

# Investigation into the optimization of low speed communication protocols for narrow band networks

by

Teraan van Staden

*Thesis presented in partial fulfilment of the requirements for the degree  
Master of Science in Electronic Engineering in the Faculty of Engineering at  
Stellenbosch University*



Department of Electrical and Electronic Engineering,  
University of Stellenbosch,  
Private Bag X1, 7602 Matieland, South Africa

Supervisor: Dr. Riaan Wolhuter

December 2012

# Declaration

By submitting this thesis electronically, I declare that the entirety of the work contained therein is my own, original work, that I am the sole author thereof (save to the extent explicitly otherwise stated), that reproduction and publication thereof by Stellenbosch University will not infringe any third party rights and that I have not previously in its entirety or in part submitted it for obtaining any qualification.

Date: December 2012.....

Copyright © 2012 Stellenbosch University  
All rights reserved.

# Abstract

## **Investigation into the optimization of low speed communication protocols for narrow band networks**

T. van Staden

*Department of Electrical and Electronic Engineering,  
University of Stellenbosch,  
Private Bag XI, 7602 Matieland, South Africa*

Thesis: MScEng (Electronic)

December 2012

In this thesis, the investigation into the optimization of low speed communication protocols for narrow band networks will be presented. The main focus will be on analysing commonly used low speed communication protocols and investigate an alternative to these protocols to provide an optimized low speed narrow band network that provides better performance at high and low channel utilization. A study of existing low speed communication networks within the field of water supply has been used to identify the most appropriate protocols to be included in the analysis. The analysis of each protocol discussed includes the development of a simulation and theoretical model, with parameters based on those obtained from implemented communication networks focusing on the parameters used within the Namib water supply scheme of NamWater in Namibia.

Of the currently implemented contention protocols, the non-persistent Carrier Sense Multiple Access (CSMA) protocol is implemented the most. Current models used for modelling these protocols make use of various assumptions. These models have been expanded to provide a more accurate representation of the non-persistent CSMA model. The Round Robin Polling (RRP) protocol is another well known protocol used within the telemetry industry and has also been modelled as an alternative to the non-persistent CSMA model.

The Adaptive Tree Walk (ATW) protocol has been identified as the limited contention protocol to be modelled as a possible alternative to the conventional methods used. A new model has been developed for modelling this protocol by making use of the same strategies and tools used in the modelling of the non-persistent CSMA and RRP protocols.

The Simulation modelling has been developed by making use of DESMO-J, an Object Orientated Simulation API based in Java, developed by the Faculty of Informatics at the University of Hamburg. DESMO-J has been chosen as an alternative to the more traditional simulation languages due to its complete documentation, support structures, ease of use and flexibility. All theoretical models have been implemented in Matlab.

# Uittreksel

## Ondersoek na optimering van laespoed protokolle vir kommunikasie deur nouband netwerke

T. van Staden

*Departement Elektries en Elektroniese Ingenieurswese,  
Universiteit van Stellenbosch,  
Privaatsak XI, Matieland 7602, Suid Afrika.*

Tesis: MScIng(Elektronies)

Desember 2012

In hierdie tesis sal die ondersoek na die optimering van laespoed protokolle vir kommunikasie oor nouband netwerke voorgelê word. Die hoof fokus is op die analise van algemene laespoed kommunikasie protokolle en die ondersoek van alternatiewe wat 'n meer optimale laespoed nouband netwerk sal lewer deur beter werkverigting by lae en hoë kanaal-verkeer. 'n Studie van praktiese laespoed nouband netwerke in die veld van waterspreiding word gebruik om die mees algemene protokolle te identifiseer wat in die analise ingesluit moet word. Die analise van die protokolle sluit in teoretiese en simulasiemodelle, met parameters soos geïdentifiseer uit die studie van 'n praktiese netwerk, naamlik die Namib waterspreidingskema van NamWater in Namibië. Die 'Non-persistent Carrier Sense Multiple Access' en 'Round Robin Polling' protokolle is geïdentifiseer as dié wat meeste geïmplementeer word. Die werkverigting van die protokolle is geanaliseer deur gebruik te maak van teoretiese en simulasiemodeleringstegnieke. Huidige modelle van die CSMA protokol is gebaseer op sekere aannames. Hierdie aannames word aangepas en verbeter vir implementering van die teoretiese model. Die model word ook verder uitgebrei om beter resultate te lewer oor 'n groter parameterstel. Die 'Adaptive Tree Walk' protokol is geïdentifiseer as 'n moontlike optimale protokol en word gemodelleer en vergelyk teen die CSMA en RRP protokolle se werkverigting. Die simulasiemodelle is ontwikkel deur gebruik te maak van die DESMO-J sagteware, soos ontwikkel as 'n Java program-koppelvlak deur die Universiteit van Hamburg se Fakulteit van Informatika. DESMO-J is gekies as 'n alternatief vir die meer tradisionele simuleringstale omrede goeie dokumentasie, maklike gebruik en buigbaarheid. Alle teoretiese modelering is uitgevoer in Matlab

# Acknowledgements

I would like to thank the following people for their inspiration, guidance, insights and motivation, without whom this thesis would not have been completed.

- My Supervisor, Dr. R Wolhuter for his insights and guidance into problems as well as his dedication under difficult circumstances.
- My family and friends, whose motivation inspired me throughout.
- My Wife, whose support has been invaluable.
- Mr Johannes Gobel, whose insights into the DESMO-J API has been invaluable.
- ProDesign, for providing information regarding their communication protocol
- NamWater, for providing information of their telemetry systems.
- God

# Dedications

For my mom and dad

# Contents

<b>Declaration</b>	<b>i</b>
<b>Abstract</b>	<b>ii</b>
<b>Uittreksel</b>	<b>iii</b>
<b>Acknowledgements</b>	<b>iv</b>
<b>Dedications</b>	<b>v</b>
<b>Contents</b>	<b>vi</b>
<b>List of Figures</b>	<b>xiii</b>
<b>List of Tables</b>	<b>xviii</b>
<b>Nomenclature</b>	<b>xix</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Introduction . . . . .	1
1.2 Project Objectives . . . . .	2
1.3 Significance of Research and Contributions to Focus Area . . . . .	2
1.4 Overview of Thesis . . . . .	3
1.4.1 Chapter 2 . . . . .	3
1.4.2 Chapter 3 . . . . .	3
1.4.3 Chapter 4 . . . . .	3
1.4.4 Chapter 5 . . . . .	4
1.4.5 Chapter 6 . . . . .	4

1.4.6	Chapter 7 . . . . .	4
1.4.7	Chapter 8 . . . . .	4
1.4.8	Chapter 9 . . . . .	4
1.4.9	Chapter 10 . . . . .	4
<b>2</b>	<b>Narrow Band Communication Networks in Industry</b>	<b>5</b>
2.1	Channel Capacity . . . . .	6
2.2	Analogue vs Digital Radio Systems . . . . .	6
2.2.1	Analogue Radio Systems . . . . .	6
2.2.2	Digital Radio Systems . . . . .	6
2.2.3	Comparison of Digital and Analogue radio systems . . . . .	7
2.3	Basic Communication Protocols . . . . .	8
2.3.1	Introduction . . . . .	8
2.3.1.1	Physical . . . . .	9
2.3.1.2	Data Link . . . . .	9
2.3.1.3	Network layer . . . . .	9
2.3.1.4	Transport layer . . . . .	9
2.3.1.5	Session layer . . . . .	10
2.3.1.6	Presentation layer . . . . .	10
2.3.1.7	Application layer . . . . .	10
2.3.2	Data Link Layer . . . . .	11
2.3.2.1	Introduction . . . . .	11
2.3.2.2	Logical Link Control Sublayer . . . . .	11
2.3.2.3	Media Access Control Sublayer . . . . .	17
2.4	Data from Practical Systems . . . . .	22
2.4.1	River Remote Station . . . . .	22
2.4.2	Repeater Remote Station . . . . .	22
2.4.3	Reservoir Remote Station . . . . .	23
2.4.4	Borehole Remote Station . . . . .	23
2.4.5	Booster Pump Remote Station . . . . .	24



2.4.6	Point to Point transmissions . . . . .	25
2.4.7	Remote Radio Diagnostics . . . . .	25
2.4.8	Critical Nature of Data Collected . . . . .	26
2.4.9	Practical Example: Namib Water Supply Scheme . . . . .	26
2.5	Summary . . . . .	31
<b>3</b>	<b>Simulation Modelling</b>	<b>32</b>
3.1	Introduction . . . . .	32
3.1.1	Motivation and Overview . . . . .	32
3.2	Basic Concepts in Discrete Event Simulation . . . . .	33
3.2.1	Discrete Event Simulation Basics . . . . .	33
3.2.2	Discrete Event Simulation Model Components and Techniques . . . . .	33
3.2.2.1	Entities . . . . .	33
3.2.2.2	Simulation Time . . . . .	33
3.2.2.3	Event List . . . . .	34
3.2.2.4	Statistical Counter and State Variables . . . . .	34
3.2.2.5	Frameworks . . . . .	34
3.2.3	Simulating Random Events . . . . .	35
3.2.4	Transient Phases and Steady State . . . . .	35
3.2.5	Process-Oriented Simulation Modelling . . . . .	35
3.3	DESMO-J . . . . .	37
3.3.1	Overview . . . . .	37
3.3.2	Process-orientated modelling . . . . .	39
3.4	Summary . . . . .	40
<b>4</b>	<b>Queuing Theory</b>	<b>41</b>
4.1	Basic Queuing Principles . . . . .	41
4.1.1	Kendall's notation . . . . .	43
4.1.2	Little's Theorem . . . . .	43
4.2	Poisson Process . . . . .	43
4.2.1	Properties . . . . .	44

4.2.1.1	Superposition Property . . . . .	44
4.2.1.2	Decomposition Property . . . . .	45
4.2.1.3	Exponentially Distributed Inter-Arrival Times . . . . .	45
4.3	Markov Process . . . . .	45
4.3.1	Birth-Death Process . . . . .	47
4.3.2	Single Queue Markovian System . . . . .	47
4.3.2.1	Global and Local Balance and M/M/1 Queue Analysis . . . . .	48
4.3.2.2	Performance Calculations . . . . .	49
4.3.3	Finite Single Queue Markovian System for use in Telemetry Systems . . . . .	50
4.4	Summary . . . . .	55
<b>5</b>	<b>Typical Network Layout and Protocol Options</b>	<b>56</b>
5.1	Introduction . . . . .	56
5.2	Network Layout . . . . .	56
5.3	CSMA Protocol . . . . .	58
5.3.1	Non-persistent CSMA . . . . .	58
5.3.1.1	Service and Acknowledgement Time . . . . .	58
5.4	Round Robin Protocol . . . . .	59
5.5	Adaptive Tree Walk Protocol . . . . .	59
5.5.1	Overview . . . . .	59
5.5.2	Computerized \ Slotted Adaptive Tree Walk Algorithm . . . . .	60
5.6	Summary . . . . .	61
<b>6</b>	<b>Carrier Sense Multiple Access: Modelling and Simulation</b>	<b>62</b>
6.1	Introduction . . . . .	62
6.1.1	Description of the protocol used . . . . .	62
6.1.2	Vulnerable Period . . . . .	64
6.1.3	Data and Acknowledgement Frame Transmission Rates . . . . .	65
6.1.4	The Probability of Error . . . . .	66
6.1.5	Modelling Considerations and Assumptions . . . . .	67
6.2	Theoretical Modelling . . . . .	68

6.2.1	Introduction . . . . .	68
6.2.2	Existing State Space Model . . . . .	68
6.2.3	Expansion of Existing State Space Model . . . . .	73
6.3	Simulation Modelling . . . . .	80
6.3.1	Processes-Orientated Modelling . . . . .	81
6.3.1.1	The Generator Process . . . . .	81
6.3.1.2	The Station Process . . . . .	82
6.3.1.3	The Server Process . . . . .	85
6.3.2	Simulation Model Layout . . . . .	87
6.3.3	Model Control Class . . . . .	87
6.4	Results . . . . .	89
6.4.1	Theoretical Results . . . . .	89
6.4.2	Simulation Results . . . . .	94
6.4.3	Comparison of Results . . . . .	105
6.5	Summary . . . . .	109
<b>7</b>	<b>Round Robin Polling: Modelling and Simulation</b>	<b>110</b>
7.1	Introduction . . . . .	110
7.2	Protocol Structure . . . . .	110
7.3	Theoretical Modelling . . . . .	113
7.4	Simulation Modelling . . . . .	115
7.4.1	Processes-Orientated Modelling . . . . .	115
7.4.1.1	The Generator Process . . . . .	116
7.4.1.2	The Station Process . . . . .	117
7.4.1.3	The Server Process . . . . .	119
7.4.2	Simulation Model Configuration . . . . .	121
7.5	Results . . . . .	121
7.5.1	Theoretical Results . . . . .	121
7.5.2	Simulation Results . . . . .	123
7.5.3	Comparison of Results . . . . .	128

7.6	Summary	132
<b>8</b>	<b>Adaptive Tree Walk: Modelling and Simulation</b>	<b>133</b>
8.1	Introduction	133
8.2	Unslotted Adaptive Tree Walk protocol	133
8.2.1	Configuration	135
8.3	Theoretical Modelling	136
8.3.1	Introduction	136
8.4	Simulation Modelling	139
8.4.1	Processes-Orientated Modelling	139
8.4.1.1	The Generator Process	140
8.4.1.2	The Station Process	142
8.4.1.3	The Server Process	143
8.4.2	Simulation Model Configuration	145
8.4.3	Simulation Program	145
8.4.3.1	The Binary Tree Iteration	145
8.5	Results	146
8.5.1	Theoretical Results	146
8.5.2	Simulation Results	150
8.5.3	Comparison of Results	156
8.6	Summary	159
<b>9</b>	<b>Comparison of CSMA, RRP and ATW protocols</b>	<b>160</b>
9.1	Overview	160
9.2	Comparison of Simulation Models	160
9.2.1	Comparison of Waiting Times and Queue Lengths	160
9.2.2	Cycle Times Comparison for the RRP and ATW Protocols	170
9.3	Summary	171
<b>10</b>	<b>Summary and Conclusion</b>	<b>172</b>
10.1	Summary	172

10.2 Contributions and Conclusion . . . . .	172
10.2.1 Theoretical Models Based on Queueing Theory . . . . .	173
10.2.2 Verifying the Accuracy of the Theoretical Models . . . . .	173
10.2.3 ATW Protocols . . . . .	173
10.3 Future Work and Recommendations . . . . .	173
<b>Bibliography</b>	<b>175</b>
<b>A First Appendix</b>	<b>A-1</b>
A.1 ProDesign Protocol Flow Chart . . . . .	A-2
A.2 Arrival Distribution Graphs . . . . .	A-3
A.3 Gaussian Distributions . . . . .	A-5
<b>B Second Appendix</b>	<b>B-1</b>
B.1 Graphical User Interfaces . . . . .	B-1
B.1.1 Matlab . . . . .	B-1
B.1.2 Java . . . . .	B-3
B.1.2.1 Main User Interface . . . . .	B-3
B.1.2.2 CSMA User Interface . . . . .	B-4
B.1.2.3 RRP User Interface . . . . .	B-4
B.1.2.4 ATW User Interface . . . . .	B-5
<b>C Third Appendix</b>	<b>C-1</b>
C.1 Proofs and theories . . . . .	C-1
C.1.1 Exponentially Distributed Arrivals have a Poisson Distribution . . . . .	C-1
C.1.2 Memoryless Markovian Property . . . . .	C-1
C.1.3 Monroe-Penrose Pseudo Inverse . . . . .	C-2

# List of Figures

2.1	Namib and Keetmans Test Sites . . . . .	8
2.2	OSI Model taken from [32] . . . . .	9
2.3	Model of Frame Transmissions taken from [34] . . . . .	11
2.4	Stop-and-Wait Flow Control . . . . .	12
2.5	Stop-and-Wait ARQ taken from [12] . . . . .	14
2.6	Sequential Numbering from [12] . . . . .	15
2.7	MAC Protocols . . . . .	18
2.8	Typical River Remote Station taken from NamWater SCADA . . . . .	22
2.9	Typical Repeater Remote Station taken from NamWater SCADA . . . . .	22
2.10	Typical Reservoir Remote Station taken from NamWater SCADA . . . . .	23
2.11	Typical Borehole Remote Station taken from NamWater SCADA . . . . .	24
2.12	Typical Booster Pump Station taken from NamWater SCADA . . . . .	25
2.13	Current Namib Water Supply Scheme . . . . .	28
2.14	Future Namib Water Supply Scheme . . . . .	29
2.15	Future Namib Water Supply Sub-Schemes . . . . .	30
3.1	Simulation Study Schematic from [22] . . . . .	32
3.2	Discrete Event Simulation Flow taken from [26] . . . . .	34
3.3	Relationship of Events, Activities and Processes from [26] . . . . .	35
3.4	Telemetry Network with Two Repeaters . . . . .	37
3.5	DESMO-J Class Hierarchy taken from [26] . . . . .	39
3.6	Description of the use of Hold, Passive and Activate Methods from [26] . . . . .	39
4.1	Basic Queue Model . . . . .	41

4.2	Poisson Superposition . . . . .	44
4.3	Poisson Decomposition Property . . . . .	45
4.4	Birth Death Process State Transition Diagram . . . . .	47
4.5	Global Balance . . . . .	48
4.6	Local Balance . . . . .	48
4.7	Finite Telemetric Queue State Diagram . . . . .	51
5.1	Adaptive Tree with Eight Stations from [37] . . . . .	60
6.1	Non-persistent CSMA Flow Diagram . . . . .	63
6.2	3 Repeater Network Coverage . . . . .	64
6.3	State Space Model from [40] . . . . .	68
6.4	State Space Model with Varying Service Times . . . . .	73
6.5	Expanded State Space Model . . . . .	74
6.6	Effect of Collisions on Total Arrivals . . . . .	79
6.7	Generator Process Activity Diagram . . . . .	82
6.8	CSMA Station Process Activity Diagram . . . . .	84
6.9	CSMA Server Process Activity Diagram . . . . .	86
6.10	CSMA Control Class Flow Chart . . . . .	88
6.11	CSMA Theoretical Parameters . . . . .	89
6.12	CSMA Theoretical Results: Large Frame Sizes . . . . .	90
6.13	CSMA Theoretical Results: Varying Timeouts for Large Frame Size . . . . .	91
6.14	CSMA Theoretical Results: Small Frame Sizes . . . . .	92
6.15	CSMA Theoretical Results: Varying Timeouts for Small Frame Size . . . . .	93
6.16	CSMA Simulation Parameters . . . . .	94
6.17	CSMA Simulation Results: Large Frame Sizes . . . . .	95
6.18	CSMA Simulation Results: Varying Timeouts for Large Frame Size . . . . .	96
6.19	CSMA Simulation Results: Small Frame Sizes . . . . .	98
6.20	CSMA Simulation Results: Varying Timeouts for Small Frame Size . . . . .	99
6.21	CSMA Simulation Results: Varying Backoff Ranges for Large Frame Size . . . . .	100
6.22	CSMA Simulation Results: Varying Backoff Ranges for Small Frame Size . . . . .	101

6.23	CSMA Simulation Results: Exponential vs Uniformly Distributed Backoff for Large Frames . . . . .	102
6.24	CSMA Simulation Results: Effect of the Number of Repeaters on Waiting Times . . . . .	103
6.25	CSMA Simulation Results: ProDesign vs non-persistent CSMA . . . . .	104
6.26	CSMA Theoretical and Simulation Comparison: Large Frame Sizes . . . . .	105
6.27	CSMA Theoretical and Simulation Comparison: Varying Timeouts for Large Frame Sizes . . . . .	106
6.28	CSMA Theoretical and Simulation Comparison: Small Frame Sizes . . . . .	107
6.29	CSMA Theoretical and Simulation Comparison: Varying Timeout for Small Frame Sizes . . . . .	108
7.1	Round Robin Protocol . . . . .	112
7.2	RRP Generator Process Activity Diagram . . . . .	116
7.3	RRP Station Process Activity Diagram . . . . .	118
7.4	RRP Server Process Activity Diagram . . . . .	120
7.5	RRP Theoretical Parameters . . . . .	121
7.6	RRP Theoretical Results: Large Frame Sizes . . . . .	122
7.7	RRP Theoretical Results: Small Frame Sizes . . . . .	123
7.8	RRP Simulation Parameters . . . . .	124
7.9	RRP Simulation Results: Large Frame Sizes . . . . .	125
7.10	RRP Simulation Results: Small Frame Sizes . . . . .	127
7.11	RRP Simulation Results: Arriving Events Serviced . . . . .	128
7.12	RRP Theoretical and Simulation Comparison: Large Frame Sizes . . . . .	129
7.13	RRP Theoretical and Simulation Comparison: Small Frame Sizes . . . . .	130
7.14	RRP Theoretical and Simulation Comparison: Large Frame Sizes with 150 Stations . . . . .	131
8.1	Adaptive Tree with eight stations from [37] . . . . .	133
8.2	ATW State diagram . . . . .	136
8.3	ATW Generator Process Activity Diagram . . . . .	141
8.4	ATW Station Process Activity Diagram . . . . .	142
8.5	ATW Server Process Activity Diagram . . . . .	144
8.6	ATW Binary Tree Representing 8 Stations . . . . .	146
8.7	ATW Theoretical Parameters . . . . .	147
8.8	ATW Theoretical Results: Large Frame Sizes . . . . .	148



8.9	ATW Theoretical Results: Small Frame Sizes . . . . .	149
8.10	ATW Simulation Parameters . . . . .	150
8.11	ATW Simulation Results: Large Frame Sizes . . . . .	151
8.12	ATW Simulation Results: Small Frame Sizes . . . . .	152
8.13	ATW Simulation Results: Effect Of Optimization on Station Waiting Time . . . . .	153
8.14	ATW Simulation Results: Optimized Tree Level, High Arrival Rates and Large Frame Sizes . . . . .	154
8.15	ATW Simulation Results: Optimized Tree Level, Low Arrival Rates and Large Frame Sizes . . . . .	155
8.16	ATW Theoretical and Simulation Comparison: Large Frame Sizes . . . . .	156
8.17	ATW Theoretical and Simulation Comparison: Small Frame Sizes . . . . .	157
8.18	ATW Theoretical and Simulation Comparison: Different Poll Frame Sizes . . . . .	158
8.19	ATW Theoretical and Simulation Comparison: Stabilizing Effect for High Arrival Rate, Small Data Frame Sizes and Data Indication Request and Response Frames' Bytes = 20 . . . . .	159
9.1	Comparison: Low Arrival Rates and Large Frame Sizes . . . . .	161
9.2	Comparison: Low Arrival Rates and Small Frame Sizes . . . . .	162
9.3	Comparison: ATW vs CSMA for Low Data Latencies and Small Frame Sizes . . . . .	163
9.4	Comparison: High Arrival Rates and Large Frame Sizes . . . . .	164
9.5	Comparison: High Arrival Rates and Small Frame Sizes . . . . .	165
9.6	Comparison Results: Very High Arrival Rates and Large Frame Sizes . . . . .	166
9.7	Comparison Results: Very High Arrival Rates and Small Frame Sizes . . . . .	167
9.8	Comparison: Expanded Namib Supply Scheme . . . . .	169
9.9	Comparison: ATW vs RRP Cycle Time . . . . .	170
A.1	ProDesign Protocol Flow Chart Part 1 . . . . .	A-2
A.2	ProDesign Protocol Flow Chart Part 2 . . . . .	A-3
A.3	CSMA Protocol Arrival Rate Histogram with Rate = 0.1 . . . . .	A-4
A.4	CSMA Protocol Arrival Rate Histogram with Rate = 0.4 . . . . .	A-4
A.5	CSMA Protocol Arrival Rate Histogram with Rate = 0.8 . . . . .	A-5
A.6	CSMA Protocol Arrival Rate Histogram with Rate = 1.2 . . . . .	A-5
A.7	CSMA Normalized Gaussian Waiting Time . . . . .	A-6
A.8	CSMA Normalized Gaussian Backoff Time . . . . .	A-6

A.9	CSMA Normalized Gaussian Collision Time . . . . .	A-7
A.10	CSMA Normalized Gaussian Number of Collisions . . . . .	A-7
A.11	CSMA Gaussian Waiting Time . . . . .	A-8
A.12	CSMA Normalized Gaussian Backoff Time . . . . .	A-8
A.13	CSMA Gaussian Collision Time . . . . .	A-9
A.14	CSMA Gaussian Number of Collisions . . . . .	A-9
A.15	Waiting Time of CSMA used for Gaussian Distributions . . . . .	A-10
B.1	Matlab GUI . . . . .	B-2
B.2	Main Java Graphical User Interface . . . . .	B-3
B.3	Main Java Graphical User Interface Menus . . . . .	B-3
B.4	CSMA Graphical User Interface . . . . .	B-4
B.5	RRP Graphical User Interface . . . . .	B-5
B.6	ATW Graphical User Interface . . . . .	B-6

# List of Tables

9.1	Parameter Set for Practical System . . . . .	168
-----	--	-----

# Nomenclature

ABSO	Arrival Bit Set Object
ACK	Acknowledgement
AMPS	Advanced Mobile Phone System
ARQ	Automatic Repeat Request
ATW	Adaptive Tree Walk
API	Application Program Interface
CDMA	Code Division Multiple Access
CSMA	Carrier Sense Multiple Access
CSMA-CA	Carrier Sense Multiple Access with Collision Avoidance
CRC	Cyclic Redundancy Check
DL	Data Latency
FDMA	Frequency Division Multiple Access
FSK	Frequency Shift Keying
GUI	Graphical User Interface
GSM	Global System for Mobile Communication
ISO	International Standards Organization
LLC	Logical Link Control Layer
MSSQL	Microsoft Structured Query Language
M-PPI	Monroe-Penrose Pseudo Inverse
NP-CSMA	Non-Persistent Carrier Sense Multiple Access
NBT	Narrow Band Telemetry
OSI	Open System Interconnection
PDF	Probability Density Function
PLC	Programmable Logic Controller
PRE	Preamble

## NOMENCLATURE

PST	Postamble
RRP	Round Robin Polling
RSSI	Received Signal Strength Indicator
RX	Receive
RXEND	Receiver End
RTU	Remote Terminal Unit
SNR	Signal to Noise Ratio
SQL	Structured Query Language
TDMA	Time Division Multiple Access
TX	Transmit
TXTURN	Transmitter Turnaround
TTL	Time Out Delay
UML	Unified Modelling Language
UMTS	Universal Mobile Telecommunications System
VSWR	Voltage Signal to Wave Ratio

# Chapter 1

## Introduction

### 1.1 Introduction

Telemetry networks are used throughout the world in various industries to monitor, control and maintain various types of infrastructure. Examples of these networks can be found in the mining, water, electrical and transport sectors. These networks can utilise various communication media, including Optical Fibre, RF based technologies such as GPRS, WiMAX, 3G and dedicated VHF/UHF based networks. The choice of carrier is generally determined by the extent of the network, its location and available infrastructure. Networks deployed over great distances are limited to only a few of these technologies, typically GPRS and licensed analogue/digital radio based telemetry systems. The latter are preferred by many organisations as the technology of choice, due to their reliability and large area coverage.

The particular frequency band will normally be licensed to allow for higher transmitter output power (compared to license free frequency bands) which is required to cover greater distances. Licensed bandwidth is usually assigned in slots of 12.5 kHz width, which imposes a limit on the maximum throughput. This type of network can easily span in excess of 50 monitoring stations, all connected to a central control station where the information gathered from the remote points is required for monitoring, control and pre-emptive maintenance purposes. The demand imposed on the communications network will depend on the number of stations and the associated information acquired from their Input/Output (I/O) points. This cannot be increased without limit, as the communications infrastructure might be strained to a point where the reliability and validity of field information would be compromised and no longer realistic and reliable. Real-time performance of the network is also critical, as this might have a direct influence on some automation processes in the network.

The background to the work done, as presented in this document, stems from personal involvement and exposure to typical systems of the type as mentioned above. The Namibia Water Corporation (NamWater) operates a telemetry network comprising various individual digital and analogue based radio networks, servicing over 350 remote monitoring stations in total. These networks are, almost without exception, of the narrow band UHF based relatively low speed type, stretching over huge distances in most cases. Involvement in operations and planning of various sections of this infrastructure has drawn attention to some of the typical problems experienced, such as long latencies, congestion and data delivery failure. Many of these issues can be traced to the inherent particular communications and data protocols. Equipment is generally obtained from off the shelf commercial sources, without too much choice as to the most appropriate strategy for the application and system loading.

The digital radio network servicing the coastal area, known as the Namib region, has been identified as a case in point. The Namib telemetry system currently consists of 3 interconnected digital repeaters and has a total reach of approximately 200 km, servicing 57 active Remote Terminal Units (RTUs), 90% of which are connected to Programmable Logic Controllers

(PLC's). This system is in the process of being expanded, and will then include an additional 50 new RTUs, as well as a fourth digital repeater. This will extend the reach and radius coverage of the system to approximately 260 km. The system monitors repeaters, boreholes, reservoirs, river levels, dams and booster pump stations which are necessary for the water supply to Rossing and Langer Heinrich Uranium mines, the towns of Henties Bay, Swakopmund, Walvisbay, as well as many smaller towns and communities. Depending on the type of station monitored, the number of associated data points can vary from 5 to as much as 150. A typical remote solar powered river station will monitor 2 river levels, station health such as battery voltages, load and supply currents as well as solar and station entry alarms. The number of data points monitored at a typical borehole is more extensive and will include outflow from the borehole, flow totalizer, voltage supply, load currents and various alarms from the motor and pump installation. A booster pump station will monitor even more data points than will a typical borehole. These will include the various alarms, voltages and currents from each motor, pump and Variable Speed Drive (VSD), inlet and outlet pressures, various flows, PLC parameters and RTU health. Additionally, the network is also used to automate the various pumping schemes in the system by integrating water levels, water demand and pump timing. Reliability of network and inherent communications is clearly critical to ensure uninterrupted water supply for the region. Currently, the system is clearly stressed, as evidenced by frequent delays in response and data acquisition. With expansion this is bound to increase. The type and extent of additions that could still be tolerated by the network, in terms of latency and general reliability, are really unknown and should ideally be predetermined.

## 1.2 Project Objectives

From experience, as briefly set out above, it became clear that the somewhat hit and miss approach followed in implementing the type of network under consideration, leaves much scope for improvement. Installation, and particularly comms traffic planning, is generally empirical, at best. The major objective of this research project was to develop a set of theoretical and simulation tools to analyse and predict the performance of the type of protocol used for telemetry networks. With this major objective as focus point and the above as background, the following supporting objectives were defined for the research undertaken as part of this project:

- Present an overview of the characteristics of common, current Narrow Band Telemetry (NBT) protocols
- Review their use in practical systems
- Investigate means of creating sound theoretical predictive models for some of these strategies
- Support the theoretical models by development of accompanying simulation routines, in order to simulate and predict behaviour of both current and envisaged NBT networks
- Investigate means of optimising the performance of some of the standard protocols
- Investigate the feasibility of implementation of an alternative strategy not commonly applied in NBT networks, in order to overcome some of the known shortcomings of prevalent schemes
- Present a performance comparison of the different strategies for a given set of network parameters, to assist with implementation planning for such networks and the more optimal use of scarce spectrum availability

## 1.3 Significance of Research and Contributions to Focus Area

The salient aspects of the work completed and associated advances offered as a result, can be summarised as follows:

- The performance and characteristics of 3 different NBT protocols have been analysed and investigated as part of the work undertaken under this project. They are:
  1. Non-Persistent Carrier Sense Multiple Access (CSMA)
  2. Round Robin Polling (RRP)
  3. Unslotted Adaptive Tree Walk (ATW)
- In all three cases, the characteristics as applicable to practical NBT networks, were specifically taken into account.
- The theoretical predictive bases created for these NBT protocols, have been found to offer good accuracy and, from the best available information, present a valuable first attempt in this area. This is a definite, practically usable improvement, over current empirical methods.
- An existing slotted ATW protocol was significantly adapted to unslotted use and optimised for application in NBT networks. This presents a very definite advance in creating a balanced approach between contention based and deterministic protocols for the type of application in question. This strategy holds much promise and should be seriously considered for practical application. It is also a first attempt in this direction.
- A full set of simulation routines have been developed as an additional tool to be used in parallel with the mathematical models. The simulations are easily adaptable and suited for practical application. In this application, they have been utilized for performance estimation for the envisaged expanded Namib supply scheme.

## 1.4 Overview of Thesis

### 1.4.1 Chapter 2

Chapter 2 presents an overview of communication networks, with emphasis on NBT networks as used for infrastructure monitoring. The principles covered include:

- RF Propagation Principles
- Typical RF interference to be expected for the type of network
- Communication protocols for NBT networks

### 1.4.2 Chapter 3

In Chapter 3 the possible simulation environments available to model protocol performance are discussed. Discrete type simulations, particularly, are discussed as relevant, with DESMO-J introduced as the discrete event simulation Application Program Interface (API) used for this work.

### 1.4.3 Chapter 4

Chapter 4 deals with the principles of theoretical modelling of communication protocols. Single Server Markov chain theory and the ensuing queueing theory, together with a state space approach, are presented as a viable means of creating a realistic mathematical base for protocol performance prediction. The principles of Poisson distributed event arrivals and service completion are introduced. Queue lengths are established, followed by wait times, or effective system latency, as derived by means of Little's law. These are the key performance indicators of the protocol.



#### **1.4.4 Chapter 5**

An overview of a typical telemetry network topology is presented, together with a discussion of protocols commonly used in these types of network. Properties particular to such a narrow band telemetry (NBT) network, are noted. Protocols commonly used are Non-Persistent Carrier Sense Multiple Access (CSMA) and Round Robin Polling (RRP). The possibility of utilising an alternative Adaptive Tree Walk (ATW) protocol is introduced.

#### **1.4.5 Chapter 6**

This chapter deals with the non-persistent CSMA protocol in depth, as it is widely used for this type of application. It is known to be very effective at lower arrival rates. A modified version of this protocol is used within the NamWater telemetry infrastructure and, therefore, analysed. For this purpose, an extension of an existing state space model type is utilised. The development of a corresponding simulation model implemented within DESMO-J is covered and the results from both models presented and compared.

#### **1.4.6 Chapter 7**

Another well known protocol used in this type of environment is RRP, and an analytical model is presented, as for the CSMA version. The protocol increases in effectiveness with increase in arrival rate, within reasonable boundaries. The development of the simulation model under DESMO-J is discussed and the results of both models presented and compared.

#### **1.4.7 Chapter 8**

ATW has been identified as a protocol which could possibly deliver the best of both RRP and CSMA. The slotted version, as normally implemented in other types of application, is adapted to an unslotted version. A theoretical model, again making use of State Space modelling principles, is derived. Once more, a simulation routine is created under DESMO-J. It is shown that the results from the two solutions compare closely. The expectations from the modified ATW are also validated.

#### **1.4.8 Chapter 9**

This chapter presents a comparison between the results of the individual protocols obtained earlier. The most important characteristics are discussed, as well as their implications for practical application under different system conditions.

#### **1.4.9 Chapter 10**

The work contained in the thesis is summarised, the main results highlighted and contributions noted. Some concluding remarks are presented, with recommendations for further research.

## Chapter 2

# Narrow Band Communication Networks in Industry

Narrow band networks are widely used throughout industry, in particular in applications where the bandwidth requirement is limited and a data rate of between 1200 and 19200 bps will suffice. This type of network is used in industries where the area to be covered by the network is extensive and repeater sites are limited. Examples of these include oil and gas mining, water, waste water and agricultural industries, to mention a few. The frequency spectrum is a precious, limited resource of each country and is usually regulated by a governmental body which follows the guidelines set out by the ITU Radio-communication Sector (ITU-R) and the World Radio Conference (WRC). Point to multi-point telemetry links are assigned within the frequency range 440-450MHz, with transmit and receive frequencies 5MHz apart[31]. Licenses are assigned in 12.5KHz and transmit power is limited. It is therefore clear that bandwidth, and therefore the amount of data that can be transferred, is limited.

Line of sight propagation is the most important propagation mechanism in UHF and VHF communication. It refers to communication links which are visible to each other over the radio horizon (which is different to the optical horizon) and the received signal is a summation of different signals from the source, where some signals have been reflected from earth bound objects [30]. Line of sight propagation is influenced by the following:

- Refraction, which can change over time, occurs in the atmosphere and alters the trajectory of radio waves.
- Diffraction caused by objects near the direct path of the radio wave front
- Reflection from the ground or other objects
- Tropospheric Scatter Loss, caused by signals scattered from the troposphere
- Clutter Loss, caused by objects in the immediate vicinity of the transmitter

This is a field that has been very comprehensively studied and more information regarding this topic can be found in [30, 29, 24]. The signal to noise ratio (SNR) and received signal strength index (RSSI) are important indicators used to determine the quality and level of the signal received. A weaker signal has a lower SNR. Manufacturers of digital radios normally provide a table of expected BER vs RSSI. In this study a BER of  $10^{-6}$  has been used, taken from the datasheet of the MDS 4710E digital radio at a receiver signal strength of -110dB. The RSSI is commonly used in the design of the type of network concerned.

## 2.1 Channel Capacity

Channel capacity, or the maximum amount of information possible for transmission over a particular channel, is defined by the Shannon law in terms of bandwidth and SNR [35]. The digital radios used within the type of system concerned are capable of a data rate of 4800 bps at  $\text{SNR} > 30$ .

## 2.2 Analogue vs Digital Radio Systems

Telemetry system communication infrastructure can make use of either analogue or digital radio systems. Analogue radios have been designed for voice communication and do not have serial input ports for binary data, whereas digital radios have been designed for data communication and have serial input ports. These systems will be discussed in the following sections.

### 2.2.1 Analogue Radio Systems

In order to utilise analogue voice radios for data transmission, a 1200 bps Frequency Shift Keying (FSK) modem is commonly used. To increase coverage, repeater stations are used, retransmitting the signal for receipt by the base station equipped with a similar analogue to digital modem.

When the attenuated signal is received at the repeater, amplified and retransmitted, the noise components are increased as well. Therefore, the further the signal propagates, the more repeaters will be necessary, resulting in a lower SNR. To overcome this problem, the signal power can be increased by using more powerful transmitters (usually 25W), thereby improving the SNR at the first repeater and overall.

Due to the fact that these radios have been designed for audio use, the keying times (time required for the radio to start transmitting data) are usually long, typically 150-200 ms, as shorter keying times are not required for voice. Each repeater required to relay the data signal from the remote station, must be keyed before information can be sent and de-keyed afterwards, to allow for successful transmission of the data. This increases the data latency per package significantly, with a subsequent decrease in protocol efficiency.

Finally, regular maintenance is required on analogue repeaters to recalibrate the settings such as squelch lift, pre-emphasis, de-emphasis and modulation deviation.

### 2.2.2 Digital Radio Systems

Digital radios referred to in this section are specifically designed for narrow band, long range telemetry applications. Digital radios usually make use of serial connections to allow binary data to be sent and received from an external device (usually RS232/485). This data can be buffered or sent immediately, depending on the availability of the communication channel. Data is usually transmitted at a rate of 4800-19200 bps, depending on installation requirements and channel bandwidth. If wide area coverage is required, repeaters are used. The repeater's receiver/modem will recover the original data from the received signal before it is retransmitted, thereby compensating for the signal attenuation and noise. With this type of regeneration, the error rate can be kept within acceptable limits for a number of repeaters. Outstation radios and repeaters have typical data latency times of  $< 10\text{ms}$ . Furthermore, all the repeaters in a transmit chain need not be keyed before a transmission from the remote station commences. This is due to the fact that each data transmission is first demodulated by the receiver before modulating the signal again for transmission, thus ending the data transmission from the remote station's radio. The maximum transmission power of digital radios is usually 5W.

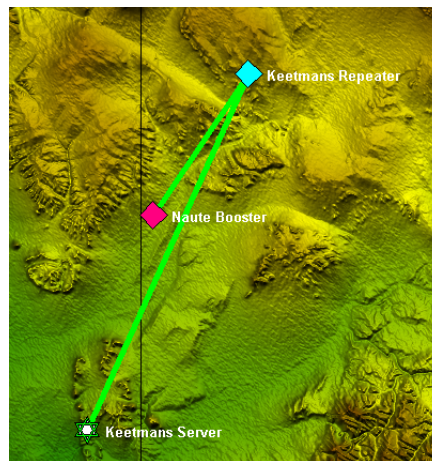
### 2.2.3 Comparison of Digital and Analogue radio systems

The main advantages of digital radio systems over analogue radio systems are as follows:

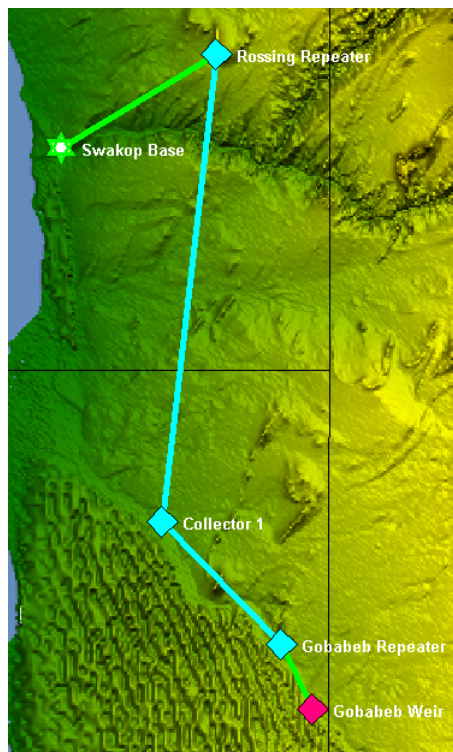
- Higher bit rates
- Lower data latency and delay times
- Lower BER
- Easier to maintain
- Less complex
- Lower power consumption (critical in solar applications)

Many advances have been made in analogue radio technology, but digital radio technology has now completely overtaken analogue types for long range telemetry solutions.

A practical experiment was done on two separate networks of NamWater, analogue and digital respectively, to determine the time required for a station to successfully communicate with the base station after information had been requested. These systems are shown in Figure 2.1. Both use precisely the same servers and software at the base stations. A stop watch was used to determine the approximate time for a remote station to return its data to the server after a polling request was made from the server. The Naute booster station, forming part of the analogue radio system of Figure 2.1a, was polled 10 times. The average time taken for the information to be received at the server was found to be 8.4 seconds. The information was relayed through one repeater. In the digital system of Figure 2.1b, Gobabeb Flood Warning station was polled 10 times. The average time for information to be received at the server was measured as 1.1 seconds. The information was relayed through three repeaters and two links. Both cases include the request for the data packet, as well as the data packet time itself. This clearly shows the difference in performance between an analogue and digital radio system, in spite of the topological disadvantage of the digital system. In the above mentioned figures, the blue icons represent the repeaters, green the server and pink the remote stations.



(a) Keetmans Analogue Radio System



(b) Namib Digital Radio System

Figure 2.1: Namib and Keetmans Test Sites

## 2.3 Basic Communication Protocols

### 2.3.1 Introduction

The well known OSI (Open System Interconnection) Model will be discussed. The OSI Model created seven different layers for the communication process. These layers are shown in Figure 2.2 and described below:

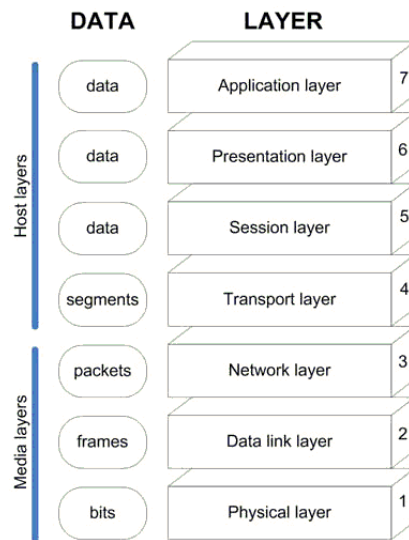


Figure 2.2: OSI Model taken from [32]

### 2.3.1.1 Physical

The physical layer deals with the low level transmission of bits over a communication network. This includes hardware, modulation and demodulation.

### 2.3.1.2 Data Link

The Data Link layer is usually assigned the following tasks [37]:

- Managing the bit stream size sent to the physical layer
- Manages error detection and correction.
- Prevents transmitters from drowning slow receivers with data
- Control access to the shared communication channel in broadcast networks through a special sublayer known as the Medium Access Control (MAC) Layer.

### 2.3.1.3 Network layer

The network layer organises data routing from source to destination and takes care of issues such as different addressing from different networks, breaking up packages into manageable sizes and mediation between different protocols used at the source and destination. It should be noted that in broadcast networks, like narrow band radio networks where a specific channel is shared, the routing problem is simple and the network layer is often small, or non-existent.

### 2.3.1.4 Transport layer

The basic function of the transport layer is to receive information from the higher levels, split it up into smaller units if required, pass these to the network layer and ensure that these arrive correctly at the destination point. [37]. The transport layer determines the type of service to provide the session layer with. These services include:

- Connection-orientated data stream
- Reliability
- Flow Control
- Multiplexing

The connection-orientated protocol is widely used and keeps track of segments transmitted. It retransmits those which fail, while delivering the messages in the order that they were sent. The associated protocol carries out error detection, error recovery, sequencing and flow control of the sent messages [12]. The transport layer is an end-to-end layer allowing communication between source and destination without regard to what happens in between. The transport layer manages various applications which are trying to access the same lower level protocols. In telemetry applications the transport layer manages only the telemetry application.

#### 2.3.1.5 Session layer

Unlike the layers below it in the OSI model, the session layer is concerned with the transportation of data between end points. The session layer manages the interactions (dialogue) between end users (applications) on the source and destination machines and provides data transfer services such as: [16]

- Half Duplex
- Full Duplex
- Quarantining
- Synchronizing

As was mentioned for the transport layer, the telemetry application accesses only the lower layers, establishing one session at a time and making the session layer unnecessary.

#### 2.3.1.6 Presentation layer

The presentation layer is concerned with the syntax and semantics of information exchanged between two systems [32]. According to [10], specific responsibilities of this layer are:

- Translation
- Encryption
- Compression

#### 2.3.1.7 Application layer

The application layer contains the various protocols that might be required by the user [37], which include protocols like MODBUS, or proprietary protocols, e.g. SSE and ProDesign, in telemetry applications.

## 2.3.2 Data Link Layer

The main focus of this thesis falls within the Data Link Layer. Elements of the Data Link Layer pertaining to narrow band radio networks will be discussed in detail below:

### 2.3.2.1 Introduction

The data link layer can be divided into two sub-layers, namely the Logical Link Control (LLC) and Media Access Control (MAC) layers. The LLC layer provides flow control, acknowledgement and error notification. The MAC layer is used to determine who gets access to the communication channel and how it is accessed.

### 2.3.2.2 Logical Link Control Sublayer

IEEE Standard 802.2 defines Logical Link Control (LLC) as a data link control layer used in 802.3 (Ethernet), 802.5 (Token Ring) and others. [35]. In telemetry applications the LLC layer will not be implemented according to this standard and will only be used for error and flow control. Error and flow control within this sublayer will be discussed next.

#### 2.3.2.2.1 Flow Control

Flow control is a technique used to regulate the transmission of data in order not to overwhelm the receiver. The receiver makes use of a buffer where data temporarily resides while it is being processed before being sent to the higher levels of the software. If flow control isn't applied, this buffer can be overwhelmed by information, overflow and therefore data losses. Figure 2.3a shows the transmission of frames between the sender and receiver of data. The time delay between the transmission and reception of the frame can be divided into two categories. The first is known as the transmission time, which is the time required for all bits of the frame to be transmitted onto the communication medium. The second is the propagation time, which is the time required for the frame to traverse the communication medium between the source and destination [35].

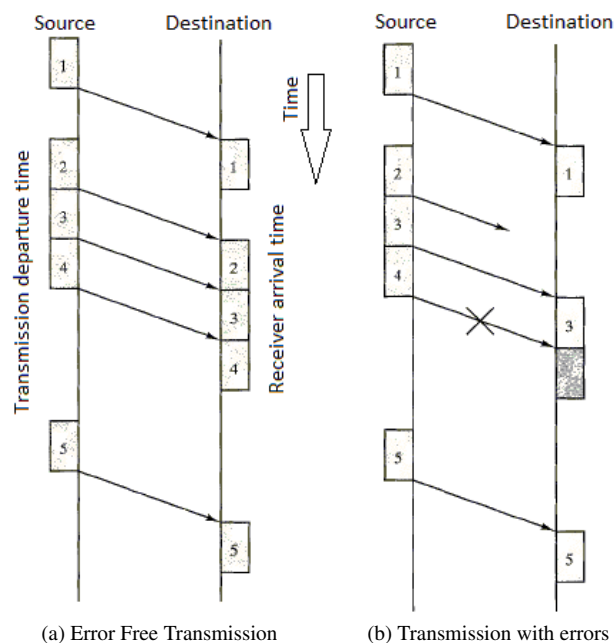


Figure 2.3: Model of Frame Transmissions taken from [34]



### 2.3.2.2.2 Stop-and-Wait Flow Control

A source entity transmits a frame of data. The receiver entity receives the frame of data after an elapsed period of time comprising the transmission and propagation time. The sender now waits for an acknowledgement (ACK) from the receiver to indicate that it has received the data frame. After receiving an acknowledgement (ACK) from the receiver, the sender will continue, sending the next frame of data to the receiver. The time line is demonstrated in Figure 2.4. Also note from the figure that the time taken to transmit the frame is depicted as equal to the propagation time. Note that this specific case of the transmission time can be longer than the propagation delay, or vice versa.

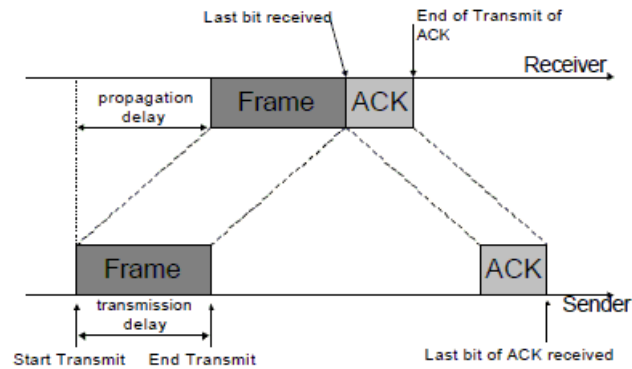


Figure 2.4: Stop-and-Wait Flow Control

Data is usually broken up into smaller frames by the source, for the following reasons:

- Buffer size of the receiver might be limited
- Larger data frames increase the probability of errors occurring during transmission
- To decrease the waiting time of other stations sharing the communication medium.

To determine the efficiency of the link we first look at the 'bit length' of the link, which can be defined as the number of bits present on the link when it is fully occupied. This is defined as follows : [35]

$$B = R \frac{d}{V} \quad (2.1)$$

where:

$B$  Length of the link in bits

$R$  data rate of the link (bps)

$d$  length of link (m)

$V$  velocity of propagation (m/s)

When the bit length of the link is greater than the frame length transmitted, the link is under utilized and inefficient. Therefore:

$$a = \frac{B}{L} \quad (2.2)$$

where:

a ratio of link length to frame length

L Length of frame in bits

Taking a typical radio network with a data rate of 4800bps, and average link distance of 50km we get the following:

$$B = 4800 \times \left( \frac{50 \times 1000}{3 \times 10^8} \right) = 0.8$$

If the typical frame is 200 bytes in length and the digital radio requires two extra bits per byte, then:

$$a = \frac{0.8}{200 \times 10} = 0.0004$$

This shows that the channel is efficiently used. If this was not the case and  $a > 1$ , the channel would be inefficiently utilized by the Stop-and-Wait protocol. In cases like this the Sliding Window Flow Control protocol can be used [35, 12, 30].

### 2.3.2.2.3 Error Control

In Figure 2.3b data is again sent as a sequence of frames from a sender, which arrive at the receiver in the same order as which they were sent, delayed by an amount of time. This time though, there is a probability that frames could contain errors. These errors might be in the form of lost or damaged frames. Lost frames do not arrive at the receiver end, which could be due to a number of things, for example burst noise. In this case the receiver is not aware that information was sent to it. Damaged frames arrive at the receiver, but contain invalid information. Among the most common error control techniques are:

- Error detection, discussed in the next section
- Positive Acknowledgement; the receiver returns a positive acknowledgement indicating that it successfully received a frame.
- Retransmission after timeout; the sender retransmits a frame after a specified period of time if the receiver doesn't return an acknowledgement.
- Negative acknowledgement and retransmission, The receiver returns a negative acknowledgement to the sender asking for a retransmission of the previous frame sent. The sender obliges by sending the frame again.

The above mechanisms are known as Automatic Repeat Requests (ARQ). The standardized ARQ models [35], are as follows:

- Stop-And-Wait ARQ
- Go-back-N ARQ
- Selective-reject ARQ

The Go-back-N ARQ and Selective-reject ARQ are techniques used with the Sliding Window Flow Control protocol and will not be discussed further. [12, 35, 30]

### 2.3.2.2.4 Stop-and-Wait ARQ

The sender will transmit a frame to the receiver. At the beginning of the transmission a timeout delay timer is started. If the receiver receives the data frame sent without any errors, it will return an ACK to the sender, indicating that the next

frame can be sent. Should the ACK be received by the sender before the timer runs out, the next frame will be sent and the timer resets. Four errors can occur within this transmission cycle:

- The data frame never reaches the receiver: In this case the delay timer will run out and the package will be sent again.
- The data frame reaches the receiver but with errors: The receiver will detect the errors and disregard the frame. The delay timer will run out at the sender's side and the frame will be sent again.
- The data frame reaches the receiver without errors, but the ACK never reaches the sender: The delay timer will run out at the sender's side and the frame will be sent again. The receiver will recognize the frame received as the same as the previous frame, disregard it, and retransmit the ACK.
- The data frame reaches the receiver without errors, the ACK reaches the sender afterwards but with errors: The sender doesn't recognize the frame received. The delay timer will run out at the sender's side and the frame will be sent again. The receiver will recognize the frame received as the same as the previous frame, disregard it, and retransmit the ACK.

The above process is depicted in Figure 2.5

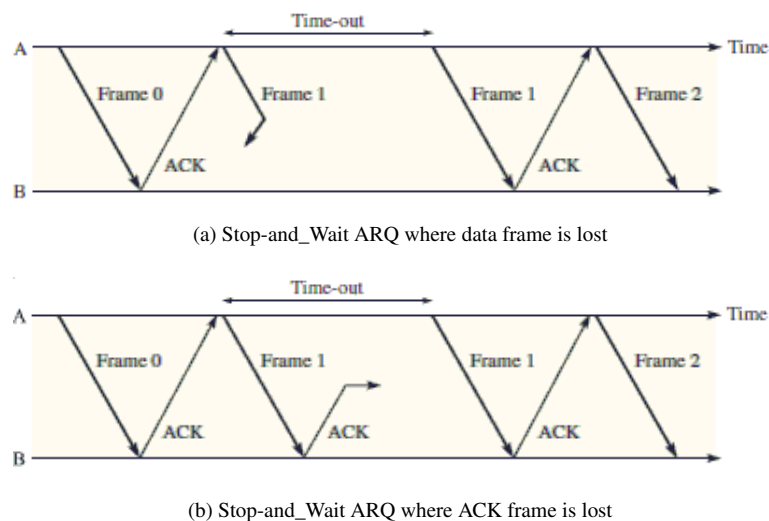


Figure 2.5: Stop-and-Wait ARQ taken from [12]

From the above the problem of duplicity arises. This occurs if the ACK isn't received by the sender on time or is received with errors; a frame already successfully processed at the receiver is resent by the sender and thus again received by the receiver. Furthermore, duplicity can also occur if the time out delay timer expires prematurely and re-sends information that has actually been successfully received by the receiver and an ACK for this frame is still under way to the sender. This problem is solved by adding sequential numbers to the data and ACK frames. This can be done by simply adding a last sent bit to the data frame and a next required bit to the ACK frame [12] as shown in Figure 2.6

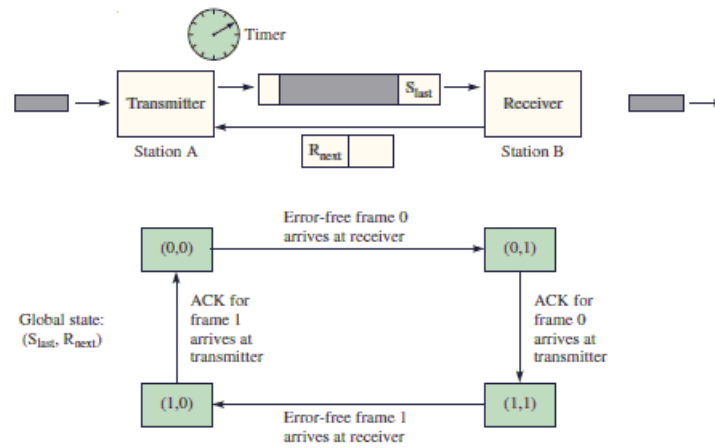


Figure 2.6: Sequential Numbering from [12]

In state (0,0) of Figure 2.6, the sender and receiver are synchronized, the sender is about to transmit a frame with its sequential bit set to 0 and the receiver is expecting a data frame with its sequential bit set to 0. The system will stay in state (0,0) until the receiver receives an error free data frame with sequential bit 0. After reception of such a frame the system will move to state (0,1) where the receiver sends an ACK to the sender signifying that it has successfully received the frame and requesting the next frame with sequential bit set to 1. The system will stay in this state until the sender successfully receives the ACK frame requesting the next data frame with sequential bit set to 1. Therefore any duplicate packets received will be ignored and the ACK asking for the data frame with sequential bit 1 will be resent. When the sender receives the ACK frame successfully, it will determine from the ACK's sequential bit that the receiver requires the next data frame to be sent with its sequential bit set to 1, implicitly indicating that it has received the previous frame successfully. The system now moves to state (1,1) where the receiver waits for the successful reception of the data frame with sequential bit set to 1 and the state won't change until it has been successfully received. Thus the sender and receiver is again synchronized and will repeat the above process again, but this time for data frame with sequential bit set to 1 [12]. Note that this means that until the sender has received an acknowledgement for the current frame it is sending, it can not accept any new frames from its higher levels.

The performance analysis of the Stop-and-Wait ARQ can be calculated as follows:

$$\begin{aligned}
 T &= \tau + t_{dat} + t_{proc} + \tau + t_{ack} + t_{proc} \\
 &= 2(\tau + t_{proc}) + t_{dat} + t_{proc} \\
 &= 2\left(\tau + \frac{d}{V}\right) + \frac{(\eta_{dat} + \eta_{inf})}{R} + \frac{\eta_{ack}}{R}
 \end{aligned} \tag{2.3}$$

where :

$T$	Total time to transmit one frame successfully (sec)
$\tau$	Propagation Time (sec)
$t_{dat}$	Transmission time of data frame (sec)
$t_{proc}$	Processing time of frame (sec)
$t_{ack}$	Transmission time of acknowledgement frame (sec)
$d$	Distance travelled (m)
$V$	Propagation Speed (m/s)

$\eta_{dat}$	Number of data bits in data frame
$\eta_{inf}$	Number of information bits in data frame
$\eta_{ack}$	Number of bits in acknowledgement frame
$R$	Channel Capacity (bps)

The effective data bits per second are calculated as follows:

$$R_{eff} = \frac{\eta_{dat} - \eta_{inf} - \eta_{ack}}{T} \text{ bps} \quad (2.4)$$

From the above equation we can therefore get the channel utilization or channel efficiency:

$$Util = \frac{R_{eff}}{R} \times 100 \quad (2.5)$$

The utilization above is the maximum utilization of the channel. The actual utilization will be lower, due to errors that might occur.

### 2.3.2.2.5 Error Detection

Parity checking is the most basic form of an error detecting code, but is very limited in capability.

Cyclic Redundancy Checks (CRC) is the most commonly used error detection scheme in network applications. CRCs are readily implemented using polynomial codes, as shown in [12, 37, 30]. CRC-16 is a widely accepted standard (e.g. ProDesign and SSE telemetry applications) and catches the following errors:

- All single and double errors
- All errors with an odd number of bits
- All burst errors of length 16 or less
- 99.997% of all 17 bit errors
- 99.998% of all errors  $\geq 18$  bits

### 2.3.2.2.6 Forward Error Correction (FEC)

Forward Error Correction is used not only to detect errors at the receiver end, but to correct a limited number of bit errors. FEC codes are usually used in systems where re-transmission is impossible or costly. Examples of these include mass storage devices, multicasting, audio and video streaming. There are two main types of FEC codes:

- Block codes which are used for fixed sized frames. Examples are Reed-Solomon coding, Golay, BCH and Hamming codes.
- Convolution codes which are used in bit streams or frames with variable length. The Viterbi algorithm is used most often with implementation in Digital TV, cellphones, blue tooth devices and satellite systems.

These codes add extra overhead to the data frame and are relatively complex to implement. Taking into account the extra overhead, the complexity of implementation and the ability to retransmit corrupted frames, this method is usually

not implemented within narrow band radio networks. Typical digital transceivers (e.g. MDS 4710E and Trio MR450) with a bit rate of 4800 bps have a typical error rate of  $10^{-6}$  at a receiver signal strength of -110dBm. Therefore, if data frames have an average size of 2000 bits and ACK frames have a size of 200 bits, then every successful communication will require 2200 bits. Taking this into account we find that it is probable that every 455<sup>th</sup> data frame will have an error bit, resulting in a frame error rate of only 0.219%. Furthermore, it is good practice not to design communication links given a signal strength as low as -110dBm, thus decreasing the probable error rate even further. Most digital radios have embedded FEC, rendering further implementation superfluous.

### 2.3.2.3 Media Access Control Sublayer

Networks can be divided into two categories: those using point to point connections and those using multi-user broadcast channels. In broadcast networks there is only one channel and multiple parties require access to this channel to convey the information they have. The MAC sublayer provides methods to effectively access this channel with the least delay. Narrow band radio networks are broadcast networks and require MAC layer protocols to function effectively. Some of the different categories of MAC layer protocols, are shown in Figure 2.7.

To determine what types of protocol are suited for narrow band radio networks with specific implementation within telemetry networks, the most important network related considerations must be identified. These are discussed hereunder:

#### a) Bandwidth

Bandwidth is assigned in 12.5KHz slots in the 440-450MHz radio spectrum assigned to telemetry applications according to the ITU-R, ICASA, SABRE in South Africa and CRAN (Communications Regulatory Authority of Namibia) in Namibia. Each slot is assigned to a specific company and has an annual licensing fee assigned to it. This can become costly if more than one channel is used.

#### b) Channel Throughput

The channel throughput is limited by the amount of bandwidth available. Newer technologies allow for higher channel throughput (up to 19200 bps), but at the cost of lower receiver sensitivities, decreasing the coverage of the network.

#### c) Type of Communication

Except for the repeaters used within narrow band radio networks, all other radios are half duplex. Full duplex radios do exist, but at a steep increase in cost. For a full duplex system each remote station would have to include either a duplexer or a dual antenna system, increasing the complexity, cost and possible failure points at each station.

#### d) Propagation distance

The distances between remote stations and repeaters are extensive and can be between 50-150 km.

#### e) Cost

In the implementation of radio networks where the core business is not built upon these networks, as with telecommunication service providers, the total implementation, maintenance and operational cost is very important and must be limited while still building an effective network that will function within the required parameters of the specific industry.

These limitations all but cancel out the use of channelization protocols, which are mostly for applications where high bandwidth is required with a continuous stream of data. Examples are GSM, UMTS, AMPS and 3G.

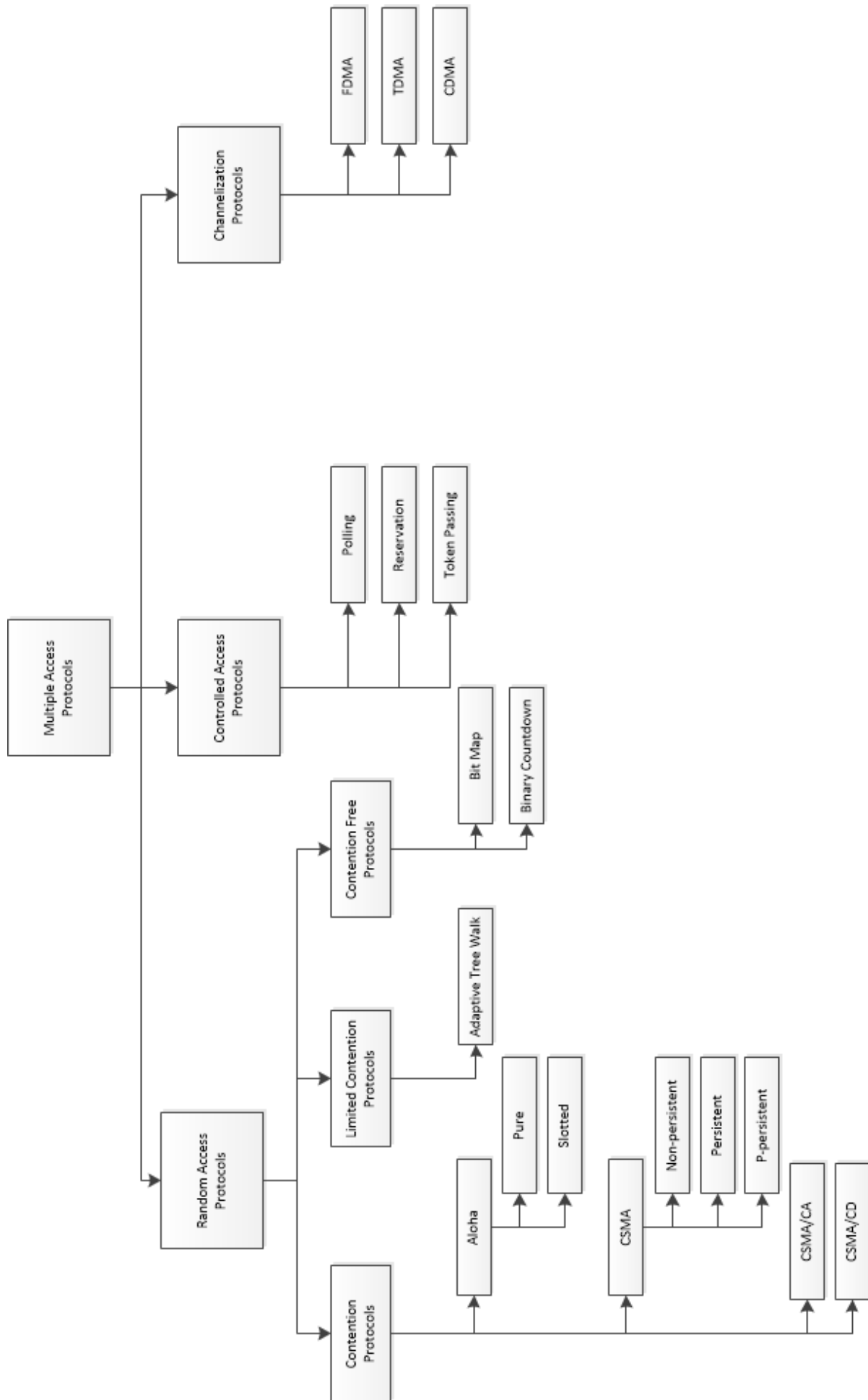


Figure 2.7: MAC Protocols

### 2.3.2.3.1 Channelization Protocols

#### a) Time Division Multiple Access

Time Division Multiple Access (TDMA) divides the usable channel space into time slots which are assigned to users as required. Synchronization of time slots by different stations is required and propagation delays are taken into account by guard bands. The more stations, the more time slots are required, thereby either increasing waiting time per station or requiring another channel to accommodate the extra stations. With large distances between stations, synchronization becomes complex. Bursty traffic also reduces the effectiveness of the protocol as time slots go underutilized [10, 12, 30].

#### b) Frequency Division Multiple Access

Frequency Division Multiple Access (FDMA) divides the usable channel into smaller frequency channels, separated by small guard bands to limit co-channel interference. Each station is assigned to its own sub channel for the duration of communication. FDMA is effective for streaming data, requires prior allocation of bandwidth and is ineffective for bursty traffic [12, 30].

#### c) Code Division Multiple Access

Code Division Multiple Access (CDMA) makes use of codes, allowing different stations to access the communication channel simultaneously. CDMA is a spread spectrum technique and therefore requires extensive bandwidth. The transmit and receive links each require a bandwidth of 1.25MHz for both Code Division Multiple Access 2000 (CDMA2000) and Code Division Multiple Access 450 (CDMA450) [9]. This is already impractical for the application concerned. The protocol also has weak performance under bursty traffic [12].

This leaves the Random Access protocols and Control Access protocols as options. An overview of these protocols is provided hereunder as supporting material.

### 2.3.2.3.2 Random Access Protocols

#### 2.3.2.3.2.1 Contention Protocols

In Contention protocols each station has the right to access the communication channel at any time by following a set of procedures defined by the protocol used. Stations compete for access to the communication channel. If two stations should access the channel at the same time a collision will occur and the data transferred will be rendered meaningless.

#### a) ALOHA

ALOHA was the first random access protocol developed in the early 1970's by Norman Abramson at the University of Hawaii. Pure ALOHA is the name given to the original ALOHA protocol developed in the 70s and basically state that any station can send information via the communication channel whenever it requires to. If a collision occurs, the acknowledgement the station is waiting for, deeming the transmission successful, will not be forthcoming and, therefore, after a predefined timeout period the station will re-transmit the packet. When two packets collide, two stations are affected and will have to retransmit their data This can cause a positive feedback loop which triggers additional collisions. Therefore, after the timeout period of a station has lapsed the protocol allows for another randomized time in order to reduce the probability of two stations' packages (that are being retransmitted) colliding again. A specific station will only be allowed  $N$  attempts to transmit a specific packet, after which it must give up and try again after an extended period of time has elapsed. Still, after each collision, there is an increase in the likelihood of more collisions. Pure ALOHA has a vulnerable period of  $2 \times T_{frame}$ , where  $T_{frame}$  is the time required for the successful transmission of one frame of data. This holds true as any station may start its own transmission soon after another station has started transmitting data, or



just before another station completes its transmission. The performance and flowchart for this protocol can be found in [10].

#### **b) Slotted ALOHA**

Slotted ALOHA reduces the probability of collisions by allowing stations to transmit only at the beginning of a slotted frame, thus changing the protocol from a continuous to a discrete protocol. The slotted time frames are of the same length as those which are required to transmit one frame successfully. Slotted ALOHA doubles the throughput of Pure ALOHA. This is due to the fact that stations can transmit only at the beginning of each slotted frame, causing the vulnerable period to be reduced from  $2 \times T_{frame}$  to  $T_{frame}$ .

#### **c) Carrier Sense Multiple Access (CSMA)**

Carrier sense protocols sense the communication channel before taking action, which is determined by the protocol used. By doing this the number of collisions can be reduced, but they can not be eliminated, due to propagation delay and data latencies found in radio equipment. Therefore, fewer repeaters and shorter propagation distances will improve the performance of the protocol. See [17] for a detailed mathematical analysis of CSMA protocols and their performance. In the following subsections an overview of several versions of the CSMA protocol will be given.

#### **d) 1-persistent CSMA**

If a station has information and, after sensing the channel, finds that it is idle, it will send its data immediately (with probability 1). If the channel is busy, the station will continue sensing the channel until it becomes free and then send its data immediately. This protocol has the highest probability for collision of the persistent protocols. This is due to the fact that if two stations sense that the channel is idle at the same time, both will transmit their data and a collision will occur, after which both stations will go into a random backoff time. Even if there were no propagation delay or equipment latencies, collisions could still occur. An example of this is when two stations sense the channel is busy; both will transmit the moment it becomes available and a collision will occur.

#### **e) Non-persistent CSMA**

In this case a station senses the channel and if the channel is found to be free, it immediately starts transmitting. If, on the other hand, the channel is busy, the station will go into a random backoff period before sensing the channel again. When a station starts to transmit and another station comes out of backoff and does not sense that the current transmission is under way, it too will start to transmit, thus causing a collision. The two stations that were part of the collision only become aware of the collision after the time of waiting for an acknowledgement has expired. If this happens, the stations will go into a random backoff period before sensing the channel again. This vulnerable period is equal to the sum of the propagation delay and equipment latency time. This protocol leads to better channel utilization, but can cause longer delays than 1-persistent CSMA. This protocol is discussed in detail in Chapters 5 and 6.

#### **f) p-Persistent CSMA**

This protocol makes use of principles used in both 1-persistent and non-persistent CSMA. A station will continuously sense the communication channel. If the channel is idle the station will transmit its data with a probability  $p$  and will go into a random backoff period with a probability of  $1 - p$ . After the backoff period has expired, the above process will be repeated. p-persistent CSMA is a slotted system with slots being the size of one contention period, which is equal to or greater than the maximum propagation time. Therefore, transmissions of any data can occur only at the beginning of each slot. The persistent CSMA performance and flowchart is available in [10].

#### **g) Carrier Sense Multiple Access with Collision Detection (CSMA/CD)**

CSMA/CD builds upon the protocols mentioned in CSMA by continuously sensing the channel while transmitting (thus requiring full duplex transceivers). If the signal energy being sensed by a station transmitting increases above that of

a single signal, (In wired systems the signal almost doubles) the station will abruptly end its transmission. By abruptly ending the transmission of the collided frame both time and bandwidth is saved. Full duplex transceivers are not applicable in narrow band radio networks, as previously discussed, therefore this protocol will be discarded in this thesis. The flowchart for this protocol is available in [10].

#### **h) Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA)**

In this protocol a station senses the communication channel continuously. If the channel is found idle, the station backs off for a random period of time (allowing any signal from any other station that has already started to transmit to reach it), after which it senses the channel again. If the channel is still idle, it will start to transmit its frame. If the channel is found to be busy, the station will backoff for a random period of time which is determined by a contention window that is divided into time slots. More detail, with flowcharts and performance analysis, is available in [10].

#### **2.3.2.3.2.2 Limited Contention Protocols**

Limited contention protocols allow for better performance in networks with higher load, while keeping the same performance at low load. This is essential for narrow band networks where the number of stations in the network is high (> 50 stations). A good candidate of this type is the Adaptive Tree Walk protocol. This protocol is discussed in detail in Chapters 5 and 8, including modelling and simulation techniques and results.

#### **2.3.2.3.2.3 Contention Free Protocols**

Both the Bit-Map protocol and the binary countdown protocol are not viable for narrow band radio networks. The Bit-Map protocol is a slotted protocol and, therefore, time synchronization will always be a complex factor in system where propagation and equipment delay play a determinant role. The Binary Countdown protocol assumes that transmission delay times are negligible, which for obvious reasons isn't possible in narrow band radio networks.

#### **2.3.2.3.3 Controlled Access Protocols**

In Control Access Protocols, stations have the right to access the communication channel only once it has been authorized to do so by a primary station (e.g. base station). The token passing protocol requires a ring topology, which will not work with narrow band radio networks. A star topology is usually used in narrow band radio networks by using a repeater on a high site (such as a mountain) to allow for greater propagation distances. It is due to these long propagation distances and accompanying equipment delay that the reservation protocol won't work either, as it is a slotted system and, as discussed above, long propagation delays would make the implementation of this protocol complex and ineffective.

#### **a) Polling**

Polling also known as Round Robin Polling (RRP), makes use of a primary station that requests information from each secondary station, in sequence. When reaching the final station, the process is repeated. All data exchanges, even if the endpoint is a secondary station, must go through the primary device. If a station has nothing to send it will respond with a negative acknowledgement (NACK), otherwise it will proceed in sending the information. The primary will respond to the newly received information by sending an ACK to the secondary station and will then proceed to the next station. Round Robin Polling will be discussed in more detail in Chapters 5 and 7.

## 2.4 Data from Practical Systems

Within a typical water supply scheme as operated by NamWater, there are various types of station that create events causing traffic on the radio network. These different types of station and event types generated by each will now be discussed.

### 2.4.1 River Remote Station

Figure 2.8 shows a typical river telemetry station. These stations are used to monitor river levels for flood management assistance. Due to their remote location and minimum power requirements they are usually solar or wind powered. The critical nature of these stations during the rainy season is evident and all aspects of the station must be closely monitored. Analogue information including river levels, currents and voltages are logged by the Remote Terminal Unit (RTU) on percentage change, (usually 5-10%) as well as at regular intervals (usually every 15 minutes). The digital inputs and outputs are used to monitor failures within the instrumentation, as well as unauthorized entry of the station, which are logged on state of change. Furthermore, the station relays all its information once every hour.

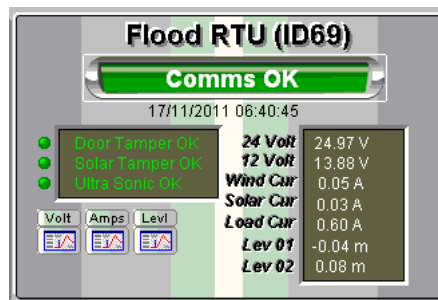


Figure 2.8: Typical River Remote Station taken from NamWater SCADA

### 2.4.2 Repeater Remote Station

A typical repeater station is shown in Figure 2.9. Repeater stations are the backbone of any radio network, therefore the power supply management of the station is critical to the overall state of the network. These stations are usually found on mountains which extends their coverage. Mountains can be very remote and the stations have a high probability of requiring solar and wind power supply. Analogue information including currents and voltages are logged by the RTU on percentage change (usually 5%), as well as at regular intervals (usually every 15 minutes). The digital inputs monitor unauthorized entry and tampering with the stations' solar panels and logged on state of change. The station relays all its information once every hour.

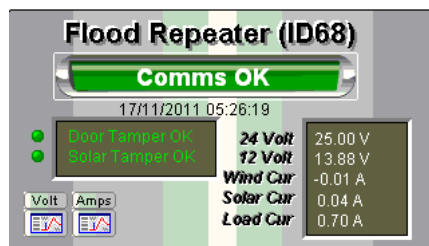


Figure 2.9: Typical Repeater Remote Station taken from NamWater SCADA

### 2.4.3 Reservoir Remote Station

In addition to monitoring the reservoir level, reservoir stations record other important values that must be monitored. These include flows, power, failures and tampering alarms. A typical reservoir station is shown in Figure 2.10. The probability that very remote stations with no pumps installed will require solar and wind power is high. Besides the flow, the number of pulses at each flow meter is monitored, providing totalised flows. Analogue information including currents, voltages, flows and counters are logged by the Remote Terminal Unit (RTU) on percentage change (usually 5-10%), as well as at regular intervals (usually every 15 minutes). The station also broadcasts its level every 10 minutes, allowing any station that might require it, e.g. borehole or booster stations, to update the previous value they had. The digital values monitoring instrumentation failure, unauthorized entry and tampering with the stations solar panels are logged on state of change. The station is designed to relay all its information once every hour. The reservoir level is an indication of whether there are problems somewhere within its water supply. A typical example would be if the pumps supplying the reservoir with water are running, outflow is average and yet the reservoir level isn't increasing. This could indicate that either the transducer measuring the reservoir level is faulty or that a pipe break has occurred and requires a maintenance team to be alerted.

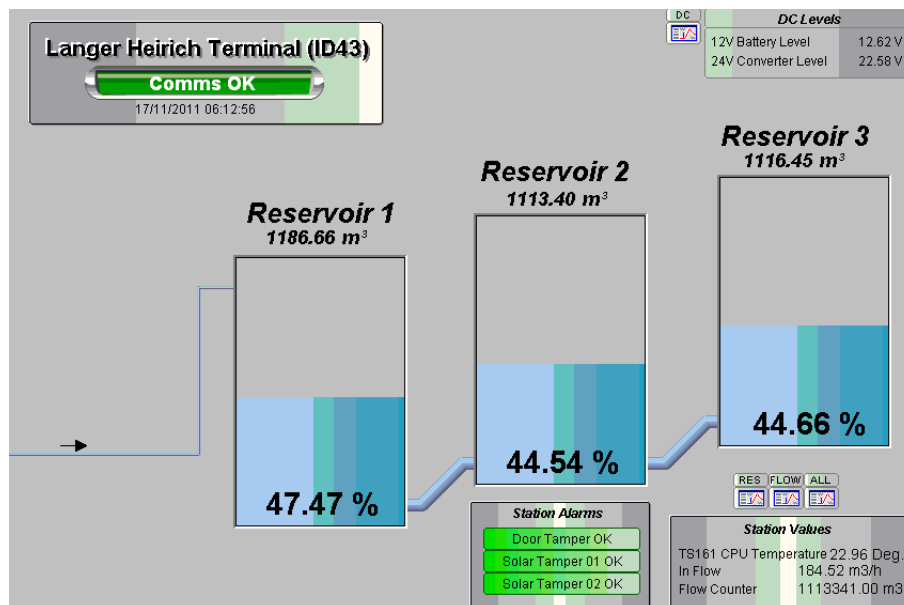


Figure 2.10: Typical Reservoir Remote Station taken from NamWater SCADA

### 2.4.4 Borehole Remote Station

Borehole stations have an electric power supply from the electrical grid and fortunately do not require solar or wind power. These stations usually have one or more pumps installed, and are controlled by a PLC (Programmable Logic Controller). The PLC is connected to the RTU and information is extracted from the PLC using a specific application layer protocol e.g. MODBUS. Information pertaining to the current status of the motor, pump, VSD (Variable Speed Drive) and power supply are obtained from the PLC. Instead of connecting the instruments to the RTU, they can also be connected to the PLC and their values will then be extracted by the RTU. Various counters are also extracted from the PLC for maintenance purposes. These, for example, can include pump running hours, which will indicate when the pump needs to be marked for routine maintenance. All the above mentioned values can be used to determine whether the pump is running within the limits of its design parameters. Furthermore, if the pump has stopped abruptly, the various monitored points can be used to run a diagnostic on the station to determine the exact cause and to assist with appropriate response action planning. The telemetry at the station is also monitored should any communication failures occur and the diagnostics taken from the telemetry can help resolve the issue. It can therefore be seen that the information must be available at regular intervals

as well as on state of change. Parameter logging is set up as for the other remote stations. This station may also monitor the communication channel for an updated value of the reservoir level, which could be used to start and stop the pumps. The value is also logged on the same principles as other analogues to compare it with the actual reservoir level. Figure 2.11 shows a typical borehole station.

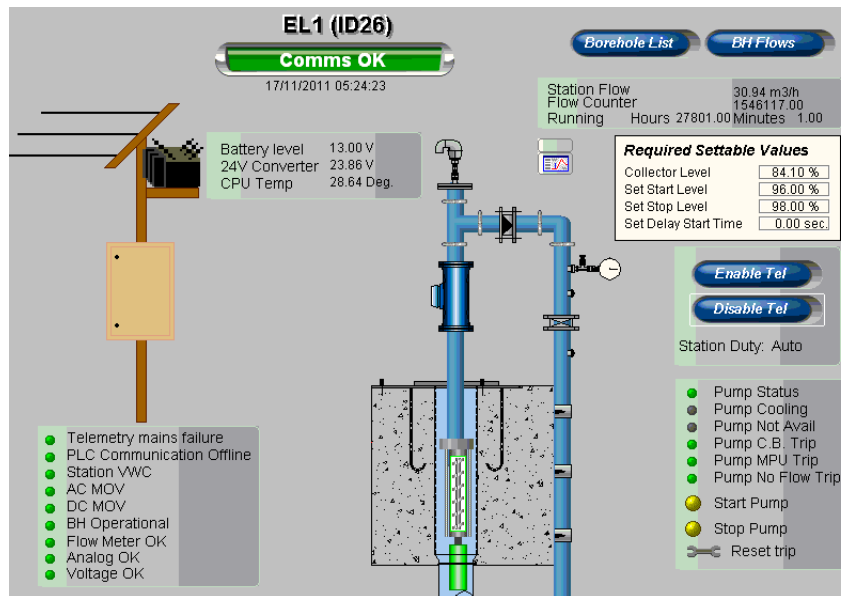


Figure 2.11: Typical Borehole Remote Station taken from NamWater SCADA

### 2.4.5 Booster Pump Remote Station

Booster pump stations are situated on long pipelines and reduce the static or operational pressure within the pipe allowing the maximum flow to be increased. The monitoring of booster pump stations follows the same basic principle as that of borehole stations, except that booster stations have multiple pumps. Pipeline pressure is an additional parameter measurement not mentioned before. Pipeline pressure is an indication of the pipeline state e.g. a sudden drop in pressure will most likely indicate a pipeline break, but it could also be an indication of a faulty instrument. The same monitoring parameters should be followed in terms of eventing and logging that has been noted for borehole stations. See Figure 2.12. Both booster and borehole stations have the added functionality of being able to start and stop the pumps remotely. This is very useful in the case of automation failure caused by instruments or a point to point link failure.

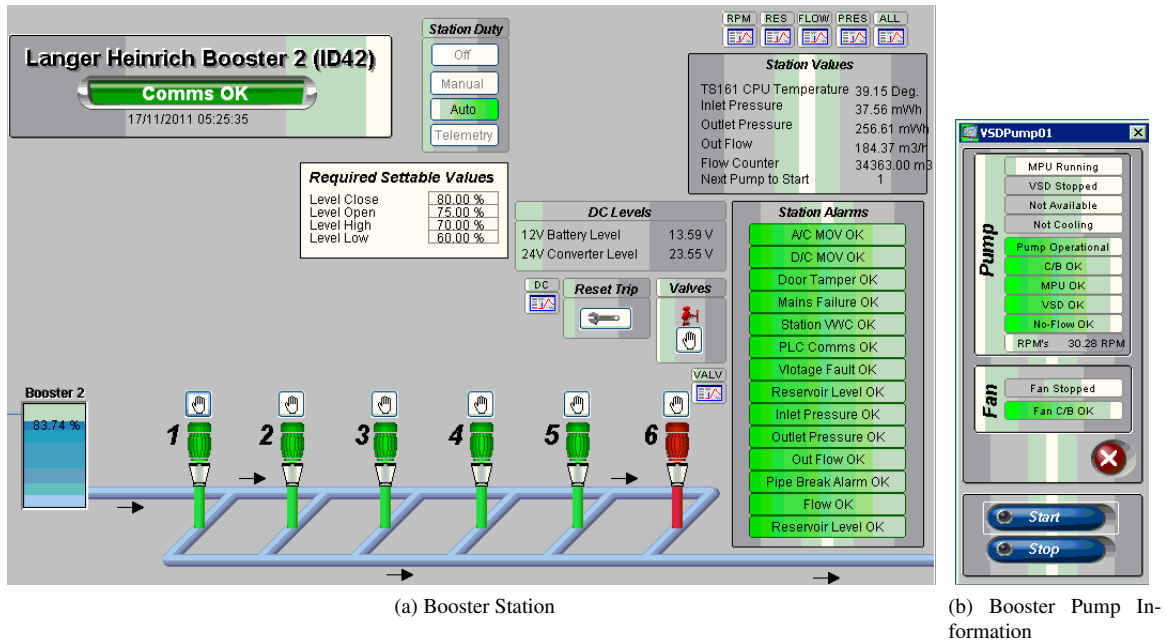


Figure 2.12: Typical Booster Pump Station taken from NamWater SCADA

## 2.4.6 Point to Point transmissions

Remote stations are set up to communicate with a base station which is usually situated at the telemetry server at the office of the water supply scheme. All eventing, logged and interval information is sent to the server. It might, however, be necessary to set up additional communication paths for specific data points. These data points will not be recognized by the base station, but by any other station that has been setup to recognize them. This transmission is usually set up as follows: Station A will be set up to broadcast specific data points at predefined intervals. Station B will be set up to recognize specific data points broadcast from Station A. This broadcast interval is usually every 10 minutes. Station B will recognize the frame sent from station A as a point to point frame, identify the data points in the frame as required and process them. Point to Point transmissions are usually set up between reservoirs, booster and borehole stations, but can also be set up between any other stations. Note that the term point to point is actually misleading, as the frame transmitted from a specific station can be used by various other stations. A typical example is given:

Reservoir A is supplied by 10 different boreholes. Each borehole requires the level from reservoir A to determine whether it is required to start its pump, stop its pump or stay in its current state. Reservoir A broadcasts its current level at intervals of 10 minutes. Each borehole will recognize the frame as a point to point frame from Reservoir A and obtain the data points within the frame, which in this case is only the reservoir level. The RTU station at each borehole will then make this new value available to the PLC, which will in turn take the necessary action according to the value it has received.

Point to Point transmissions add extra traffic to the radio network.

## 2.4.7 Remote Radio Diagnostics

To determine the status of each radio within the radio network, remote radio diagnostics can be used. This is supported by most new digital radio products (e.g. MDS SD4, Trio MR450). Radio diagnostics will include values such as the RSSI, VSWR, SNR, temperature and voltage. More advanced diagnostics can also include channel utilization and statistics. This information can help determine whether a specific link within the network is degrading and if so, why. The diagnostics can be set up either to be made available at regular intervals by allowing the base station to poll all radios within the

network or, in a less intrusive manner, by piggy backing it onto frames from the RTU. Therefore, network traffic will be increased either by more frames being sent over the network or by longer frames. Over long radio links, resulting in lower receiver signal levels, longer frames will be more prone to error and the first method mentioned above is advisable.

### 2.4.8 Critical Nature of Data Collected

From the above it is clear that while the information collected from remote stations is used to monitor and maintain the specific water supply scheme, it is also used for engineering, hydrological and operational analysis of the scheme. This information is vital to assist with network diagnostics, optimization, maintenance, operational planning, water resource monitoring and conservation.

### 2.4.9 Practical Example: Namib Water Supply Scheme

The current layout of the Namib Water Supply Scheme's radio network, is presented in Figure 2.13. Key characteristics are as follows:

- Currently, this system comprises 8 reservoir, 7 booster, 1 river, 4 repeater and 35 borehole stations
- There are 21 point to point transmission configurations, broadcasting their information once every 10 minutes.
- The system in total has about 3400 monitored data points. Each of these points is logged every hour.
- 275 Data points have been set up to transmit on change of state .
- 50 data points have been set up to transmit on 5% value change.
- All data points not set up to transmit immediately after an event or percentage change, have been set to log events to the RTU queue and transmit them only once every hour.
- A significant effort has been made to optimise the system as much as possible.

The respective data frame sizes can be estimated as follows:

- Average borehole station: 60 bytes
- Average booster station: 80 bytes
- Average reservoir station: 50 bytes
- Average repeater station: 40 bytes
- Average river station: 40 bytes
- Acknowledgement: 13 bytes

The average data frame size is therefore 60 bytes, with a acknowledgement frame size of 13 bytes. To determine the average event arrival rate of the current system we proceed as follows:

- 55 Stations log all data points once every hour, giving an arrival rate of  $\frac{55}{3600}$  /s
- 21 Point to Point transmissions once every 10 minutes, giving an arrival rate of  $\frac{21}{600}$  /s
- 275 Events transmit on a change of state, which can be estimated as once every hour, giving a rate of  $\frac{275}{3600}$  /s

- Finally, 50 events transmit on a 5% change in value, which can be estimated as once every hour, giving a rate of  $\frac{50}{3600}$  /s
- Operators will request stations for data on an estimated average of twice per hour per station, giving a rate of  $\frac{110}{3600}$  /s

Therefore, the total arrival rate at this stage will be  $\frac{55}{3600} + \frac{126}{3600} + \frac{275}{3600} + \frac{50}{3600} + \frac{110}{3600} = \frac{616}{3600} = 0.171$  arrivals per second.

Figure 2.14 shows the planned expansion of the current system, while Figures 2.15a,b and c expand the clustered view of stations in Figure 2.14.

The total system will be expanded to 111 stations with 15 reservoir, 10 booster, 1 river, 5 repeater and 80 borehole stations, with a further possible expansion which would include another 42 reservoirs, bringing the eventual station count to 153. If the point-to-point, state of change and percentage change transmissions are increased threefold, we get a new arrival rate of  $\frac{153}{3600} + \frac{372}{3600} + \frac{825}{3600} + \frac{150}{3600} + \frac{110}{3600} = \frac{1610}{3600} = 0.447$  arrivals per second. The system is to be expanded with a new repeater, which will introduce extra equipment latencies and propagation delays, which in turn will increase the total vulnerable time per transmission and increase the probability of collisions. If any of the streamlining features currently implemented are removed or reduced, the arrivals per second will increase dramatically, with a high probability of causing network failure.

Also note that radio diagnostics will be included in the new expanded network, which will also increase network traffic by a number of arrivals per second or by increasing the frame size of each station by a fixed amount.



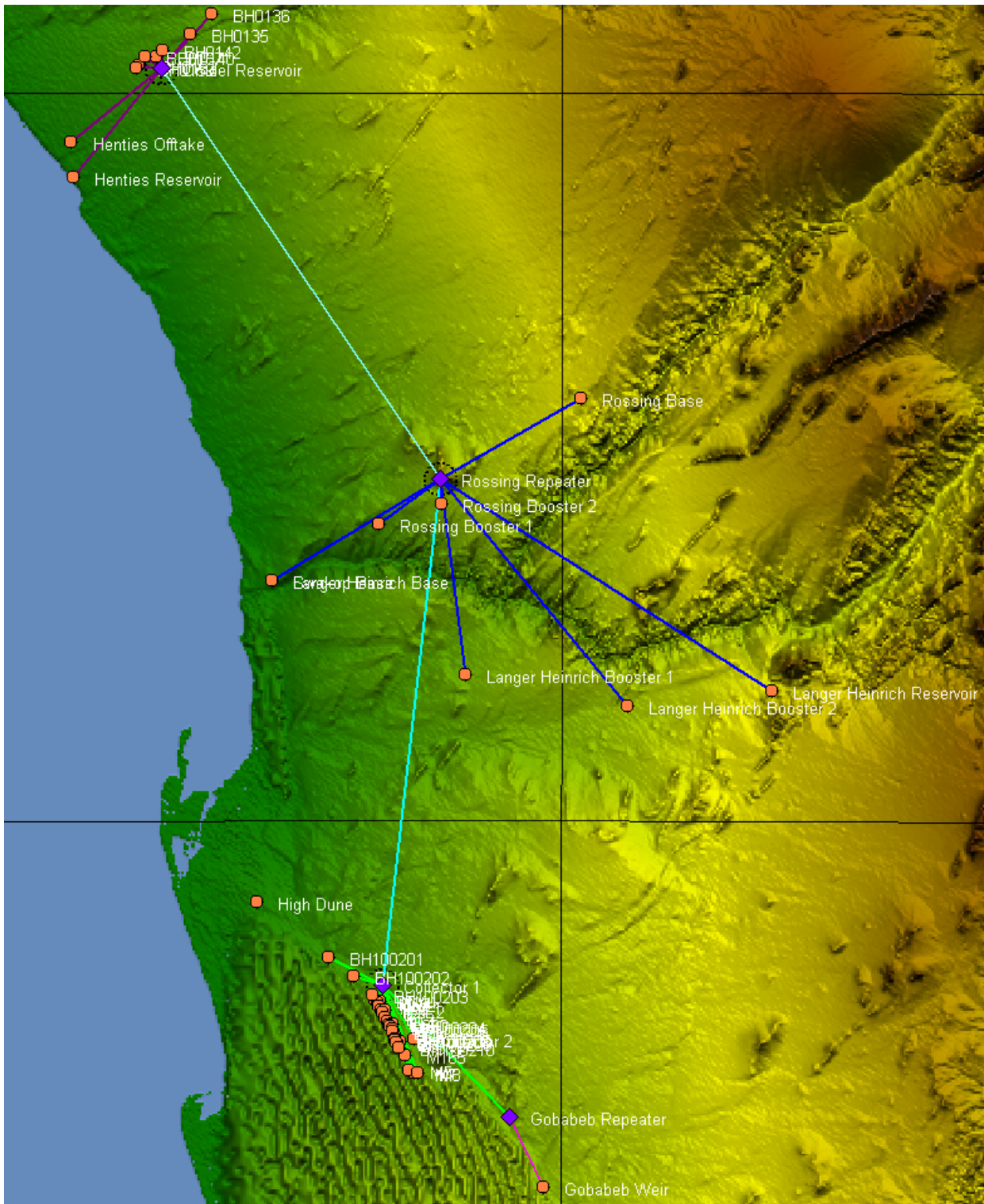
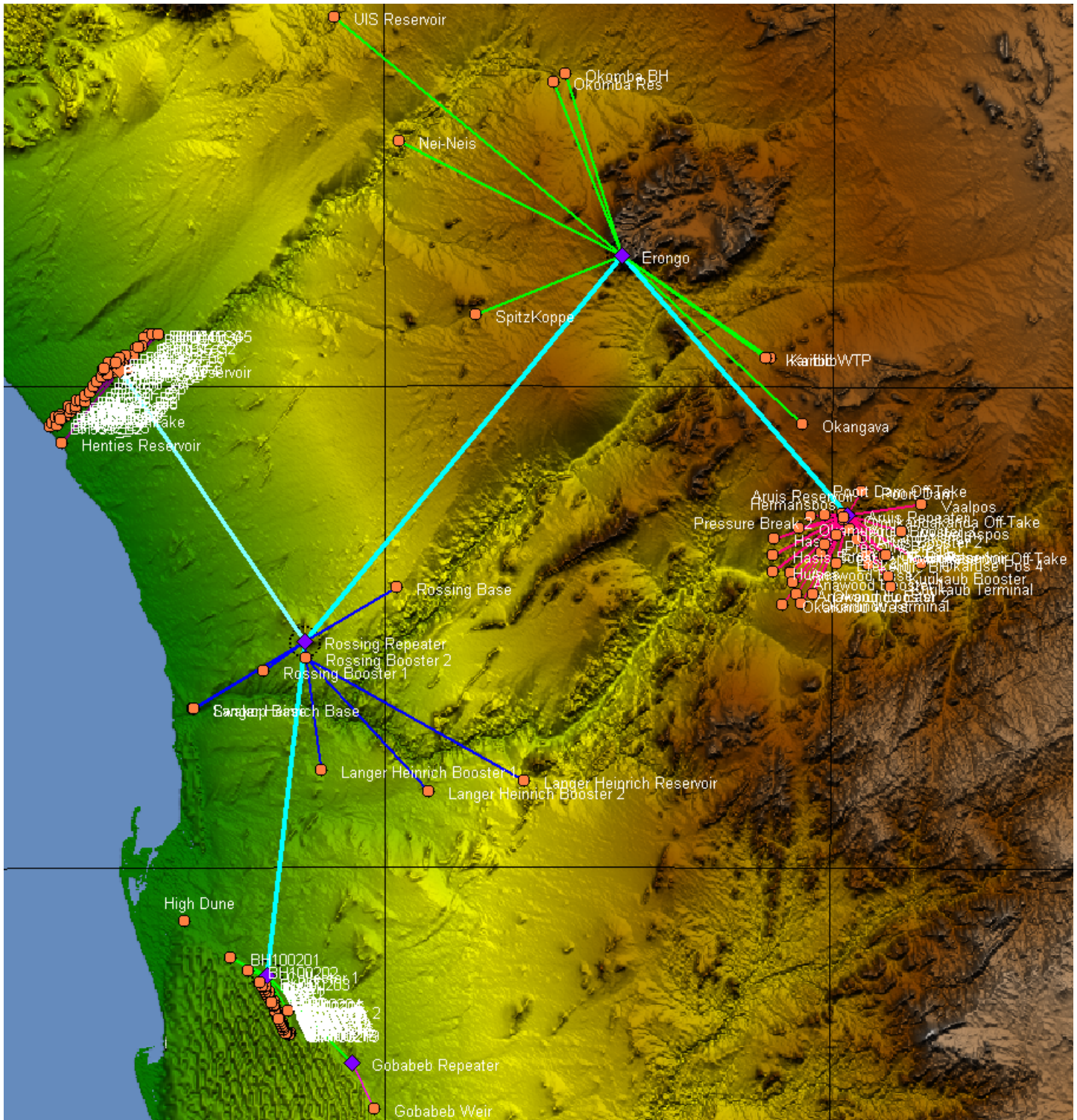
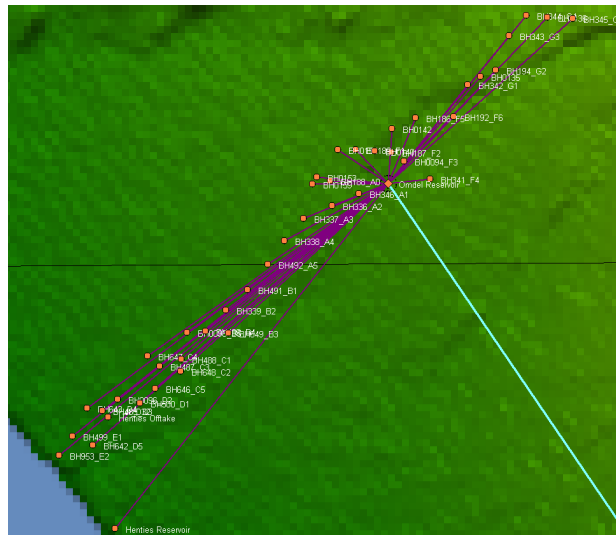


Figure 2.13: Current Namib Water Supply Scheme

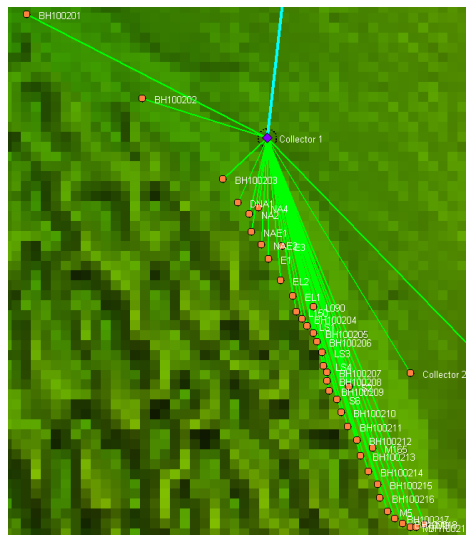


(a) Overview of System

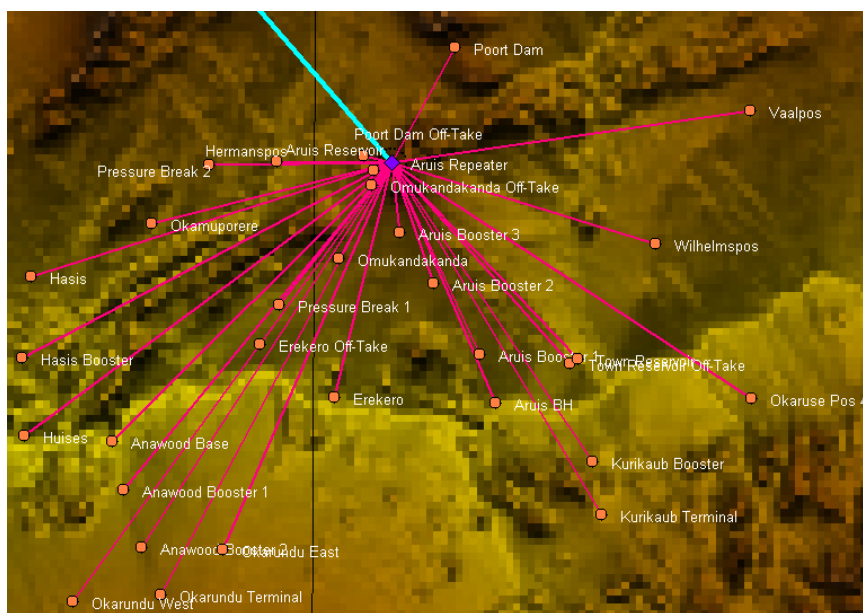
Figure 2.14: Future Namib Water Supply Scheme



(a) Omdel Borehole Scheme



(b) Kuiseb Borehole Scheme



(c) Erongo Water Supply Scheme

Figure 2.15: Future Namib Water Supply Sub-Schemes

## 2.5 Summary

This chapter has provided an overview of typical narrow band communication telemetry networks. The difference between analogue and digital radio communication is explained, highlighting the advantages of digital radio communication. The OSI model is explained, with emphasis being placed on the Data Link and Medium Access Control layers. Various methods of implementing the two layers are described. Protocols from these layers will later be used in further analysis. Finally, a practical telemetry system making use of narrow band communication network technology is discussed. The various monitoring stations within the network are outlined and the communication strategy between these stations defined. The resultant amount of traffic of the network is approximated, as well as the increase in traffic caused by future expansions of the system.

Simulation modelling to estimate the performance of practical networks is discussed in the next chapter.

## Chapter 3

# Simulation Modelling

### 3.1 Introduction

Computer simulation is an important tool for the modelling and analysis of complex systems, such as the one that is the subject of this thesis. The process follows the typical flow as depicted in Figure 3.1.

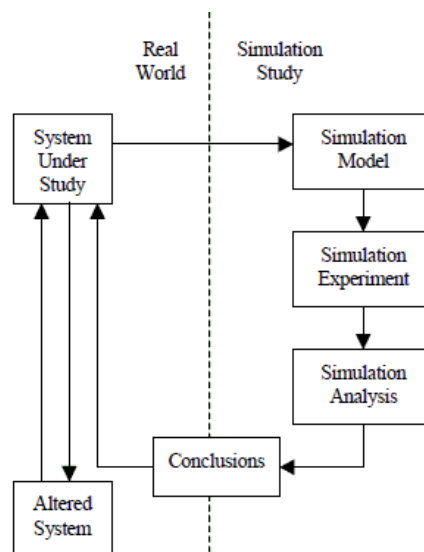


Figure 3.1: Simulation Study Schematic from [22]

#### 3.1.1 Motivation and Overview

In this thesis simulation modelling is applied for three specific reasons:

- To verify the results obtained from theoretical modelling of different MAC layer protocols which were considered for the type of system concerned.
- To provide a more accurate real life representation of the performance of the MAC layer protocols concerned.
- To expand the analysis of the protocols beyond the limitation of the theoretical models.

One of the advantages of simulation modelling is that it enables one to predict the performance of a specific theoretical model before attempting implementation. In the context of this thesis discrete event simulation will be used, which describes the change within a model when a specific event occurs. The principle of discrete event simulation falls in line with the modelling of event driven MAC layer protocols. Discrete event simulation maps the change of state within a particular system into a discrete event and assumes that nothing relevant has happened in between.

The relevance of the simulation model is determined by the correct interpretation of the particular assumptions made during the modelling process and must be approached with care, otherwise significant divergence might occur between the model and real life systems.

## **3.2 Basic Concepts in Discrete Event Simulation**

A good understanding of the concepts and tools within the simulation model is essential. Some of the more important considerations will be discussed in this section.

### **3.2.1 Discrete Event Simulation Basics**

The most important characteristic of discrete event simulation is the representation of the system state and its changes with time. Most discrete event simulations, including those in this thesis, are stochastic, meaning that their behaviour is intrinsically non-deterministic and the system's state is determined by the occurrence of random events. Modelling of non-deterministic systems is strongly reliant upon pseudo-random numbers or event generation as part of the simulation process.

### **3.2.2 Discrete Event Simulation Model Components and Techniques**

Core components and techniques forming the basis of simulation modelling architectures with relevance to this project, are as follows.

#### **3.2.2.1 Entities**

Entities or objects are simulation model components mimicking the real system. The state of an entity is determined by its inputs, which have a time variant behaviour. A RTU station is an example of such an entity. Each station will have certain attributes (e.g. data frame size, station number).

#### **3.2.2.2 Simulation Time**

Time procession within a model is referred to as model time. Model time passes within the model itself and is independent of real time e.g. the time required to run the model. For example, the performance of a specific MAC protocol over one day can be simulated within 10 minutes.

Discrete event models trace the behaviour of entities at specific points in time when events occur, such as new frame arrival or successful event servicing. By defining entities as objects, instances of the entities can be created (as in object orientation), each with its own "life cycle" tracing the object's condition over time.

In discrete event simulation you have a pseudo real time clock, with the tick intervals defined as required. Input changes and events occur only at these tick points. All results are recorded at these points.

The simulation clock is therefore completely independent of “real time” and, therefore, platform and CPU speed independent.

### 3.2.2.3 Event List

From the above it is clear that an event list needs to be kept to keep track of all the different events, their type and the time of occurrence. The event list is ordered according to ascending time. When a simulation run is executed it iterates through the event list (selecting the next event to occur, updating model time, change model state where necessary and adding new events to the event list caused by the current event occurring). This will continue until the event list becomes empty or something else terminates the cycle. A typical flow of the simulation process, is shown in Figure 3.2.

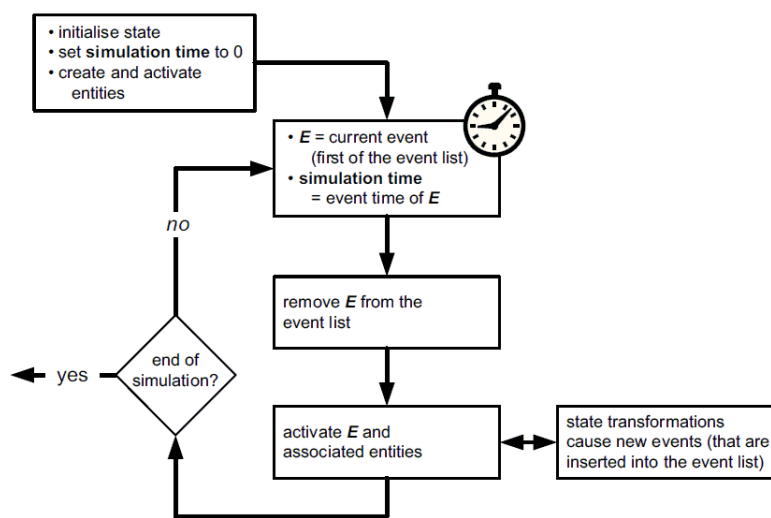


Figure 3.2: Discrete Event Simulation Flow taken from [26]

### 3.2.2.4 Statistical Counter and State Variables

Performance statistics must be available from the model. Statistical variables store the condition of the model at a specific point in simulation time. This would usually be done after a specific event has occurred. These variables can later provide useful statistical information on a monitored point, e.g. mean waiting time and number of stations in a queue. State variables are used to describe the state of the model at a specific point in time.

### 3.2.2.5 Frameworks

Discrete event simulation offers three different frameworks e.g. event, activity and process orientated. Figure 3.3 shows the relationships between events, activities and processes in an ordering system.

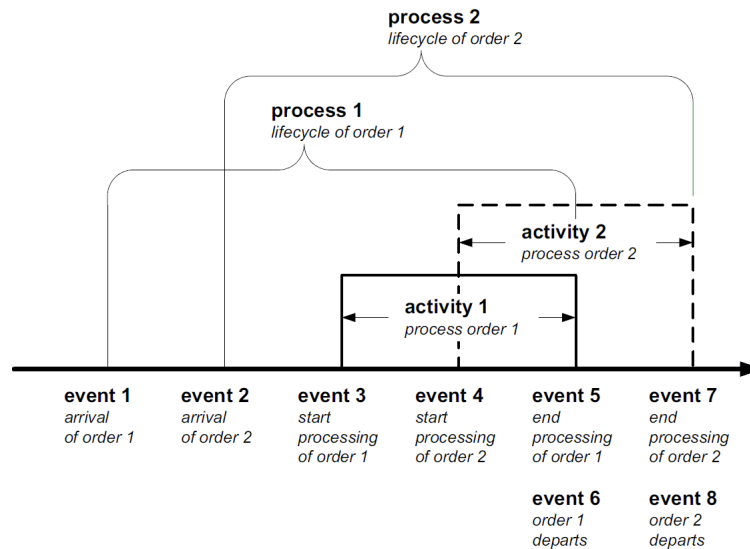


Figure 3.3: Relationship of Events, Activities and Processes from [26]

### 3.2.3 Simulating Random Events

Pseudo-random numbers, created by a pseudo-random number generator (PRNG), are central to discrete event simulation. These numbers are used to simulate events like frame arrivals, backoff and frame rates in a MAC layer simulation. The quality of the random numbers created and their statistical independence are very important. The most commonly used PRNG is the linear congruential generator (e.g. that is used for Java's random number generator).

### 3.2.4 Transient Phases and Steady State

The transient phase of a simulation model can be seen as the "warm-up" period of the simulation, meaning that the processes within the simulation are still dependent on time and not solely on the properties that they were assigned. This is known as a stationary stochastic process. In analysis of the MAC layer protocol used within a network this is important, as we want to see how the system performs in a steady state format, therefore where the system performance is dependent only on the specific properties of the network and not the time within the network.

### 3.2.5 Process-Oriented Simulation Modelling

Process-oriented simulations are the most common discrete event simulation modelling framework. A simulation model is generated from a set of interacting objects and a process is represented by a series of events related to a certain object. These events can be grouped into activities as discussed in 3.2.2.5. The life cycle of the process then consist of the activities of a certain entity that are grouped together.

The life cycles of different processes can run in parallel and the execution of each and the order in which they are executed is handled by the model's scheduler. Therefore, certain activities can be performed in one process life cycle up to a certain point when it requires input from another process life cycle, or requires another process life cycle to first complete before it can continue. Figure 3.4 shows a typical telemetry network containing RTU/remote and repeater stations and is a typical example of a network to be modelled.



During the active phase of a process, model state changes will occur. In a process life cycle a process can:

- modify the properties of entities (e.g. change the backoff time for a specific RTU station)
- generate new entities and their lifecycles (e.g. create new RTU stations which have data to be sent)
- activate other process lifecycles that have been passive (e.g. a RTU station being activated after its backoff time has expired)
- change or cancel active phases of active entities (e.g. an interrupt can be called inhibiting the active processes)
- deactivate itself and pass control to the model scheduler, which will activate another process (e.g. a RTU station process determines that the communication channel is busy and goes into a passive state for a specific time)
- terminate its own life cycle or other process lifecycles (e.g. a RTU station's data has been processed, therefore its life cycle has completed and it terminates itself)

By updating the model clock, time passes during the activities of a process life cycle. There are different ways in which time can be passed. An activity within a process's life cycle can be required to wait for a specific fixed time, go into a passive state where it waits to be activated, or be activated by another activity in another process life cycle.

The management of process lifecycles is done by the scheduler. It is the heart of any discrete event simulation modelling tool and controls when processes are activated and in which order, by making use of the event list. When a process life cycle is active, the scheduler waits for control to be handed back. When this happens it will access the event list and decide which processes must be activated next. The scheduler is a behind the scenes function and doesn't consume any modelling time. It merely manages the modelling time and the processes that run within this time.

To describe the high level flow of processes and their interaction with each other, UML can be used, which is then implemented using object orientation by identifying different entities, their properties and behaviour.

The simulation models in this thesis take a process-oriented approach and more information on this approach can be found in [26].

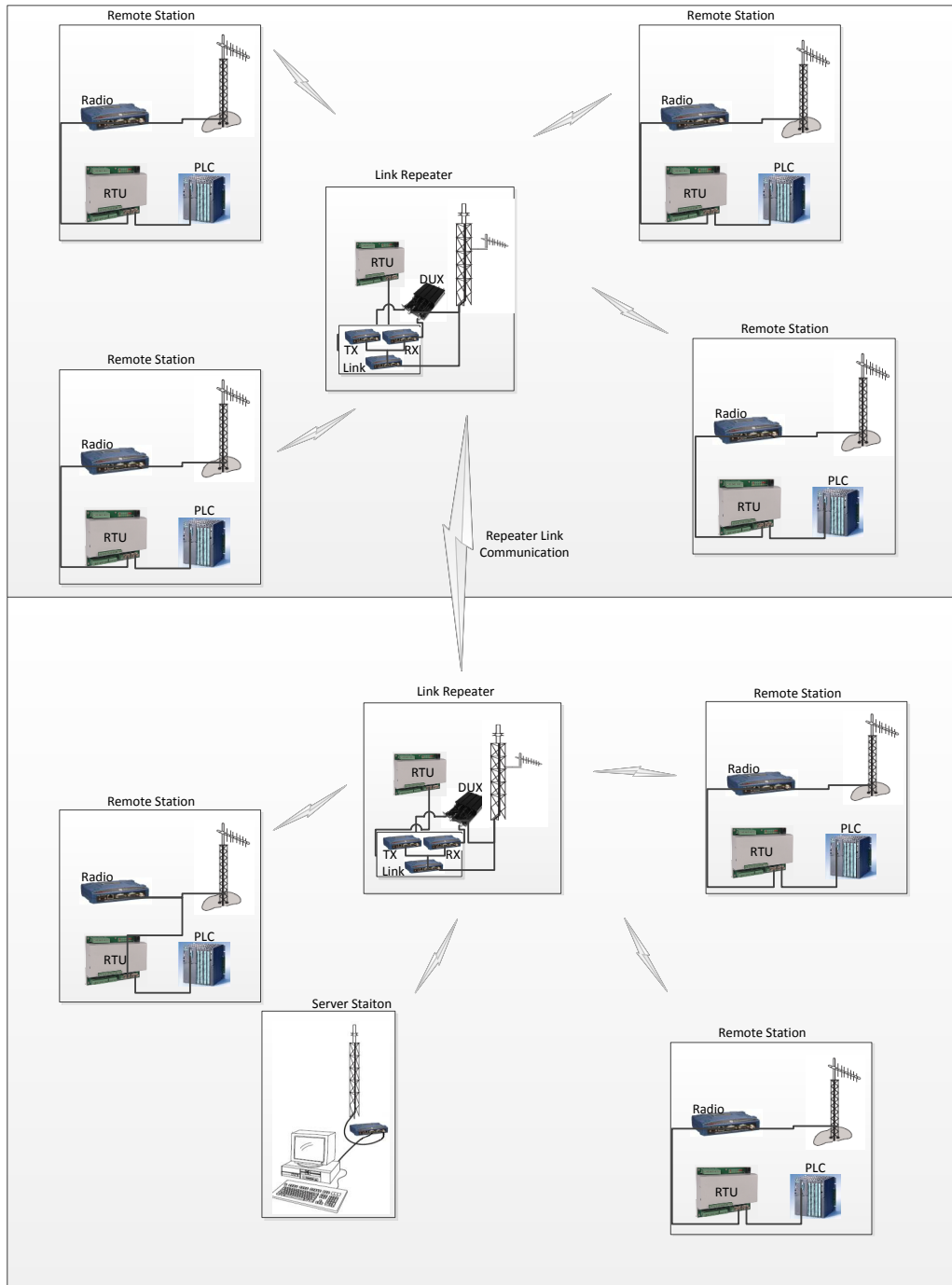


Figure 3.4: Telemetry Network with Two Repeaters

### 3.3 DESMO-J

#### 3.3.1 Overview

DESMO-J (Discrete Event Simulation Modelling in Java) is an object orientated software development framework for discrete event simulation. DESMO-J contains components that can be selected and customized to deliver a simulation

model as required. These components fall within two categories:

- A Black box framework, which only requires initiation of predefined components. Modifications are restricted to simple parameter changes.
- White box frameworks offers abstract classes which must be adapted and customized to serve the needs of a specific model.

DESMO-J is framework software, requiring setup by the user in accordance with the model requirements. DESMO-J makes use of both black and white box components in its framework.

To support discrete event simulation modelling, a framework must contain the following components:

- Entities, which are used to model real world objects, their properties, behaviours and relationships with other entities
- Stochastic distributions, to model random behaviour that might be required within the model
- Data collectors, which are used to collect statistical information within the model for later analysis of the simulation
- Event list, which stores the pending events or process activations
- Scheduler is the model executive used to control the execution of the model, which includes the interactions between events, activities and processes.

DESMO-J supports these components in white and black box components. The black box components provided are:

- A discrete event simulation infrastructure that includes the event list, simulation clock and event scheduler. This functionality is stored within the Model class.
- A means of storing simulation results and data within files (e.g. reports, event traces, debugging information and errors). This functionality is found in the Experiment class.
- Components for modelling different queues and random processes.
- Components for the collection of statistical data.
- Process-Oriented modelling components

DESMO-J allows the basic black box features to be used to develop model specific entity types and behaviour. This is done by allowing generic Model, Entity, Event and SimProcess objects which are sub classed within the user defined model as required. This feature provides white box functionality. Figure 3.5 shows DESMO-J's class hierarchy with the term "hot-spot" indicating white box features that require to be detailed by the user. Black box components only need to be initiated.

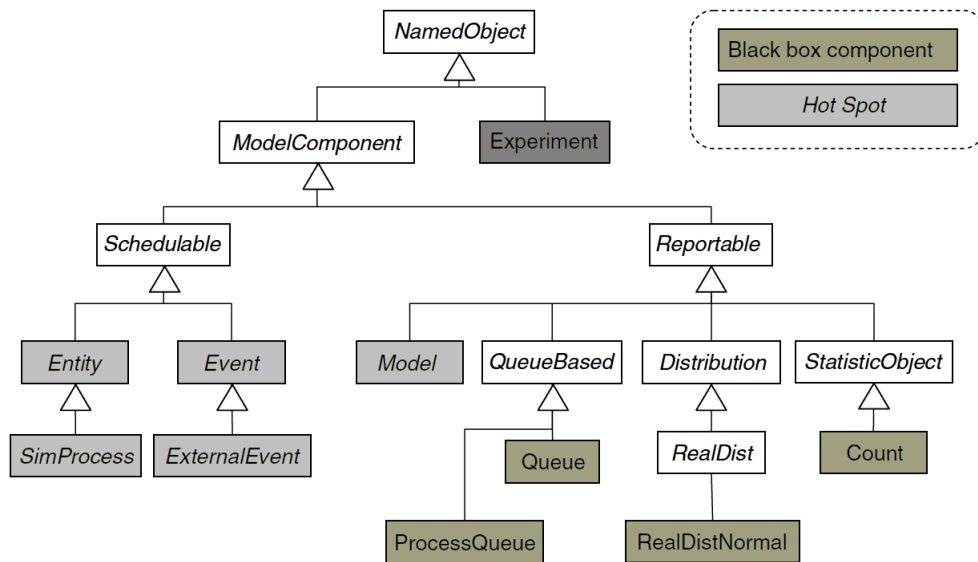


Figure 3.5: DESMO-J Class Hierarchy taken from [26]

### 3.3.2 Process-orientated modelling

Processes can be seen as active entities which encapsulate both behaviour and properties of the entity. The properties are defined as specific data structures and the behaviour is described within the lifecycle method that must be contained in the process class. The user defined process is an extension of the DESMO-J SimProcess class and must implement the lifeCycle() method, as indicated earlier. A process is either active, causing the model state to change, passive and waiting for another process to activate it, or delayed for a specific period of time. Processes can also be interrupted by other processes or by themselves and will resume, when activated again, from the point where they were interrupted. The hold(), passive() and activate() methods are used within DESMO-J to implement above-mentioned functionality. Figure 3.6 shows how these methods are used within DESMO-J.

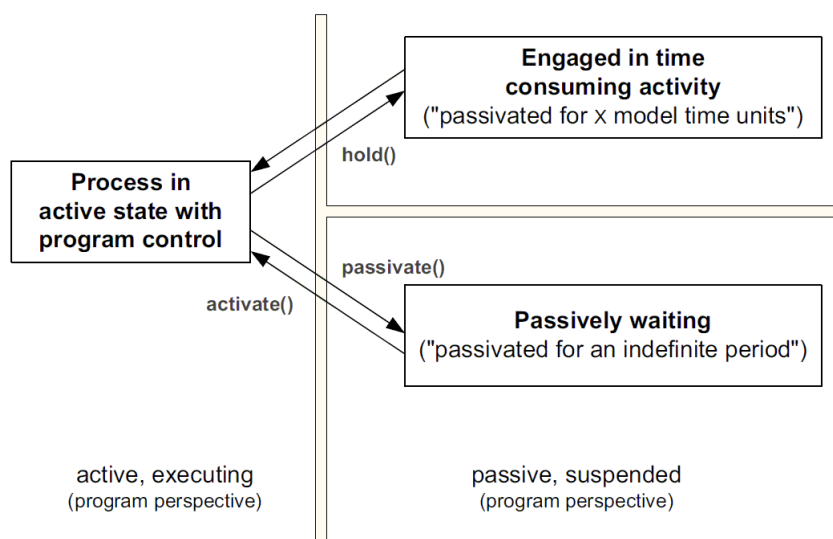


Figure 3.6: Description of the use of Hold, Passive and Activate Methods from [26]

### 3.4 Summary

This Chapter has given an introduction to simulation modelling, with the main focus placed on discrete event simulation models. The basic elements required of a discrete event simulation model are outlined and discussed. DESMO-J is introduced as a discrete event simulation API implemented on the Java platform. A layout of DESMO-J is given and the most important elements within the API discussed. Finally, process-orientated modelling is defined and explained and is the basis on which all models have been built within this thesis.

In the next chapter queueing theory is introduced, with emphasis placed on the Markovian process.

## Chapter 4

# Queuing Theory

All MAC layer protocols are, in essence, based upon queues. Data frames waiting to be serviced or waiting for acknowledgements can be seen as a queue, as each is waiting for a previous frame to complete its transaction before it can continue with its own. Certain queuing principles are used throughout the analysis of MAC layer protocols and require a good understanding to build representative simulation and theoretical models. This chapter will introduce some of these principles as applied in the construction of the models.

### 4.1 Basic Queuing Principles

The most basic model used to analyse queues is shown in Figure 4.1. A queue is supplied by an infinite or finite source population, where each source arrives at the queue at a specific rate. The arrival rate of all sources gives an overall arrival rate of  $\lambda$  arrivals per second. The arrivals will move into the queue and will be serviced by the server according to a specific strategy which is either First in First Out (FIFO), where the first arrival to the queue is the first to be serviced (most commonly used) or Last in Last Out (LIFO). The server will service the arrivals at a rate of  $\mu$  arrivals per second.

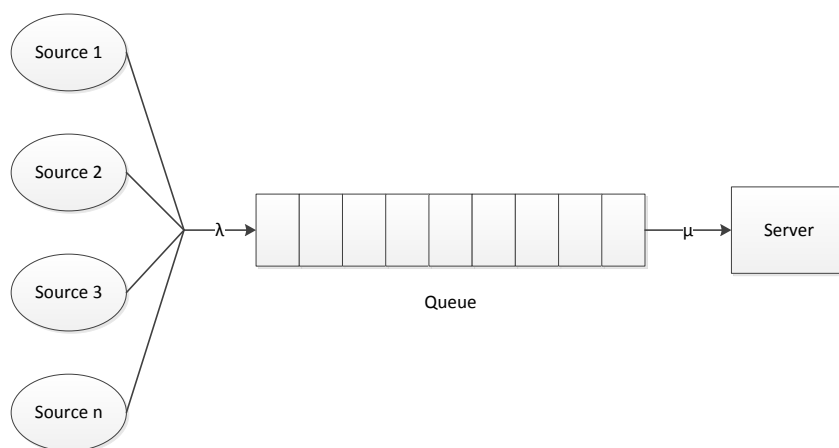


Figure 4.1: Basic Queue Model

Queues like the one in Figure 4.1 can be characterized by six parameters [30]. These parameters are:

### **Source Population**

The source population comprises of the objects that will join the queue. Cars arriving at a service station is an example of an infinite source population, as different cars will arrive at the service station and their arrival is not limited or bounded by specific parameters. RTU stations within a communication network is an example of a finite source population, as there are only a finite number of remote monitoring points within a network. Each station will have events that occur at certain intervals and will require the use of the communication channel to be transferred to the base station. If each event is taken as an arrival and requires the use of the communication channel alone, this again becomes an infinite source population, but if each event is seen as part of a station buffer, the contents of which are transmitted as one packet, then the finite source assumption holds true. Note that when an event occurs at a station an arrival has taken place and the specific station is no longer part of the source population that can create new arrivals. Only when the station has been able to transmit its information successfully to the base station, can it rejoin the source population. This is important because, as the source population increases and decreases, the arrival rate from the population follows suit.

### **Arrival process**

The arrival process is the overall rate at which new arrivals join the queue. In an infinite source population, the average arrival rate will stay constant, but in a finite source population this rate will vary with the number of stations already in the queue. If the example of a RTU station source population is taken again, each station as described above can have events occurring which are placed in a buffer and transmitted as one package. Therefore the moment that the buffer receives a new entry, the station will try to transmit the entry and the station is in the queue and no longer part of the source population that can create new arrivals. Therefore the source population has decreased, which means the arrival rate has decreased as well. The moment the station in the queue has transmitted its packet, it is no longer in the queue and joins the source population again, meaning that the arrival rate has increased again.

The probability that an arrival will occur can be described by a specific Probability Density Function (PDF). These can vary depending on the arrival process. A Poisson PDF can be used to describe the arrival process of events at remote stations in a telemetry network, as will later be discussed.

### **Server process**

The server process determines the rate at which arrivals within the queue are serviced. The arrival discipline, together with the server discipline, will determine the queue length in terms of the relationship of queue parameters, as later discussed. The probability density function of the server process is important and determined by the specific network that it is applied to. A Poisson PDF is commonly used, as will be discussed in Section 4.2.

### **Number of servers**

In the MAC layer protocols used in telemetry applications, there is only one true server. In other applications, if the number of servers is increased the total service time will decrease, which in turn means that arrivals will be serviced faster.

### **Queuing discipline**

Queues can be serviced in different ways. These include FIFO and LIFO strategies to which further prioritization can be added. The type of MAC layer protocol used will determine the type of strategy implemented, which in most cases is FIFO. This can become difficult though, as will be seen in the CSMA protocol. Within this protocol a specific station might have entered the queue first due to an event occurring at the station. Although the station entered the queue first, it might not be the first to be serviced as it might go into backoff or timeout due to collisions or corrupt data.

## Queue length

The queue length has a definite influence on the stability of a specific network. The queue length is determined by the particular type of queue and its operational parameters. Queue buffer space can be finite, or infinite. In the first case (the most general one) arrivals will simply disappear if the buffer is full. Some common types of queues (that will be used in this thesis) and their associated analysis, will be discussed in Section 4.3

### 4.1.1 Kendall's notation

Kendall's notation is used to describe the queuing process and takes the form  $A/B/m/k/l$ , where:

A	Describes the arrival process
B	Describes the service process
m	Is the number of servers used in the system
k	Is the system capacity
l	Queuing discipline

All MAC layer protocols to be described are  $M/M/1/N/FIFO$  queuing systems where the M stands for Markovian, N stands for the maximum queue length and FIFO is First in First Out. This notation is usually written shorthand as  $M/M/1$  where the last two terms are assumed as infinite queue length and a FIFO strategy. Other common notations include G (General), D (Deterministic), GI (General Independent), H (Hyper-exponential) and E (Erlang).

### 4.1.2 Little's Theorem

Little's Theorem states that the average number of customers ( $N$ ) in a stable system is equal to the average effective arrival rate ( $\lambda$ ) multiplied by the average time a customer spends in the system ( $T$ ), as shown in Equation 4.1.

$$N = \lambda T \quad (4.1)$$

This result is somewhat surprising and very general. It doesn't rely on the specific distribution of the particular service and arrival processes, the number of servers, or queuing discipline and can be applied to any queuing system with the proper interpretation of  $N$ ,  $\lambda$  and  $T$ . Further detail on the implementation of this theorem within the theoretical modelling of the different MAC layer protocols will be given in the next chapters.

## 4.2 Poisson Process

The Poisson process closely resembles various physical phenomena and is considered a good model for any arriving process which involves a large number of similar and independent sources. A large number of common queueing systems assume that the arrival and service processes exhibit a Poisson distribution. The inter-arrival and inter-service times are, therefore, exponentially distributed [5]. The Poisson distribution is given as

$$P[X(t) = k] = \frac{(\lambda t)^k}{k!} e^{-\lambda t} \quad (4.2)$$



where:

$\lambda$  is the rate of events occurring

$\lambda t$  is the mean of the Poisson random variable, i.e. the average of the number of occurrences within the interval  $(0, t)$ .

It is also possible to see the Poisson process as a specific case of Bernoulli trials. If time interval  $(0, t)$  is made discrete and divided into  $n$  slots, then each event occurring within the time interval is assumed to take place within one of the  $n$  time slots and can be viewed as a success. The probability of  $k$  successes within  $n$  time slots can then be viewed as

$$P[k \text{ successes in } n \text{ time slots}] = \binom{k}{n} p^k (1-p)^{n-k} \quad (4.3)$$

If the number of time slots  $n$  is increased, while the probability of a successful occurrence of an event is decreased to allow the average arrival rate within the time interval  $(0, t)$  to stay constant,  $np = \lambda t$ , then Equation 4.3 can be re-written as:

$$P[k \text{ successes in } n \text{ time slots}] \equiv P[k \text{ arrivals in } (0, t)] = \lim_{n \rightarrow \infty} \binom{k}{n} \left(\frac{\lambda t}{n}\right) \left(1 - \left(\frac{\lambda t}{n}\right)\right)^{n-k} \quad (4.4)$$

Solving for the limit, we again get the Poisson process as shown in Equation 4.2.

## 4.2.1 Properties

The Poisson process properties is important in determining the arrival and service processes within the models of the MAC layer protocols to be analysed.

### 4.2.1.1 Superposition Property

The superposition property says that if there are  $k$  independent Poisson processes  $A_1 \dots A_k$  the sum of these processes will also be a Poisson process with rate  $\lambda$  equal to the sum of rates  $\lambda_1 \dots \lambda_k$  [5].

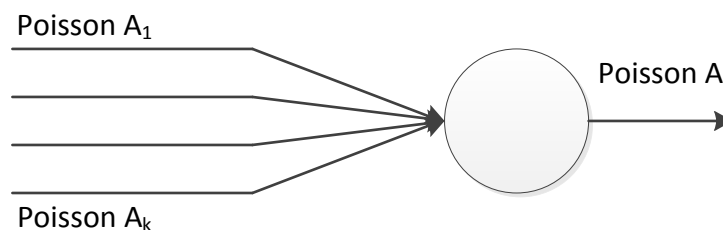


Figure 4.2: Poisson Superposition

### 4.2.1.2 Decomposition Property

The decomposition property is just the reverse of the superposition property, where Poisson process  $A$  can be divided into  $k$  independent Poisson processes using probability  $p_i$  ( $i = 1 \dots k$ ).  $A_i$  will then have an average arrival rate of  $\lambda_i = p_i \lambda$ , where  $\lambda$  is the arrival rate of Poisson process  $A$  [5].

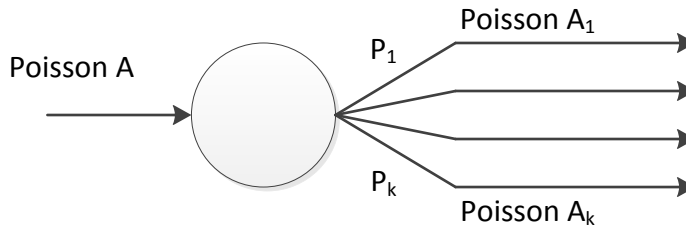


Figure 4.3: Poisson Decomposition Property

### 4.2.1.3 Exponentially Distributed Inter-Arrival Times

The exponential process and Poisson distribution mirror each other. If the inter-arrival time of a source population is exponentially distributed, then the number of arrivals in a time interval  $(0, t)$  will be given by the Poisson distribution and the arrival process will be Poisson. It can be shown that the probability distribution for the arrivals of sources with new events in a time interval  $(0, t)$  is:

$$\begin{aligned}
 P(\tau \leq t) &= 1 - P(\tau > t) \\
 &= 1 - P[X(t) = 0] \\
 &= 1 - e^{-\lambda t}
 \end{aligned} \tag{4.5}$$

where:

$P(\tau \leq t)$  Probability that an arrival occurs in time  $(0, t)$

$P(\tau > t)$  Probability that no arrivals occur in time  $(0, t)$

$P[X(t) = 0]$  Probability that exactly 0 arrivals will occur in a time interval  $(0, t)$

which is the exponential distribution. The derivation of Equation 4.5 is provided in Appendix C.1.1.

## 4.3 Markov Process

To model a specific real time process a mathematical model known as the stochastic (random) process can be used. If a stochastic process has the property that future conditions of the process are dependent only on the most recent condition,

and how the process arrived in the current condition is irrelevant, the stochastic process is known as a Markov process. A Markov Process links together different discrete states in which a real life process can find itself and this is known as a Markov Chain. This is best shown by making use of an example that is used in [5], if a coin toss experiment is taken. There are only two outcomes, head(1) or tail(0) and these can be described by random variable  $X_k$ . Random variable  $Y_k$  describes the accumulation of heads or 1s. The system starts at state  $Y_k = 0$ . It can then be said that  $X_k$  describes a normal stochastic process whereas  $Y_k$  is a special case stochastic process known as the Markov chain which is described as  $Y_{k+1} = Y_k + X_k$ . Which means that the next value of  $Y$  is determined by its current value as well as the next value of  $X$ . This shows the dependency described above.

Markov chains can be divided into discrete time and continuous time. This study will only look at the continuous time Markov chain. For more information on the discrete time Markov chain, please refer to [5].

If the transition from one state to another can happen at any instant of time it is known as a Continuous Time Markov Chain. In a Continuous Time Markov Chain the past state of the chain is described by its present state, but future states depends only on the current state. The single most important continuous distribution in a Continuous Time Markov Chain is the exponential distribution, which has a memoryless property. This property is due to the relationship between the Poisson process and the exponential distribution of inter-arrival times. A very important property of the Markovian process is that what has occurred in the past has no effect on occurrences in the future [5]. This is fundamental to the analysis of the majority of queuing systems. This property is shown below:

$$\begin{aligned} P(\tau \leq t_0 + t \mid \tau > t_0) &= P(\tau \leq t) \\ &= 1 - e^{-\lambda t} \end{aligned} \quad (4.6)$$

Equation 4.6 shows that the conditional distribution of inter-arrival times, given that a certain amount of time has passed, is the same as the unconditional distribution. The derivation of Equation 4.6 is provided in Appendix C.1.2.

The transition from one state to another can happen at any time and it is thus necessary to describe how long a process has stayed in a specific state before transitioning to the next one. Taking above into account, the Continuous Time Markov Chain must satisfy the following conditional probability [5],

$$P[X(t_{k+1}) = j \mid X(t_1) = i_1, X(t_2) = i_2, \dots, X(t_k) = i_k] \quad (4.7)$$

$$= P[X(t_{k+1}) = j \mid X(t_k) = i_k] \quad (4.8)$$

where  $X(t_{k+1})$  is the future stochastic process which is only dependent on the current stochastic process,  $X(t_k)$ , and not on the past processes. This is known as the transition probability, which describes the probability that Continuous Time Markov Chain will be in a specific state. It is also required to know with what probability or rate the chain will leave state  $i$  and move to state  $j$  ( $i \neq j$ ) in the next infinitely small time  $\Delta t$ . This can be described as

$$p_{ij}(t, t + \Delta t) = q_{ij} \Delta t \quad (4.9)$$

where  $q_{ij}$  is the instantaneous transition rate of leaving state  $i$  for state  $j$ . Generally  $q_{ij}$  is a function of time but for the requirements of the text it is assumed that  $q_{ij}$  is independent of time, making the Markov chain homogeneous. It makes sense that the total rate at which state  $i$  is left for any other state  $j$  is then:

$$\sum_{i \neq j} q_{ij} \quad (4.10)$$

### 4.3.1 Birth-Death Process

A Markovian queuing system that is characterized by a Poisson arrival process (only one arrival can occur in time  $\Delta t$ ) and exponentially distributed service times (only one arrival can be serviced in time  $\Delta t$ ) can be characterized by the Birth-Death Process, therefore the Birth-Death Process is a special case of the Continuous Time Markov Chain in which the chain can only transition to its neighbouring state from its current state, as shown in Figure 4.4.

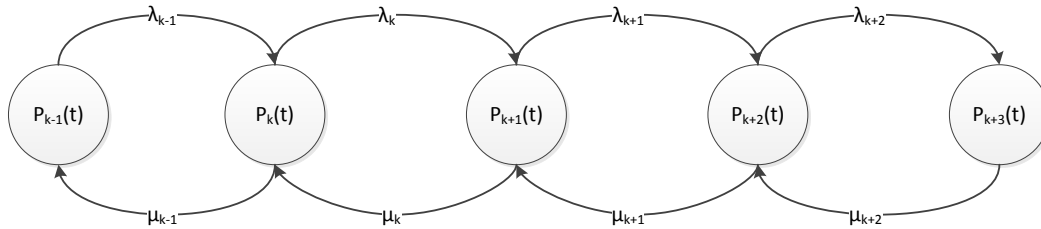


Figure 4.4: Birth Death Process State Transition Diagram

When a queue population size is  $k$  at time  $t$  then it will be in state  $k$  with probability  $P_k(t)$ . A move from state  $k$  to state  $k + 1$  signifies a birth and the rate at which this transition occurs is  $\lambda_k$ , where as the move from state  $k$  to state  $k - 1$  signifies a death and the transition between the two states occurs at a rate  $\mu_k$ . From Equation 4.9

$$\lambda_k = q_{k,k+1} \quad (4.11)$$

$$\mu_k = q_{k,k-1} \quad (4.12)$$

The Birth-Death Process principle will be used in Single Queue Markovian Systems described in the following subsection.

### 4.3.2 Single Queue Markovian System

A special case of the Continuous Time Markov Chain is the Birth-Death Process, where transitions can only be made to the neighbouring states of the current state. A queuing system which consists of a single queue can be modelled making use of the Birth-Death Process.

In a Birth-Death Process where the system only has one queue and an infinite source population, it is described by Kendall's notation as a  $M/M/1$  queue. Within this system the Poisson arrival rate ( $\lambda$ ) and the exponential service time ( $\mu$ ) are not dependent on the number of source events within the queue, therefore it is state independent and  $\lambda_k$  and  $\mu_k$  can be changed to  $\lambda$  and  $\mu$ . An arrival to the queue is seen as a birth and an arrival that has been serviced and leaves the queue is seen as a death.

In these queuing systems the interest usually lies in the steady-state performance analysis of the queue, therefore the state where the Birth-Death Process is in equilibrium.

### 4.3.2.1 Global and Local Balance and M/M/1 Queue Analysis

The probability  $P_k$  describes the probability that a system will find itself in state  $k$ . In other words, it is the fraction of normalized time that the system will find itself in state  $k$ . Therefore  $\lambda P_k$  can be seen as the expected rate of transition from state  $k$  to  $k+1$  whereas  $\mu P_{k-1}$  can be seen as the the expected rate of transition from state  $k$  to  $k-1$ . This quantity is known as the stochastic or probability flow from state  $k$  to  $k+1$  for  $\lambda P_k$  and the same applies for  $\mu P_{k-1}$ . For equilibrium conditions the stochastic flow out of state  $k$  must equal the stochastic flow into state  $k$  and can be written as follows [5]:

$$(\lambda + \mu) P_k = \lambda P_{k-1} + \mu P_{k+1} \quad (4.13)$$

Figure 4.5 shows the global balance concept.

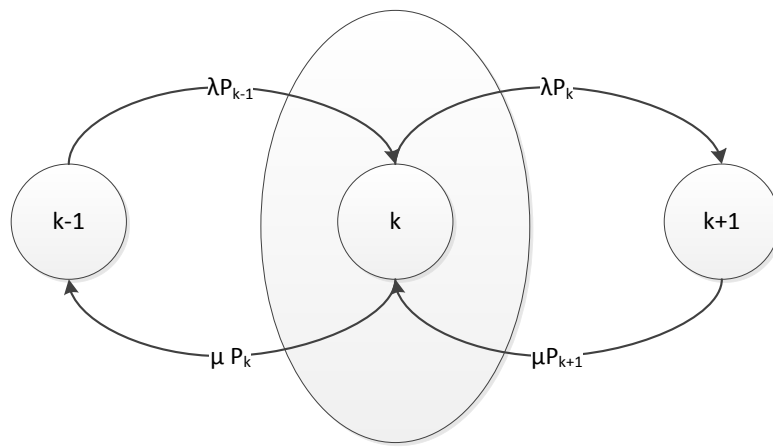


Figure 4.5: Global Balance

A special case of the Global Balance is the Local Balance shown in Figure 4.6.

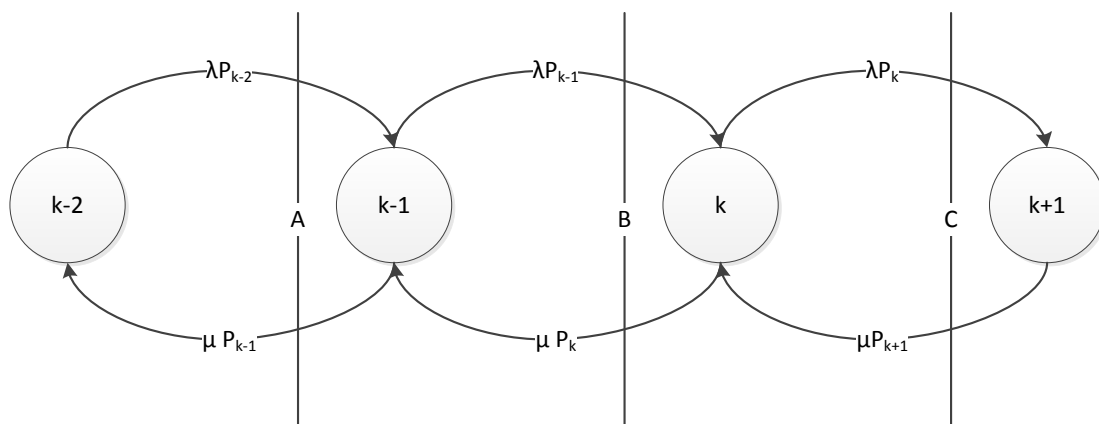


Figure 4.6: Local Balance

Creating an equilibrium equation for the boundary B, we get:

$$\lambda P_{k-1} = \mu P_k \quad (4.14)$$

Equation 4.14 captures information from all states up to state  $k$ . This can be shown as follows: If boundary B is as defined in Equation 4.14, then boundary A will be:

$$\lambda P_{k-2} = \mu P_{k-1} \quad (4.15)$$

Rearranging the terms in Equation 4.14 and 4.15,  $P_k$  is defined as follows:

$$\begin{aligned} P_k &= \frac{\lambda}{\mu} P_{k-1} \\ &= \frac{\lambda}{\mu} \left( \frac{\lambda}{\mu} P_{k-2} \right) \\ &= \left( \frac{\lambda}{\mu} \right)^2 P_{k-2} \\ &= \left( \frac{\lambda}{\mu} \right)^k P_0 \\ &= \rho^k P_0 \end{aligned} \quad (4.16)$$

where  $\rho = \frac{\lambda}{\mu}$ .

#### 4.3.2.2 Performance Calculations

From Equation 4.16, a useful way is provided to represent probability  $P_k$  in terms of  $P_0$ . For a queuing system, the system must be in one of the  $k$  states that exists, which means that a 100% probability exists that the system is indeed in one of these states, therefore the summation of all probabilities of being in a specific state up to  $k$  must be 1:

$$\sum_k P_k = 1 \quad (4.17)$$

Substituting Equation 4.16 into 4.17 and making use of the geometric series [36], specifically Equation 4.19:

$$\sum_{n=1}^{\infty} ar^{n-1} = \frac{a}{1-r}, \quad |r| < 1 \quad (4.18)$$

$$\sum_{n=0}^{\infty} ar^n = \frac{a}{1-r}, \quad |r| < 1 \quad (4.19)$$

$$\sum_{n=1}^{\infty} a(nr^n) = \frac{a}{(1-r)^2}, \quad |r| < 1 \quad (4.20)$$

it can be found that:

$$\begin{aligned} \sum_{k=0}^{\infty} P_k &= \sum_{k=0}^{\infty} \rho^k P_0 = 1 \\ \frac{P_0}{1-\rho} &= 1 \\ P_0 &= 1 - \rho \end{aligned} \quad (4.21)$$

$$\begin{aligned}
 P_k &= \rho^k P_0 \\
 &= \rho^k (1 - \rho)
 \end{aligned}
 \tag{4.22}$$

Having probability  $P_k$  it is possible to determine the number of arrivals in the queue. If the probability of  $k$  customers in the queue is  $P_k$ , then  $L_q = kP_k$  is the queue length for the probability that there are  $k$  customers in the queue. Summing all probabilities will provide the actual queue length:

$$\begin{aligned}
 L_q &= \sum_{k=0}^{\infty} kP_k \\
 &= \sum_{k=0}^{\infty} k\rho^k (1 - \rho) \\
 &= \rho (1 - \rho) \sum_{k=0}^{\infty} k\rho^{k-1} \\
 &= \frac{\rho(1 - \rho)}{(1 - \rho)^2} \\
 &= \frac{\rho}{(1 - \rho)}
 \end{aligned}
 \tag{4.23}$$

where the geometric series as given in Equation 4.20 has been used. It can be seen from the geometric series boundaries that  $\lambda < \mu$  for the series to converge, else the series will diverge and the queue will grow boundlessly. To determine the average waiting time, Little's Theorem as in Equation 4.1 can be used:

$$\begin{aligned}
 T &= \frac{L_q}{\lambda} \\
 &= \frac{L_q}{\frac{\rho}{(1 - \rho)}} \\
 &= \frac{1}{\mu - \lambda}
 \end{aligned}
 \tag{4.24}$$

where  $T$  is the total average waiting time for a specific arrival and service time per source event. This is very important.

### 4.3.3 Finite Single Queue Markovian System for use in Telemetry Systems

The above presents the performance analysis of an infinite single server queue using Markovian processes. In telemetry networks the source population, which from now on will be referred to as stations, is bounded and not infinite. The theory still applies and the required adjustments for the implementation of a Finite Single Queue Markovian System will be covered in this section. Firstly, the Kendal notation now changes from a  $M/M/1$  queue to a  $M/M/1/N$  queue where  $N$  represents the number of stations in the network. The state diagram now changes as shown in Figure 4.7.

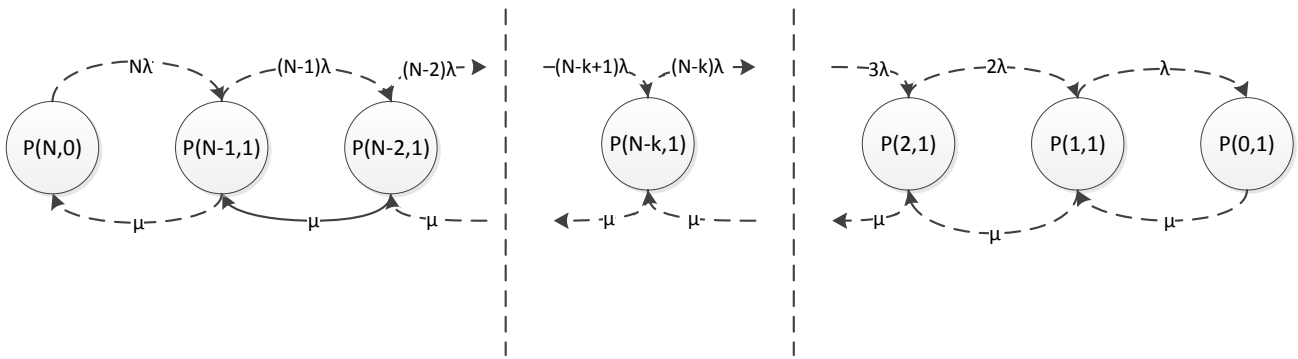


Figure 4.7: Finite Telemetric Queue State Diagram

The arrival of stations with events is still Poisson distributed with exponentially distributed inter-arrival times. The servicing times are still exponentially distributed, because of the varying frame size, data latencies and propagation delay of each station. These characteristics of the telemetry network allow the application of the Markov process. Due to the arrivals being Poisson distributed and the fact that only one station's events can be serviced at a time, the current state can still only transition to one of its neighbouring states, therefore still allowing the application of the Birth-Death Process. The main difference from the  $M/M/1$  queue is that the total Poisson arrival rate used to transition between states is now dependent on the current state that the system is in. This is due to the fact that if a station has new events and joins the queue, the source population reduces, with a corresponding reduction in total arrival rate. After a station has been serviced and is no longer part of the queue, the total arrival rate again increases by an individual station's arrival rate. Here the superposition property of the Poisson process allows us to add and remove stations with Poisson characteristics to the total Poisson arrival rate.

As can be seen from Figure 4.7, the probability term has been given within each state. This is used to define the expected rate of transition from one state to another. The probability that the system will find itself in a specific state is defined by  $P(n, m)$ , where:

- $n$  describes the current state that the system is in, as well as the number of stations that do not have any events, but have the probability to produce new events and therefore are not part of the queue.
- $m$  describes the state of the server of the queue. If  $m=0$ , the server is idle and not servicing any events from any station. If  $m=1$ , the server is busy and currently servicing the events of one particular station.

The arrival time per station is exponentially distributed and the rate is described by  $\lambda$ . The total number of stations in the network is  $N$  and thus the total possible arrival rate is  $\lambda_{max} = N\lambda$ , which is also a Poisson process. A description of the different states in the state diagram and transitions is as follows:

- $P(N, 0)$  is the probability that  $N$  stations can produce new events. It also means that  $N$  stations are currently idle and not waiting for service from the server, therefore currently not part of the queue. Due to the fact that  $N$  stations are idle, the server has nothing to service and is idle. From the state  $P(N, 0)$  the system can transition to state  $P(N - 1, 1)$  at a rate of  $N\lambda$ , due to the fact that  $N$  stations have the probability to create new events. This means that the next station with new events will arrive at an average rate of  $N\lambda$  and move the system to the new state  $P(N - 1, 1)$ . The system can only transition to its closest neighbour which is  $P(N - 1, 1)$ .
- $P(N - 1, 1)$  is the probability that  $N - 1$  stations can produce new events and 1 station is waiting for its service to be completed. The server is no longer idle, as there is a station to be serviced. This state has two immediate neighbours







$$\begin{bmatrix} P(N,0) \\ P(N-1,1) \\ P(N-2,1) \\ \vdots \\ \vdots \\ \vdots \\ P(N-k+1) \\ P(N-k,1) \\ P(N-k-1) \\ \vdots \\ \vdots \\ \vdots \\ P(2,1) \\ P(1,1) \\ P(0,1) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ \vdots \\ \vdots \\ 0 \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \tag{4.28}$$

This is solvable and is the same approach as taken in Equations 4.14 to 4.22, but matrix based for convenience. The matrix, however, isn't square and therefore without inverse. It is therefore necessary to use the M-PPI (Monroe-Penrose Pseudo Inverse) approach as an over specified variable set exists. Matrix B is then given as:

$$\begin{aligned} B &= A^+BC \\ &= (AA^T)^{-1} A^T C \end{aligned} \tag{4.29}$$

Knowing the values of  $P(N,0) \dots P(0,1)$ , the queue length can again be determined. Due to the fact that this is no longer an infinite queue, the geometric series applied in Equation 4.23 can no longer be used. Furthermore the probability  $P(N,0)$  refers to the condition where all stations are idle, therefore no stations are part of the queue. This means that this probability doesn't contribute to the queue length. Therefore, queue length  $L_q$  is:

$$L_q = \sum_{k=1}^N kP(N-k,1) \tag{4.30}$$

To determine the average waiting time Little's Law is used, but it must be remembered that  $\lambda$  differs for each state. This means that the waiting time will have to be calculated iteratively, making use of the queue length contribution at each specific state. For example, for the probability that there are  $k$  stations in the queue, the queue length would be

$$L_{qk} = kP(N-k,1) \tag{4.31}$$

and the corresponding waiting time using Little's Law will be:

$$\begin{aligned} T_k &= \frac{L_{qk}}{\lambda_k} \\ &= \frac{L_{qk}}{[(N+1)-k]\lambda} \end{aligned} \tag{4.32}$$

The above state can be explained by taking  $N = 50$  and calculating the waiting time for a probable queue length of  $k = 50$ . This will change Equation 4.32 to  $T_{50} = \frac{50P(0,1)}{\lambda}$ , which means that given that the arrival rate to state  $P(0,1)$  is  $\lambda$ , the probable queue length will be  $50P(0,1)$  and therefore Little's law can be used to calculate  $T_{50}$ . Making use of Equation

4.30 and Little's Law, the total waiting time will therefore be:

$$T = \sum_{k=1}^N \frac{kP(N-k, 1)}{[(N+1)-k]\lambda} \quad (4.33)$$

An important fact to remember from finite source queues is that the queue will not grow boundlessly if  $N\lambda > \mu$ , due to the fact that the queue is bounded by  $N$ , which means that the queue length will only grow closer to  $N$ .

## 4.4 Summary

Queueing theory is a very useful fundamental analytical tool for analysing the performance of communication networks. The Kendall notation used to describe the type of queue implemented, is introduced and discussed. Little's Theorem, which provides the average queue length and waiting time of a specific queueing model, has been outlined. Narrow band communication networks implemented within telemetry systems usually have an arrival rate which is in line with the Poisson process. The Poisson process is discussed and its properties outlined, as they are a requirement for the Markovian process. The memoryless Markovian Birth-Death process is discussed and used in the analysis of a single server  $M/M/1$  queueing model, for infinite and finite sources. The matrix based state model is introduced and will be used in the theoretical modelling of CSMA and ATW protocols.

The next chapter introduces the typical network layout and protocols under consideration.

## Chapter 5

# Typical Network Layout and Protocol Options

### 5.1 Introduction

In order to provide the necessary background for material presented in later chapters, it might be useful to briefly describe a typical real life network of the type under consideration, as well as the characteristics of a few appropriate and practical protocols.

### 5.2 Network Layout

Each remote station comprises an RTU which gathers and logs data from directly connected inputs as well as serial (using RS232) inputs like PLC's. The RTU makes use of a user defined configuration to determine when events occur and require to be sent to the server or base station. The RTU is serially connected (using RS232) to a MDS 4710E radio. The radio has the following features, relevant to the modelling of each protocol used.

- Makes use of Continuous Phase Frequency Shift Keying (CP-FSK)
- A data latency of 11ms
- Simplex and half duplex functionality
- A channel capacity of 4800 bps
- Error rate of  $10^{-6}$  at a RSSI level of -110 dBm
- Byte length of 10 bits
- The ability to buffer information
- Operating Frequencies are from 440 to 445 MHz for Remote stations and 450 to 460 MHz for Repeater Link frequencies.
- Transmitter gets preference over the receiver e.g. if a signal is received on the serial port, the transmitter is keyed.
- Uses 12.5 KHz channel spacing

- Maximum Transmit power of 5 Watt

All remote stations relay their information via the MDS 4710E radio which in turn relays its information through one or more repeaters. Each repeater comprises 2 MDS 4710E radios, where one is used only as a transmitter (Its RS232 receiver pin has been disconnected) and the other only to receive (Its RS232 transmitter pin has been disconnected). If the repeater has to relay its information to another repeater, an extra radio is added to the repeater system that acts as a point to point link. All repeaters' RS232 serial output is converted to RS485 in order to allow the three radios' serial interfaces to communicate with each other. The information is received at the base station as well as all other stations within the radio network. The server consists of an MDS 4710E radio connected serially to a server, which runs a telemetry application processing the information received and responding as required, usually with an acknowledgement. The layout is shown in Figure 3.4.

Each RTU station has a data carrier detect connected to the radio, which is used to sense activity on the data channel. Whenever a remote station transmits information, the information is received by the repeater and transmitted in the local network as well as transmitted to the next repeater via the Repeater Link Radio. This process is repeated at the next repeaters, allowing all stations within the whole network to receive the signal transmitted by the remote station as shown in Figure 2.13. The server is situated within one of the repeater networks and will therefore receive the signal from the remote station after a delayed period of time. The data will be processed by the server and an acknowledgement of the data will be sent back to the remote station. All stations will receive the signal but only the RTU which initially sent the information will recognize the acknowledgement as that of the data it has just sent to the server. All other data transactions between two stations or from the server will be received by all other stations in the network when transmitted. The following elements cause time delays in the transmission of any type of frame:

#### **Preamble**

Preamble is the time that only the carrier is transmitted by the radio. The preamble setup in the RTU must match the key up time of the radio transmitter.

#### **Postamble**

Postamble is the time that only the carrier is transmitted after a data or acknowledgement frame has been sent. The postamble setup of the RTU must match the de-key time of the radio transmitter.

#### **Turnaround**

The turnaround time is the time that the RTU or server will wait before it can reply to a message received e.g. the time the server must wait before it can send an acknowledgement.

#### **RXEnd**

RXend is the time used to determine the end of a received frame.

#### **Data Bytes**

Data bytes are the actual data that will be transmitted within a data frame.

#### **Information Bytes**

Information bytes are the bytes required to describe the type of frame, from where it is sent and to whom. The Information bytes also include the bytes required for error detection. In the case of the ProDesign and non-persistent CSMA protocol that will be modelled, the error detection method is CRC-16.

**Number of repeaters**

Is the number of repeaters that each data frame will have to relay through before reaching its end destination, which is used in the total data latency calculation.

**Propagation Distance**

Is the distance (km) from the remote station to the server and is used to calculate the propagation delay time.

**Data Latency**

Is the delay caused by the equipment used within the transmission path from a remote station to the server. This will include the delay from the time that data is received on a radio's serial interface to the time that it is transmitted, as well as delay time that could be experienced by other equipment, such as the serial converters used at the repeater.

## 5.3 CSMA Protocol

The CSMA protocol contains many variants, as discussed in Chapter 2. In telemetry networks the most commonly used variant is the non-persistent CSMA protocol.

### 5.3.1 Non-persistent CSMA

A variant of the non-persistent CSMA protocol will be modelled. The arrival times of events at each station are exponentially distributed and this is therefore a Poisson process. The data frames of different types of station will vary in size and, due to the Poisson process of the arriving events, the variation in size will also be Poisson. The number of information bytes found within each data and acknowledgement frame has been determined from the ProDesign protocol and used in the non-persistent CSMA protocol. The service time is the time from the occurrence of an event at the remote station and its transmission up to the time that the package is successfully received at the server, processed and an acknowledgement received by the remote station without any errors occurring.

#### 5.3.1.1 Service and Acknowledgement Time

The service times of different types of frames will differ. In this thesis, only the following types of frames are considered:

- Data frame
- Acknowledgement frame

A data frame is used to send events from the remote station to the server and an acknowledgement frame is sent from the server to inform the remote station that its data frame has been received successfully.

Note that negative acknowledgements (server informing a station that its package hasn't been successfully received and must be sent again) is not used within the protocol as it causes extra overhead on the communication channel. If a corrupted package has been received by the server, it will simply be disregarded. The remote station will re-transmit its package after a timeout period.

In the theoretical and simulation modelling of the protocol a factor has been built in to add extra overheads to the service time because of the probability of unsuccessful transmission of a frame. The probability is extremely low and will contribute only a small amount to the total service time.

## 5.4 Round Robin Protocol

Round Robin Polling (RRP) is a centrally controlled protocol, where the base station or server controls the communication channel and determines which station will be granted access to the channel next. The protocol falls within the Controlled Access Protocol genre, but can also be seen as a collision free protocol, due to the fact that only the base station can initiate communication to any specific station and no collisions can therefore occur. The base station will poll each station requesting the current status of the station, which will then reply with either a control message indicating that it has no new information or it will send a data message containing the latest information from the station. The base station will wait for a reply from the station for a maximum timeout period before either moving on to the next station or resending the request to the specific station. This depends on the way the protocol has been set up. When the base station has requested information from all the stations in the network, it will again start polling the first station within the network. Round Robin Polling is used widely in industry and can be used as a stand alone protocol (e.g. MODBUS) or as part of a mixed strategy (e.g. ProDesign and SSE combine CSMA with RRP). The protocol is best used in systems with:

- Medium to high load conditions
- Sources generating frequent, but constant, data. Bursty traffic might render the protocol inefficient.
- Short propagation delays between the base station and the remote stations
- Stable communication channel with minimum interference

Because of the simplicity and strengths of the protocol various schemes have been developed to optimize the protocol such as:

- Adaptive polling, where the base station builds a statistical record of each station's traffic demand, polling stations with higher traffic demand more frequently by making use of the information that is gathered over time.
- Priority polling, where certain stations are known to have higher traffic demand than others and are assigned priority accordingly. Various priority levels can exist.

## 5.5 Adaptive Tree Walk Protocol

### 5.5.1 Overview

In the previous sections, we have considered the use of a contention protocol (non-persistent CSMA) and a controlled access protocol (RRP). The non-persistent CSMA protocol performs well under low load with low waiting times. As the load on the channel increases though, the protocol becomes less attractive due to the number of collisions that takes place in the contention for the channel. The reverse is true for the RRP protocol, where under low load, the waiting times are high, but as the load increases the channel efficiency increases rather than worsens. Using the Adaptive Tree Walk protocol better channel efficiency can be provided under low and high load demand incorporating the best parts of the non-persistent CSMA and RRP protocols.

This algorithm as computerized in [4], will be discussed next.



### 5.5.2 Computerized \ Slotted Adaptive Tree Walk Algorithm

For the computerized algorithm a proper binary tree is used where the leaves of the tree represent stations, as shown in Figure 5.1. For more information regarding the properties of a proper binary tree see [13]. The protocol has been defined as a slotted protocol, but can also be implemented as unslotted, as will be discussed in Chapter 8. In the first contention slot, slot 0, all stations are permitted to contend for access to the channel. If only one station contends then it transmits its data and the process is done. If, however, a collision occurs, then only those stations falling under node 2 of the tree may contend for channel access in time slot 1. If a station acquires the channel, then the next time slot is reserved for stations under node 3. If, however, there is another collision in time slot 1, then only stations under node 4 may contend for the channel in time slot 2. If only one station contends for the time slot, it will successfully obtain the channel in the time slot and transmit its data. Time slot 3 will then be contended for by stations under node 5. If the channel stays idle for the duration of time slot 3, time slot 4 will belong to stations under node 3, as no stations under node 5 require service and the specific branch of the tree will not be searched further. The process will be continued until no more collisions or successful contentions occur or until the complete tree has been iterated, whereafter the process will be repeated.

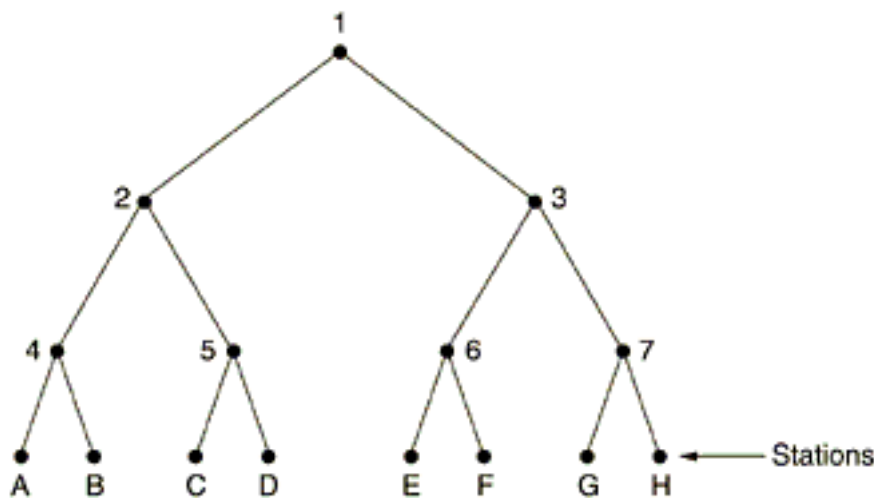


Figure 5.1: Adaptive Tree with Eight Stations from [37]

It can be seen that if a collision occurs within a time slot, the algorithm moves a level down within the tree and sends a request again to stations falling under that specific node. If during this time slot, the channel stays idle or only one station replies, that specific node no longer has to be searched and the algorithm moves to the sibling node where it again allows stations under the specific node to contend for the time slot if they require it. Therefore each time a collision occurs the tree will be searched recursively within the specific node's left and right children.

The more traffic, the more nodes need to be searched. In time slot 0, sending a request to all stations for data will only be effective if there is only one station at a time requiring to send data. To optimize the tree, the search algorithm can start lower down in the tree, depending on historical statistics that has been gathered regarding the traffic of the network. To determine this, the top of the tree is seen as level one of the tree. In Figure 5.1 the tree has 4 levels numbered level 0 to 3 which means that a level 4 tree can, at maximum, have  $2^3$  stations assigned to it. The size of the tree can therefore be generalized as follows, if the tree is balanced:

$$N_s = 2^i \quad (5.1)$$

$$i = \text{Tree Levels} - 1 \quad (5.2)$$

where:

$N_s$           Number of stations in a balanced tree

Going a level down in the tree, the number of the stations under each node is halved e.g. nodes 2 and 3 will each have  $2^{i-1} = 2^2 = 4$  stations below them. If, for example, there are two stations that have data ready and they are uniformly distributed, one under node 2 and one under node 3, it makes sense that the level to start the search at is level 1 instead of level 0 to prevent collisions from occurring. The level  $j$  to start the search at to reduce the probable amount of collisions to 1, will be the level where only one station has data under the specific node [37]:

$$\begin{aligned}2^{-j}q &= 1 \\j &= \log_2 q\end{aligned}\tag{5.3}$$

where:

$q$           Average number of data ready stations uniformly distributed

$j$           Tree level to start search at to reduce collisions

An unslotted adaptive tree walk protocol which will be used in the modelling has been developed and is discussed in Chapter 8 . The unslotted protocol is a variant of the one described above and has been developed to cater for long distances and delays which are found in telemetry radio networks. These delays negatively effect the performance of the slotted protocol, as will be discussed later.

## 5.6 Summary

This chapter has given an overview of the network layout and protocols typically used in telemetry networks. These protocols will be modelled and discussed in the next chapters, making use of the telemetry network layouts previously described.

## Chapter 6

# Carrier Sense Multiple Access: Modelling and Simulation

### 6.1 Introduction

The non-persistent CSMA protocol or variations thereof is used extensively throughout the field of telemetry for narrow band radio networks. South African RTU manufacturers, SSE (Specialized Systems Engineering) and ProDesign are companies that makes use of a non-persistent CSMA protocol. Both of these products have been implemented widely within South Africa, Namibia and other parts of Southern Africa. These systems are also implemented in the case study of this thesis. The manufacturers have graciously given information regarding their communications strategies for analysis within this thesis. In the central coastal water supply scheme of Namibia (aka Namib water supply scheme), ProDesign RTUs have been implemented for the monitoring of the various pumping systems, see Chapter 2 for more detail.

Further discussion will be focused on the essence of the ProDesign protocol and how it has been implemented within the Namib water supply scheme. The specific case study has been used as it consists of a large network of remote monitoring stations, which is planned to be expanded within the near future, and problems within this expansion because of high traffic load are foreseen.

#### 6.1.1 Description of the protocol used

When an event occurs at a remote station, the station will first go into a uniform backoff. After that the RTU will sense the channel to determine whether it is busy. If it senses that the channel is busy, it will go into a uniformly distributed random backoff state. If the channel is not sensed as busy, either after backoff or initially, the station will send a poll request to the server in a poll request frame. The station will then wait for a uniformly distributed timeout period. If the station receives a poll request from the server before the timeout period expires, it will again sense the channel to determine availability; if not available, it will again back off as before until the channel is available. When available the station will send its data frame, piggybacking the acknowledgement of the poll request. Upon receiving this message the server will respond with an acknowledgement informing the remote station that the frame has been received successfully. If the stations' timeout period expires before receiving an acknowledgement from the server, the station will assume that the transmission was unsuccessful and retransmit its frame, after it has backed off again. If the station has not received a request to send its information to the server after 10 poll requests, it will go into a slower poll request mode where the random backoff period between requests is increased. The flow diagram of this process is shown in Appendix A.1.

The protocol is unique in that it sends an extra poll request packet before communication can be initiated. This causes extra overheads per data frame, which increases the probability for collisions upon the increase of the arrival rate. This does not follow the IEEE 1815 standard, implemented by manufacturers as an underlying base. The non-persistent CSMA protocol follows this implementation closely and will be used in modelling. In this protocol, if an event arises at a remote station it will sense the channel to determine if it is busy, if not, it will send its data frame. If the channel is busy, it will go into a uniformly distributed random backoff period, after which it will again sense the channel. After sending its data frame, the RTU will wait for a fixed time-out period to receive an acknowledgement from the server. If the timeout occurs before the ACK frame has been received, the station will again repeat the above process. The flow diagram of this protocol is shown in Figure 6.1. The LLC layer's Stop-and-Wait ARQ protocol and the non-persistent CSMA protocol, both described in Chapter 2, have been merged into one protocol in the special case of the non-persistent CSMA protocol used in telemetry systems. All discussions from now on will refer to the non-persistent CSMA protocol as described above.

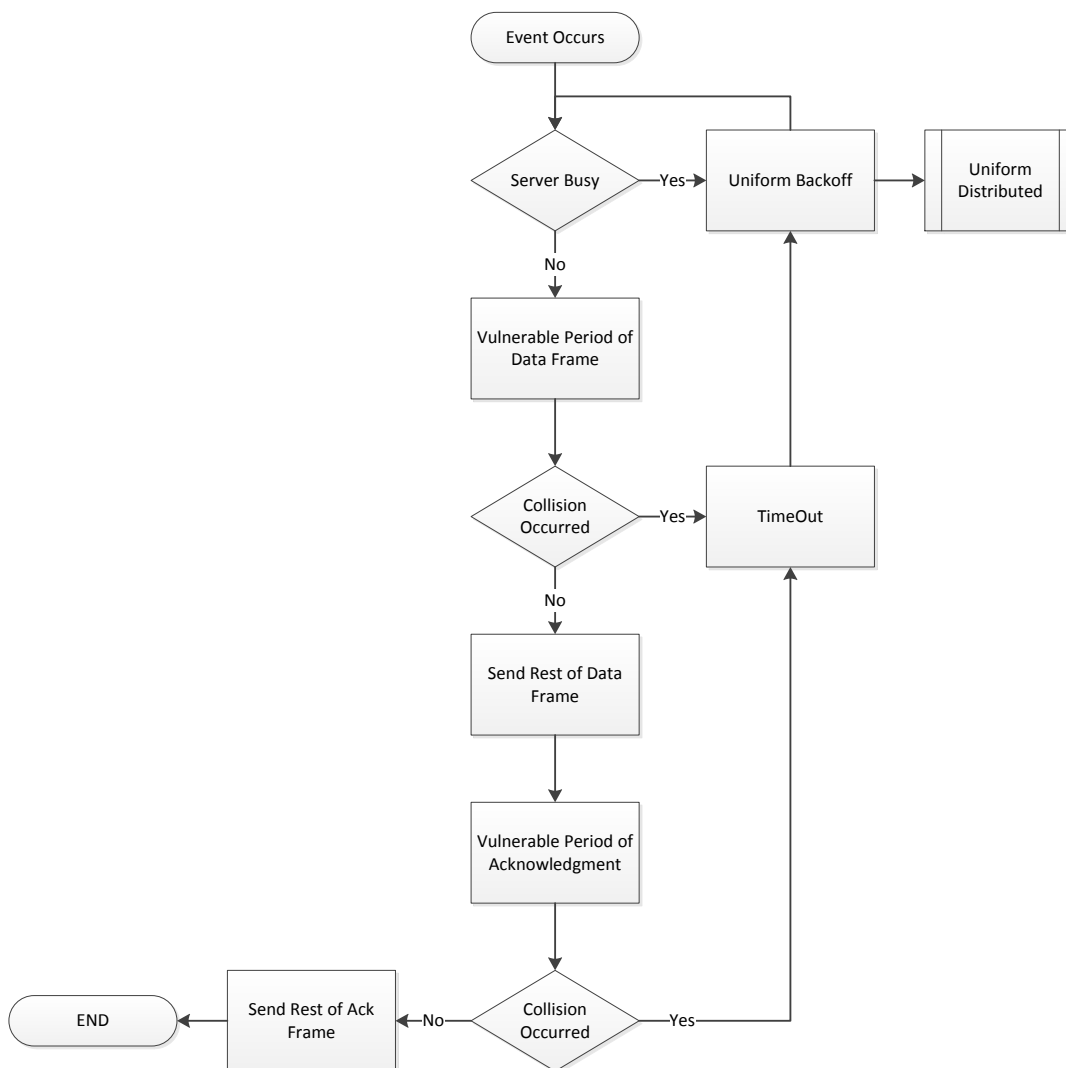


Figure 6.1: Non-persistent CSMA Flow Diagram

### 6.1.2 Vulnerable Period

The main cause for frames not reaching their end destination intact is collisions with other frames being transmitted at the same time. This happens during the period when a transmitted signal from one station has not yet reached the receiver of another, therefore the second station doesn't know of the transmission taking place and will transmit, causing a collision. Both stations involved in the collision will wait until their timeout periods have expired, backoff and then try again, causing a large delay in the transmission of each stations frame. This can happen in the case of a data frame or the acknowledgement of a data frame being sent with the same probability of collision. The period within which collisions can occur is referred to as the vulnerable period and is caused by delays in equipment and RF propagation delay of the signal. The Hidden Node problem doesn't apply to this network as all stations are able to communicate with each other. The signal sent from a distant point A will take time before it reaches point C on the other side of the network and it is this delay that is causing the collisions. Figure 6.2 shows the coverage of each repeater within the three-repeater network used in the Namib water supply scheme. Each repeater covers all stations within its local network and all repeaters are linked to each other in a star topology. This shows that all stations can receive transmissions from any other station within the network. From the indicated scale, it can be seen that the distance covered is extensive, therefore the propagation delay, together with the number of repeaters, results in increased delay between stations.

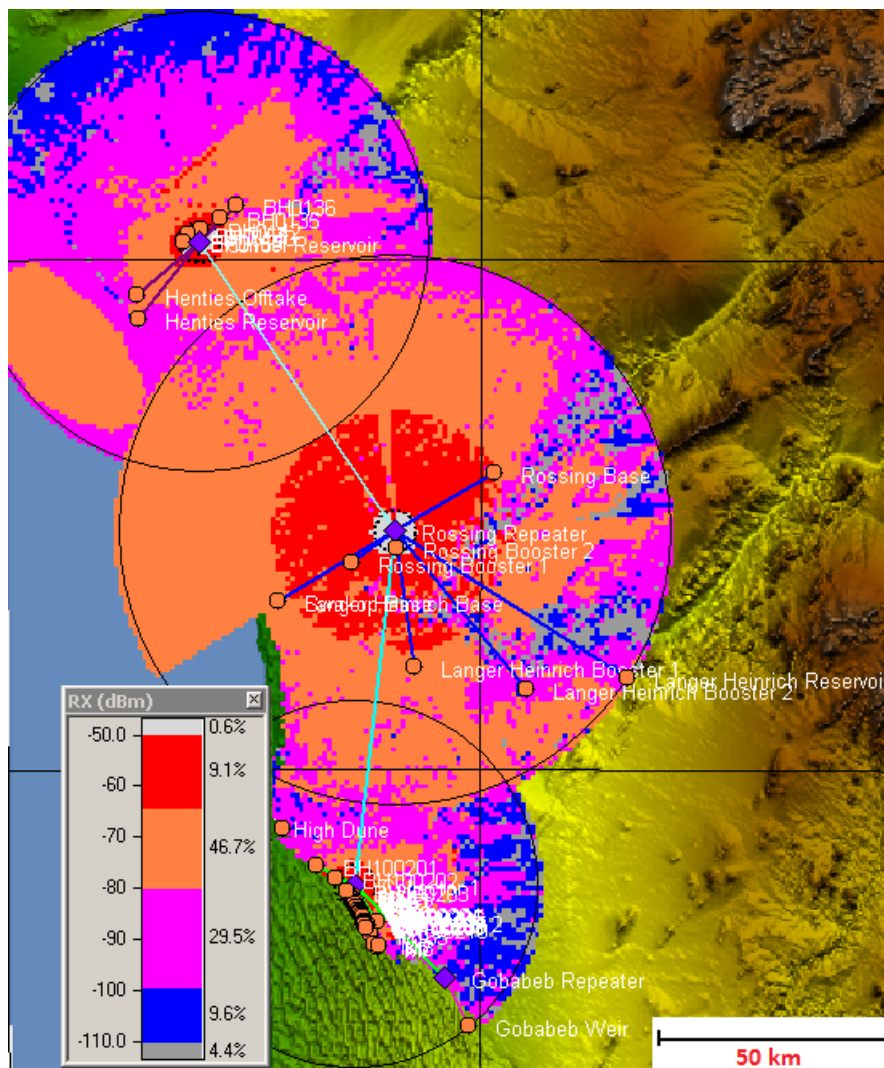


Figure 6.2: 3 Repeater Network Coverage

Each radio in the network has a specified data latency or delay time. This is the time from when data is received on the serial interface, until the time it is transmitted (the delay from the RTU to the radio has been omitted as insignificantly

small). This is also known as the rise time of the transmitter. It is also the time from when data is received by the radio until it is made available to the serial interface. For the MDS 4710E this time is given as 11ms, meaning that from the time radio A receives information on its serial interface, to the time that radio B can produce this information on its serial interface, is approximately equal to 22 ms, excluding propagation delay. This, however, varies in accordance with the propagation distance. Therefore, any station that transmits a data frame will experience delays. These delays are caused by the stations radio and repeater transmitter key-up times, the receiver delay of the repeater (more than one repeater is possible), the receiver delay of any other station and the propagation delay. This whole period can be seen as vulnerable, as at any point in time during this period all other stations in the network will not be aware of the transmission taking place and may commence a transmission of their own. It is clear that the more repeaters in the network, the higher the data latency. If two repeaters are linked, four radios are added to the transmission path of the data frame.

$$t_{dl} = t_{Stx} + t_{Rrx} + t_{Rtx} + t_{Srx} + t_{pdl} \quad (6.1)$$

$$t_{dl} = t_{Stx} + nt_{Rrx} + nt_{Rtx} + (n-1)t_{RLrx} + (n-1)t_{RLtx} + t_{Srx} + t_{pdl} \quad (6.2)$$

where:

$t_{dl}$	Data latency (ms)
$t_{Stx}$	Station transmitter data latency (ms)
$t_{Rrx}$	Repeater receiver data latency (ms)
$t_{Rtx}$	Repeater transmitter data latency (ms)
$t_{Srx}$	Station receiver data latency (ms)
$t_{pdl}$	Propagation delay (ms)
$t_{RLrx}$	Repeater link receiver data latency (ms)
$t_{RLtx}$	Repeater link transmitter data latency (ms)
$n$	Number of repeaters

Equation 6.1 shows the delay experienced in a single repeater network, while Equation 6.2 shows the delay for a network containing  $n$  repeaters. It is therefore evident that a higher event arrival rate with longer propagation distances and more repeaters in the network, results in an increased probability of collision.

### 6.1.3 Data and Acknowledgement Frame Transmission Rates

In each data transmission from a remote station to the server two frames are involved, one being the data layer frame and the other a shorter acknowledgement frame. Each data frame contains actual data bytes. Information bytes contain the station, server and network addresses, CRC-16 code, type of frame being sent and other control information. The acknowledgement frame also contains all the information bytes described above, but no data payload, making it much shorter than the data frame. The data frame can vary in size, depending on the amount of data, whereas the acknowledgement frame is a fixed size. Each data byte within a frame, requires a certain amount of time to be sent. The time is determined by the channel capacity; as discussed in Chapter 2 and can be expressed as follows:

$$t_{ib} = \frac{\text{Information Bytes} * 10}{\text{Channel Capacity}} * 1000 \quad (6.3)$$

$$t_{db} = \frac{\text{Data Bytes} * 10}{\text{Channel Capacity}} * 1000 \quad (6.4)$$

where:

$t_{ib}$  time required to send information bytes (ms)

$t_{db}$  time required to send data bytes (ms)

The MDS 4710E adds 2 bits to each byte sent, therefore the number of information and data bytes is multiplied by 10 to get both in terms of bits as the channel capacity is measured. Making use of all the different times involved in sending a frame, as found in Section 5.3.1.1, the transmission times of data and acknowledgement frames, without any errors and collisions are:

$$t_{df} = t_{pr} + t_{ps} + t_{ta} + t_{rd} + t_{dl} + t_{ib} + t_{db} \quad (6.5)$$

$$t_{af} = t_{pr} + t_{ps} + t_{ta} + t_{rd} + t_{dl} + t_{ib} \quad (6.6)$$

where:

$t_{df}$  Data frame transmission time (ms)

$t_{af}$  Acknowledgement frame transmission time (ms)

$t_{pr}$  Preamble time (ms)

$t_{ps}$  Postamble time (ms)

$t_{ta}$  Turnaround time (ms)

$t_{rd}$  RXEnd time (ms)

$t_{dl}$  Data latency time (ms)

$t_{ib}$  Information bytes transmission time (ms)

$t_{db}$  Data bytes transmission time (ms)

#### 6.1.4 The Probability of Error

If the communication channel used is noisy or the remote station is located on the border of repeater coverage with a minimal signal, the SNR and RSSI (Receiver Signal Strength Indicator) will be low, increasing the probability of the bits transmitted being received incorrectly. In the MDS 4710E, the error rate is  $10^{-6}$  at a RSSI level of  $-110$  dBm. With improved RSSI levels, the SNR level should increase and the error rate will decrease even more. Because of the small influence that the error rate will have on the overall transmission, it is not included in the model simulation, but rather calculated as an extra time delay within each frame. The extra time for data and acknowledgement frames is calculated as follows:

$$t_{de} = \left( \frac{(\text{Data Bytes} + \text{Information Bytes}) * 10}{10^6} \right) * (t_{df} + t_{to} + t_{bf}) \quad (6.7)$$

$$t_{ae} = \left( \frac{\text{Information Bytes} * 10}{10^6} \right) * (t_{af} + t_{to} + t_{bf}) \quad (6.8)$$

where:

$t_{de}$  Data frame error time (ms)

$t_{ae}$  Acknowledgement frame error time (ms)

$t_{bf}$  Average backoff time (ms)

$t_{to}$  Timeout time (ms)

The error time for both data and acknowledgement frames is computed by finding the rate of error per frame and multiplying it by the extra time that will be required to send a new frame. If the protocol doesn't make use of backoff times,  $t_{bf}$  can be left out of the equations. The transmission time for each type of frame, including extra time for possible re-transmissions due to bit errors, are as follows:

$$t_{dat} = t_{df} + t_{de} \quad (6.9)$$

$$t_{ack} = t_{af} + t_{ae} \quad (6.10)$$

where:

$t_{dat}$  Data frame transmission time including extra time for the probability of errors

$t_{ack}$  Acknowledgement frame transmission time including extra time for the probability of errors

These are the data and acknowledgement frame times used in the model. The total transmission or service time can therefore be seen as the summation of  $t_{dat}$  and  $t_{ack}$  and is given by:

$$t_{ser} = t_{dat} + t_{ack} \quad (6.11)$$

### 6.1.5 Modelling Considerations and Assumptions

The events occurring at the different stations are exponentially distributed as regards time and the rate at which the RTU sends data frames to the radio will be Poisson distributed. As different RTU stations will have different I/O counts, as seen in Chapter 2, the service rate will also be Poisson distributed. Data frames sent from each station will be independent of the previous frame sent from the specific station or any other station, making it a memoryless occurrence. Taking this information into account, the waiting time per station and the queue length of stations waiting to be serviced can be modelled using a finite single queue Markovian chain. It is assumed that after an event has occurred at a station, no other events will occur during the time that the station waits for its data frame to be serviced. The assumption stays valid while the total arrival rate is smaller than the total service rate, meaning that the summation of the rate at which all arrivals occur, will be smaller than the rate at which the stations are being serviced. Therefore the system can be defined as follows:

- Each station has a Poisson arrival rate and an assumed maximum queue length of 1.
- The combined arrival rate of all stations is Poisson and forms a system queue with maximum length equal to the number of stations in the system.
- The service rate of each station is Poisson due to the variable size of the data frame.
- The summed service rate of all stations will also be Poisson.

A further assumption has been made that all stations in the system will have the same Poisson arrival rate and subsequently the average Poisson service rate is determined using an average data frame size which is determined from the average of the maximum and minimum frame size throughout the system. The backoff rate is uniform and an average of the maximum and minimum rates has been taken.



## 6.2 Theoretical Modelling

### 6.2.1 Introduction

The non-persistent CSMA protocol can be modelled as a  $M/M/1/N$  queue. This protocol can be modelled using the same techniques described in Section 4.3.3. Both the arrival and service rates are Poisson processes and with arrivals occurring independently, the requirements of modelling the queue as a Markovian chain are satisfied. An expanded state diagram is used within the existing and expanded state space models, where the current state can still only move to its neighbouring states which is in line with the Birth-Death process requirements. An initial state model will be presented where the results hold only for a narrow band of backoff and timeout values. The model is therefore expanded to allow the modelling of previously encapsulated states in the expanded model, resulting in accuracy over a wider range of parameters when compared to the simulation.

### 6.2.2 Existing State Space Model

Figure 6.3 shows the state diagram as presented by [40]. The figure shows a system with 5 stations as an example. This system will be analysed to give an understanding of the state space modelling principle.

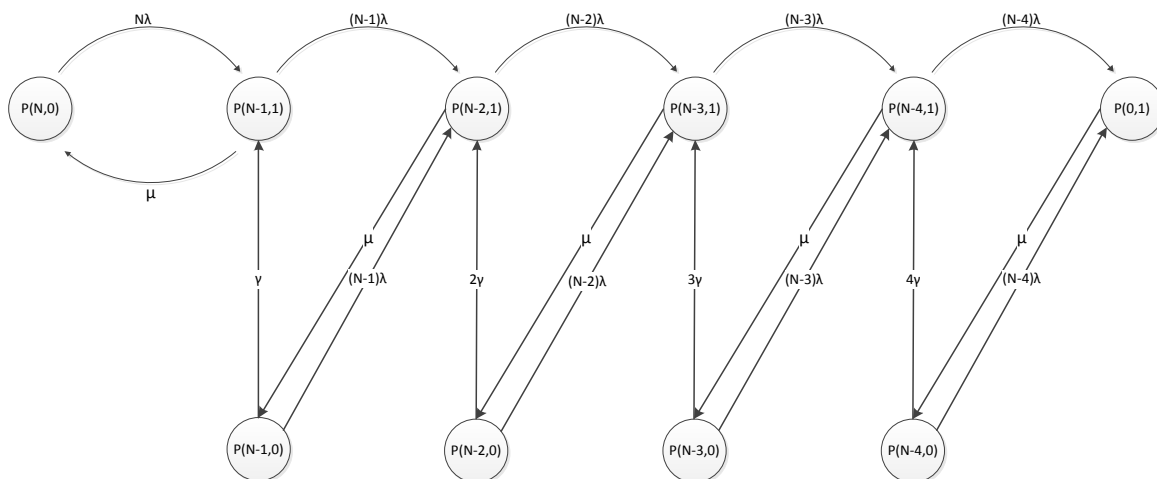


Figure 6.3: State Space Model from [40]

The symbols used in the diagram are as follows:

- $\lambda$  Poisson arrival rate per station
- $\mu$  Poisson service rate of the system
- $\gamma$  Backoff rate per station after collision
- $N$  Number of stations in the system
- $P(n, m)$  Probability that  $n$  stations are inactive and can create newly arriving events to add to the queue and  $m$  describes the state of the server. If  $m = 0$ , then the server is idle and not servicing any station event, whereas if  $m=1$ , the server is busy servicing an event from a station.

The service rate  $\mu$  is determined by:

$$\mu = \frac{1}{t_{dat} + t_{ack}} = \frac{1}{t_{ser}} \quad (6.12)$$

The description of different states within Figure 6.3 can be described as follows:

#### **P(N,0):**

Currently no station events are being serviced in the system or waiting to be serviced. A probability therefore exists, that a new event can be created at N stations with a Poisson arrival rate of  $\lambda$ . This means that the next event to be serviced will arrive at a rate of  $N\lambda$ , with a probability of  $P(N,0)$ . After the arrival of the new event, the number of stations that can create a new event is reduced to  $(N-1)$  and the server will be busy servicing the new arrived event, which implies that the new state the system is in is  $P(N-1,1)$ . The service rate of the station events is  $\mu$ . After the station event has been serviced at this rate, the system will return to a state where the server is idle and N stations have the ability to create new events for the  $P(N,0)$  state. It makes sense that the system can only revert to this state if the service rate is greater than the next arrival rate i.e.  $\mu > (N-1)\lambda$ .

#### **P(N-1,1):**

For the system to be in state  $P(N-1,1)$  means that an arrival has occurred. The occurrence of this arrival could be due to a new event occurring, thus the system has moved from state  $P(N,0)$  to  $P(N-1,1)$ , or the occurrence of this arrival could also be from an event that occurred a while back, but the station where the event occurred couldn't get access to the communication channel, as it was already busy and went into backoff. The station returned from backoff to find the communication channel idle and sent its event at a rate  $\gamma$ , which moves the system from state  $P(N-1,0)$  to  $P(N-1,1)$ . When in state  $P(N-1,1)$  two things could occur:

- The event being serviced gets serviced before a new arrival occurs and the system moves to state  $P(N,0)$  at a rate  $\mu$ .
- A new station event occurs before the event being serviced is complete and the system will, therefore, move to a state  $P(N-2,1)$  where one station event is being serviced and another station has an event to be serviced, but has gone into backoff due to the unavailability of the communication channel. The transition from state  $P(N-1,1)$  to  $P(N-2,1)$  occurs at a rate of  $(N-1)\lambda$ , which is the arrival of a new event from any station that doesn't have any events to be serviced yet. The probability of a new event occurring at a rate of  $(N-1)\lambda$  is  $P(N-1,1)$ .

#### **P(N-2,1):**

In state  $P(N-2,1)$ , two stations have events that need to be serviced. One of these station events is being serviced while the other station has gone into backoff due to the unavailability of the server. As described earlier, the event being serviced could be either a new event that has occurred or an event from a station that has returned from backoff. When in state  $P(N-2,1)$  two things could occur:

- The current event could complete service before a new event occurs at any station that does not yet have an event to be serviced. The system will then move to state  $P(N-1,0)$  at a rate of  $\mu$ . This will occur at a probability of  $P(N-2,1)$ .
- A new event could occur while the server is busy servicing an event and the system will move to state  $P(N-3,1)$ . The new event found the server busy and also went into backoff. Therefore, in state  $P(N-3,1)$  two stations are in backoff and one station is being serviced, meaning that 3 stations are waiting for their events to be serviced. The system will move to state  $P(N-3,1)$  from  $P(N-2,1)$  at a new arrival rate of  $(N-2)\lambda$ , with a new event occurring with probability  $P(N-2,1)$ .

**P(N-1,0):**

In state  $P(N-1,0)$ , the server is idle as the station with an event to be serviced is in backoff and no new arrival has occurred. When in state  $P(N-1,0)$  two things could occur:

- The station in backoff returns from backoff at a rate of  $k\gamma$ , where  $k = 1$  in this instance, as only one station is in backoff. It finds the server idle and sends its information. Therefore, the system can move from state  $P(N-1,0)$  to state  $P(N-1,1)$  at a rate of  $k\gamma$  with a probability of  $P(N-1,0)$ , with  $k = 1$ . It should be noted that the number of stations capable of creating new events stays  $N-1$ , as no new event has occurred and an existing event which was already waiting to be serviced has returned from backoff.
- A new event can occur at a station which is not waiting to be serviced. Where the station is in backoff and the server idle, the system will change from  $P(N-1,0)$  to  $P(N-2,1)$ , where the new arrival is being serviced and the other still in backoff. This will occur at a rate of  $(N-1)\lambda$  with a probability of  $P(N-1,0)$  if new arrivals occur at a greater rate than the station coming out of backoff.

**P(0,1):**

All states up to  $P(0,1)$  will have the same transition process as described for  $P(N-2,1)$  and  $P(N-1,0)$ . In this state all stations have events to be serviced and one station event is being serviced, with  $N-1$  events waiting to be serviced. As no new events can occur, the only transition from this state to  $P(N-4,0)$  (the state can also be seen as  $P(1,0)$ ) will occur at a rate of  $\mu$  with a probability of  $P(0,1)$ . See Figure 6.3.

State inflows have to balance state outflows for the system to be in equilibrium. This is done using global balance equations as shown below:

$$\begin{aligned}
& P(N,0)N\lambda - P(N-1,1)\mu = 0 \\
& P(N-1,1)((N-1)\lambda + \mu) - P(N,0)\lambda - P(N-1,0)\gamma = 0 \\
& P(N-1,0)((N-1)\lambda + \gamma) - P(N-2,1)\mu = 0 \\
& P(N-2,1)((N-2)\lambda + \mu) - P(N-1,1)(N-1)\lambda - P(N-1,0)(N-1)\lambda - P(N-2,0)(2\gamma) = 0 \\
& \dots \dots \dots \dots \dots \dots = 0 \\
& \dots \dots \dots \dots \dots \dots = 0 \\
& P(1,0)(\lambda + (N-1)\gamma) - P(0,1)\mu = 0 \\
& P(0,1)\mu - P(1,1)\lambda - P(1,0)\lambda = 0
\end{aligned} \tag{6.13}$$

$$P(N,0) + P(N-1,1) + P(N-1,0) + P(N-2,1) + \dots + P(1,0) + P(0,1) = 1 \tag{6.14}$$

Equation 6.14 represent the summation of all state probabilities and must equal one. The above equations can again be represented in state matrix format, as  $AB = C$ :

$$\begin{bmatrix}
 N\lambda & -\mu & 0 & 0 & 0 \\
 -N\lambda & (N-1)\lambda + \mu & -\gamma & 0 & 0 \\
 0 & 0 & (N-1)\lambda + \gamma & -\mu & 0 \\
 0 & -(N-1)\lambda & -(N-1)\lambda & (N-2)\lambda + \mu & -2\gamma \\
 \vdots & \vdots & \vdots & \vdots & \vdots \\
 \vdots & 0 & 0 & \lambda + (N-1)\gamma & -\mu \\
 \vdots & 0 & -\lambda & -\lambda & \mu \\
 1 & 1 & 1 & 1 & 1
 \end{bmatrix} \times \begin{bmatrix}
 P(N,0) \\
 P(N-1,1) \\
 P((N-1),0) \\
 P(N-2,1) \\
 P((N-2),0) \\
 \vdots \\
 \vdots \\
 P(1,01) \\
 P(1,0) \\
 P(0,1) \\
 0
 \end{bmatrix} = \begin{bmatrix}
 0 \\
 0 \\
 0 \\
 0 \\
 0 \\
 \vdots \\
 \vdots \\
 0 \\
 0 \\
 0 \\
 1
 \end{bmatrix} \tag{6.15}$$

Matrix A isn't square, but has full rank, and the special case of the pseudo inverse, as given in Equation 4.29, can be applied to find the values of the probabilities, as described in Section 4.3.3. Note that full rank is only applicable to matrix A if  $\lambda$  and  $\mu$  are both greater than 0. The pseudo inverse is easily found in Matlab. The average queue length can be found by making use of the principles presented by Equation 4.30. The system can have a probable queue length of  $(N - k)$  where the server is idle or busy. The sum of  $P(N - k, 1)$  and  $P(N - k, 0)$  gives the actual probability that the system has a queue length of  $(N - k)$ . Therefore, Equation 4.30 is altered to take this into account and the average queue length is calculated as follows:

$$N = \sum_{k=1}^N k [P(N - k, 1) + P(N - k, 0)] \tag{6.16}$$

The average waiting time can be determined by making use of Little's theorem,  $T = \frac{N}{\lambda}$ , as per Equation 4.33. Due to the fact that  $\lambda$  isn't constant, Little's theorem must be applied iteratively. Each probable queue length as calculated above will represent a fraction of the final queue length. The probabilities represented in the calculation of each probable queue length, also represent a specific state within the state diagram, which has a specific new arrival rate that will cause a transition to a new state. Using this specific  $\lambda$ , together with the probable queue length, (calculated from the probabilities representing the states, which implies that a certain number of stations are waiting for service), a probable waiting time per state can be calculated. This probable waiting time represents a fraction of the final average waiting time of the system. The final average waiting time is calculated as follows:

$$T = \sum_{k=1}^N \frac{k [P(N - k, 1) + P(N - k, 0)]}{[(N + 1) - k] \lambda} \tag{6.17}$$

The waiting time calculated above, makes use of an underlying service time that has been extended to include the probability that an arrival will be corrupted by noise as discussed in Section 6.1.4. This follows from the fact that the failure reduces overall system throughput and, therefore, has the same effect as increasing the service time of each event. In [40] the probability of error due to noise is modelled as burst noise ( $\delta_n$ ), which in effect will reduce the service rate ( $\mu$ ) and can be modelled as such. In this protocol, a station will implement a timeout period before it considers the transmission as failed, after which it will backoff and will then sense the channel upon return from backoff to retransmit the data frame. This time is represented by  $t_{err}$ , which is multiplied by the burst noise ratio  $\delta_n$  to provide the increased service time. The effective service time is therefore extended to incorporate the delay caused by retransmission, as given by the following equations:

$$\frac{1}{\mu_{eff}} = t_{ser-eff} = t_{ser}(1 + \delta_n t_{err}) \quad (6.18)$$

$$t_{err} = t_{to} + t_{bf} \quad (6.19)$$

The burst noise is already incorporated in the service time as discussed in 6.1.4. In this model though, the burst noise is fixed, as obtained from the MDS 4710E data sheets for specific data and acknowledgement frame lengths. The theoretical model as implemented is accurate when compared with the simulation model, as long as the vulnerable period of the simulation is set to zero. This, however, is unrealistic as with any practical system there will be propagation and equipment latencies as discussed before. When the vulnerable period is increased, the model no longer corresponds with the simulation. This result is expected, as the model accounts only for burst noise. The burst noise ratio ( $\delta_n$ ) however does not account for collisions that could occur. These are directly affected by the arrival rate, timeout period, average backoff time and the range between the minimum and maximum backoff periods. An increase in average backoff time will increase the waiting time per station as the arrival rate increases, but might reduce the number of collisions because fewer stations are contending for the communication channel and transmitting during the vulnerable period of another station. The effect of the backoff period on the waiting time and consequently the queue length, will also vary with the range of backoff values and not only the average backoff time. Furthermore, there is also a correlation between the chosen ranges of average backoff time, backoff range and timeout. All these will affect the waiting time and queue length of the system and must be accommodated. According to [40], the following assumptions are valid:

- For data from two stations not to collide, the communication channel must be free from any transmission for a period of  $2t_{pdl}$ , where  $t_{pdl}$  is the propagation time delay between two stations.
- A station returning from backoff will have the possibility of colliding with a new arrival during the data latency or rise time ( $t_r$ ) of the radio equipment within its communication path. Therefore, the higher the backoff rate and the longer the data latency of the equipment in the transmission path, the higher the probability of collisions taking place.
- It makes sense that a station leaving the backoff state can only collide during the vulnerable period of a transmission in progress and not during the idle period of the communication channel. This effect can be modelled as a function of the traffic density  $\rho = \frac{N\lambda}{\mu}$ .

Taking the above factors into account, the effective service time can now be adjusted to:

$$\frac{1}{\mu_{eff2}} = t_{ser-eff2} = t_{ser}(1 + \delta t_{err}) + 2t_{pdl} + n\gamma t_r \rho t_{err} \quad (6.20)$$

The adjusted service time ( $\mu_{eff2}$ ) causes the service time to change with each state and the state diagram will change as shown in Figure 6.4. With the fixed burst noise employed into the service time for a specific data and acknowledgement frame length, the above equation has been adapted as follows:

$$\frac{1}{\mu_{eff2}} = t_{ser-eff2} = t_{ser} + 2t_{pdl} + n\gamma t_r \rho t_{err} \quad (6.21)$$

Equation 6.21 makes use of a fixed burst noise, which will be approximately 1.4% for a data frame size of 180 bytes with an acknowledgement of 20 bytes. For a smaller data frame of 47 bytes and acknowledgement of 13 bytes, the percentage changes to 0.7%. The effective service rate changes for each state and, therefore, the state diagram changes as depicted in Figure 6.4. This model has been implemented with acceptable results at low backoff rates, high timeouts and realistic arrival times. However, changing the timeout period is modelled less successfully with higher arrival rates, (where a timeout period of  $2t_{pdt}$  no longer suffices) if higher backoff rates are implemented for better performance. The model provides better performance for shorter timeouts at higher backoff and arrival rates. This is not in accordance with the simulation results. The theoretical model has, therefore, been expanded to increase its accuracy and range of variables for which acceptable results are obtained and is discussed next.

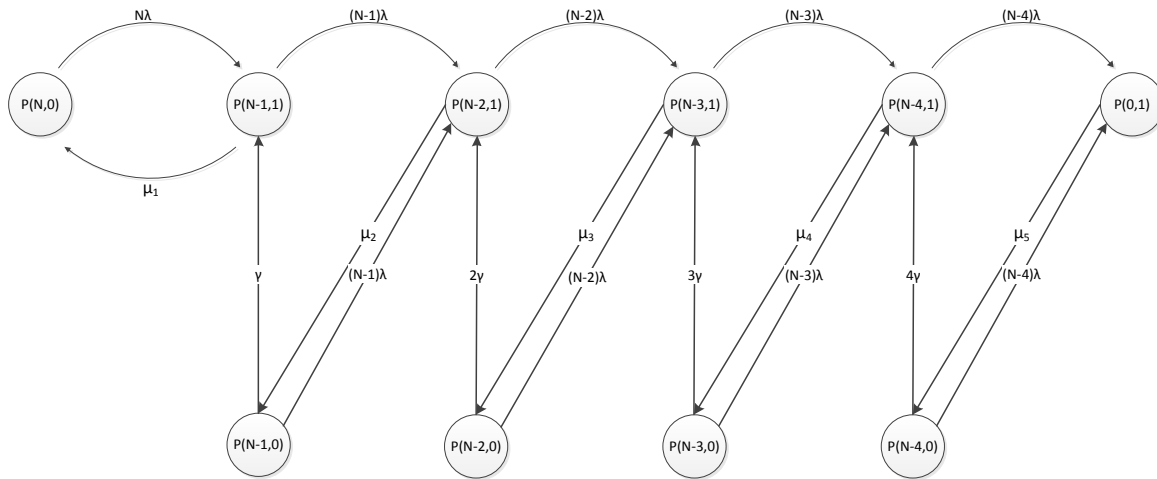


Figure 6.4: State Space Model with Varying Service Times

### 6.2.3 Expansion of Existing State Space Model

The previous model allowed for two possible states, i.e. one state in which the server was busy and another where the server was idle. This requires the service time in the model to be adjusted for collision conditions. The expanded model builds upon the idea of the previous one, but is expanded to include various other states which exist within the protocol and it therefore reduces the number of adjustments required in the service time. All requirements for a birth-death process are still valid. The state space diagram for five stations is shown in Figure 6.5. The model now has six different states in which the model can reside.

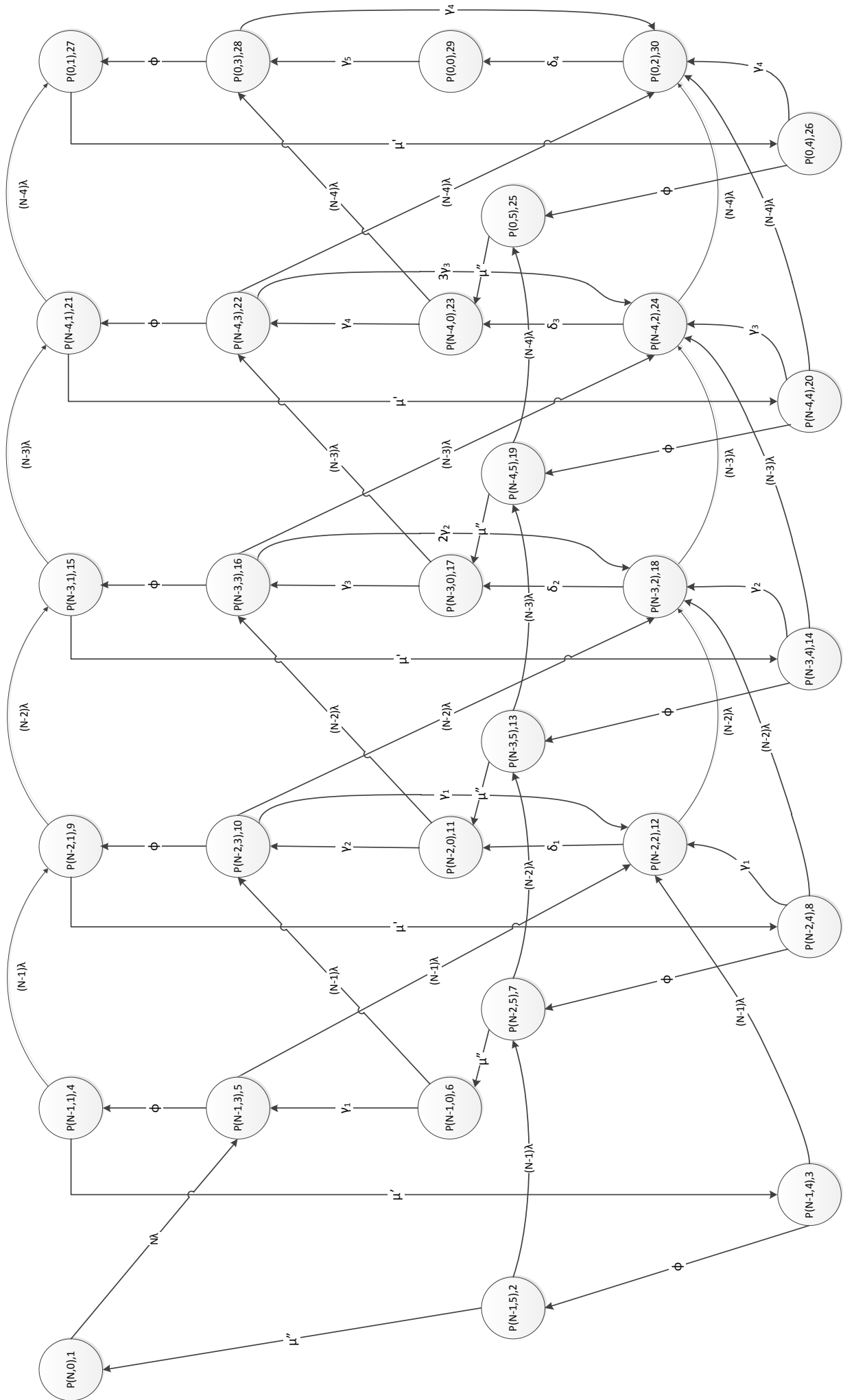


Figure 6.5: Expanded State Space Model

The symbols within this diagram are defined as follows:

$\lambda$	Arrival rate per station
$\mu'$	Service rate of the data frame, excluding the vulnerable period
$\mu''$	Service rate of the Acknowledgement frame, excluding the vulnerable period
$\phi$	Vulnerable rate of each frame sent.
$\delta_k$	Return rate from the collision state
$\gamma_k$	Variable backoff rate, adjusted for the number of stations in the queue
$N$	Total number of stations modelled
$P(m, n)$	The probability of being within one of 6 particular states.

To assist understanding of the state diagram, a number has been added to each state  $P(m, n)$ , *State Number* in Figure 6.5. This number also indicates the order in which the global balance equations should be added to the state matrix.  $P(m, n)$  describes the probability of being in each of six different states:

$P(m, 0)$	$n = 0$ , represents the state in which the server and communication channel is idle.
$P(m, 1)$	$n = 1$ , represents the state in which the server is busy with servicing a data frame and the communication channel is not vulnerable to collisions.
$P(m, 2)$	$n = 2$ , represents the state where the server is seen as idle, but the communication channel is in collision state, where more than one station has contended for the communication channel during a vulnerable period of a transmission.
$P(m, 3)$	$n = 3$ , represents the state in which the server is busy servicing a data frame and the communication channel is vulnerable to collisions.
$P(m, 4)$	$n = 4$ , represents the state in which the server is busy servicing an acknowledgement frame and the communication channel is vulnerable to collisions.
$P(m, 5)$	$n = 5$ , represents the state in which the server is busy servicing an acknowledgement frame and the communication channel is not vulnerable for collisions.

It is important to note that  $m$  is represented by  $N - k$  in any of the six different states and represents the number of stations that are not waiting for service, therefore  $k$  represents the number of stations that are waiting for service.

The description of the flow of the state model follows:

#### **P(N,0):**

Currently no station events are being serviced in the system or waiting to be serviced. Therefore, a probability exists that a new event can be created at  $N$  stations with a Poisson arrival rate of  $\lambda$ . This means that the next event to be serviced will arrive at a rate of  $N\lambda$ , with a probability of  $P(N, 0)$ . After the arrival of the new event, the number of stations that can create a new event is reduced to  $(N - 1)$  and the server will commence in servicing the newly arrived event. The system will change state from  $P(N, 0)$  to  $P(N - 1, 3)$ .



**P(N-1,3):**

In this state there is currently only one station waiting to be serviced and, consequently, it is therefore also the station currently being serviced. The data frame of the station is being serviced, but the communication channel is still vulnerable to interference from other stations as they can't sense that the communication channel is currently busy. In this state a probability exists that a new event can be created at any of the  $(N - 1)$  stations that are not waiting to be serviced. Therefore, a new arrival will occur with an arrival rate of  $(N - 1)\lambda$ , which will occur with a probability of  $P(N - 1, 3)$ . If this arrival occurs before the vulnerable period of the data frame service time has passed ( $(N - 1)\lambda > \phi$ ), a collision occurs and the system will change state to  $P(N - 2, 2)$  where both the newly arrived station event and the one being serviced have data corrupted due to the collision. If, however, the vulnerable time of the data frame transmission passes before a new arrival can occur ( $(N - 1)\lambda < \phi$ ), the system will move to state  $P(N - 1, 1)$  at a rate of  $\phi$  with a probability of  $P(N - 1, 3)$ .

**P(N-1,1):**

In this state, there is still only one station in the system waiting for its event service to be completed, but the communication channel is no longer vulnerable and the rest of the data frame can be sent without any chance of collisions. A new arrival could still occur at a rate of  $(N - 1)\lambda$  with a probability of  $P(N - 1, 1)$  but the new arrival will sense that the channel is busy and initiate a move to state  $P(N - 2, 1)$ . Therefore, one station is now being serviced and another has backed off and is waiting in the queue to be serviced. If, however, the arrival rate is lower than the service rate of the rest of the data frame transmission ( $\mu' > (N - 1)\lambda$ ), the system will change state to  $P(N - 1, 4)$  at a rate of  $\mu'$ , with a probability of  $P(N - 1, 1)$ .

**P(N-1,4):**

In this state the data transmission has been successfully processed by the server and the acknowledgement frame is under way to the station from which the data frame came, but the communication channel is vulnerable, as not all stations have received the signal yet. A new event can arrive at a rate of  $(N - 1)\lambda$  with a probability of  $P(N - 1, 4)$ . If this arrival occurs before the vulnerable period of the acknowledgement transmission has passed ( $(N - 1)\lambda > \phi$ ), the the system will move to the collision state  $P(N - 2, 2)$  where both the station that newly arrived and the one being serviced has data corrupted due to the collision. If however, the vulnerable time of the acknowledgement frame transmission passes before a new arrival can occur ( $(N - 1)\lambda < \phi$ ), the system will move to state  $P(N - 1, 5)$  at a rate of  $\phi$  with a probability of  $P(N - 1, 4)$ .

**P(N-1,5):**

In this state the acknowledgement frame transmission of the station being serviced is no longer vulnerable to collisions and the transmission of the frame is being completed. A new arrival can still occur at a rate of  $(N - 1)\lambda$  with a probability of  $P(N - 1, 5)$  but the new arrival will sense that the channel is busy and initiate a move to state  $P(N - 2, 5)$ . Therefore one station is now being serviced and another has backed off and is waiting in the queue to be serviced. If, however, the arrival rate is lower than the service rate of the rest of the acknowledgement frame transmission ( $\mu'' > (N - 1)\lambda$ ), the system will change state to  $P(N, 0)$  at a rate of  $\mu''$  with a probability of  $P(N - 1, 5)$  completing the service of the first arrival.

**P(N-2,5):**

State  $P(N - 2, 5)$ , is very similar to state  $P(N - 1, 5)$ , except that one station is being serviced, one station is in backoff and  $N - 2$  stations can still arrive with new events at a rate  $(N - 2)\lambda$  with a probability of  $P(N - 2, 5)$ . If a new arrival occurs before the acknowledgement frame transmission of the station being serviced has completed ( $(N - 2)\lambda > \mu''$ ), the the system will transition to state  $P(N - 3, 5)$  and two stations will be in backoff while the acknowledgement frame transmission of the current station being serviced, is still being completed. If, however, the arrival rate is lower than the service rate of the rest of the acknowledgement frame transmission ( $\mu'' > (N - 2)\lambda$ ), then the system will change state to  $P(N - 1, 0)$  at a rate of  $\mu''$  with a probability of  $P(N - 2, 5)$  and the first arrival has completed service. All  $P(N - k, 5)$ ,

where  $N \neq k$ , states from here on will follow the same process as the one described above. Where  $N = k$ , new arrivals are no longer possible and therefore, the system can only transition to state  $P(1, 0)$  (represented as  $P(N - 4, 0)$  in Figure 6.5) still at a rate of  $\mu''$  with a probability of  $P(0, 5)$ .

#### **P(N-1,0):**

In this state the server is idle, one station is in backoff and will return from backoff at rate of  $\gamma_1$  with a probability of  $P(N - 1, 0)$  and  $(N - 1)$  stations does not have any events, but a new event can occur at any of these stations at a rate of  $(N - 1)\lambda$  with a probability of  $P(N - 1, 0)$ . If the arrival rate of a new event is greater than that of the return rate from backoff ( $(N - 1)\lambda > \gamma$ ) then the system will transition to state  $P(N - 2, 3)$  where the newly arrived station's data frame is being serviced. If the backoff rate is greater than the arrival rate ( $(N - 1)\lambda < \gamma$ ), then the system will transition to state  $P(N - 1, 3)$  where the data frame transmission from the station that was in backoff is now being serviced. All  $P(N - k, 0)$ , where  $N \neq k$ , states from here on will follow the same process as the one described above. When  $N = k$ , no new arrivals can occur and therefore the system can only transition to  $P(0, 3)$  at a rate of  $\gamma_N$  (represented as  $\gamma_5$  in Figure 6.5) with a probability of  $P(0, 0)$ .

#### **P(N-2,3):**

In this state there is one station being serviced and one station in backoff. The data frame transmission of the station is being serviced, but the communication channel is still vulnerable to interference from other stations as they can't sense that the communication channel is currently busy. A probability exists that a new event can be created at any of the  $(N - 2)$  stations that are not waiting to be serviced. Therefore a new arrival will occur with an arrival rate of  $(N - 2)\lambda$ , which will occur with a probability of  $P(N - 2, 3)$  if this arrival occurs before the vulnerable period of the data frame service time has passed ( $(N - 2)\lambda > \phi$ ), a collision occurs and the system will move to state  $P(N - 3, 2)$  where both the data of the station that has newly arrived and the one being serviced has been corrupted due to the collision. Furthermore, the station in backoff can also return from backoff at a rate  $\gamma_1$ . If the rate of return from backoff is greater than the vulnerable period service rate ( $\gamma > \phi$ ), then the data transmission of the station returning from backoff and the station being serviced will collide and the system will transition to state  $P(N - 2, 2)$ . If, however, the vulnerable time of the data frame transmission passes before a new arrival can occur or a station can return from its backoff state ( $(N - 1)\lambda < \phi$ ,  $\gamma < \phi$ ), the system will move to state  $P(N - 2, 1)$  at a rate of  $\phi$  with a probability of  $P(N - 2, 3)$ . All  $P(N - k, 3)$ , where  $N \neq k$ , states from here on will follow the same process as the one described above except that  $\gamma_1$  will become  $\gamma_k$ . When  $N = k$ , no new arrivals can occur and therefore the transition to the collision state due to a new arrival occurring in the vulnerable time of the data frame transmission can no longer occur.

#### **P(N-2,1):**

In this state there is one station being serviced and one station in backoff, but the communication channel is no longer vulnerable and the rest of the data frame can be sent without any chance of collisions. If the station currently in backoff should return, it will sense the channel is busy and go into backoff again. This is not shown in the diagram, as no state transition takes place. A new arrival can still occur at a rate of  $(N - 2)\lambda$  with a probability of  $P(N - 2, 1)$  but the new arrival will sense that the channel is busy, go into backoff (thus join the queue of stations waiting) and cause a transition to state  $P(N - 3, 1)$ . If, however, the arrival rate is lower than the service rate of the rest of the data frame transmission ( $\mu' > (N - 2)\lambda$ ), then the system will change state to  $P(N - 2, 4)$  at a rate of  $\mu'$  with a probability of  $P(N - 2, 1)$ . All  $P(N - k, 1)$ , where  $N \neq k$ , states from here on will follow the same process as the one described above. Where  $N = k$ , new arrivals can no longer arrive and, therefore, the system can only transition to state  $P(0, 4)$  still at a rate of  $\mu'$  with a probability of  $P(0, 1)$ .

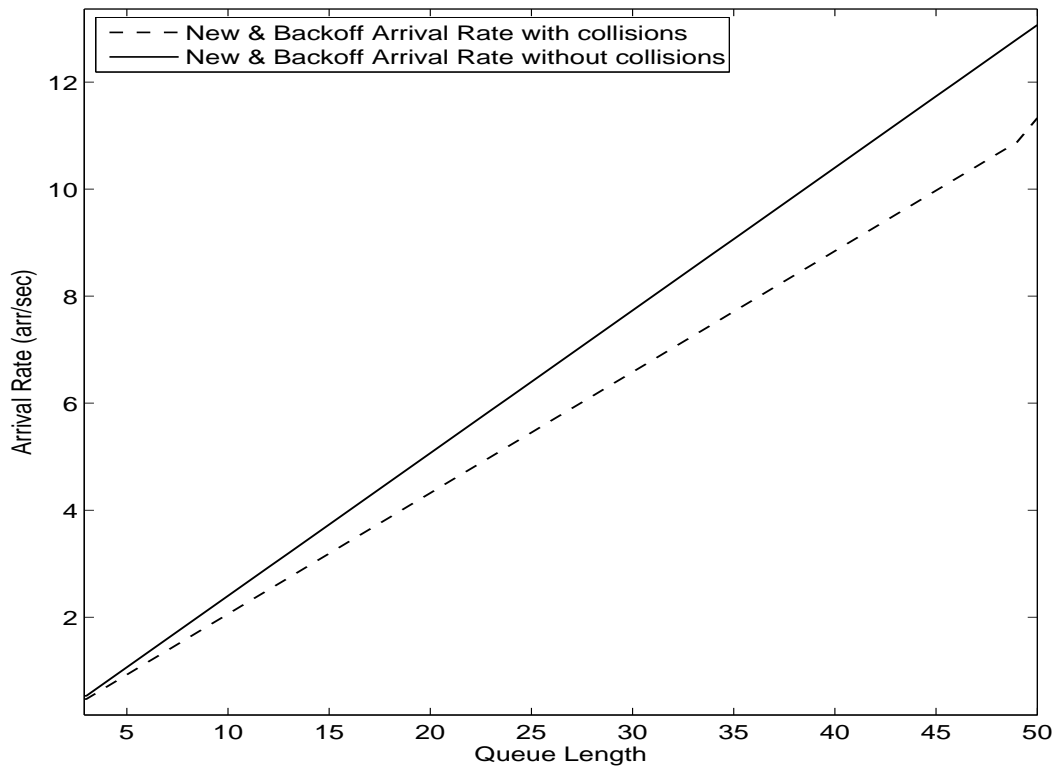
#### **P(N-2,4):**

In this state the data transmission has been successfully processed by the server and the acknowledgement frame is under way to the station from which the data frame came, but the communication channel is vulnerable as all stations have not

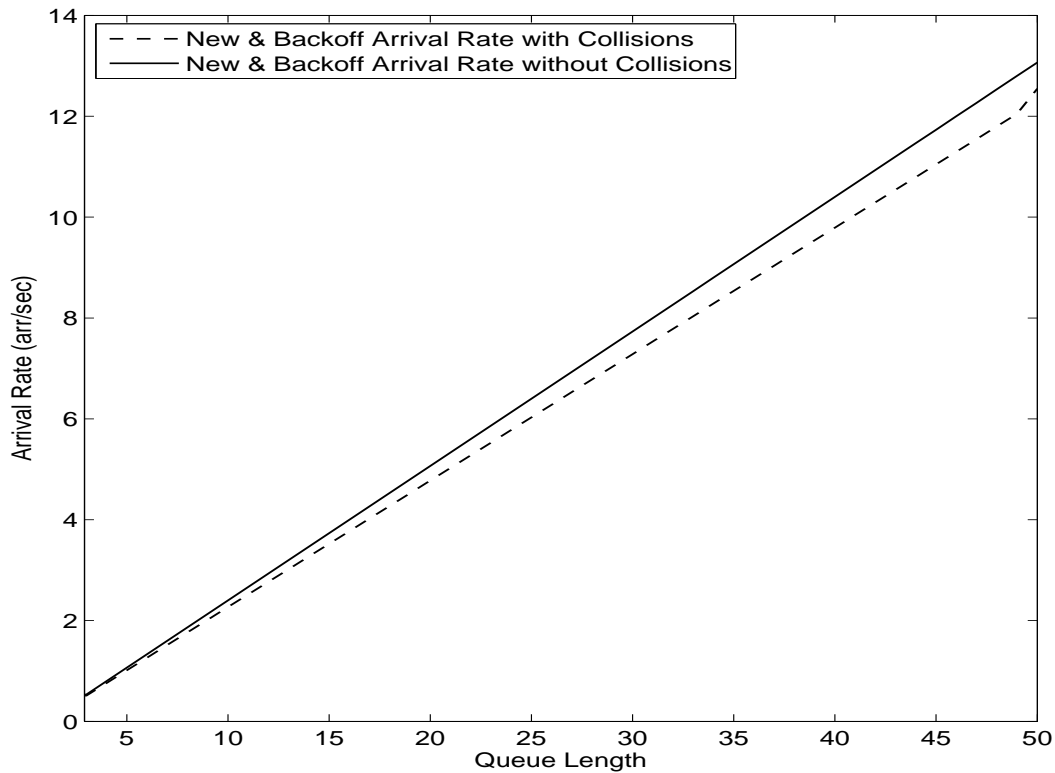
yet received the signal. A new station can arrive at a rate of  $(N-2)\lambda$  with a probability of  $P(N-2,4)$ . If this arrival occurs before the vulnerable period of the acknowledgement transmission has passed ( $(N-2)\lambda > \phi$ ), the system will move to the collision state  $P(N-3,2)$  where the data of the station that has newly arrived and that of the one being serviced has been corrupted due to the collision. If, however, the vulnerable time of the acknowledgement frame transmission passes before a new arrival can occur ( $(N-2)\lambda < \phi$ ), the system will move to state  $P(N-2,5)$  at a rate of  $\phi$  with a probability of  $P(N-2,4)$ . Furthermore, the station in backoff can also return from backoff at a rate  $\gamma_1$ . If the rate of return from backoff is greater than the vulnerable period service rate ( $\gamma > \phi$ ), then the data transmission of the station returning from backoff and that of the station being serviced will collide and the system will transition to state  $P(N-2,2)$  at a rate of  $\gamma_1$  with a probability of  $P(N-2,4)$ . If, however, the vulnerable time of the data frame transmission passes before a new arrival can occur or a station can return from its backoff state ( $(N-2)\lambda < \phi$ ,  $\gamma < \phi$ ), the system will move to state  $P(N-2,5)$  at a rate of  $\phi$  with a probability of  $P(N-2,4)$ . All  $P(N-k,4)$ , where  $N \neq k$ , states from here on will follow the same process as the one described above except that  $\gamma_1$  will become  $\gamma_k$ . Where  $N = k$ , there are no new arrivals that can any longer arrive and therefore, the system can only transition to state  $P(0,5)$  still at a rate of  $\phi$  with a probability of  $P(0,4)$  if ( $\phi > \gamma_{N-1}$ ), but if this is not the case, the transition will be to the collision state  $P(0,2)$  at a rate of  $\gamma_{N-1}$  (represented as  $\gamma_4$  in Figure 6.5) with a probability of  $P(0,4)$ .

### **P(N-2,2):**

In this state the server is not available and the communication channel is in a state of collision. A station returning from backoff while the system is in a collision state will also collide with the already collided frames, but will not cause a change of state within the system. If a new arrival should occur while the system is in this state, it will also be in a state of collision. A new station can arrive at a rate of  $(N-2)\lambda$  if this occurs before the frame transmission of the last station that joined the collision state has completed, the newly arrived station will also be in a state of collision and the system will transition to  $P(N-3,2)$  at a rate of  $(N-2)\lambda$  with a probability of  $P(N-2,2)$  if  $(N-2)\lambda > \delta_1$ . If, however, the next arrival occurs only after the frame transmission of the last station that joined the collision state is completed  $(N-2)\lambda < \delta_1$ , the system will transition to state  $P(N-2,0)$  at a rate of  $\delta_1$  with a probability of  $P(N-2,2)$ . The same process as the one described above will be followed for all  $P(N-2,k)$  states where  $N \neq k$  and the collision return rate equal to  $\delta_k$ . Where  $N = k$ , there are no longer new arrivals that can arrive and therefore, the system can only transition to state  $P(0,0)$  at a rate of  $\delta_{N-1}$  (represented as  $\delta_4$  in Figure 6.5) with a probability of  $P(0,2)$ . Because of collisions, the actual backoff rate will become lower due to the extra time added to the backoff of a station when its timeout period is factored in. A typical example is shown in Figure 6.6 which depicts the total arrival (backoff and new arrivals) when no collision occurs and when collisions occur.



(a) Total Arrival Rate with Timeout Equal to 5000 ms



(b) Total Arrival Rate with Timeout Equal to 1000 ms

Figure 6.6: Effect of Collisions on Total Arrivals

To determine the actual backoff rate,  $\gamma_k$ , the utilization must first be determined when  $k$  stations have events, which can then be used to determine the actual backoff rate per queue length  $k$ , as shown in Equation 6.24.

$$\lambda_{tot-k} = \frac{(N-k)\lambda}{k} \gamma + 1 \quad (6.22)$$

$$Util = \frac{\lambda_{tot-k}}{\mu} \quad (6.23)$$

$$\gamma_k = (k \times Util) \gamma + (k(1 - Util)) \left( \frac{1}{\frac{1}{timeout} + \frac{1}{\gamma}} \right) \quad (6.24)$$

The rate at which stations return from collisions,  $\delta$ , will also change with queue length  $k$ . A simplified calculation has been used in which  $\delta_1 = \delta_2 = \dots = \delta$ . An average has been determined for a specific new event arrival rate that will only be used in determining the collision return rate,  $\delta$  and the process of obtaining it is shown below.

$$\mu_\delta = \frac{1}{t_{ser}} \quad (6.25)$$

$$\lambda_{col-k} = (N-k)\lambda + Util \left[ (k(1 - Util)) \left( \frac{1}{\frac{1}{timeout} + \frac{1}{\gamma}} \right) + ((N-k)(1 - Util) \gamma) \right] \quad (6.26)$$

$$\delta_{util} = \frac{\sum_{k=1}^N \left[ \frac{\lambda_{col-k}}{\mu_\delta} \times \frac{\gamma_{col-k}}{k\gamma} \right]}{N} \quad (6.27)$$

$$\delta_{factor} = (N\gamma - \gamma_{col-N}) \times \left( 1 - \delta_{util} \times \frac{N\lambda}{\mu} \right) \quad (6.28)$$

$$\delta = \frac{\mu_\delta + \delta_{factor}}{2} \quad (6.29)$$

where:

$\mu_\delta$	The service rate per collisions
$\lambda_{col-k}$	New event arrival rate that takes into account arrivals that are returning from collision as well as backoff
$\delta_{util}$	Utilization of the channel as a result of collisions due to backoff arrivals and new arrivals
$Util$	Utilization of the channel as a result of new arrivals
$\delta_{factor}$	The rate at which the collision service rate is increased or decreased.
$\delta$	The new service rate per collision taking into account the average backoff rate, including collisions, and the arrival rate of new events to the communication channel
$N$	Number of stations

The results of the expanded state space model is discussed and compared to the simulation model in 6.4.

### 6.3 Simulation Modelling

As discussed in Chapter 3, DESMO-J has been used to implement the simulation of the non-persistent CSMA protocol, as discussed in the previous section. A process oriented modelling approach has been taken to implement the protocol. The protocol has been implemented as outlined in Figure 6.1 with the following assumptions being made:

- The arrival rate is a Poisson process with exponentially distributed arrival times
- Once an event occurs at a station, no other event occurs at the station until the first has been serviced

- The data frame sizes are exponentially distributed between a minimum and maximum size
- The bit error rate is modelled as part of the service time of the acknowledgement and data frames
- The timeout period after a collision is static

### 6.3.1 Processes-Orientated Modelling

The process based modelling design takes an object orientated perspective making use of Java and DESMO-J. All the activities of a specific entity are grouped together as a process and are viewed as the process's life cycle. Each life cycle of a process is described from the entity's perspective and contains the activities of the specific entity as well as the entity's relationship with other entities. The life cycle of each process is described using diagrams based on UML activity diagrams. The non-persistent CSMA protocol is broken down into three main processes:

- The Generator process which creates new arrivals
- The Station process which describes the contention process of a station when a new event arrives
- The Server process which describes the service and acknowledgement of events from different stations

These processes are described in detail in the following sections with the help of UML based activity diagrams.

#### 6.3.1.1 The Generator Process

The life cycle of the Generator process is shown in Figure 6.7. The ID of a specific station is not relevant in the simulated non-persistent CSMA protocol and therefore any arrival occurring is seen as a new station with data to transmit to the server. The number of stations that can arrive is defined by the number of stations in the network. When simulation starts, a new station with data is created. The statistical counter used to calculate the average queue length at the end of simulation, is incremented by one. The statistical counters are black box objects from DESMO-J and can be chosen to measure the specific elements of the model required for analysis at the end of simulation. In this specific case, the queue length is modelled as a Tally statistical counter object. According to the DESMO-J API, the Tally class provides the mean value and standard deviation of a specific value and is calculated on the basis of the total number of observations. After the station process has been created and the queue length statistics updated, the newly created station process is scheduled to be activated after the Generator process has been passivated. This means that the scheduler inserts the newly created station process into the event list to be activated after the current active process (the Generator process) is passivated. After passivation, a process can be re-activated either by another process, or when it is scheduled to re-activate. The last mentioned applies to the Generator process. The next station is created when the next event occurs, but the arrival rate has reduced, as one station is already waiting to be serviced and therefore can not produce another event under the assumption mentioned earlier. The average arrival rate is reduced by the arrival rate generator (which produces exponential arrival times). The next arrival time is determined and the Generator process is scheduled to re-activate at that time. The scheduler will schedule the new time to activate the Generator process and enter it into the event list. The Generator is then passivated. When re-activated, the process is repeated, but if the queue is full (all stations have events to be serviced), the Generator process will passivate itself. In the last mentioned activity, the scheduler does not insert the Generator into the event list, as it is not scheduled for re-activation at a specific time, but must wait to be activated by another process.

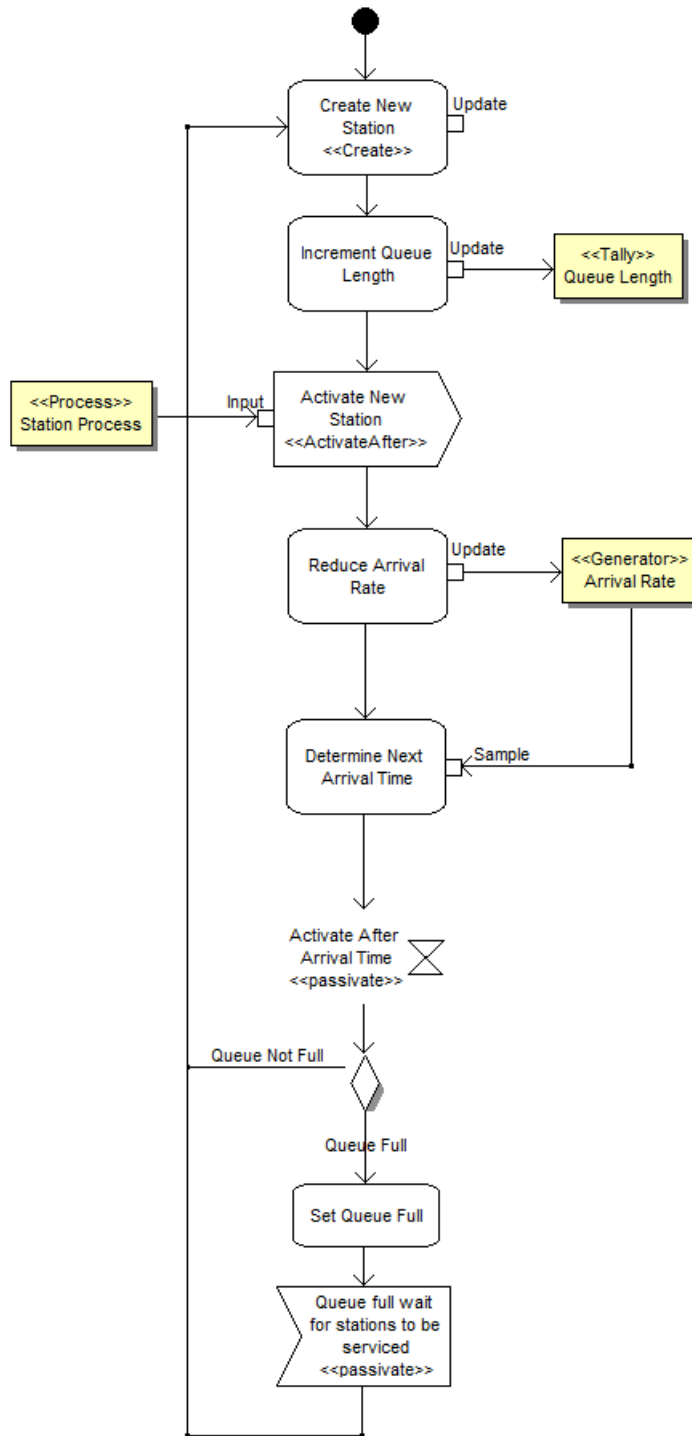


Figure 6.7: Generator Process Activity Diagram

### 6.3.1.2 The Station Process

The Station process life cycle is shown in Figure 6.8. A new Station process is created by the Generator process and activated. When activated, the Station process will store the current simulation process time used in determining the total time that the station waits to complete its event service. If this is the first Station process activated in the simulation, it will set the Server process as unavailable to other Station processes and create an instance of the Server process. There

can only be one Server process, as the protocol makes use of only one server. The Station process will schedule the server to be activated after it has been passivated and set itself as the active station. The active station is the first to contend for service time or is the station that is currently being serviced by the Server process. The Station process will then passivate itself to be re-activated by the Server process. Re-activation occurs either after service has been completed, or when the Server process has determined that a collision has occurred. Upon re-activation, the Station process will check to see if its service has been completed, or if it has been involved in a collision.

If the service has been completed, the process will get the current time of the simulation and subtract the start time from it to get the service waiting time for the station. This time is then used to update the statistical counter used to calculate the average waiting time at the end of the simulation. The Waiting Time counter is again defined as an instance of the Tally class as described in Section 7.4.1.1. With the service of the station completed, it can again create new events and, therefore, the arrival rate generator must be updated by incrementing the average arrival rate by the arrival rate of one station. The total number of stations waiting for service has also decreased and the Queue Length statistical counter must also be updated by decreasing its length by one. If the reduction in queue length has caused the queue length to decrease from its maximum value, the Generator process will be scheduled to be activated next by adding it to the event list. After the Station process has scheduled the Generator to be activated next, if required, the process is completed and the instance of the Station process will be terminated and garbage collected by Java.

If the process, upon re-activation, finds that its service has not been completed and a collision has occurred, it will obtain the time provided for timeout. As it is the active station and last notified of the collision by the Server process (at this time the vulnerable period has passed), it will reset all the flags to indicate that a collision has occurred and make the server unavailable to all stations. After the vulnerable period of a transmission has passed, all stations in the system should know that the communication channel is busy. The stations that contended for service time will be waiting for a reply from the server that will never come, therefore they will timeout and back off. The server is therefore unavailable to all. The remaining data frame transmission time of the last station in the collision is determined and the active station is passivated for this time period. With the Server process unavailable and the collision and vulnerable flags set to false, all stations arriving with new events will go into backoff until this specific transmission time has passed. When this time has passed, a new timeout period is calculated for the active station by subtracting the time that has passed since the station's data transmission started. The station is then scheduled to re-activate at the end of this time. After the station returns from its timeout period it will schedule a backoff time. The backoff time is determined by making use of a Backoff Time generator, which is an instance of DESMO-J's uniform time generator (DiscreteDistUniform) which is bounded by a minimum and a maximum backoff time. The station will be scheduled to re-activate after the newly determined backoff time has passed, entered into the event list and passivated. When the station is re-activated it will again try to access the channel.

If a newly initiated Station process is activated and it is not the first station to do so, it will try to access the communication channel to be serviced. For this to occur, the server must be available. If the server is not available, the station will schedule a backoff time, be placed in the event list and passivated. After re-activation, the station will check to see if the server is available. If not, the process is repeated. If the server is available, but the channel is vulnerable (indicated by the vulnerable flag), the Station process will obtain its timeout period and set the collision flag to true. The station process will now be scheduled to re-activate after its timeout period, entered in the event list and passivated. After re-activation, a new backoff time is determined and the station is again scheduled for re-activation after this time, entered into the event list and passivated, after which it will again try to gain access to the server.

Finally, if a station should find the server available, no collisions have occurred and the channel is not vulnerable, the station will be able to capture the channel. The station process will then schedule the Server process for activation and enter it into the event list as the next process to be activated. The station sets itself as the active station and goes passive waiting to be activated after service completion, or if a collision has occurred.





### 6.3.1.3 The Server Process

The Server process is depicted in Figure 6.9. The Server process handles the transmission of data and acknowledgement frames. When activated it immediately sets the vulnerable flag as true, calculates the vulnerable time, schedules the server to re-activate after this time and passivates. Upon re-activation, the server will check the collision flag to determine whether a collision has taken place. If true, it will schedule the active station to be activated and enter it into the event list as the process to be activated next. The server will passivate itself and be activated by the next active Station process.

If the collision flag has not been set to true, no collisions have occurred during the vulnerable period of the transmission of the data frame. Therefore the server is set as unavailable and the rest of the data transmission time is determined, after which the Server process will be re-activated. The server is again entered into the event list and passivated. After re-activation, the server will start with the transmission of the acknowledgement frame. The vulnerable flag is set, vulnerable time calculated and the server is scheduled to re-activate afterwards. The Server process is entered into the event list and passivated. Once again, upon re-activation a collision could have occurred and the same process as before will be followed. If no collisions have taken place, the remaining time of the acknowledgement frame transmission is determined and the server is scheduled to re-activate afterwards and entered into the event list. The Server process is set to unavailable and passivated. Re-activated, the Server process has successfully completed the service of the active station's event. The station service done flag is set to true, the active Station process is scheduled to be activated next and entered into the event list. The Server process is set to available and passivated afterwards.

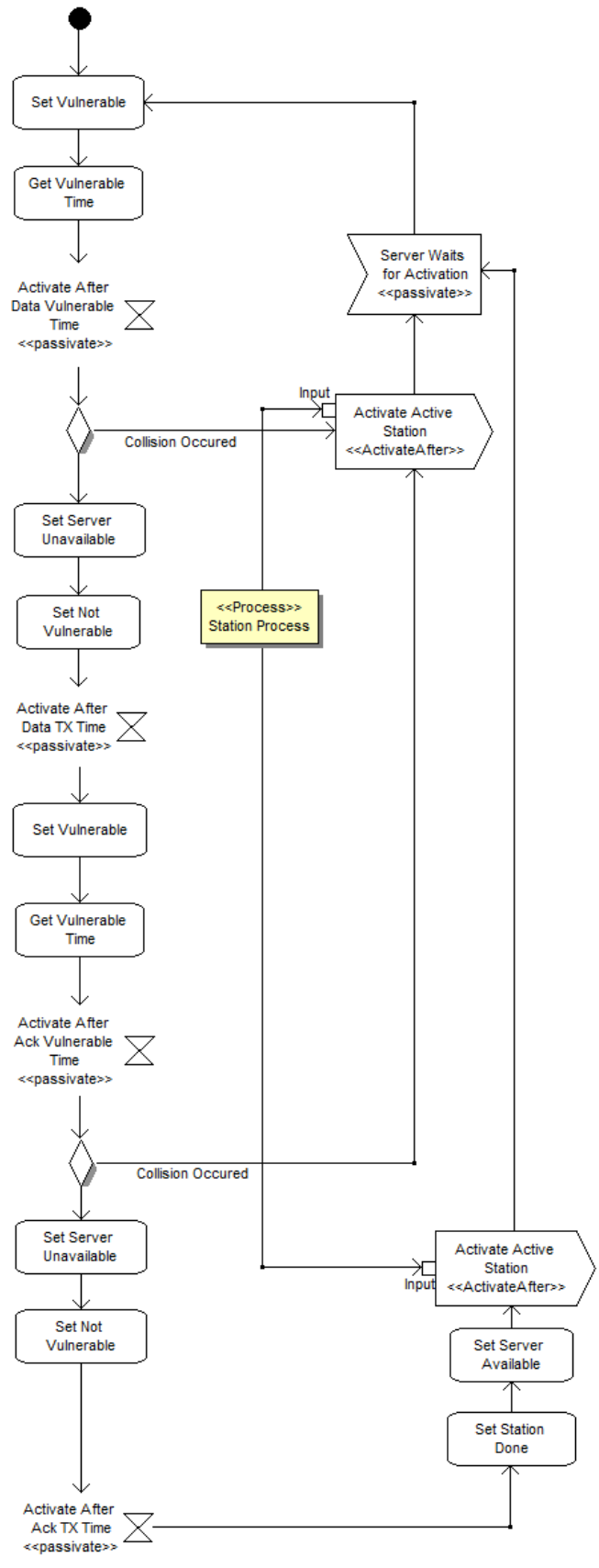


Figure 6.9: CSMA Server Process Activity Diagram

### 6.3.2 Simulation Model Layout

The process lifecycles described in the previous section are the heart of the program. All processes are defined as part of a model. A model class which includes three main abstract methods that must be implemented must therefore be created extending the DESMO-J model class. These methods are:

- `description()` method which is used to define a model description
- `init()` method which is used to initialize all statistical counter, random generators and other important model variables. Examples are the flags indicating active station, vulnerable, collision and server available.
- `doInitialSchedules()` method which is used to create and define the process which will activate the model at the beginning of the simulation. In our model this process is the Generator process described in Section 7.4.1.1.

As mentioned in the previous section, processes are activated, re-activated, scheduled, entered into event lists and passivated. These activities are handled by the scheduler, which is part of the experiment class of DESMO-J. The experiment class is a black box class and is only initiated. An initiated model class is connected to an initiated experiment class and used to run the simulation model. The program itself runs various simulations increasing the average arrival rates to determine the effect on the waiting time and queue length of the system. The next section describes the setup of the class used to initiate the model and experiment classes, as well as the management of running numerous simulations and storing the results for analysis.

### 6.3.3 Model Control Class

A flow diagram of the model control class is shown in Figure 6.10. The class begins by getting all the user defined variables from the graphics user interface (GUI) and defining them in such way that they are available to the process class instances. The model class makes use of static variables such as the minimum and maximum backoff times and data frame sizes; these are defined and initialized next. In the GUI a minimum and maximum arrival rate, as well as step size, is defined which is used to determine the number of simulations that must be executed to obtain the required statistical values. The minimum or begin arrival rate is set and a new experiment class variable is defined and initialized, after which the CSMA model class (extending the DESMO-J user class as discussed in the previous section) is defined and connected to the newly initialized experiment class variable. When connecting the experiment to the model, the `init()` method is executed, initializing all statistical counters, generators and other important model variables. A transient time is set for each simulation. This is not user defined but hard coded and is set between 600 and 3600 seconds which should suffice for reaching steady state in the protocol modelled. The experiment is then started, executing the `doInitialSchedules()` method of the model class. The scheduler, which is part of the experiment class, loads the Generator process as the first event in the event list and activates it, after which it schedules new processes to be activated and re-activates and passivates existing process instances as they are defined in the three process life cycles described before. After the transient simulation time has been reached, the simulation is paused. At this point the statistical counters used in the model are reset so that only steady state statistics is gathered. The steady state simulation time is set in the experiment as defined in the GUI. The experiment is resumed again stopping at the end of the steady state user defined time. The statistical counters are finalized, averages obtained and written into the CSMA database which can either be MySQL or MSSQL databases, although MSSQL is preferred and all results in this thesis have been provided from it. After this has been done, the first simulation is completed.

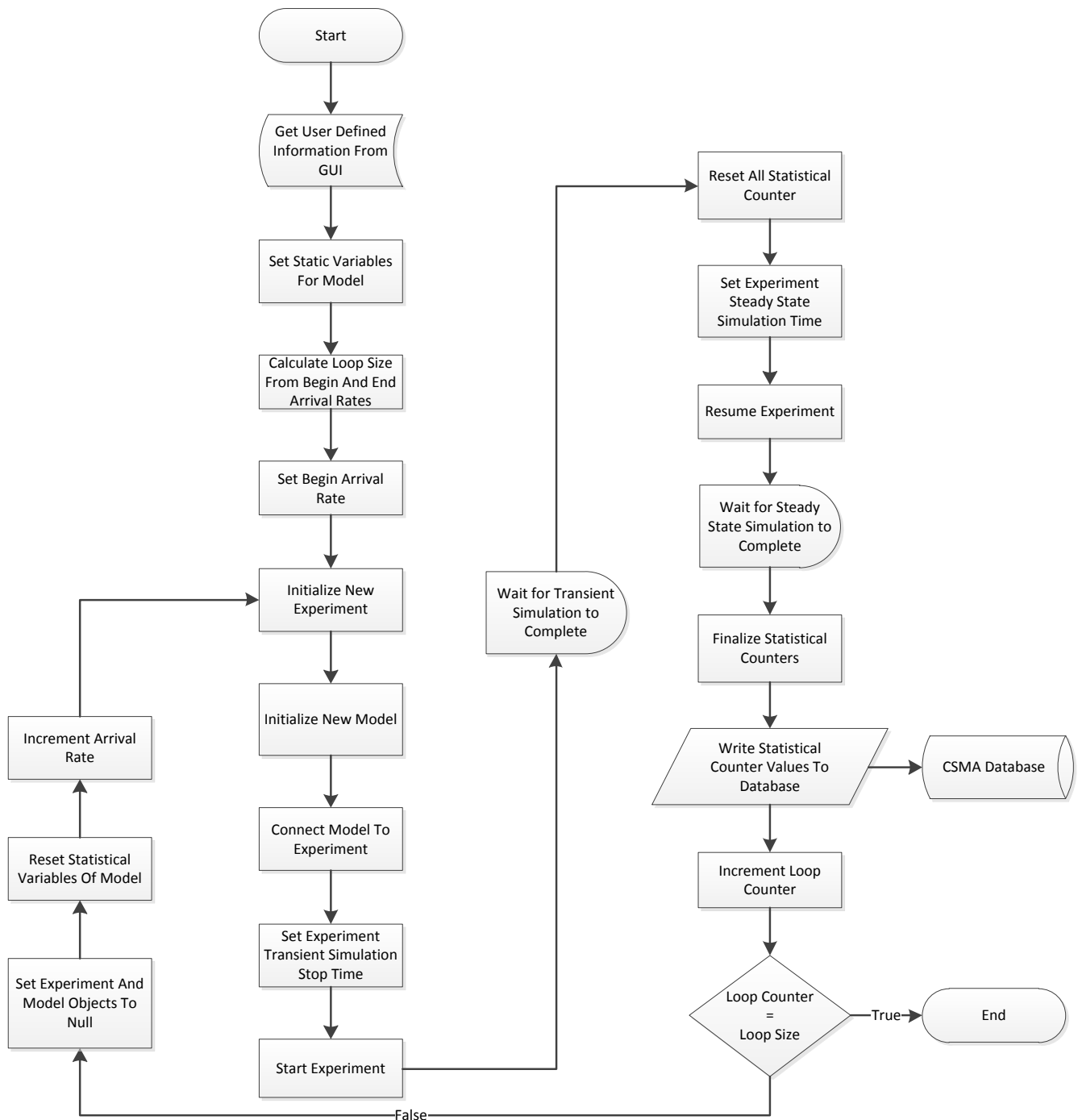


Figure 6.10: CSMA Control Class Flow Chart

The loop counter is incremented and weighed against the number of simulations to be computed. If all simulations have not been completed, the model and experiment objects and static variables of the model class are reset and the new arrival rate is incremented by the simulation step size as defined by the user. A new experiment is defined, initialized and the same process as described above is followed. When the loop counter equals the number of simulations to be computed, the simulation is completed and the statistics can be extracted from the database and analysed. It is these results that are shown in the following section. The Model Control Classes of the two remaining protocols are similar to the one described above, except that the statistical analysis is stored in different databases to allow the results of the different

protocols to be compared.

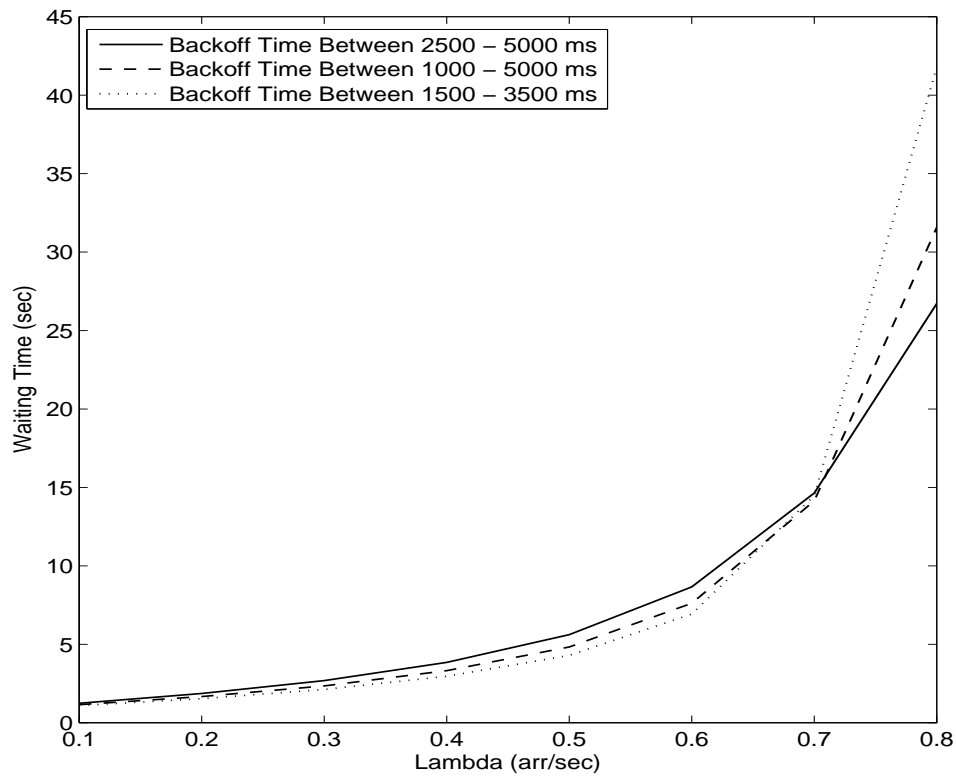
## 6.4 Results

### 6.4.1 Theoretical Results

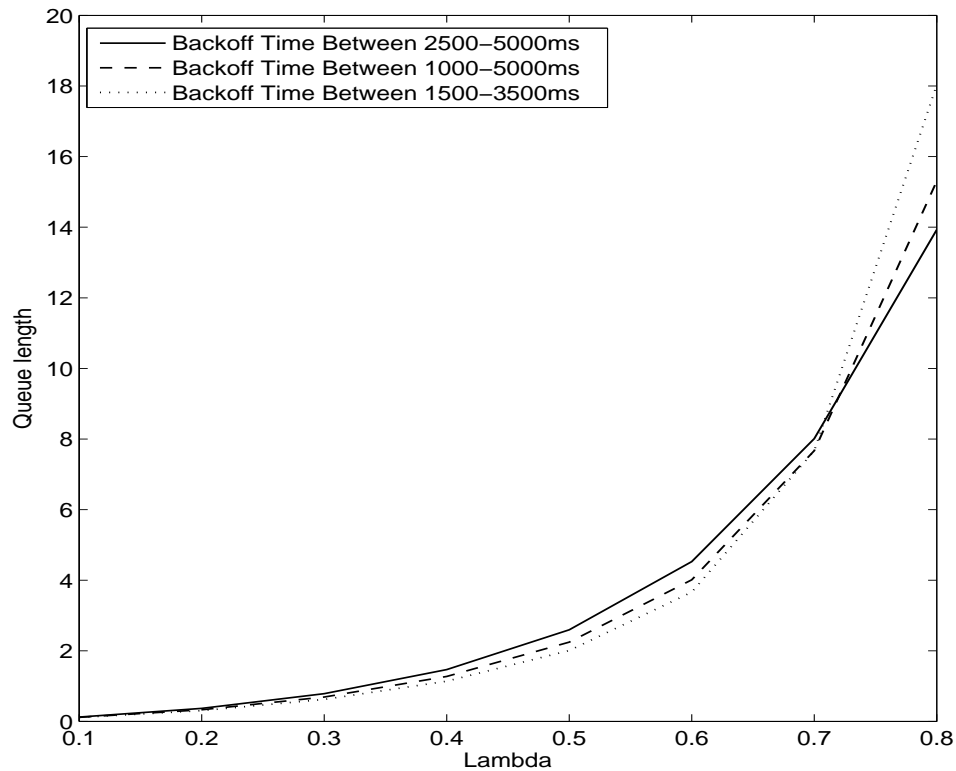
The theoretical results provided make use of the parameters shown in Figure 6.11. The backoff rate have been altered to show three different results in Figures 6.12 and 6.14 with a constant timeout of 5000ms. Varying timeouts have been used in Figures 6.13 and 6.15. The timeout periods have been chosen to be 2, 4, 6 and 8 times the total service time required for a station event with a backoff rate between 1000 and 5000ms. Large and small frame sizes have been used, with 180 bytes in Figures 6.12 and 6.13 and 47 bytes in Figures 6.14 and 6.15.

Number of stations [1.250]	50
Min Propagation distance [Km]	30
Max Propagation distance [Km]	80
Backoff time min [ms]	5000
Backoff time max [ms]	2500
Preamble [ms]	1
Postamble [ms]	1
Data latency [ms]	11
Frame Length data [bytes]	180
Frame Length info [bytes]	20
Channel Capacity [bps]	4800
RXend [ms]	30
Step Size	0.1
Min Arrival Time(arr/sec)	0.1
Max Arrival Time(arr/sec)	0.8
Time Out (ms)	5000
Number of Repeaters	3
Turn Around Time (ms)	1

Figure 6.11: CSMA Theoretical Parameters



(a) Waiting Time

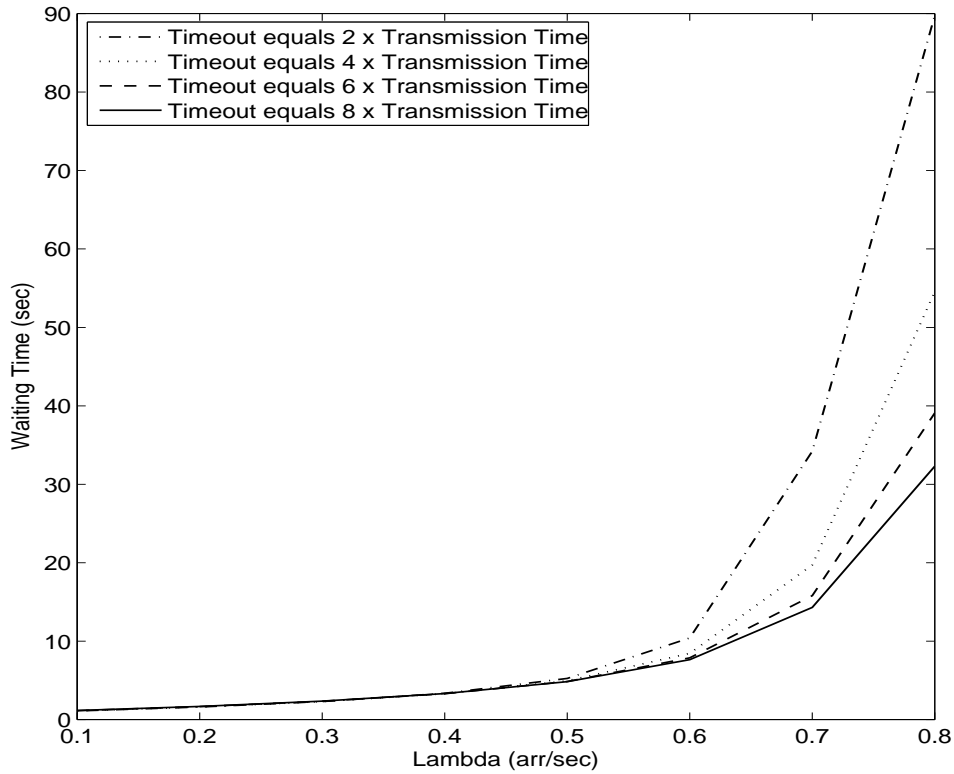


(b) Queue Length

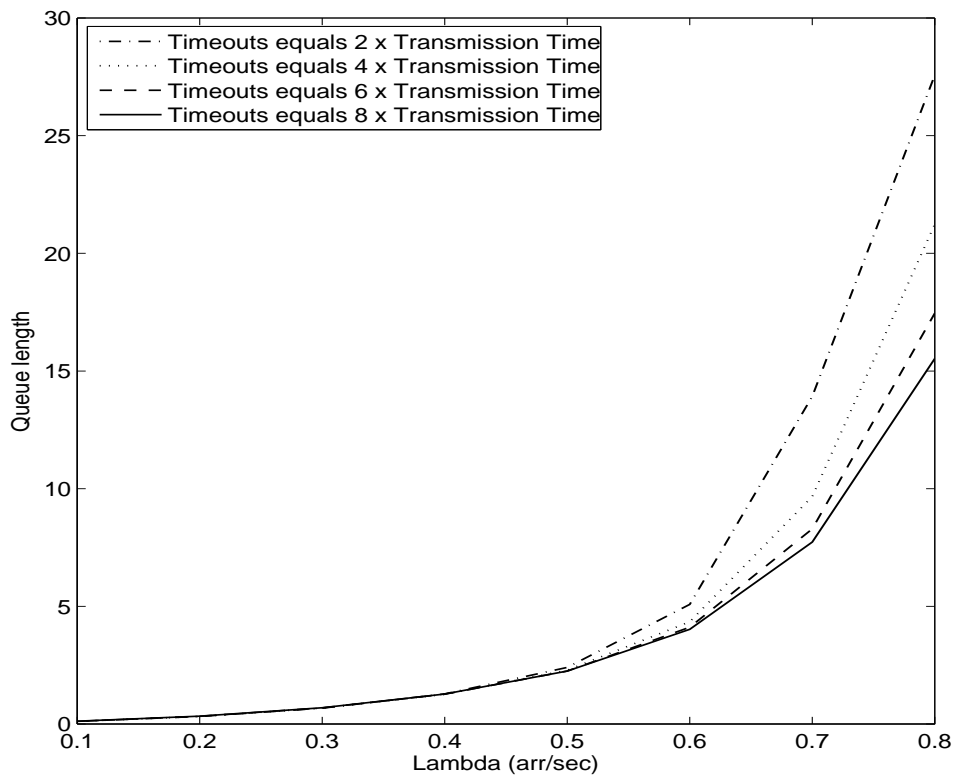
Figure 6.12: CSMA Theoretical Results: Large Frame Sizes

Figure 6.12 shows that at lower arrival rates and therefore at lower channel utilization, smaller backoff rates produce a shorter waiting time and are more efficient, but as the utilization increases and approaches 1, the lower backoff rates reduce in efficiency. The utilization approaches 1 between an arrival rate of 0.6 and 0.7 arrivals per second, where a definite increase in the waiting time for all backoff rates occurs. It is also clear that stations with lower backoff rates will

now wait longer for service, because of more collisions occurring as the channel is sensed more often.



(a) Waiting Time



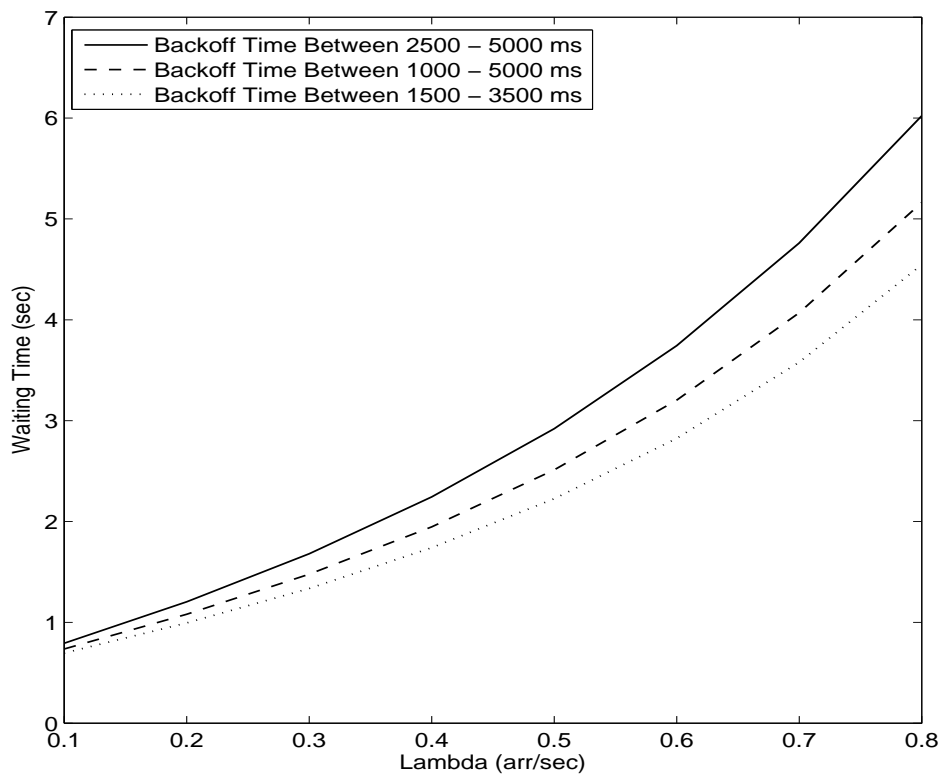
(b) Queue Length

Figure 6.13: CSMA Theoretical Results: Varying Timeouts for Large Frame Size

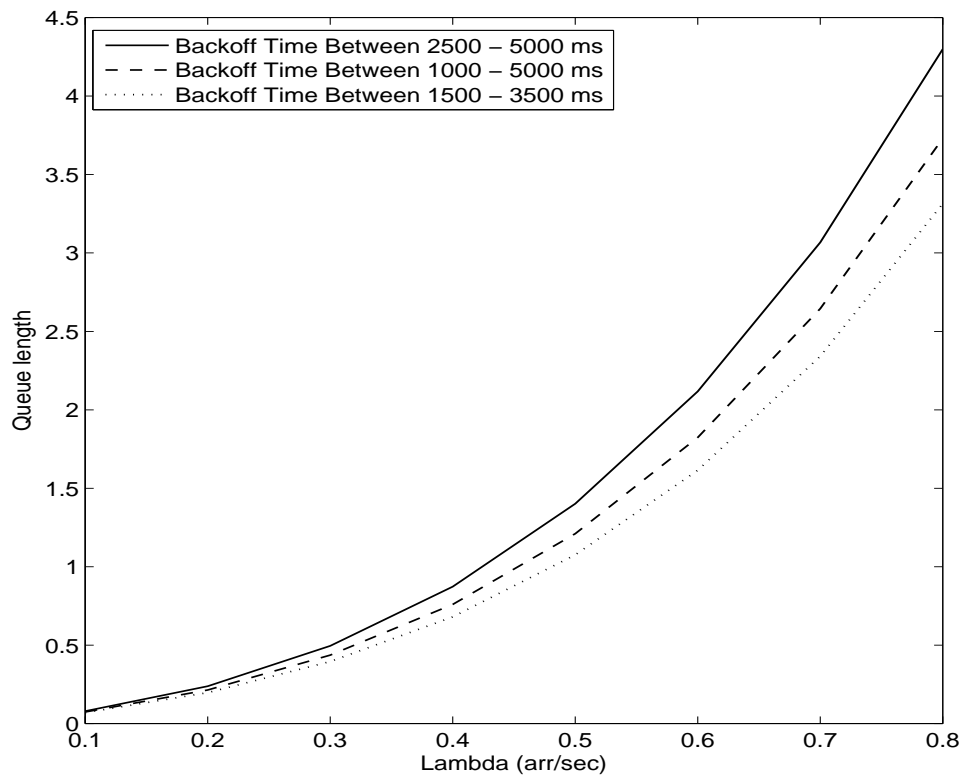
Figure 6.13 shows that with low channel utilization, the different timeout rates do not have a significant effect on the waiting time of each station in the network, but as the utilization approaches 1, the lower timeout rates increase the



waiting time substantially. The timeouts lengthen the actual backoff time when a collision occurs and therefore the higher the channel utilization, the higher the probability of collision with lower timeouts. The timeout also affects the data and acknowledgement transmission time, as the probability for error due to burst noise is integrated into this time. This means that there is an intricate balance between high and low timeouts, and these must be chosen carefully.

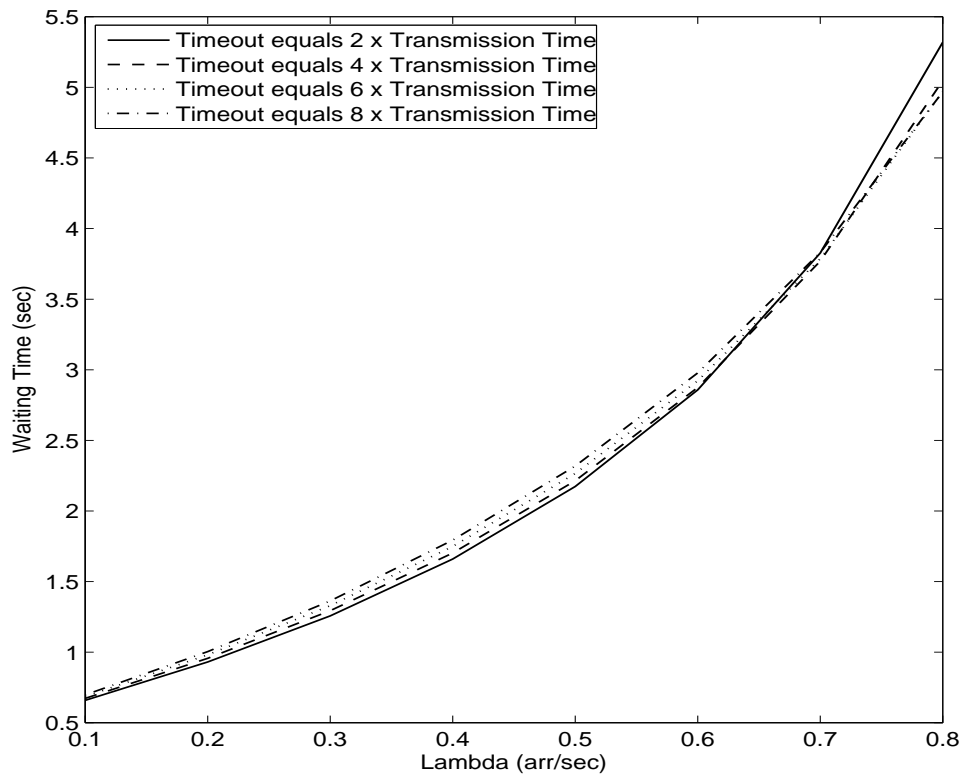


(a) Waiting Time

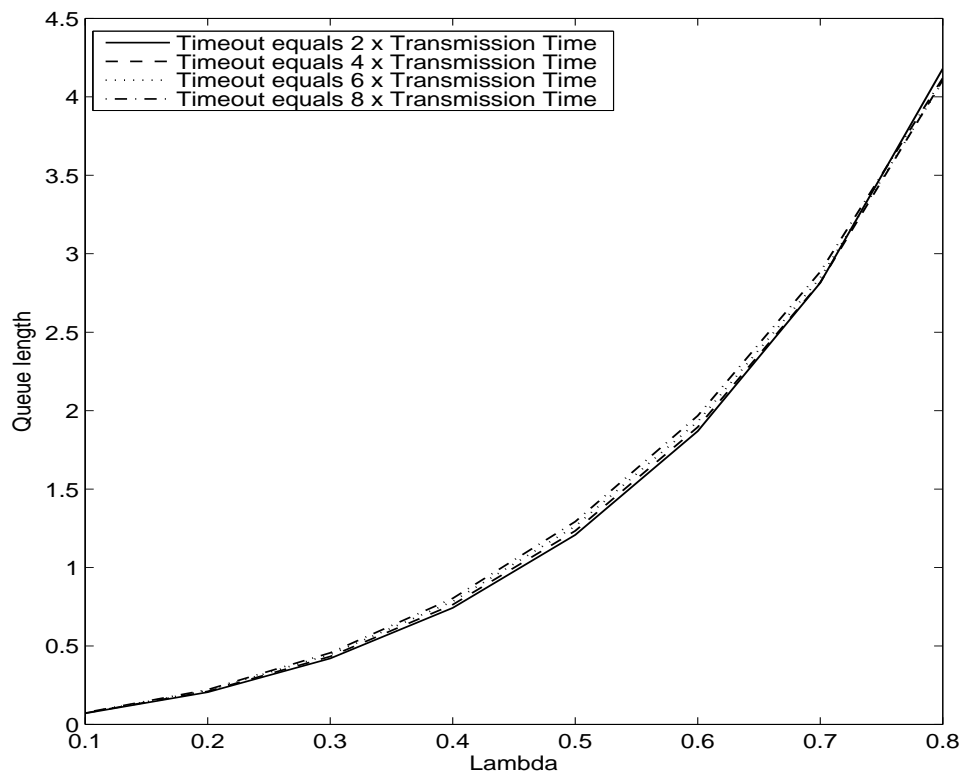


(b) Queue Length

Figure 6.14: CSMA Theoretical Results: Small Frame Sizes



(a) Waiting Time



(b) Queue Length

Figure 6.15: CSMA Theoretical Results: Varying Timeouts for Small Frame Size

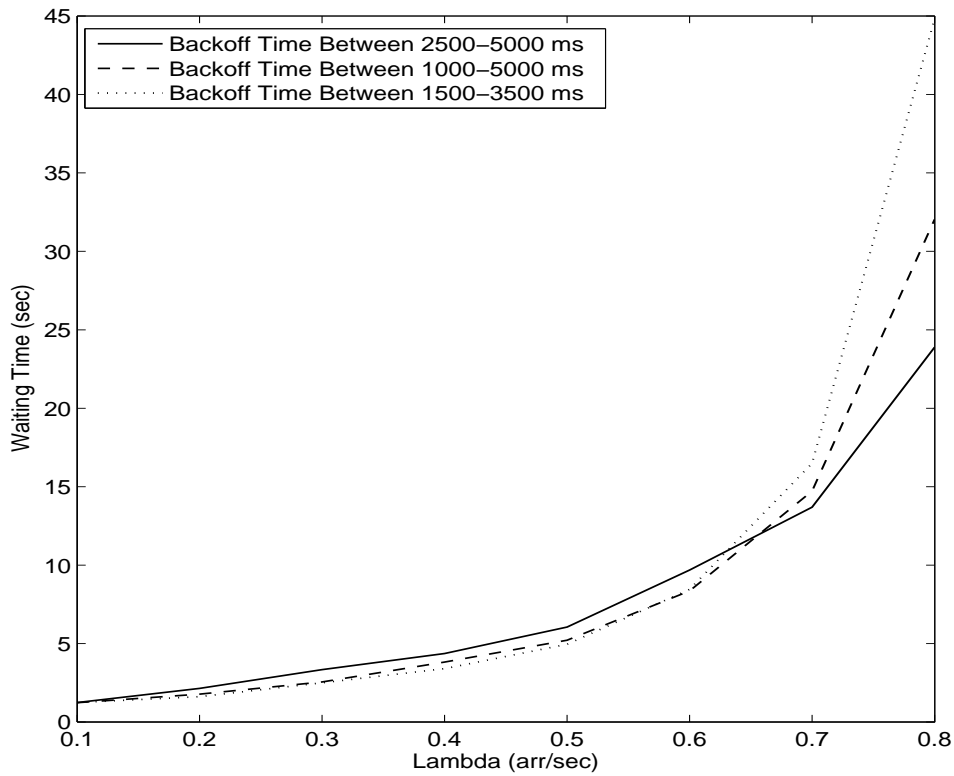
Figures 6.14 and 6.15 show the results for smaller frame sizes. The same principles as mentioned above still apply, but smaller frame sizes provide for a faster service time and the utilization will only reach 1 at higher arrival rates. This is clearly seen in Figure 6.14 where the lower average backoff rate results have the lowest waiting times for all arrival rates. The different timeout periods have very little effect as the utilization is low and few collisions occur.

## 6.4.2 Simulation Results

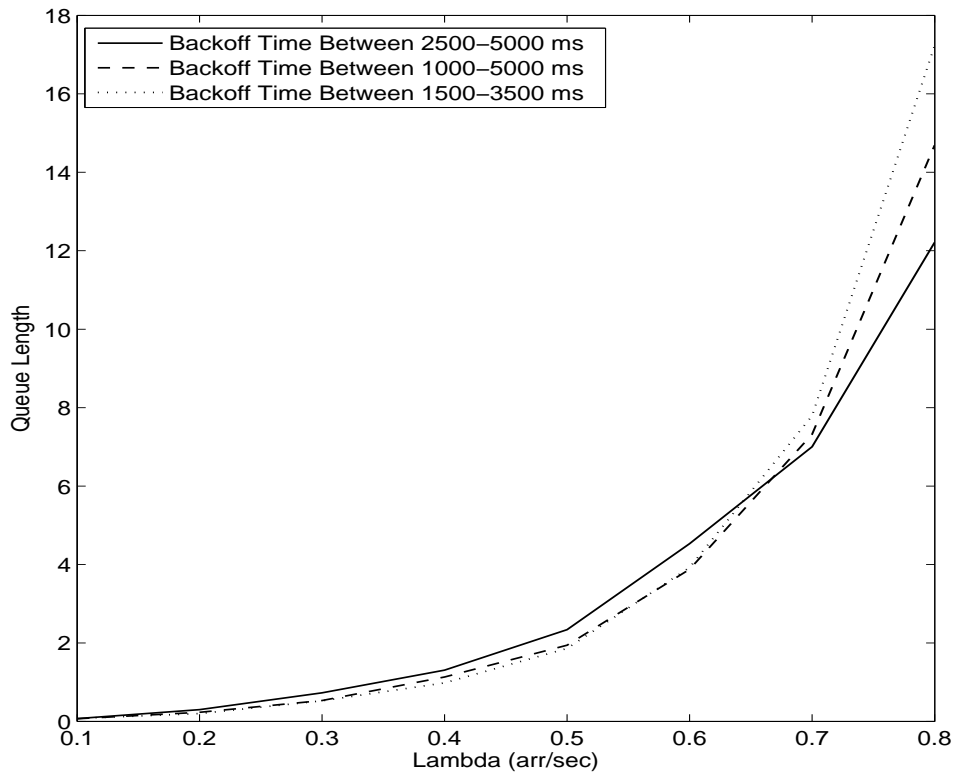
The simulation model results are determined by making use of the parameters shown in Figure 6.16. Some of the parameters will be adjusted for various results shown. The backoff and timeout values used are chosen as realistically as possible. Using the parameter set, three different graphs are drawn. The first is for the backoff range as given, the second with a backoff range 1000 to 5000 ms and the third with a range of 1500 to 3500 ms. With a lower average backoff rate the protocol tends to perform better at lower arrival rates, but quickly loses performance with an increase in arrival rate. This makes sense, as the more frequent return from backoff together with a high arrival rate increase the probability of collision as more stations sense the channel to contend for service time.

Preamble(ms)	1
Postamble(ms)	1
RXEnd(ms)	30
Data Latency(ms)	11
Information Bytes	20
Data Bytes Minimum	170
Data Bytes Maximum	190
Lower Distance(km)	30
Upper Distance(km)	80
System Arrival rate begin(arv/sec)	0.1
System Arrival rate end(arv/sec)	0.7
Simulation Time length(sec)	25000
Step size(sec)	0.1
Backoff Start(ms)	2500
Backoff End(ms)	5000
Channel Capacity (bps)	4800
Number of Stations	50
Number of Repeaters	3
TurnAround (ms)	1
TimeOut (ms)	5000

Figure 6.16: CSMA Simulation Parameters



(a) Waiting Time

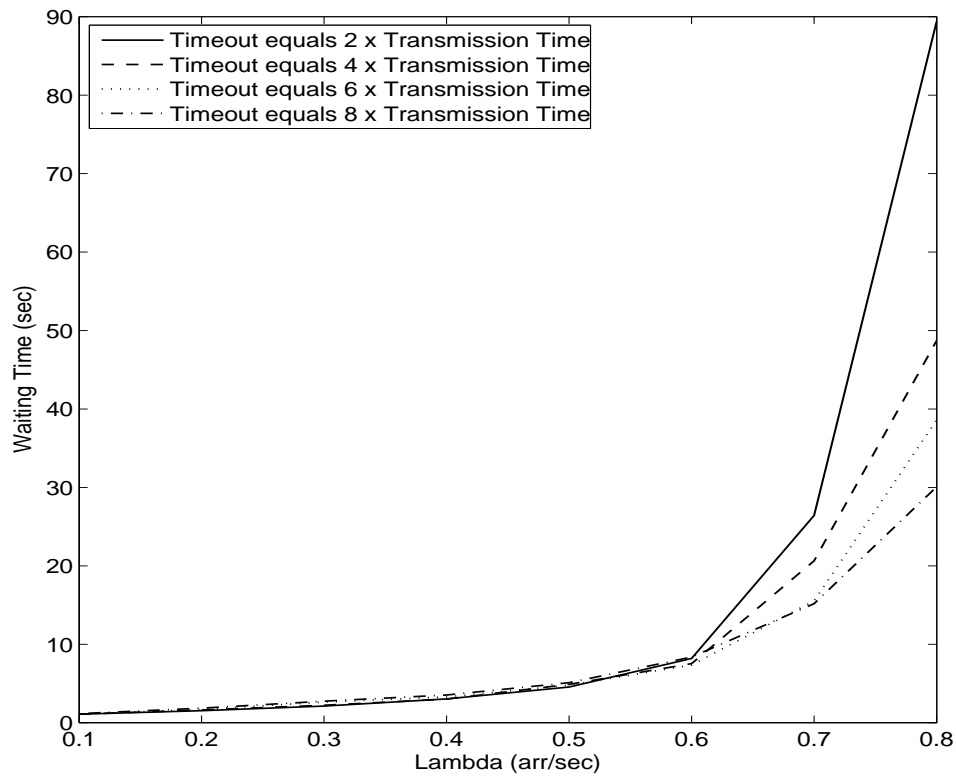


(b) Queue Length

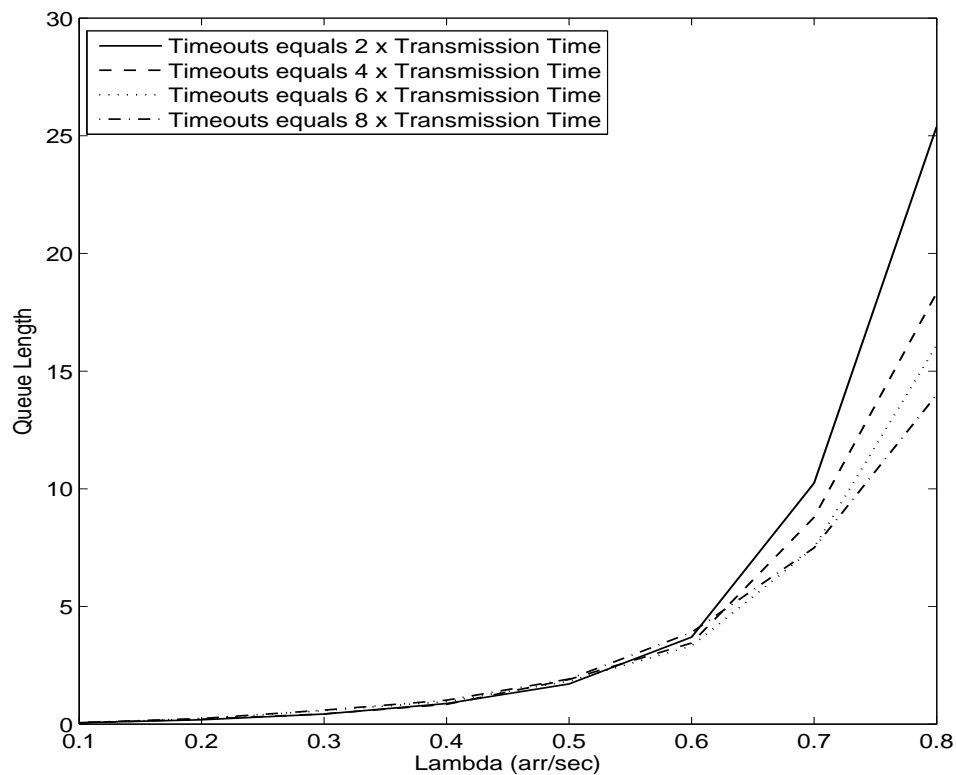
Figure 6.17: CSMA Simulation Results: Large Frame Sizes

The backoffs of Figure 6.17 have all been taken with an timeout period of 5000 ms. The variation in the timeout period with a backoff range of 1000 to 5000 ms is shown in Figure 6.18. The timeout periods have been chosen to be 2, 4, 6 and 8 times the total service time. Again at lower arrival rates, the lower timeouts give better results, but performance is reduced greatly at higher levels, where a pivotal change in performance occurs when the utilization becomes  $>1$  when

the arrival rate increases above 0.6 arrivals per second. It is also at this point that the timeout rate needs to be increased dramatically to reduce performance degradation.



(a) Waiting Time

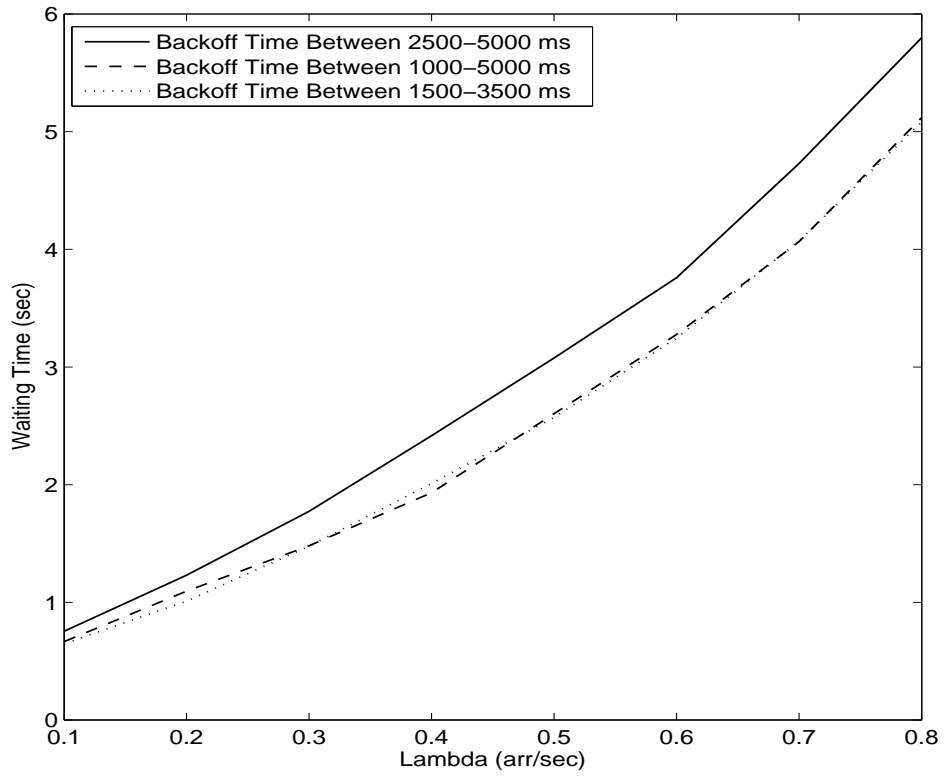


(b) Queue Length

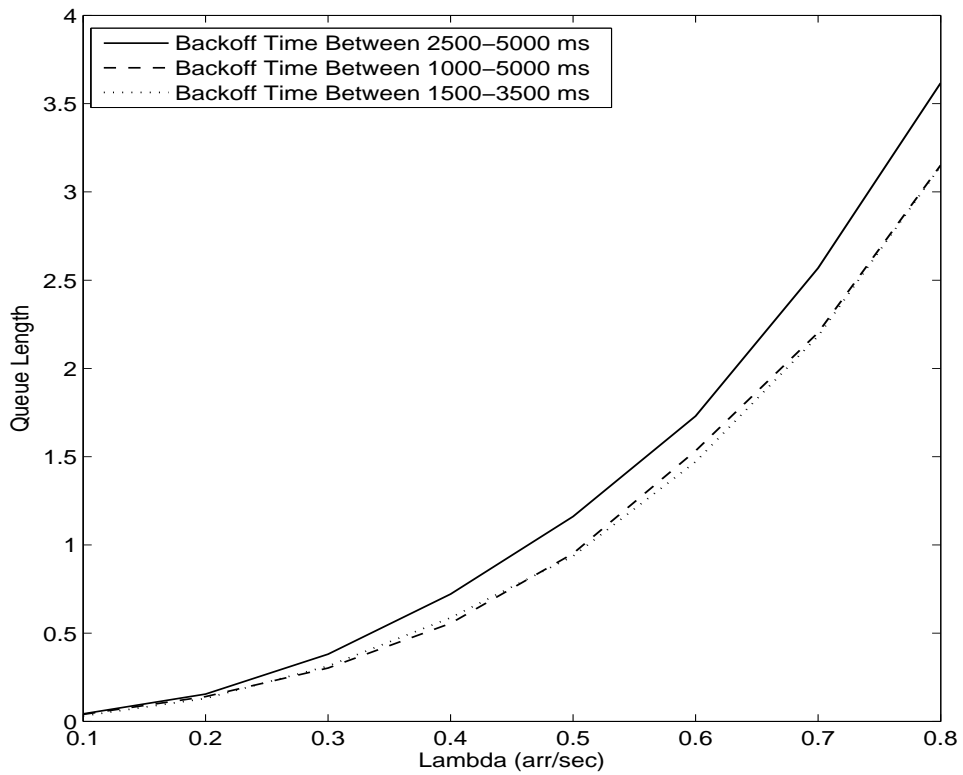
Figure 6.18: CSMA Simulation Results: Varying Timeouts for Large Frame Size

The same parameters used for the above two figures has been used for a smaller data and acknowledgement frame size

using a data bytes range of 27 to 67 and information bytes equal to 13. The results are shown in Figures 6.19 and 6.20. The inverse is true for the smaller frame size with respect to both backoff and timeout time. This is the norm and as such is confirmed by many reputed sources. The reasoning behind this is because channel utilization is considerable lower than with the larger frame sizes, meaning that every time a station backs off or is involved in a collision, the backoff and timeout period need not have to be that long, as the traffic over the communication channel is low. Furthermore, the utilization is never greater than one, and allows the normal assumption of a timeout period that is equal to 2 times the transmission time to hold true.

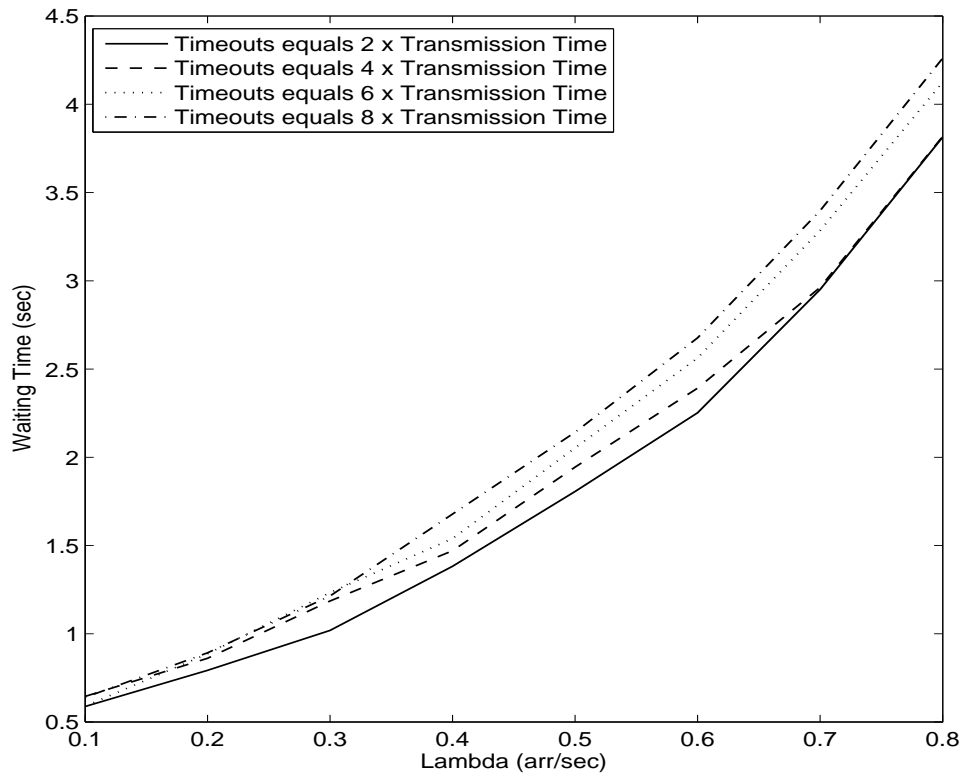


(a) Waiting Time

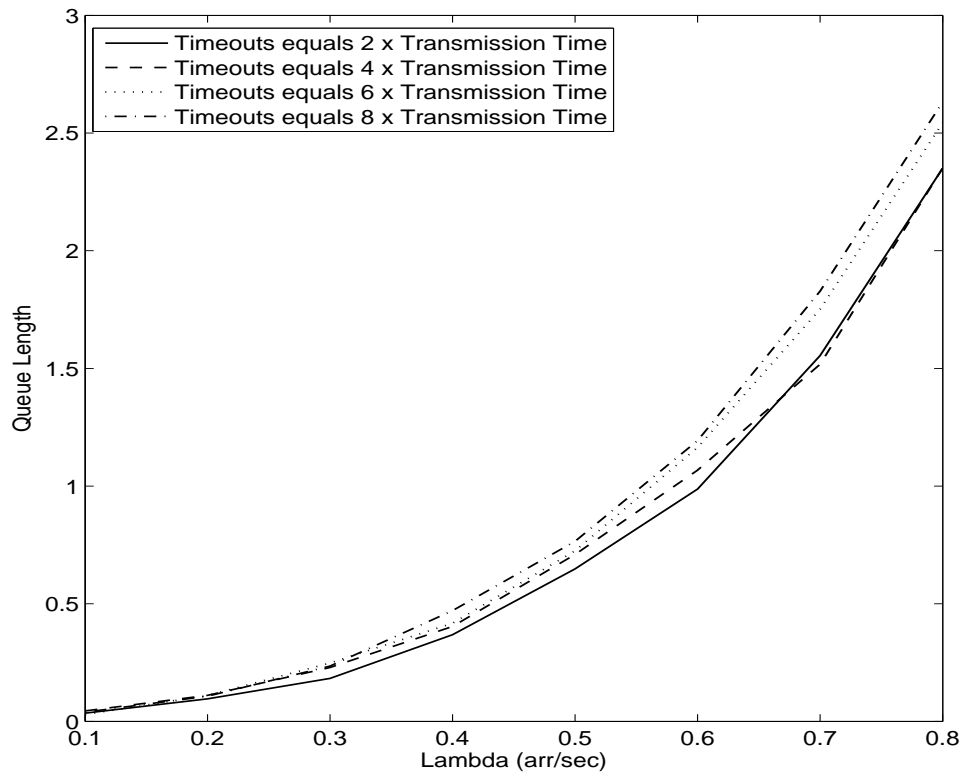


(b) Queue Length

Figure 6.19: CSMA Simulation Results: Small Frame Sizes



(a) Waiting Time



(b) Queue Length

Figure 6.20: CSMA Simulation Results: Varying Timeouts for Small Frame Size

Low and very high backoff rates cause long waiting times and long queue lengths. Figure 6.21 shows the effect of various backoff rates with different arrival rates for large and Figure 6.22 for small frame sizes (large frame sizes make use of timeouts of 5000 ms and small frame sizes of 2500 ms).



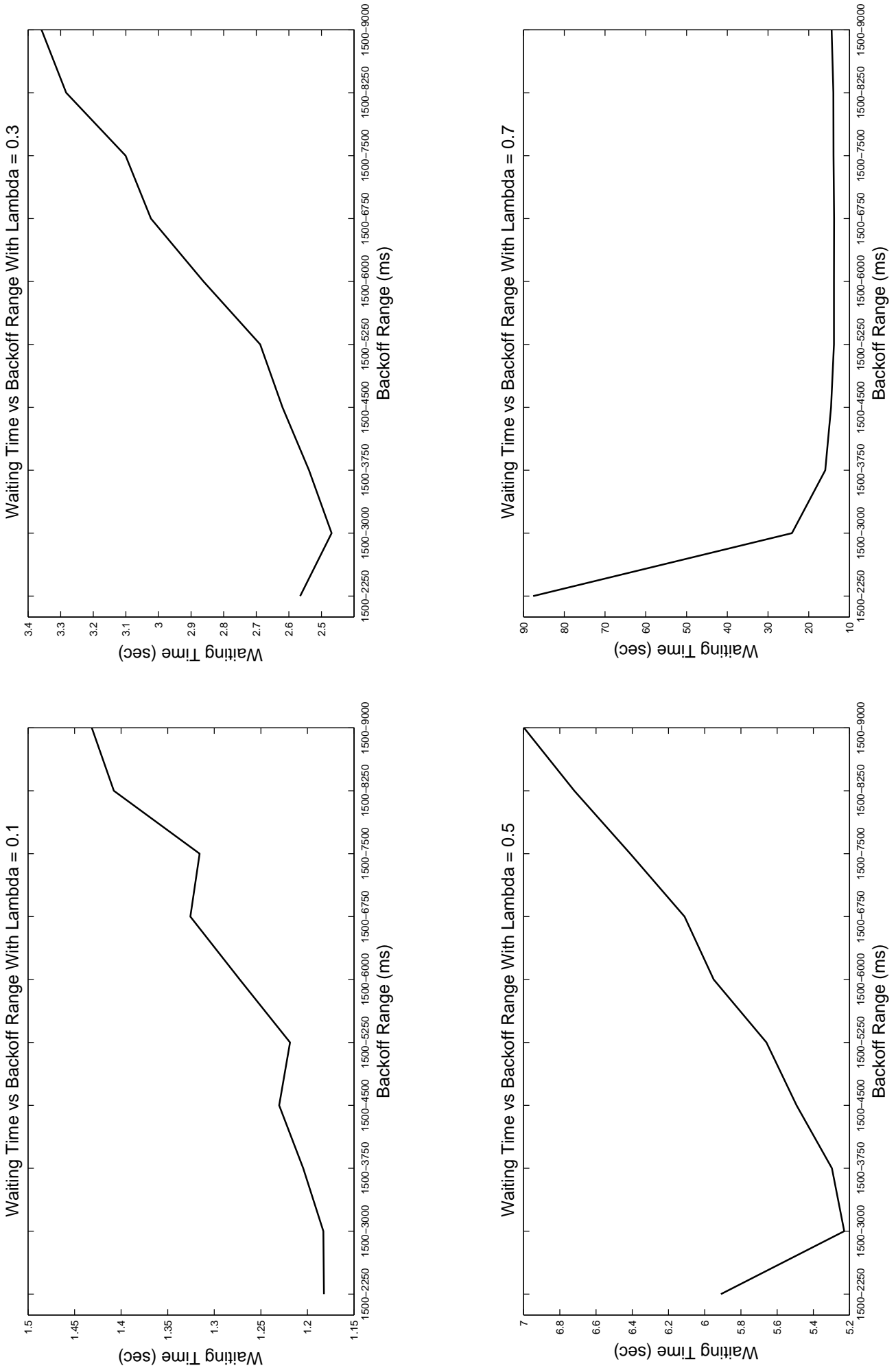


Figure 6.21: CSMA Simulation Results: Varying Backoff Ranges for Large Frame Size

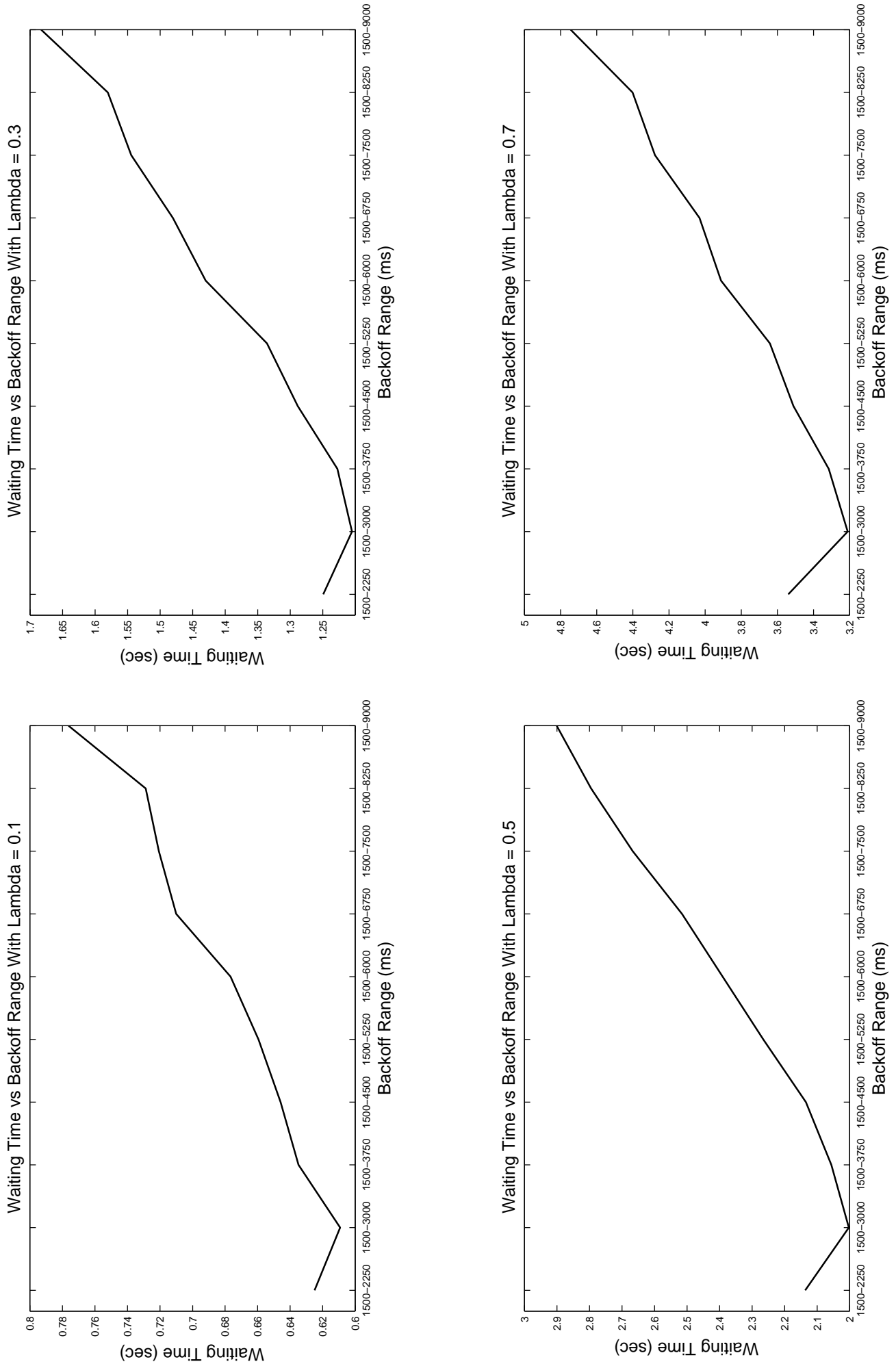


Figure 6.22: CSMA Simulation Results: Varying Backoff Ranges for Small Frame Size

Many CSMA protocols are optimized using exponential backoff instead of uniformly distributed backoffs. This has been implemented by increasing the backoff rate exponentially each time the station encounters a collision or backoff. The exponential increase can be user defined. Figure 6.23 shows the exponential backoff implemented for large frame sizes with a step size of 0.2, meaning that each time a station goes into backoff or collides with another station its backoff time will be incremented by  $e^{0.2k}$  where  $k$  represents the number of backoffs the station has encountered. The equation for the new backoff time is:

$$t_{backoff} = e^{0.2k} \times \text{Uniform Backoff Sample} \quad (6.30)$$

The uniform backoff sample represents the uniformly distributed random backoff sample that the station would have obtained and was defined between minimum and maximum boundaries by the user.

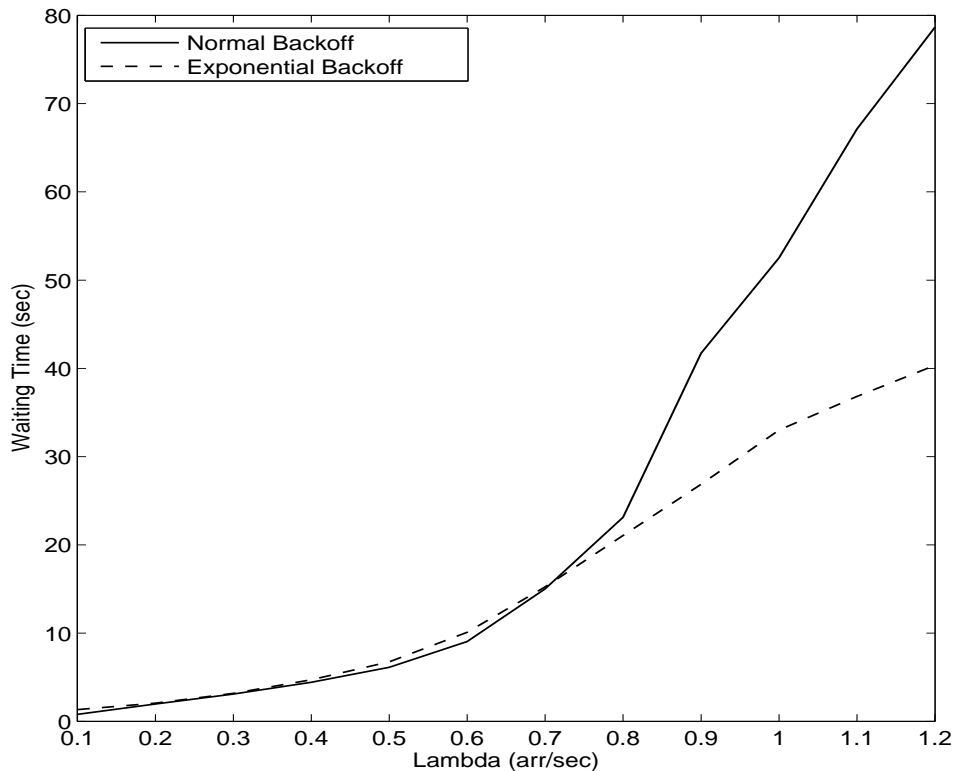


Figure 6.23: CSMA Simulation Results: Exponential vs Uniformly Distributed Backoff for Large Frames

The data latency plays a critical role in the performance of the protocol. The major cause of these latencies is delays in the equipment, especially repeater sites. All simulations and theoretical models throughout the thesis have been based on 3 repeaters unless otherwise stated. The performance is greatly improved with less data latency, thus less repeaters. Figure 6.24 shows the differences between three repeaters using the parameters for large frame sizes.

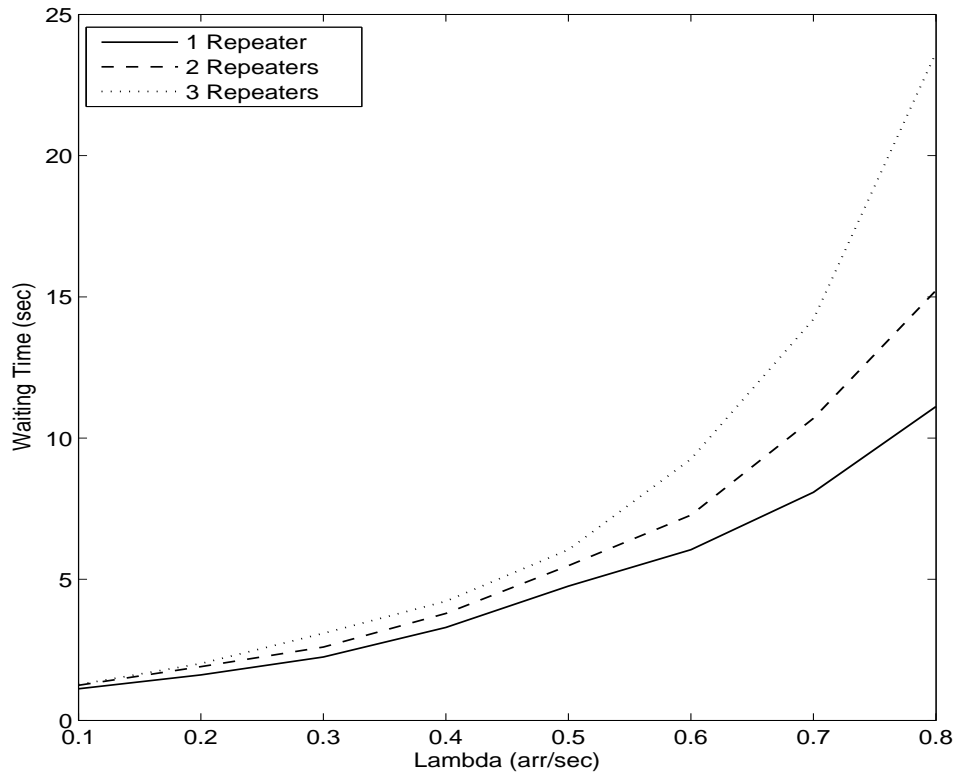
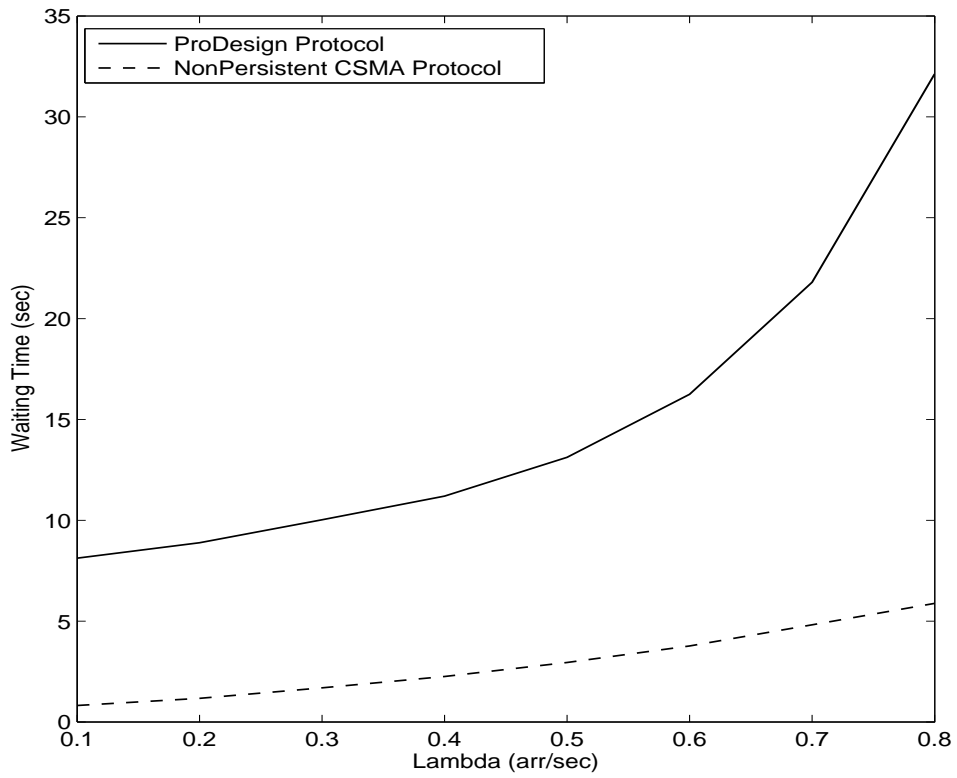
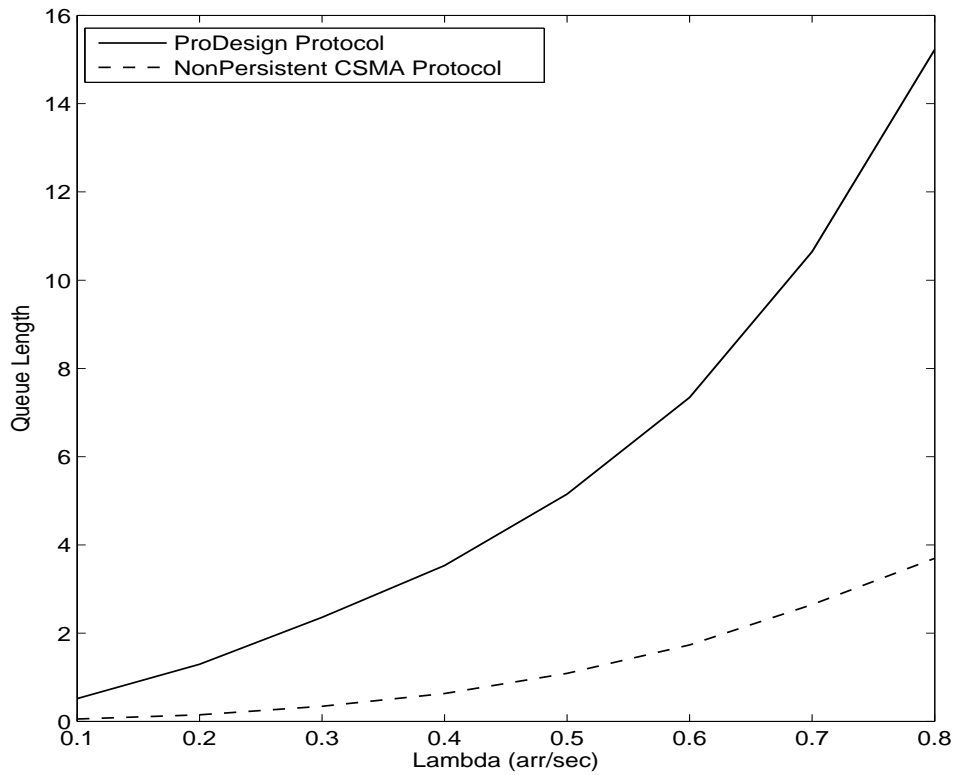


Figure 6.24: CSMA Simulation Results: Effect of the Number of Repeaters on Waiting Times

The last comparison to be presented in this section is of the non-persistent CSMA protocol and the ProDesign protocol (the protocol flowchart can be found in Appendix A). The ProDesign protocol makes use of an extra negotiating transmission, allowing for extra vulnerable time per station event transmission. This can clearly be seen in Figure 6.25. A small frame size has been used in this comparison.



(a) Waiting Time

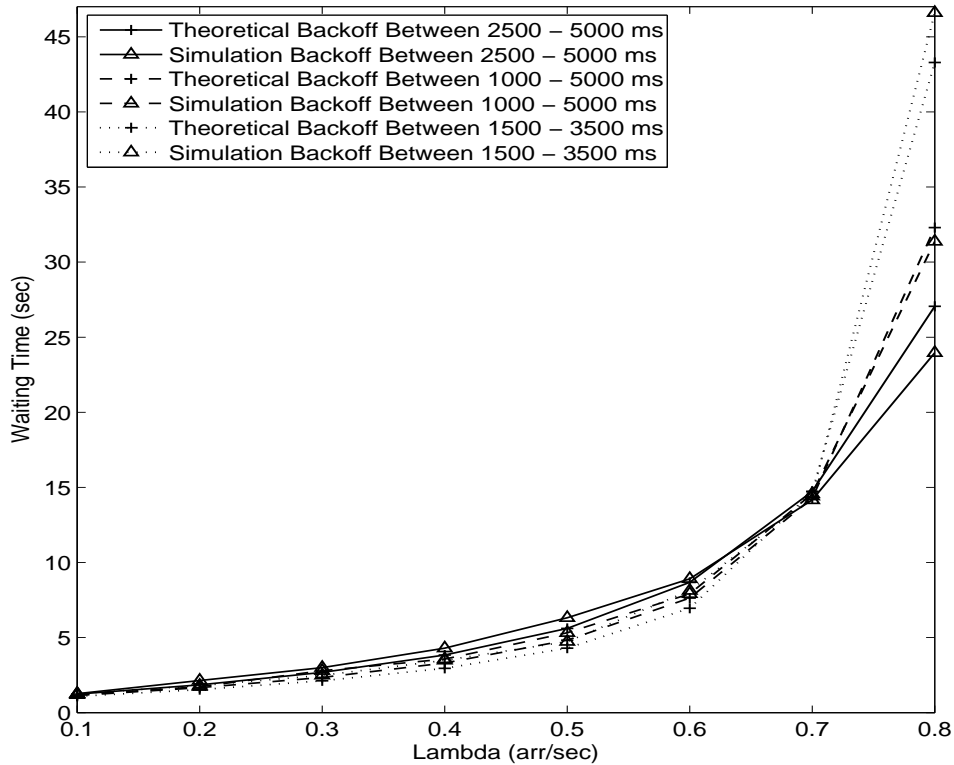


(b) Queue Length

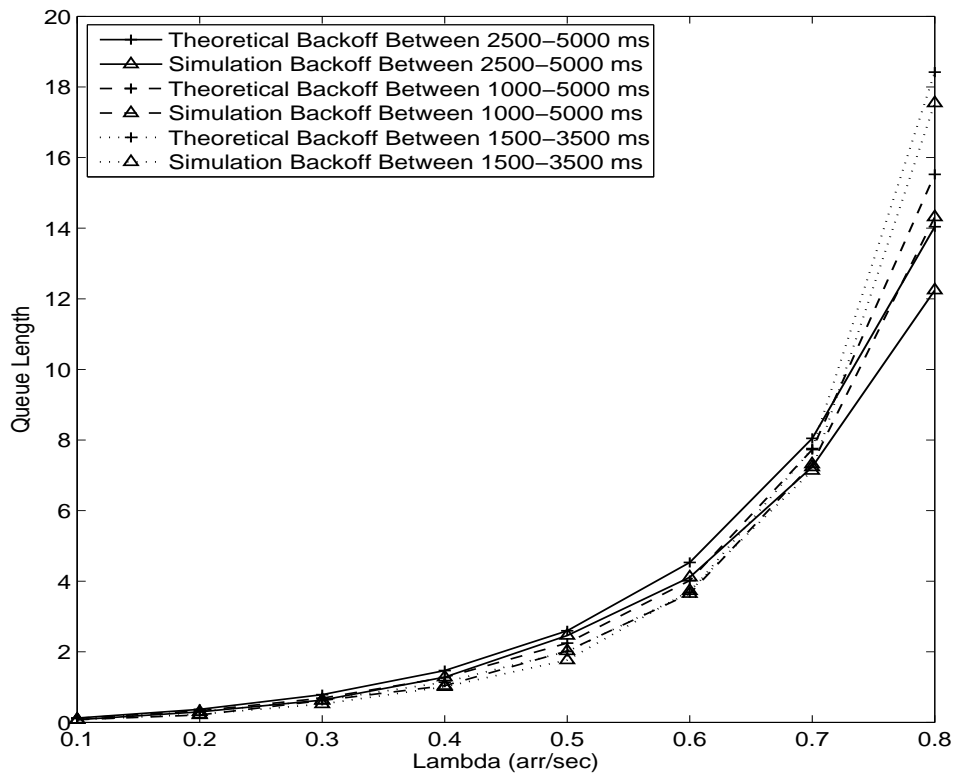
Figure 6.25: CSMA Simulation Results: ProDesign vs non-persistent CSMA

### 6.4.3 Comparison of Results

The comparison between the theoretical and simulation results is demonstrated in the following graphs. The same parameters as stated in the theoretical and simulation sections above have been used. It can clearly be seen from these results that the theoretical and simulation results follow closely within a bounded parameter set.

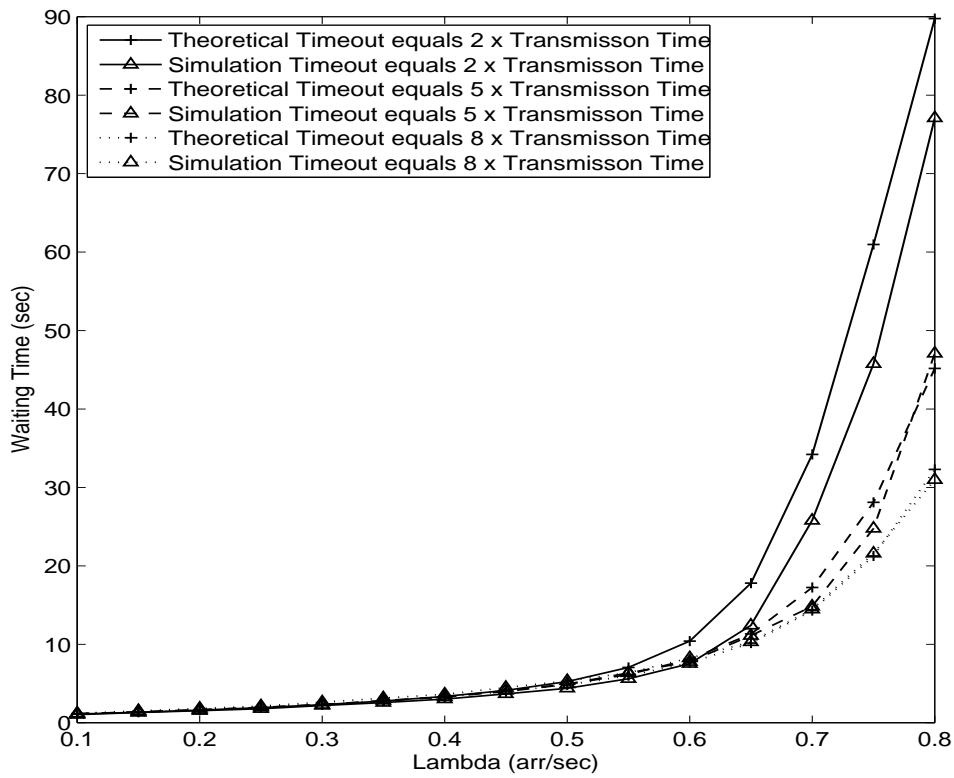


(a) Waiting Time

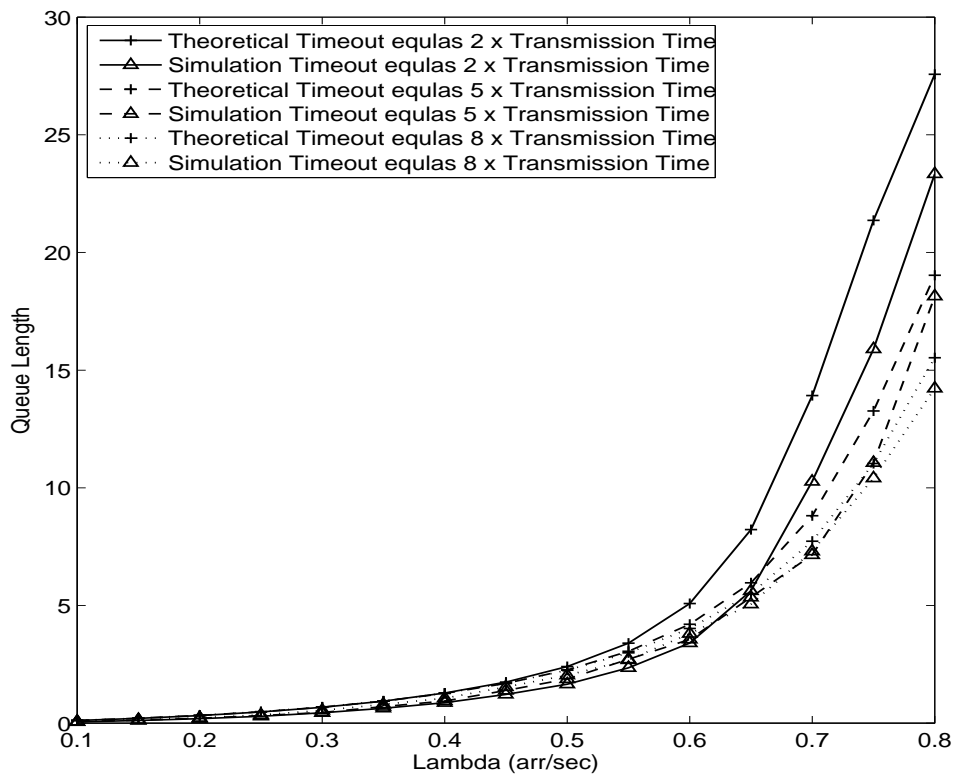


(b) Queue Length

Figure 6.26: CSMA Theoretical and Simulation Comparison: Large Frame Sizes

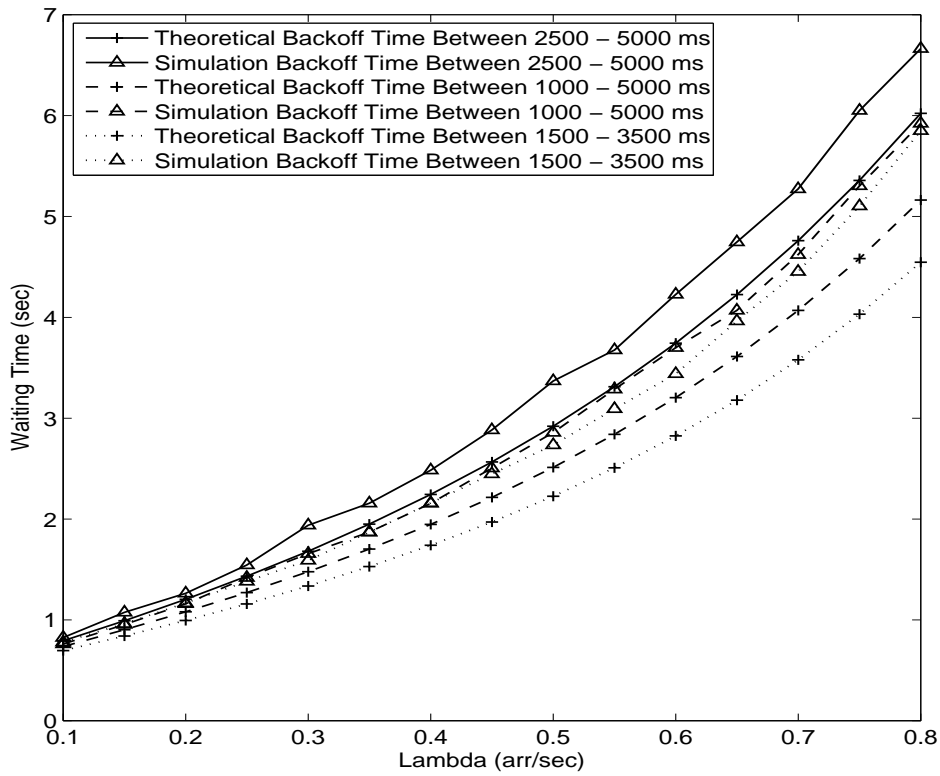


(a) Waiting Time

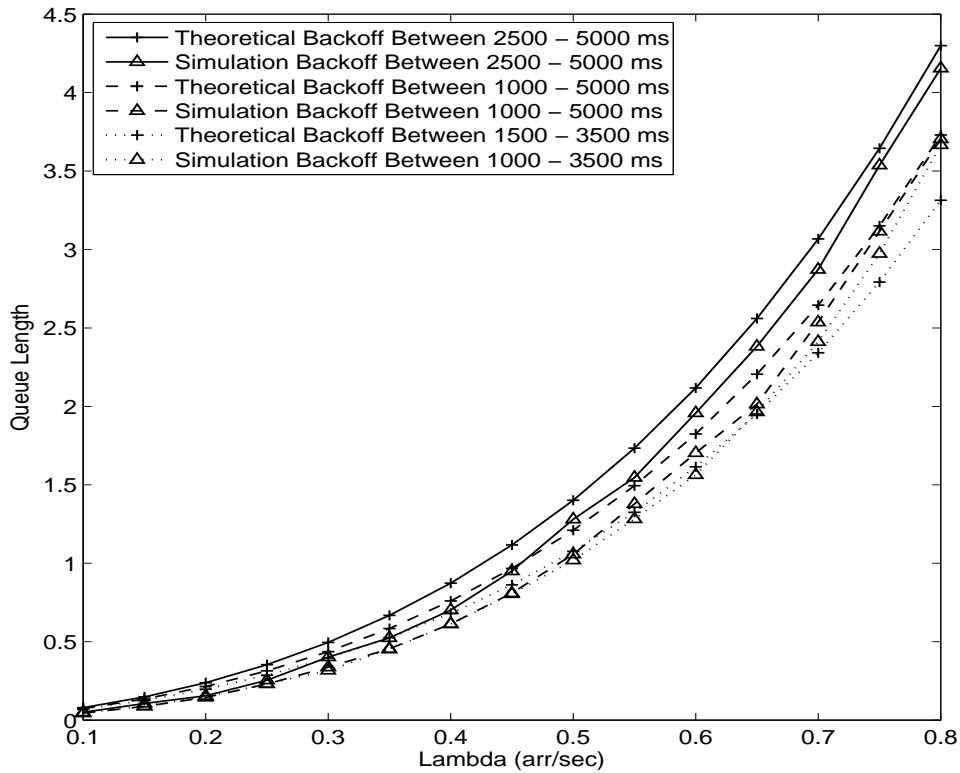


(b) Queue Length

Figure 6.27: CSMA Theoretical and Simulation Comparison: Varying Timeouts for Large Frame Sizes



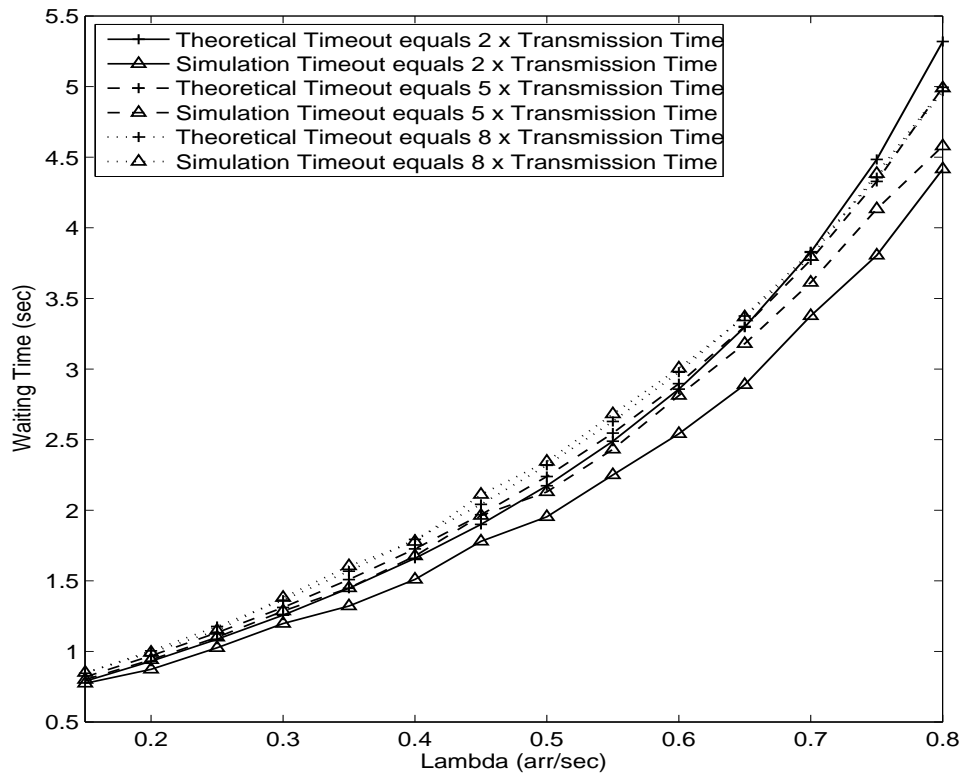
(a) Waiting Time



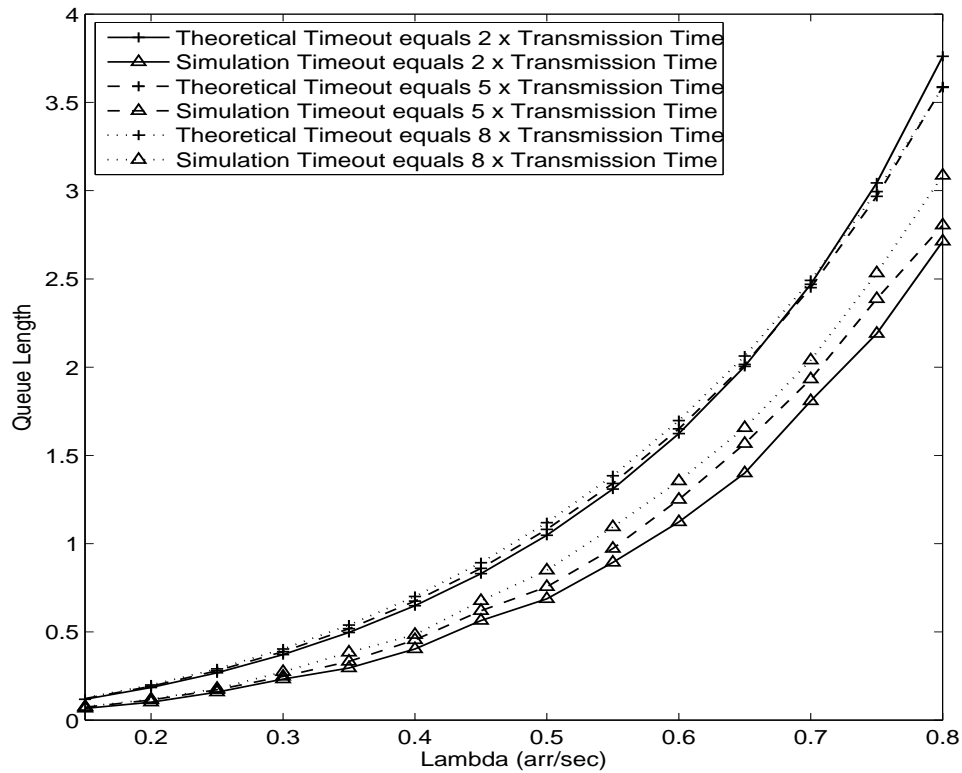
(b) Queue Length

Figure 6.28: CSMA Theoretical and Simulation Comparison: Small Frame Sizes





(a) Waiting Time



(b) Queue Length

Figure 6.29: CSMA Theoretical and Simulation Comparison: Varying Timeout for Small Frame Sizes

## 6.5 Summary

This chapter discussed and analysed the theoretical and simulation model for the non-persistent CSMA protocol. A detailed analysis of the transmission time of the frames used within the protocol has been given. Part of this transmission time includes a vulnerable period in which conclusions within the communication channel can take place. The layout of this vulnerable period is determined by the equipment and propagation latencies within the network and is a critical component within the protocol as results has shown. The State Space Model introduced in [40] has been expanded to provide better results for a broadened parameter set. Both models have been explained and the results for the expanded model shown. The simulation model layout makes use of three main discrete processes namely; the Generator, Server and Station processes. These processes have been explained with the help of UML based activity diagrams. The implementation of the simulation model within Java have also been described. The results have proven previous assumptions of good performance under low traffic demand and has also shown the quick degradation in performance with the increase in station arrivals within a given network.

Chapter 7 deliberates the implementation of the RRP protocol.

## Chapter 7

# Round Robin Polling: Modelling and Simulation

### 7.1 Introduction

The Round Robin Protocol is used extensively throughout telemetry systems and is sometimes used in conjunction with the non-persistent CSMA protocol. In this chapter the protocol structure, theoretical and simulation modelling of the Round Robin Protocol will be discussed.

For this investigation a basic continuous polling cycle with single buffered stations is assumed. Each station can only reply with one data frame when requested for information from the base station. If the base station timeout period lapses, it will continue to the next station in the network and will not retransmit the data request to the current station. The model and simulation is built upon the assumption that each station can only have one message queued at a time. It is therefore assumed, that after an event occurred at a station that no further event will occur before the remote station is polled by the server again or if it does, the event is entered into a sufficiently sized station event buffer. Furthermore, it is assumed that all stations within the network are stable operating within acceptable standards.

### 7.2 Protocol Structure

The base station will start communication with a remote station by sending a data request to the station, on which the station can reply with either the data in its buffer, or an acknowledgement frame indicating that it has no new data to relay. The data request frame from the server and the acknowledgement frame from the remote station, will have the same time requirements. The data, data request and acknowledgement frame transmission times are as follows:

$$t_{dqf} = t_{acf} = t_{pr} + t_{ps} + t_{ta} + t_{rd} + t_{dl} + t_{ib} \quad (7.1)$$

$$t_{dtf} = t_{pr} + t_{ps} + t_{ta} + t_{rd} + t_{dl} + t_{ib} + t_{db} \quad (7.2)$$

where:

$t_{dqf}$  Time required to send a data request frame from the base station (ms)

$t_{acf}$  Time required to send an acknowledgement frame from the remote station (ms)

$t_{dtf}$  Time required to send a data frame from the remote station (ms)

$t_{pr}$	Preamble (ms)
$t_{ps}$	Postamble (ms)
$t_{ta}$	Turnaround (ms)
$t_{rd}$	RXEnd (ms)
$t_{dl}$	Data Latency (ms)
$t_{ib}$	Information bits transmission time (ms)
$t_{db}$	Data bits transmission time (ms)

The description of the different time segments of the acknowledgement or data request frame times is in keeping with those described in Chapter 6. The protocol structure is shown in Figure 7.1. As can be seen from Figure 7.1, each frame has the probability of being influenced by burst noise and therefore not being received correctly by its end destination. When noise is incorporated, the frame sent by the remote or base station will be received corrupted. As discussed in Chapter 6 this is usually caused by high signal to noise ratio's (SNR), or low signal strength levels (RSSI) at the remote station. Again, the radio used is the MDS 4710E, which has a specified bit error rate of  $10^{-6}$  at a signal strength level of  $-110dBm$ . Due to the small influence that the error rate will have on the overall transmission, this isn't simulated on its own within the models but rather calculated as an extra time delay within each package as before. This time delay is calculated as follows:

$$t_{de} = \left( \frac{Data\ Bytes * 10}{10^6} \right) * t_{to} \quad (7.3)$$

$$t_{ie} = \left( \frac{Information\ Bytes * 10}{10^6} \right) * t_{to} \quad (7.4)$$

where:

$t_{de}$	Time contributed by data bytes being in error (ms)
$t_{ie}$	Time contributed by information bytes being in error (ms)

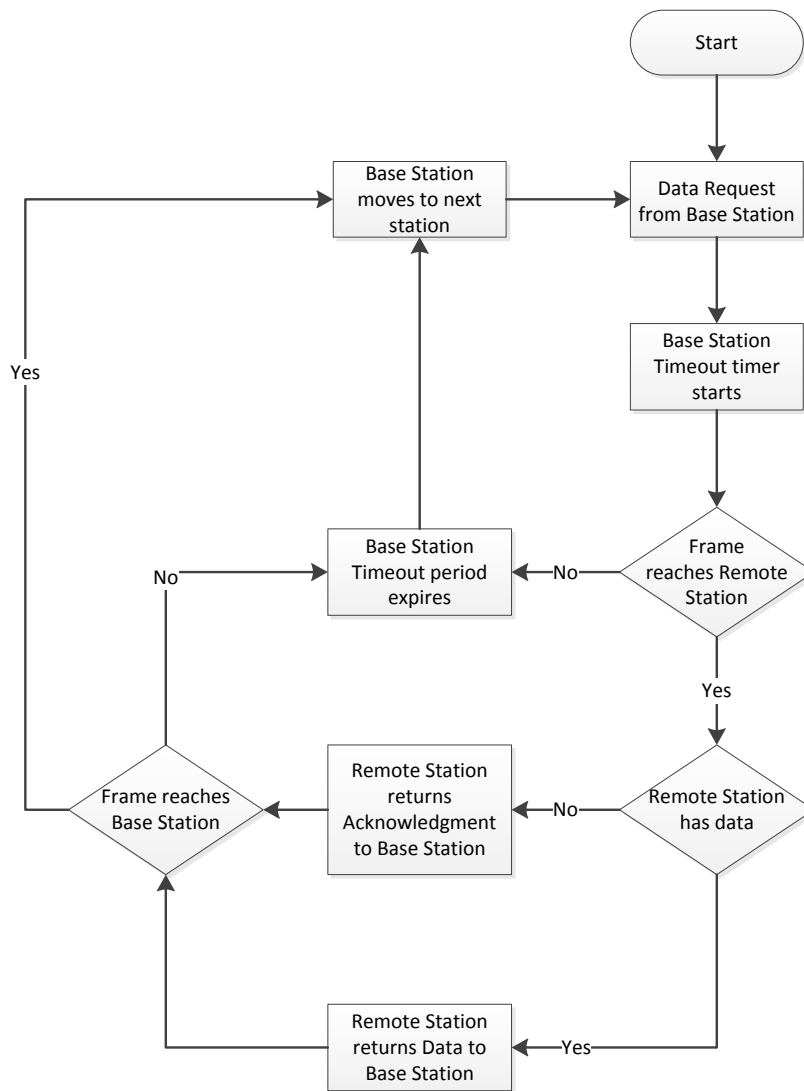


Figure 7.1: Round Robin Protocol

The timeout period  $t_{to}$  of the server must be chosen to be greater than the sum of the data request and data frame response times:

$$t_{to} > t_{dqf} + t_{dtf} \quad (7.5)$$

where the time of  $t_{dqf}$  and  $t_{dtf}$  must be calculated for the maximum number of bytes that can be sent during the length of the timeout period. The additional time added due to the probability of an error occurring depicts the added delay of a timeout occurring when a data request is sent to the remote station from the base station and a reply is not received back. The transmission time for each frame can now be calculated as follows:

$$t_{drq} = t_{dqf} + t_{ie} \quad (7.6)$$

$$t_{ack} = t_{acf} + t_{ie} \quad (7.7)$$

$$t_{dat} = t_{dtf} + t_{de} + t_{ie} \quad (7.8)$$

where:

$t_{drq}$	Data request transmission time (ms)
$t_{ack}$	Acknowledgement transmission time (ms)
$t_{dat}$	Data transmission time (ms)

The above information will be used in the setup of both the theoretical and simulation model as described in the following sections.

### 7.3 Theoretical Modelling

Figure 7.1 shows that there are two types of replies that the base station could receive from the remote station. The first involves the base station sending a data request to the remote station which replies with an acknowledgement frame, indicating that it doesn't have any new data to convey. Within this communication both the data request frame and acknowledgement frame will have a static frame size consisting of only information bytes. The time required for such a communication to complete is as follows:

$$t_{dra} = t_{drq} + t_{ack} \quad (7.9)$$

where:

$t_{dra}$	Transmission time for a data request with an acknowledgement response (ms)
-----------	--

The second type involves the base station sending a data request to the remote station which replies with its event data. The data request frame will again be static, but the data frame sent back from the remote station can vary depending on the size of the remote station (e.g. a borehole station will have less data than a booster station). The size of the data frame can, therefore, be seen as exponentially distributed, as the arrival of events at different stations are of different sizes and exponentially distributed. The size of the number of bytes conveyed within the data frame will be taken as the average of the Poisson process. The time required for this communication to complete is as follows:

$$t_{drd} = t_{drq} + t_{dat} \quad (7.10)$$

where:

$t_{drd}$	Transmission time for a data request with a data response (ms)
-----------	--

The arrival of events at different stations, as well as the size of the data frames to be sent are assumed to be exponentially distributed. Although the data frames are exponentially distributed, the service time is not and will have a general distribution. Queuing theory can be used to solve this queue, which is seen as a M/G/1/N queue with vacations [5]. In this thesis, a queuing approach has not been taken, but a mathematical analytical method is used.

If an event occurs at a remote station, the time that the station will wait before being requested to send its data depends on the specific place in the polling cycle that the server is in. If the event occurs directly after the server has requested information from the specific station then the minimum time that the station will wait before the server will reach it again is:

$$t_{min} = N_s \times t_{dra} \quad (7.11)$$

where:

- $t_{min}$  Minimum waiting time if an event occurred at a remote station just after being serviced (ms)  
 $N_s$  Number of stations in the network

$t_{min}$  assumes that no other station has any events pending and all of them reply with an acknowledgement to confirm that they have received the data request from the base station. It can also happen that an event could occur at a remote station just before a data request arrives from the server, meaning that its waiting time for the base station is zero. From this it can be seen that if only one event can occur (with exponential distribution) in a cycle time of the base station, then over time the average waiting time of the remote station will be:

$$t_{min-avg} = \frac{t_{cyc-min}}{2} \quad (7.12)$$

This can be seen as the baseline waiting time of the protocol and even with minimal traffic, the average waiting for a remote station will be  $\geq t_{min-avg}$ . As traffic picks up and more remote stations create events, the waiting time per station will increase, therefore increasing the utilization of the channel. The utilization of any communication channel is determined by the number of arrivals that occur per second and the service rate of these arrivals:

$$\rho = \frac{\lambda}{\mu} \quad (7.13)$$

where:

- $\rho$  Channel utilization  
 $\lambda$  Arrival rate (arrivals per second)  
 $\mu$  Service rate (arrivals serviced per second)

Each station in the network has an individual arrival rate and according to the properties of a Poisson process, these rates can be summed to give the total arrival rate. The channel utilization can therefore also be written as follows:

$$\rho = \frac{N_s \lambda}{\mu} \quad (7.14)$$

where  $\lambda$  now indicates the average arrival rate of events at each remote station. The average service rate is determined by the time required to service a remote station having an event:

$$\mu = \frac{1}{t_{drd}} \quad (7.15)$$

It is clear that if the channel utilization is  $\rho$ , then the channel is idle for  $(1 - \rho)$ , bounded by the fact that the average arrival rate must always be smaller than the average service rate;  $N_s \lambda < \mu$  for realistic results. Making use of the channel utilization, the average waiting time per station can be calculated as:

$$T_w = t_{drd} \rho + t_{dra} (1 - \rho) \quad (7.16)$$

$$T_{avg} = \frac{T_w}{2} \quad (7.17)$$

where:

- $T_w$  Maximum waiting time per station (sec)  
 $T_{avg}$  Average waiting time per station (sec)

from this the average queue length can be calculated by making use of Little's theorem:

$$\begin{aligned} N &= \lambda T \\ &= N_s \lambda T_{avg} \end{aligned} \quad (7.18)$$

However, it is assumed that each remote station can produce only one event before it has been serviced and only after that can it create another event. Whenever a station has received an event, its contribution to the total arrival rate must be deducted, therefore also changing the channel utilization. This means that throughout the cycle time of the base station, the arrival rate changes and therefore the channel utilization changes. To accommodate this fact, the waiting time must be calculated iteratively, taking into account the change in channel utilization due to the change in arrival rate throughout the base station service cycle. The waiting time is now calculated as follows:

$$\rho = \frac{k\lambda}{\mu} \quad (7.19)$$

$$T_w = \sum_{k=1}^N \left[ t_{drd} N_s \left( \frac{k\lambda}{\mu} \right) + t_{dra} N_s \left( 1 - \frac{k\lambda}{\mu} \right) \right] \quad (7.20)$$

$$= N_s \times \sum_{k=1}^N \left[ t_{drd} \left( \frac{k\lambda}{\mu} \right) + t_{dra} \left( 1 - \frac{k\lambda}{\mu} \right) \right] \quad (7.21)$$

$$T_{avg} = \frac{T_w}{2N_s} \quad (7.22)$$

Making use of Equation 7.21, the average queue length can be calculated using Little's Theorem iteratively:

$$N = \frac{\sum_{k=1}^N \left[ (k\lambda) \left( t_{drd} N_s \left( \frac{k\lambda}{\mu} \right) + t_{dra} N_s \left( 1 - \frac{k\lambda}{\mu} \right) \right) \right]}{\sum_{k=1}^N k} \quad (7.23)$$

The above equations for the queue length and waiting for the specific RRP protocol have been implemented in Matlab and the results compared to that of the simulation. These are discussed in Section 7.5 of this chapter.

## 7.4 Simulation Modelling

As discussed in Chapter 3, DESMO-J has been used to implement the simulation of the RRP protocol as discussed in the previous section. A process oriented modelling approach has been taken to implement the protocol. The protocol has been implemented as outlined in Figure 7.1 with the same assumptions made as in Chapter 5.

### 7.4.1 Processes-Orientated Modelling

The RRP protocol is broken down into three main processes:

- The Generator process which creates new arrivals
- The Station process which covers the addition of stations with events to the polling cycle
- The Server process which constitutes the service and acknowledgement of events from different stations, as well as the request for data and acknowledgement from stations with no data readily available.

These processes are described in the following sections:



### 7.4.1.1 The Generator Process

As in the non-persistent CSMA protocol, the Generator process is used to create new stations, simulating events arriving at stations. As the station ID is critical in the implementation of the protocol, the same approach cannot be taken and a Java ArrayList is implemented, which is a variable array which holds Station process objects. The ArrayList is known as the Station Events Queue. The implementation of the Station Events Queue will only be seen in the station process, but is mentioned for better understanding of the Generator process life cycle as per Figure 7.2.

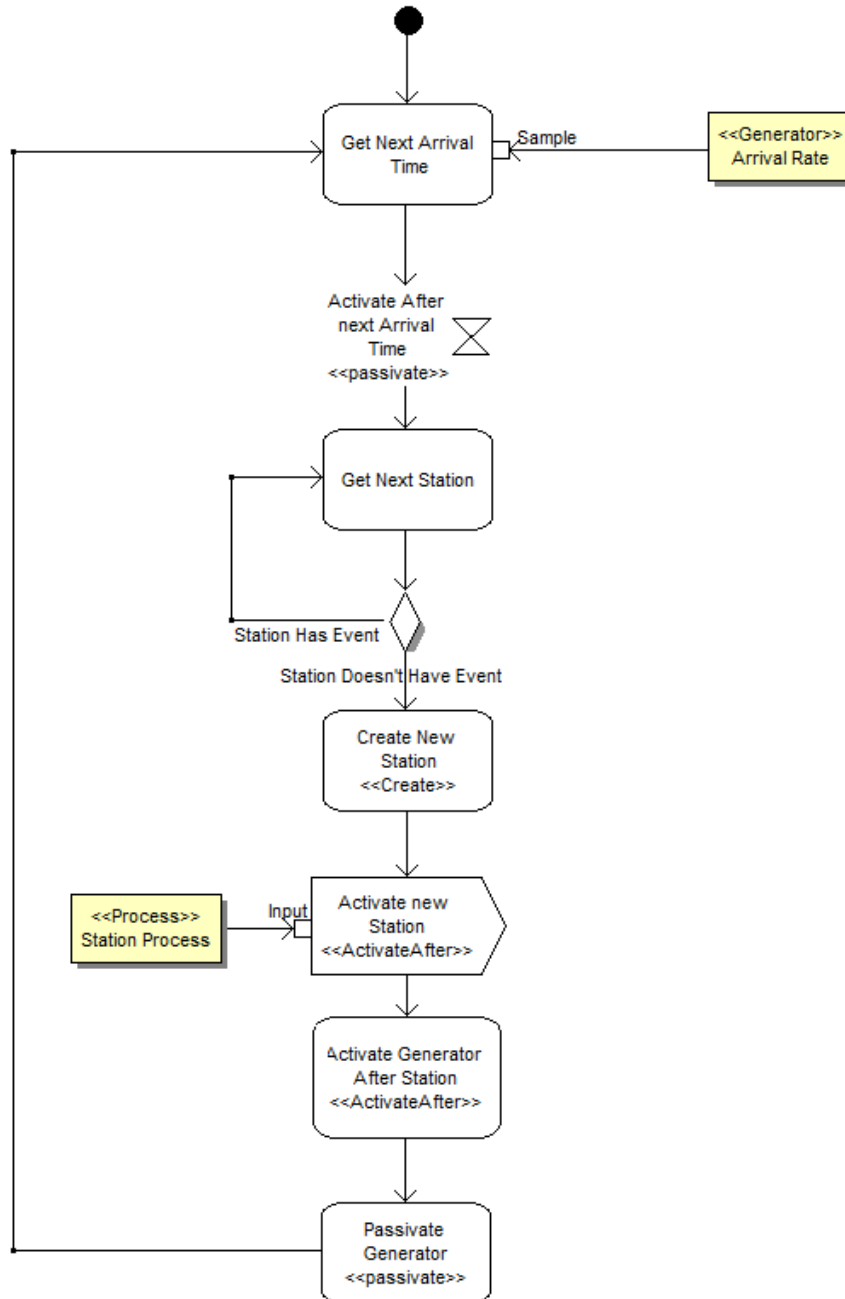


Figure 7.2: RRP Generator Process Activity Diagram

The process's life cycle starts by obtaining the arrival time of the next event to occur within the network. The Arrival Rate Generator creates exponentially distributed times for new events, making use of the average arrival time and Java's uniform random generator. The Generator process is scheduled to re-activate when this time has passed and then entered

into the event list and passivated. Upon re-activation a random uniform generator is applied to get the next station ID. The Station Events Queue (described previously in this section) contains Station process objects. These objects describe stations that have events assigned to a specific station ID. If the generator produces a station ID of a station that already has an event, a new ID is randomly generated, until a station which does not have an event waiting, is identified. A new station instance is then created, entered into the event list and scheduled by the scheduler to be activated next. The Generator process is then scheduled to re-activate after the newly assigned Station process is entered into the event list and passivated. Upon re-activation the above process is repeated.

#### **7.4.1.2 The Station Process**

The Station process's life cycle activity diagram is shown in Figure 7.3. The Station process life cycle begins by getting the current time, which is used to determine the waiting time of the station and finally the average waiting time of the protocol. As discussed earlier the Station Events Queue is used to store a list of stations with an event to be transmitted. Therefore, the station is added to the Stations Event Queue according to its ID. The name Stations Event Queue can be misleading as it is actually an array, where each station's ID equals the index of the array. The Station process has produced an event and due to the assumption made that a station can only produce one event at a time, the arrival rate must be reduced by the arrival rate of one station. The Arrival Rate Generator is therefore set accordingly and the Station process passivates itself and can only be activated after the server has completed its service of the station.

When activated, the current time is again obtained. The start time, obtained at the start of the Station processes life cycle, is subtracted from this time and the resultant waiting time of the station entered in the Waiting Time statistical counter, which is of type Tally (described in Chapter 6). The station is then removed from the Station Event Queue and the Arrival Rate Generator is updated with the new arrival rate (one less station has an event, therefore increasing the average arrival rate). The Station process life cycle is completed and garbage collected by DESMO-J and Java.

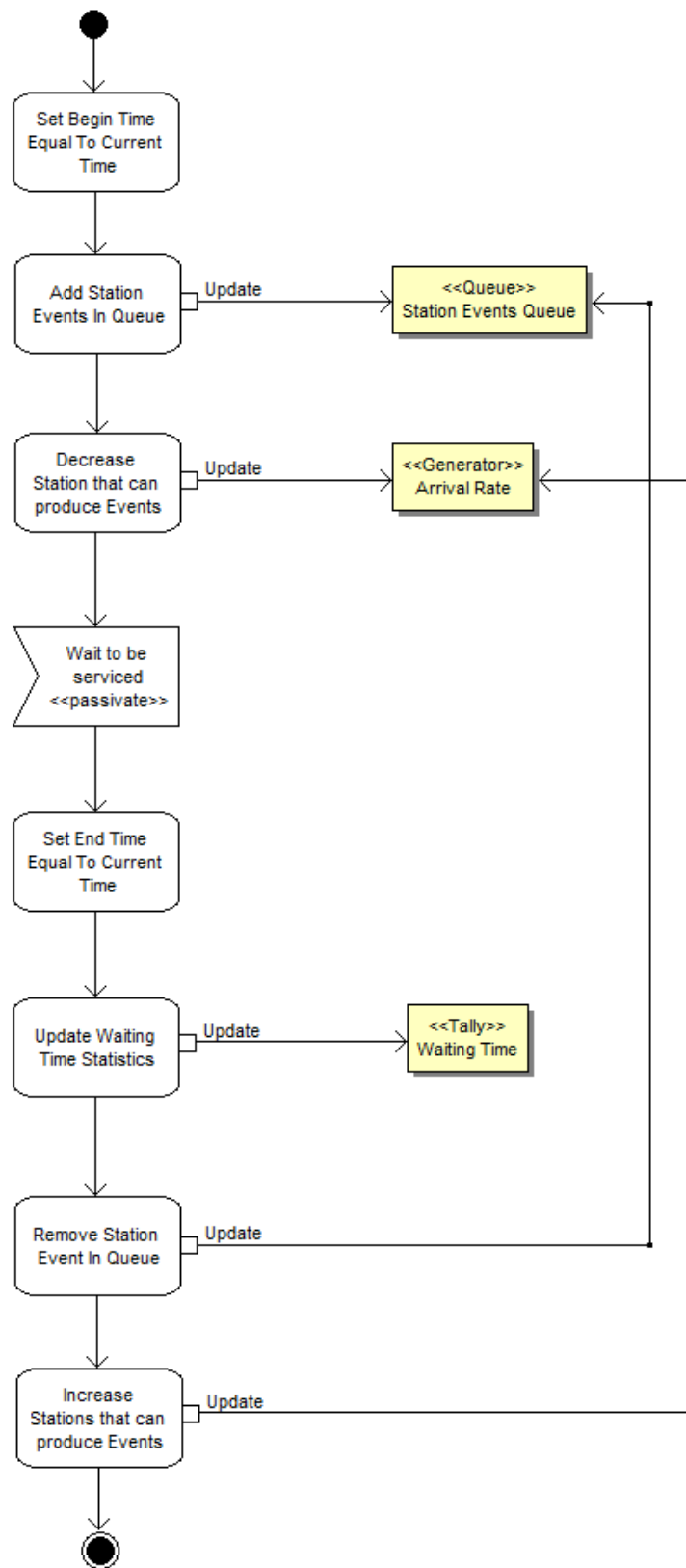


Figure 7.3: RRP Station Process Activity Diagram

### 7.4.1.3 The Server Process

The Server process life cycle runs through the polling cycle of the protocol. The Data request, Acknowledgement and Data Transmission times are calculated in this process in accordance with the user defined values set at the start of the simulation. The Server process's life cycle activity diagram is shown in Figure 7.4. At the beginning of the Server process life cycle, the current time is obtained to determine the time required to complete one polling cycle. The process will look at the first position in the Station Event Queue and determine if it contains a Station process object, meaning that the specific station has an event to be serviced. If this is the case the station event will be serviced.

The data request and service time is obtained and the Server is scheduled to re-activate after this time has passed. It is entered into the event list and passivated. Upon re-activation, the Station process, obtained from the Station Event Queue, is scheduled to be activated and the Server scheduled to be re-activated after the station. Both are entered into the event list accordingly and the Server process passivated. The cycle counter is incremented by one and will be used to determine the average queue length of the protocol. If the end of the cycle has not yet been reached, the last station in the Station Event Queue has not yet been evaluated and the process repeated.

If the position of the evaluated Station Event Queue doesn't contain an instance of a Station process, the specific station represented by the position of the Station Event Queue does not have an event at this stage. The server will then obtain the data request and acknowledge transmission time. After which it will schedule itself for re-activation, be entered into the event list and passivated. Upon re-activation the process will again check if it is at the end of its cycle. If this is the case, the cycle counter will be entered into the Queue Length statistical counter which is used to determine the average queue length. After this the current cycle time will be obtained to determine the length of the polling cycle and finally the average cycle time of the protocol. The cycle counter is reset and the Station Event Queue is again evaluated from the beginning.

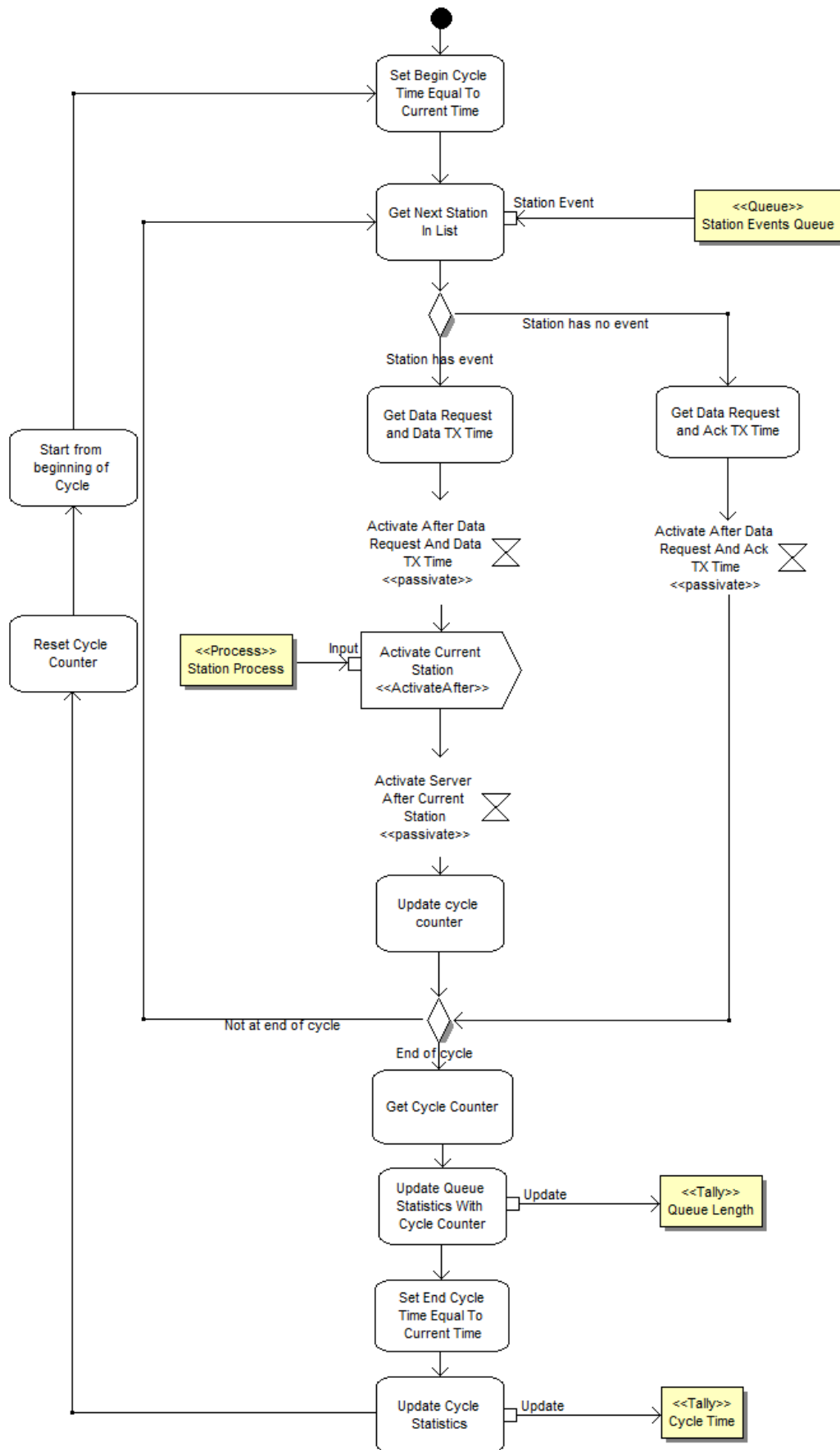


Figure 7.4: RRP Server Process Activity Diagram

## 7.4.2 Simulation Model Configuration

The RRP simulation has been set up using the same DESMO-J facilities as outlined for CSMA. Various simulations were run, using increasing average arrival rates to determine the effect on the waiting time and queue length of the system.

## 7.5 Results

The results of the theoretical and simulation models will subsequently be compared and analysed in this section.

### 7.5.1 Theoretical Results

Theoretical results for large frame sizes are presented in Figure 7.6, based on the parameters of 7.5.

Number of stations [1,250]	50
Min Propagation distance [Km]	30
Max Propagation distance [Km]	80
Preamble [ms]	1
Postamble [ms]	1
Data latency [ms]	11
Frame Length data [bytes]	180
Frame Length info [bytes]	20
Channel Capacity [bps]	4800
RXend [ms]	30
Step Size	0.1
Min Arrival Time(arr/sec)	0.1
Max Arrival Time(arr/sec)	0.9
Time Out (ms)	5000
Number of Repeaters	3
Turn Around Time (ms)	1

Figure 7.5: RRP Theoretical Parameters

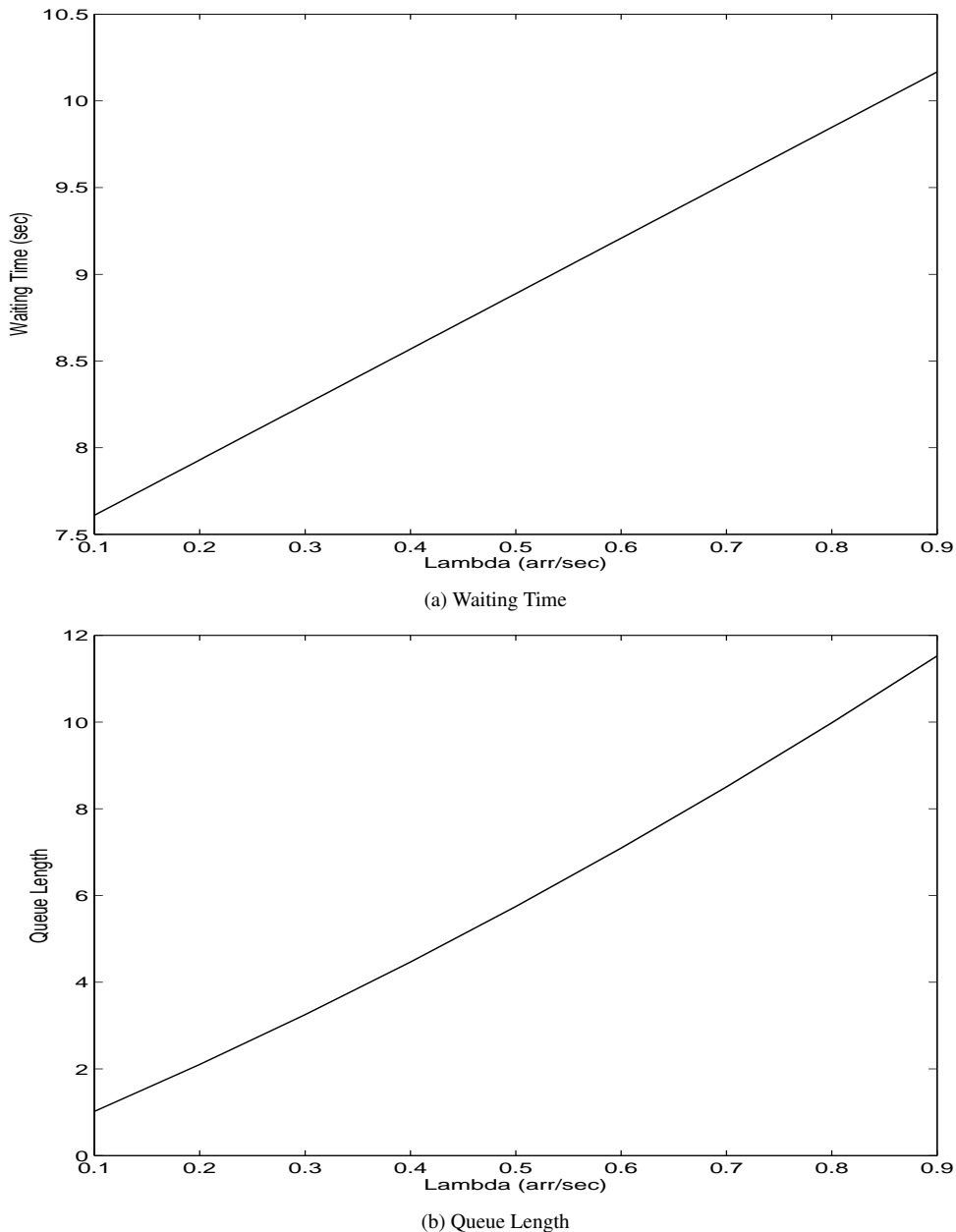
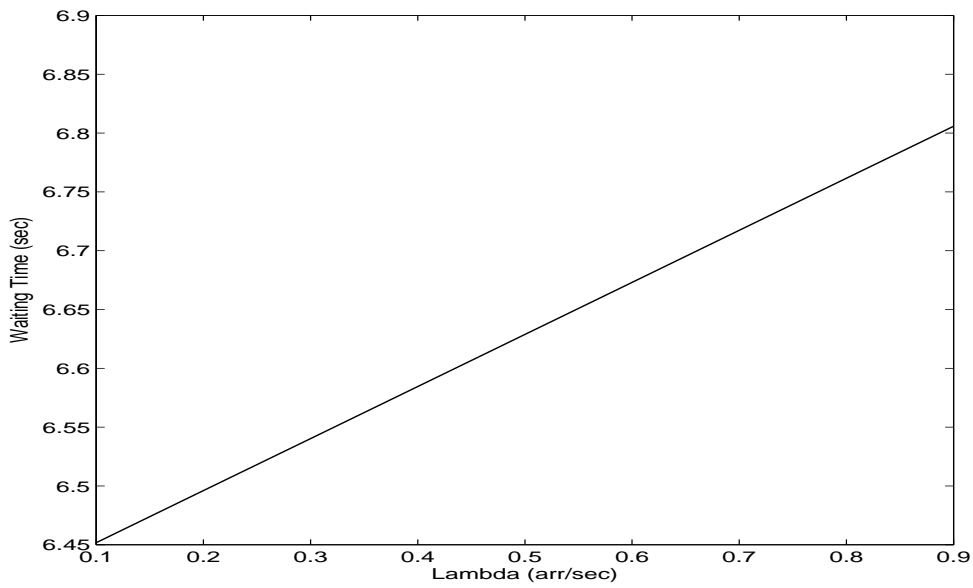


Figure 7.6: RRP Theoretical Results: Large Frame Sizes

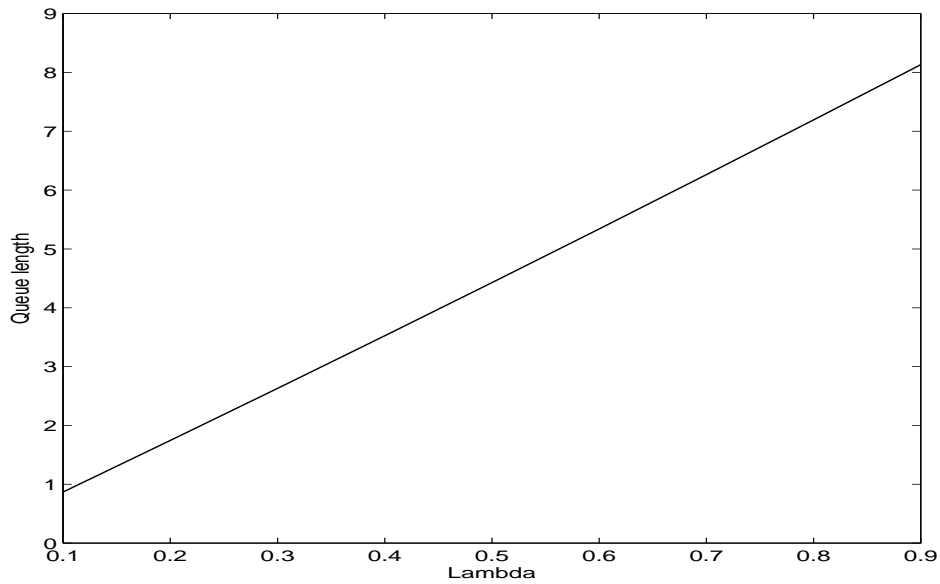
From the waiting time of low arrival rates shown in Figure 7.6a, the polling cycle overhead can clearly be seen due to the fact that the queue length is very low as is shown in Figure 7.6b. With an increase in arrival rate, the queue length and waiting time also increase. It can be seen that the increase in the queue length and waiting time shown in Figure 7.6 do not correlate if the time to transmit a data frame is taken into account. This can easily be explained by the fact that the transmission of a data frame replaces the transmission of an acknowledgement frame from each station with an event. The difference in the two frames' transmission time is added to the minimum polling cycle (polling cycle with no station events) for each station with an event to determine the waiting time. The protocol therefore performs better as the load is increased.

The waiting time and queue length, shown in Figure 7.7 uses the same parameters as used in Figure 7.6 with the exception of the data and information byte parameters. These parameters are reduced to the same values as expected in the practical example of the Namib water supply scheme, as mentioned earlier. Therefore, the data bytes reduce to an average of 47 and the information bytes to 13. The difference between the large and smaller frame sizes can clearly be seen in the reduced waiting time. This is because the time to transmit the small data frame has decreased significantly from the time required

to transmit a large data frame.



(a) Waiting Time



(b) Queue Length

Figure 7.7: RRP Theoretical Results: Small Frame Sizes

## 7.5.2 Simulation Results

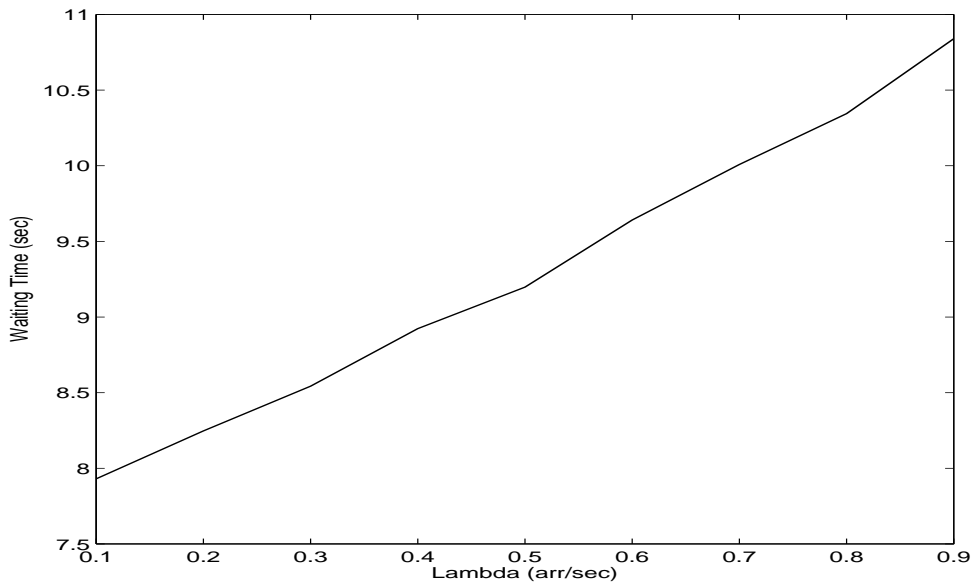
The same parameters as provided in Figure 7.5 will be used for the simulations. The only difference is that the data bytes and propagation distance are specified between a minimum and maximum value. These values as set up in the simulation GUI are shown in Figure 7.8.



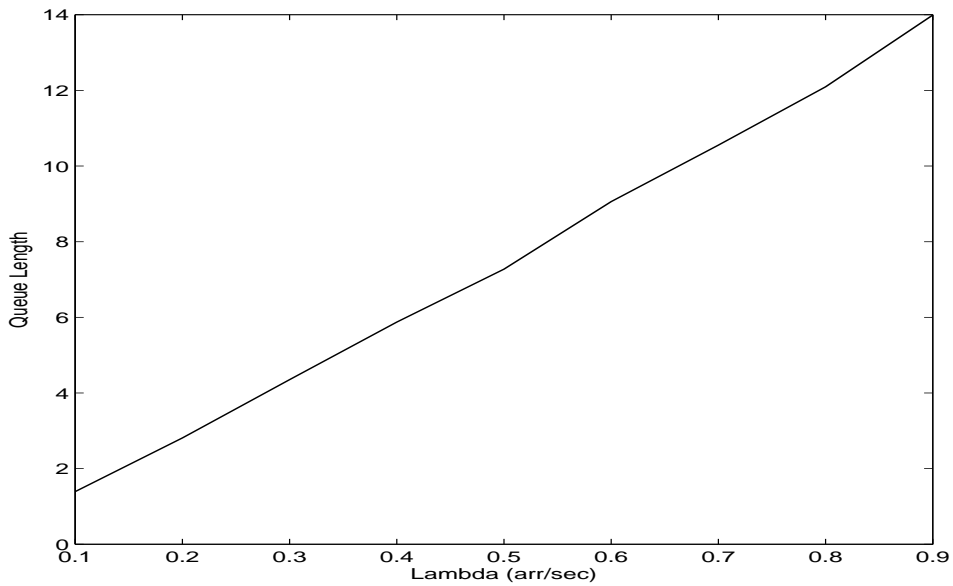
Number of Stations	<input type="text" value="50"/>	TimeOut (ms)	<input type="text" value="5000"/>
Channel Capacity (bps)	<input type="text" value="4800"/>	RXEnd (ms)	<input type="text" value="30"/>
Data Bytes Minimum	<input type="text" value="170"/>	Data Latency (ms)	<input type="text" value="11"/>
Data Bytes Maximum	<input type="text" value="190"/>	System Begin Arrival Rate (arv/sec)	<input type="text" value="0.1"/>
Information Bytes	<input type="text" value="20"/>	System End Arrival Reate (arv/sec)	<input type="text" value="0.9"/>
Preamble (ms)	<input type="text" value="1"/>	Simulation Length (sec)	<input type="text" value="25000"/>
Postamble (ms)	<input type="text" value="1"/>	Step Size	<input type="text" value="0.1"/>
Upper Distance	<input type="text" value="80"/>	Number of Repeaters	<input type="text" value="3"/>
Lower Distance	<input type="text" value="30"/>	Turn Around (ms)	<input type="text" value="1"/>

Figure 7.8: RRP Simulation Parameters

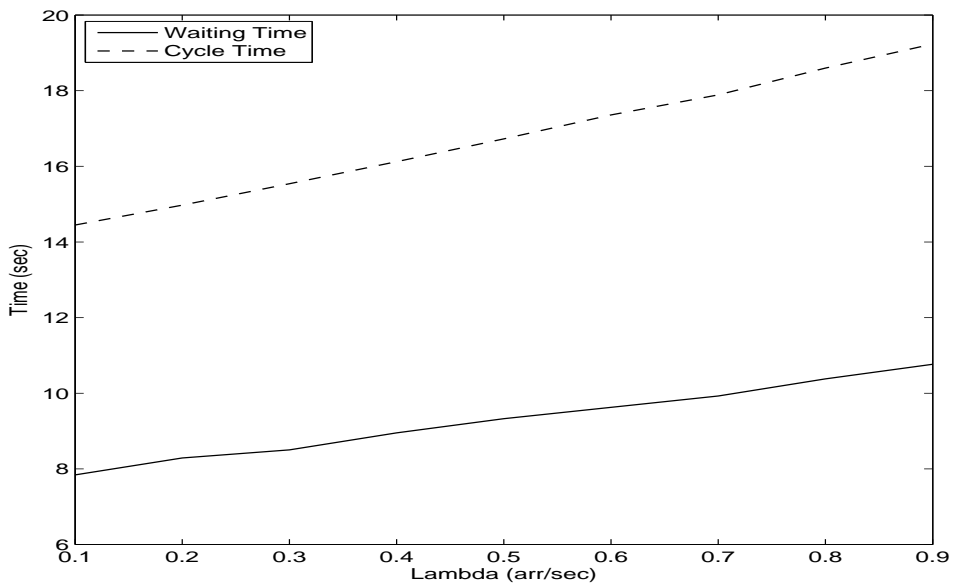
In the simulation the data frame sizes and arrival rates are exponentially distributed and the propagation distance uniformly distributed. The simulations are run with a simulation time of 25000 seconds and a step size of 0.1 between the minimum and the maximum arrival rates. The resultant waiting time and queue length is shown in Figure 7.9. From the figure it can be seen that the simulation has the same tendencies as the model, although more pessimistic. It can also be seen that with a transient simulation time of 600 seconds, steady state is reached.



(a) Waiting Time



(b) Queue Length

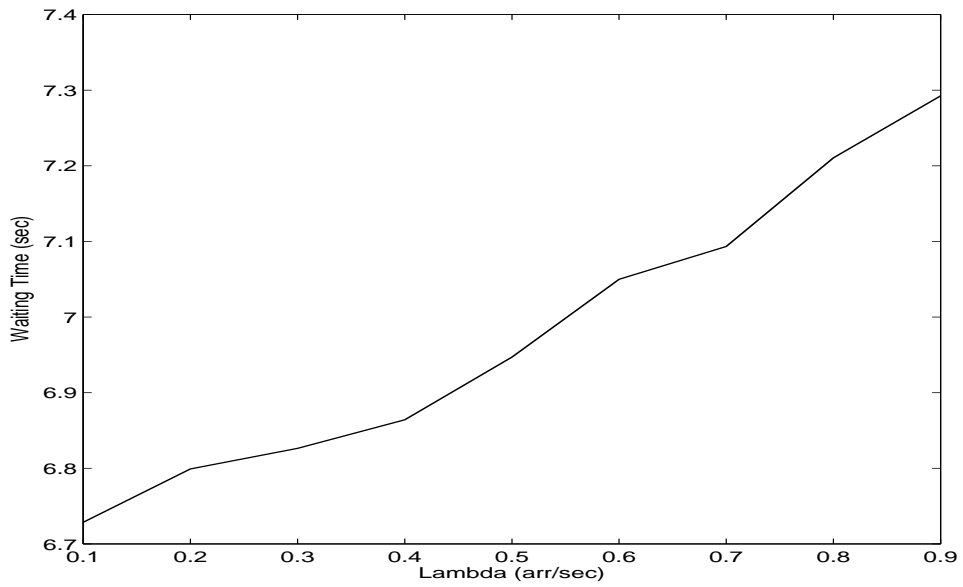


(c) Cycle Time

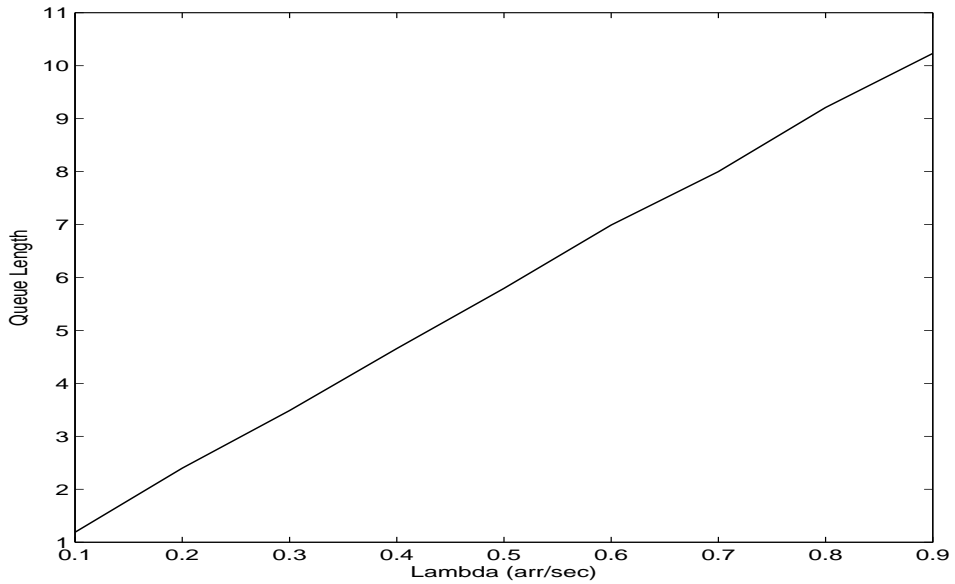
Figure 7.9: RRP Simulation Results: Large Frame Sizes

Figure 7.9c shows the polling cycle time average, compared to the average waiting time of the simulation. It could be expected that the average cycle time should be twice the average waiting time, but this is not quite the case. This can be explained by taking into account that the waiting time is based on the average waiting time of each station, while the average cycle time also takes into account polling cycles where there are no events, resulting in the average value being smaller than twice the waiting time.

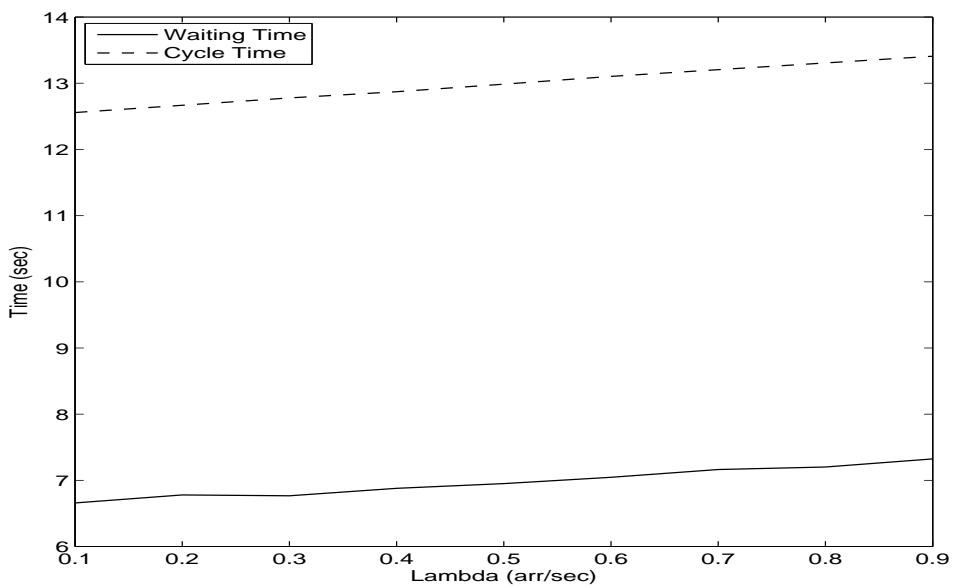
Figures 7.10a, 7.10b and 7.10c show the results of a smaller frame size where the minimum and maximum data bytes have been changed to 27 and 67 respectively and the information bytes changed to 13.



(a) Waiting Time



(b) Queue Length



(c) Cycle Time

Figure 7.10: RRP Simulation Results: Small Frame Sizes

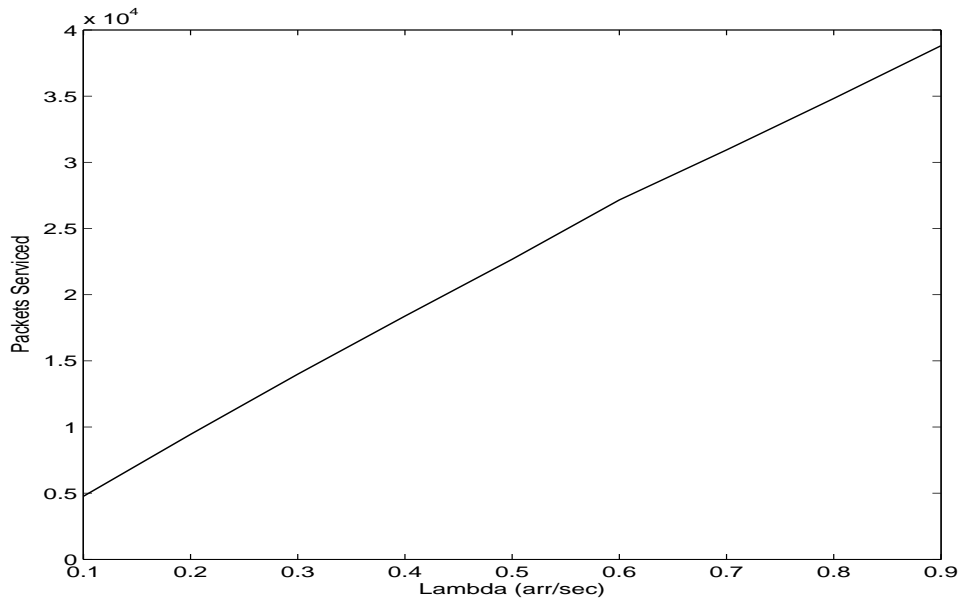
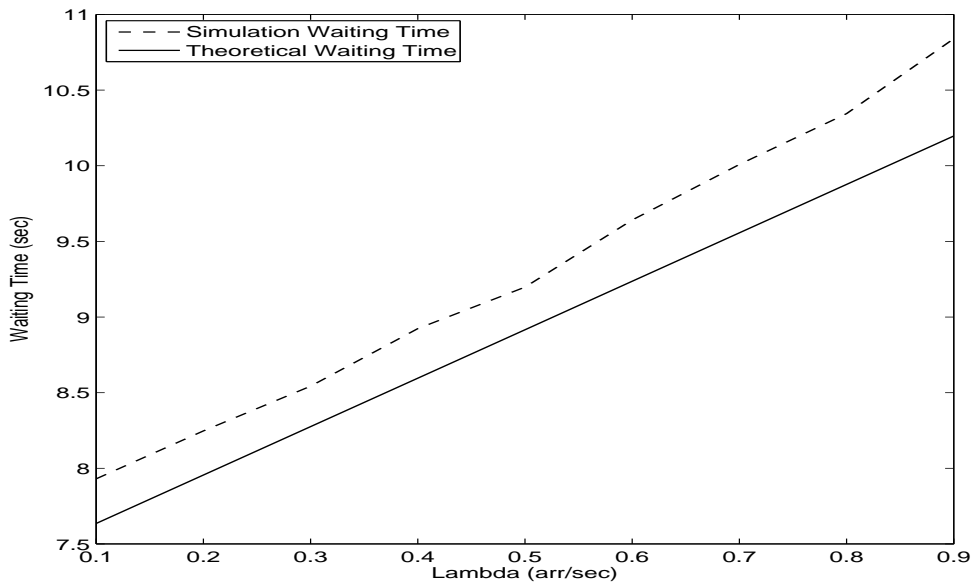


Figure 7.11: RRP Simulation Results: Arriving Events Serviced

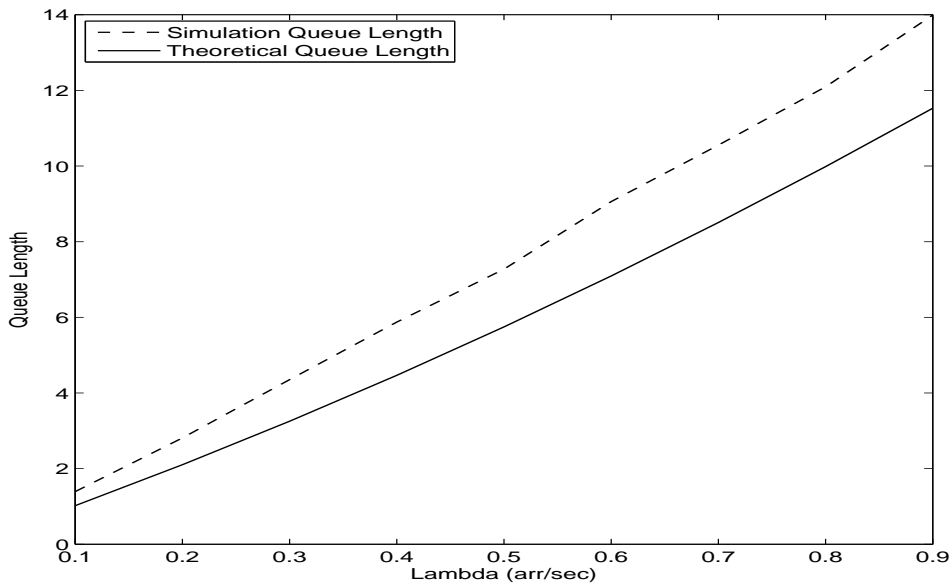
The same trend is followed as that found in the theoretical model. Figure 7.11 has been added to indicate the number of events serviced vs lambda. The theoretical and simulation results will be compared in the next section.

### 7.5.3 Comparison of Results

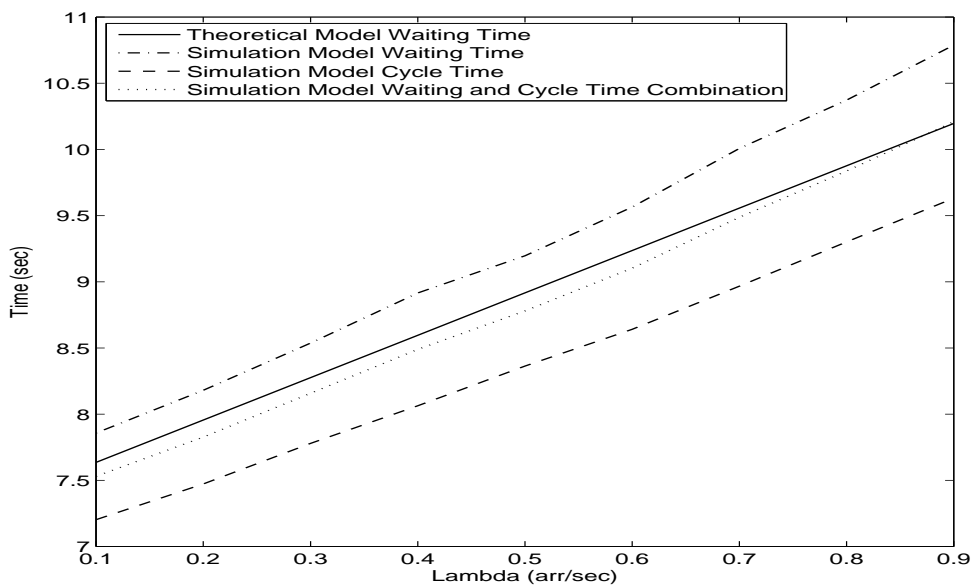
Figure 7.12 compares the waiting times and queue lengths of the large data frame sizes of the simulation and theoretical models. The results correspond to 3.8 - 6.2% difference for the waiting time and 9 and 21% for the queue lengths. The divergence of the queue length can be explained by the fact that the theoretical queue length is determined by the utilization of the communication channel. When the arrival rate  $>$  service rate the system is unbounded and the queue length for the theoretical model will no longer be accurate. The theoretical model's waiting time is also bounded by the utilization of the channel and will exhibit the same behaviour. If the simulation's average waiting time and average cycle time are added together and averaged, a closer approximation to the theoretical model waiting time is provided. This is presented in Figure 7.12c and gives a difference of between 0.1-1.3%. The same comparisons has been done for the smaller frame sizes, as shown in Figure 7.13. It can clearly be seen that with a lower utilization the protocol exhibits better performance with regards to waiting time vs  $\lambda$  and queue length vs  $\lambda$  (accuracy of 0.1-2.5% between the model and the simulation combination). As the utilization picks up, the protocol performance deteriorates as illustrated in Figure 7.14. In Figure 7.14 150 Stations have been implemented with a large frame size and the rest of the parameters are the same as for Figure 7.12.



(a) Waiting Time

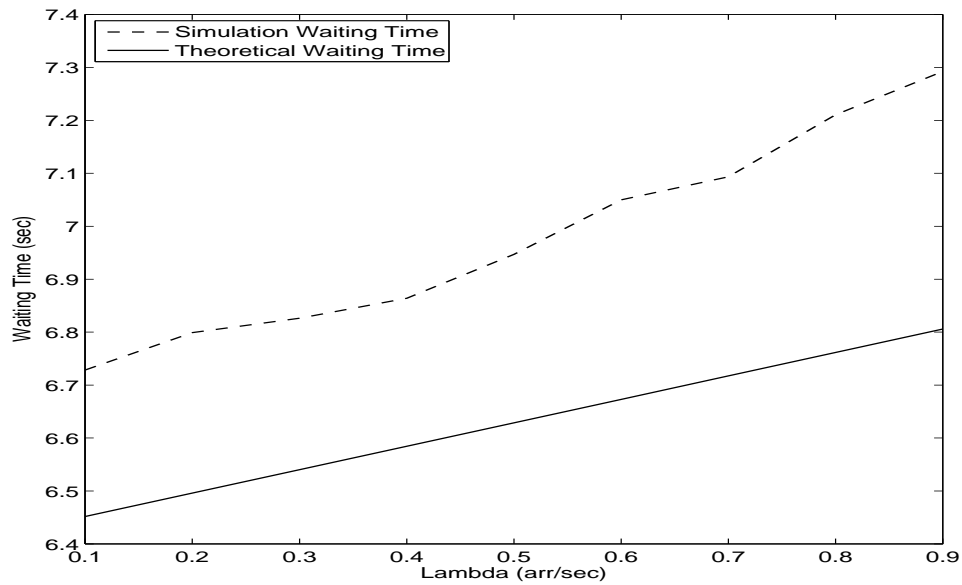


(b) Queue Length

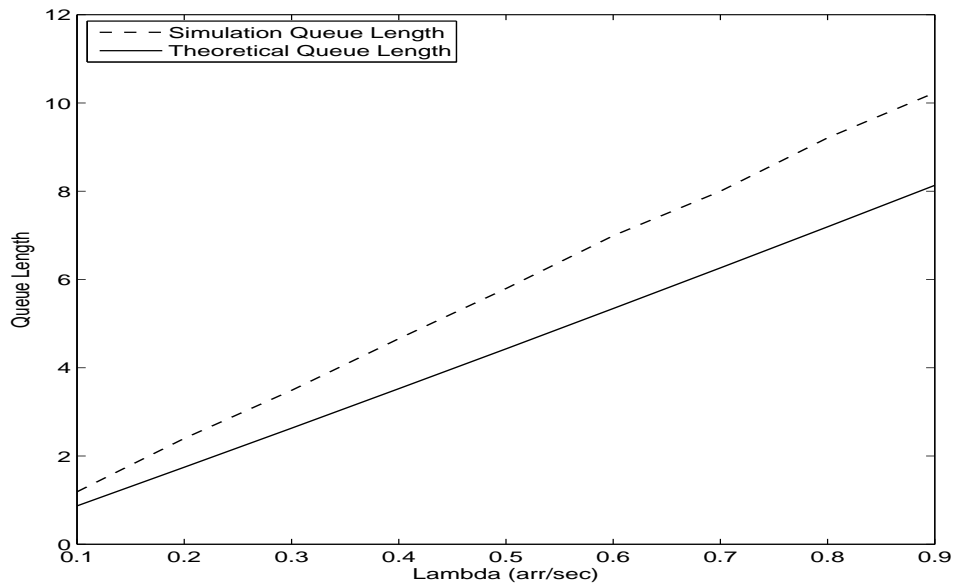


(c) Cycle Time

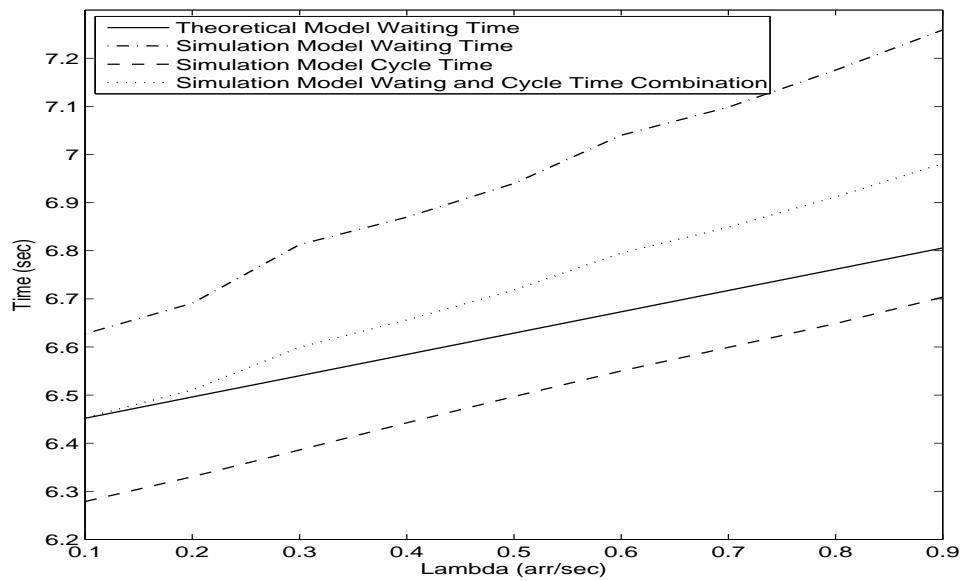
Figure 7.12: RRP Theoretical and Simulation Comparison: Large Frame Sizes



(a) Waiting Time

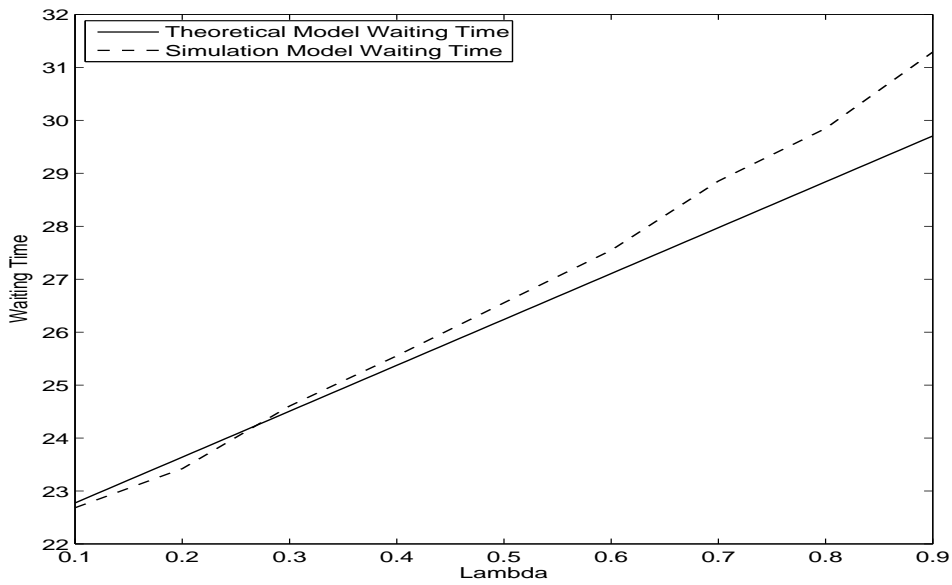


(b) Queue Length

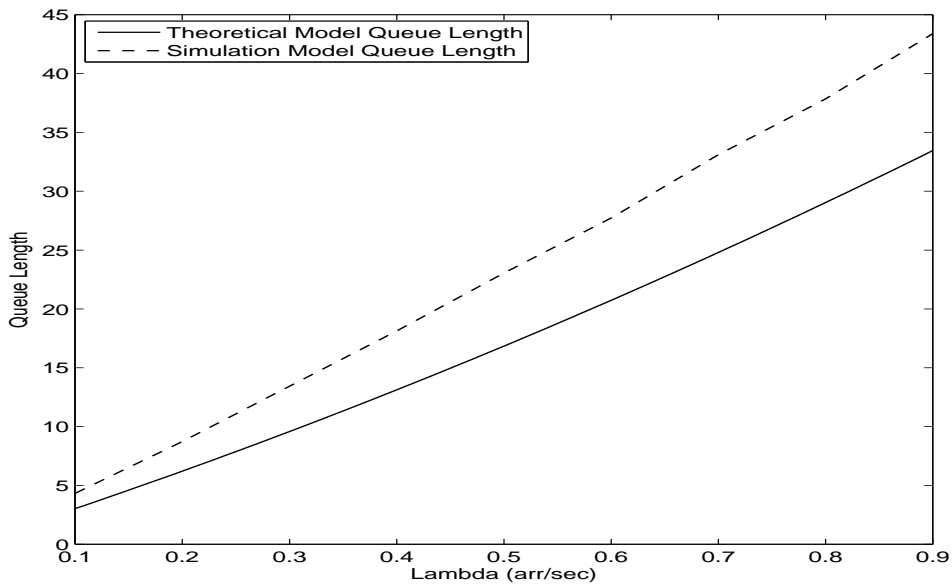


(c) Cycle Time

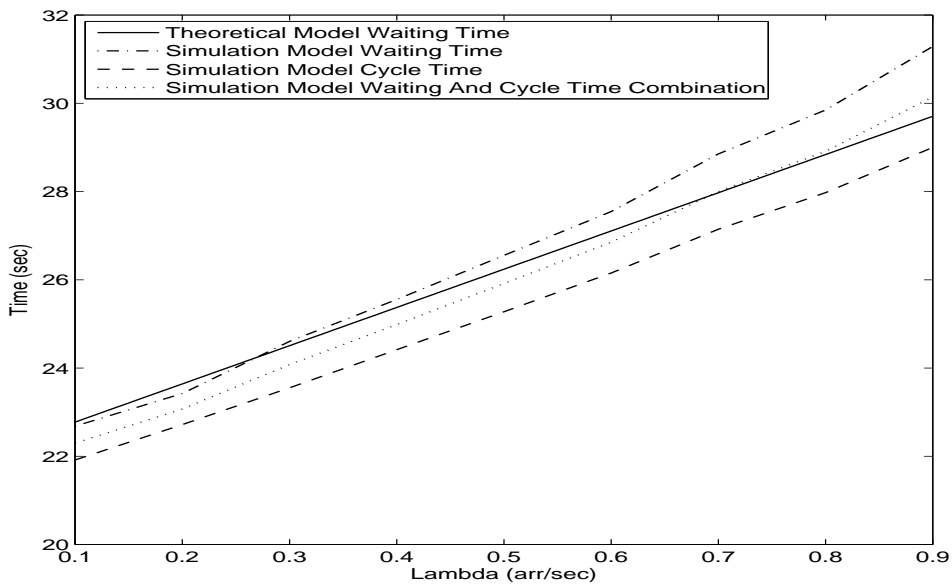
Figure 7.13: RRP Theoretical and Simulation Comparison: Small Frame Sizes



(a) Waiting Time



(b) Queue Length



(c) Cycle Time

Figure 7.14: RRP Theoretical and Simulation Comparison: Large Frame Sizes with 150 Stations



## 7.6 Summary

This chapter has discussed the theoretical and simulation models for the RRP protocol. A detailed analysis of the frame transmission time has been provided. The theoretical model has been implemented using an analytical mathematical approach. The simulation model configuration makes use of three main discrete processes, namely the Generator, Server and Station processes. These processes have been explained with the help of UML based activity diagrams. The implementation of the simulation model within Java is the same as that discussed in Chapter 6. The results have validated previous assumptions that the protocol has a high overhead at low arrival rates, but increases in performance in keeping with increased network traffic. The results of the theoretical and simulation models compared very favourably within realistic boundaries.

The implementation of the ATW protocol is covered in Chapter 8.

## Chapter 8

# Adaptive Tree Walk: Modelling and Simulation

### 8.1 Introduction

As far as could reasonably be determined the ATW protocol has not been used in telemetry networks, but was suspected to have the properties to deliver the best combination of the non-persistent CSMA and RRP protocols. An unslotted ATW protocol has been developed as an alternative to the known slotted version, to cater for the special needs of telemetry networks. The configuration and theoretical and simulation modelling thereof, are described in this chapter.

### 8.2 Unslotted Adaptive Tree Walk protocol

A typical station tree is presented in Figure 8.1 to assist with further discussions.

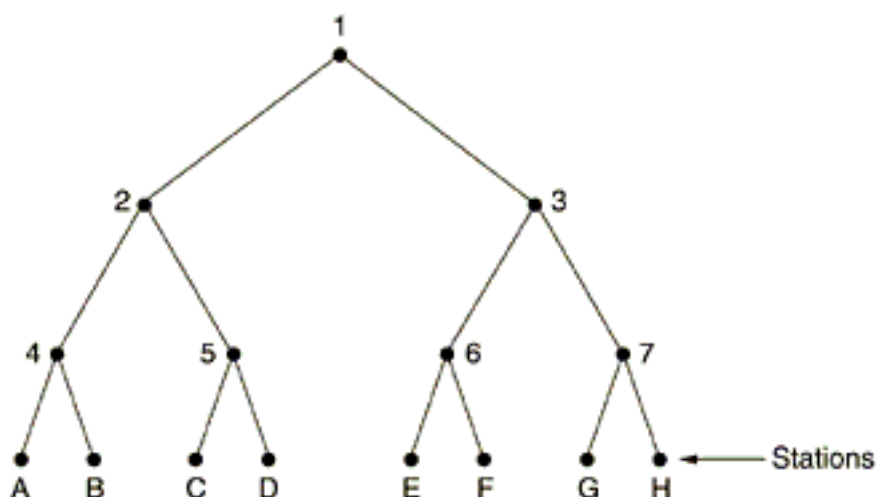


Figure 8.1: Adaptive Tree with eight stations from [37]

The more generalized computer network ATW algorithm is based on time slots. This version also assumes that each station will try to transmit its data when a request is sent from the base station, as in the case of RRP. With high data

latencies this approach would be inefficient because the required slot time would have to be increased to accommodate the latencies as well as the data transmitted. Furthermore, the various remote stations can be routed through different numbers of repeaters, causing different latencies and each slot size will have to be given a size equal to the sum of the maximum latency and data frame transmission time in the system. To substantiate this, a system is presented where each station in the system has data to transmit. The total time to cycle through the tree and service all stations will then be (with no burst noise assumed):

$$t_{cyc} = (2^i - 1)(t_{dqf} + t_{dtf}) + (2^i)(t_{dqf} + t_{dtf}) \quad (8.1)$$

$$t_{dqf} = t_{pr} + t_{ps} + t_{ta} + t_{rd} + t_{dl} + t_{ib} \quad (8.2)$$

$$t_{dtf} = t_{pr} + t_{ps} + t_{ta} + t_{rd} + t_{dl} + t_{ib} + t_{db} \quad (8.3)$$

where:

$t_{cyc}$  Time required to cycle through the tree, servicing all stations in the network (ms)

$t_{dqf}$  Time required to send a data request frame from the base station (ms)

$t_{dtf}$  Time required to send a data frame from the remote station (ms)

$t_{pr}$  Preamble (ms)

$t_{ps}$  Postamble (ms)

$t_{ta}$  Turnaround (ms)

$t_{rd}$  RXEnd (ms)

$t_{dl}$  Data Latency (ms)

$t_{ib}$  Information bits transmission time (ms)

$t_{db}$  Data bits transmission time (ms)

From Equation 8.1 it can be seen that the overhead  $[(2^i - 1)(t_{dqf} + t_{dtf})]$  is almost equal to the actual servicing time required per station  $[(2^i)(t_{dqf} + t_{dtf})]$ . The original ATW protocol has, therefore, been adapted to service networks with higher data latencies, by making it unslotted and changing the remote station to base station communication flow. The base station will start by sending a request to all stations asking for an indication of data. If only one station replies, confirming that it has data, the server will again send a request to that specific station requesting its data, to which the station will reply with its data frame, completing communication. If more than one station replies with a data indication, the system will go down one level in the tree and the base station will send a data indication to the stations available at this level. The process continues in the same recursive way as before. The cycle time will now be altered as follows:

$$t_{cyc} = (2^i - 1)(t_{dif} + t_{drf}) + (2^i)(t_{dif} + t_{drf} + t_{dqf} + t_{dtf}) \quad (8.4)$$

$$t_{dif} = t_{drf} = t_{pr} + t_{ps} + t_{ta} + t_{rd} + t_{dl} + t_{ib} \quad (8.5)$$

where:

$t_{dif}$  Time required to send a data indication frame from the base station (ms)

$t_{drf}$  Time required to send a data indication response frame from the remote station (ms)

$t_{dqf}$  Time required to send a data request frame from the base station (ms)

$t_{dtf}$  Time required to send a data frame from the remote station (ms)

The number of frames sent is increased but the size of the data indication and response frames are smaller than the data request frames, which are again smaller than the data frames, reducing the total transmission time of the system. This approach will be used in the theoretical and simulation modelling of the protocol.

## 8.2.1 Configuration

In the protocol configuration there are four types of frames which can be sent. They are the data request, data, data indication and data indication response frames. The data frame will contain the data bytes of the event which occurred at the remote station and its length is determined by the amount of data to be sent, whereas the data request, data indication and response frames should be as short as possible. To allow the protocol to work correctly, no retries for data indication and response frames are allowed. The effect of failure of these two frames, will require an average extra waiting time of  $\frac{t_{cyc}}{2}$  at the remote station, where  $t_{cyc}$  is dependent on the utilization of the network. If a data request or data frame is being sent, the station with data has already been identified and a retry can be invoked if data is found to be corrupted at the base station. The probability of error for the frames is again determined from the bit error rate of the radio. This is taken into account by lengthening the frame transmission time accordingly:

$$t_{die} = \left( \frac{2 * Information\ Bytes * 10}{10^6} \right) * \frac{t_{cyc}}{2} \quad (8.6)$$

$$t_{dre} = \left( \frac{(Information\ Bytes + Data\ Bytes) * 10}{10^6} \right) * (t_{dtf} + t_{dqf} + t_{to}) \quad (8.7)$$

where:

$t_{die}$  Time contributed to a data indication cycle in error (ms)

$t_{dre}$  Time contributed to a data request cycle in error (ms)

$t_{cyc}$  is the average of the maximum and minimum utilization of the channel. As described in the RRP protocol, it is assumed that once a remote station has an event to transmit, it cannot create another one and this will, in effect, reduce the total arrival rate. The average cycle time is determined iteratively:

$$\mu = \frac{1}{t_{dif} + t_{drf} + t_{dqf} + t_{dtf}} \quad (8.8)$$

$$\rho = \frac{k\lambda}{\mu} \quad (8.9)$$

$$t_{cyc} = \sum_{k=1}^N \left[ \left( 2^{\log 2^k} - 1 \right) (t_{dif} + t_{drf}) \left( \frac{k\lambda}{\mu} \right) + \left( 2^{\log 2^k} \right) (t_{dif} + t_{drf} + t_{dqf} + t_{dtf}) \left( \frac{k\lambda}{\mu} \right) \right] \quad (8.10)$$

$$t_{cyc-avg} = \frac{t_{cyc}}{2} \quad (8.11)$$

Compensating the frame transmission time for probability of error:

$$t_{dib} = t_{dis} = t_{dif} + t_{cyc-avg} = t_{drf} + t_{cyc-avg} \quad (8.12)$$

$$t_{drq} = t_{dqf} + t_{dre} \quad (8.13)$$

$$t_{dat} = t_{dtf} + t_{dre} \quad (8.14)$$

where:

$t_{dib}$  Time required to send a data indication request from the base station (ms)

$t_{dis}$  Time required to send a data indication response from the remote station (ms)

$t_{drq}$  Time required to send a data request from the base station (ms)

$t_{dat}$  Time required to send the data from the remote station (ms)

The above approach has been implemented in the simulation as an unslotted ATW. Furthermore there is no base-remote acknowledgement as this would increase the probability of failure. If the data request or data frame should fail, the server will timeout and resend the data request frame. The results from the simulation are presented in Section 8.5.2.

### 8.3 Theoretical Modelling

#### 8.3.1 Introduction

The unslotted ATW protocol can be modelled as a M/M/1/N/FIFO queue, using the same techniques as discussed in Section 4.3.3. Both the arrival and service rates are Poisson processes. With arrivals occurring independently, it is in line with the requirements of modelling the queue as a Markovian chain. The state diagram of the queue is shown in Figure 8.2

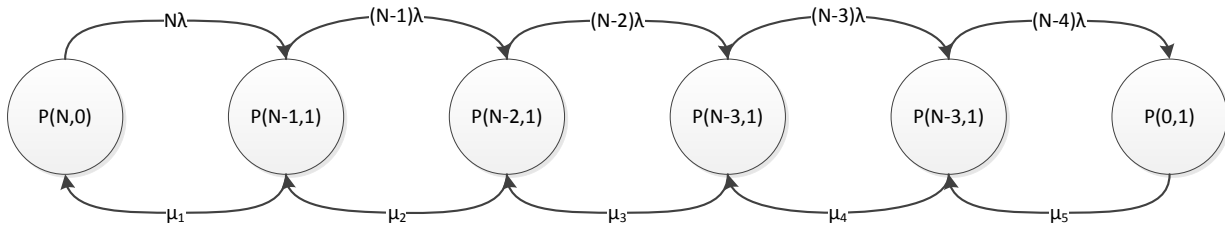


Figure 8.2: ATW State diagram

The increase in polling overhead caused by the traversal of the tree has been implemented as an increase in service time. Overhead will increase with stations in the queue. This requires that the level of the tree first be determined by:

$$i = \log_2 N_s \tag{8.15}$$

where:

- $i$  Represents the tree level
- $N_s$  Number of stations in the network

A fractional tree level will be obtained if all the external nodes of the tree are not filled. This is not possible and with the requirement for a balanced tree, Equation 8.15 is therefore modified to:

$$i = \log_2 N_s + 1 \tag{8.16}$$

All stations can now be accommodated as external nodes at this tree level. Equation 8.15 can now be used to calculate the factor by which the service time must be increased:

$$f_{pf} = \frac{[(2^{i-1} - (2^{i-1} - N_s)) + (2^{i-1} + 1) / 2] / N_s}{2} \tag{8.17}$$

$$f_{u_k} = f_{pf} + f_{util} \times \frac{l_{req}}{2} \times f_{pf} \tag{8.18}$$

where:

- $f_{pf}$  The average number of polls for a tree containing  $N_s$  stations
- $f_{u_k}$  The polling factor
- $l_{req}$  Required tree level to start sending data indication requests with a single data indication response back
- $f_{util}$  Represents the utilization of the communication channel for  $k$  stations in the queue

$f_{pf}$  makes use of the maximum internal and external nodes of the tree, to determine the average number of polls for a tree containing  $N_s$  stations. This is then used to determine the polling factor  $f_{u_k}$ .  $f_{u_k}$  takes into account the utilization of the channel and the effect that it will have on the total number of poll requests required.  $l_{req}$  is determined from Equation 5.3, which has been modified to account for the fact that the number of stations in the network might not fill all the external nodes of the tree.  $l_{req}$  can be expressed as follows:

$$l_{req} = \log \left( 2^{i-1} \left( \frac{2^{i-1}}{N_s} \right) \right) \quad (8.19)$$

$f_{util}$  is iteratively calculated for different states and expressed as follows:

$$f_{util} = \frac{(N_s - k) \lambda}{\mu_{util}} \quad (8.20)$$

$$\mu_{util} = \frac{1}{t_{dat} + t_{drq} + t_{dib} + t_{dis}} \quad (8.21)$$

where  $k$  is the current queue length. The service time for  $k$  stations in the queue can now be calculated:

$$\mu_s = \frac{1}{t_{dat}} \quad (8.22)$$

$$\mu_q = \frac{1}{t_{drq}} \quad (8.23)$$

$$\mu_{i_k} = \frac{f_{u_k}}{t_{dib} + t_{dis}} \quad (8.24)$$

$$\mu_k = \mu_s + \mu_q + \mu_{i_k} \quad (8.25)$$

where:

- $\mu_s$  Service rate for the data frame
- $\mu_q$  Service rate for the data request frame
- $\mu_{i_k}$  Service rate for the data indication and data indication response frames for queue length  $k$
- $\mu_k$  Service rate for the network with  $k$  stations in the queue

The state diagram shown in Figure 8.2 represents a network with 5 stations and will be used as example. The symbols used are defined as follows:

- $N$  Number of stations in the network
- $\lambda$  Arrivals per second for each station
- $\mu_k$  Service rate for a queue length of  $k$

$P(m, n)$  Probability that  $n$  stations are inactive and can create new events to the queue, where  $m$  describes the state of the server. If  $m = 0$ , then the server is idle and not servicing any station event, whereas if  $m = 1$ , the server is busy servicing an event from a station.

Parameter definitions and characteristics are as before.  $N$  is the total number of stations in the network and thus the total possible arrival rate is  $\lambda_{max} = N\lambda$ . The different states and transitions can be set out as follows:

- $P(N, 0)$  is the probability that  $N$  stations can produce new events. It also means that  $N$  stations are currently idle and not waiting for service from the server, therefore currently not part of the queue. Due to the fact that  $N$  stations are idle, the server has nothing to service and is idle as well. From the state  $P(N, 0)$  the system can transition to state  $P(N - 1, 1)$  at a rate of  $N\lambda$ , due to the fact that  $N$  stations have the probability to create new events. This means that the next station, with new events, will arrive at an average rate of  $N\lambda$  and move the system to the new state  $P(N - 1, 1)$ . The system can only transition to its closest neighbour which is  $P(N - 1, 1)$ .
- $P(N - 1, 1)$  is the probability that  $N - 1$  stations can produce new events and 1 station is waiting for its service to be completed. The server is no longer idle as there is a station to be serviced. This state has two immediate neighbours and can transition to either of them, depending on whether the next arrival occurs first or service to the current station is completed first. The system can therefore move to state  $P(N - 2, 1)$  at a rate of  $(N - 1)\lambda$  or it can transition back to state  $P(N, 0)$  at a rate of  $\mu_1$ . If the average service rate is greater than the average arrival rate,  $\mu_1 > (N - 1)\lambda$ , the system will transition back to  $P(N, 0)$ ; if, however, this is not the case and the average arrival rate is greater than the average service rate,  $\mu_1 < (N - 1)\lambda$ , it is probable that a new station will arrive before service to the current station is completed. The system will then transition to the next state  $P(N - 2, 1)$ .
- $P(N - 2, 1)$  is the probability that  $N - 2$  stations can produce new events, i.e. 1 station is being serviced and 1 station is waiting to be serviced. The process of transition to other states follows the same process as described above. The difference is that the average arrival rate has now decreased, as there are fewer stations that can produce new events and the fact that one station is waiting in the queue to receive service, as the server is already busy servicing another station. The service rate will also decrease, due to more nodes of the tree being traversed. This process continues up to the last state  $P(0, 1)$ .
- $P(0, 1)$  is the probability that 0 stations can produce new events, 1 station is being serviced and  $N - 1$  stations waiting to be serviced. No more arrivals can occur as all stations are in the queue, waiting to be serviced. The only transition that can take place is to the previous state ( $P(1, 1)$ ) after the current station has finished its service period. Again, this transition will take place at the service rate,  $\mu_s$ , which has decreased due to the number of nodes traversed during the search for stations to be serviced.

The global balance equations for the state diagram are as follows:

$$\begin{aligned}
 P(N, 0)N\lambda - P(N - 1, 1)\mu_1 &= 0 \\
 P(N - 1, 1)((N - 1)\lambda + \mu_1) - P(N, 0)\lambda &= 0 \\
 P(N - 2, 1)((N - 2)\lambda + \mu_2) - P(N - 1, 1)(N - 1)\lambda - P(N - 3, 1)\mu_3 &= 0 \\
 \dots \dots \dots \dots \dots \dots &= 0 \\
 \dots \dots \dots \dots \dots \dots &= 0 \\
 P(0, 1)\mu_5 - P(1, 1)\lambda &= 0
 \end{aligned} \tag{8.26}$$

$$P(N, 0) + P(N - 1, 1) + P(N - 2, 1) + \dots + P(0, 1) = 1 \tag{8.27}$$

Equation 8.27 represent the summation of all state probabilities and must equal one. The above equations can be represented in a full rank state transition matrix:

$$\begin{bmatrix}
 N\lambda & -\mu_1 & 0 & 0 & 0 & & & & & \\
 N\lambda & (N-1)\lambda + \mu_1 & -\mu_2 & 0 & 0 & & & & & \\
 0 & -(N-1)\lambda & (N-2)\lambda + \mu_2 & -\mu_3 & & & & & & \\
 & & & & & & & & & \\
 & & & & & & \dots & \vdots & \vdots & \vdots \\
 & & & & & & \dots & 0 & -\lambda & \mu_5 \\
 1 & 1 & 1 & 1 & 1 & \dots & 1 & 1 & 1 & 1
 \end{bmatrix} \times$$

$$\begin{bmatrix}
 P(N,0) \\
 P(N-1,1) \\
 P(N-2,1) \\
 \vdots \\
 \vdots \\
 P(0,1) \\
 0
 \end{bmatrix} = \begin{bmatrix}
 0 \\
 0 \\
 0 \\
 \vdots \\
 \vdots \\
 0 \\
 1
 \end{bmatrix} \tag{8.28}$$

To determine the average waiting time, Little’s Law is used, but it must be remembered that  $\lambda$  differs for each state. This means that the waiting time will have to be calculated iteratively, making use of the queue length contribution at each specific state. The queue length can be calculated as follows:

$$N = \sum_{k=1}^N kP(N-k,1) \tag{8.29}$$

and the corresponding waiting time will be:

$$\begin{aligned}
 T_k &= \frac{N_k}{\lambda_k} \\
 &= \sum_{k=1}^N \frac{kP(N-k,1)}{[(N+1)-k]\lambda}
 \end{aligned} \tag{8.30}$$

The model has been implemented in Matlab and the results been compared to the simulation model in Section 8.5.

## 8.4 Simulation Modelling

As discussed in Chapter 3, the unslotted ATW protocol has been implemented with DESMO-J. A process oriented modelling approach has been taken to implement the protocol. The protocol has been implemented as outlined in Section 8.2 with the same fundamental assumptions as for RRP.

### 8.4.1 Processes-Orientated Modelling

The simulation modelling under DESMO-J is also process orientated, as set out for CSMA and RRP.



### 8.4.1.1 The Generator Process

The activity diagram of the Generator process is shown in Figure 8.3. The process lifecycle contains an interruptible region that can only be interrupted by the Server process. This region is used to gather station events during the binary tree polling cycle of the Server process. When the server is done with its polling cycle, it will interrupt this process, from where the Generator process will write the station events into the format that the server requires. The interruptible region will be described first.

The interruptible region of the Generator process is entered immediately after the process has been initiated. If the Arrival Bit Set Object (ABSO) flag has not been set, the ABSO class has not been initiated yet and the process will do so. This class is used to indicate that a station has an event, by making use of a BitSet (a BitSet is a Java Object that consists of a list of bits that can be set high or low) and enters the Station process object with relevant information into an ArrayList. After the ABSO class has been initiated, the ABSO flag is set not to clear or garbage collect the ABSO class object. The next arrival time of a station with an event is obtained from the Arrival Rate Generator. The same principle, as described in RRP, to obtain the next station ID, is followed. A newly created Station object will be added to the Station Event List Generator, which is a BitSet, by making use of its Station ID. The above process is repeated. If interrupted, the process completes and exits the interruptible region of the Generator process. The information required by the server is set up which includes a list of stations with events, a queue of station objects containing relevant information of each station and a queue which contains the Station process instances. The reason a queue is used for this is the fact that the stations must be entered into the queue according to their ID, as that is the sequence in which they will be accessed by the binary tree and which corresponds to the station event list. After the Generator Station Waiting Queue has been copied into the Server Station Waiting Queue, the Generator Station Waiting Queue must be reset and the ABSO flag is set to clear all relevant variables, queues and BitSets used within the Generator process. The Server process is scheduled to activate after the Generator process is passivated and entered into the event list. A check is done to determine whether the interruptible region has been interrupted after a new arrival time has already been determined. The current time can't be changed when interrupted, therefore the new event can only be scheduled now, and the next arrival occurred flag is set; else the flag is set to indicate that the next arrival has not yet occurred. The Generator process will now begin with gathering new events for the start of the next Server process polling cycle. This can be seen as the moment that the server or base station sends a data indication request for the first time. Any station that creates an event after that initial message has been sent will not be able to participate in the current polling cycle. The Server process will be activated by the scheduler, as it is the next process in the event list.

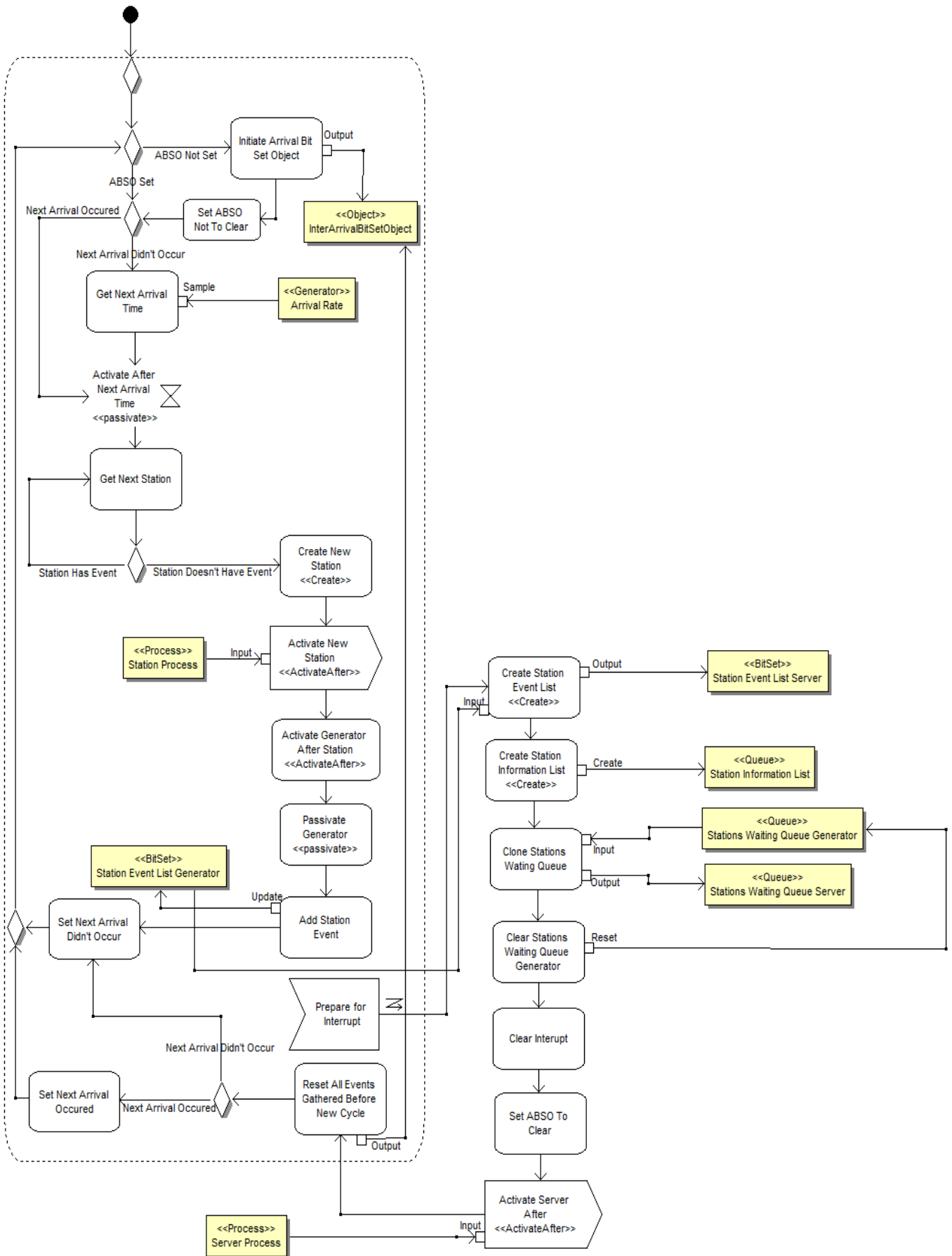


Figure 8.3: ATW Generator Process Activity Diagram

### 8.4.1.2 The Station Process

The Station process activity diagram is shown in Figure 8.4 and follows the same principles as the RRP Station process.

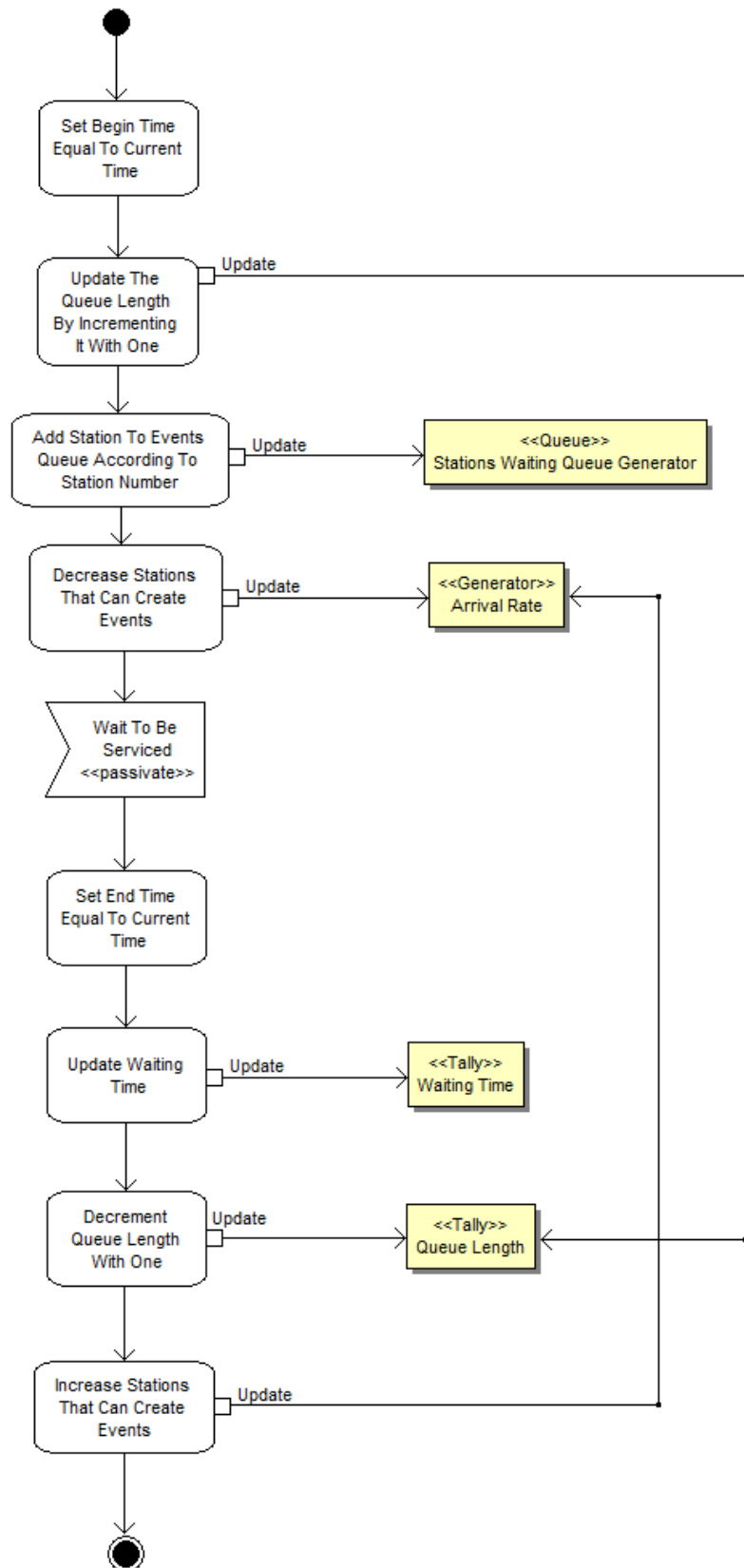


Figure 8.4: ATW Station Process Activity Diagram

### 8.4.1.3 The Server Process

The Server process activity diagram is shown in Figure 8.5. When the server process is initialized, it will immediately set the interrupt for the Generator process and passivate itself, waiting for the Generator process to schedule it for activation. When activated, the server creates a new binary tree for the number of stations in the network. The Server process loads the Station Event List into the Binary Tree allowing each node in the tree to contain the number of bits corresponding to the number of stations that fall under it. To measure the total time required to complete a polling cycle, the current time is obtained and set as the begin time of the cycle. Next the root of the tree is obtained, meaning the complete Binary Tree object is obtained and the recursive region (named TreePolling in Figure 8.5) of the tree is entered.

The TreePolling region will be called recursively until all the stations set to have events in the Station Event List have been serviced. To do this the Station Event List of the current Binary Tree Object, which will change as the TreePolling region is called recursively, is obtained. The data indication and data indication response times are determined and the Server process schedules itself for re-activation, is entered into the event list by the scheduler and is passivated. Upon re-activation, the Station Event List of this Binary Tree Object is re-visited. If it contains more than one active station, then a garbled data indication response frame has been received by the server. If the tree still has a left node, this node is loaded as the new active tree, effectively creating a new Binary Tree Object, and the TreePolling region is called recursively until only one active station remains, or a left recursive call is made and no active stations exists. If the latter is the case, then the right child node will be loaded as the active tree and the TreePolling region is called recursively as before. If one active station remains after recursive calls, the data request and data response times are determined and the Server schedules itself for re-activation, is entered into the event list by the scheduler and passivated. Upon re-activation by the scheduler the first station in the Station Waiting Queue for the Server is obtained and removed from the Station Waiting Queue. The Station process object obtained from the Station Waiting Queue is scheduled for re-activation and the Server is scheduled to be re-activated afterwards. Both processes are entered into the event list and the Server process is passivated. This specific TreePolling region is completed and no further recursion will be done from it and it returns to the previous level of recursion, if it exists. If the previous level returns from a left child recursion, the Station Event List is checked to determine if more than one station still has events. If this is the case then the right node is loaded as the active tree and the TreePolling region is called recursively. Upon return from a right child node's recursive call, there is nothing left to be done and it will return to the previous level of recursion if it exists.

If no more recursion levels exist, all stations have been serviced and the TreePolling region is exited. The current time is obtained and the polling cycle counter updated. The Generator process is then again interrupted, obtaining all the station events that occurred during the polling cycle time that has just passed. The above process is then repeated until the simulation time is completed. The concept of recursion and the binary tree will be discussed in more detail in Section 8.4.3.1.

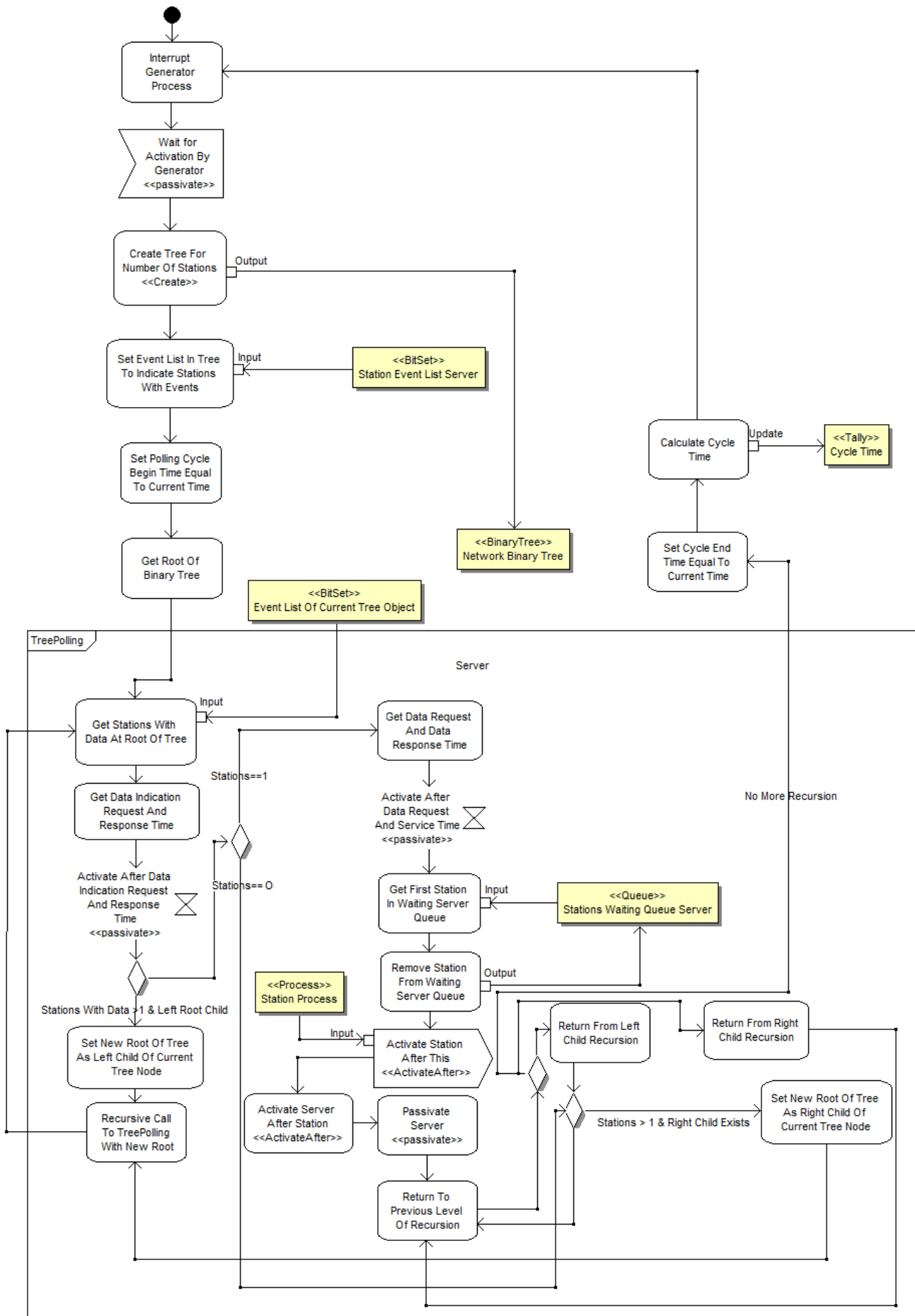


Figure 8.5: ATW Server Process Activity Diagram

## 8.4.2 Simulation Model Configuration

The simulation has been configured for the ATW strategy. Detailed internal mechanisms are similar to those described for CSMA and RRP. The only difference is the use of different database tables to store the statistical values.

## 8.4.3 Simulation Program

Some of the more important aspects of the simulation program will be described in this section. This includes the Binary Tree Iteration, which forms the heart of the protocol, as well as the ArrivalBitSetObject class, which contains important methods used within the Generator process. Furthermore, an important new class from DESMO-J, known as the ProcessQueue class, is used to represent the Station Waiting Queue for the Server and Generator processes. This class allows simulation processes to be queued in a list that can be prioritized. This queue class handles only objects derived from the SimProcess class, which is the super class for all processes. The default sort order for this queue is FIFO, and the SimProcess can be inserted into the queue by priority. In this case the station ID is used to assign the priority to the Station SimProcess entered into the queue.

### 8.4.3.1 The Binary Tree Iteration

The Binary Tree is iteratively traversed using a method derived from the preorder traversal of a binary tree, using the following algorithm from [13]:

```

Algorithm preOrderBinaryTree(Tree, NodeV)
    perform the action for node NodeV
    if v is an internal node then
        preOrderBinaryTree(Tree, NodeV.leftChild(NodeV))
            {recursive traverse left subtree}
        preOrderBinaryTree(Tree, NodeV.righChild(NodeV))
            {recursive traverse right subtree}

```

#### 8.4.3.1.1 Binary Tree Example

The following binary tree example will assist with explanation of the recursion used within the TreePolling region of the Server process. Please refer to Figure 8.6, which shows a tree representing a network with 8 stations. Stations 1,2,4,5,7 and 8 have events that need to be serviced by the server. The sequence of zeros and ones next to each node depicts the Station Event List (of type BitSet) used in the Server process. The TreePolling routine is entered, starting with the full tree. From the Station Event List, '11011011', it can be seen that more than one station has an event to be serviced. A recursive call is made, setting the left child of the root equal to the new root in the recursive call. Again there is more than one station requiring servicing, i.e. '1101', and the left child of the current tree is set as the new root and the recursive call made. The same happens in the next recursive call, '11'. In the new recursively called TreePolling routine the root node now has no more children and represents station 1. This one event is serviced, no right child exists and the current TreePolling routine ends and, therefore, moves up one level in the TreePolling routine. This is now subsequent to the state where it made the recursive call containing the left child, as the new root. Therefore, the number of events to be serviced at this level is still 2, but as the left child recursive call has already been made, the process moves to the right child. More than one event exists and, the right child is set as the new root and the recursive call is made. In the new Tree polling routine the root of only one event waiting to be serviced now exists, which is that of station 2. The one event is serviced,

no right child exists and the current TreePolling routine ends and therefore moves up one level. The left hand and right hand halves of the tree are differentiated by the addressing scheme. The entire tree is searched by following the same iterative process as set out above.

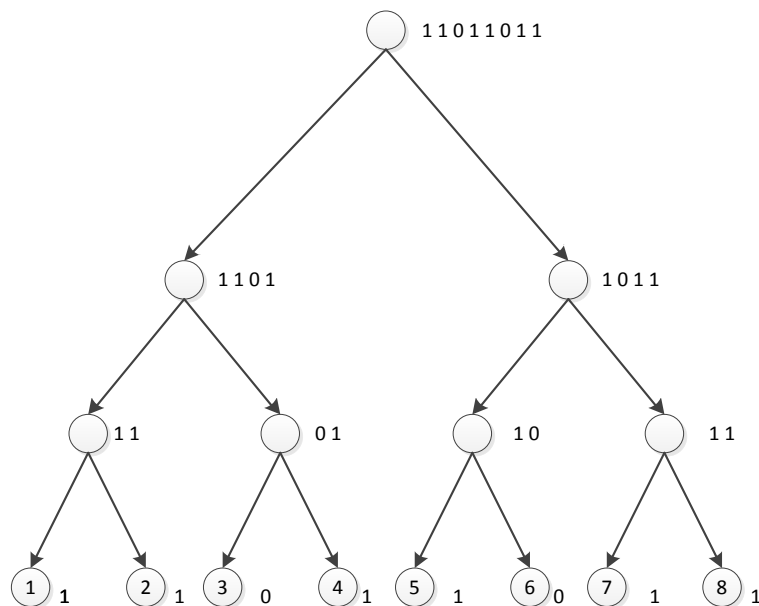


Figure 8.6: ATW Binary Tree Representing 8 Stations

The main use of the ArrivalBitSetObject class is to create the BitSet event lists used by the Server and the Generator process. In this scope the number of events that can occur at each station before it is serviced is limited to one, but the simulation also allows for multiple events. All station events are logged in an ArrayList.

## 8.5 Results

In this section the results of the theoretical and simulation models are compared and analysed.

### 8.5.1 Theoretical Results

The theoretical model has been based on the parameters as per Figure 8.7. Note that data indication request and response frame sizes have been referred to as poll bytes in all figure legends.

Number of stations [1.250]	50
Min Propagation distance [Km]	30
Max Propagation distance [Km]	80
Frame Length poll [bytes]	5
Preamble [ms]	1
Postamble [ms]	1
Data latency [ms]	11
Frame Length data [bytes]	180
Frame Length info [bytes]	20
Channel Capacity [bps]	4800
RXend [ms]	30
Step Size	0.1
Min Arrival Time(arr/sec)	0.1
Max Arrival Time(arr/sec)	0.7
Time Out (ms)	2500
Number of Repeaters	3
Turn Around Time (ms)	1

Figure 8.7: ATW Theoretical Parameters

The results for the waiting time and queue length, are presented in Figure 8.8. The figure also depicts the change in waiting time with the change in data indication request and response frame sizes. From this, it is clear that overall system performance deteriorates with the increase in these frame sizes, as can be expected.



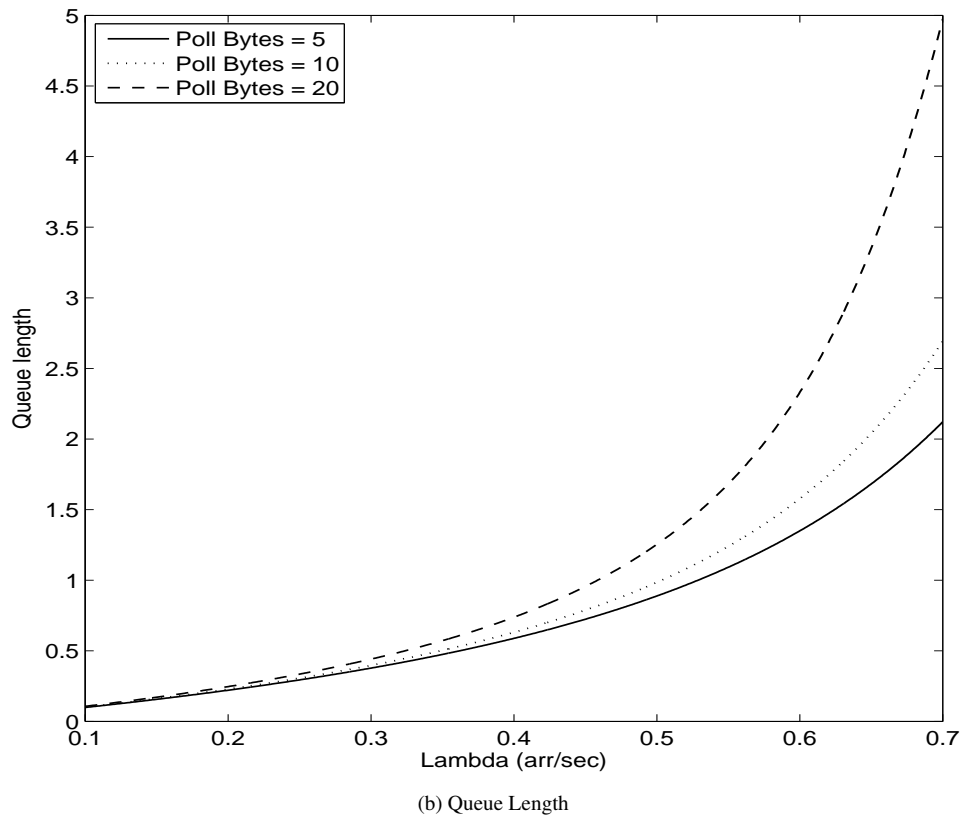
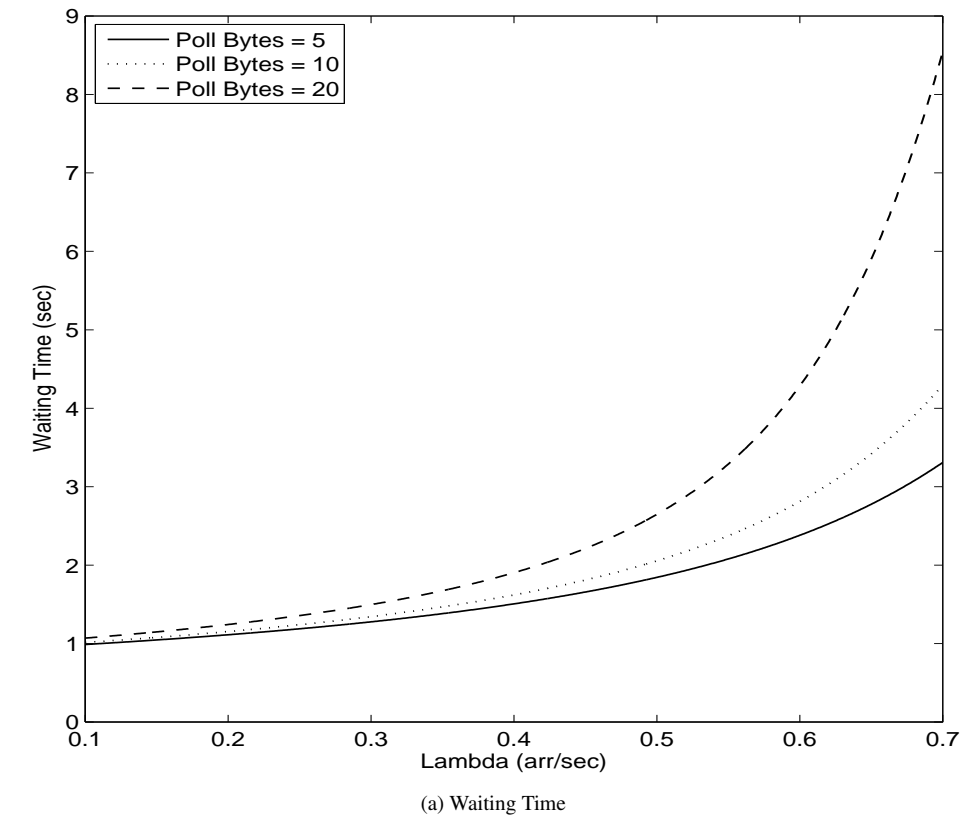
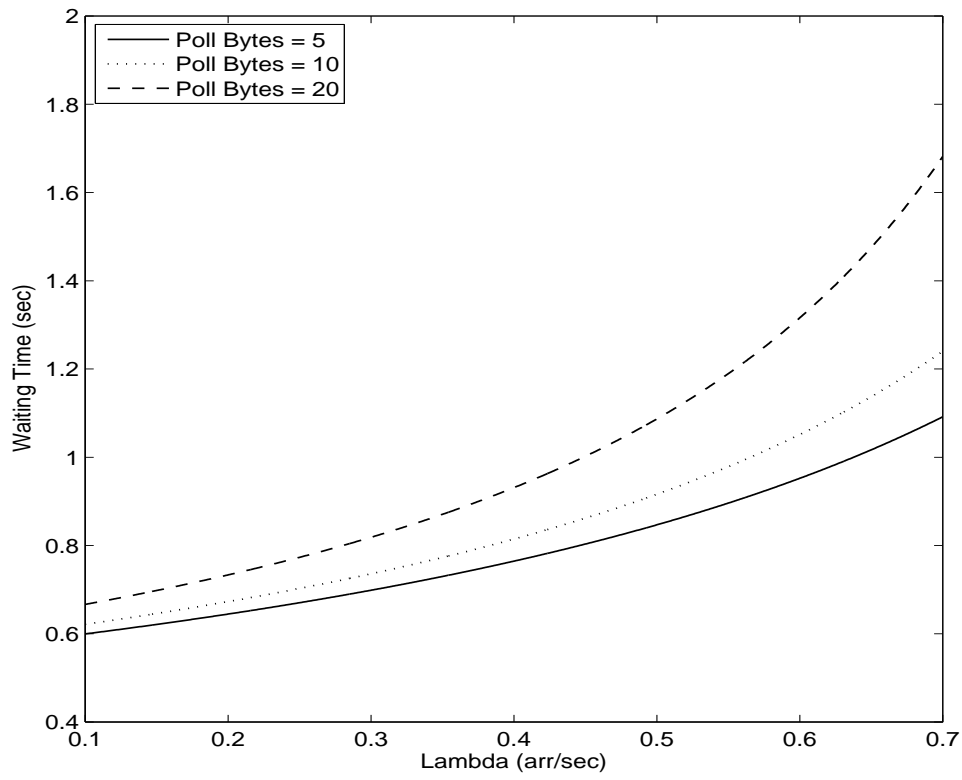
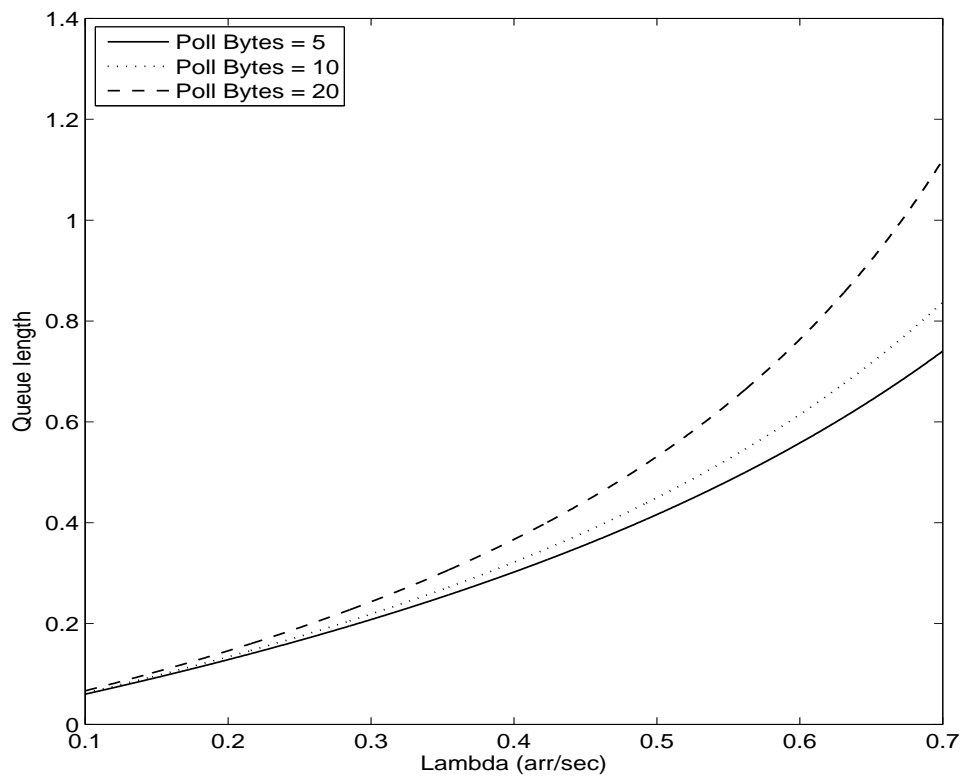


Figure 8.8: ATW Theoretical Results: Large Frame Sizes

Figure 8.9 makes use of the same parameters as Figure 8.8, except that the data response and data request frame sizes have been decreased by decreasing the data (averaged as 47 bytes) and information (average as 13 bytes) bytes. Different data indication request and response frame sizes have been implemented, giving the same effect as seen in the previous figure. The overall efficiency of the system increases drastically with a decrease in the data indication frames' sizes.



(a) Waiting Time



(b) Queue Length

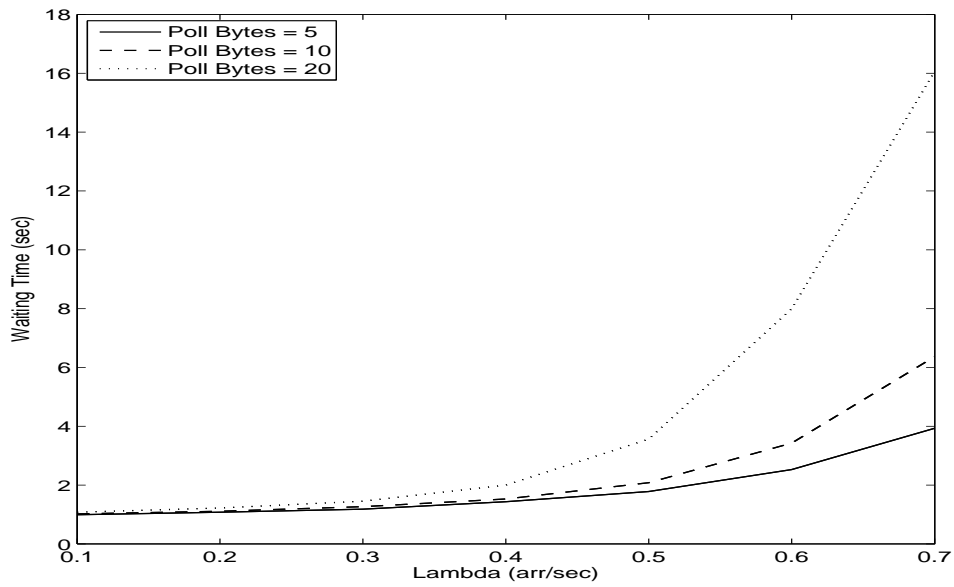
Figure 8.9: ATW Theoretical Results: Small Frame Sizes

## 8.5.2 Simulation Results

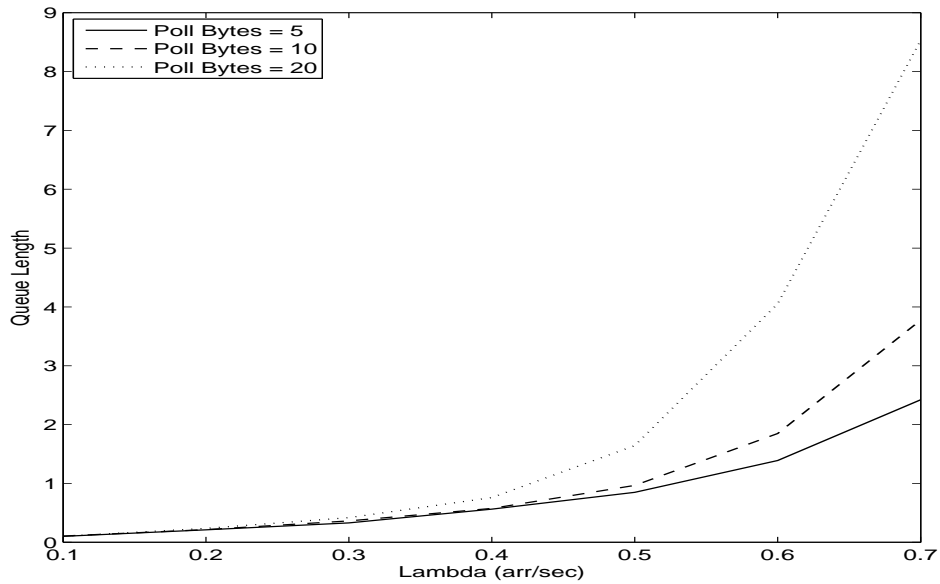
The parameters used for the simulation are similar to those of the Theoretical results discussed in the previous section, except that data frame size varies between minimum and maximum, as does the propagation distance. The simulation parameters are listed in Figure 8.10. The simulation time has been taken as 25000 seconds with a transient time of 3600 seconds. The step size is 0.1 arrivals per second. Figure 8.11 presents the average waiting and cycle times, as well as the queue length for large frame sizes (data bytes between 170 and 190 bytes, information bytes equal to 20 bytes). Data indication request and response frames with sizes of 5,10 and 20 bytes have been used. Figure 8.12 shows the average waiting and cycle times, as well as the queue length for small frame sizes (data bytes between 27 and 67 bytes, information bytes equal to 13 bytes), with data indication request and response frame sizes varied as before. Although the performance of the protocol decreases with larger data indication request and response frames, it can also be seen that the degradation in performance is less with smaller data frame sizes. This is expected, as the channel is occupied for shorter periods of time.

Number of Stations	50	Upper Distance (km)	80
Channel Capacity (bps)	4800	RXEnd (ms)	30
Data Bytes Mimimum	170	Data Latency (ms)	11
Information Bytes	20	System Begin Arrival Rate (arv/sec)	0.1
Preamble (ms)	1	System End Arrival Rate (arv/sec)	0.9
Postamble (ms)	1	Simulation Length (sec)	5000
Lower Distance (km)	30	Step Size (sec)	0.1
Data Bytes Maximum	190	Poll Information Bytes	5
Number of Repeaters	3	TurnAround (ms)	1
Time Out (ms)	2500		

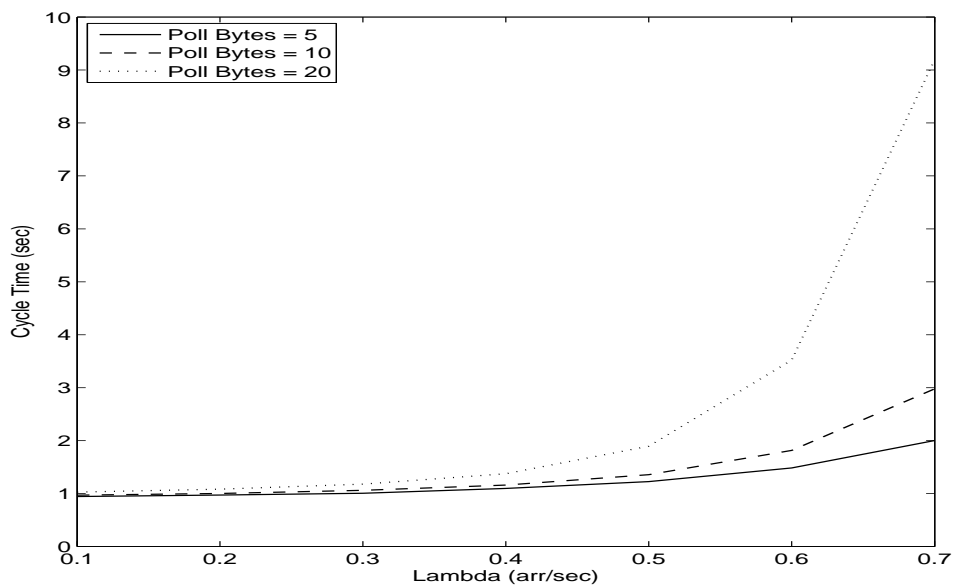
Figure 8.10: ATW Simulation Parameters



(a) Waiting Time

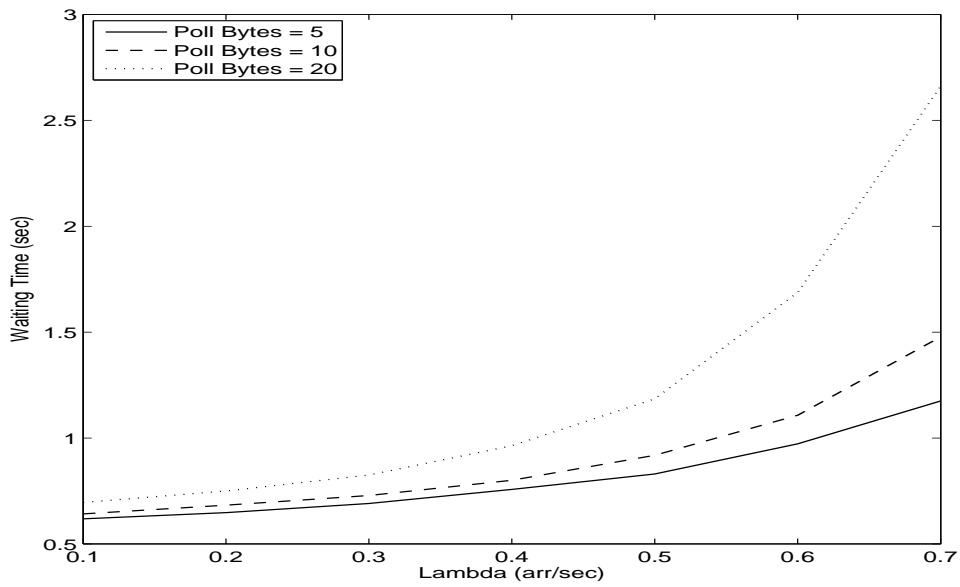


(b) Queue Length

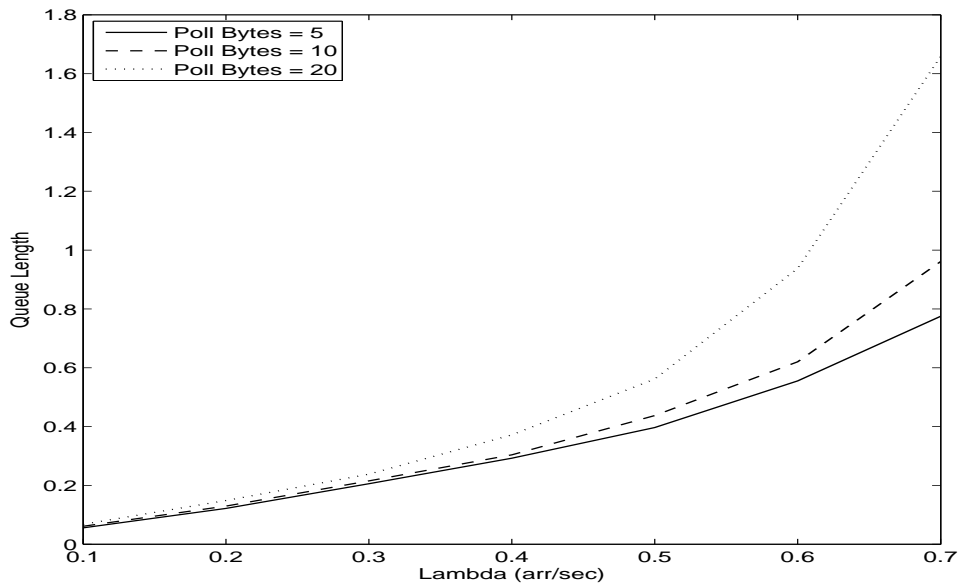


(c) Cycle Time

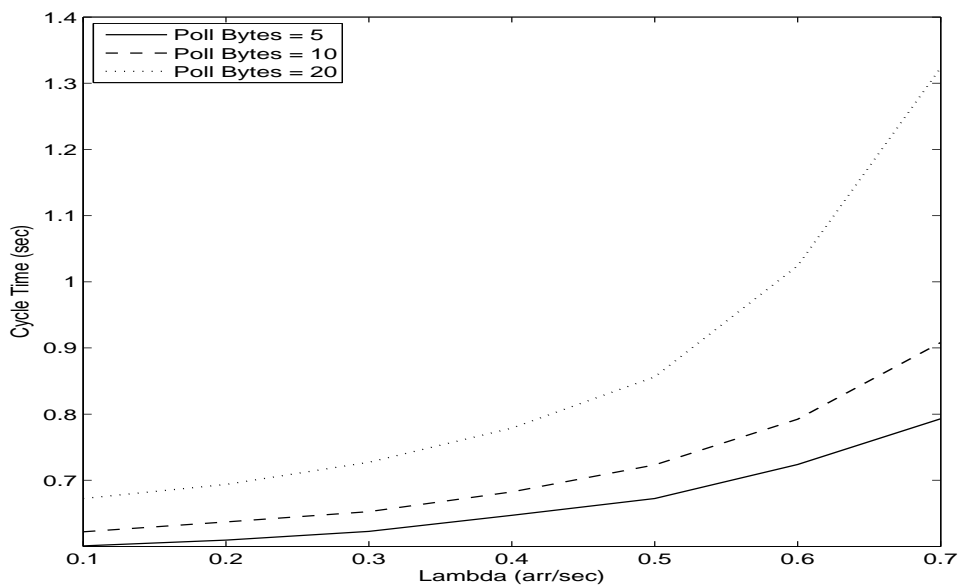
Figure 8.11: ATW Simulation Results: Large Frame Sizes



(a) Waiting Time



(b) Queue Length



(c) Cycle Time

Figure 8.12: ATW Simulation Results: Small Frame Sizes

Due to the fact that the system always starts with the station with the lowest station ID, it makes sense that the station with the largest station ID will always have a longer waiting time. This is proven in Figure 8.13 with an average arrival rate of 0.7 arrivals per second. An optimization has been implemented allowing the recursive process to be swapped around by running the right child nodes recursively first as indicated in the algorithm below. The two different types of algorithms are swapped around every 10 cycles, giving the results as per Figure 8.13. The optimization has no effect on the waiting time, cycle time and queue length of the complete system.

```

Algorithm preOrderBinaryTree(Tree, NodeV)
    perform the action for node NodeV
    if v is an internal node then
        preOrderBinaryTree(Tree, NodeV.rightChild(NodeV))
        {recursive traverse left subtree}
        preOrderBinaryTree(Tree, NodeV.leftChild(NodeV))
        {recursive traverse right subtree}

```

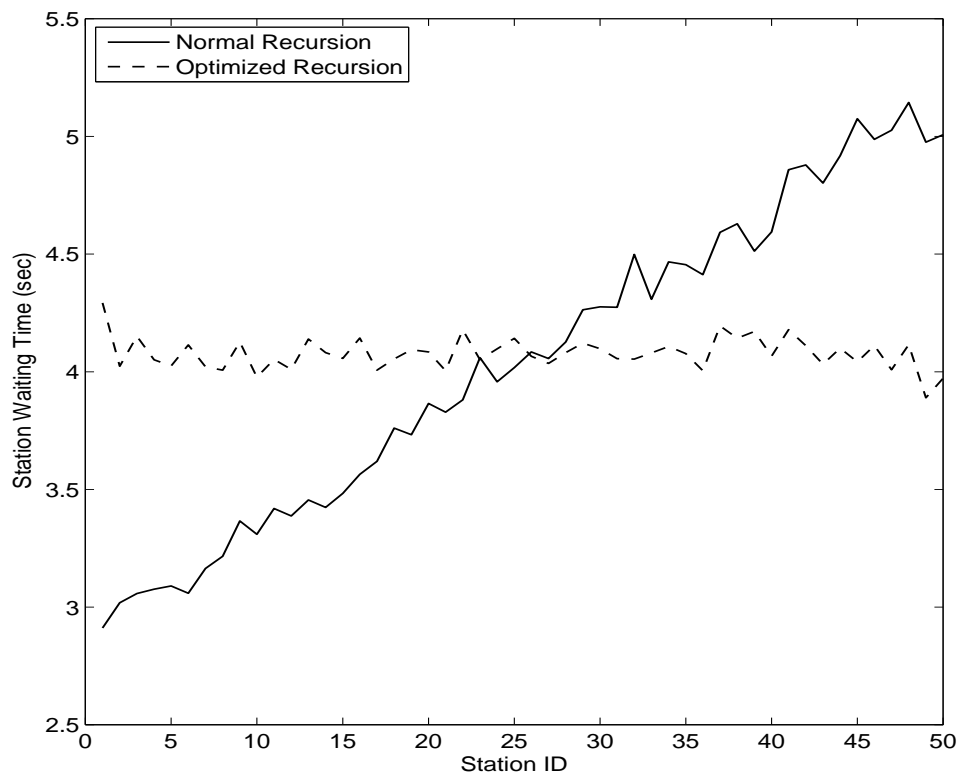
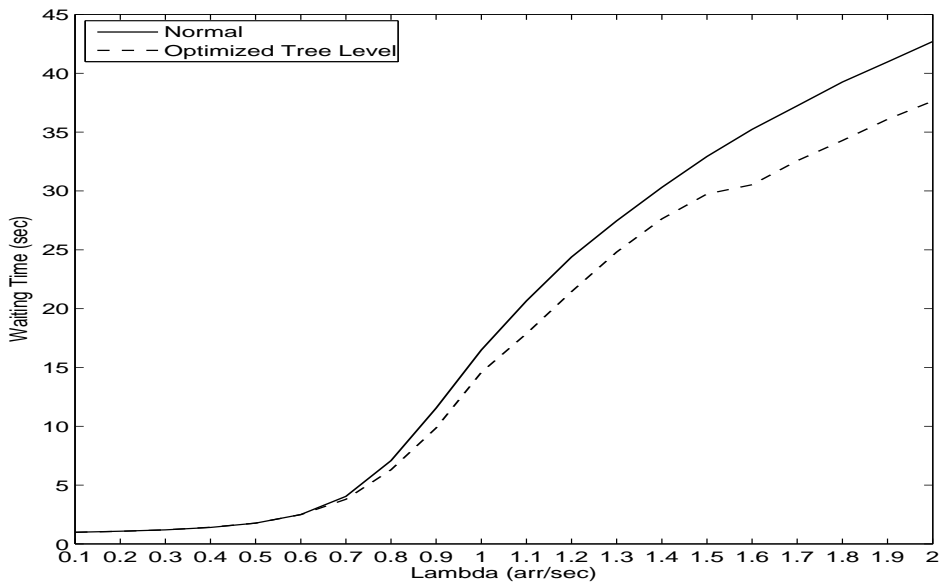
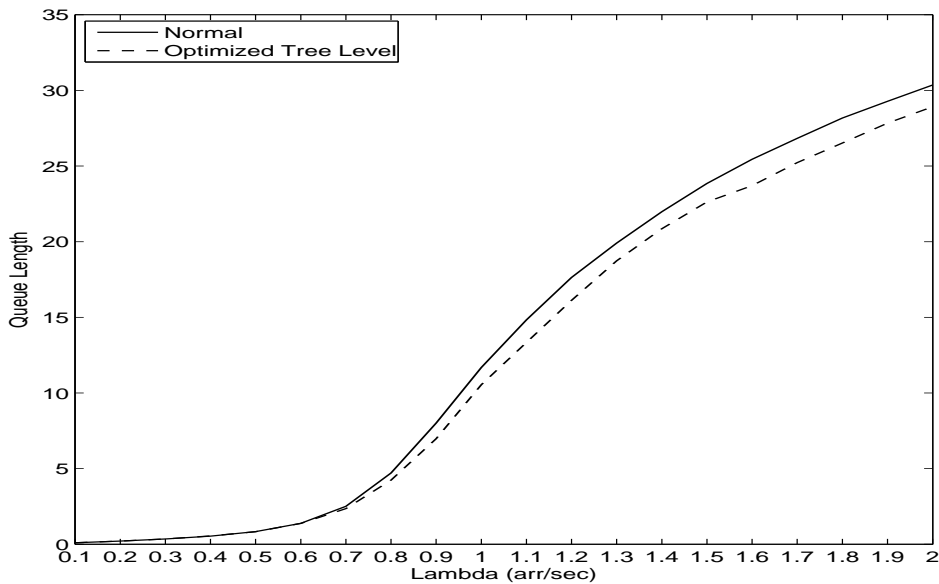


Figure 8.13: ATW Simulation Results: Effect Of Optimization on Station Waiting Time

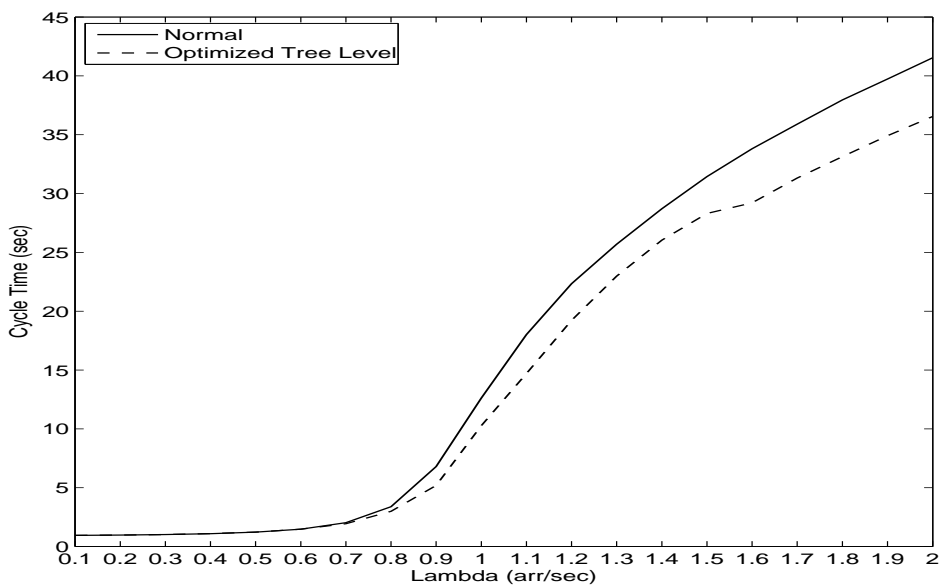
The simulation has been given the added functionality of starting at the optimized level of the tree as discussed in Section 5.5.2. Therefore, the `TreePolling` routine which handles the recursion of the Server process will only start data indication requests and responses once the optimal tree level has been reached. This is determined by making use of the average queue length to determine the number of stations requiring service on average. Equation 5.1 is used to determine the optimized level of the tree, thus where to start sending and receiving data indication frames. This has been implemented with a new arrival rate range of 0.1 to 2.0 arrivals per second, to better demonstrate the effect. The data indication request and response frame sizes have been set to 5 bytes. This is demonstrated in Figure 8.14. It has also been implemented for an arrival rate range of 0.1 to 0.7 arrivals per second with different sized data indication request and response frames, by setting these frames' bytes equal to 5, 10 and 20 bytes respectively, for large data frame sizes. This is shown in Figure 8.15.



(a) Waiting Time

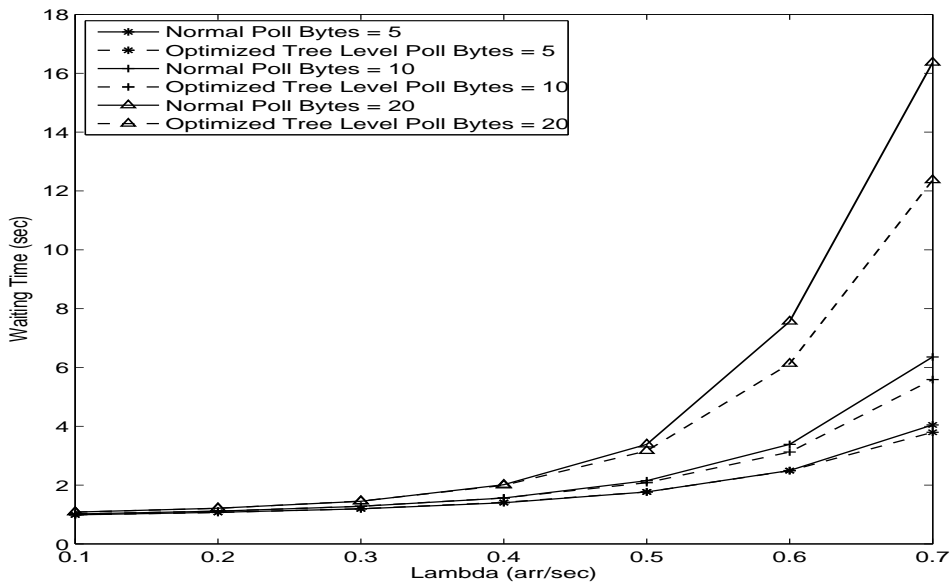


(b) Queue Length

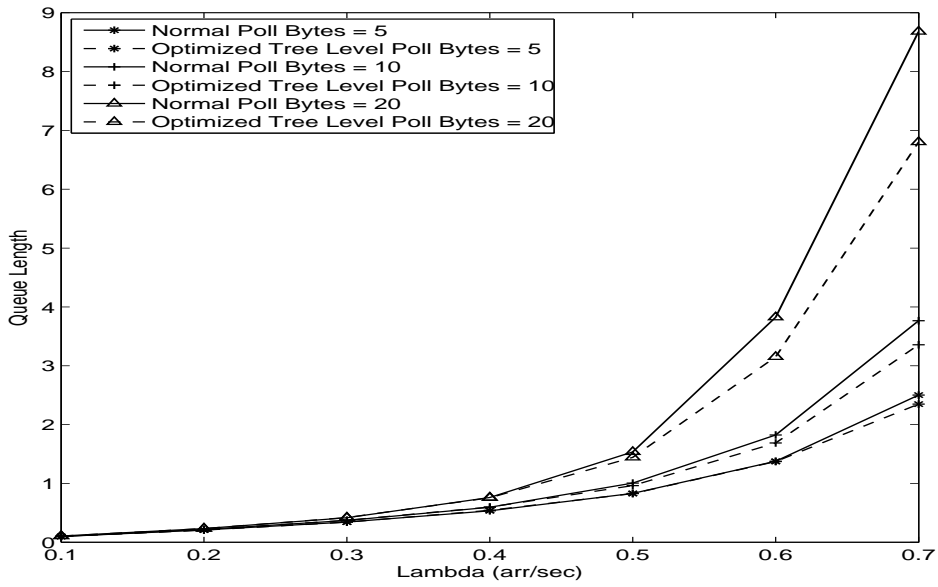


(c) Cycle Time

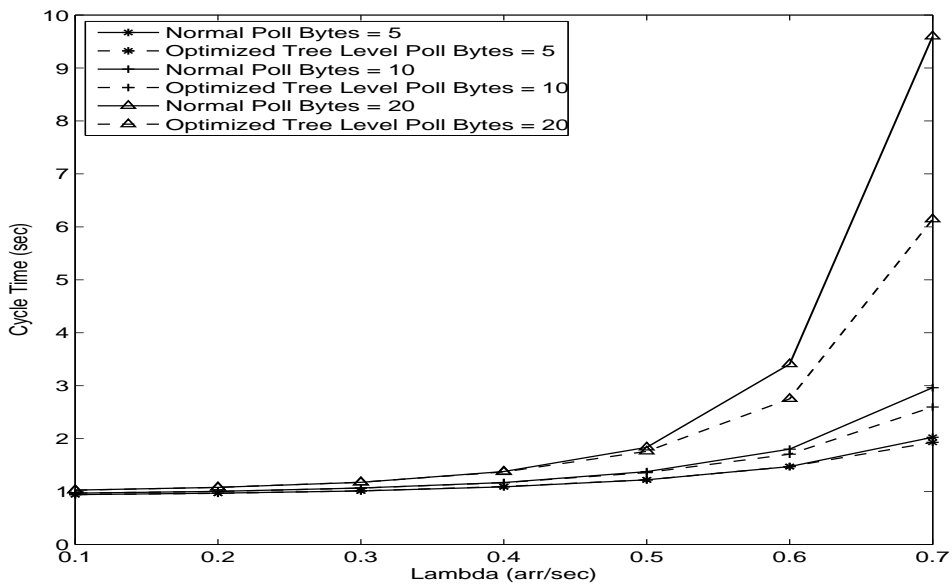
Figure 8.14: ATW Simulation Results: Optimized Tree Level, High Arrival Rates and Large Frame Sizes



(a) Waiting Time



(b) Queue Length



(c) Cycle Time

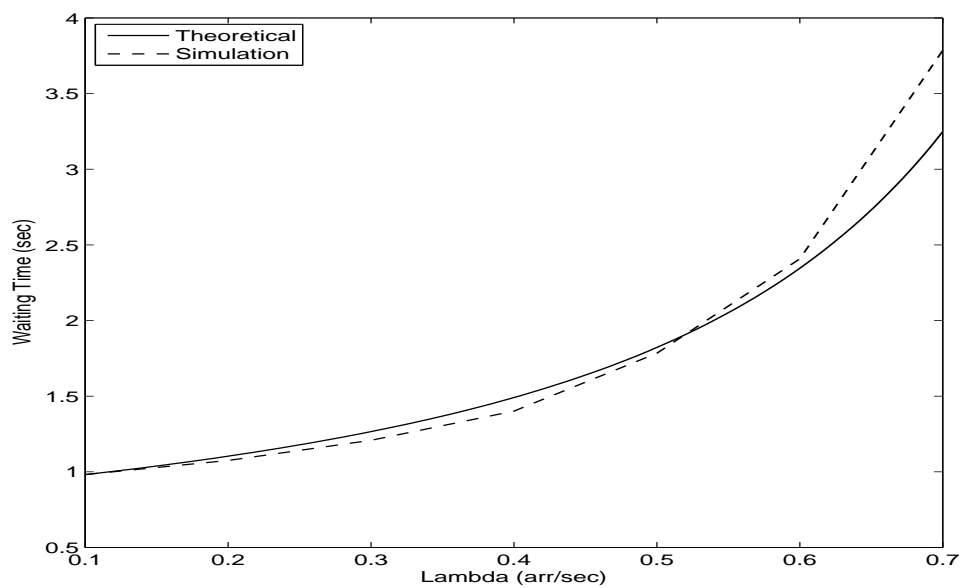
Figure 8.15: ATW Simulation Results: Optimized Tree Level, Low Arrival Rates and Large Frame Sizes



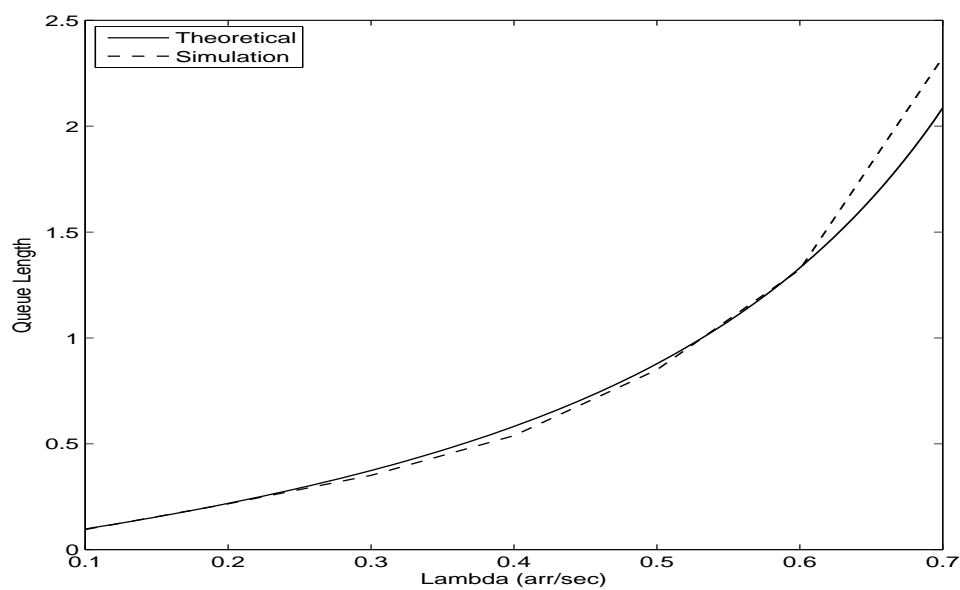
### 8.5.3 Comparison of Results

Figures 8.16 and 8.17 compare theoretical and simulation results of the large and small data frame sizes respectively, with the parameters presented in previous sections. For both large and small data frame sizes, the theoretical comparison of the waiting time is within 5% of the simulation, except at the higher rates where it diverges to 9 and 12% for large and small data frames respectively. This stabilizes at higher arrival rates with an average difference of 6.5%. It is clear that within realistic limits, there is a very good comparison between the two sets of results.

Figure 8.18 contains results for different data indication request and response frame sizes for large data frames. It can be seen from the graphs that as the utilization picks up, results from the simulation and theoretical model diverge, but start to converge again later. The reason for this is that as the average queue length increases, the effective arrival rate decreases, thus decreasing the effective utilization. The model and simulation comparison is still within 3% for arrival rates below 0.5 arrivals per second. Finally, Figure 8.19 presents the stabilizing effect for a small data frame and a data indication and request frame size of 20 bytes.

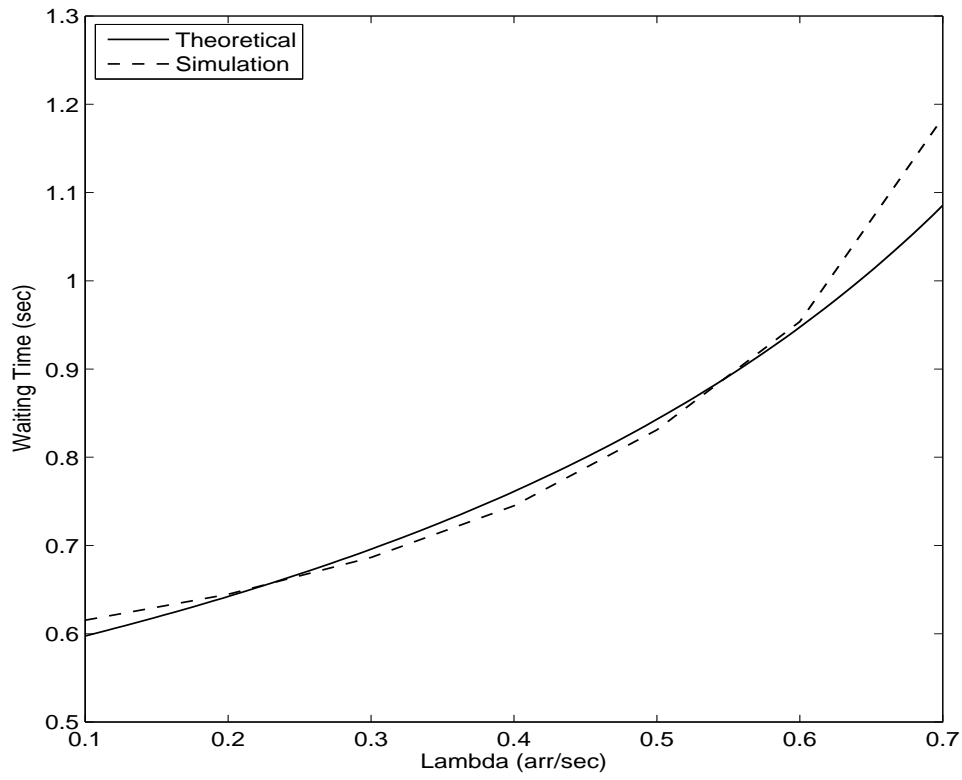


(a) Waiting Time

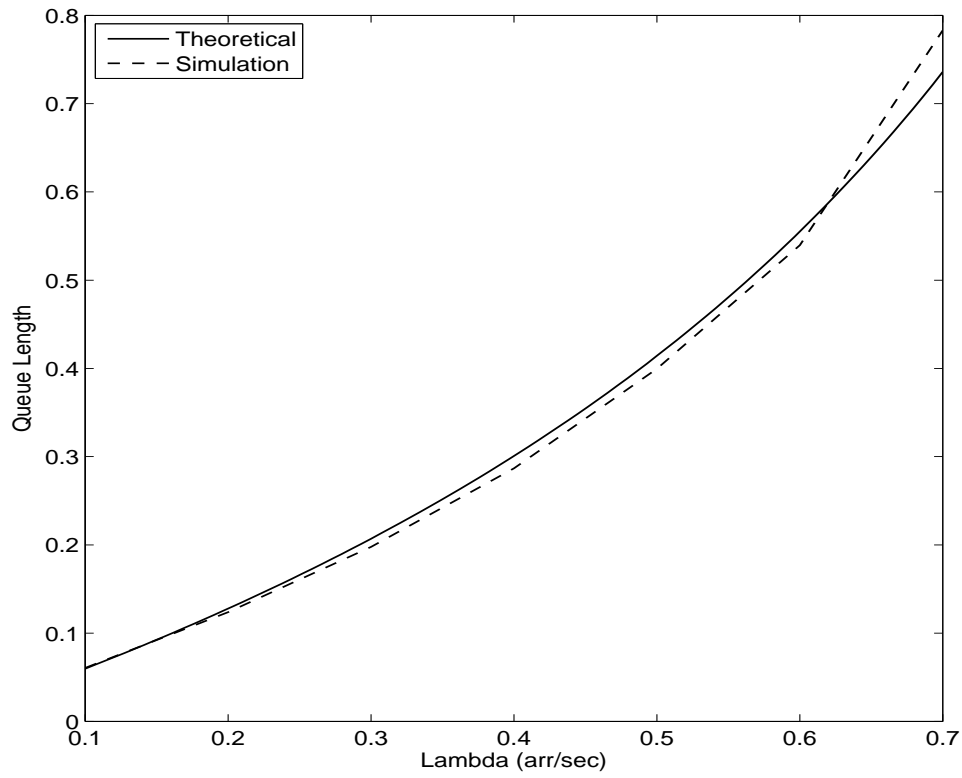


(b) Queue Length

Figure 8.16: ATW Theoretical and Simulation Comparison: Large Frame Sizes

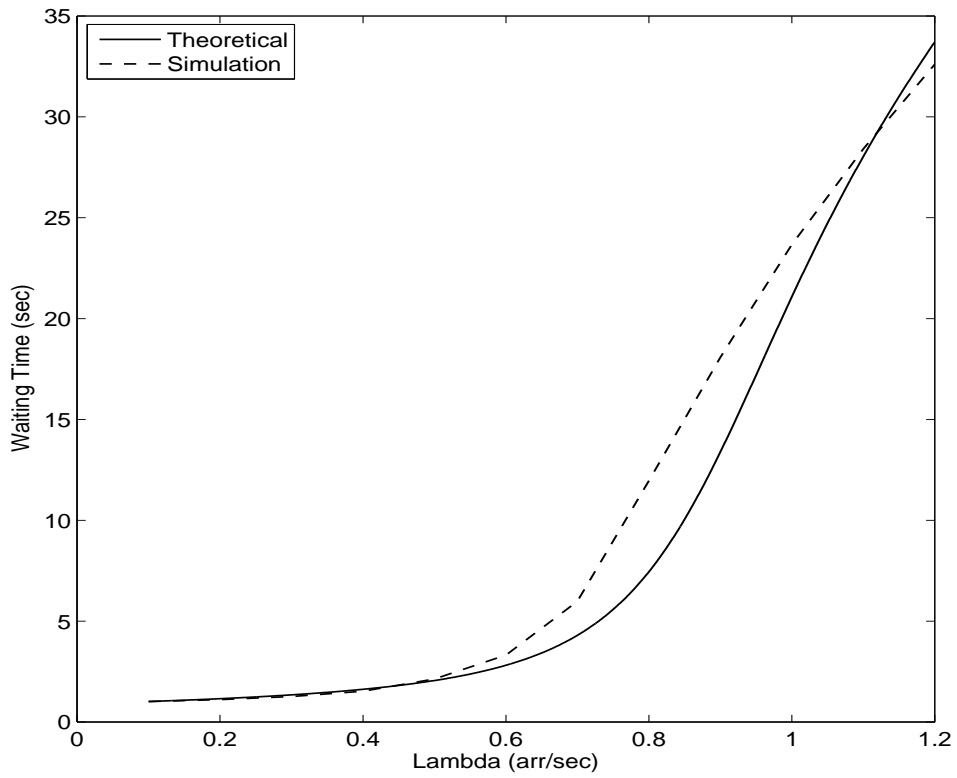


(a) Waiting Time

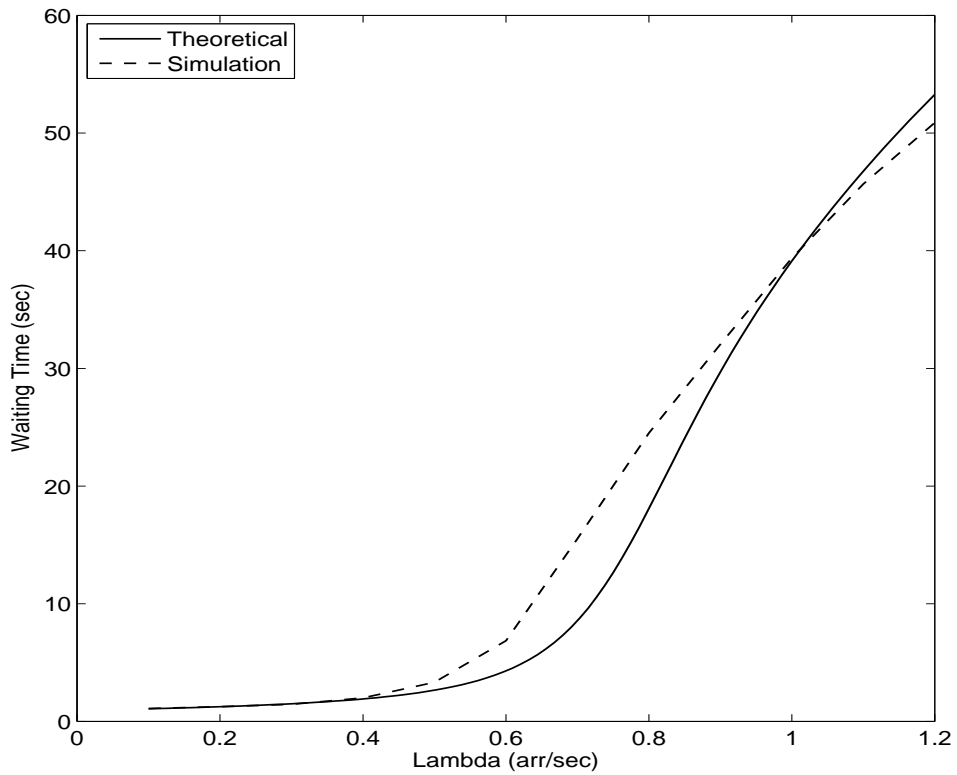


(b) Queue Length

Figure 8.17: ATW Theoretical and Simulation Comparison: Small Frame Sizes



(a) Waiting Time with Data Indication Request and Response Frames' Bytes = 10



(b) Waiting Time with Data Indication Request and Response Frames' Bytes = 20

Figure 8.18: ATW Theoretical and Simulation Comparison: Different Poll Frame Sizes

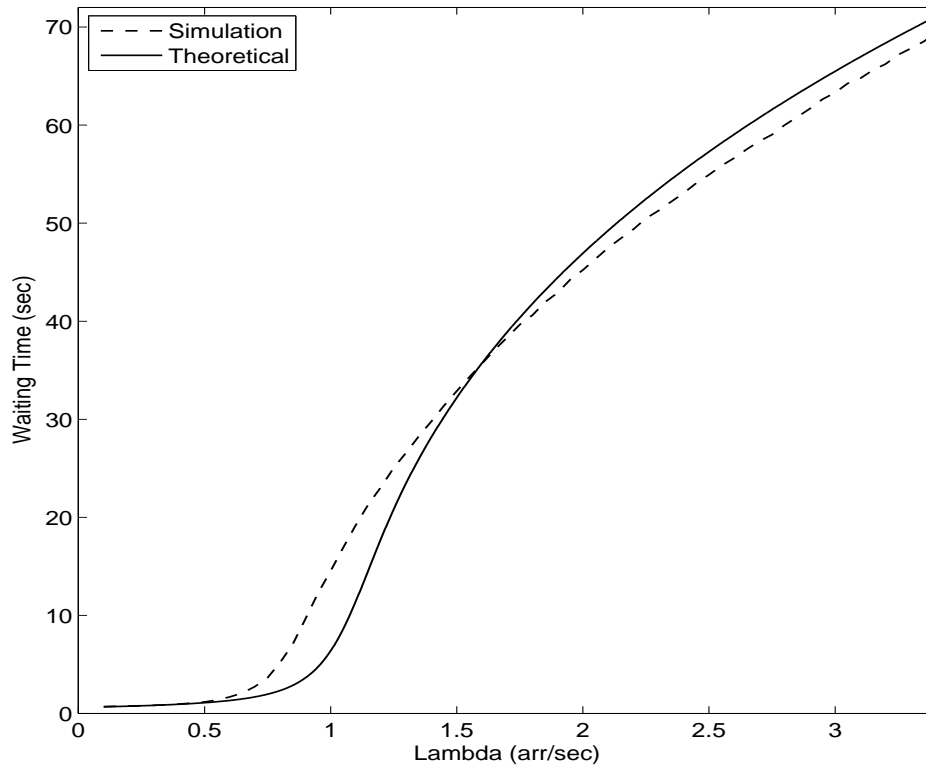


Figure 8.19: ATW Theoretical and Simulation Comparison: Stabilizing Effect for High Arrival Rate, Small Data Frame Sizes and Data Indication Request and Response Frames' Bytes = 20

## 8.6 Summary

The ATW protocol has been investigated as a possible strategy to combine the best performance characteristics of the CSMA and RRP protocols. The protocol is not known to be used within narrow band telemetry networks and is usually applied within a slotted timebase. The definition of the protocol as commonly found has been altered to follow an unslotted methodology. Simulation and theoretical models have been constructed to provide an analysis of the protocol performance. The various types of frame used within the protocol have been described, together with the transmission time for each. The theoretical model is based on a Markovian process State Space Model approach. The simulation model makes use of three discrete event processes as explained with UML activity diagrams. The results from the simulation and theoretical models compare very closely within similar parameter sets. While these results supported initial assumptions, the protocol was further optimized by defining an optimal tree level, thereby reducing the number of stations responding to the initial data indication poll. This optimisation was a success, with general performance improvement, but more so at higher event arrival rates.

A summarised comparison between the different strategies examined, will be presented in Chapter 9.

## Chapter 9

# Comparison of CSMA, RRP and ATW protocols

### 9.1 Overview

In this chapter the performance of each of the different protocols investigated, will be compared and discussed. The comparison has to be seen against the particular area of application for the strategies. The discussion is centred on the results from the different simulations as presented. These are acceptably close to the theoretical results, so the ensuing comparison does not explicitly distinguish between the two. Various optimisations, as implemented, are also taken into account. From the analysis thus far and the theory presented, the protocols' performance can be summarised as follows:

- CSMA gives the best performance at low arrival rates but degrades substantially as the rate increases
- RRP has weaker performance at low arrival rates but improves with an increase in arrival rate
- ATW presents a compromise and can be seen as combining the best of CSMA and RRP

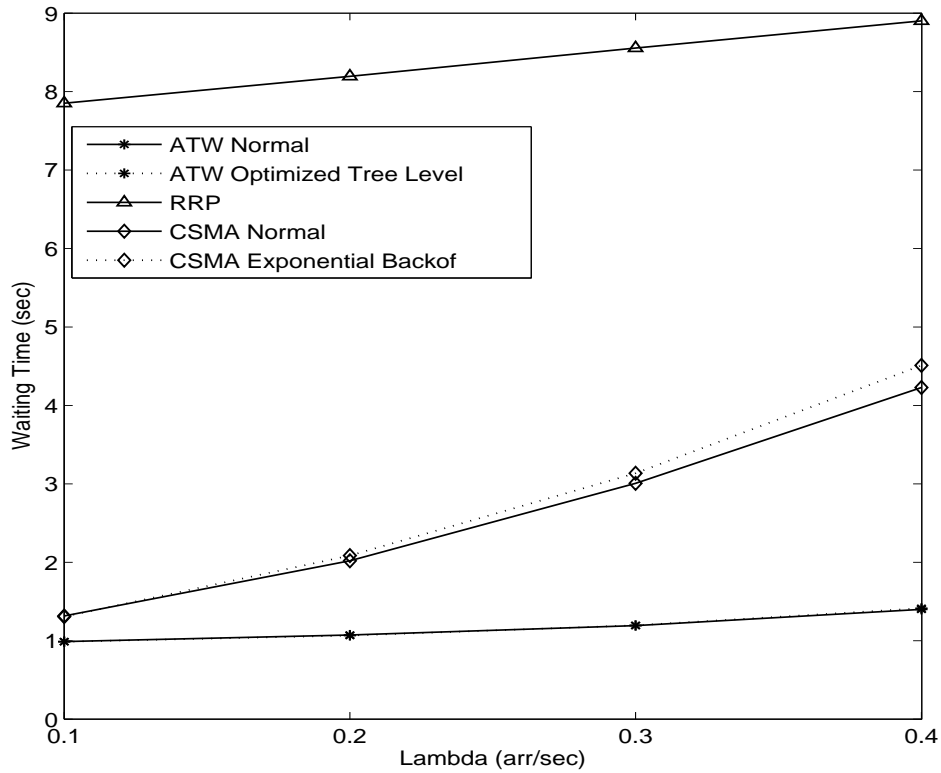
This will be motivated in subsequent sections.

### 9.2 Comparison of Simulation Models

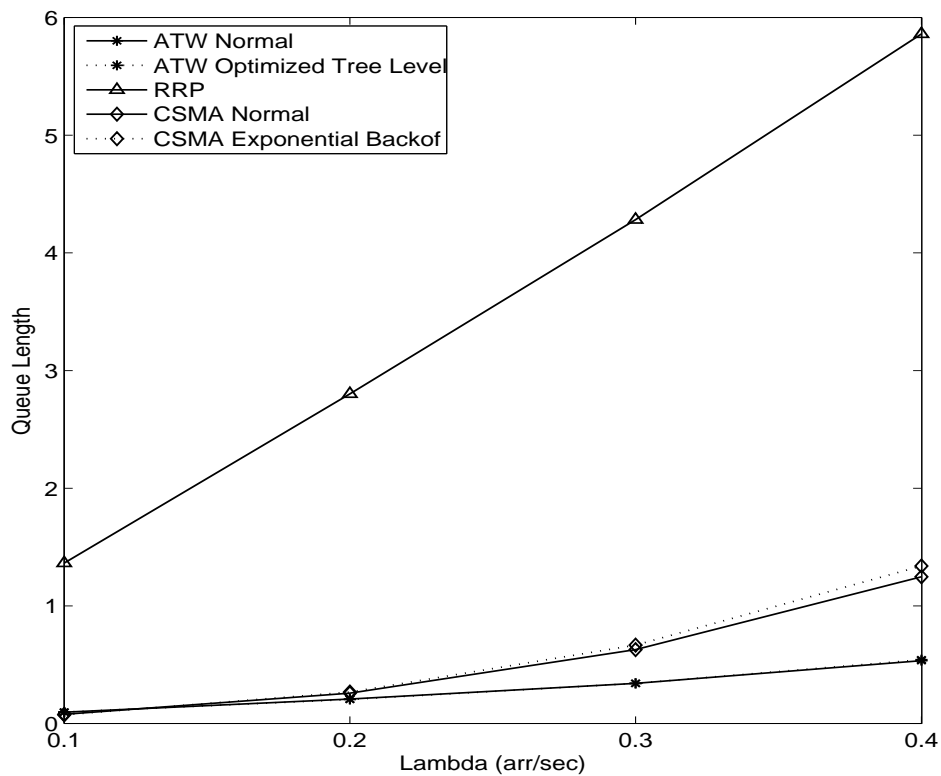
#### 9.2.1 Comparison of Waiting Times and Queue Lengths

The main performance indicators of the different protocols are based on high and low system utilization. The main contributors to the utilization are of course, the arrival rates and data frame sizes used within the network. For the sake of sensible comparison, the same basic set of parameters, e.g. frame sizes, arrival distributions, etc., has been used for all protocols. The exponential backoff optimization for the non-persistent CSMA protocol and the tree level optimization used in the ATW protocol will be included in the comparison. The data indication request and response frame sizes of the ATW protocol has been set to 5 bytes in the comparison figures. Figures 9.1 and 9.2 compare waiting times and queue lengths of the three different protocols for low arrival rates of 0.1 to 0.4 arrivals per second, for large and small frame sizes respectively. As expected, both ATW and CSMA perform better than RRP. ATW performs the best in this case. This can be explained by the high vulnerable period of the CSMA protocol, as three repeaters are used within the setup.

Even with one repeater in the network, the ATW protocol would still do better, due to the fact that backoff and collisions are inevitable in the CSMA protocol. The same applies to smaller frames, but if the number of repeaters in the system is reduced to one in this case and the CSMA protocol backoff range reduced to 1250-2500ms with a timeout period of 1000ms, the performance of the CSMA protocol increases dramatically and outperforms ATW. This is shown in Figure 9.3.

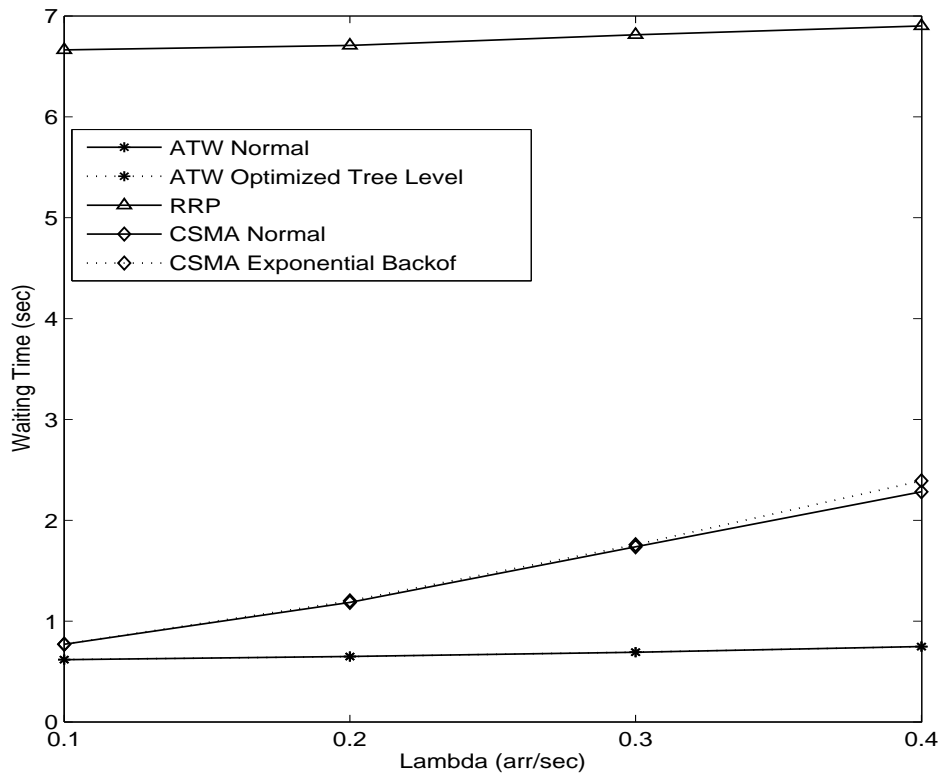


(a) Waiting Time

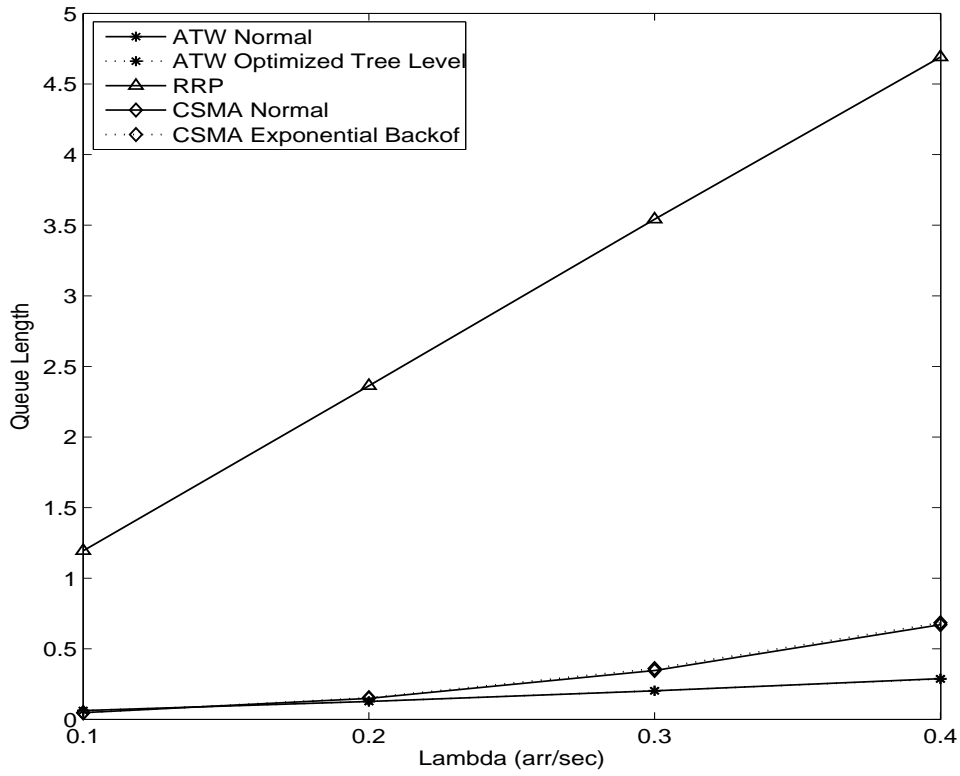


(b) Queue Length

Figure 9.1: Comparison: Low Arrival Rates and Large Frame Sizes



(a) Waiting Time



(b) Queue Length

Figure 9.2: Comparison: Low Arrival Rates and Small Frame Sizes

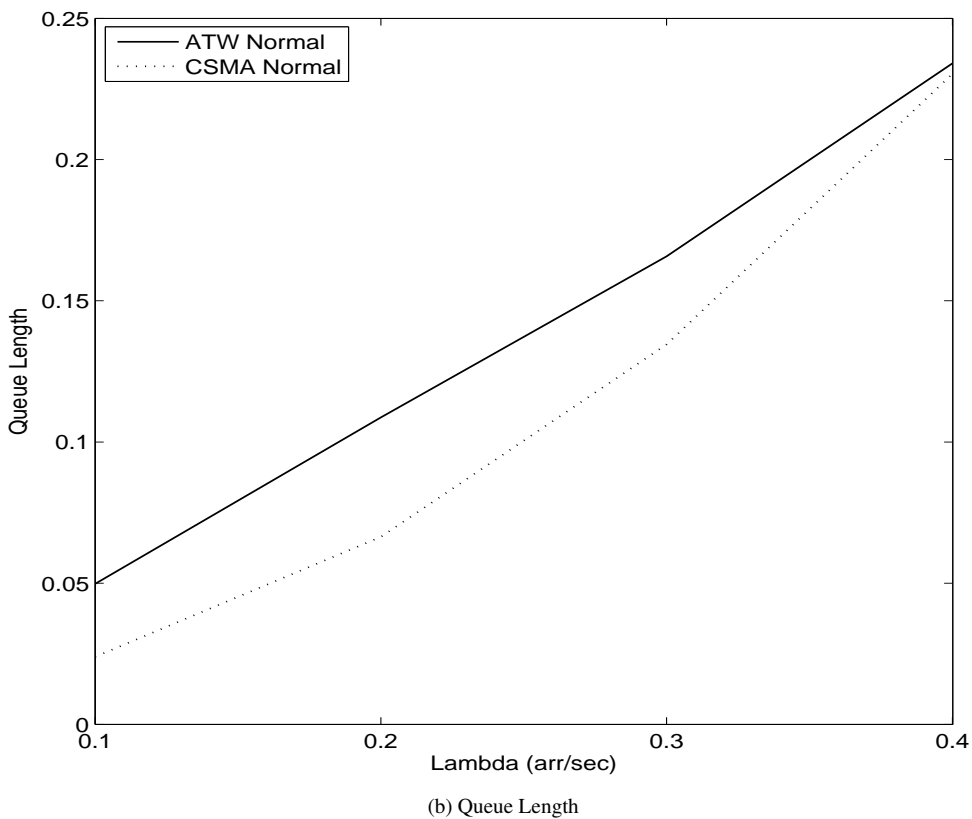
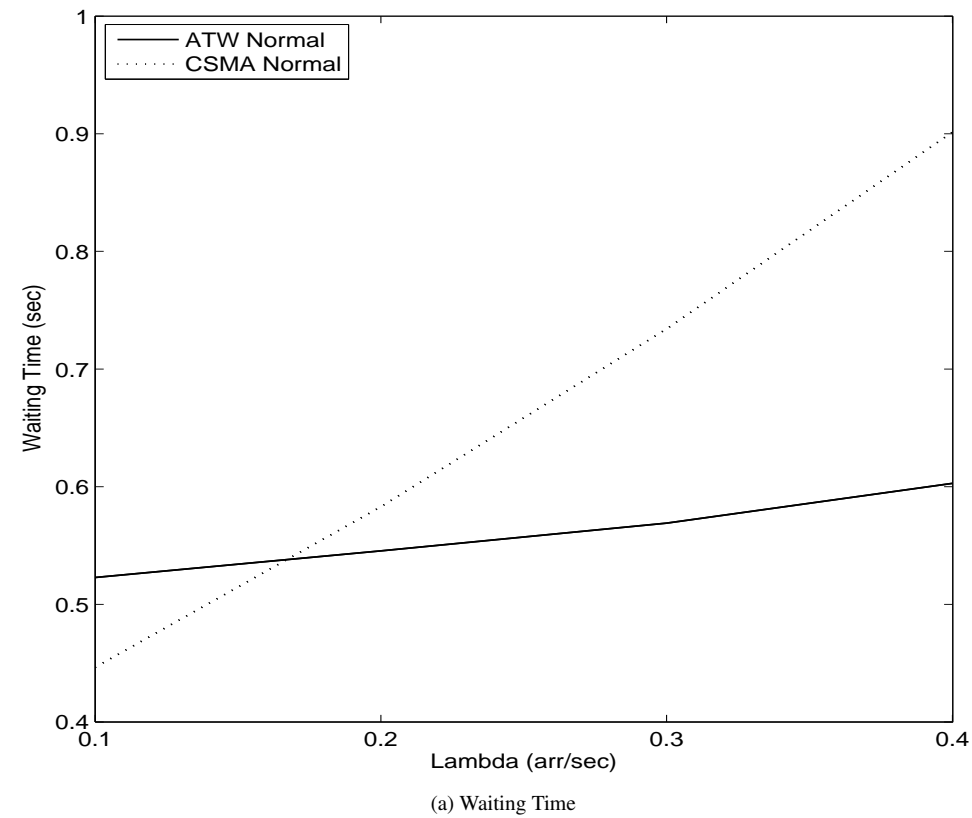
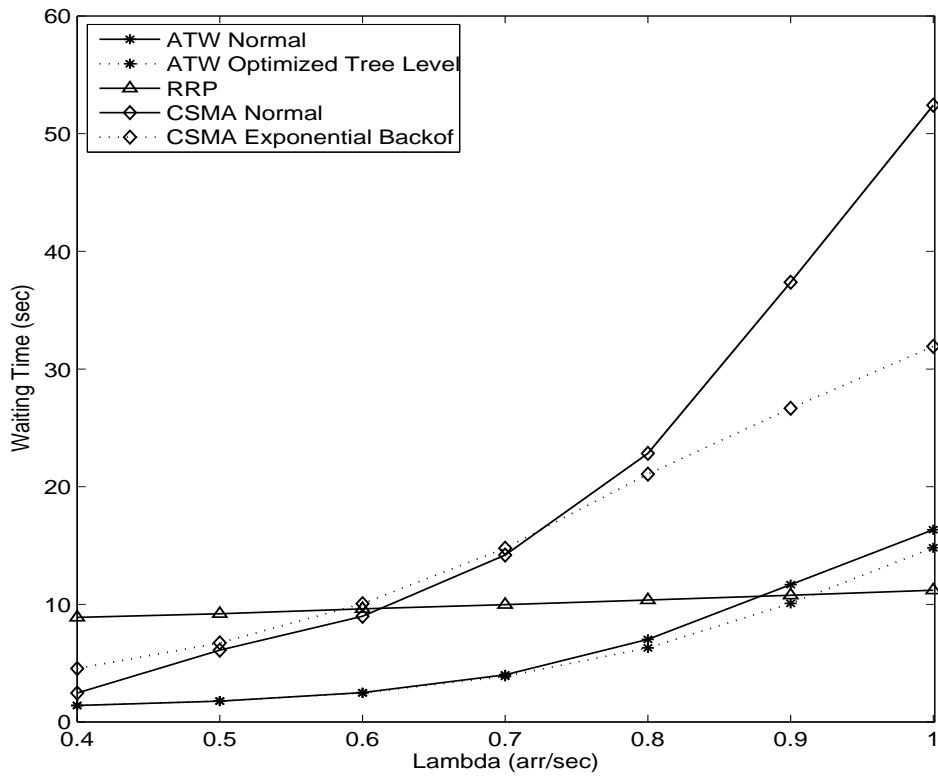


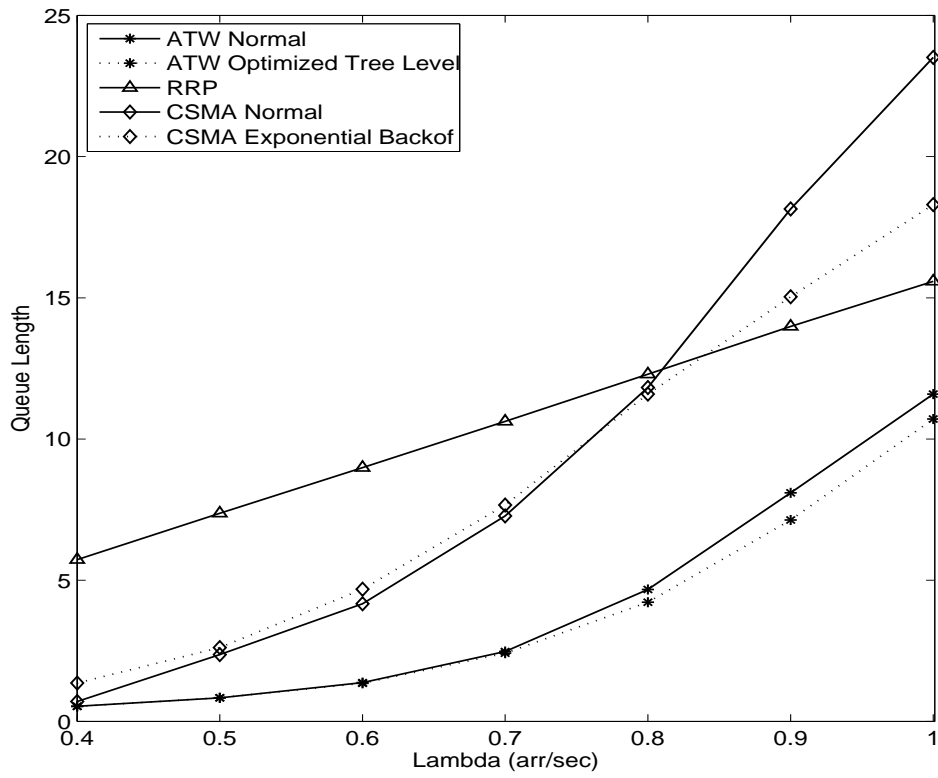
Figure 9.3: Comparison: ATW vs CSMA for Low Data Latencies and Small Frame Sizes

Figures 9.4 and 9.5 present the comparison of the three different protocols for an arrival rate of 0.4 to 1 arrivals per second. For larger frame sizes it can be seen that the performance of CSMA deteriorates quickly. The ATW performance is also decreased but less so than for CSMA. As expected, RRP is the better option with the increase in arrival rate. Even with smaller frame sizes, RRP begins to outperform CSMA, but both are still significantly outperformed by ATW.



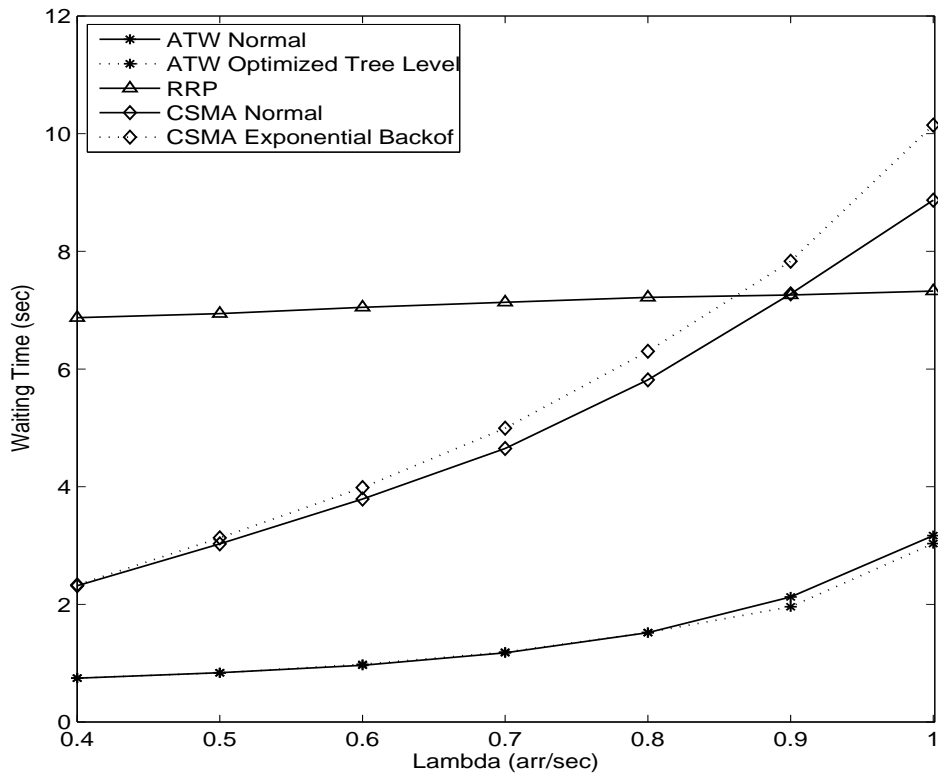


(a) Waiting Time

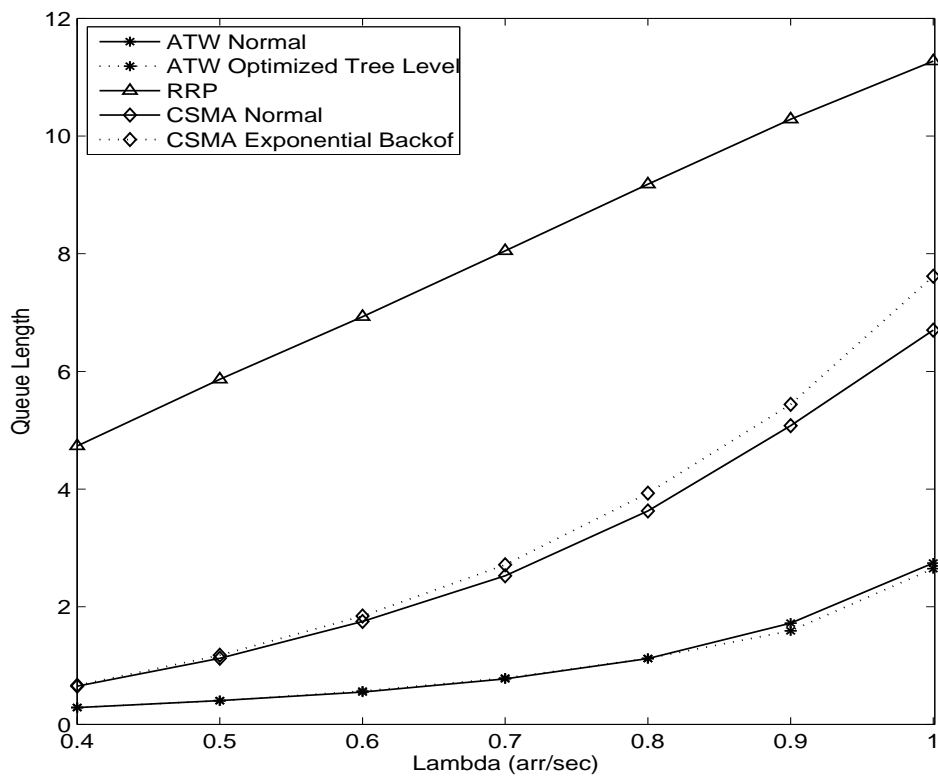


(b) Queue Length

Figure 9.4: Comparison: High Arrival Rates and Large Frame Sizes



(a) Waiting Time

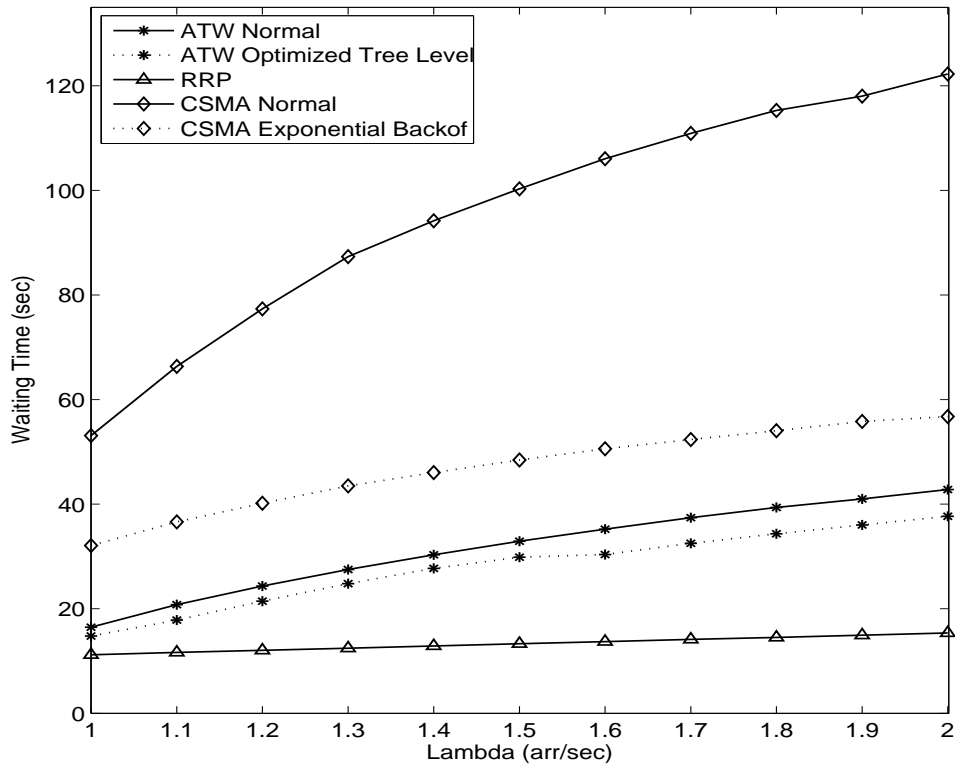


(b) Queue Length

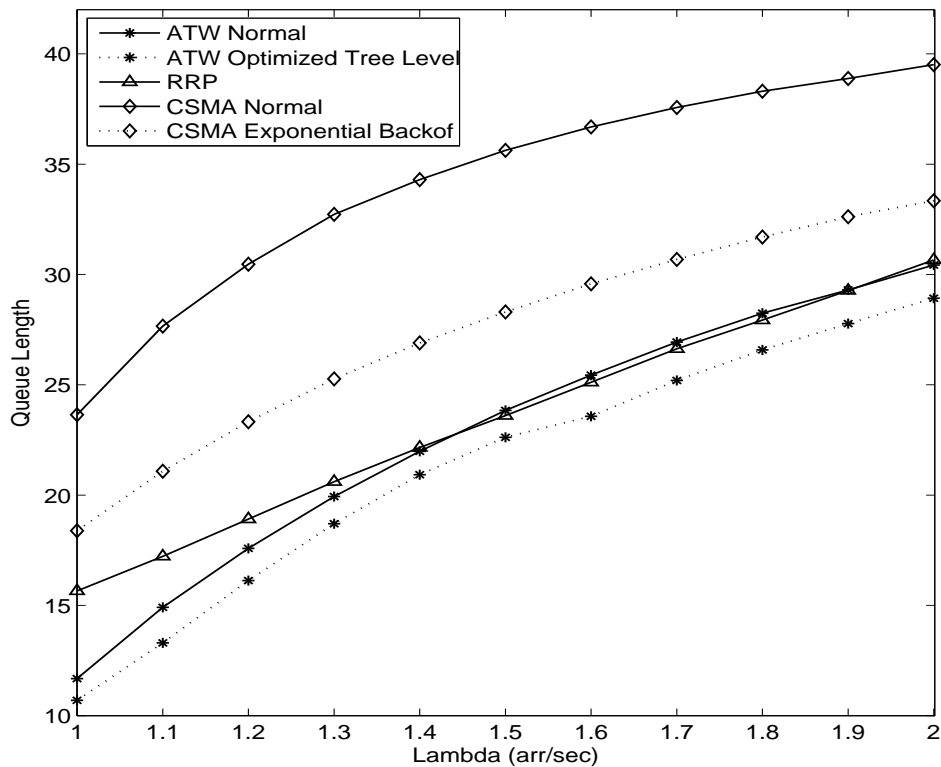
Figure 9.5: Comparison: High Arrival Rates and Small Frame Sizes

Figures 9.6 and 9.7 present the performance differences at very high arrival rates of 1 to 2 arrivals per second. With large frame sizes, CSMA is rendered completely ineffective and ATW struggles, whereas RRP's performance improves as expected and is now by far the best. With smaller frame sizes CSMA still performs worse than the other two, and at rates of 1 to 1.25 arrivals per second, ATW still outperforms the other two. Above that, RRP is again the best choice. It

might be expected that with ATW's optimized tree level it should perform just slightly weaker than RRP. The problem is that the scheme is implemented so that only stations which have arrived with events during the previous polling cycle of the binary tree, can be polled during the next polling cycle. Any events arriving during the active polling cycle will have to wait until the next cycle. If the protocol is modified to allow stations to be taken into consideration within the polling cycle during which they arrive, the performance should improve to almost that of RRP.

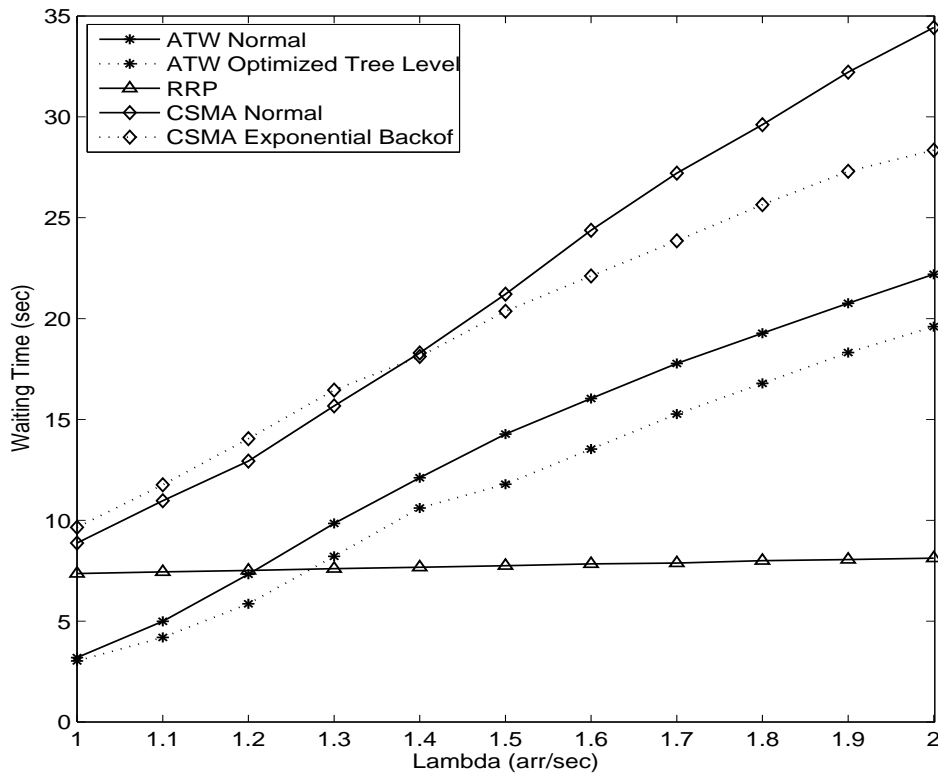


(a) Waiting Time

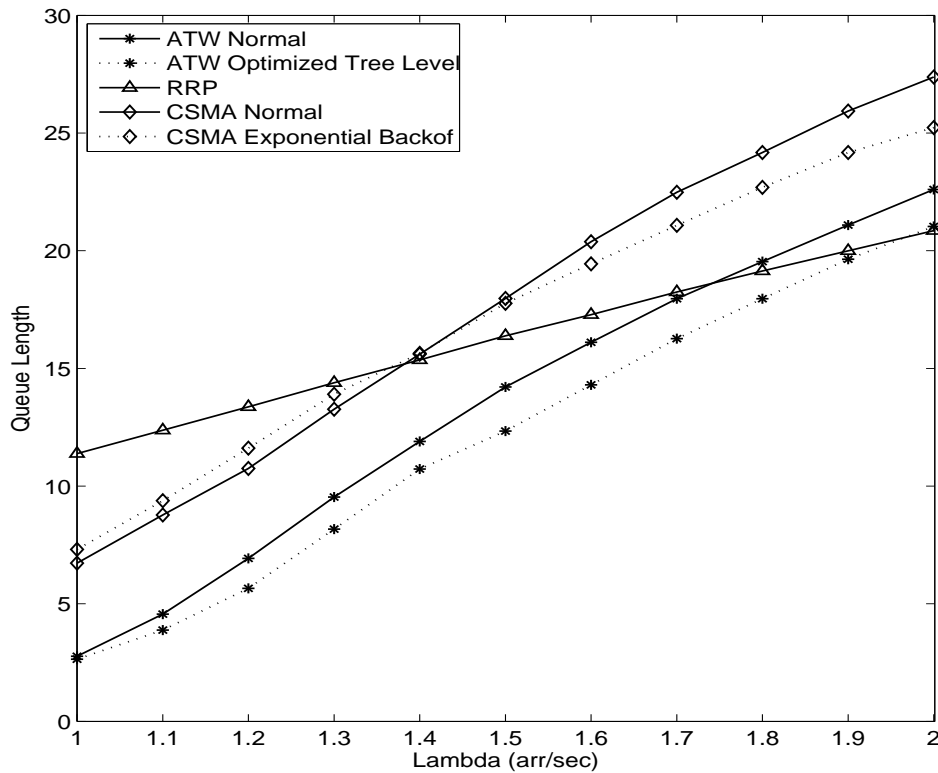


(b) Queue Length

Figure 9.6: Comparison Results: Very High Arrival Rates and Large Frame Sizes



(a) Waiting Time



(b) Queue Length

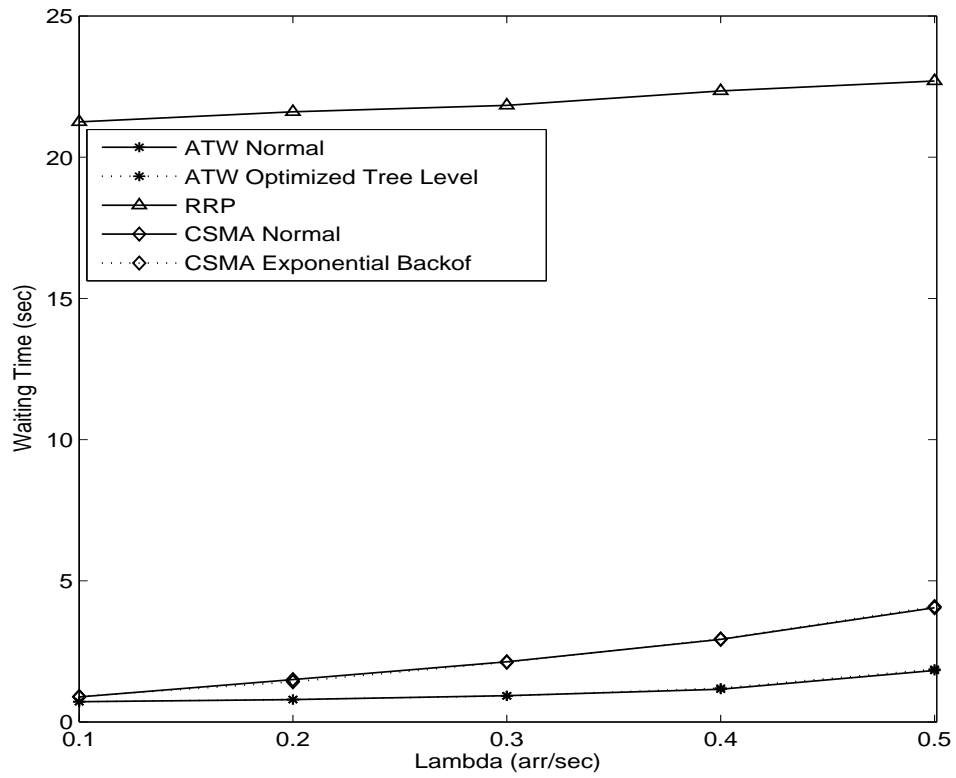
Figure 9.7: Comparison Results: Very High Arrival Rates and Small Frame Sizes

As discussed in Chapter 2, the extent of the Namib water supply scheme will be increased to include approximately 150 stations. Four repeaters will be included and an average arrival rate of close to 0.5 arrivals per second is expected. The data bytes to be transmitted during each transmission should range from 37 to 77 bytes. In order to test the validity of the work done so far and to obtain realistic planning information for the enlarged system, a simulation was run using the

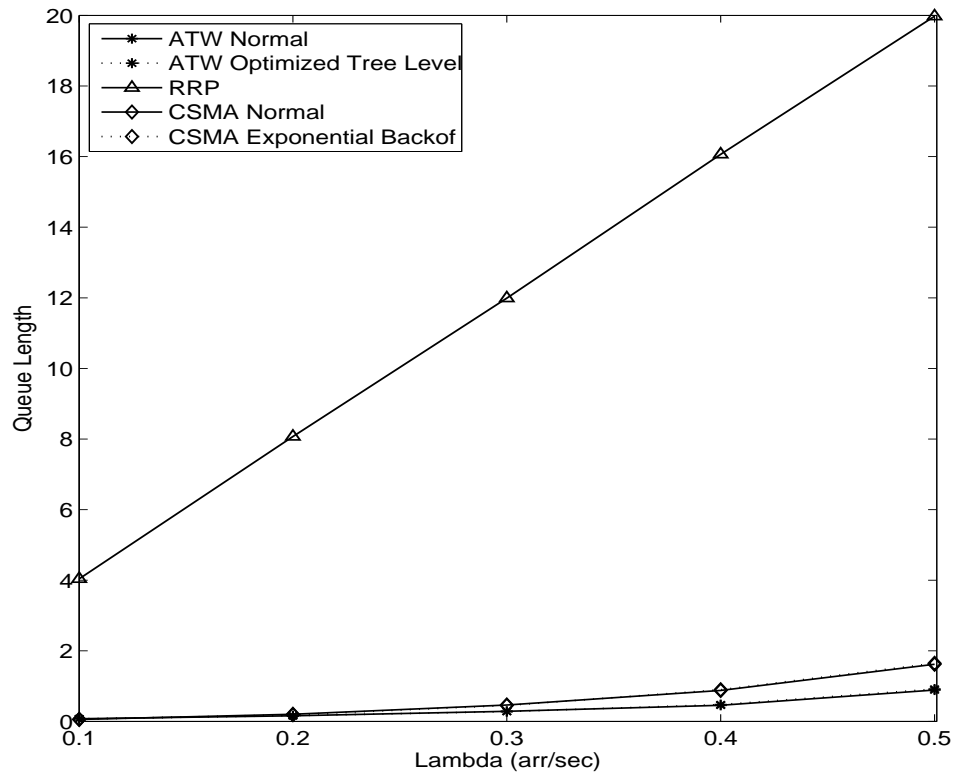
parameter set as per Table 9.1. From these results it is clear that the performance of the ATW protocol is clearly better than the other two, resulting in shorter waiting times and queue lengths at both low and high arrival rates. RRP is very ineffective at these rates, due to high station and repeater overhead, which has the effect of increasing the length of each polling cycle and therefore also the waiting time and queue length.

Parameters	CSMA	RRP	ATW
Number of Stations	150	150	150
Channel Capacity (bps)	4800	4800	4800
Information Bytes	13	13	13
Data Bytes Minimum	37	37	37
Data Bytes Maximum	77	77	77
Preamble (ms)	1	1	1
Postamble (ms)	1	1	1
Min Propagation Distance (km)	30	30	30
Max Propagation Distance (km)	80	80	80
Number of Repeaters	4	4	4
RXEnd (ms)	30	30	30
Turn Around (ms)	1	1	1
Step Size	0.1	0.1	0.1
Time Out (ms)	1500	1500	1500
Data indication frames' bytes	-	-	5
Data Latency (ms)	11	11	11
Begin Arrival Rate (arr/sec)	0.1	0.1	0.1
End Arrival Rate (arr/sec)	0.5	0.5	0.1
Simulation Time (sec)	99000	99000	99000
Backoff Minimum (ms)	2500	-	-
Backoff Maximum (ms)	5000	-	-

Table 9.1: Parameter Set for Practical System



(a) Waiting Time

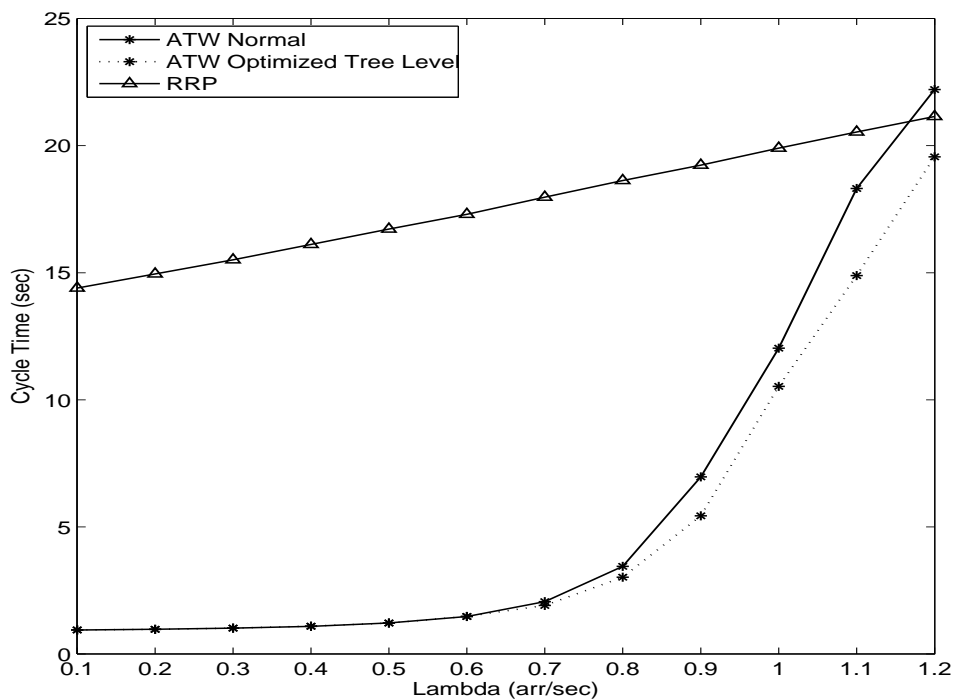


(b) Queue Length

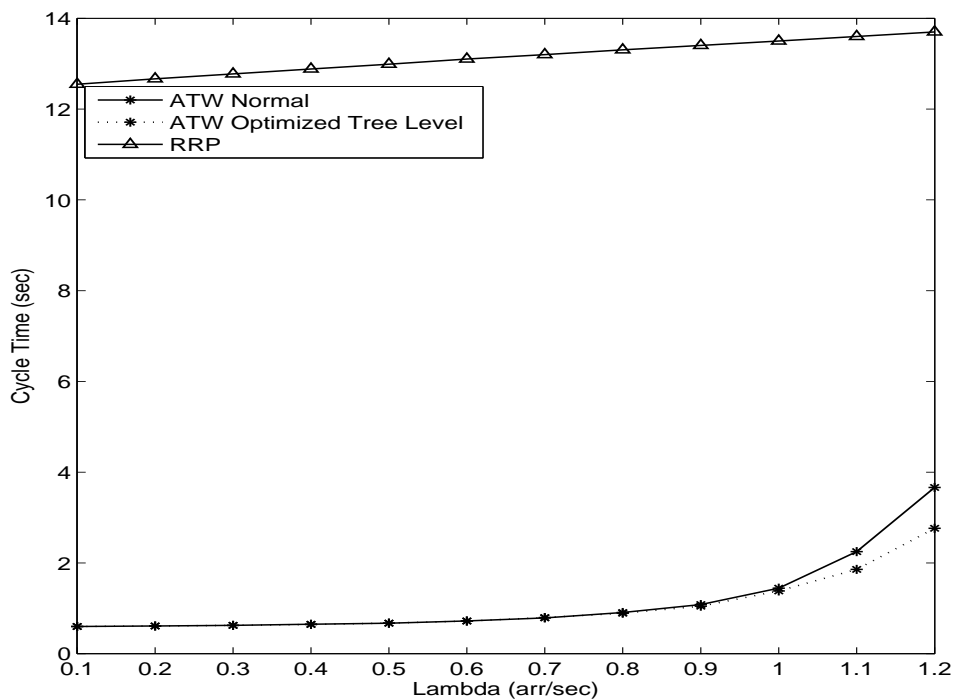
Figure 9.8: Comparison: Expanded Namib Supply Scheme

## 9.2.2 Cycle Times Comparison for the RRP and ATW Protocols

For the sake of interest, the cycle times for RRP and ATW were also compared for both large and small frame sizes, but with an increased arrival rate of 1.2. The results are presented in Figure 9.9. It can clearly be seen that the initial assumption is correct, as also confirmed in the previous section, that the ATW protocol outperforms RRP for lower arrival rates. As the arrival rate increases, the performance of ATW deteriorates and that of RRP improves. With small frame sizes, however, the rate of deterioration of ATW is significantly diminished and ATW is only outperformed by RRP at very high arrival rates.



(a) Large Frames



(b) Small Frames

Figure 9.9: Comparison: ATW vs RRP Cycle Time

### 9.3 Summary

This chapter has presented a comparison between the performance of different protocols under similar system conditions. Performance is adjudicated on queue lengths and resultant wait times, these being the most important metric for this type of application. The results prove that RRP mostly outperforms the ATW and CSMA schemes by far, at very high arrival rates. It is also shown that at low and high arrival rates, with smaller frames, ATW performs best. With larger frames and a high arrival rate, the performance of ATW and RRP converges at the higher rates with RRP performing slightly better. It has also been determined that the high data latency of the system caused by the number of repeaters in the network reduces the performance of CSMA. Reducing the chances of backoffs and collisions, by reducing the data latency (one repeater in the system) and employing small frame sizes, CSMA will outperform both ATW and RRP. At very high arrival rates, the performance of CSMA degrades completely, but ATW still performs better than RRP up to a point where it degrades to a waiting time twice as high as that of RRP. It has also been explained that if the ATW protocol is optimized by allowing stations to complete within the current polling cycle of the adaptive tree, its performance will further improve. If the optimized tree level is also implemented, the performance will be slightly below that of RRP at higher arrival rates. Finally, with an increase of stations and average arrival rate the RRP performance decreases significantly and CSMA and ATW have the better performance while ATW further outperforms CSMA.

The practicality of the set of tools developed has also been proven, by running simulations to predict the performance of a planned, expanded practical system. The work done under this project will be summarised and concluded in the final chapter, Chapter 10.



## Chapter 10

# Summary and Conclusion

### 10.1 Summary

The main objective of this research project was to develop a set of theoretical and simulation tools to analyse and predict the performance of the type of protocol used for telemetry networks. This was done in an attempt to identify the type of approach that might provide better performance at both low and high channel utilisation.

The investigation was centred on 3 protocols, i.e. Non-persistent CSMA, Round Robin Polling and Adaptive Tree Walk. In the case of CSMA, a more standard version was used instead of the commercially available version currently implemented in some of NamWater's schemes. This particular example was found to be overly complex. The ATW strategy as modelled, was developed for unslotted use, from the standard slotted approach.

In all 3 cases theoretical modelling was based on Markov chain derived queueing theory, using an expanded state space approach. The theoretical modelling was followed by creating protocol simulation models under DESMO-J. The simulation models for both CSMA and ATW were further optimised by backoff modification and tree level optimising, respectively. It was found that the theoretical and simulation results track very well for realistic system parameters, providing a satisfactory level of confidence in using both as complementary planning tools.

Actual results obtained during the research indicated that the ATW protocol offers an excellent solution, incorporating good characteristics of both CSMA and RRP, for the type of application concerned. This applies to a wide range of network utilisation and seems to be a strong consideration for use in future narrow band telemetry networks.

### 10.2 Contributions and Conclusion

Narrow band telemetry networks play a very important role and are commonly applied in the real-time monitoring of infrastructure such as water, sewage and electrical networks. In spite of this, an "install and see if it works ok" approach is the norm, without too much attention being paid to proper advance modelling, or protocol optimisation thereof. The latter is really fundamental to adequate system performance and optimal use of available communications bandwidth. The work completed under this project sought to establish a more deterministic approach in order to facilitate planning and performance prediction, for networks utilising different, commonly applied protocols. To this end, successful outcomes were obtained and is briefly summarised under the subheadings hereunder.

### 10.2.1 Theoretical Models Based on Queueing Theory

Theoretical models based on Queueing Theory were derived for 3 protocols, i.e.:

- Non-Persistent CSMA
- Round Robin Cyclical Polling
- Unslotted Adaptive Tree Walk

The models were developed using a comprehensive State Space approach and took specific operational conditions for the type of network concerned into account, such as timeouts, backoffs and error occurrence. This is a considerable expansion on the type of general analysis to be found in references. As such, they offer an accurate set of deterministic planning tools.

### 10.2.2 Verifying the Accuracy of the Theoretical Models

In order to verify the accuracy of the theoretical models, extensive simulations were created for the protocols under DESMO-J, a Java based package. The routines were developed in such a way that errors, delays and other protocol system parameters could easily be selected and altered. Very good correlation was obtained between theory and simulations. The simulation routines present another set of performance prediction tools. As a trial, it was found to be practical and useful in application to a real life network under consideration for expansion.

### 10.2.3 ATW Protocols

ATW protocols are, of course, well known, but the unslotted ATW protocol developed during this work has characteristics specifically aimed at the type of network in question. It is, as far as can be determined, a first time approach for this type of telemetry network and seems to hold much promise in improving system performance under appropriate conditions.

In summary, it can be said that considerable attention has been given to an important, but much neglected, area of practical protocol implementation. This resulted in the establishment of a useful theoretical base as well as a set of accompanying simulation routines.

## 10.3 Future Work and Recommendations

In order to further capitalise on the work as documented, the following is recommended in terms of future trials and developments:

- As a consequence of the very promising results obtained by the modified ATW protocol, it seems logical to evaluate a practical implementation thereof on one of NamWater's networks. A retrofit is always time consuming, but in this case it might be thoroughly worthwhile and financially viable over time. It is therefore recommended that additional research be conducted to substantiate the results obtained from this research.
- It is tempting to build some machine learning capability into the RRP and ATW strategies. It should be considered to implement some form of traffic aware polling variants in both of them respectively. This should be based on continuous statistical analysis of traffic patterns and individual station arrival rates, in order to bias service to some nodes and to improve general channel efficiency. This is not a field that has received much attention and will fit

in well with the set of tools developed under this project. It is therefore recommended that further research be conducted with the aim to improve current practises and to contribute towards the advancement of intelligence within this domain.

# Bibliography

- [1] D. Bailey, *Practical radio engineering and telemetry for industry*. Newnes, 2003.
- [2] J. Burdick, "The moore-penrose pseudo inverse," <http://robotics.caltech.edu/~jwb/courses/ME115/handouts/pseudo.pdf>, n.d.
- [3] R. Cadenhead and L. Lemay, *Sams teach yourself Java 6 in 21 days*. Sams Publishing, 2007.
- [4] J. Capetanakis, "Tree algorithms for packet broadcast channels," in *IEEE Trans. on Information Theory*, vol. IT-25, 1979.
- [5] N. Chee-Hock and S. Boon-Hee, *Queuing modeling fundamentals with applications in communication networks, Second edition*. John Wiley & Sons Ltd, 2008.
- [6] P. Cherry, "ProDesign proprietary protocol," ProDesign.
- [7] "Understanding logical link control," [http://www.cisco.com/en/US/tech/tk331/tk336/technologies\\_tech\\_note09186a0080094777.shtml](http://www.cisco.com/en/US/tech/tk331/tk336/technologies_tech_note09186a0080094777.shtml), n.d.
- [8] "A brief history of optics and wave theory," [http://www.phy.davidson.edu/stuhome/cabell\\_f/diffractionfinal/pages/history.htm](http://www.phy.davidson.edu/stuhome/cabell_f/diffractionfinal/pages/history.htm), 2000.
- [9] "Potential of CDMA450 for rural network connectivity," <http://tier.cs.berkeley.edu/docs/wireless/cdma450.pdf>, n.d.
- [10] B. A. Forouzan, *Data communications and networking, Fourth edition*. McGraw Hill, 2007.
- [11] R. L. Freeman, *Reference manual for telecommunications engineering*. Wiley Online Library, 2002.
- [12] L. Garcia and I. Widjaja, *Communication networks, fundamental concepts and key architectures, Second edition*. McGraw Hill, 2004.
- [13] M. T. Goodrich and R. Tamassia, *Data structures and algorithms in Java*. John Wiley & Sons Inc, 2001.
- [14] K. Greene, "Huygens principle/construct" <http://www.math.ubc.ca/~cass/courses/m309-04a/HuyghensOverheads.pdf>, 2004.
- [15] H. Sizon, *Radio wave propagation for telecommunication applications*. Springer, 2003.
- [16] L. Hughes, *Introduction to data communications: A practical approach*. Jones & Bartlett Publishers Inc, 1997.
- [17] L. Kleinrock and F. A. Tobagi, "Packet switching in radio channels: Part i - carrier sense multiple-access modes and their throughput-delay characteristics," in *IEEE Transactions on Communications*, vol. COM-23, December 1975.
- [18] D. E. Knuth, *The art of computer programming, Third edition*. Addison Wesley Longman, 1997.
- [20] D. C. Lay, *Linear algebra and its applications, Second Edition*. Addison Wesley Long, 2000.

- [21] J. Liebeherr, "Data link layer lecture," <http://www.cs.virginia.edu/~zaher/classes/CS457/lectures/flow-control.pdf>, 1998.
- [22] A. Maria, "Introduction to modeling and simulation," in Winter Simulation Conference, 1997.
- [23] E. McCune, *Practical digital wireless signals*. Cambridge University Press, 2010.
- [24] B. McLarnon, *Vhf/uhf/microwave radio propagation: A primer for digital experimenters*. ICTP-URSI School on digital and multimedia communications using terrestrial and satellite radio links, 2001.
- [25] P. Nielsen, M. White, and U. Parui, *Microsoft SQL server 2008*. Wiley Publishing, Inc, 2008.
- [26] B. Page and W. Kreutzer, *The Java simulation handbook*. Shaker Verlag Aachen, 2005.
- [27] P. Z. Peebles, Probability, *Random variables and random signal principles. Fourth edition*. McGraw-Hill, 2001.
- [28] L. L. Peterson and B. S. Davie, *Computer networks: A systems approach, Third edition*. Morgan Kaufman, 2003.
- [29] D. M. Pozar, *Microwave and RF design of wireless systems*. John Wiley & Sons, 2001.
- [30] R. Wolhuter and G. van Rooyen, "Elements of telecommunications systems design and teletrac analysis," n.d.
- [31] *South African band re-planning exercise, band plan for the frequencies 20MHz to 3GHz, version 2*. South Africa: Department of Communications, 1997.
- [32] C. Rossouw, "The design of low cost ad-hoc network for short distance data acquisition," Master's thesis, University of Stellenbosch, Department of Electrical and Electronic Engineering, Stellenbosch, 2008.
- [33] J. S. Seybold, *Introduction to RF propagation*. John Wiley & Sons, Inc, 2005.
- [34] W. Stallings, *Data and computer communications, Fifth edition*. Prentice-Hall, 1997.
- [35] W. Stallings, *Data and computer communication, Eighth edition*. Pearson Prentice Hall, 2007.
- [36] J. Stewart, *Calculus, Fourth edition*. Brooks/Cole, 1999.
- [37] A. Tanenbaum, *Computer Networks, Fourth edition*. Prentice-Hall, 2002.
- [38] "Wireless networking in the developing world," <http://wndw.net/>, 2006.
- [39] S. Winder and J. Carr, *Newnes radio and RF engineering pocket book, Third Edition*. Newnes, 2002.
- [40] R. Wolhuter, "The Determining of Optimum Protocol Stratageies of Optimum Protocol Strategies for Half-Duplex Telemetry Communication Links," Ph.D. dissertation, University of Stellenbosch, Department of Electrical and Electronic Engineering, Stellenbosch, 2002.
- [41] R. Ziemer and W. Tranter, *Principles of communications, Fifth edition*. John Wiley & Sons Inc, 2002.
- [42] "Linear congruential generator," [http://en.wikipedia.org/wiki/Linear\\_congruential\\_generator](http://en.wikipedia.org/wiki/Linear_congruential_generator), n.d.
- [43] "Chapter 32. ieee 802.2 llc - logical link control layer," [http://www.rigacci.org/docs/biblio/online/intro\\_to\\_networking/c5048.htm](http://www.rigacci.org/docs/biblio/online/intro_to_networking/c5048.htm), n.d.



# Appendix A

## First Appendix

### A.1 ProDesign Protocol Flow Chart

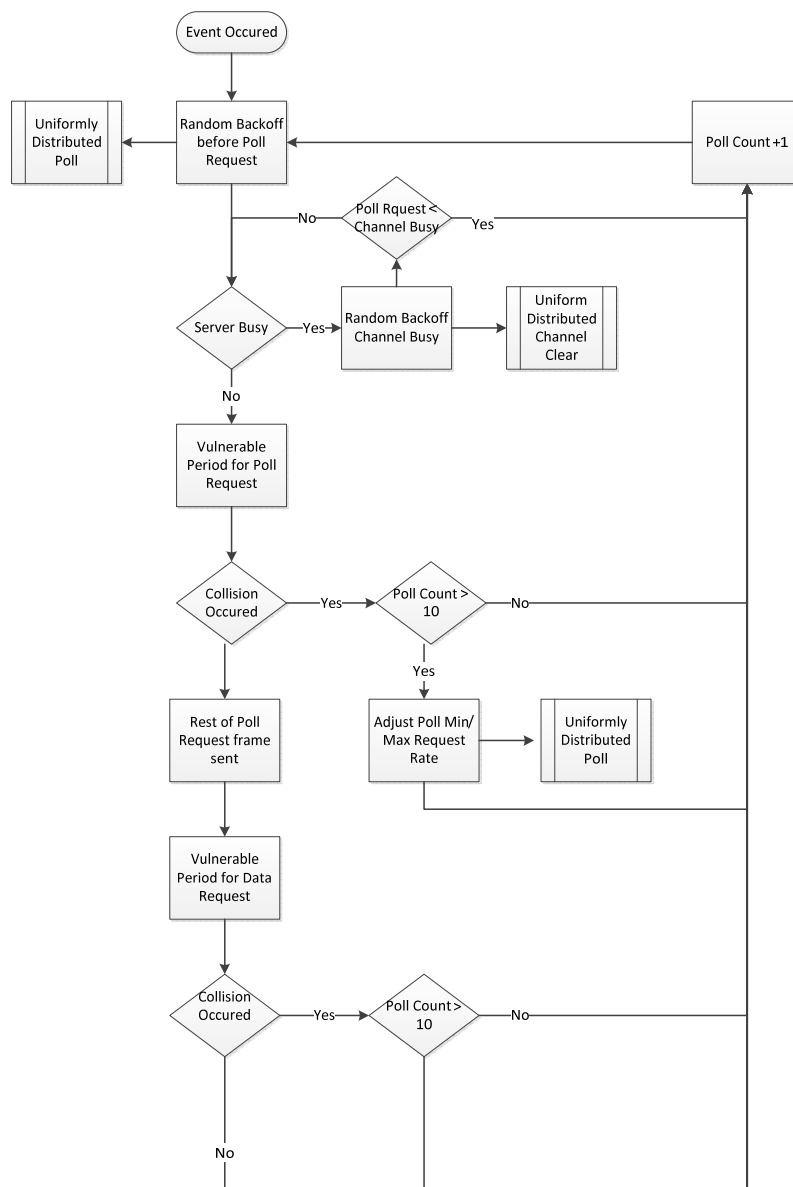


Figure A.1: ProDesign Protocol Flow Chart Part 1

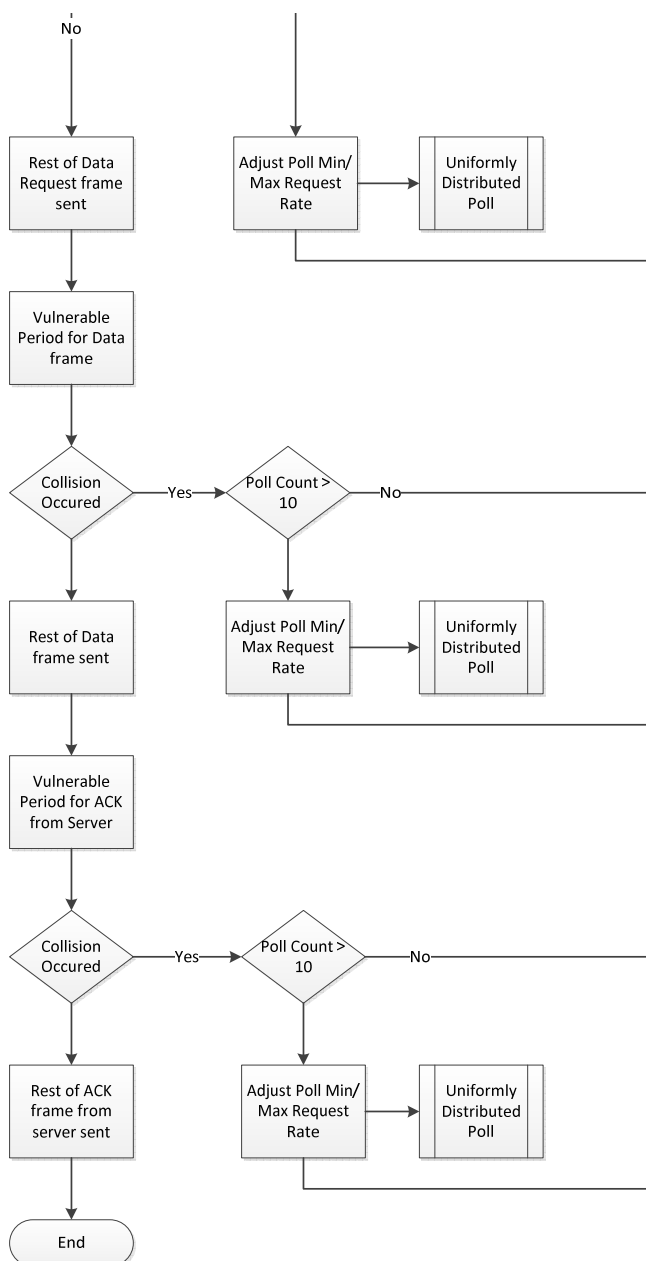


Figure A.2: ProDesign Protocol Flow Chart Part 2

## A.2 Arrival Distribution Graphs

The following arrival distribution graphs have been drawn from the CSMA protocol making use of the parameter set used for large frame sizes. Note that the highest frequency is not centred at the average arrival rate set for the simulation but is lower, due to the fact that after each event occurs the arrival rate decreases and will finally settle at the steady state condition of the system. This can clearly be seen in the figures presented. Note that histograms used have been limited to a certain number of samples and therefore the actual distribution extends beyond the boundaries of the histogram.



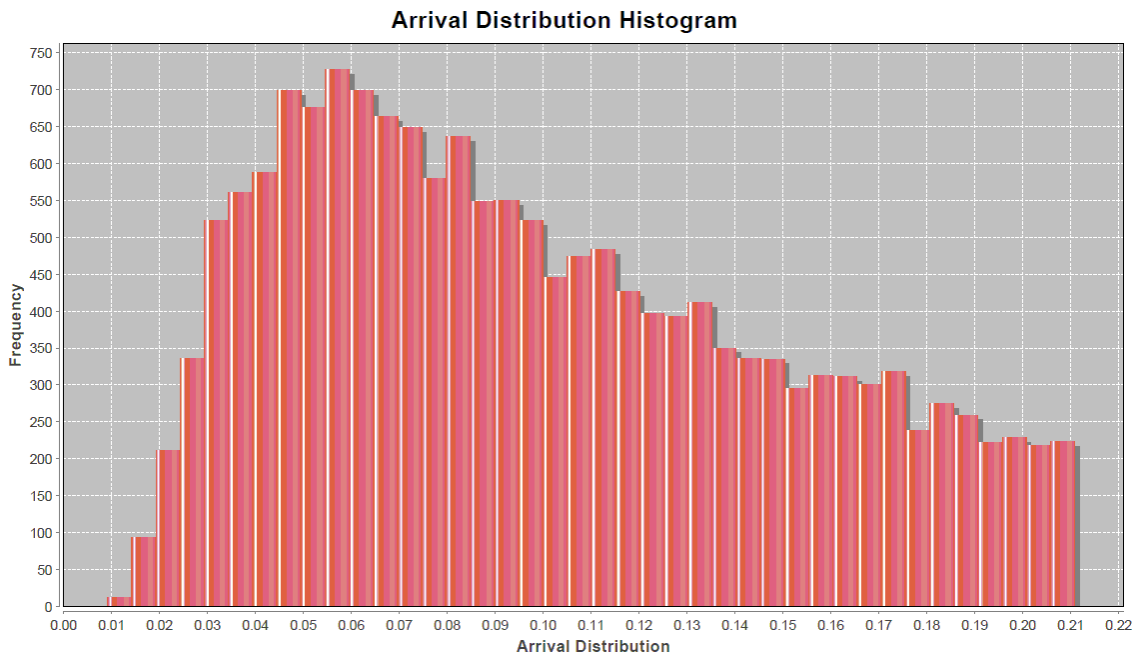


Figure A.3: CSMA Protocol Arrival Rate Histogram with Rate = 0.1

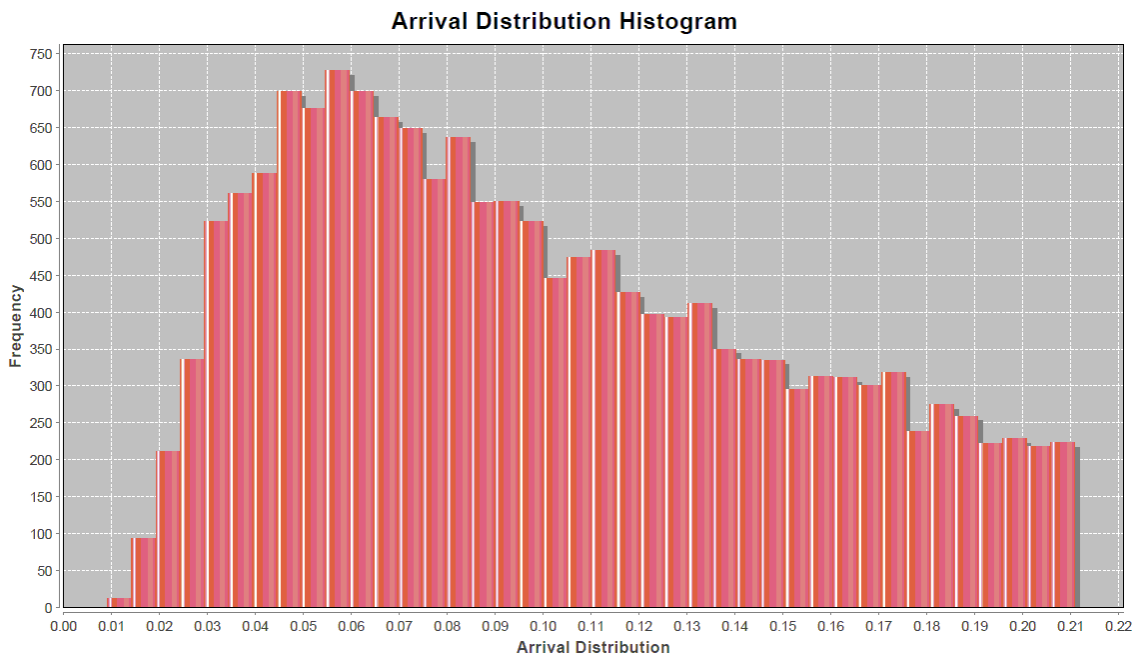


Figure A.4: CSMA Protocol Arrival Rate Histogram with Rate = 0.4

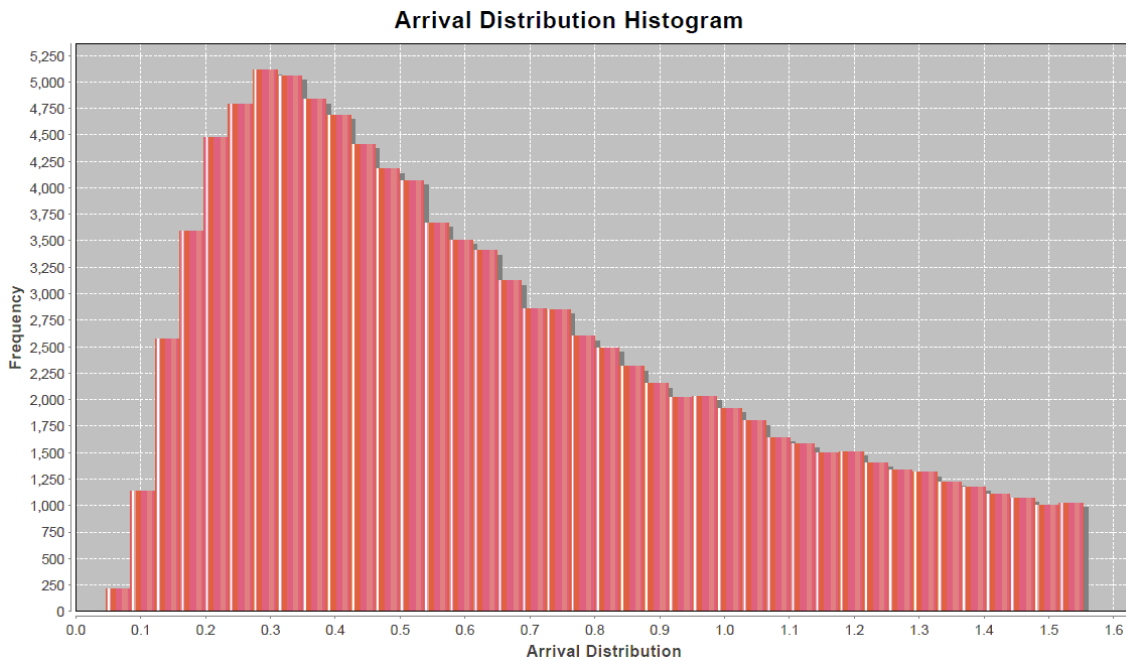


Figure A.5: CSMA Protocol Arrival Rate Histogram with Rate = 0.8

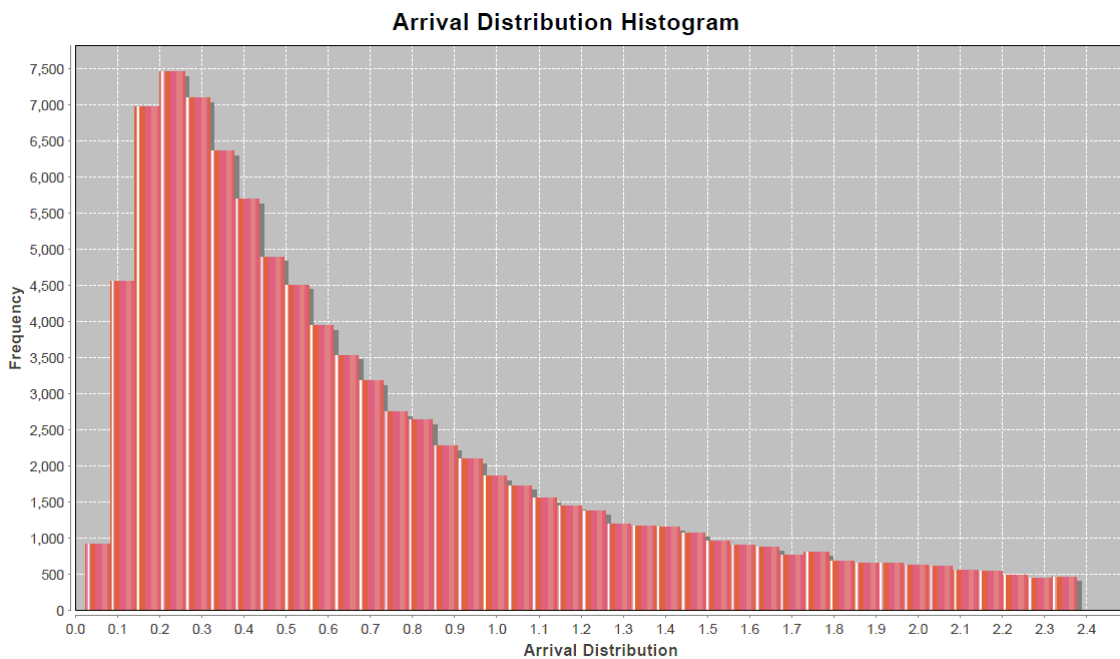


Figure A.6: CSMA Protocol Arrival Rate Histogram with Rate = 1.2

### A.3 Gaussian Distributions

The Central Limit Theorem states that the mean of a large enough sample space of independent random variables, with a finite mean and variance can be approximated by the Gaussian or Normal distribution. This is shown in Figures A.7 to A.10 where the average waiting time, collision time, queue length and number of collisions have been taken for a specific queue length within the simulation period and normalized. Figures A.11 to A.14 show the same graphs as before, but have not been normalized. Figure A.15 shows the waiting times that have been used in the Gaussian graphs previously

mentioned. Different average waiting times will be obtained within a simulation for different queue lengths and these waiting times for the various queue lengths approximate a Gaussian distribution. The same principle applies for the average collision time, backoff time and number of collisions. This proves that the events that take place within the protocol are independent random variables. The same will apply for both the ATW and RRP protocols. An simulation time of 250 000 ms has been used.

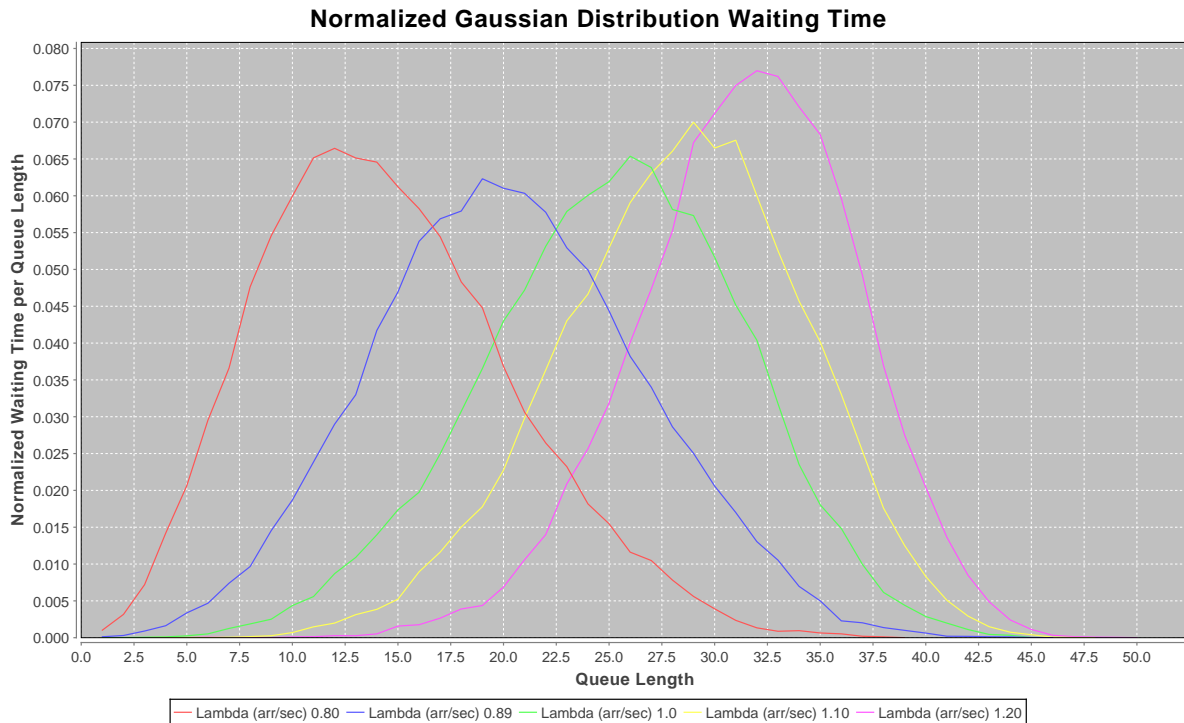


Figure A.7: CSMA Normalized Gaussian Waiting Time

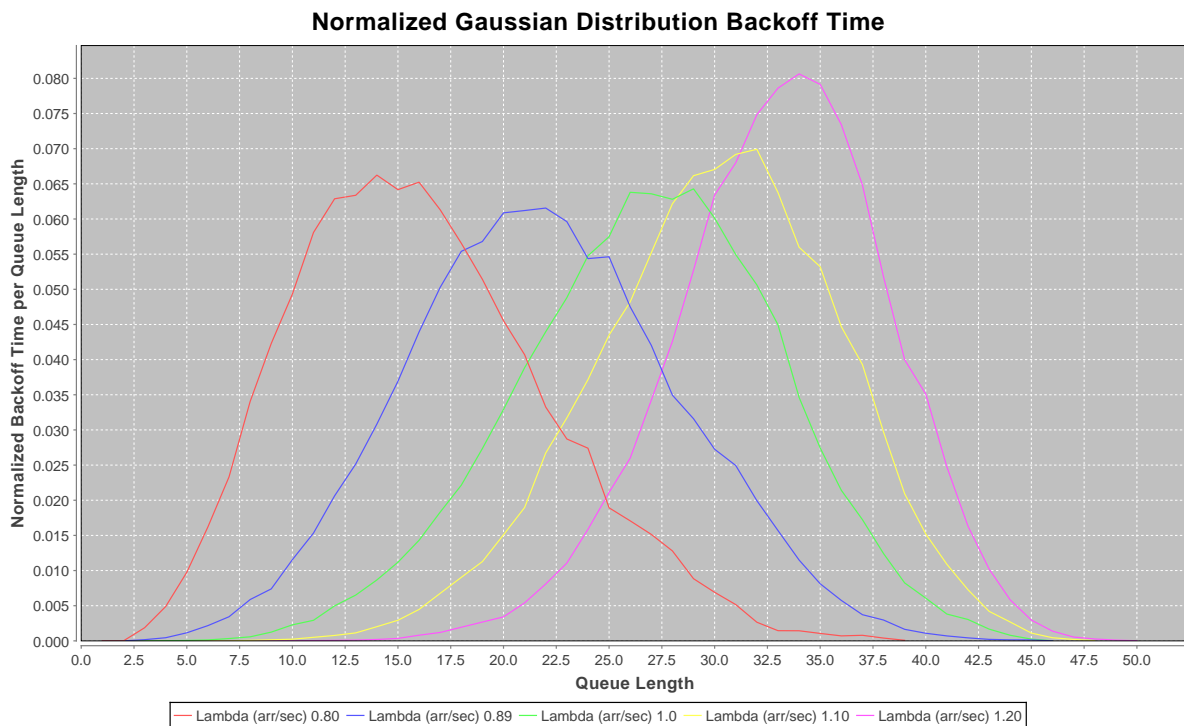


Figure A.8: CSMA Normalized Gaussian Backoff Time

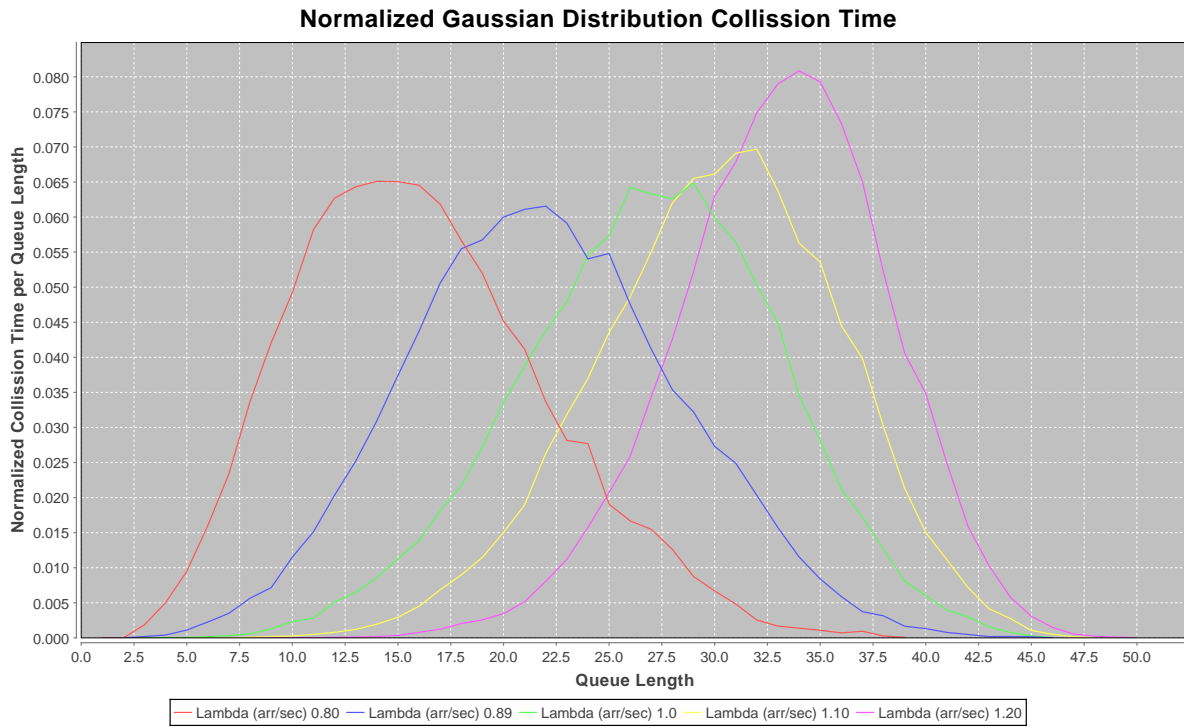


Figure A.9: CSMA Normalized Gaussian Collision Time

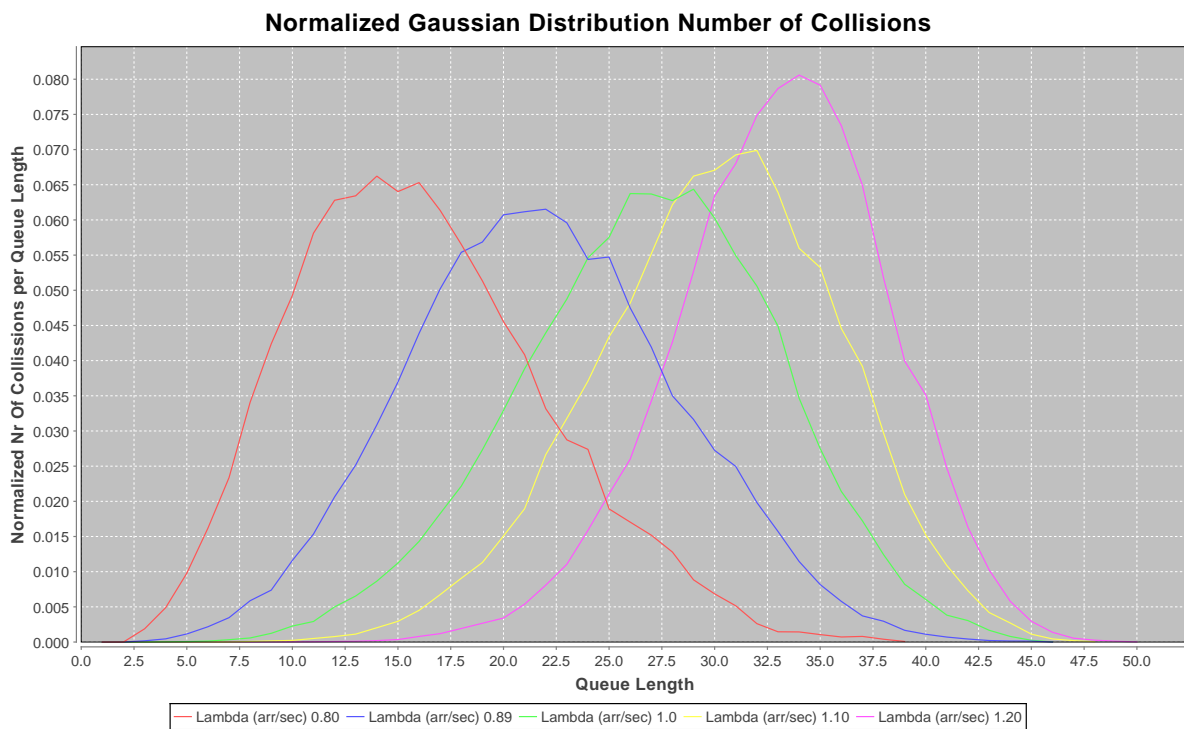


Figure A.10: CSMA Normalized Gaussian Number of Collisions

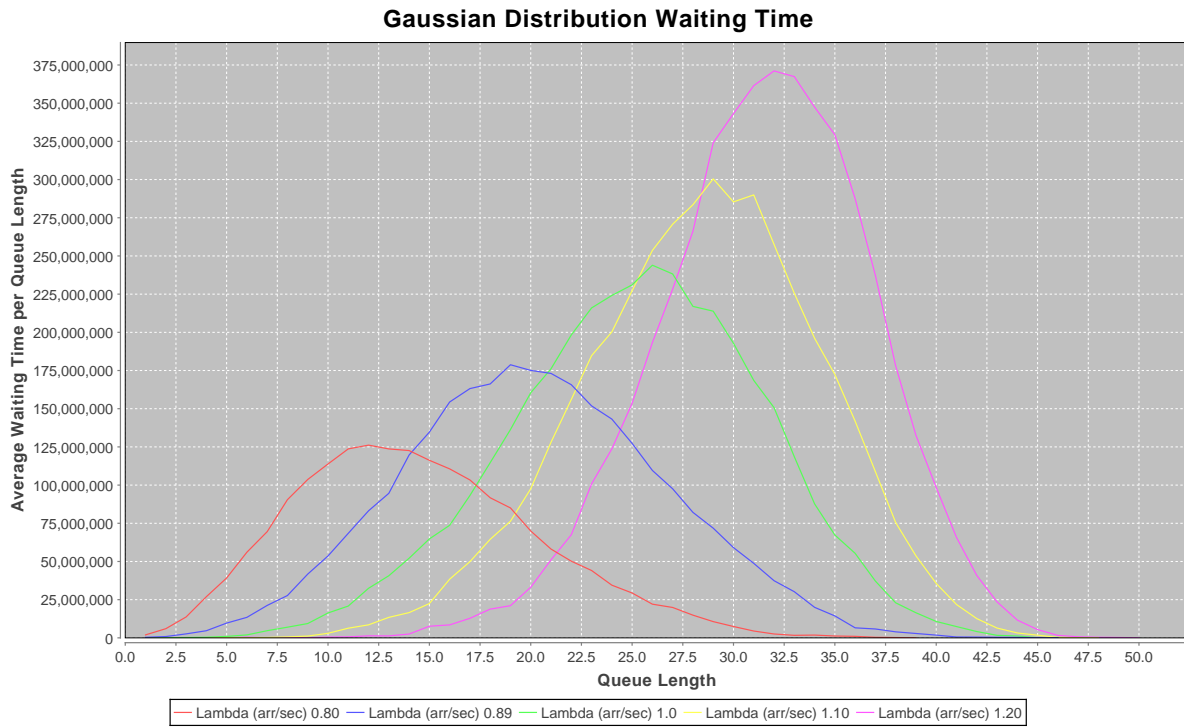


Figure A.11: CSMA Gaussian Waiting Time

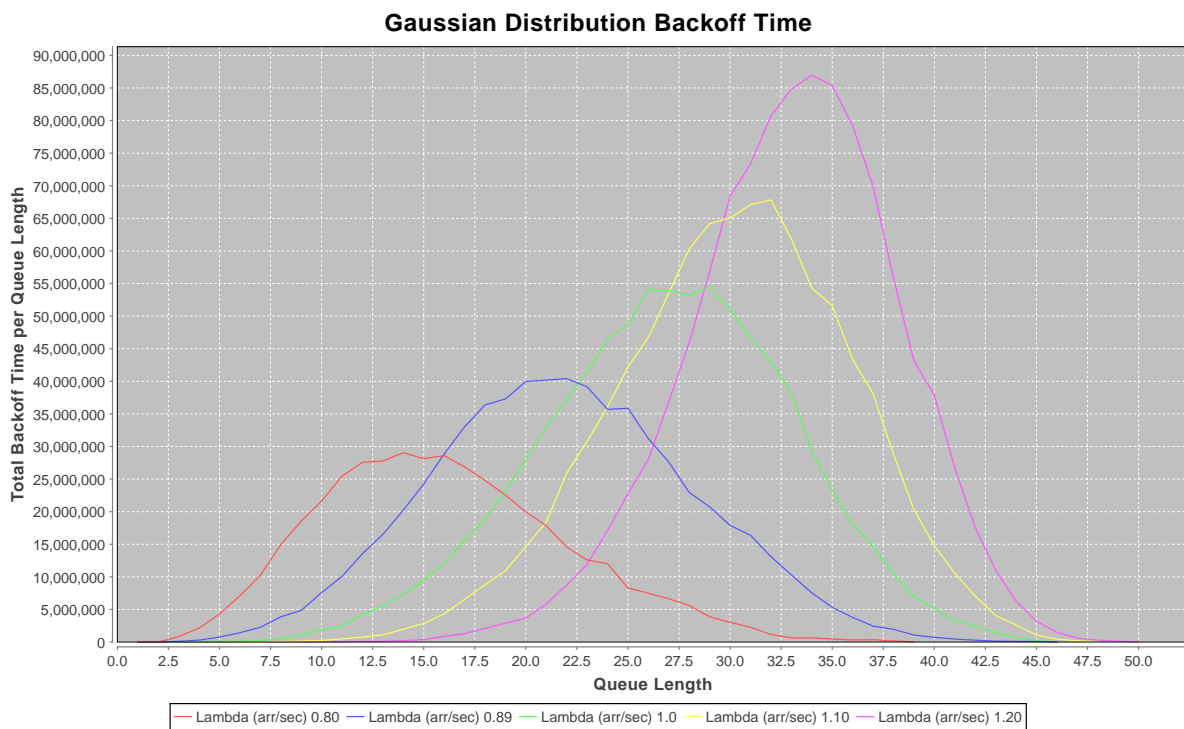


Figure A.12: CSMA Normalized Gaussian Backoff Time

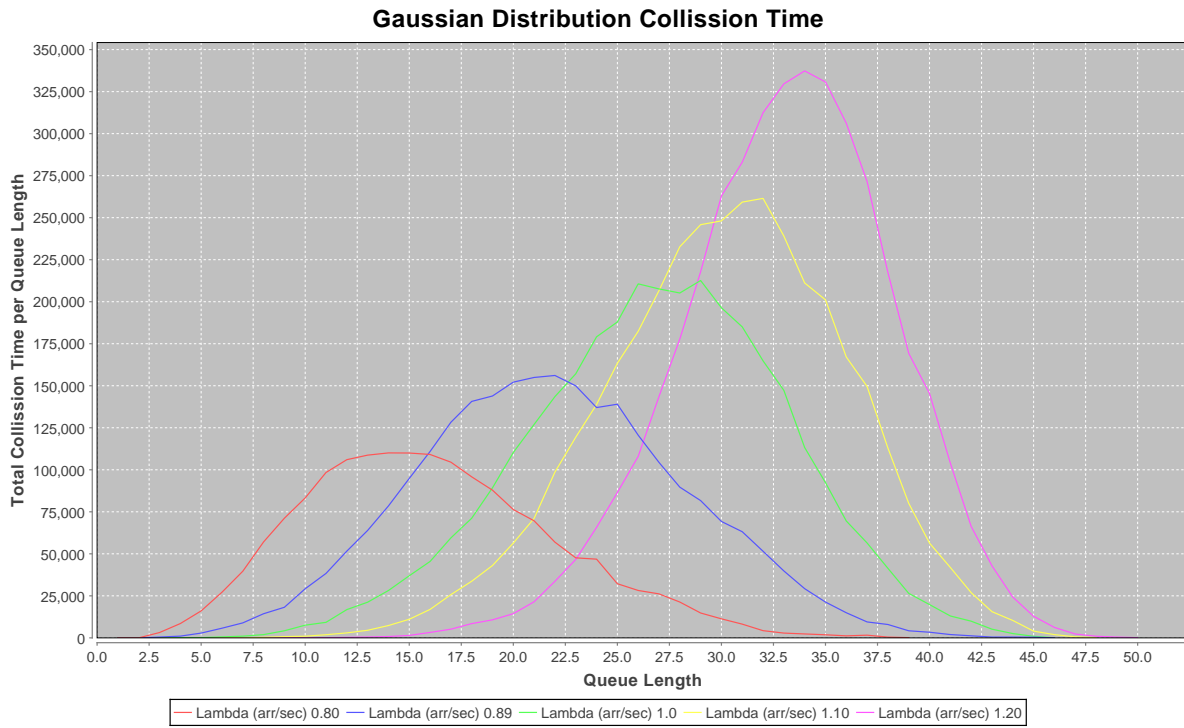


Figure A.13: CSMA Gaussian Collision Time

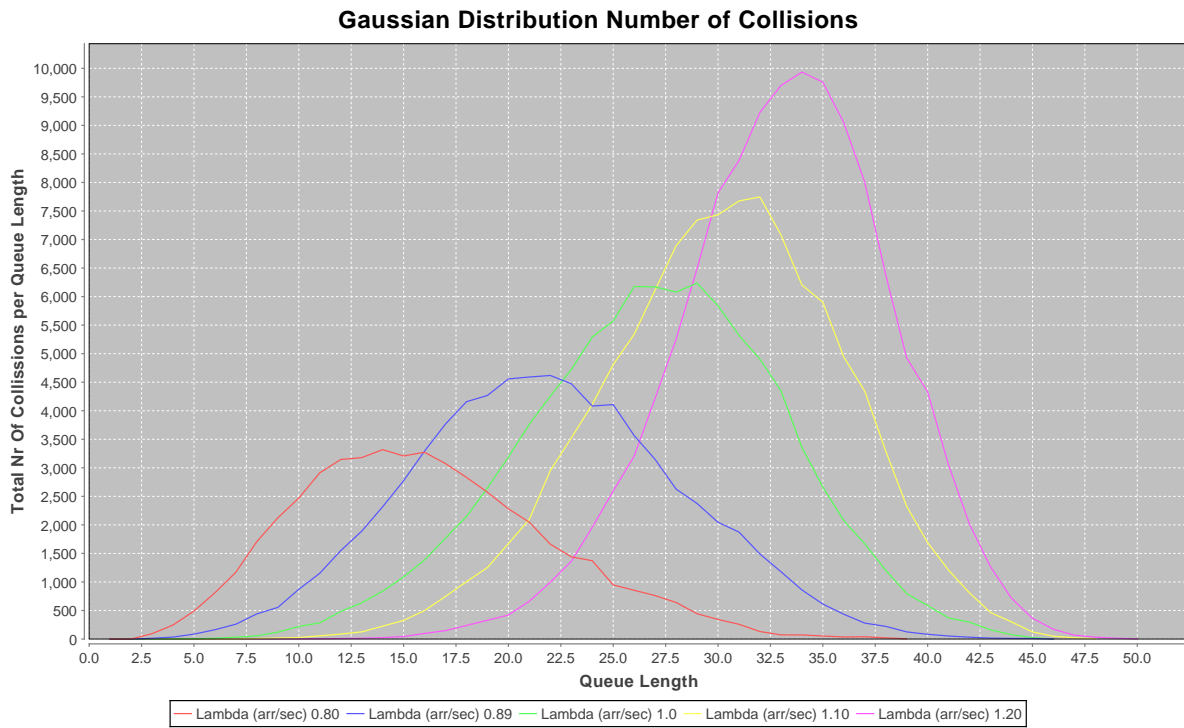


Figure A.14: CSMA Gaussian Number of Collisions

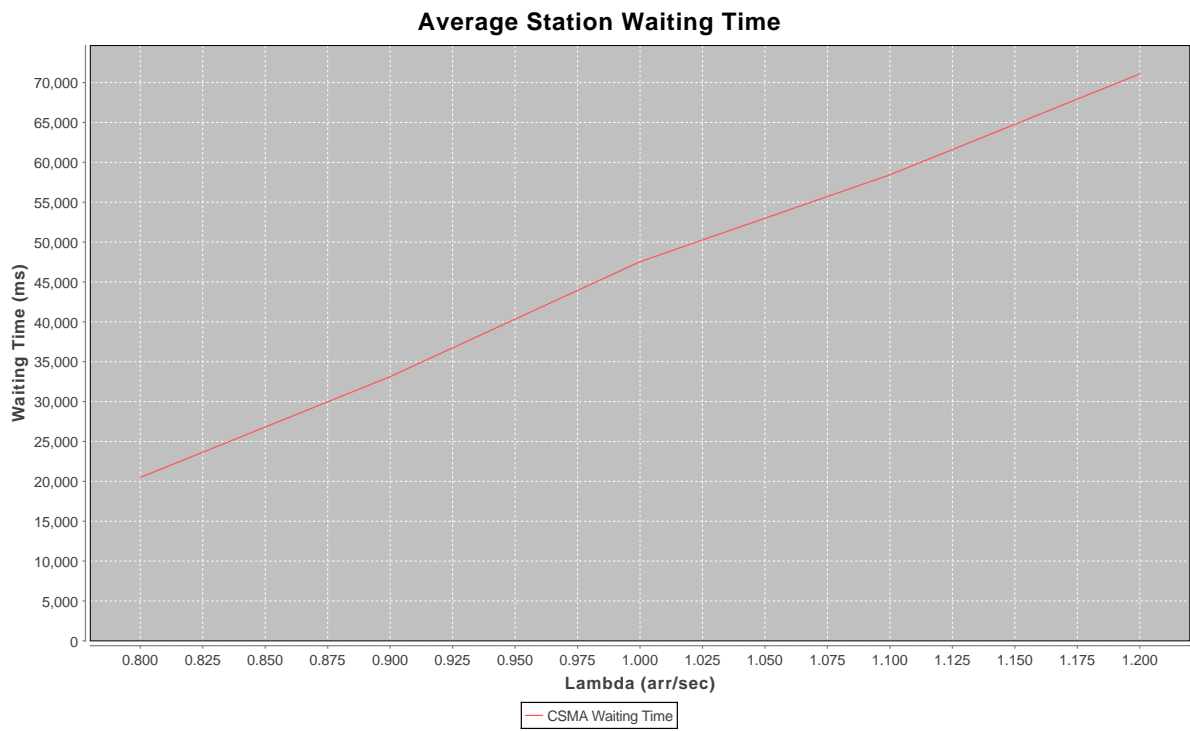


Figure A.15: Waiting Time of CSMA used for Gaussian Distributions

## **Appendix B**

# **Second Appendix**

### **B.1 Graphical User Interfaces**

Various graphical user interfaces have been designed for use in setting up the parameters of each model and analysing the results. These user interfaces are shown in the following subsections.

#### **B.1.1 Matlab**

The Matlab graphical user interface is used to select the type of theoretical model to be modelled as well as to allow theoretical and simulation results to be compared.



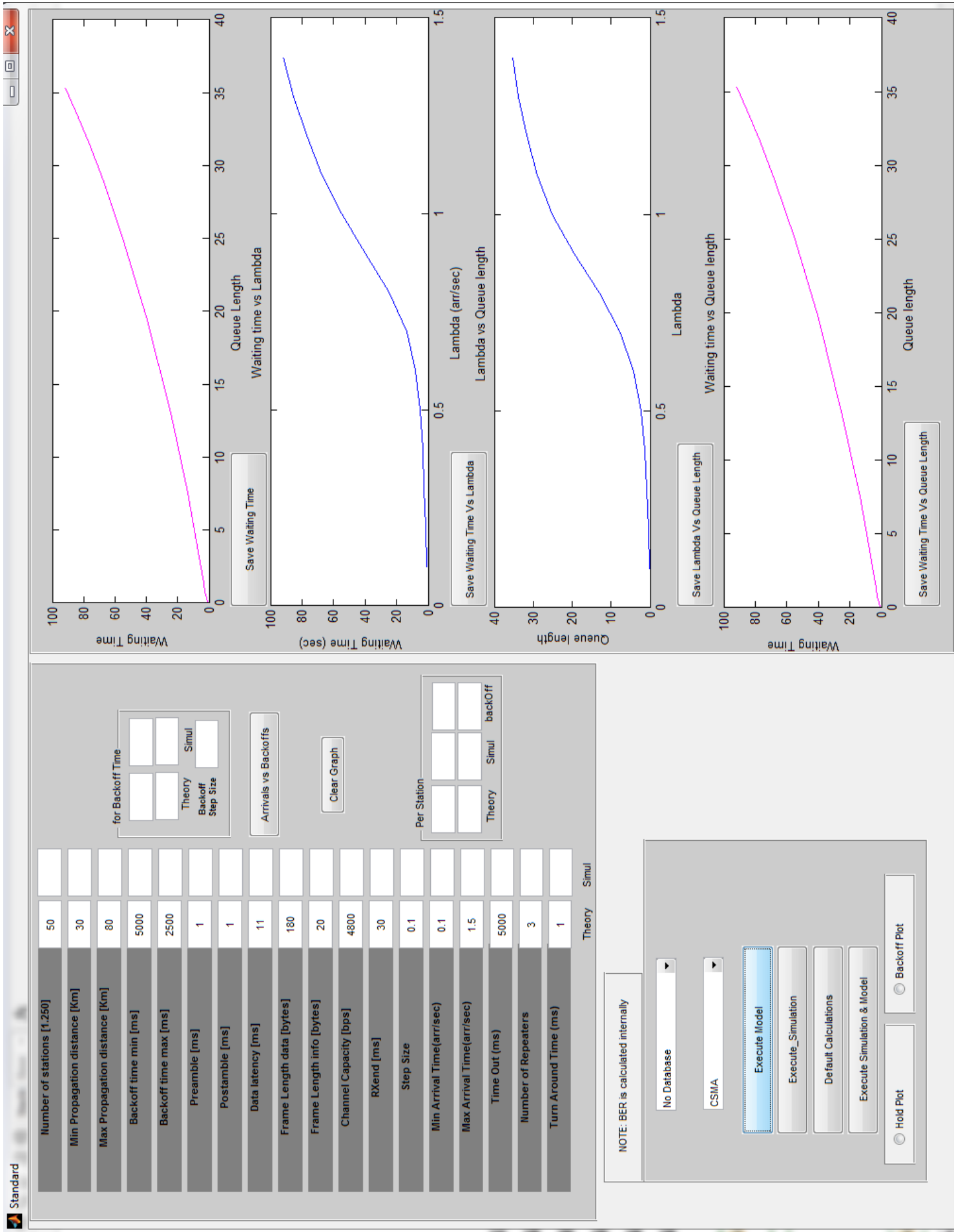


Figure B.1: Matlab GUI

## B.1.2 Java

### B.1.2.1 Main User Interface

The main graphical user interface is used to activate any of the simulation models, as well as to select various graphs from different models so that they can be compared with each other.

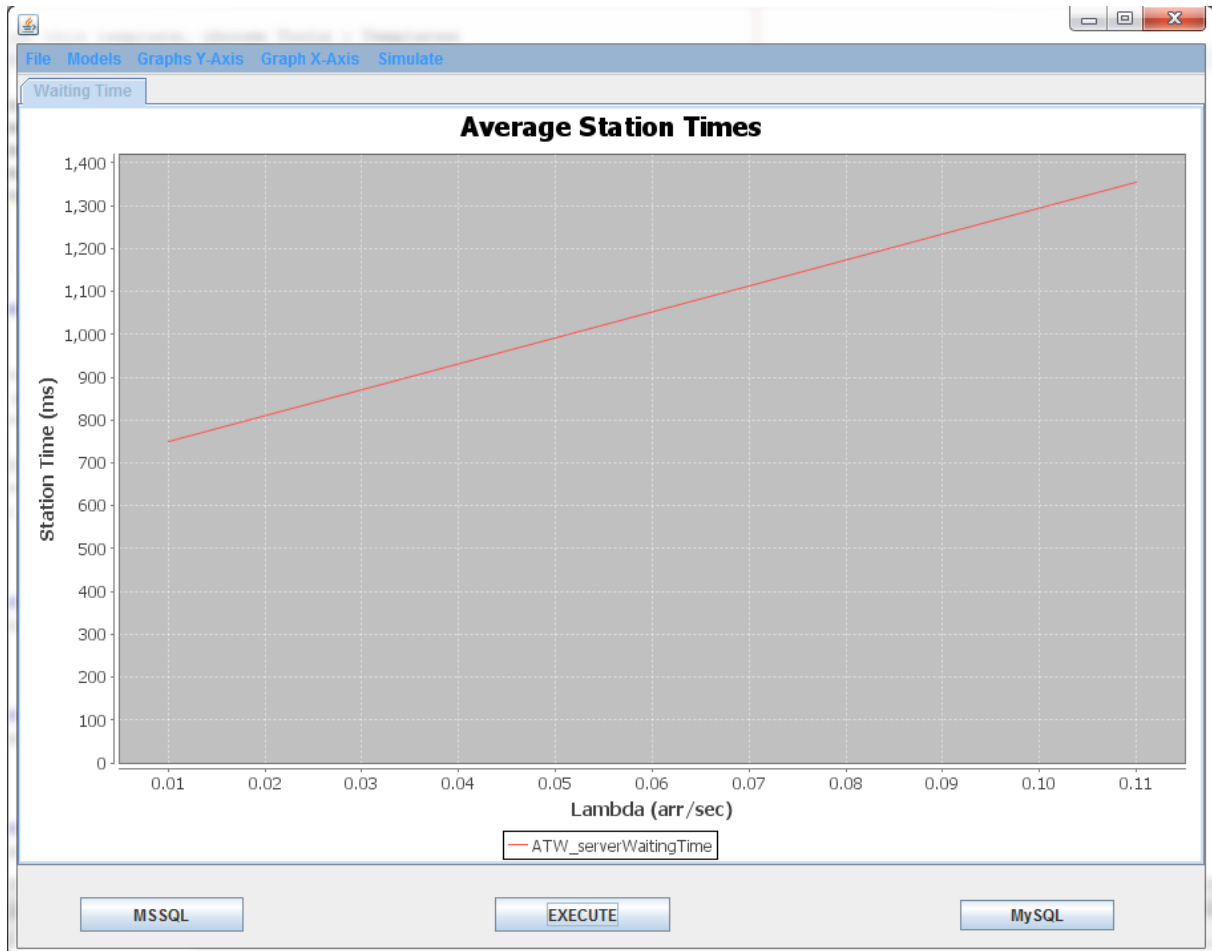


Figure B.2: Main Java Graphical User Interface

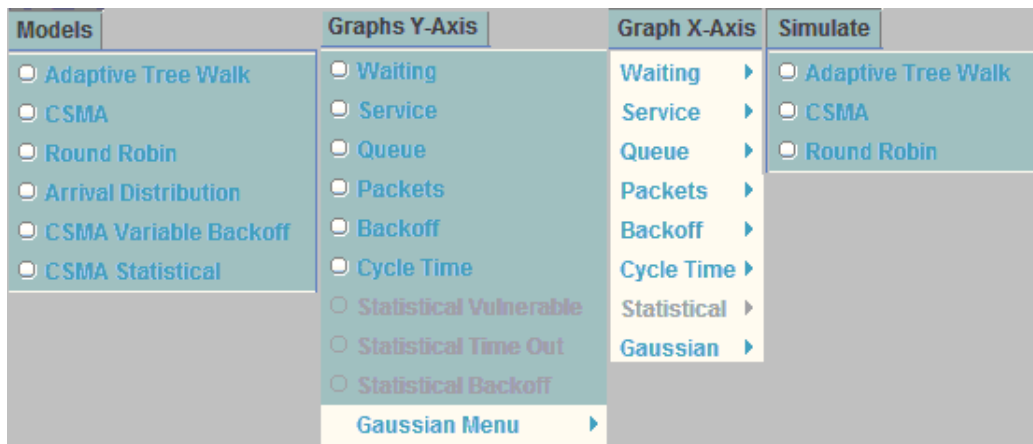


Figure B.3: Main Java Graphical User Interface Menus

### B.1.2.2 CSMA User Interface

The CSMA graphical user interface is used to set the parameters used within the simulation and to select various options to adjust the type of CSMA model required.

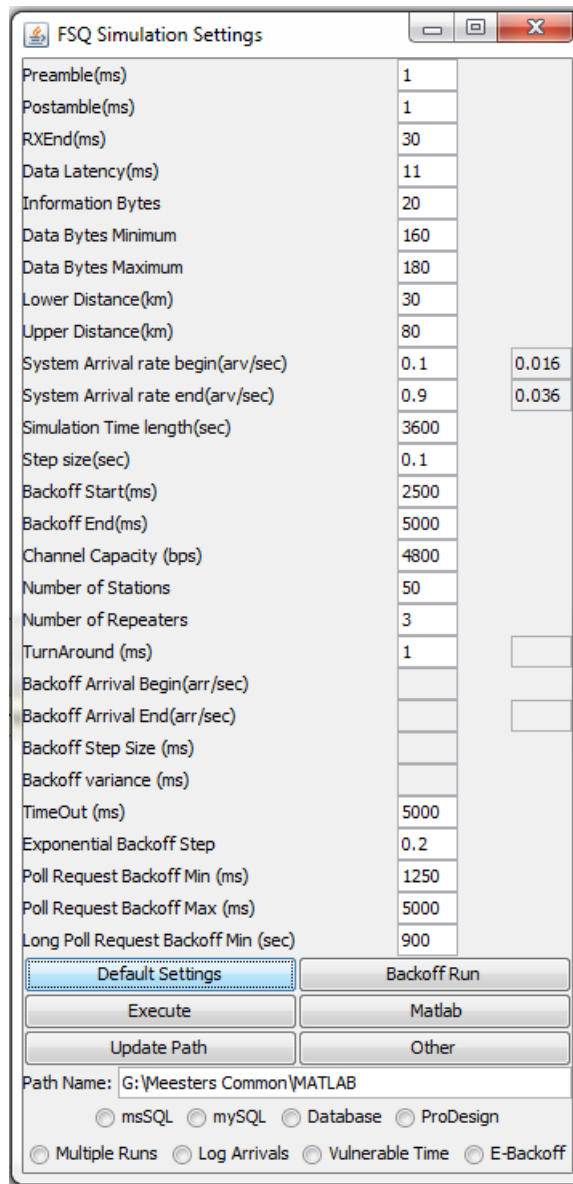


Figure B.4: CSMA Graphical User Interface

### B.1.2.3 RRP User Interface

The RRP graphical user interface is used to set the parameters used within the simulation.

**Round Robin Simulation**

Number of Stations	50	TimeOut (ms)	5000
Channel Capacity (bps)	4800	RXEnd (ms)	30
Data Bytes Minimum	160	Data Latency (ms)	11
Data Bytes Maximum	180	System Begin Arrival Rate (arv/sec)	0.95
Information Bytes	10	System End Arrival Reate (arv/sec)	1.5
Preamble (ms)	1	Simulation Length (sec)	7200
Postamble (ms)	1	Step Size	0.1
Upper Distance	120	Number of Repeaters	3
Lower Distance	20	Turn Around (ms)	1

Path Name: G:\Meesters\RoundRobin

Simulation Time:

Select Database:

- MSSQL
- MY SQL
- Multiple Run

Execute Defaults Graphs

Progress Bar

Exit

Figure B.5: RRP Graphical User Interface

#### B.1.2.4 ATW User Interface

The ATW graphical user interface is used to set the parameters of the simulation and select various options to adjust the type of ATW model to be modelled.

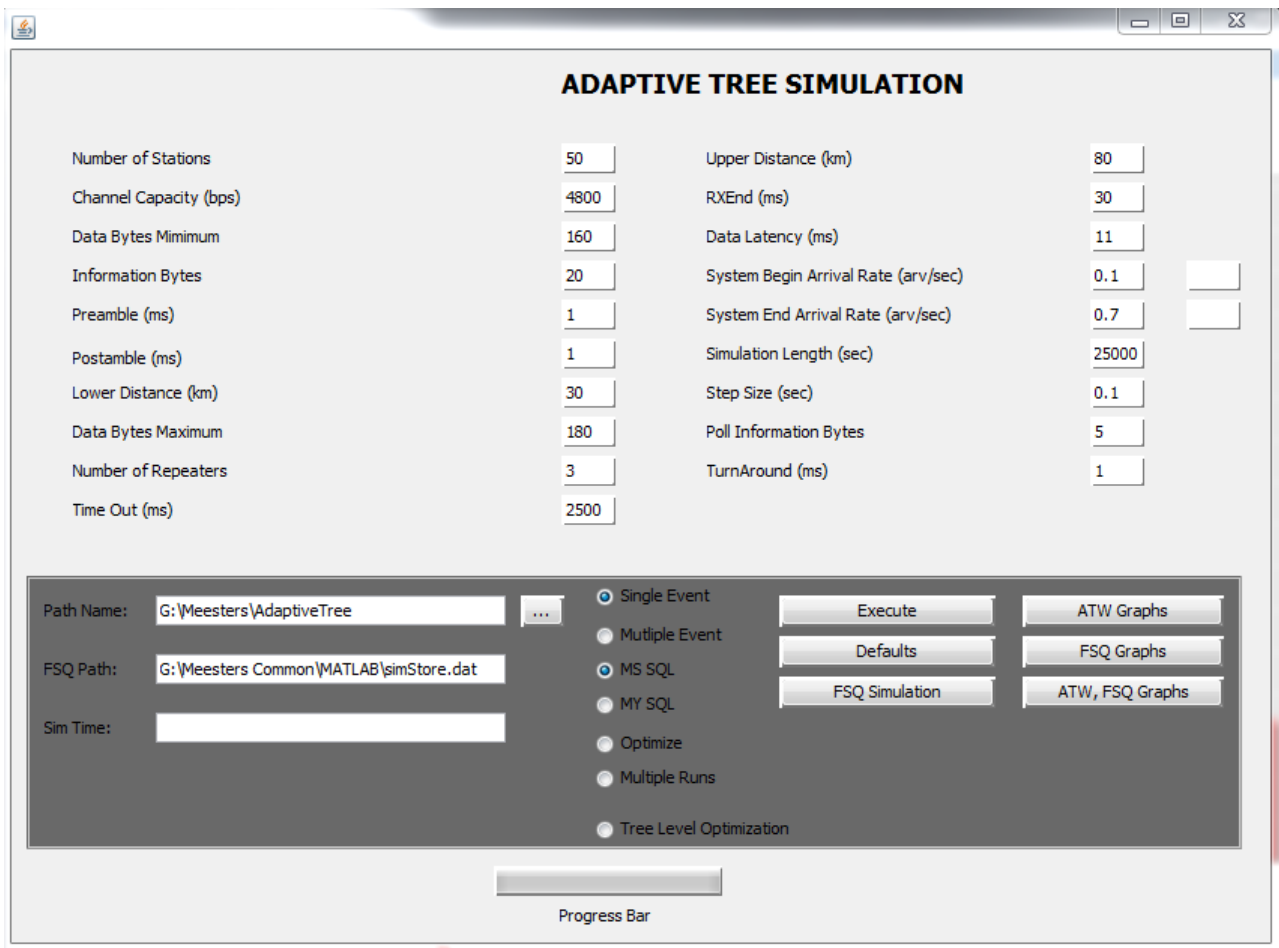


Figure B.6: ATW Graphical User Interface

## Appendix C

# Third Appendix

### C.1 Proofs and theories

#### C.1.1 Exponentially Distributed Arrivals have a Poisson Distribution

If the inter-arrival time is  $\tau$ , which means a source arrives with a new event every  $\tau$  seconds, then the probability that an arrival will take place in time  $(0, t)$  will be 1- the probability that no arrivals occur in  $(0, t)$ . This can be written as follows:

$P(\tau \leq t)$  Probability that an arrival occurs in time  $(0, t)$

$P(\tau > t)$  Probability that no arrivals occur in time  $(0, t)$

from this it can be shown that

$$P(\tau \leq t) = 1 - P(\tau > t) \quad (\text{C.1})$$

it is also know from Equation 4.2 that the probability that exactly 0 arrivals will occur in a time interval  $(0, t)$  is:

$$\begin{aligned} P(\tau > t) = P[X(t) = 0] &= \frac{(\lambda t)^0}{0!} e^{-\lambda t} \\ &= e^{-\lambda t} \end{aligned} \quad (\text{C.2})$$

therefore, making use of the above information, it can be shown that the probability distribution for the arrivals of sources with new events in a time interval  $(0, t)$  is

$$\begin{aligned} P(\tau \leq t) &= 1 - P(\tau > t) \\ &= 1 - P[X(t) = 0] \\ &= 1 - e^{-\lambda t} \end{aligned} \quad (\text{C.3})$$

#### C.1.2 Memoryless Markovian Property

This property is possible due to the fact that the Poisson process is connected to the exponential distribution as the exponential distribution is the only continuous distribution that possesses the memoryless property. To show that this is

true, the conditional distribution law is required as given in Equation C.4.

$$P(A | B) = \frac{P(A \cap B)}{P(B)} \quad (\text{C.4})$$

If the last arrival occurred at time  $t = 0$ , then it can be shown that the probability for no arrivals in time  $(0, t_0)$  is

$$P(\tau > t_0) = e^{-\lambda t_0} \quad (\text{C.5})$$

as shown in Equation C.2. It can also be shown that the probability of an arrival in time interval  $(0, t_0 + t)$  is

$$\begin{aligned} P(\tau \leq t_0 + t) &= 1 - P(\tau > t_0 + t) \\ &= (1 - e^{-\lambda(t_0 + t)}) \\ &= 1 - e^{-\lambda(t_0 + t)} \end{aligned} \quad (\text{C.6})$$

Equation C.7 shows an arrival that occurs in time interval  $(0, t_0 + t)$ , given that no arrivals occur between  $(0, t_0)$ .

$$\begin{aligned} P(\tau \leq t_0 + t | \tau > t_0) &= \frac{P[(\tau \leq t_0 + t) \cap (\tau > t_0)]}{P(\tau > t_0)} \\ &= \frac{P(\tau \leq t_0 + t) - P(\tau \leq t_0)}{P(\tau > t_0)} \\ &= \frac{P(\tau \leq t_0 + t) - [1 - P(\tau > t_0)]}{P(\tau > t_0)} \\ &= \frac{(1 - e^{-\lambda(t_0 + t)}) - (1 - e^{-\lambda t_0})}{e^{-\lambda t_0}} \\ &= \frac{e^{-\lambda t_0} - e^{-\lambda(t_0 + t)}}{e^{-\lambda t_0}} \\ &= \frac{e^{-\lambda t_0}(1 - e^{-\lambda t})}{e^{-\lambda t_0}} \\ &= 1 - e^{-\lambda t} \end{aligned} \quad (\text{C.7})$$

It can also be shown that the probability that an arrival can occur in a time interval  $(0, t)$  is

$$\begin{aligned} P(\tau \leq t) &= 1 - P(\tau > t) \\ &= 1 - e^{-\lambda t} \end{aligned} \quad (\text{C.8})$$

### C.1.3 Monroe-Penrose Pseudo Inverse

The Monroe-Penrose Pseudo Inverse (pseudo inverse) may be used as a general way to solve linear equations of the form  $AB = C$ , where A is a  $(m \times n)$  matrix, B is a  $(n \times 1)$  matrix and C is a  $(m \times 1)$  matrix. In this case  $m > n$ , there are more constraining equations than there are free variables which are the probabilities found in matrix B. The pseudo inverse gives the solution to B such that  $A^+B \approx C$ , where  $A^+$  is the pseudo inverse. A special case of the pseudo-inverse is given when matrix A is full rank [2]:

- case  $m < n$ :  $A^+ = A^T (AA^T)^{-1}$
- case  $m > n$ :  $A^+ = (AA^T)^{-1} A^T$

For matrix A which has dimension  $(m \times n)$ , the maximum possible rank is the smaller of  $m$  and  $n$ . In the above case this is  $n$ . If the rank of matrix A is therefore equal to  $n$  (therefore having  $n$  linearly independent columns), then it has full

rank. By making use of elementary row operations and obtaining the echelon matrix of  $A$ , it can be seen that there are  $n$  non-zero rows, meaning that there are  $n$  linearly independent columns or rows [20]. A vector set  $\{v_1, v_2, \dots, v_k\}$  is said to be linearly independent if

$$x_1 v_1 + x_2 v_2 + \dots + x_k v_k = 0 \quad (\text{C.9})$$

Determining that we can use the special case of the pseudo-inverse for  $m > n$ , the probability matrix  $B$  can then be determined as:

$$\begin{aligned} B &= A^+ B C \\ &= (A A^T)^{-1} A^T C \end{aligned} \quad (\text{C.10})$$