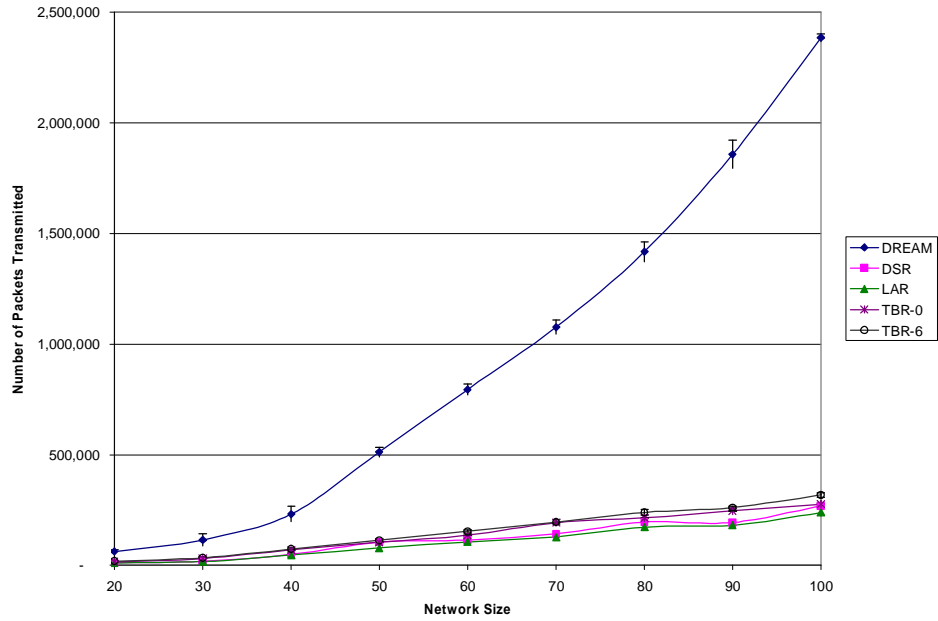


Figure 151: Number of Packets Transmitted (Hilly, Low Mobility, 1 pkt/sec)

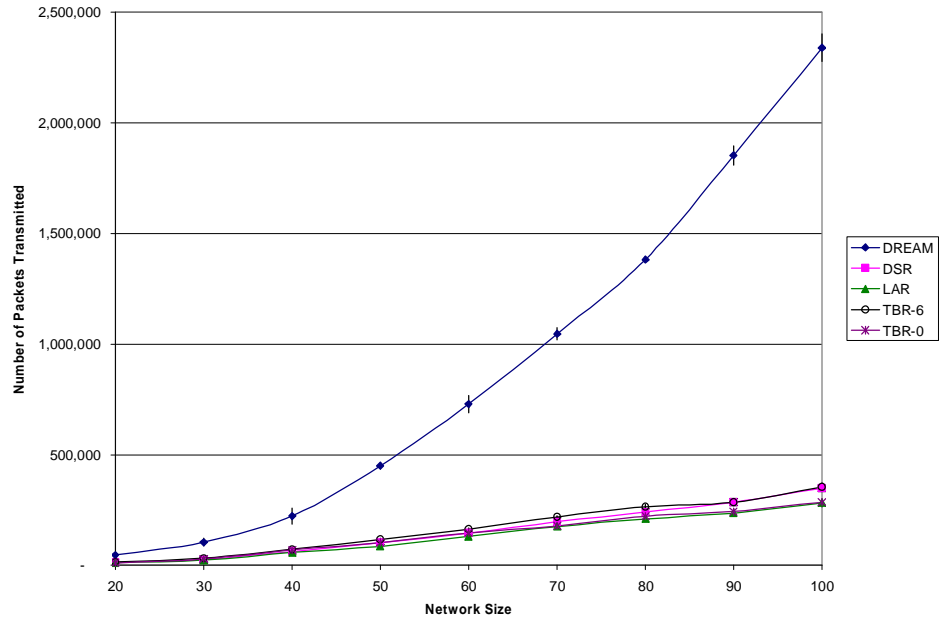


Figure 152: Number of Packets Transmitted (Hilly, Medium Mobility, 1 pkt/sec)

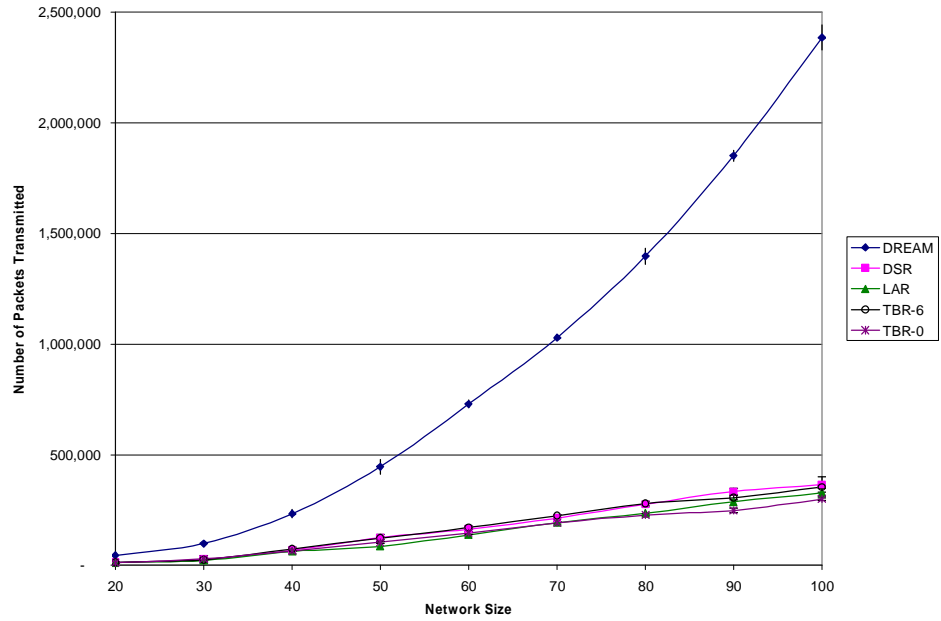


Figure 153: Number of Packets Transmitted (Hilly, High Mobility, 1 pkt/sec)

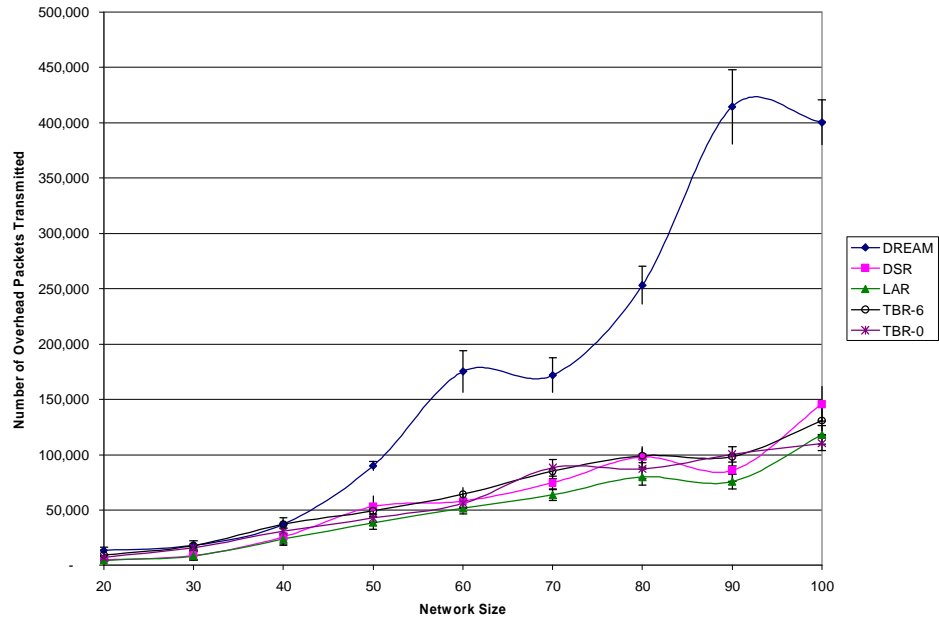


Figure 154: Total Number of Overhead Packets (Hilly, Low Mobility, 1 pkt/sec)

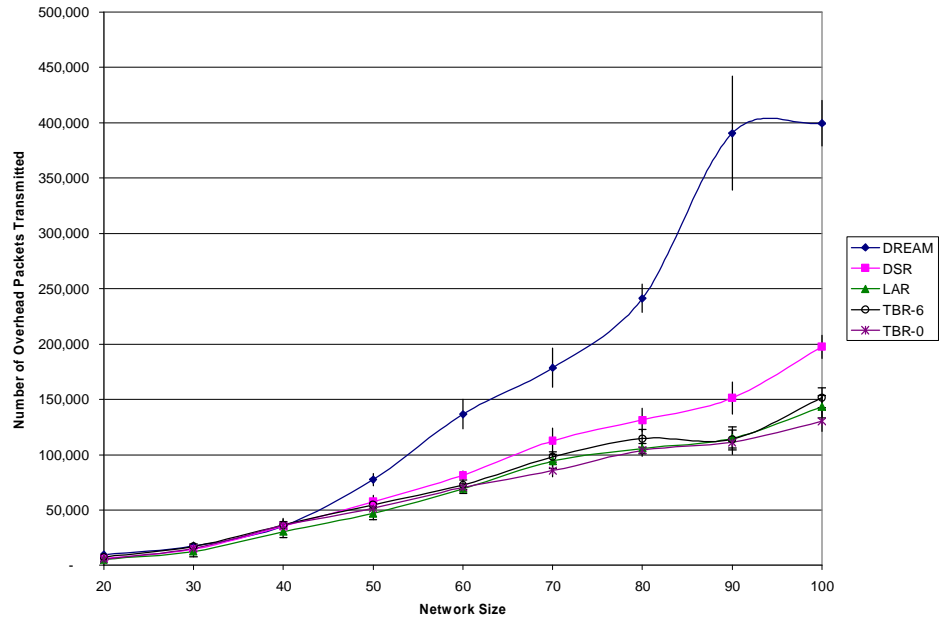


Figure 155: Total Number of Overhead Packets (Hilly, Medium Mobility, 1 pkt/sec)

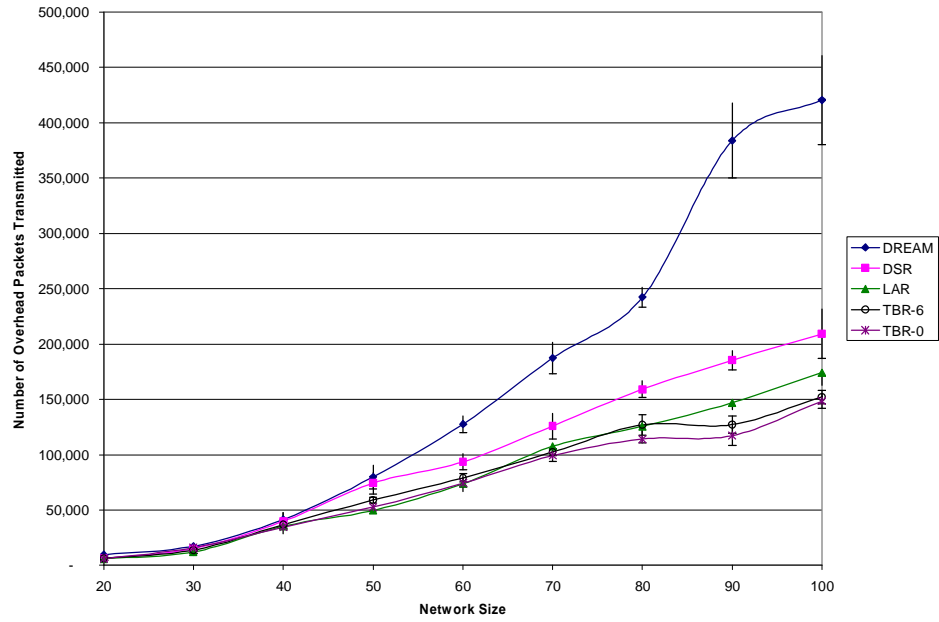


Figure 156: Total Number of Overhead Packets (Hilly, High Mobility, 1 pkt/sec)

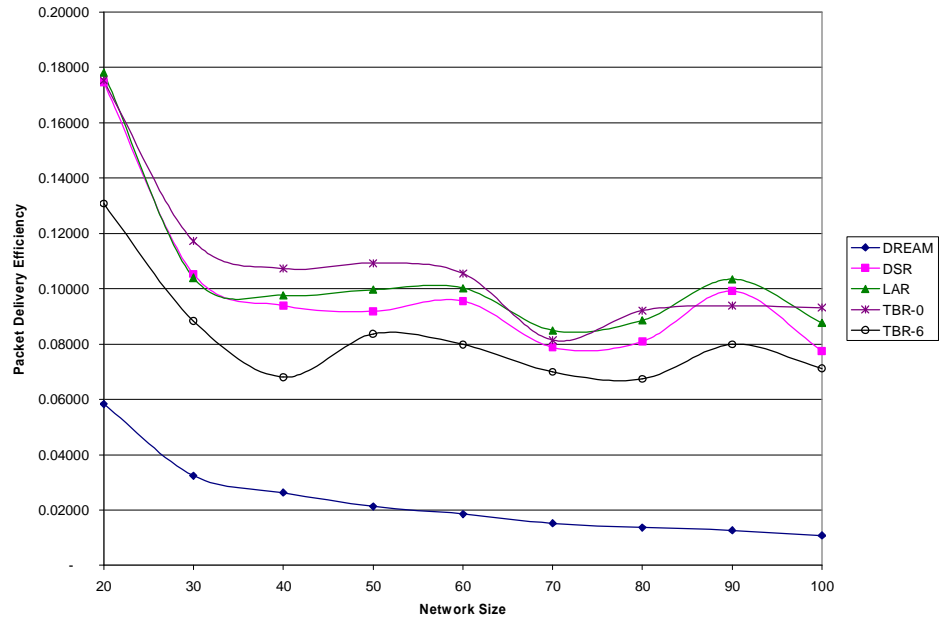


Figure 157: Packet Delivery Efficiency (Hilly, Low Mobility, 1 pkt/sec)

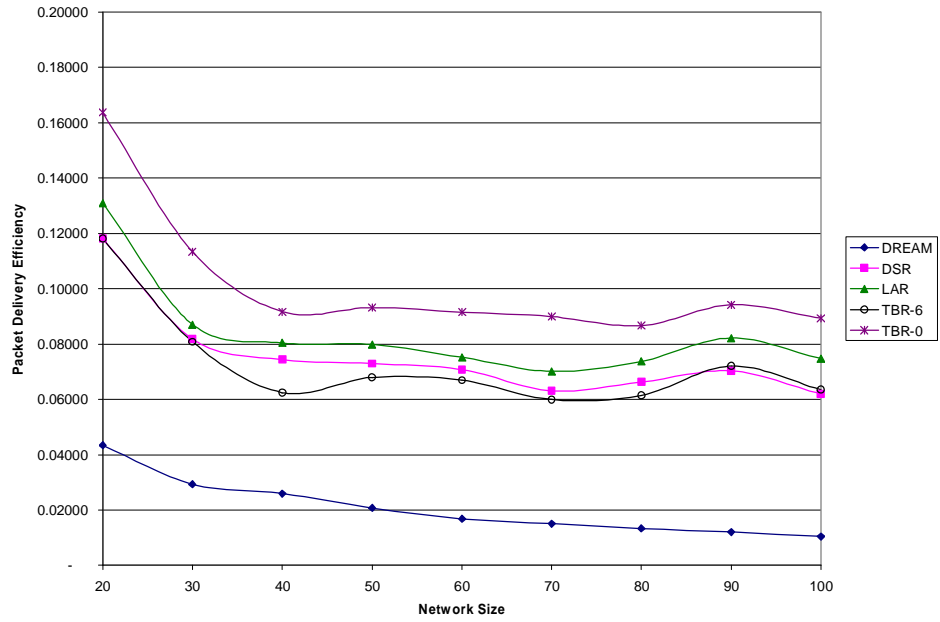


Figure 158: Packet Delivery Efficiency (Hilly, Medium Mobility, 1 pkt/sec)

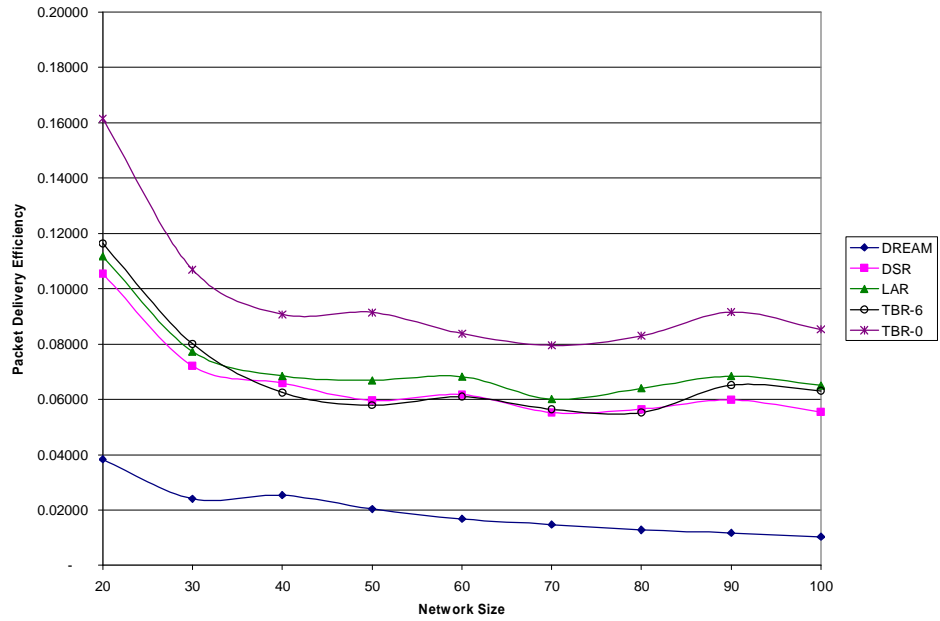


Figure 159: Packet Delivery Efficiency (Hilly, High Mobility, 1 pkt/sec)

9.4.2 Hilly Terrain, 2 Packets/Second

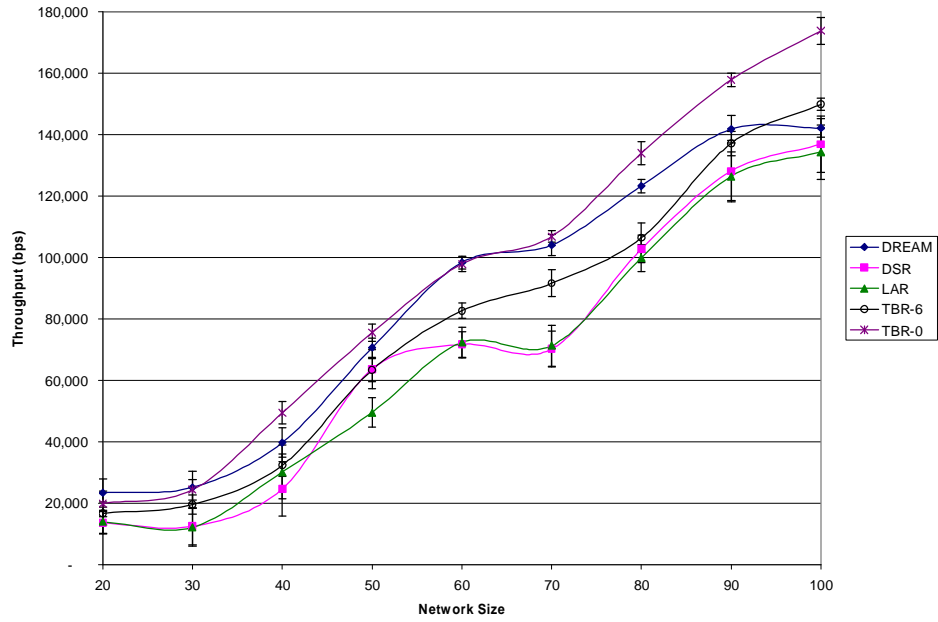


Figure 160: Throughput (Hilly, Low Mobility, 2 pkts/sec)

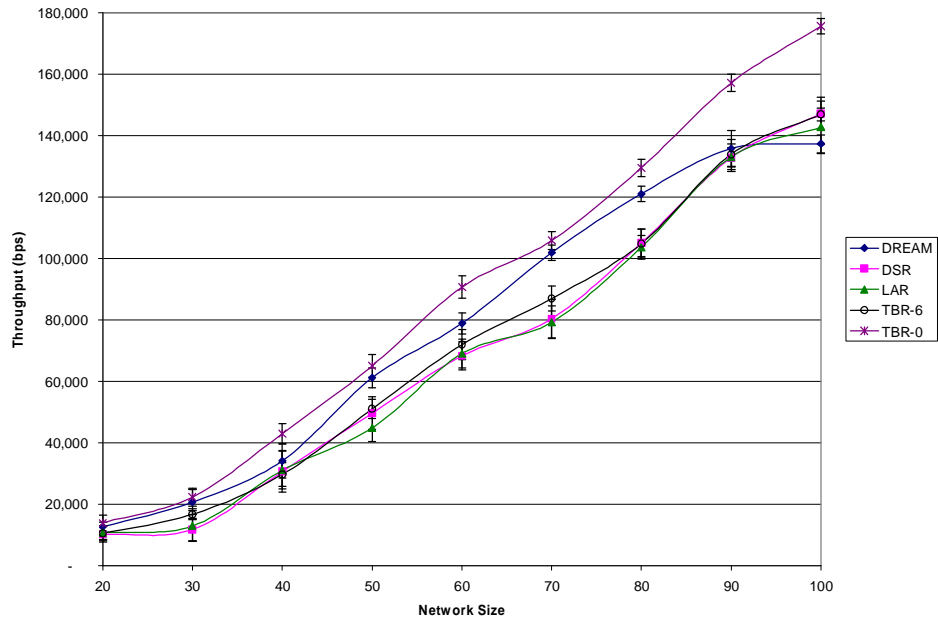


Figure 161: Throughput (Hilly, Medium Mobility, 2 pkts/sec)

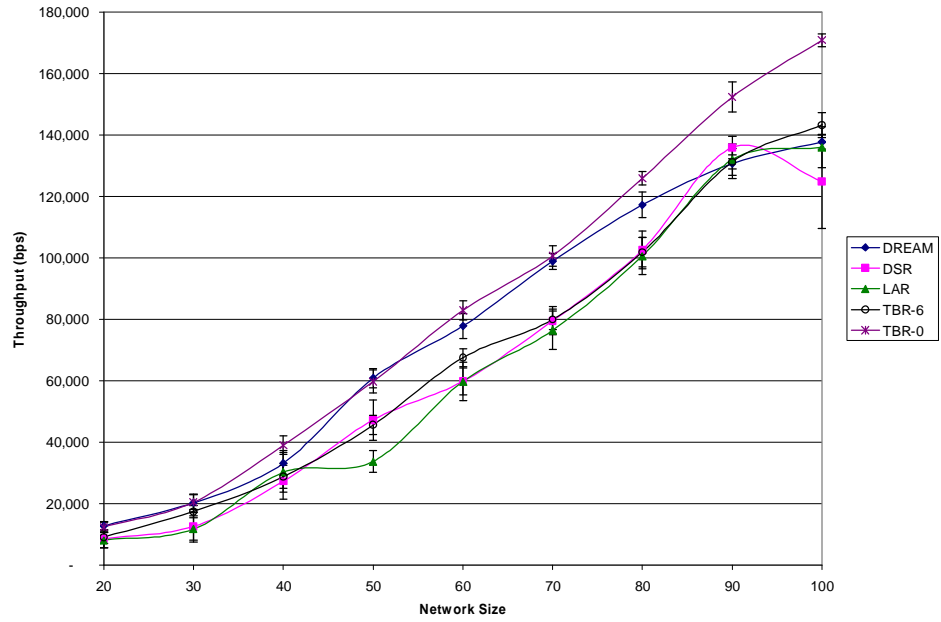


Figure 162: Throughput (Hilly High Mobility, 2 pkts/sec)

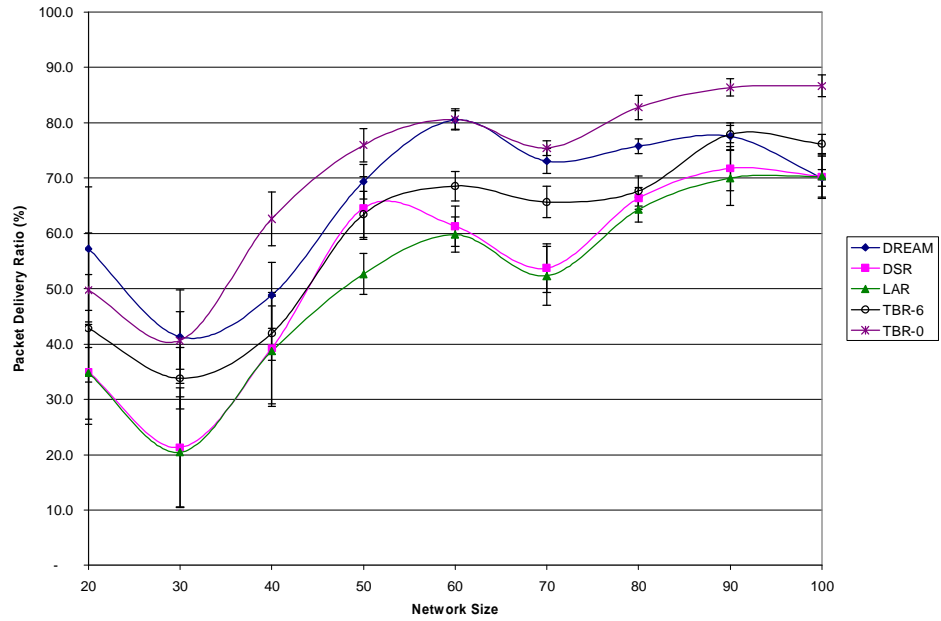


Figure 163: Packet Delivery Ratio (Hilly, Low Mobility, 2 pkts/sec)

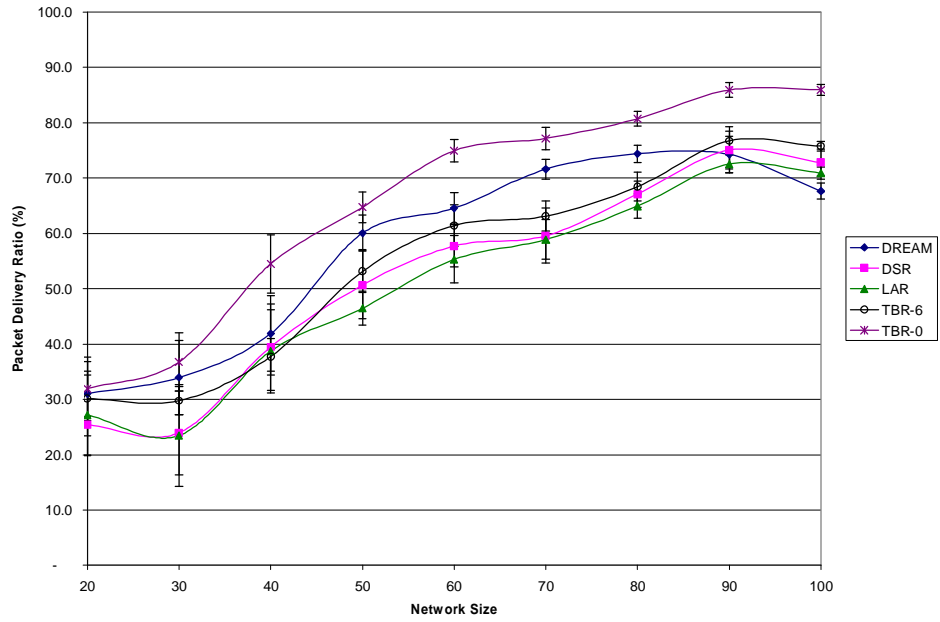


Figure 164: Packet Delivery Ratio (Hilly, Medium Mobility, 2 pkts/sec)

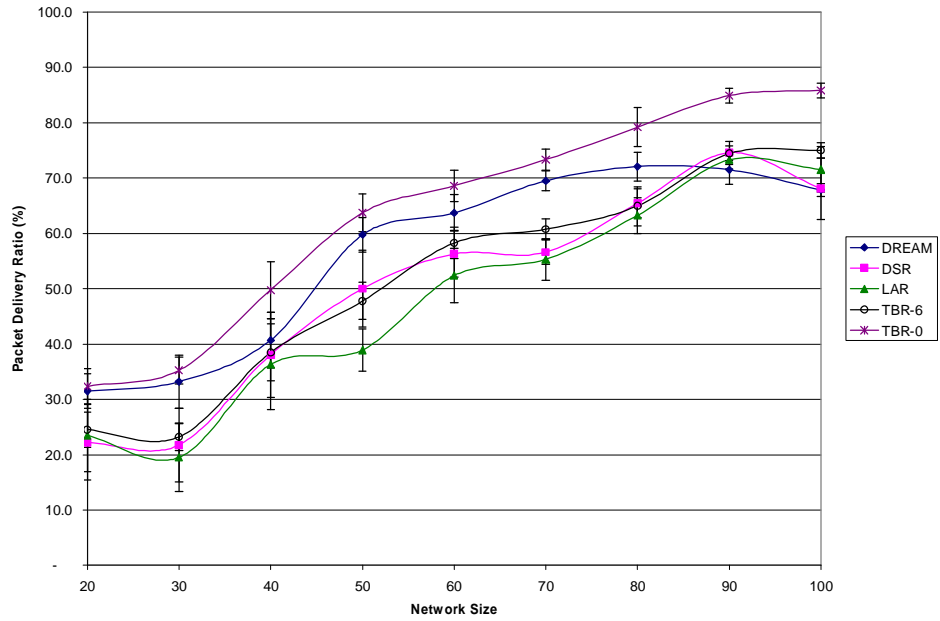


Figure 165: Packet Delivery Ratio (Hilly, High Mobility 2 pkts/sec)

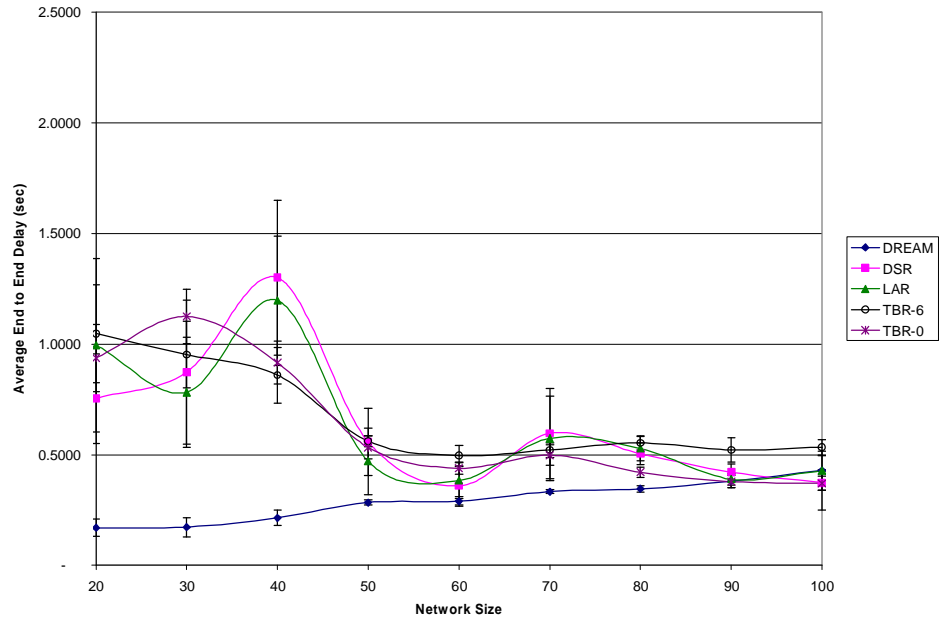


Figure 166: Average End to End Delay (Hilly, Low Mobility, 2 pkts/sec)

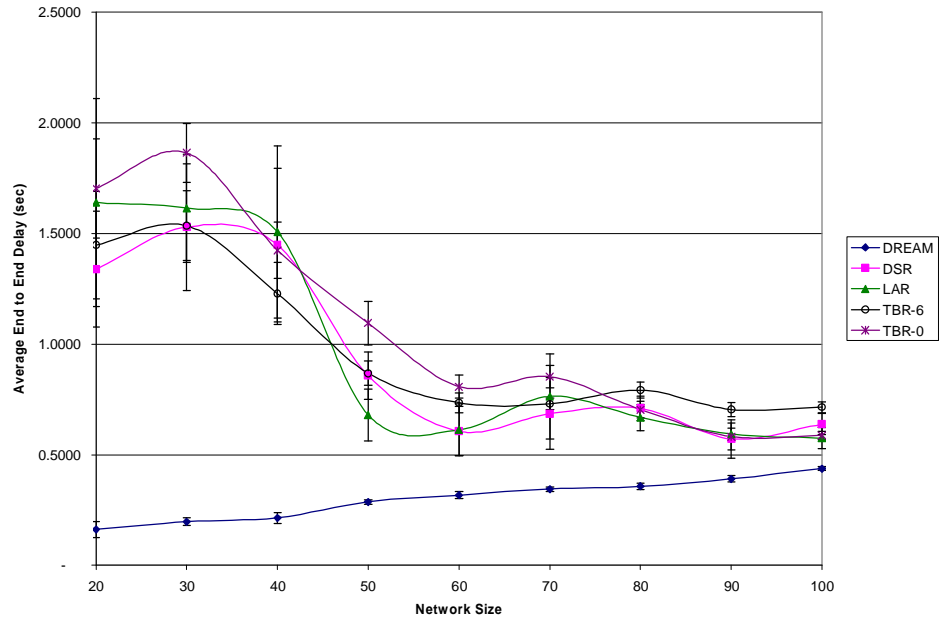


Figure 167: Average End to End Delay (Hilly, Medium Mobility, 2 pkts/sec)

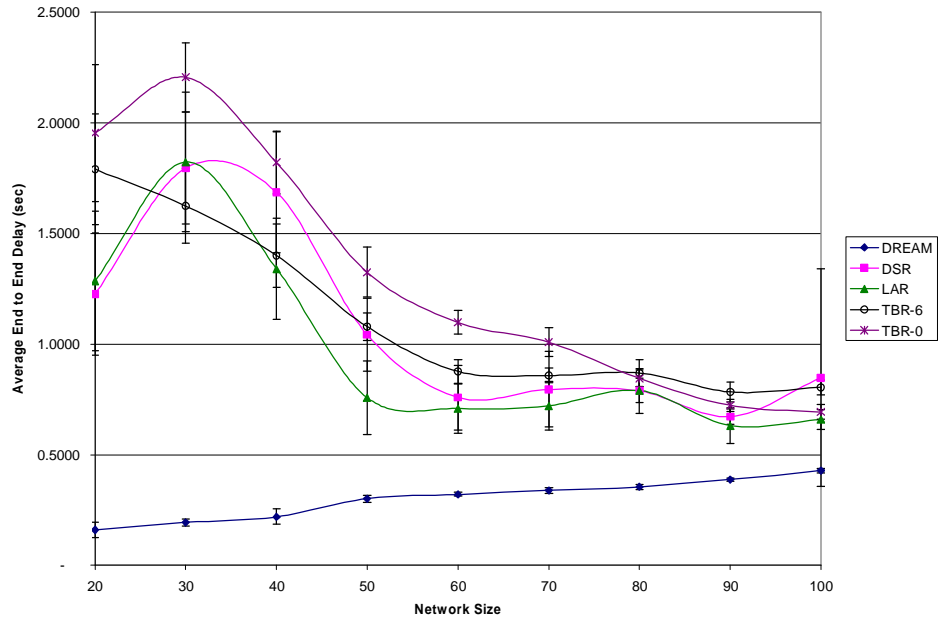


Figure 168: Average End to End Delay (Hilly, High Mobility, 2 pkts/sec)

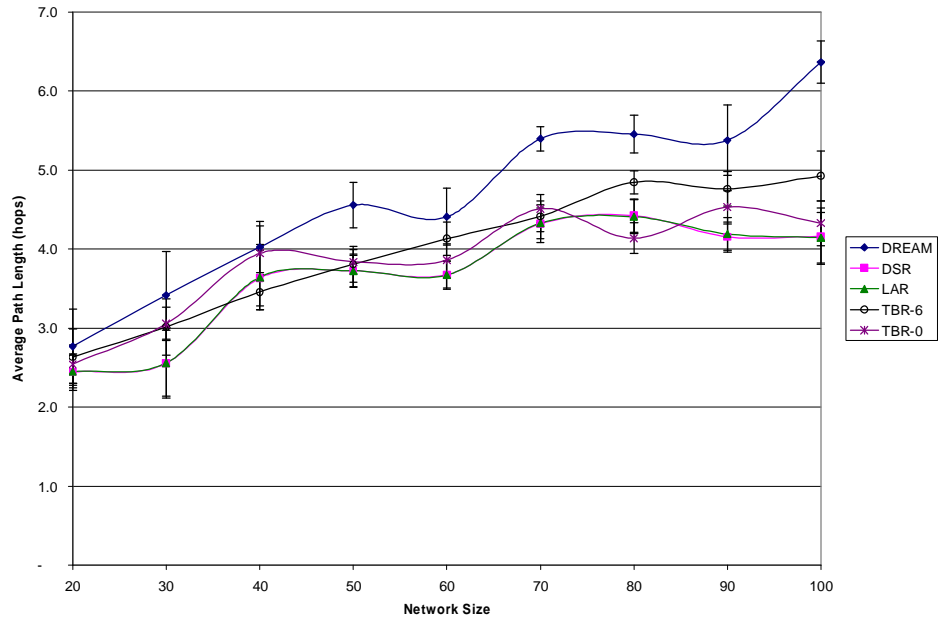


Figure 169: Average Path Length (Hilly, Low Mobility, 2 pkts/sec)

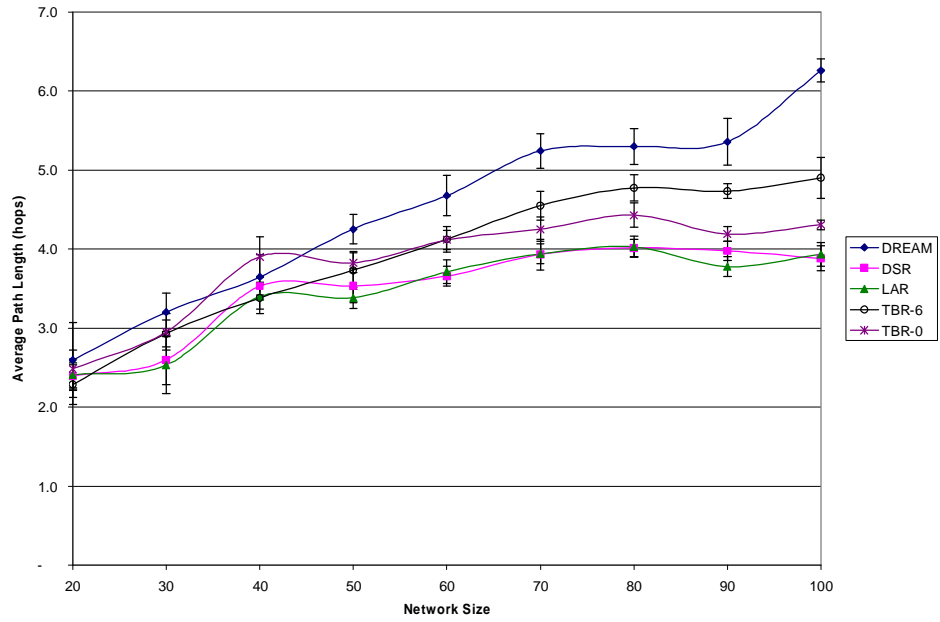


Figure 170: Average Path Length (Hilly, Medium Mobility, 2 pkts/sec)

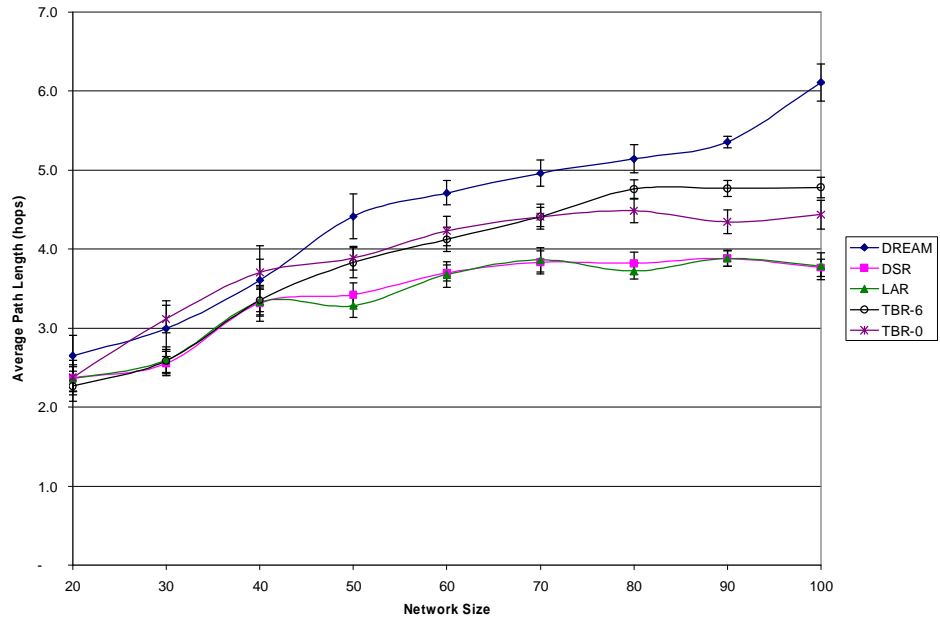


Figure 171: Average Path Length (Hilly, High Mobility, 2 pkts/sec)

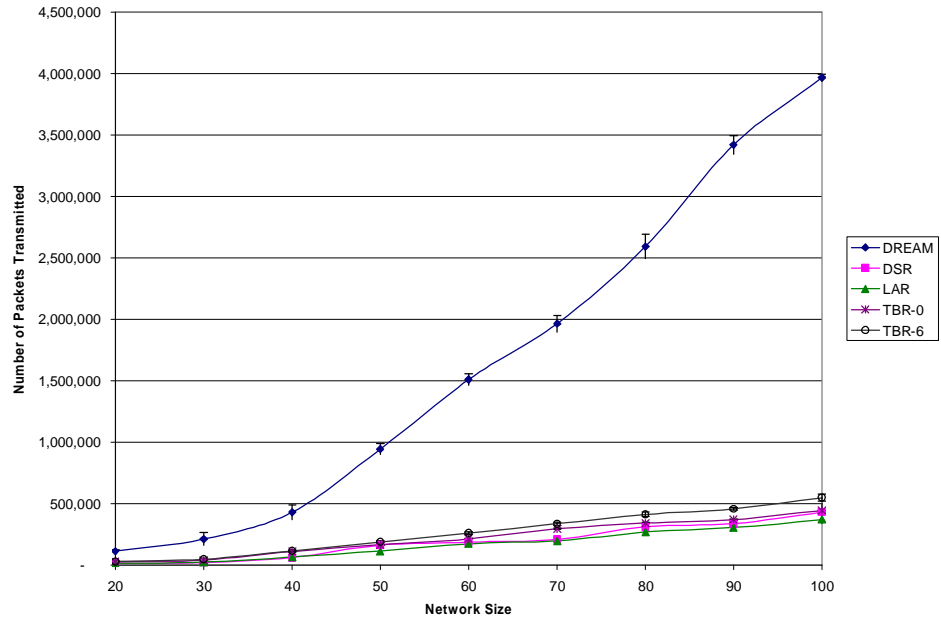


Figure 172: Number of Packets Transmitted (Hilly, Low Mobility, 2 pkts/sec)

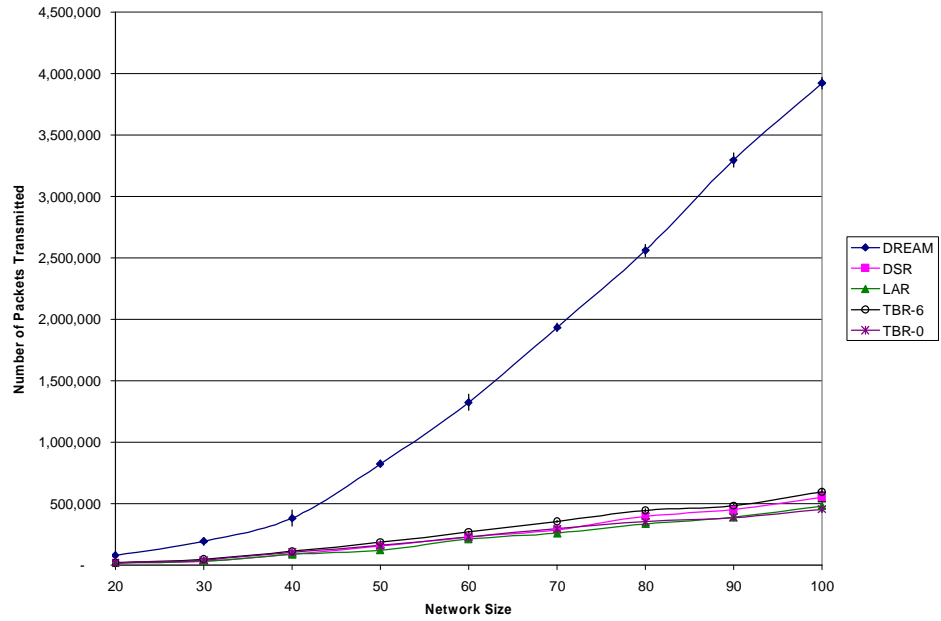


Figure 173: Number of Packets Transmitted (Hilly, Medium Mobility, 2 pkts/sec)

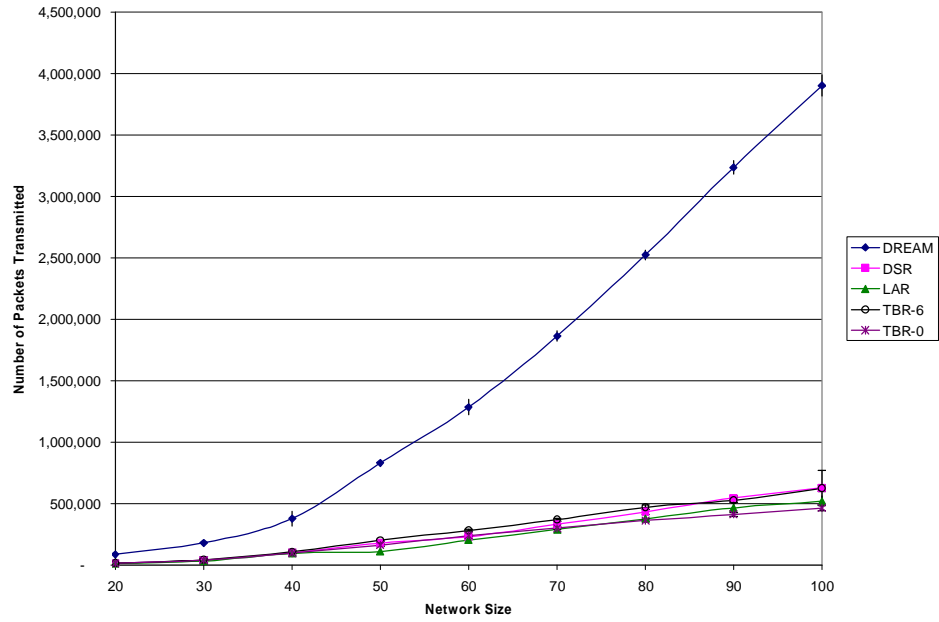


Figure 174: Number of Packets Transmitted (Hilly, High Mobility, 2 pkts/sec)

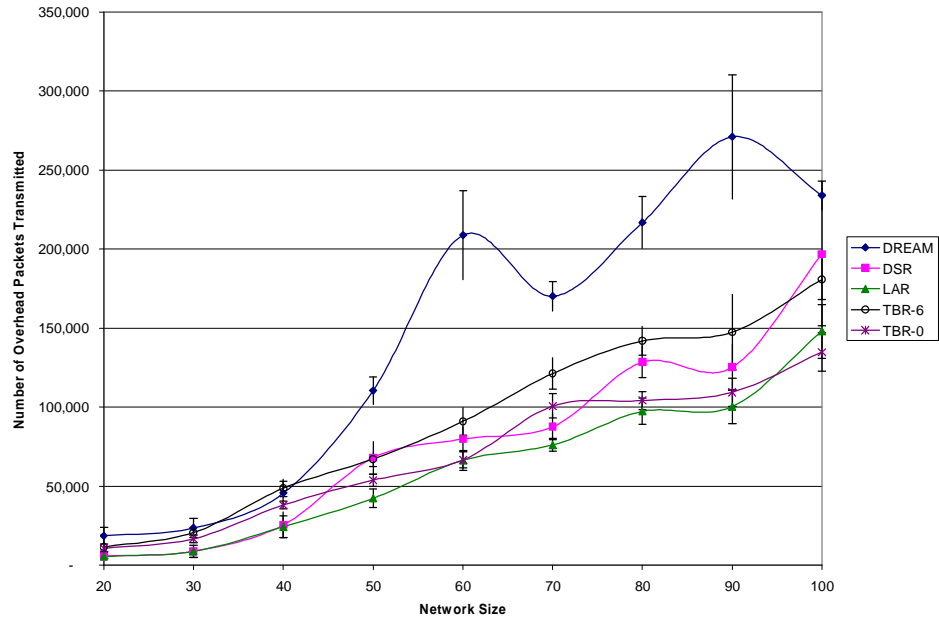


Figure 175: Number of Overhead Packets (Hilly, Low Mobility, 2 pkts/sec)

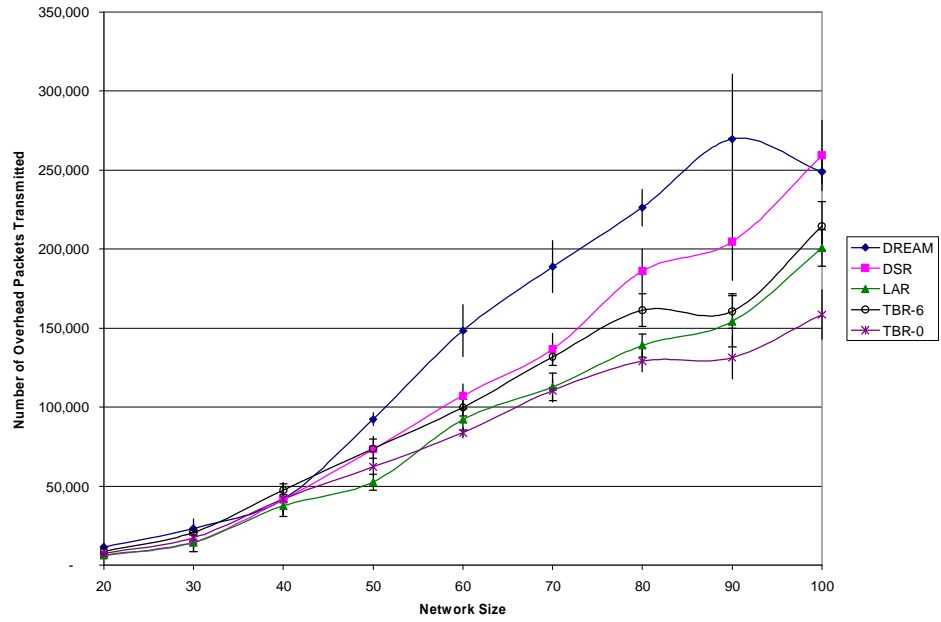


Figure 176: Number of Overhead Packets (Hilly, Medium Mobility, 2 pkts/sec)

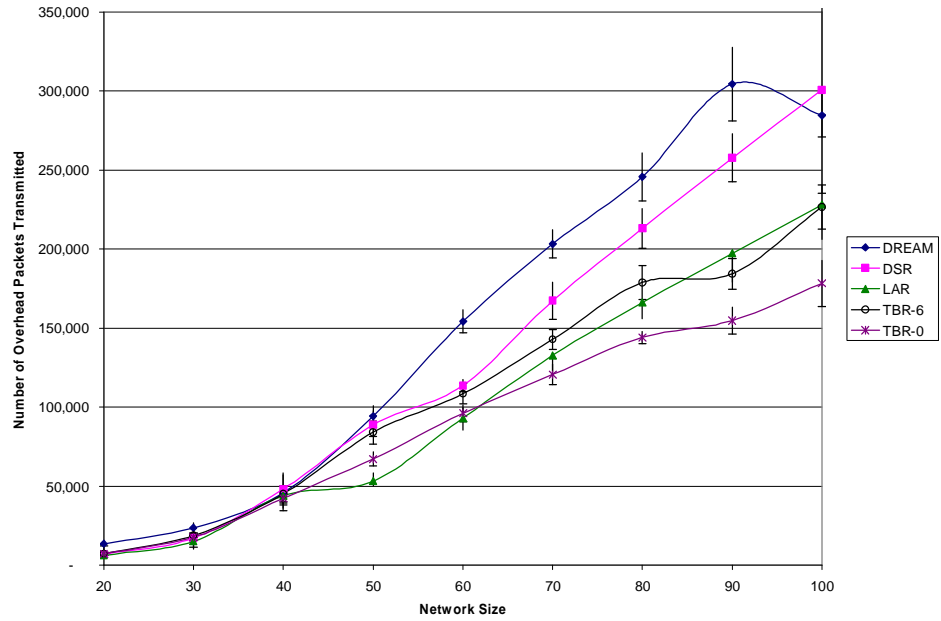


Figure 177: Number of Overhead Packets (Hilly, High Mobility, 2 pkts/sec)

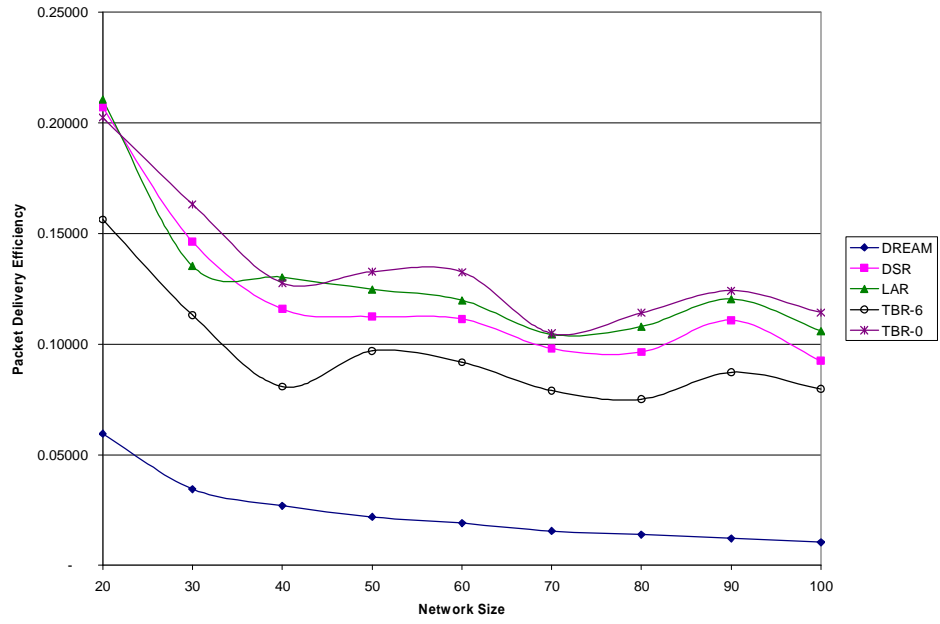


Figure 178: Packet Delivery Efficiency (Hilly, Low Mobility, 2 pkts/sec)

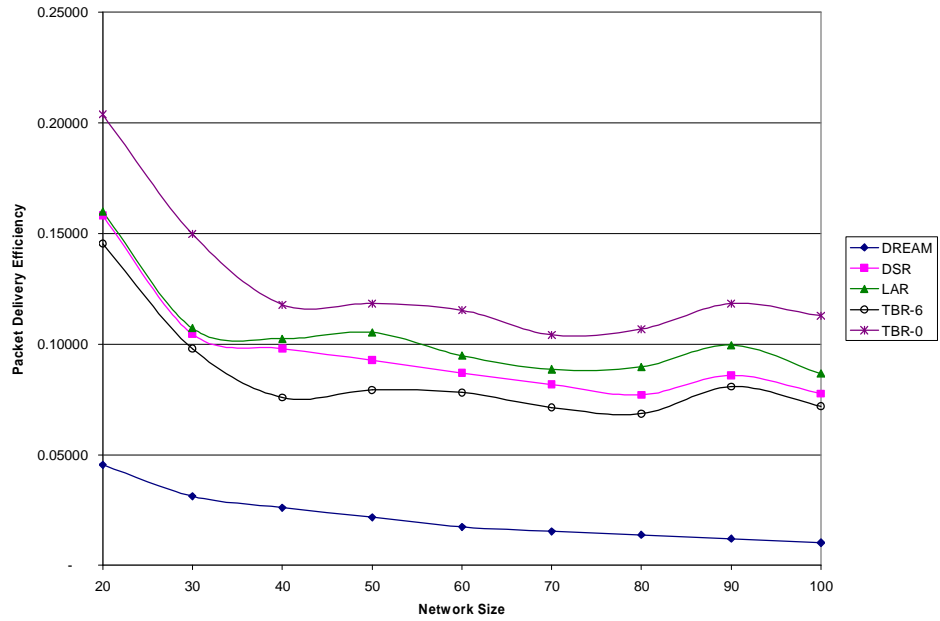


Figure 179: Packet Delivery Efficiency (Hilly, Medium Mobility, 2 pkts/sec)

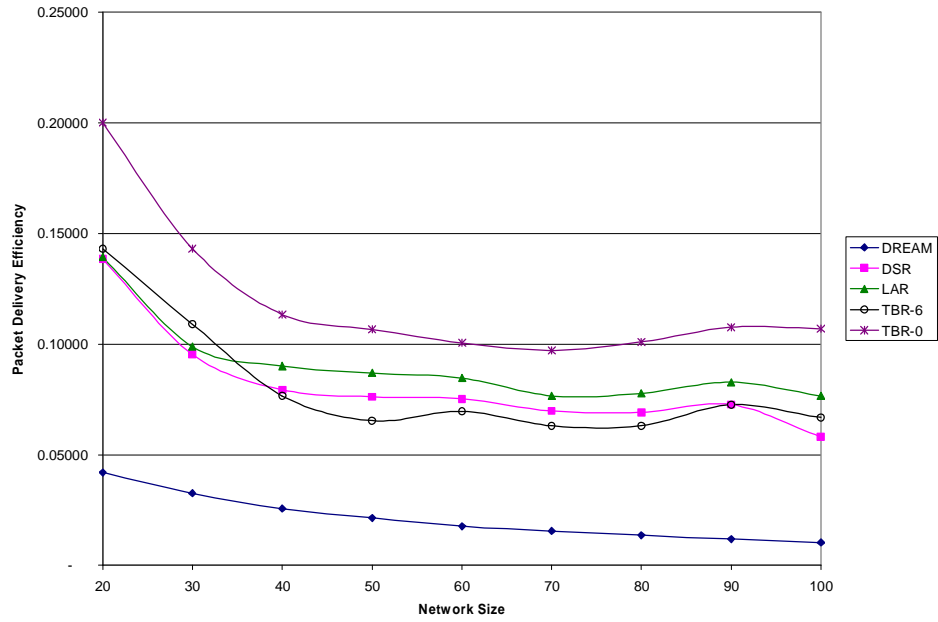


Figure 180: Packet Delivery Efficiency (Hilly, High Mobility, 2 pkts/sec)

9.4.3 Hilly Terrain, 3 Packets/Second

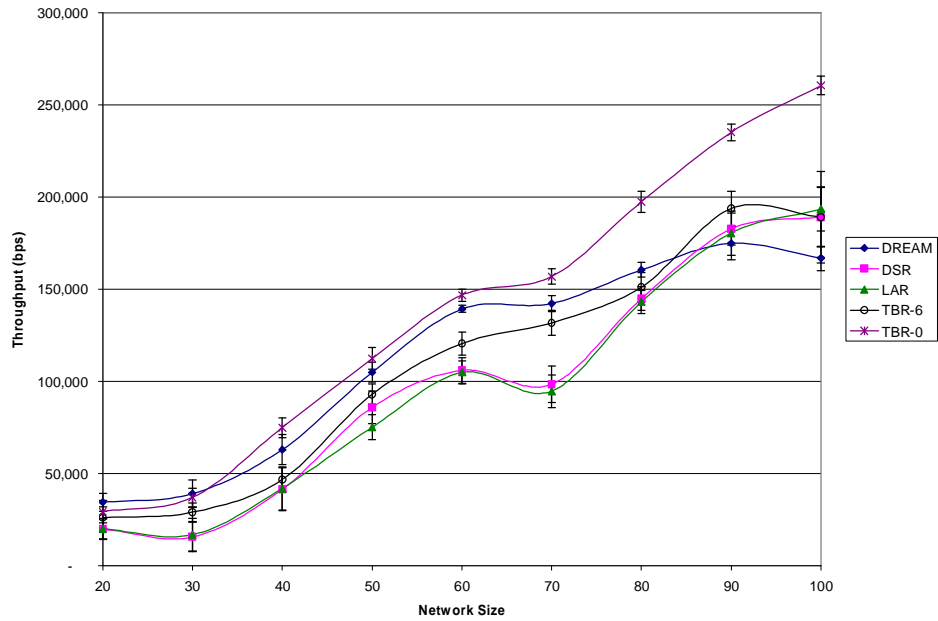


Figure 181: Throughput (Hilly, Low Mobility, 3 pkts/sec)

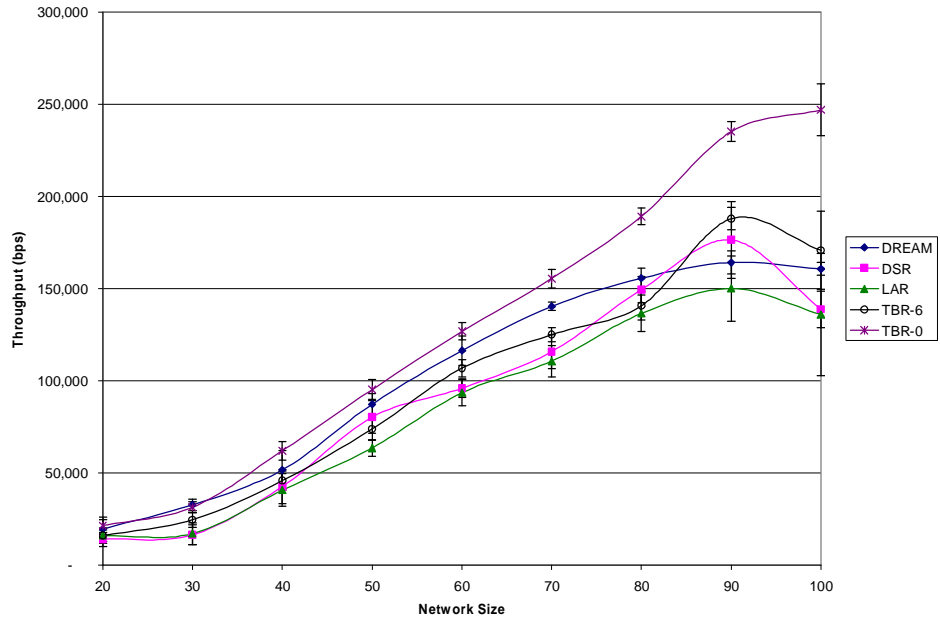


Figure 182: Throughput (Hilly, Medium Mobility, 3 pkts/sec)

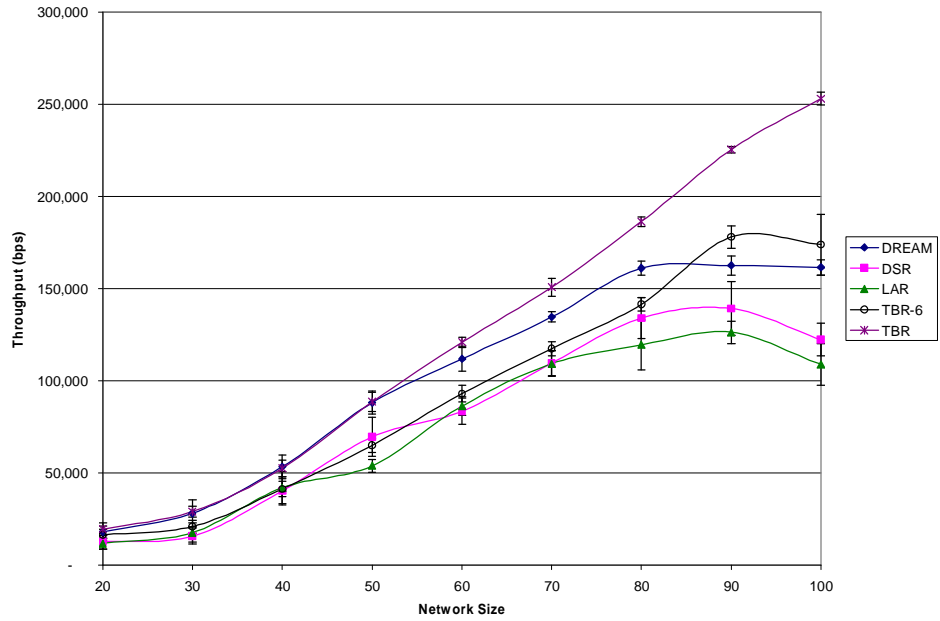


Figure 183: Throughput (Hilly, Medium Mobility, 3 pkts/sec)

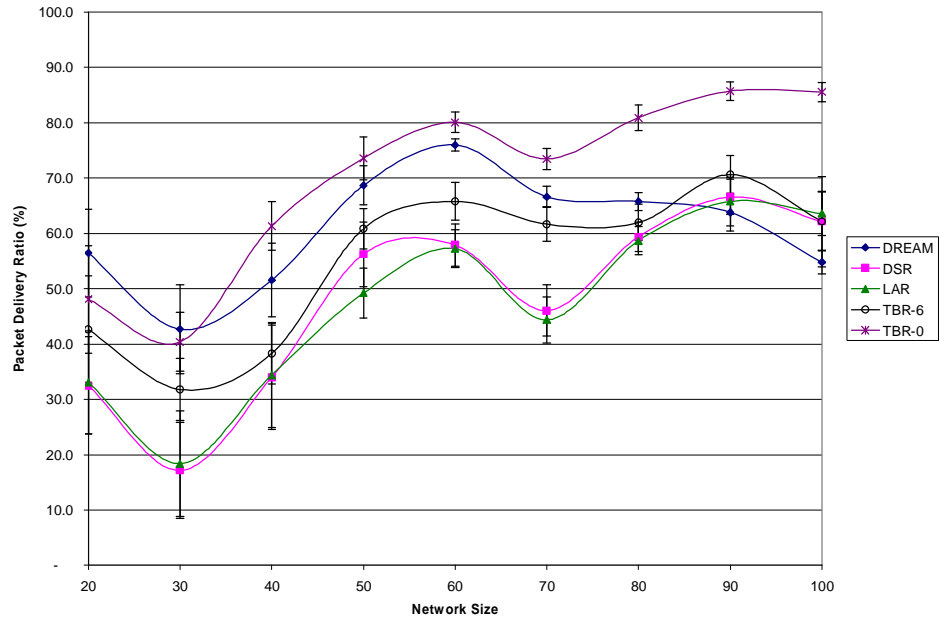


Figure 184: Packet Delivery Ratio (Hilly, Low Mobility, 3 pkts/sec)

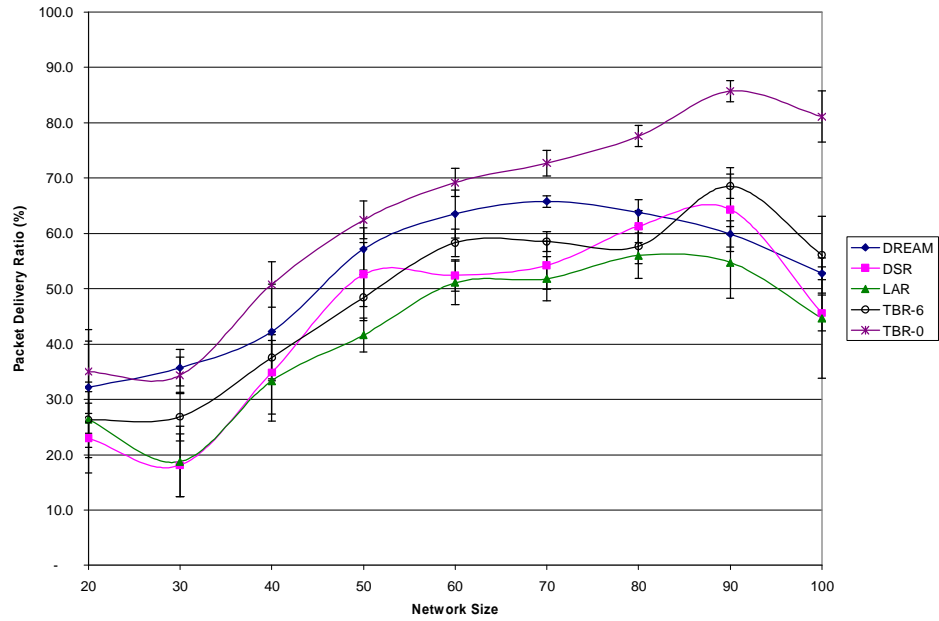


Figure 185: Packet Delivery Ratio (Hilly, Medium Mobility, 3 pkts/sec)

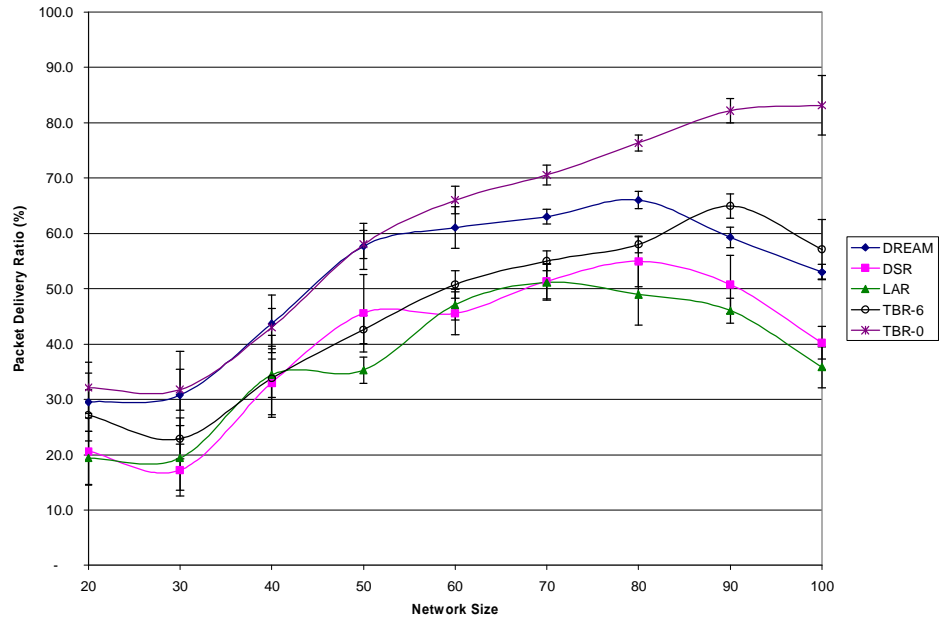


Figure 186: Packet Delivery Ratio (Hilly, High Mobility, 3 pkts/sec)

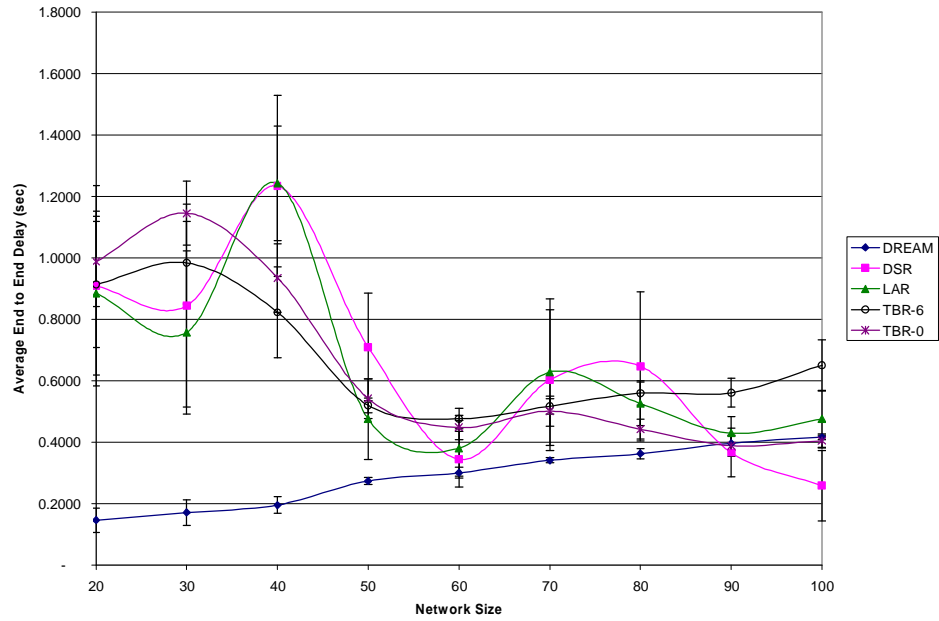


Figure 187: Average End to End Delay (Hilly, Low Mobility, 3 pkts/sec)

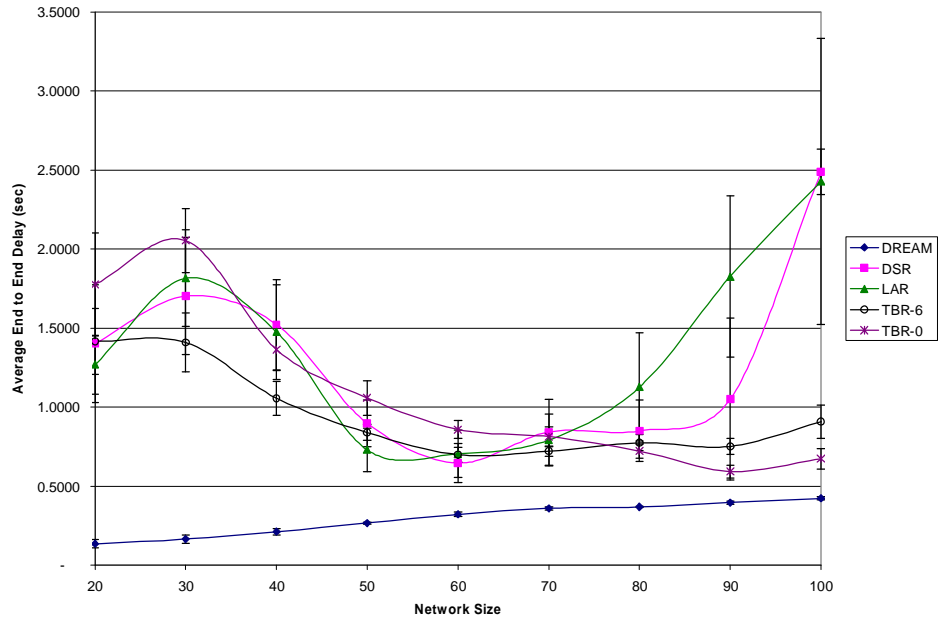


Figure 188: Average End to End Delay (Hilly, Medium Mobility, 3 pkts/sec)

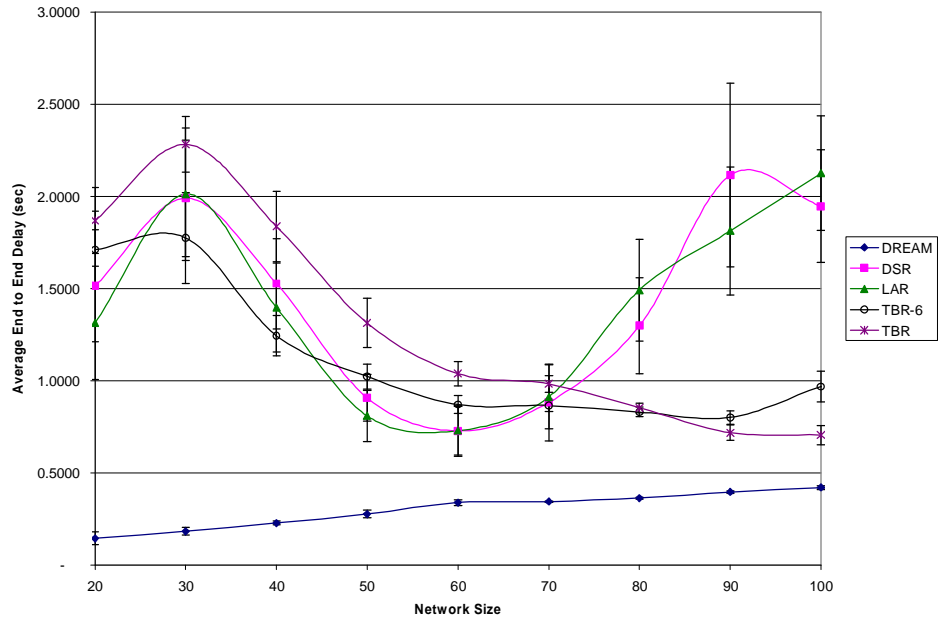


Figure 189: Average End to End Delay (Hilly, High Mobility, 3 pkts/sec)

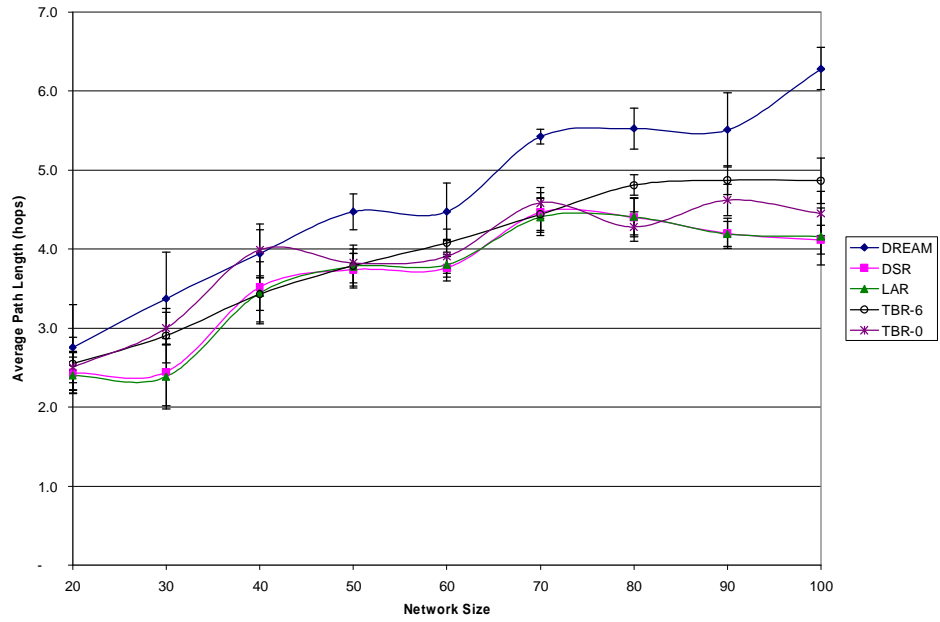


Figure 190: Average Path Length (Hilly, Low Mobility, 3 pkts/sec)

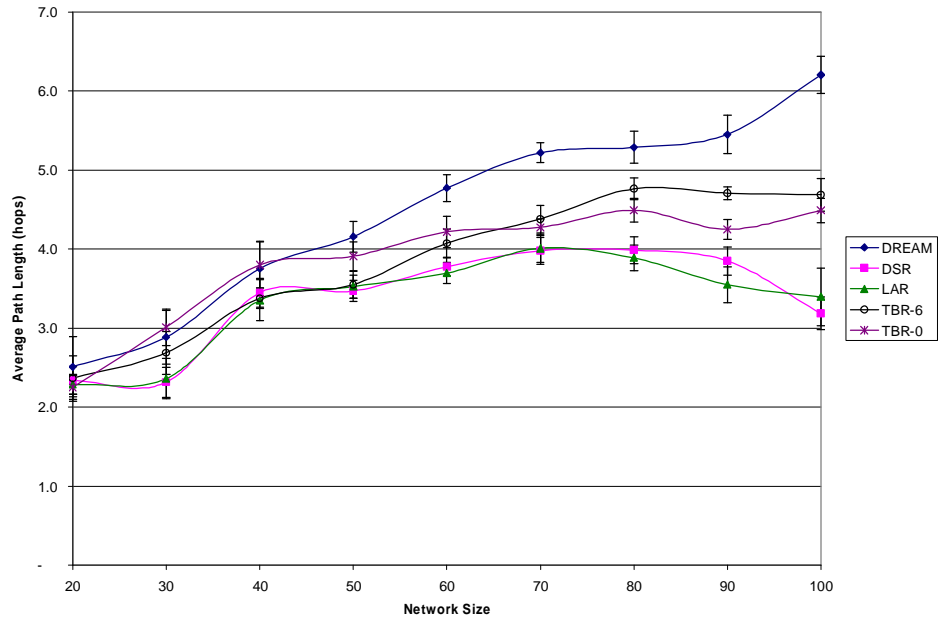


Figure 191: Average Path Length (Hilly, Medium Mobility, 3 pkts/sec)

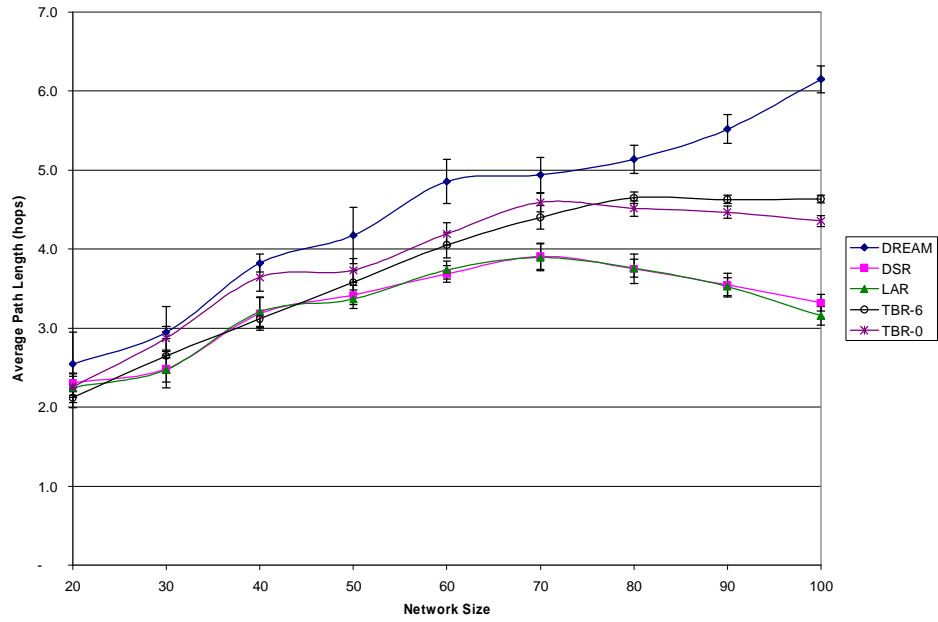


Figure 192: Average Path Length (Hilly, High Mobility, 3 pkts/sec)

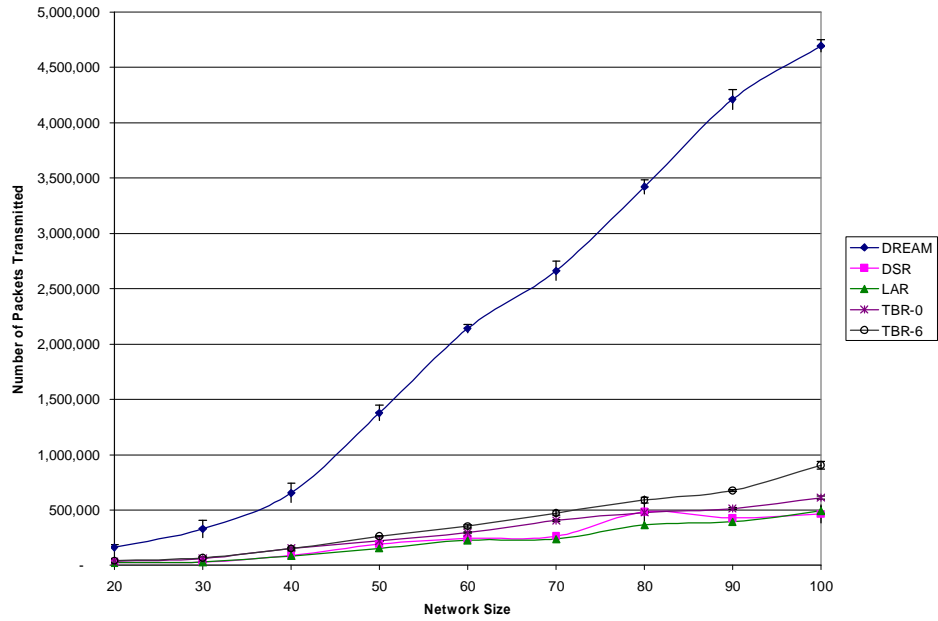


Figure 193: Number of Packets Transmitted (Hilly, Low Mobility, 3 pkts/sec)

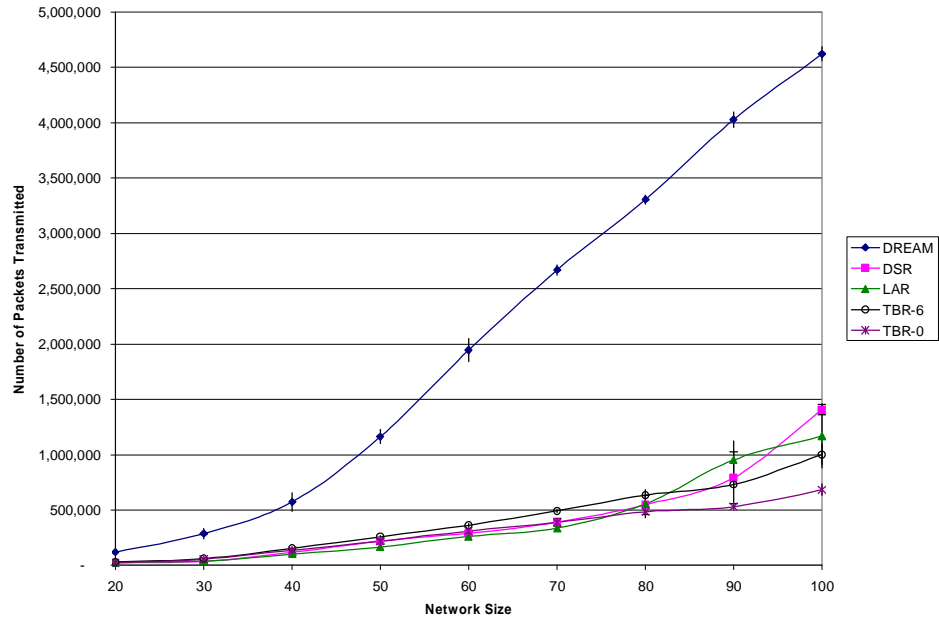


Figure 194: Number of Packets Transmitted (Hilly, Medium Mobility, 3 pkts/sec)

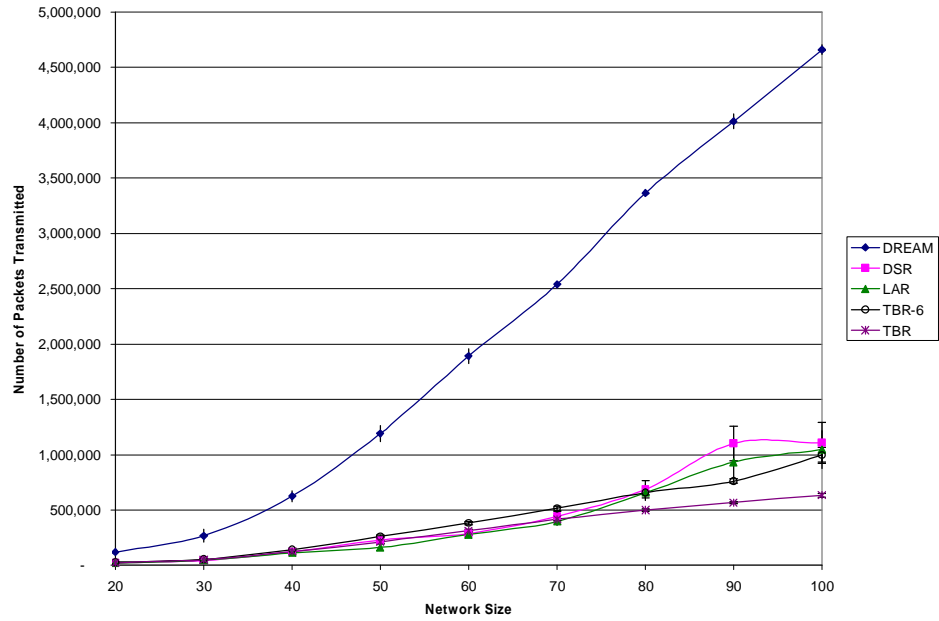


Figure 195: Number of Packets Transmitted (Hilly, High Mobility, 3 pkts/sec)

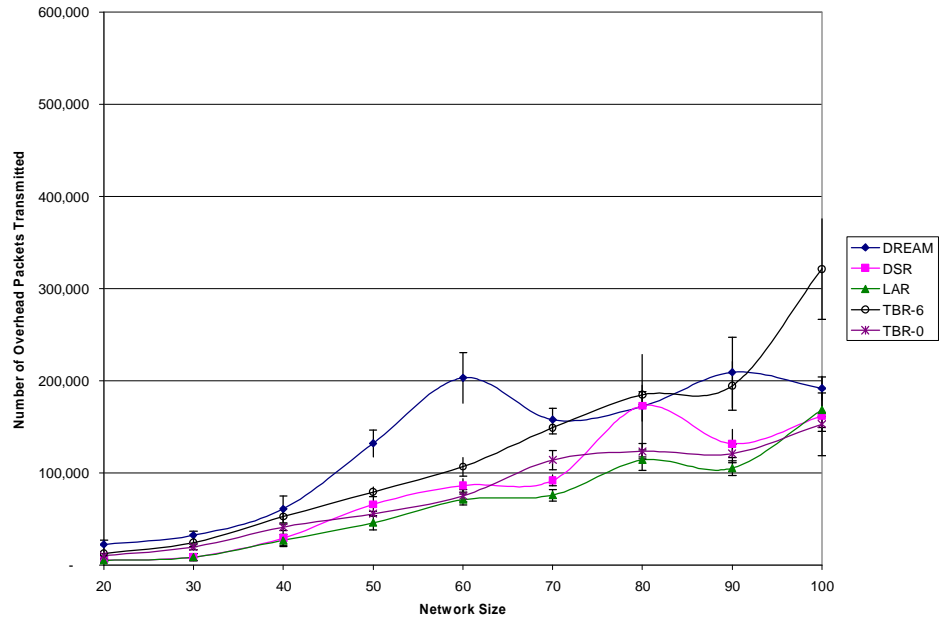


Figure 196: Number of Overhead Packets (Hilly, Low Mobility, 3 pkts/sec)

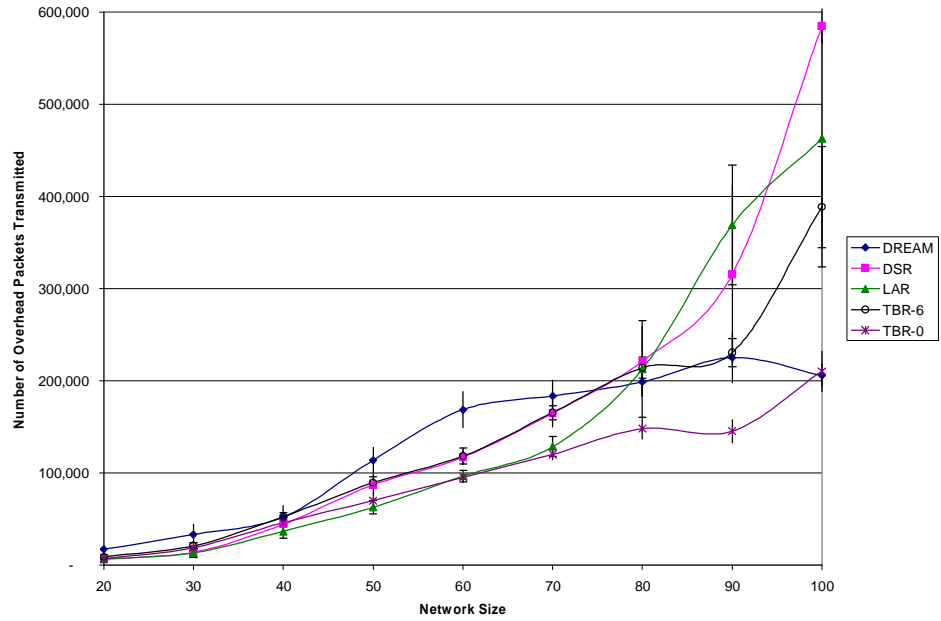


Figure 197: Number of Overhead Packets (Hilly, Medium Mobility, 3 pkts/sec)

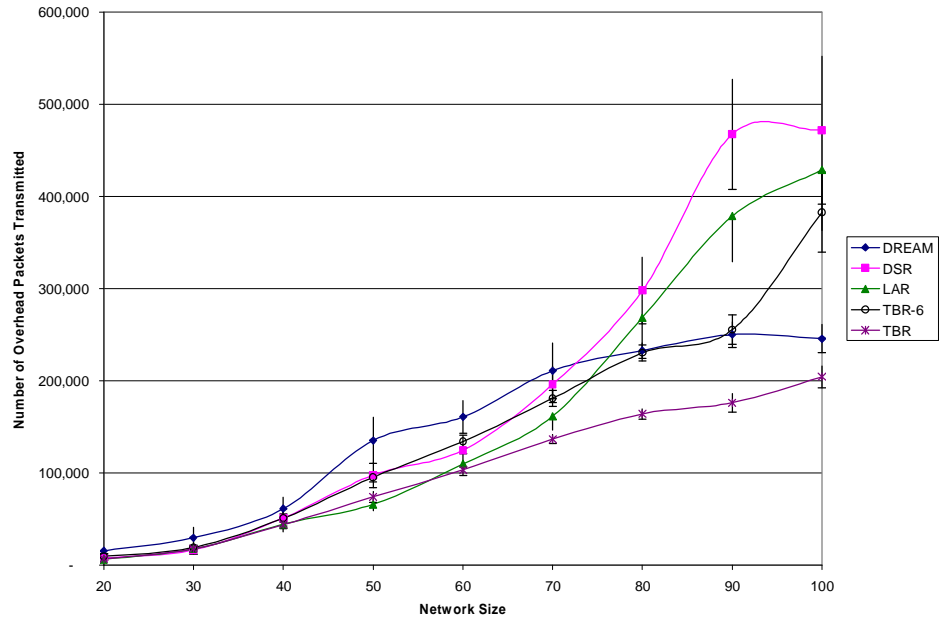


Figure 198: Number of Overhead Packets (Hilly, High Mobility, 3 pkts/sec)

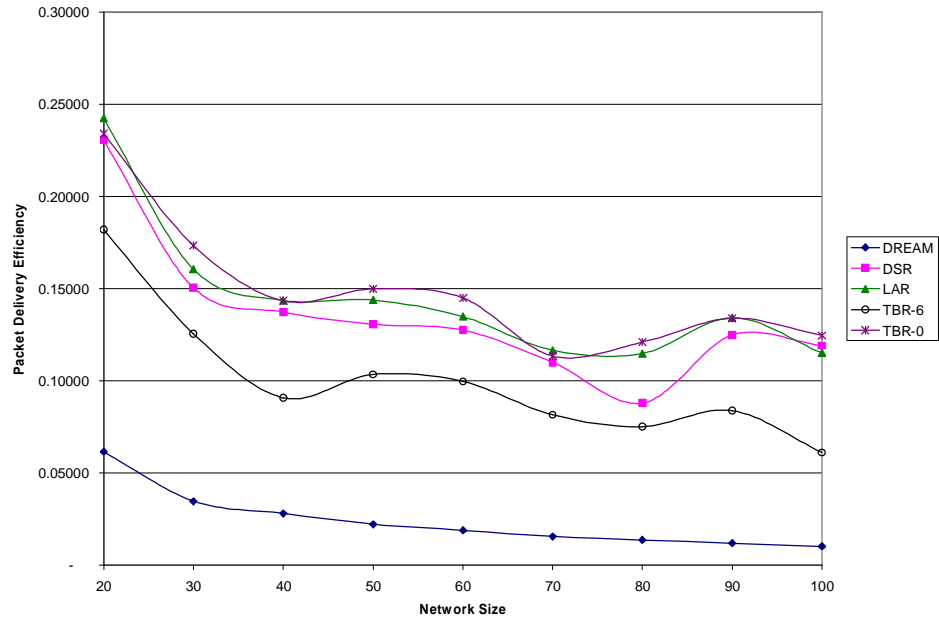


Figure 199: Packet Delivery Efficiency (Hilly, Low Mobility, 3 pkts/sec)

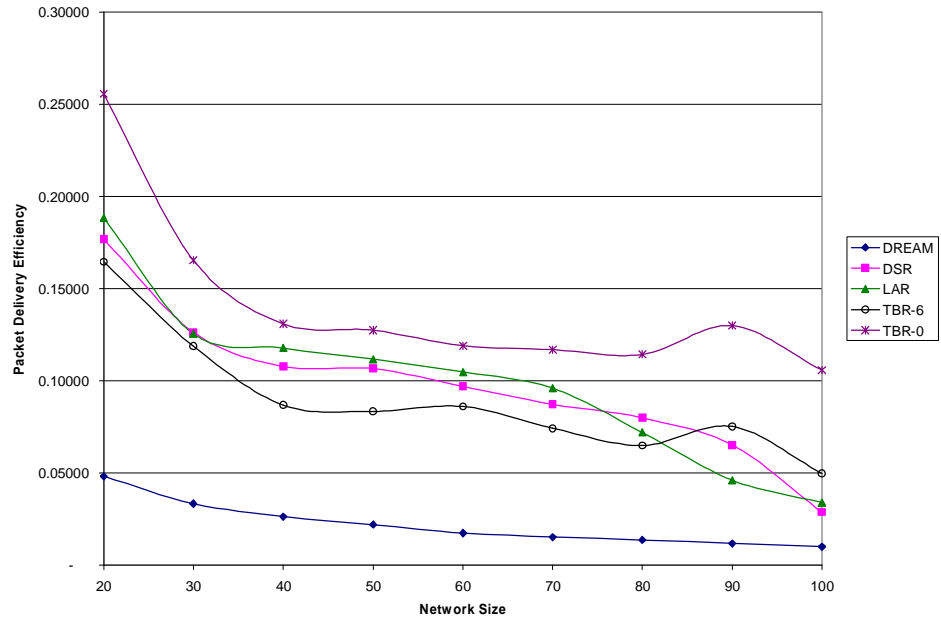


Figure 200: Packet Delivery Efficiency (Hilly, Medium Mobility, 3 pkts/sec)

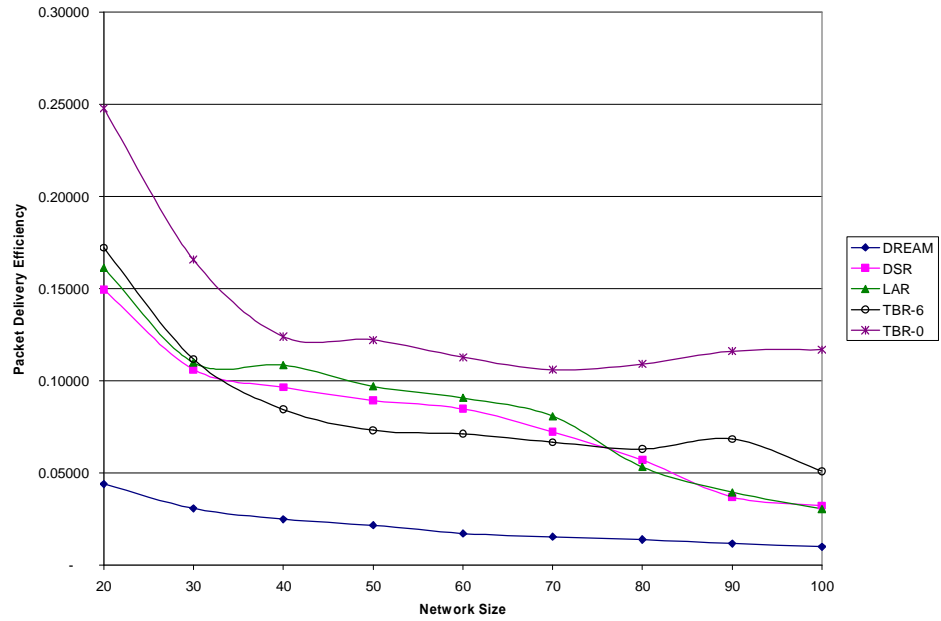


Figure 201: Packet Delivery Efficiency (Hilly, High Mobility, 3 pkts/sec)

9.4.4 Hilly Terrain, 4 Packets/Second

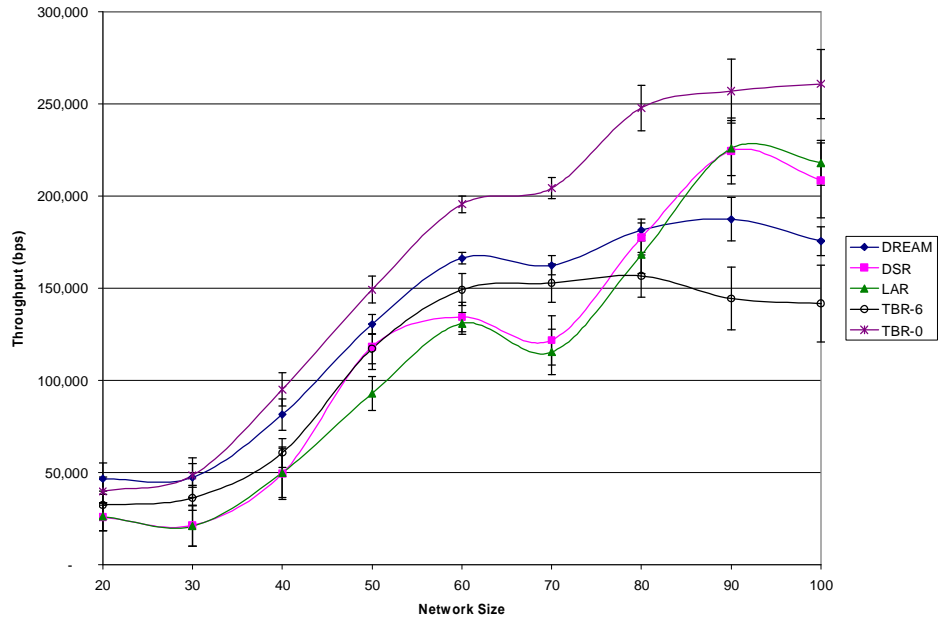


Figure 202: Throughput (Hilly, Low Mobility, 4 pkts/sec)

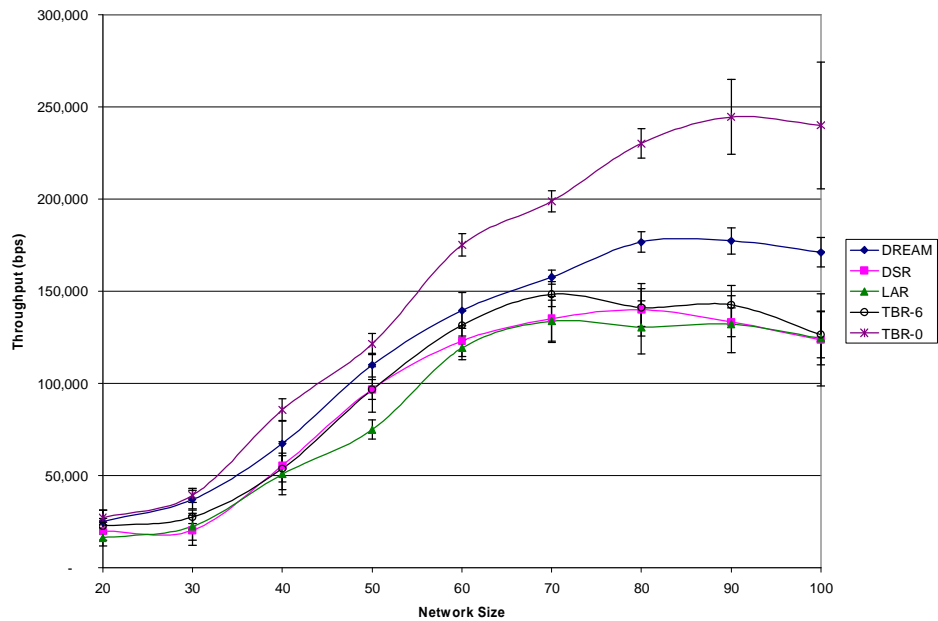


Figure 203: Throughput (Hilly, Medium Mobility, 4 pkts/sec)

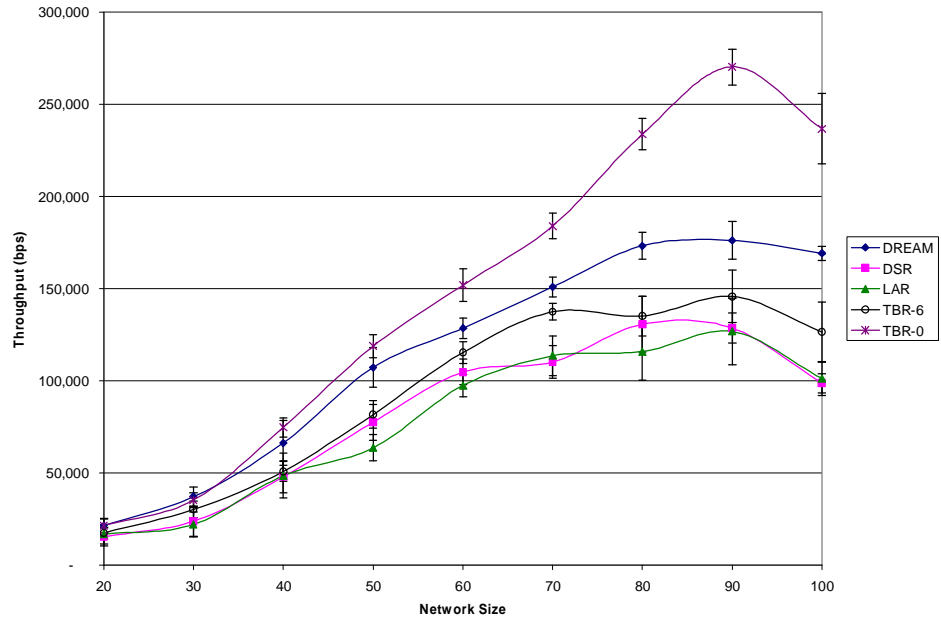


Figure 204: Throughput (Hilly, High Mobility, 4 pkts/sec)

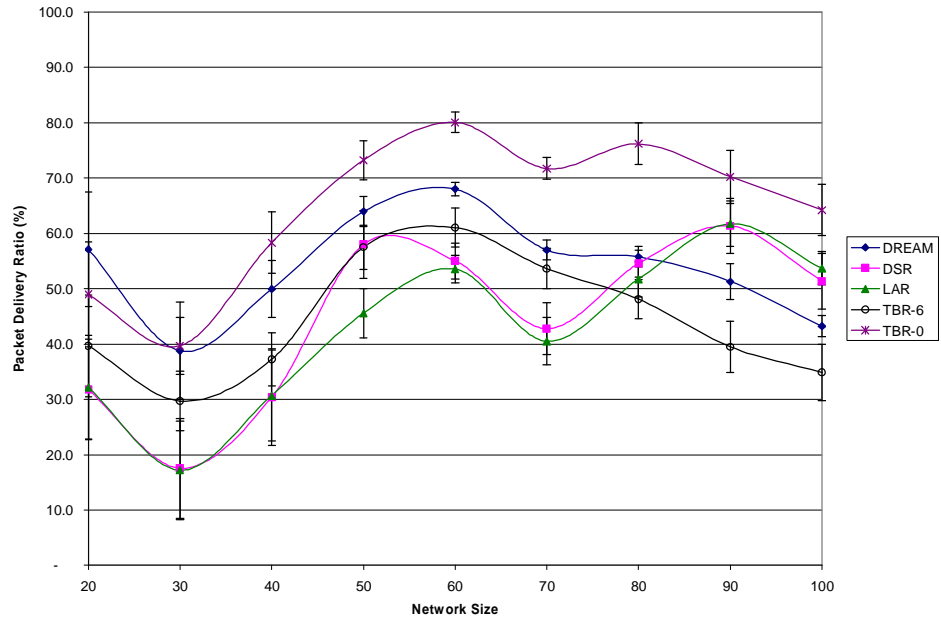


Figure 205: Packet Delivery Ratio (Hilly, Low Mobility, 4 pkts/sec)

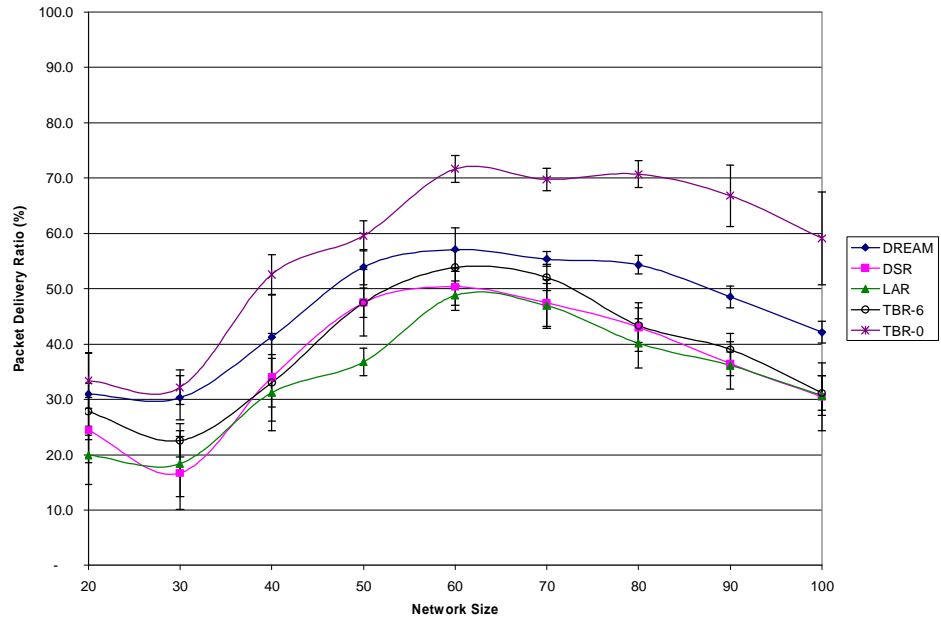


Figure 206: Packet Delivery Ratio (Hilly, Medium Mobility, 4 pkts/sec)

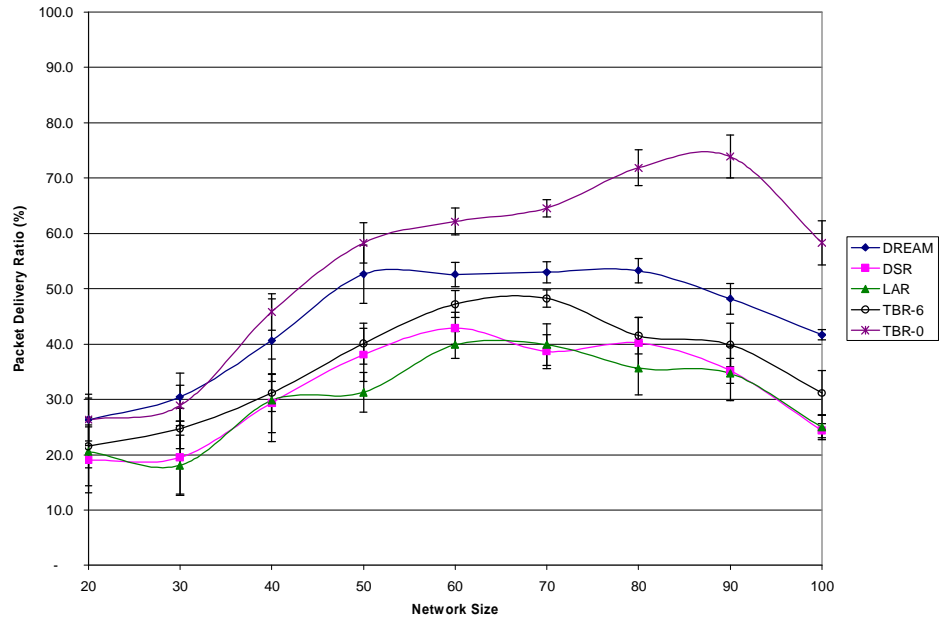


Figure 207: Packet Delivery Ratio (Hilly, High Mobility, 4 pkts/sec)

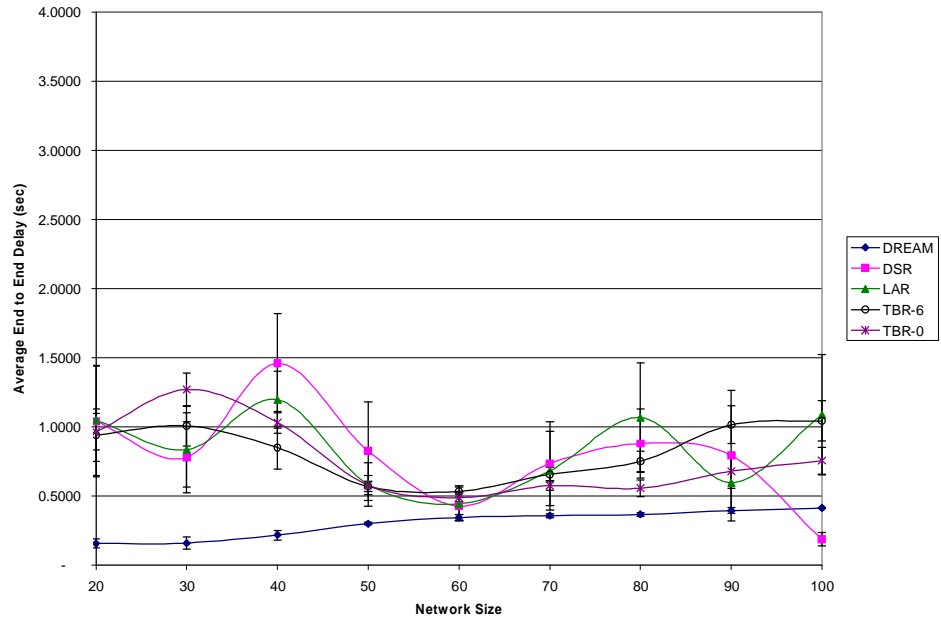


Figure 208: Average End to End Delay (Hilly, Low Mobility, 4 pkts/sec)

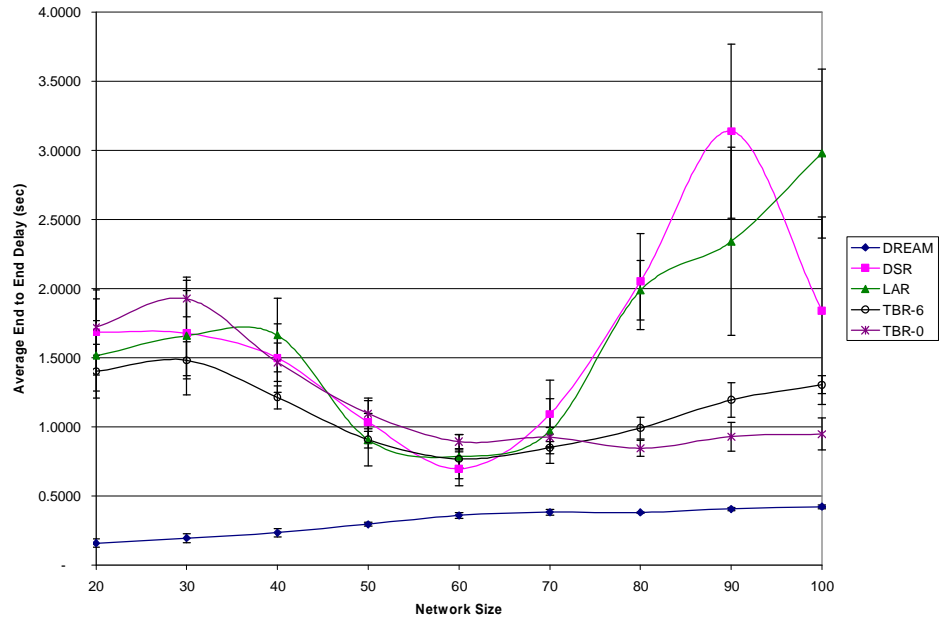


Figure 209: Average End to End Delay (Hilly, Medium Mobility, 4 pkts/sec)

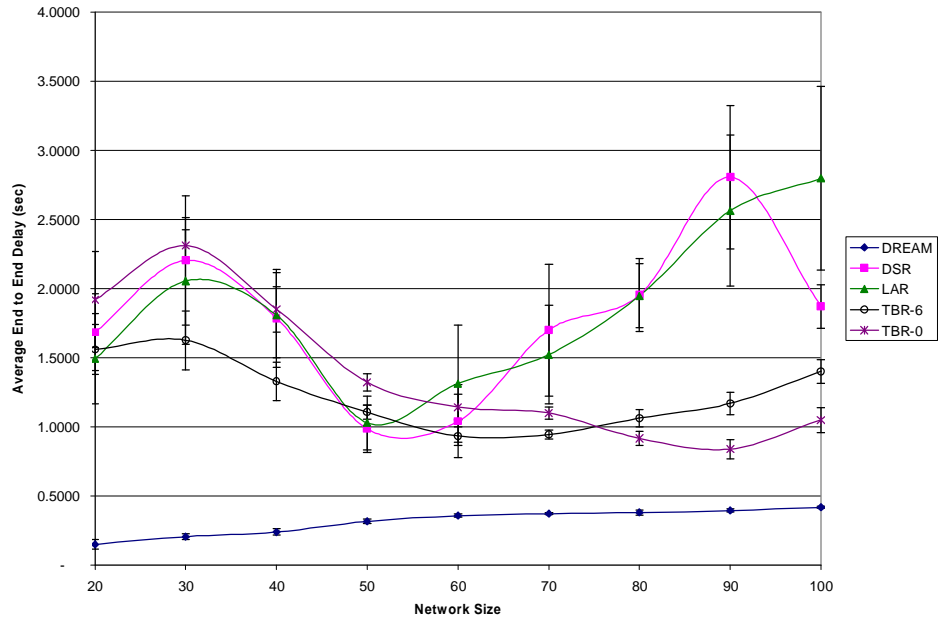


Figure 210: Average End to End Delay (Hilly, High Mobility, 4 pkts/sec)

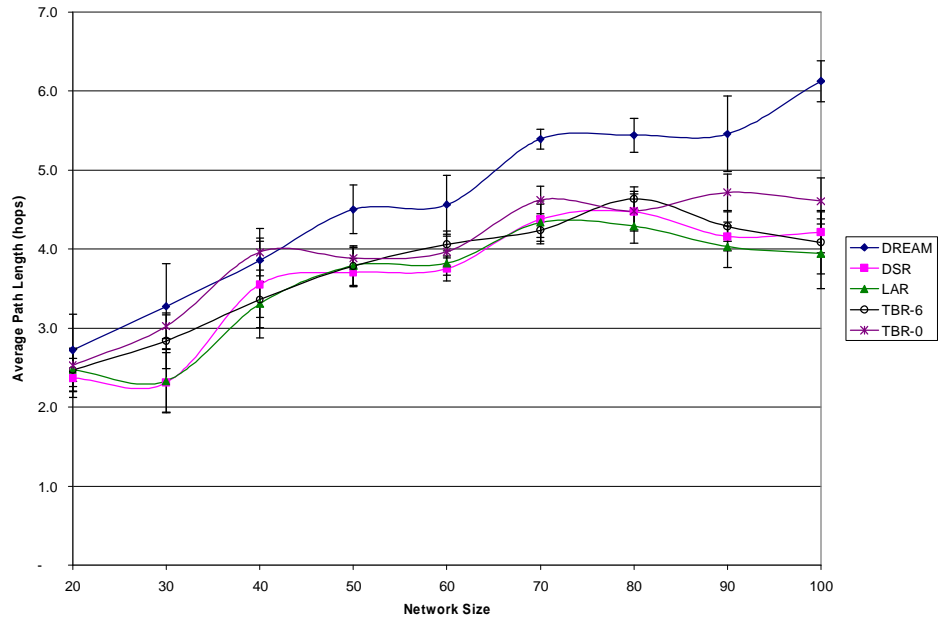


Figure 211: Average Path Length (Hilly, Low Mobility, 4 pkts/sec)

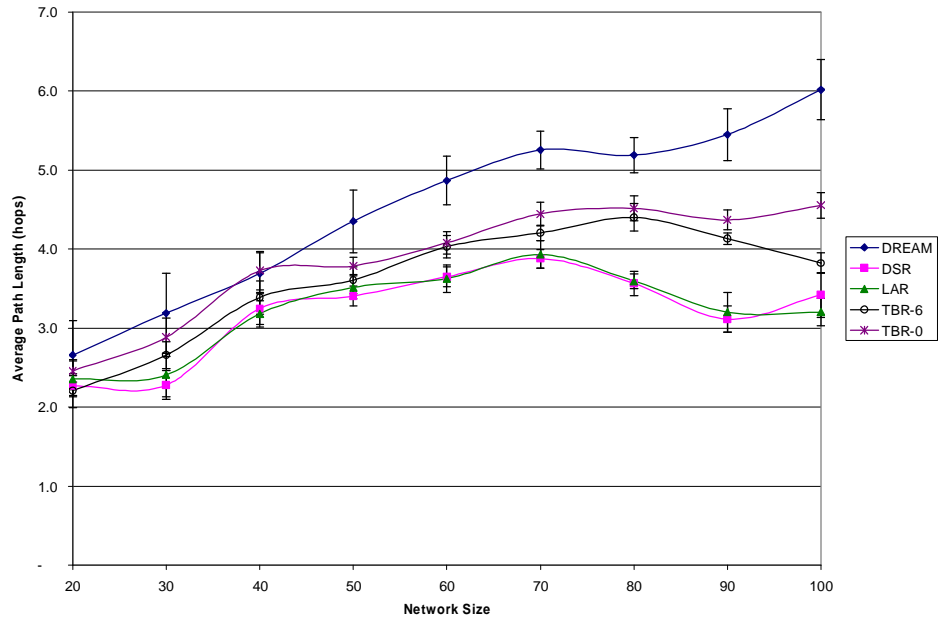


Figure 212: Average Path Length (Hilly, Medium Mobility, 4 pkts/sec)

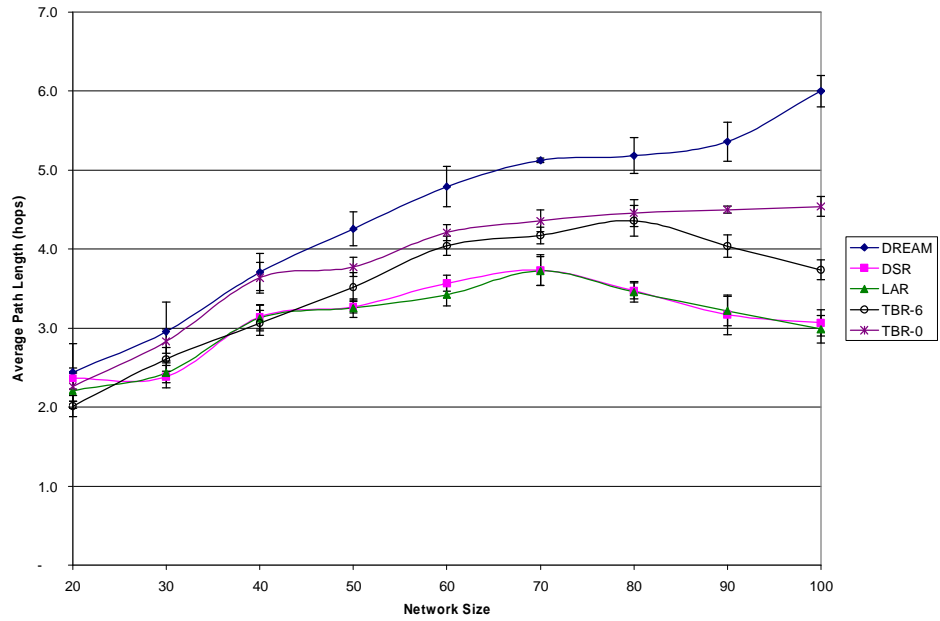


Figure 213: Average Path Length (Hilly, High Mobility, 4 pkts/sec)

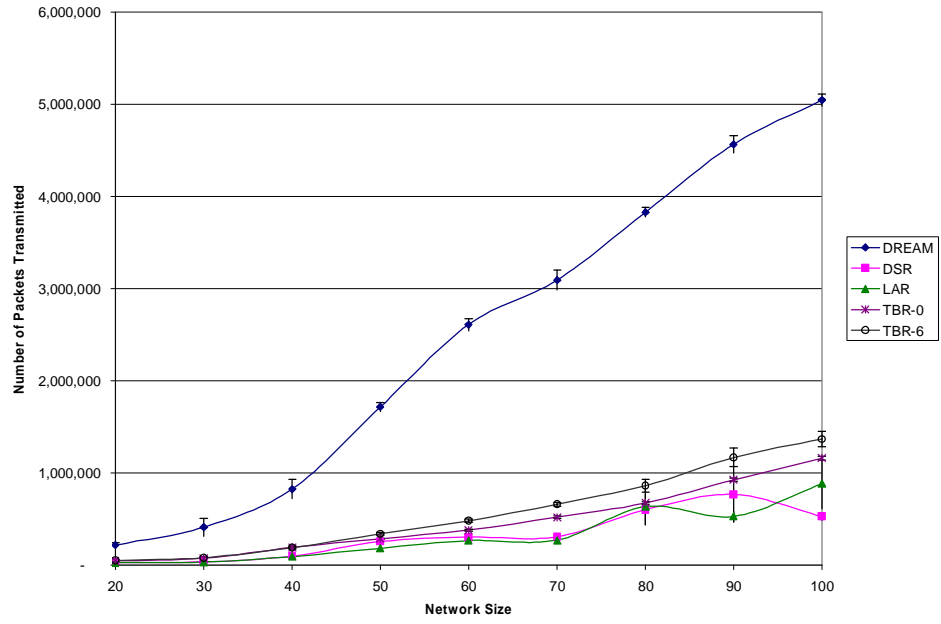


Figure 214: Number of Packets Transmitted (Hilly, Low Mobility, 4 pkts/sec)

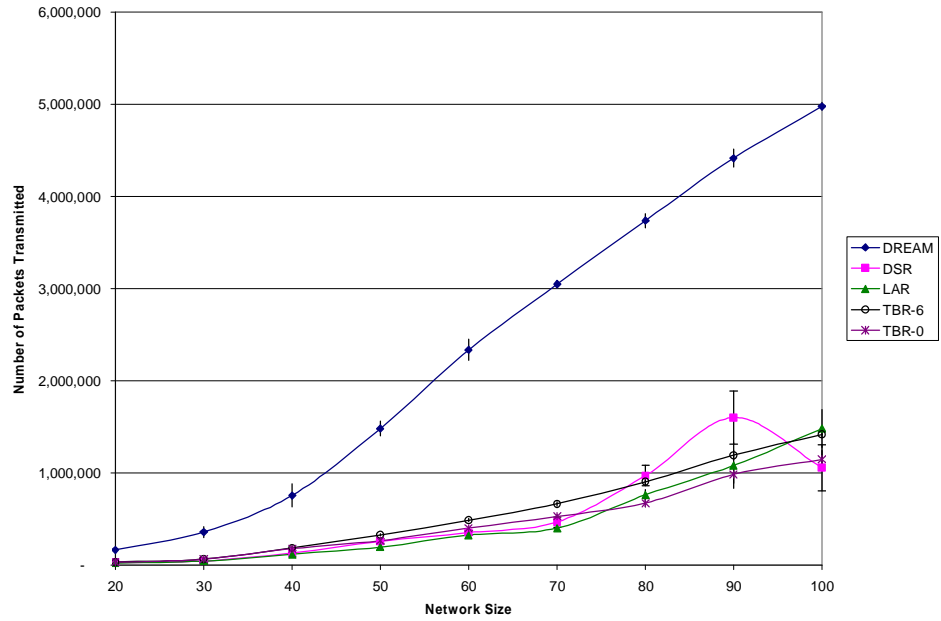


Figure 215: Number of Packets Transmitted (Hilly, Medium Mobility, 4 pkts/sec)

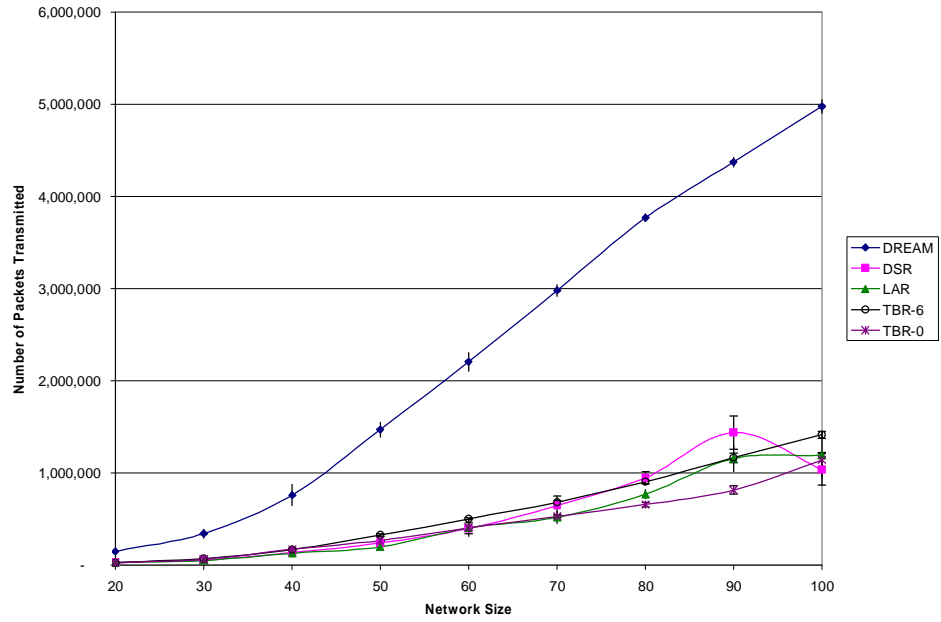


Figure 216: Number of Packets Transmitted (Hilly, High Mobility, 4 pkts/sec)

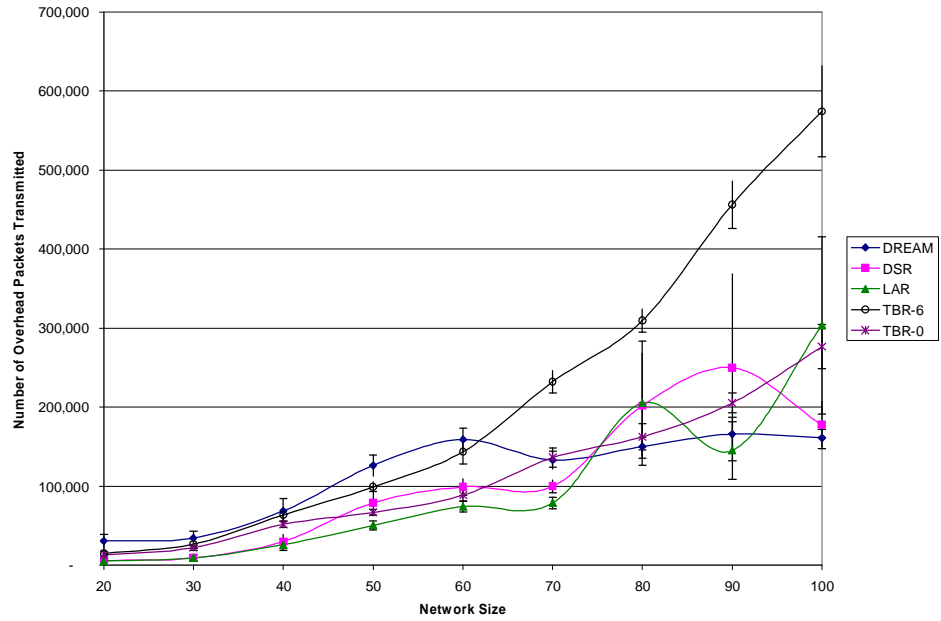


Figure 217: Number of Overhead Packets (Hilly, Low Mobility, 4 pkts/sec)

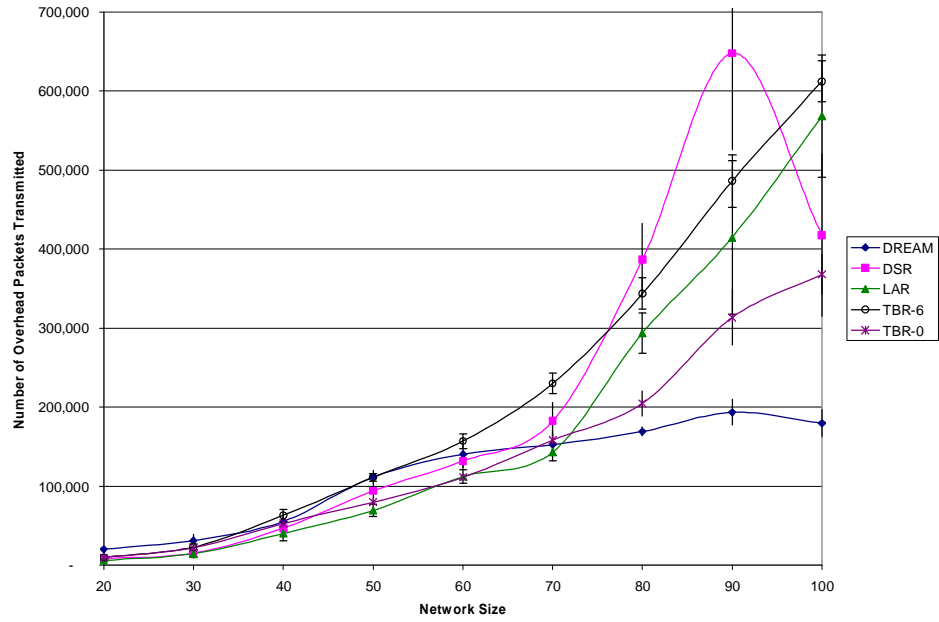


Figure 218: Number of Overhead Packets (Hilly, Medium Mobility, 4 pkts/sec)

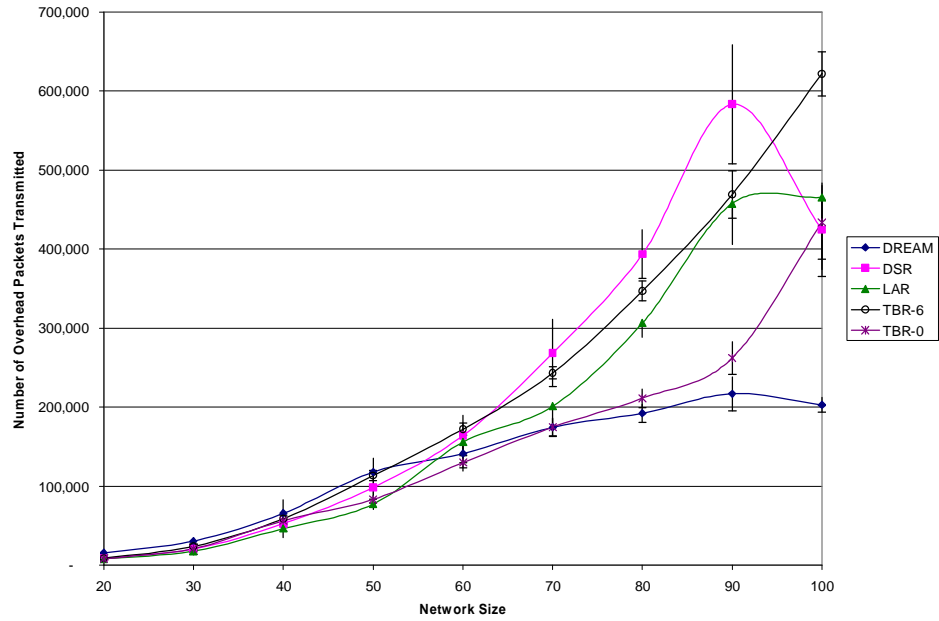


Figure 219: Number of Overhead Packets (Hilly, High Mobility, 4 pkts/sec)

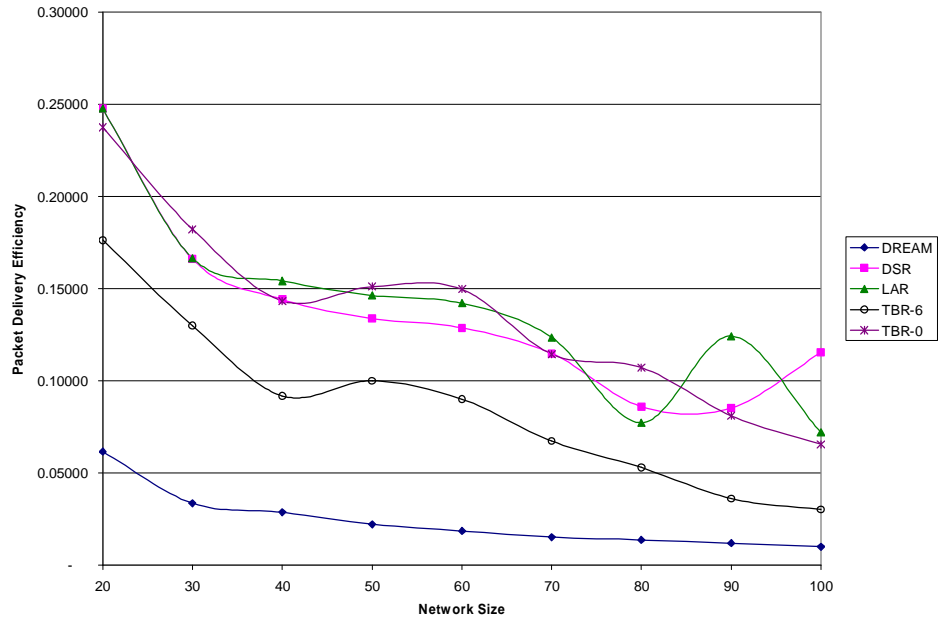


Figure 220: Packet Delivery Efficiency (Hilly, Low Mobility, 4 pkts/sec)

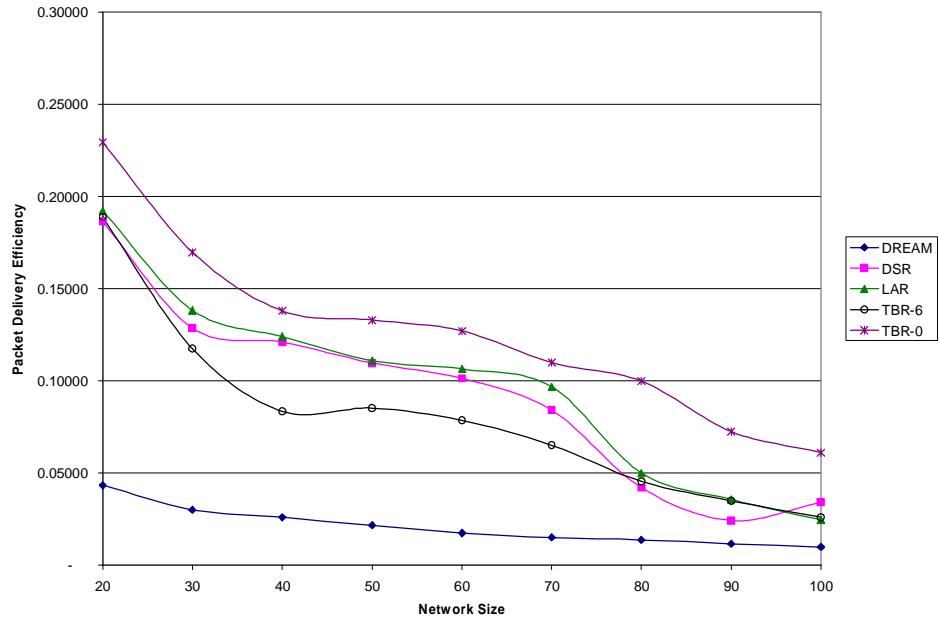


Figure 221: Packet Delivery Efficiency (Hilly, Medium Mobility, 4 pkts/sec)

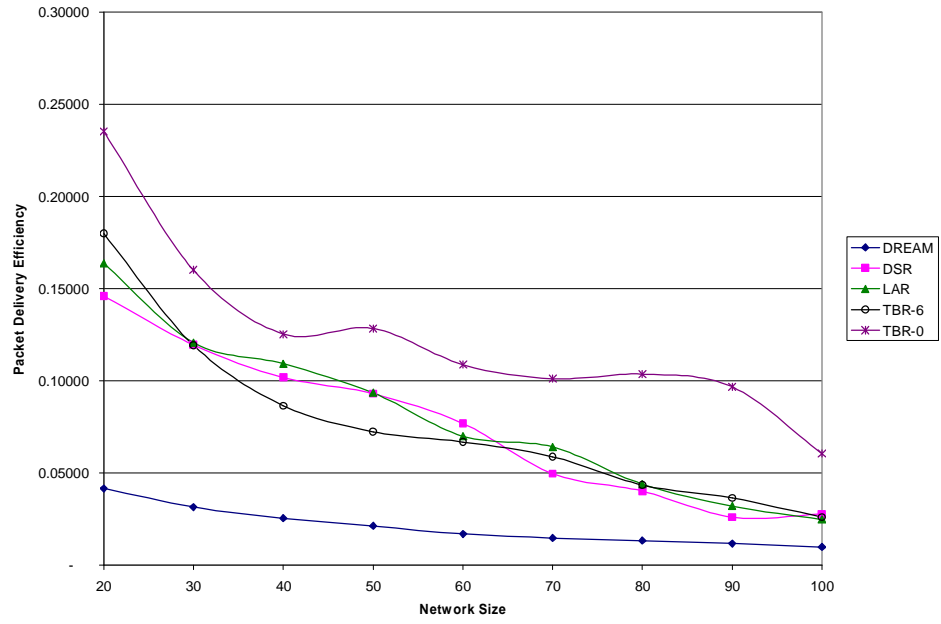


Figure 222: Packet Delivery Efficiency (Hilly, High Mobility, 4 pkts/sec)

REFERENCES

- [1] Z. Haas and S. Tabrizi, "On Some Challenges and Design Choices in Ad-Hoc Communications," *Proceedings of IEEE MILCOM'98*, Bedford, MA, October 18-21, 1998.
- [2] R. Ramanathan and J. Redi, "A Brief Overview of Ad Hoc Networks: Challenges and Directions," *IEEE Communications Magazine*, May 2002, pp20-22.
- [3] B. Leiner, D. Nielson, and F Tobagi, "Issues in Packet Radio Network Design," *Proceeding of IEEE*, Vol 75, No 1, January 1987.
- [4] J. Burbank, P. Chimento, B. Haberman & W. Kasch, "Key Challenges of Military Tactical Networking and the Elusive Promise of MANET Technology," *IEEE Communications Magazine*, November 2006, pp39-45.
- [5] C S Murthy and B.S. Manoj, *Ad Hoc Wireless Networks: Architectures and Protocols*, Prentice Hall 2004.
- [6] C-K. Toh, "Associativity Based Routing For Ad Hoc Mobile Networks," *Wireless Personal Communications Journal*, Vol. 4, No. 2, March 1997.
- [7] C. Perkins, E. Royer, "Ad Hoc On-Demand Distance Vector Routing," *Proceedings of the 2nd IEEE Workshop on Mobile Computing Systems and Applications*, New Orleans, LA, February 1999, pp90-100.
- [8] J. Broch, D. Johnson, and D. Maltz, "Dynamic Source Routing Algorithm for Mobile Ad-Hoc Networks," IETF Internet Draft, draft-ietf-manet-dsr-02.txt, June 1999. (work in progress)
- [9] V. Park and S. Corson, "A Highly Adaptive Distributed Routing Algorithm for Mobile Wireless Networks," *Proceedings of INFOCOM'97*, Kobe, Japan, April 1997.
- [10] P. Jacquet; P. Muhlethaler; T. Clausen, A. Laouiti; A. Qayyum, and L. Viennot, "Optimized Link State Routing Protocol for Ad Hoc Networks," *Proceedings IEEE International Multi Topic Conference 2001*, pp62 -68.

- [11] J.J. Garcia-Luna-Aceves and M. Spohn, "Source-Tree Routing in Wireless Networks," *Proceedings of the IEEE ICNP 99: 7th International Conference on Network Protocols*, Toronto, Canada, November, 1999.
- [12] B. Bellur; and R.G. Ogier, "A Reliable, Efficient Topology Broadcast Protocol For Dynamic Networks," *Proceedings of the Eighteenth Annual Joint Conference of the IEEE Computer and Communications Societies*, Volume: 1 , 1999 pp178-186.
- [13] A. Tanenbaum, *Computer Networks 3rd Edition*, Prentice Hall, 1996.
- [14] Z. Haas and M. Pearlman, "The Zone Routing Protocol for Highly Reconfigurable Ad Hoc Networks," *The Proceedings of ACM SIGCOMM '98*, September, 1998.
- [15] Z. Haas, "The Routing Algorithm for the Reconfigurable Wireless Networks," *ICUPC'97*, October 12-16, 1997.
- [16] D. Johnson, D. Maltz, "Dynamic Source Routing in Ad Hoc Networks," *Mobile Computing*, Kluwer Academic Publishers, 1996.
- [17] C. Perkins and P. Bhagwat, "Highly Dynamic Destination Sequenced Distance Vector Routing (DSDV) for Mobile Computers," *The Proceedings of ACM SIGCOMM '94*, October, 1994.
- [18] D. Maltz, J. Broch, J. Jetcheva and D. Johnson., "The Effects of On-Demand Behavior in Routing Protocols for Multihop Wireless Ad Hoc Networks," *IEEE Journal on Selected Areas in Communications*, vol. 17, no. 8, pp. 1439-1453, August 1999.
- [19] G. Dommety and R. Jain, "Potential networking applications of Global Positioning Systems (GPS)," Technical Report TR-24, CS Dept., Ohio State University, April 1996.
- [20] J. Navas and T. Imielinski, "Geographic Addressing and Routing", *Proceedings of the Third ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom'97)*, Budapest, Hungary, September 26-30 1997.
- [21] T. Imielinski and J. Navas, " GPS-Based Addressing and Routing", *Technical Report (LCSR-TR-262)*, Rutgers University Computer Science, March 7, 1996 (updated August 9, 1996).

- [22] M Mauve and J Widmer, "A Survey on Position-Based Routing in Mobile Ad Hoc Networks", *IEEE Network Magazine*, November/December 2001, pages 30-39.
- [23] I. Stojmenovic, "Position-Based Routing in Ad hoc Networks," *IEEE Communications Magazine*, Vol. 40 No. 7, July 2002, pp128-134.
- [24] S Giordano, I Stojmenovic, L. Blazevic, "Position Based Routing Algorithms for Ad Hoc Networks: A Taxmony," in: '*Ad Hoc Wireless Networking*', X. Cheng, X. Huang and D.Z. Du (eds.), Kluwer, 2003.
- [25] S. Basiagni, I. Chlamtac, V. Syrotiuk and B. Woodward, "A Distance Routing Effect Algorithm for Mobility (DREAM)," *The Proceedings of MOBICOM 98*, Dallas, TX, 1998.
- [26] B. Karp, "Geographic Routing in Wireless Networks", doctoral dissertation, October 2000.
- [27] B Karp & H.T Kung, "GPSR: Greedy Perimeter Stateless Routing for Wireless Networks," *Proceedings of the 6th Annual International Conference on Mobile Computing and Networking (MOBICOM 2000)*, 2000.
- [28] G. Finn, "Routing and Addressing Problems in Large Metropolitan-Scale Internetworks," Technical Report ISI/RR-87-180, Information Sciences Institute, March, 1987.
- [29] S-C Woo and S Singh, "Scalable Routing in Ad Hoc Networks," Technical Report TR00.001, Oregon State University, March 2000.
- [30] T-C Hou and V. Li, "Transmission Range Control in Multihop Packet Radio Networks," *IEEE Transactions on Communications*, Vol. COM-34, No. 1, January 1986.
- [31] Y-B Ko and N Vaidya, "Using Location Information to Improve Routing in Ad Hoc Networks," Technical Report 97-013, CS Dept., Texas A&M University, December 1997.
- [32] Y-B Ko and N Vaidya, "Location Aided Routing (LAR) in Mobile Ad Hoc Networks," Technical Report 98-012, CS Dept., Texas A&M University, June 1998.

- [33] Y-B. Ko and N. Vaidya, "Location-aided Routing (LAR) in Mobile Ad Hoc Networks," *Proceedings of the 4th Annual International Conference on Mobile Computing and Networking (MOBICOM 98)*, 1998.
- [34] R. Dube and C. Rais and K. Wang and S. Tripathi, "Signal stability based adaptive routing (SSA) for ad hoc mobile networks" *IEEE Personal Communication*, 1997.
- [35] A. Kershenbaum, *Telecommunications Network Design Algorithms*, McGraw Hill, 1993, pp157-158.
- [36] R Sciandra, "TIREM/SEM Programmer's Reference Manual", JSC Consulting Report ECAC-CR-90-039, Electromagnetic Compatibility Analysis Center, July 1990.
- [37] D Eppink and W Kuebler, "TIREM/SEM Handbook", JSC Handbook ECAC-HDBK-93-076, Electromagnetic Compatibility Analysis Center, March 1994.
- [38] <http://standards.ieee.org/getieee802/802.11.html> (assessed November 2005).
- [39] S Shakkottai, T Rappaport and P Karlsson, "On Cross-Layer Design of Wireless Networks", *IEEE Communications Magazine*, October 2003, pp74-80.
- [40] U. Kozat, I. Koutsopoulos and L. Tassiulas, "Cross-layer Design for Wireless Networks," *Proceedings of IEEE INFOCOM 2004*, March 2004,
- [41] S-J. Lee, M. Gerla, and C-K. Toh, "A Simulation Study of Table-Driven and On-Demand Routing Protocols for Mobile Ad Hoc Networks," *IEEE Network*, July 1999.
- [42] E. Royer, C-K Toh, "A Review of Current Routing Protocols for Ad Hoc Mobile Wireless Networks." *IEEE Personal Communications*, April 1999, pp46-55.
- [43] S. Das, C. Perkins, and E. Royer, "Performance Comparison of Two On-demand Routing Protocols for Ad Hoc Networks," *Proceedings of INFOCOM '00*, March 2000.
- [44] T. Camp, et al, "Performance Comparison of Two Location Based Routing Protocols," *Proceedings of IEEE INFOCOM 2002*, June 2002, pp1678-1687.

- [45] http://w3.antd.nist.gov/wctg/prd_dsrfiles.html (Accessed October 2006).
- [46] http://www.opnet.com/support/cont_models.html (Accessed June 2004).
- [47] C. Bettstetter; G. Resta; and P. Santi, "The Node Distribution Of The Random Waypoint Mobility Model For Wireless Ad Hoc Networks," *IEEE Transactions on Mobile Computing*, Vol 2, Issue 3, Jul-Sep 2003,pp257-269.
- [48] T. Camp, J. Boleng, and V. Davies, "A Survey of Mobility Models for Ad Hoc Network Research," *Wireless Communication & Mobile Computing (WCMC): Special Issue on Mobile Ad Hoc Networking Research, Trends and Applications*, Vol. 2, No. 5, 2002, pp. 483-502.
- [49] W. Su, S. Lee, and M. Gerla, "Mobility prediction and routing in ad hoc wireless networks," *International Journal of Network Management*, vol. 11, no. 1, Jan. 2001, pp. 3–30.
- [50] P Pathirana, A Savkin and S Jha, "Robust extended Kalman filter applied to location tracking and trajectory prediction for PCS networks ,"*Proceedings of the 2004 IEEE International Conference on Control Applications*, 2-4 Sept. 2004, Volume: 1, pp63- 68.
- [51] T. Liu, P. Bahl and I. Chlamtac, "Mobility Modeling, Location Tracking, and Trajectory Prediction in Wireless ATM Networks," *IEEE Journal on Selected Areas of Communications*, Vol 16, No 6, August 1998, pp922-935.
- [52] J. Chan and A. Seneviratne, "A Practical User Mobility Prediction Algorithm for Supporting Adaptive QoS in Wireless Networks," *Proceedings of IEEE International Conference on Networks*, Brisbane, Australia, Sep. 1999.
- [53] A. Jardosh, E. M. Belding-Royer, K. C. Almeroth, and S. Suri, "Towards Realistic Mobility Models for Mobile Ad Hoc Networks," *ACM Mobicom 2003*, San Diego, CA, September 2003
- [54] F. Tobagi and L. Klienrock, "Packet Switching in Radio Channels: Part II--The Hidden Terminal Problem in Carrier Sense Multiple-Access and the Busy-Tone Solution" *IEEE Transactions on Communications*, Vol 23, Issue 12, Dec 1975, pp1417-1433.

- [55] A. Jayasuriya, S. Perreau, A. Dadej, S. Gordon, "A Hidden vs. Exposed Terminal Problem in Ad hoc Networks," *Proceedings of the Australian Telecommunication Networks and Applications Conference*, Sydney, Australia, 8-10 Dec 2004.