TIME DOMAIN MEDIUM ACCESS CONTROL PROTOCOLS FOR UNDERWATER ACOUSTIC NETWORKS

SHIRAZ SHAHABUDEEN

NATIONAL UNIVERSITY OF SINGAPORE

2011

TIME DOMAIN MEDIUM ACCESS CONTROL PROTOCOLS FOR UNDERWATER ACOUSTIC NETWORKS

SHIRAZ SHAHABUDEEN

(B.Eng., NUS, M.Tele.Eng., University of Melbourne)

A THESIS SUBMITTED

FOR THE DEGREE OF DOCTOR OF PHILOSOPHY

DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

NATIONAL UNIVERSITY OF SINGAPORE

2011

Acknowledgments

This work would not have been possible if not for the support from Acoustic Research Laboratory and the Defence Science Technology Agency (DSTA), Singapore, under the guidance of Dr. Mandar Chitre and Dr. Mehul Motani. Many of my close colleagues at ARL have been of immense help in this endeavour, especially Dr. John Potter, Dr. Venugopalan Pallayil, Mr. Mohan Panayamadam, Mr. Shankar Satish, Mr. Alan Low, Mr. Koay Teong Beng and Mr. Iulian Topor.

Dedication

This work is dedicated to my parents and my dear wife who have been my support throughout, and also to the rest of my wonderful family.

Contents

Summary **xv**

List of Figures **xvii**

List of Symbols **xxv**

Abbreviations **xxix**

Chapter 1 Introduction **1**

1.1 Background and Motivation 2

1.2 Objectives 4

1.3 Methodology 6

1.4 Outline 7

1.5 Novel Contributions 8

Chapter 2 Literature Review **11**

2.1 Media Access Control 11

2.1.1 Static protocols 12

2.1.2 Dynamic contention-based MAC 13

2.1.3 Conflicting opinions and results 15

2.1.4 Dynamic contention-free MAC 15

2.2 Energy Conservation 16

2.3 AUV Networking 17

2.4 Standardization and Software Frameworks 17

2.5 Conclusion 18

Chapter 3 Investigation of MAC Protocol Choices for UANs 21

3.1 Introduction 21

3.1.1 Topology 23

3.1.2 Spatial re-use, channelization and allocation 25

3.1.3 Need for dynamic channelization and allocation in UANs . . 27

3.2 Selection of MAC Protocols for UANs 28

3.2.1 The general equivalence of static TDMA, FDMA and CDMA 29

3.2.2 General strengths and weaknesses of CDMA, FDMA and TDMA 31

3.2.2.1 Full duplex requirement for CDMA and FDMA . . 32

3.2.2.2 CDMA performs better in terrestrial cellular networks? 34

3.2.3 Dynamic allocation protocols 36

3.2.4 Dynamic TDMA protocol and MACA based protocols . . . 38

3.2.5 Re-use, topology selection 41

3.2.6 Propagation delay and its impact 42

3.3 Conclusion 43

Chapter 4 A high performance MAC protocol for underwater acoustic networks: MACA-EA **45**

4.1 Review 45

4.2 System Model 49

4.2.1 Input-output models 49

4.2.2 Packet detection, error and collision model 50

4.2.3 MACA-based protocol model 51

4.2.4 Performance measures 52

4.2.5 A brief note on simulations 54

4.3 Analysis of Service Time and Throughput 55

4.3.1 Markov chain model for the protocol excluding retries 55

4.3.2 Enhanced retry mechanism 61

4.3.3 Expected throughput for reliable transfer 65

4.3.4 Expected packet service time s_p 67

4.3.5 Comparison with previous analyses 67

4.3.6 Sea trial results 68

4.4 Service Time Distribution Analysis 71

4.5 Queuing Analysis 75

4.5.1 Unsaturated queuing analysis 75

4.5.2 Expected waiting time 77

4.5.3 Waiting time variation with batch size 79

4.6 Protocol Enhancement: MACA-SEA 83

4.6.1 Algorithm outline 84

4.6.2 MACA-SEA performance 85

4.7 Discussion 86

4.7.1 Pre-emptive contention 86

4.7.2 Optimum RTS window 89

4.7.3 Physical Carrier Sense 91

4.7.4 Single Long DATA Instead of Batches 91

4.7.5 Optimum number of ACKs 92

4.7.6 Forward Error Correction (FEC) and Power Control 93

4.7.7 Multi-hop and hidden nodes 94

4.8 Conclusion 95

Chapter 5 Adaptive Multi-mode Medium Access Control for Underwater Acoustic Networks 99

5.1 Introduction 100

5.2 Protocol Modes in MAC-AMM 102

5.2.1 Level-1 compliance 102

5.2.2 Level-2 compliance 104

5.2.3 Distributed mode of Level-2 MAC: MACA-EA 105

5.2.4 Centralized mode of Level-2 MAC: MACA-C 106

5.2.5 Level-2 distributed mode with no handshaking: DATA-ACK 107

5.2.6 Adaptive multi-mode MAC 107

5.3 Throughput Analysis of LEVEL-2 MAC 109

5.3.1 DATA-ACK 110

5.3.2 MACA-EA 111

5.3.3 MACA-C . . . 111

5.4 Mode Adaptation Based on Traffic Intensity . . . 115

5.4.1 Service time distribution . . . 116

5.4.2 MACA-EA . . . 118

5.4.3 MACA-C . . . 118

5.4.4 DATA-ACK . . . 120

5.4.5 TDMA-REF . . . 121

5.4.6 Effect of traffic intensity . . . 123

5.4.7 Adaptation algorithm . . . 126

5.5 State dependent DATA-ACK protocol . . . 130

5.6 Discussion . . . 130

5.7 Conclusion . . . 132

Chapter 6 Twin-MAC Protocols **135**

6.1 Introduction . . . 136

6.1.1 Throughput greater than 1? . . . 136

6.1.2 2-node network . . . 137

6.1.3 4-node regular tetrahedron network . . . 138

6.1.4 4-node stretched tetrahedron network . . . 140

6.1.5 N-node networks . . . 141

6.1.6 Bounded network geometries . . . 143

6.1.7 Remarks . . . 145

6.2 Pair-wise Transmission Protocols . . . 145

6.2.1 Twin-TDMA . . . 147

6.2.1.1 Throughput 147
6.2.1.2 Queuing delay 150
6.2.2 Dynamic Twin-TDMA 152
6.2.2.1 Centralized dynamic Twin-TDMA 153
6.2.2.2 Performance 154
6.2.3 Twin-ALOHA 155
6.3 Conclusion 158

Chapter 7 A Multi-channel MAC Protocol for AUV Networks 159

7.1 Introduction 159
7.2 Multi-channel Modelling 161
7.2.1 BER performance modelling 161
7.2.2 The packet train model and packet loss ratio 163
7.3 Network Architecture and Algorithms 165
7.3.1 The physical layer 165
7.3.2 Network layer and data transmission model 165
7.3.3 The overall architecture 166
7.3.4 The basic DLL algorithm 166
7.4 Simulation Setup 168
7.4.1 Modelling of AUV node motion 168
7.4.2 Some factors affecting performance 170
7.5 Simulation Results 170
7.5.1 MACA-MCP effective data rate performance 171
7.5.2 MACA-MCP throughput performance 173

7.5.3 Adaptive clustering behaviour 175

7.6 Conclusion 176

Chapter 8 Unified Simulation and Implementation Software Framework 177

8.1 Introduction 177

8.2 FAPI and UNA 180

8.3 ARL Modem 182

8.4 The Simulator 183

8.4.1 The unified simulator and modem software model 184

8.4.2 Simulator physical layer details 185

8.4.3 Channel model details 187

8.4.4 Simulator limitations 188

8.5 Writing MAC Code 189

8.6 Modem Trials and Results 191

8.6.1 Sea trials 192

8.7 Conclusion 194

Chapter 9 Conclusion 195

9.1 Future Work 198

Bibliography 201

Appendix A MACA Analysis 213

A.1 The Performance of the Standard ACK Model 213

A.2 Distribution Analysis Markov Chain 213

Appendix B Analysis for MAC-AMM **216**

B.1 DATA-ACK throughput analysis 216

B.2 M/D^B/1 Waiting Time Analysis for MACA-C and TDMA 220

B.3 Inter-cell or inter-MC interference 220

Appendix C Super TDMA **223**

C.1 Prototype Schedule for Odd-even Distance Networks 223

C.2 ρ-schedule 224

Appendix D UNA, FAPI, DLL Utilities and Sample MAC Code **225**

D.1 UNA Messages 225

D.1.1 Key physical layer messages 225

D.1.2 Key data link layer messages 225

D.2 Framework API 226

D.3 Data link layer Utility Functions 227

D.3.1 Reception handling 227

D.3.2 HPA control 227

D.3.3 PDU handlers 228

D.4 Sample MAC Code 229

D.4.1 The main handler interface 229

D.4.2 MAIN_MAC_HANDLER 230

D.4.3 DLL_SEND_PKT_REQ 230

D.4.4 FAPI_TIMER_EXPIRED_NTF 231

D.4.5 PHY_PKT_XMIT_RSP 231

D.4.6 PHY_INCOMING_PKT_NTF 232

Appendix E Publications From This Thesis **233**

Summary

The central objective of this thesis is an investigation of time domain medium access control (MAC) protocols, specifically those based on Medium Access Collision Avoidance (MACA), for Underwater Acoustic Networks (UAN). A review of the key developments in data link layer (DLL) at the start of the work revealed many gaps in research on the relative merits of MAC protocols for UANs and the performance of protocols. Re-analysis of the design choices for MAC in UANs led to the observation that in a distributed topology, CDMA and FDMA require full-duplex and multi-channel functionality to have similar performance as TDMA. Time domain protocols, including those based on MACA, are found to be fundamentally best suited for UAN MAC.

A key objective was to develop new high performance MACA-based protocols for UANs. A novel ARQ variation called Early-Multi-ACK for batch-node data transmissions and some other enhancements to MACA give rise to the novel MACA-EA protocol. An in-depth analysis of this protocol, including factors such as propagation delay, detection and decoding errors not considered in many previous analyses, gives new closed form metrics for mean service time and through-

put for reliable batch transmission. The batch service time distribution closely matches the exponential distribution. Queuing analysis of the waiting time shows that there is an optimum batch size that minimizes total waiting time.

Other novel protocol refinements have been developed such as the MACA-SEA protocol, that can achieve higher performance through a pseudo-TDMA "taking turns" behaviour. A novel multi-channel protocol called MACA-MCP for effective networking in a small AUV network, exploits mobility through multiple acoustic modems operating at different frequency bands suited for different ranges. A multi-mode protocol suite – MAC-AMM, incorporates novel adaptation techniques and uses a centralized MACA-C protocol mode, a distributed MACA-EA mode and a novel state dependent DATA-ACK mode to achieve efficient communications under varying environmental and traffic intensity. Motivated by a recently published observation that in an N-node UAN, the upper bound normalized throughput is $N/2$ and not 1 as is the case in networks with negligible propagation delay, three novel protocols – Twin-TDMA, Dynamic Twin-TDMA and Twin-ALOHA that utilize simultaneous transmissions were also developed.

A new unified software framework has been developed that aids seamless simulations and sea-trials with the same MAC code, which helped ensure that the performance measures in this thesis are reliable and the protocols are guaranteed to work in real acoustic modems.

List of Figures

3.1 An example UAN architecture with two cells, one using a centralized topology while the other using a distributed topology (Chitre et al., 2008) 25

3.2 A schematic representation of D-TDMA, D-FDMA and D-CDMA. . 37

3.3 MACA protocol model with RTS/CTS/PACKET-TRAIN. Node A sends an RTS to Node B and Node B sends a CTS back to Node A. Node A then sends a DATA batch to Node B. Reception of CTS at another node C is shown which then performs a VCS to avoid interference with Node A's transmission. A potential collision from Node C is shown. How back-off starts after completion of one batch transmission is also indicated. 39

4.1 Main Markov chain for computing expected service time 57

4.2 Markov chain for Early-Multi-ACK 63

4.3 Throughput vs. batch size for $k = k_D = 0.36$, 0.49, 0.81 and 1.0, simulations (S), analysis (A). Parameters: $L = L_D = 0.5s, N = 3, D = 0.5s, W = 4, i = 3$. 65

4.4 Throughput vs. batch size. Parameters: $[L = 0.5s, k = 0.81, N = 3, D = 0.5s, W = 9, i = 3]$ 66

4.5 Markov chain to compute s_p 67

4.6 Analysis comparison with sea trials and simulations. Parameters: $N = 3, D = 0.4s, W = 10, P_d = 1, P = 0.9, i = 1, L = 0.9s, L_d = 0.6s$. A sample image that was transferred between modems in a recent sea trial is also shown. This file transfer used the MACA based protocol with the Early-ACK retry mechanism. 70

4.7 Part of Markov chain with dummy states for computing service time distribution 71

4.8 Service time CDF. Parameters: $L = L_D = 0.5s, N = 3, D = 0.5s, W = 4, i = 3, k = k_D = 0.81$. “Analytical” curve uses Exponential fit. 73

4.9 W_T (in seconds) vs. Batch size (B), Parameters: $L = L_D = 0.5s, N = 3, D = 0.5s, W = 4, i = 3,\ \lambda = 0.05,\ k = k_D = 0.81$ 80

4.10 W_T (in seconds) vs. Batch size (B), $N = 10, D = 1.0s, L = 0.5s, L_D = 1.5s, k = 0.81, k_D = 0.9, W = 23, i = 3, \lambda = 0.033$ 81

4.11 Variation of optimum (‘Opt’) and minimum (‘Min’) batch size with the number of nodes N and arrival delays. Parameters: $L = L_D = 0.5s, N = 3, D = 0.5s, W = 4, i = 3,\ k = k_D = 0.81$. 82

4.12 Waiting time behaviour illustration – the solid curve is W_T. Other key characteristics are as indicated. 83

4.13 MACA-SEA flowchart 85

4.14 Performance of MACA-SEA and MACA-EA. Parameters: $L = 0.5s, N = 5, D = 0.5s, i = 3, k = 0.64$. Contention window W as indicated. 86

4.15 Comparison with MACA-EA-WAIT, arrival delay = 20s. Parameters: $L = 0.5s, N = 3, D = 0.5s, W = 6, i = 3,\ k = 0.81$. 88

4.16 Comparison with MACA-EA-WAIT, arrival delay = 50s. Parameters: $L = 0.5s, N = 3, D = 0.5s, W = 6, i = 3,\ k = 0.81$. 89

4.17 Variation of network throughput with W (legend shows batch size B). Parameters: $L = 0.5s, N = 10, D = 0.5s, W = 4, i = 3,\ k = 0.81$. 90

4.18 Hexagonal cell model and sample traffic pattern 94

5.1 A sample physical layer packet structure shows the preamble and data signal portion. Physical layer compliance levels are as indicated. 103

5.2 MAC-AMM adaptation 109

5.3 Markov Chain for computing Expected Service Time for DATA-ACK protocol 110

5.4 Network throughput of MACA-EA, MACA-C and DATA-ACK (Packet duration $L = 0.5s, L_D = 1.0s$, detection and decoding probability $k = 0.81, k_D = 0.63$, one-way propagation delay $D = 0.5s$, number of nodes $N = 7$, contention window $W = 17$, Multi-ACK $i = 3$). . . . 112

5.5 Network throughput of MACA-EA, MACA-C and DATA-ACK: behaviour at low batch size, showing that DATA-ACK is better than RTS/CTS protocols with B=1. Parameters: $L = L_D = 0.5s, k = k_D = 0.81, N = 4, D = 0.5s, W = 11, i = 3$. 113

5.6 Markov chain for the MACA-C protocol 114

5.7 Service time distributions of MACA-EA and MACA-C. Parameters: $L = L_D = 0.5s, N = 4, D = 0.5s, i = 3, k = k_D = 0.81$. 117

5.8 Service time distribution of DATA-ACK. Parameters: $L = L_D = 0.5s, N = 4, D = 0.5s, k = k_D = 0.81$. 118

5.9 Waiting time for the different modes. Parameters: $L = 0.5s, L_D = 1.0s, N = 4, D = 0.5s, W = 11, i = 3, k = 0.81, k_D = 0.72$. 119

5.10 Waiting time for the different modes. Analysis (MACA-C deterministic service). Parameters: $L = 0.5s, L_D = 1.0s, N = 4, D = 0.5s, W = 11, i = 3, k = 0.81, k_D = 0.72$. 120

5.11 Comparison of deterministic and Markov models for TDMA-REF analysis. Parameters: $L = 0.5s, N = 4, D = 0.5s, W = 11, i = 3, k = 0.81$. 123

5.12 Varying batch size B, DATA-ACK (simulated) and MACA-EA at a given arrival delay. Parameters: $L = L_D = 0.5s, N = 4, D = 0.5s, W = 11, i = 3, k = k_D = 0.81$. 127

5.13 Comparing $B = 5$ and $B = 2$ MACA-EA with DATA-ACK (all analytical) at different arrival delays. Parameters: $L = L_D = 0.5s, N = 4, D = 0.5s, W = 11, i = 3, k = k_D = 0.81$. 128

5.14 State dependent variation for the DATA-ACK protocol mode. Parameters: $L = 0.5s, N = 4, D = 0.5s, W = 11, k = 0.81$. DATA-ACK modes simulated, TDMA-REF analytical. 131

6.1 Simultaneous transmissions in a two-node network. It shows exchange of packets of duration equal to the propagation delay (i.e., $L = D$) between node 1 and node 2. At time $t = 0$, the transmissions start. At $t = D$, the packets have fully left the transmitters and are just reaching the receivers. At $t = 2D$, the packets receptions are complete and the process repeats. 137

6.2 Regular tetrahedron and stretched tetrahedron networks 139

6.3 Simultaneous transmissions in a regular tetrahedron network. This shows a snapshot of the cyclic process viewed at node 1 in steady state. By symmetry, it is the same for all nodes. At time $t = 0$, interfering receptions arrive at node 1 from node 3 and 4 and node 1 starts transmitting to node 2. By $t = D$, node 1 has finished transmitting to node 2, and the expected packet from its peer node 2 has just arrived. The interfering receptions from node 4 and node 3 are also over. At $t = 2D$, the reception of the valid packet from node 2 is complete and the process repeats. 140

6.4 A 2-dimensional N-node network for even N 142

6.5 Throughput trade-off with delay 145

6.6 Twin-TDMA 149

6.7 Centralized topology 150

6.8 Dynamic TDMA 153

6.9 Throughput for slotted Aloha, un-slotted Aloha and twin-Aloha in a four node network 156

6.10 Optimal throughput for Aloha and Twin-Aloha 157

7.1 Coded BER curves for all channels. The BER is modelled as having a rapid increase after their maximum range in this study. 163
7.2 Packet train 163
7.3 PLR curves for all channels for 1000 bits per packet. At the optimum range, the packet loss is about 15%. 164
7.4 Network stack architecture 166
7.5 AUV nodes motion during one realization of the simulation 169
7.6 Effective data rates for the different protocols as a function of packet train size. Parameters: bits/packet = 1000, baud rates (SR/MR/LR)= 7200/2400/720 bits/s, propagation delay is time varying based on motion model, number of nodes $N = 8$. 171
7.7 Network throughput normalized to individual BAUD. Parameters: bits/packet = 1000, baud rates (SR/MR/LR)= 7200/2400/720 bits/s, propagation delay is time varying based on motion model, number of nodes $N = 8$. 174
7.8 Node clustering behaviour 175

8.1 Message nomenclature in UNA (adapted from (Chitre et al., 2006)) 181
8.2 The ARL modem used in the tests 183
8.3 Simulator and modem software model 185
8.4 Simplified physical layer state diagram (adapted from (Shahabudeen and Chitre, 2005)) 186
8.5 Key sub-functions of data link layer 190

8.6 Node deployment geometry 192
8.7 Sea trial location off the north of Singapore and below Pulau Ubin island. Boats were separated by about 400 to 500 meters during trials. Water depth was 6 to 12 meters. 193
A.1 Markov chain for Standard ACK model 214
B.1 Neighbouring cells Scenario 1 221
C.1 An illustration of a ρ-schedule 224

List of Symbols

α	Utility variable for queueing analysis	79
b	Bandwidth	29
B	Batch size	49
β	Utility variable for queueing analysis	79
C	Number of user channels	29
D	Propagation delay	50
$\mathbf{D}$	Delay Matrix for SuperTDMA concept	142
E	Bit Error Rate	164
$E(m,n)$	Expected no: of times in state n if starts in state m	60
$\mathbf{F}$	Fundamental matrix	60
G	Girth in SuperTDMA concept	143
γ	Utility variable for service time computation	62
i	Number of multi-ACKs	61
k	Probability of correct detection and decoding of a packet	50
k_D	Probability of correct detection and decoding of a data packet	50
κ	The spreading factor for CDMA	30
l	Slot duration	56

L Time duration of a packet 50

L_D Time duration of a data packet 50

L_T Expected system total queue length 78

λ Poisson arrival rate 75

$\mathbf{M}$ Markov matrix 60

$\mathbf{M_R}$ Markov matrix for the complete retry process 72

μ_b Service rate 77

N Number of nodes 50

N_0 Gaussian ambient noise with power per unit bandwidth 29

P Packet decoding probability 50

P_D Decoding probability of data packets 50

P_d Detection probability 50

P_R Received power per unit bandwidth 29

P_T Transmission power per unit bandwidth 29

$P(g, h)$ State transition probability from state g to state h 58

$P(z)$ Probability generating function 77

$\mathbf{Q}$ Transient state matrix 60

r_0 Root of the characteristic equation 77

R Data rate 29

R_{PL} Packet Loss Ratio 163

s_b Mean batch service time 53

s_{CTS} Time till successful reception of CTS from state 1 to state 6 62

s_p Mean packet service time 53

S	Slot length in SuperTDMA concept	136
ς	Received signal-to-noise ratio	29
Ψ	Signal-to-interference-noise ratio (SINR)	34
t	Time window	29
t_A	Timer to wait for CTS or ACK	51
t_B	Time for batch transmission	58
t_{B_VCS}	Time for batch transmission in VCS states	58
T	Normalized throughput	53
ω	Packet transmission probability	58
W	RTS back-off window size	51
W_Q	Waiting time in the queue	54
W_T	Total waiting time	54
W'	Expected value of the uniformly distributed contention window	58
χ	Transmission schedule for SuperTDMA concept	138

Abbreviations

ARL	Acoustic Research Laboratory, NUS	5
AWGN	Additive White Gaussian Noise	188
BER	Bit Error Rate	22
BS	Base Station	24
CCL	Command and Control Language	17
CDMA	Code Division Multiple Access	22
CSMA	Carrier Sense Multiple Access	13
CTS	Clear to Send	13
DACAP	Distance Aware Collision Avoidance Protocol	14
DCF	Distributed Coordination Function	23
DLL	Data link layer	7
FAMA	Floor Acquisition Multiple Access	14
FAPI	Framework API	182
FDMA	Frequency Division Multiple Access	22
FEC	Forward Error Correction	91
MAC	Medium Access Control	1
MACA	Medium Access Collision Avoidance	1

MACA-C	Centrally Controlled Polled MACA	104
MACA-EA	MACA-Early-ACK	52
MACA-SEA	MACA-Sequenced-Early-ACK	84
MAI	Multiple Access Interference	30
MC	MAC Controller	104
NAV	Network Allocation Vector	57
OFDMA	Orthogonal Frequency Division Multiple Access	35
PA	Power Amplifier	69
PCF	Point Coordination Function	24
PCS	Physical Carrier Sense	43
RRTS	Request for RTS	15
RTR	Request-to-receive	13
RTS	Request to Send	13
SDMA	Space Division Multiple Access	12
SINR	Signal-to-interference-noise Ratio	34
SNR	Signal-to-noise Ratio	162
TDMA	Time Division Multiple Access	22
UAN	Underwater Acoustic Network	1
UID	Unique Identification Number	52
UNA	Underwater Network Architecture	179
VCS	Virtual Carrier Sense	51

Chapter 1

Introduction

Ad hoc underwater acoustic networks (UAN) have been an essential part of Oceanography for many years. Connectivity between underwater sensors, surface vessels, submarines, Autonomous Underwater Vehicles (AUV) etc is required for many scientific, commercial and military applications such as ocean monitoring and target tracking. Such networks primarily use acoustic modems to provide point-point communication between nodes, since radio transceivers have severely limited propagation range in sea-water. When more than two nodes are present in the same geographical region as a network, media access control (MAC) becomes a key challenge due to the low data rates of the acoustic communication links, large propagation delays as compared to terrestrial radio networks, high error rates and channel variability. Node mobility also can add to the challenge if AUVs etc are present. MAC for UANs have been actively researched for many years. This thesis presents novel research findings on time-domain and MACA-based (Medium Access with Collision Avoidance) MAC protocols for UAN MAC.

1.1 Background and Motivation

At the start of the work towards this thesis, there were many outstanding questions pertaining to the ad hoc UAN MAC problem. It was an active area of research with diverse and at times conflicting views on protocol choices and their relative merits, some of which is elaborated in Chapter 2. There were no published experimentally verified performance results on metrics such as normalized throughput and waiting time for reliable transfer (these metrics are defined in Chapter 4) for protocols such as MACA in UANs. Some experimental results (Rice et al., 2000) were available, but not sufficient to ascertain the metrics as mentioned above. In most cases, simulation results were also not obtained through independently verifiable simulation platforms. In some papers, mathematical modelling of proposed protocols was done, but had many shortcomings. For e.g., many papers omitted the relationship to system parameters such as detection and decoding probabilities and only modelled collisions (a more complete review is provided in Section 4.1). Most importantly, few protocol proposals had been validated through sea-trials.

Many papers were published on problems similar to those addressed in this thesis, during the same period (for e.g., Peleato and Stojanovic (2006); Peng and Cui (2006); Kredo et al. (2009); Freitag et al. (2005); Shusta et al. (2008), which will be reviewed in the related chapters). Novel ad hoc UAN MAC protocol models and simulation based results were published, some of which were based on the MACA-protocol family, the fundamental protocol model used in this thesis. Apart from novelty in protocol refinements, no experimentally verified performance results were provided, nor were the simulation results obtained through an independently verifiable open simulation platform. Mathematical modelling of such proposals also continued to have some limitations as mentioned earlier. Thus, the methodology used could not assure that the protocols were guaranteed to work in a similar manner in a sea-trial using acoustic modems[1].

MAC protocol standardization for UANs is going to be vital in future. In terrestrial radio wireless networks, standardization has been in practice for decades. However, most of the UANs around the world have been setup in isolation and use proprietary hardware and protocols. Up to the year 2006, relatively few attempts were made towards standardization in UANs and some existing standards included (Freitag and Singh, 2000; Stokey et al., 2005; Chitre et al., 2006), which are briefly discussed in Section 2.4. Even today there are no universal UAN MAC protocol standards, but many research groups are putting much effort towards this goal. The 2009 Janus workshop[2] at NURC (NATO Undersea Research Centre), was one such initiative aimed at UAN MAC standardization. The workshop highlighted that there is no consensus on the best choices for UAN MAC and a more reliable body of work was required on MAC protocols to aid the standardization process.

[1] There were known instances where a published and acclaimed UAN MAC protocol failed when tried in real sea-trials, due to unaccounted characteristics of the actual underwater channel in the simulations (see discussion on T-LOHI in Section 8.4.4). Many other published UAN MAC protocols have possibly never been tested in sea-trials using acoustic modems.

[2] http://www.januswiki.com/

1.2 Objectives

The fundamental objectives behind this thesis were thus born out of the above background in the UAN MAC domain. Clearer answers were needed on the best protocol choice for ad hoc UANs, primarily for the distributed topology. Some earlier works had given a strong hypothesis that favoured MACA-based protocols for UAN MAC (Rice et al., 2000; Shahabudeen and Chitre, 2005). However, much more in depth insights were needed to understand the various UAN MAC options. Reliable and more comprehensive performance results had to be obtained on chosen protocols, based not only on simulations but also experiments and accurate mathematical modelling. Chosen protocol models also needed to be refined and enhanced to improve performance, for the severely challenged UAN environment.

To aid the UAN MAC standardization process, suitable protocols for a universal UAN scenario had to be evaluated. Since there is certainly no single solution to the diverse requirements of a general UAN, adaptive protocol suites need to be explored. Reliable performance measures are also needed.

If very good time synchronization is possible, suitable TDMA based protocols can be viable as discussed in Chapter 3. Though there are many other challenges for TDMA protocols such as providing ad hoc functionality, scalability and robustness (to time synchronization errors), they still provide a useful performance benchmark for the other time-domain protocols. TDMA has been successfully used in small non-ad hoc UANs (Rice et al., 2000). Most UAN MAC protocols aim to mitigate the effects of propagation delay. Propagation delay could offer a different strategy for UANs since it enables spatial multiplexing. This possibility also had to be explored.

Most importantly the protocols have to be tested at sea, transferring data successfully in a real ad hoc UAN. New simulation and implementation software platforms need to be developed to enable such combined evaluation – simulation as well as experiments. One of the reasons why sea-trials were usually not done was the fact that simulation platforms were not linked to acoustic modem software. Simulated protocols need to be ported to modem platforms, incurring substantial manpower costs and other problems (discussed in Chapter 8). Modem sea-trials are also not easy to carry out frequently. Software frameworks for seamless simulation and deployment were practically non-existent at the starting period of this work, but since then other researchers have started to address this problem as well (Shusta et al., 2008). Thus, what was required is a simulation platform that emulates a real acoustic modem and allows the same code to be used for both simulations and deployment. With occasional validation of protocols at sea-trials, even purely simulation based results on such platforms can be relied upon, since they are guaranteed to work at sea in a similar manner.

The focus in this thesis is on short (50m to 500m) to medium range (500m - 5km) UANs. The OFDM modems built by the NUS Acoustic Research Laboratory (ARL, URL) have about 2 kilometre range in tropical shallow waters, such as in Singapore coastal waters. Many practical AUV networks etc are low propagation delay ($\approx 1s$) networks. In a general and universal UAN scenario, it is not easily possible to have stable and scalable time synchronization. The focus of this work throughout has been practical UANs and hence protocols not critically dependent on time synchronization were preferred. At the same time, the performance of the proposed protocols must improve if there is time synchronization. Other critical requirements include ad hoc functionality (node arrivals and departures), scalability and robustness (especially with respect to time synchronization).

1.3 Methodology

In this thesis, the methodology used involves a combination of simulations, mathematical analysis and sea-trials. The simulations are done in a software framework that allows the same code to run in the acoustic modem for sea-trials. Reliable-transfer-based metrics with normalization with respect to the physical layer data rate are used. Correct queueing theory based analysis for metrics such as waiting time is also used.

In terms of performance metrics, many papers use non-reliable data transfer based metrics. Metrics such as throughput, if defined for unreliable transfer, lose their significance substantially. If there are no acknowledgement (ACK) based retries to ensure reliability at the data link (MAC) layer, this overhead will have to be done at the network layer and will give overestimated and misleading performance results at the MAC level. An effective measure of UAN MAC protocol performance is normalized throughput, where the normalization with respect to the physical layer one-way data rate helps to isolate the performance of the MAC protocol. Without such normalization, it is hard to compare results for UAN MAC protocols, which have been implemented on different physical layers. Queueing analysis can be used to capture the waiting time behaviour of these protocols.

1.4 Outline

There have been significant advances in UANs over the last few decades. In Chapter 2, some of the key developments in data link layer (DLL), MAC and routing protocols are reviewed, to provide a brief overview of existing body of work in the domain of this thesis and help identify unsolved problems and issues in UAN MAC. Based on the observation from the review that there were differences of opinion on the best choices for UAN MAC, Chapter 3 explores the design choices for MAC in a UAN primarily for a distributed topology. The various protocols explored in research until then were assessed and their relative merits for use in a UAN evaluated. A key question was whether time domain and specifically MACA and related protocols are effective choices for UANs.

After establishing the usefulness of MACA-based protocols for UANs, Chapter 4 presents a mathematical analysis of a MACA-based protocol with batch data for ad hoc underwater acoustic networks (UAN). Sea trial results corroborated simulation and analytical results. Various novel protocol enhancements to improve performance are also proposed. Since a single protocol cannot meet all the requirements of a more general UAN scenario, Chapter 5 looks at a proposal for an adaptive multi-mode MAC protocol suite (MAC-AMM) for use in a heterogeneous underwater network. The chapter uses mathematical models and simulations to characterize three modes – centralized MACA-C protocol, distributed MACA-EA protocol and a DATA-ACK protocol (the term DATA shall be used for data packets) and show how a combined adaptive scheme can be useful in UAN MAC. This protocol suite is aimed as a candidate for UAN MAC standardization.

Much of the work in this thesis looks at the UAN MAC problem where time synchronization is not necessarily available. However, as mentioned earlier, TDMA continues to provide a benchmark for time-domain protocol performance, despite its many limitations in providing ad hoc functionality, scalability and robustness (due to time synchronization errors). Chapter 6 relooks at TDMA and the fascinating possibility of utilizing the propagation delay in networks more constructively than traditionally done. It outlines the "SuperTDMA" concept and a new performance target for future UAN MAC protocols. Three new MAC protocol variants that utilize this "SuperTDMA" concept are presented – Twin-TDMA, Dynamic Twin-TDMA and Twin-ALOHA.

Chapter 7 presents results from a study on using multiple communication channels simultaneously for effective networking in a small AUV network. Chapter 8 presents the unified simulation and implementation software framework that was used for most of the UAN MAC protocol evaluation described in this thesis. And finally, Chapter 9 summarizes the observations, insights and conclusions of this thesis. It also outlines some of the potential future work.

1.5 Novel Contributions

This is a summary of the key novel contributions in this thesis.

- Insights into why time domain MAC schemes are best suited for most of the UANs and why MACA-based schemes and Dynamic-TDMA based schemes are good choices in UAN MAC

- An accurate Markov chain based analytical model for a novel MACA based protocol (MACA-EA) for UANs has been obtained. A queuing analysis for reliable transfer of batch data for the MACA-EA protocol has been done. The analysis shows that an optimal batch size minimizes the total waiting time, whereas increasing the batch size arbitrarily, maximizes the throughput

- A comprehensive adaptive multi-mode MAC protocol termed MACA-AMM to address a heterogeneous UAN has been proposed. The thesis presents analytical characterization of the DATA-ACK protocol mode with queueing of incoming data and reliable communication with retries, analytical characterization of the centralized MACA-C protocol mode for reliable communications in UANs and waiting time comparison between TDMA, MACA-EA, MACA-C and DATA-ACK modes. It illustrates the transition traffic intensity between MACA-EA and DATA-ACK modes for minimizing waiting time

- A novel, state-dependent DATA-ACK protocol for low traffic intensity, MACA-SEA, a self-sequencing variant of MACA-EA that achieves a pseudo-TDMA behaviour adaptively and a position based multi-channel extension to a MACA-based MAC protocol have been presented

- An ARQ protocol variation (Early-Multi-ACK) for efficient and reliable data transfer. The throughput performance of this batch-mode MACA-EA protocol is better than other proposed ad hoc protocols for UANs

- A part of work on interference-alignment-based time domain scheduling for achieving $N/2$ throughput in N-node UANs, and Twin-TDMA and Twin-ALOHA protocols that utilize simultaneous transmissions to achieve better performance than their counterparts TDMA and ALOHA

- Unified software framework for UAN MAC protocol simulation and deployment. This enabled comparative sea trial results that validate that the proposed protocols can indeed be implemented in a real network, and that its performance will similar to the analytical predictions, even in cases where sea-trials are not conducted

Chapter 2

Literature Review

There have been significant advances in underwater networking over the last few decades. This section reviews some of the key developments in DLL, MAC and routing protocols, to provide a brief background on the existing body of work in the domain of this thesis up until 2008 (with a few exceptions). It is also aimed at highlighting gaps in ongoing research that existed at the early period of this work that helped identify unsolved problems and issues in UAN MAC to pursue for this thesis. Most of the content in this literature review has been published (Chitre et al., 2008). More recent and specific review for each chapter will be provided within the chapter itself.

2.1 Media Access Control

In static MAC protocols, nodes are allocated predetermined data channels, are contention-free (also referred to as scheduled or deterministic protocols). Static protocols are inherently non-scalable. In dynamic and ad-hoc schemes, nodes typically use a shared control channel over which data channels are requested. Two main MAC topologies used are centralized or distributed. In centralized topology (also referred to as clustered, cellular etc) a master node controls media access for nodes in its neighbourhood. In a distributed topology, there are no controlling master nodes and all nodes asynchronously handles data transfers. Dynamic MAC protocols in distributed topology are contention-based. In centralized topology, they could also use polling methods with no contention.

2.1.1 Static protocols

Contention-free static MAC protocols include TDMA, FDMA, and CDMA. Space division multiple access (SDMA) is rarely used. Among these, a general consensus in underwater network research is that FDMA is inefficient for underwater applications (Rice et al., 2000). TDMA has been reported to be better in some aspects but requires good time synchronization in nodes (Sozer et al., 2000). In some publications, CDMA is favoured over TDMA and FDMA (Proakis et al., 2001; Jun-Hong et al., 2006; Chan and Motani, 2007). Akyildiz et al. (2006) favours CDMA over TDMA and FDMA. PCLS, a loosely synchronized form of TDMA with non-overlapping timeslots, has been proposed for low capacity sensor networks (Turgay and Erdal, 2006). At a fundamental level, there is still some difference of opinion on the best MAC protocol choice for UANs.

2.1.2 Dynamic contention-based MAC

Some of the simpler contention-based distributed protocols include half duplex ALOHA, carrier sense multiple access (CSMA) and MACA using Request to Send (RTS)-Clear to Send (CTS) handshaking (Sozer et al., 2000). Smith et al. (1997) describes a CSMA based contention protocol. MACA based protocols use RTS, CTS, DATA, ACK sequences and were shown to be effective for underwater use compared with scheduled protocols early on in the Seaweb project (Rice et al., 2000). The authors observe that in the physical and MAC layers, adaptive modulation and power control are the keys to maximizing both channel capacity and channel efficiency and RTS/CTS handshaking permits that, along with addressing, ranging and channel estimation. MACA based protocols are found to be highly suited in many scenarios underwater where scalability is important and time-synchronization is not available (Kebkal et al., 2005; Molins and Stojanovic, 2006; Peng and Cui, 2006; Heidemann et al., 2006). However in some sensor networks, RTS/CTS mechanisms could perform poorly due to propagation delay issues and inefficiency for small payload packets (Turgay and Erdal, 2006). Protocol extensions and enhancements of MACA have been investigated to suit them better to underwater channel. For example, Doukkali et al. (2006) and Sozer et al. (2000) investigated a WAIT command extension. A WAIT command is sent back by the receiver if it is currently busy and intends to send a CTS later on. Rice et al. (2000) uses a selective ARQ, initiated by recipient should it not receive packets in a specified time instead of using ACK packets. Guo et al. (2006) proposed to counter the wasted bandwidth in handshaking due to the high propagation delay, using a variant called PCAP which pipelines other actions while waiting for CTS from receiver. Packet trains have been shown to greatly improve the performance of protocols such as MACA (Molins and Stojanovic, 2006; Garcia-Luna-Aceves and Fullmer, 1998). Some analytical results for optimal packet size as a function of the acoustic link parameters (transmission rate, link distance, and error probability) and the train or group size have been presented (Stojanovic, 2005). Floor acquisition multiple access (FAMA), a family of protocols of which MACA is a variant, was originally proposed for terrestrial networks. It uses carrier sensing (absent in MACA) and puts restrictions on RTS/CTS time durations. Time-slotting can also be implemented to enhance performance (Fullmer and Garcia-Luna-Aceves, 1995). FAMA in its original form is quite unsuited to underwater networks, but with enhancements such as slotting, it can be used underwater effectively (Molins and Stojanovic, 2006). Distance aware-collision avoidance protocol (DACAP) is also based on MACA (Peleato and Stojanovic, 2006). It adds a warning message if a RTS is overheard while waiting for a reply to its own RTS. While waiting for reply, if another CTS or a warning is heard, a random back-off is used. Dolc and Stojanovic (2007) looks at optimal power control for DACAP. The optimal power is found to be that which minimizes connectivity.

One of the observations was that performance results were often not conclusive. Most were purely simulation based (Petrioli et al., 2008), and some were purely numerical (Peng and Cui, 2006). Cross verification between simulations, numerical analysis and sea-trials for most of such protocol proposals have not been presented. Exceptions include the Sea-web experiments where some experimental results on RTS/CTS based protocols were presented (Rice et al., 2000).

More specific reviews on MACA-based protocols in the context of the MACA-EA protocol presented in this thesis is provided in Section 4.1.

2.1.3 Conflicting opinions and results

Some of the conflicting views were mentioned above in Section 2.1.1. There were other views against TDMA protocol whilst favouring FDMA. For example, Doukkali and Nuaymi (2005) say "TDMA techniques are generally not suitable in the case of underwater communications". There we were also views against RTS/CTS based protocols as well, in contrast to some of the positive assessment stated above. For example, Jun-Hong et al. (2006) say "it has been observed that contention-based protocols that rely on carrier sensing and handshaking are not appropriate in underwater communications". Some papers also refuted another's claims of good performance (for example, PCAP performance (Guo et al., 2006, 2009) vs. (Petrioli et al., 2008)). Such difference of opinion provided the motivation for a thorough investigation of MAC protocol choices for UANs (presented in Chapter 3).

2.1.4 Dynamic contention-free MAC

The polling-based protocol called FAMA-CF uses request for RTS (RRTS), RTS, CTS, DATA, ACK handshaking to communicate with the central node (Kebkal et al., 2005). The central node initiates the RRTS to its peers. This is closely related to the centralized MACA-C protocol described in Section 5.2.4. Salva-Garau and Stojanovic (2003) looks at one CDMA code per cluster and spatial re-use of codes. TDMA is used within each cluster. Nodes are assumed to be able to handle multiple CDMA codes simultaneously. Casari et al. (2007) show a similar scheme in which clusters are allocated either different CDMA codes or FDMA bands and within each cluster TDMA is used. Since cluster heads are tasked with TDMA slot allocation to ordinary nodes, the above can be classified as centralized MAC topology. An underwater acoustic cellular network is an extension of centralized topology. Analysis of frequency re-use between adjacent clusters and optimal cell-radius selection criteria has been carried out (Stojanovic, 2002). Peleato and Stojanovic (2007a) present a related work on channel allocation and scheduling protocol for underwater cellular networks.

2.2 Energy Conservation

A key DLL/MAC aspect is energy conservation. PCLS incorporates a power control and sleep-wake up scheme (Turgay and Erdal, 2006). Rodoplu and Min Kyoung (2005) present another example on energy minimization and show an ultra-low duty cycle MAC protocol focusing on energy conservation at low data rate. A sensor wakeup scheme – adaptive wakeup schedule function (AWSF), suitable for underwater sensor systems uses a time cyclic wakeup schedule for each node such that at any one time only a few nodes are active (Wong et al., 2006).

2.3 AUV Networking

Due to ever increasing applications, networking of mobile AUVs is currently a very active area of research. The mobility and ad-hoc requirements for such networks pose many challenges. Stojanovic et al. (2002) describe a TDMA protocol for AUVs. Exchanged packets contain position information for localization. Molins and Stojanovic (2006) presented simulated results from a FAMA based MAC for an AUV network. AUVs are sometimes equipped with multiple modems (Freitag et al., 2005). One of the author's own work explores a related concept of multi-channel communications in AUV networks and is elaborated in Chapter 7 (Shahabudeen et al., 2007) .

2.4 Standardization and Software Frameworks

Some of the existing standardization initiatives are reviewed here. The physical layer of the WHOI micro-modem was published as a standard (Freitag and Singh, 2000). A commercial modem maker Benthos implemented compatible modems to this standard (Freitag et al., 2005). Standardized communications to acoustic modems includes the WHOI micro-modem that supports a standard NMEA 0183 protocol (Freitag et al., 2005). At a higher layer, the command and control language (CCL) specifications for AUV networks outline a TCP/IP based protocol for access to a CCL gateway (Stokey et al., 2005). MAC standards are being attempted across the world(McCoy, 2009). Chitre et al. (2006) presented a framework called Underwater Network Architecture (UNA) and a Framework API

(FAPI) aimed at unifying programming interfaces between layers in a UAN protocol stack. Shusta et al. (2008) presented one of the earlier proposals for a unifying software framework, and Guerra et al. (2009) presented another work towards simulation and testing frameworks for UAN MAC. Otnes et al. (2009) presented a roadmap for this process and Petroccia et al. (2011) presents one of the latest results in this area, where a ns2 based simulation system makes it easy for developers to take protocol implementations to sea trials. However, standardization in underwater networks is still in its infancy compared to terrestrial networks. A brief review of software frameworks is given in Section 8.1.

2.5 Conclusion

This brief review is meant to provide a background for the work presented in the thesis. A large body of work exists on UANs. However there were many ongoing debates on the relative merits of MAC protocols for UANs and the performance of ad hoc protocols such as MACA and its variants, which had to be addressed during the time frame of the work that is summarized in this thesis. MACA-based protocols also remained open to further novel enhancements to improve performance. The existing body of work on UAN MAC protocols were also largely simulation based and few experiments had been reported on MACA-based protocol performance measurements using actual sea-trials. Detailed mathematical analysis of MACA-based protocols for metrics such as normalized throughput and waiting time for reliable data transfer in UANs was also not presented. Standardization of ad hoc UAN protocols is also in its infancy, and much could be done on providing a strong proposal towards the process. Open protocol development software frameworks were also non-existent and provided another key area for novel contributions. In the subsequent chapters, updated and more specific reviews with respect to the chapter's content will be provided.

Though lack of sea-trials were mentioned in this and the previous chapter as a shortcoming in many of the papers on UAN MAC, it should be noted that some research groups (WHOI, NPS etc) have actively been doing UAN experiments involving MACA-based protocols for more than a decade. They have immense sea-trial capabilities. Based on available publications, relatively few of these experiments have been used to characterize protocols such as MACA, using the normalized throughput and waiting time metrics for reliable transfer as undertaken in this thesis, through a unified simulation-analysis-sea trials methodology.

Chapter 3

Investigation of MAC Protocol Choices for UANs

A key problem that any underwater acoustic network (UAN) has to address is that of medium access control (MAC) – how do multiple nodes coordinate access to the acoustic channel? As we saw in the previous review chapter, there have been numerous MAC proposals and some conflicting views on what is best for UAN MAC. In this chapter various possible MAC options are evaluated to provide useful insights into the selection of MAC for UANs. This work was completed in 2008 and published as a book chapter (Shahabudeen et al., 2010).

3.1 Introduction

The MAC problem for UANs is conceptually similar to the MAC problem for terrestrial radio wireless networks. However, there are some key differences between the two problems: larger propagation delay due to low sound speed, extremely low point-to-point data rates and high raw bit error rate (BER). Thus, researchers often suggest that MAC protocols for UANs should developed from the ground up and not directly adopted from existing terrestrial protocols (Heidemann et al., 2006; Jiejun et al., 2005).

The overall networking problem includes MAC, multi-hop routing, reliability, data transfers to the wider Internet infrastructure, etc. In this chapter, the focus is only on the MAC layer. Two of the most relevant topologies for UANs are distributed and centralized. There are other specialized topologies such as linear, ring etc that are not considered, as they are only applicable in special situations in UANs. In order to avoid collisions, code, frequency or time division can be used to separate logical communication channels. This leads to the common multiple access schemes – code division multiple access (CDMA), frequency division multiple access (FDMA) and Time Division Multiple Access (TDMA). Space division multiple access (SDMA) is rarely used as it is impractical to implement it except in special scenarios. *Channelization* is defined as the process of dividing the total available channel capacity in the acoustic channel into a set of logical *channels* for the purposes of multiple access and spatial re-use. Channelization can either be static or dynamic. Spatial re-use of channels is important in UANs since they are both power and bandwidth-limited and consequently capacity-limited. UANs could also employ either static or dynamic *channel allocation* to associate nodes with channels. The channel allocation schemes used by MAC protocols can be categorized into two basic types – contention based (the term contention or random access is used interchangeably) or contention-free (no possibility of packet collisions).

The aim of this chapter is to provide important insights into the MAC problem and options available to the designer of an UAN. Some of the results from the terrestrial wireless domain are re-visited to evaluate its applicability to UANs. It is important to have a holistic picture of the various aspects of the problem so that the designer can make an informed choice based on the exact requirements of the network being set up.

The rest of this chapter is organized as follows. The remaining part of this section establishes some basic concepts and terminology. In Section 3.2, the different channelization options of time, frequency and code division are compared to illustrate why time division is a good choice for UANs. Medium access collision avoidance (MACA) based protocols (Karn, 1990) are illustrated as a special case of dynamic TDMA, and are shown to address the UAN MAC problem well in Section 3.2.4. Some of the other important aspects of MACA based protocols in UANs are also then discussed.

3.1.1 Topology

In a *distributed topology* (sometimes referred to as peer-to-peer), there are no controlling central nodes and all nodes asynchronously and equally handle MAC functionality. All nodes are deemed equal in the MAC function. The IEEE 802.11 distributed coordination function (DCF) in ad hoc mode is a distributed protocol used by WLAN networks (Bianchi, 2000). 802.11 DCF protocol can be used for peer-to-peer networks in ad hoc mode.

In a *centralized topology* (also referred to as clustered, cellular etc), the central node controls medium access for nodes in its neighbourhood and is tasked with allocation of data channels to client nodes. For example, in a cellular wireless network, only the base station (BS) has the MAC and routing gateway functionalities. In such a network, ordinary nodes only communicate with such BSs. 802.11 point coordination function (PCF) is a centralized topology protocol and the access point (AP) polls and allocates channels to clients (Bianchi, 2000). 802.11 PCF is implemented on top of DCF. The AP acquires channels in a distributed fashion among its peers and utilizes the acquired channels for nodes under its PCF control. So it can be viewed as a centralized topology within a distributed topology.

A *cell* is defined as an area under the control of one central controller in centralized topology. In a distributed topology there are no defined cells, but as nodes get farther apart they will not interfere with each other due to signal attenuation.

Figure 3.1 shows a possible UAN. This network consists of two cells, one using a centralized topology (all communications are between sub-sea nodes and a central surface BS) and the other using a distributed topology (there are links between sub-sea nodes directly). The UAN cells in this example are interconnected via surface radio and to the wider Internet. Centralized topology is a good choice when at least one node in the cell has high-speed connectivity to other cells. However, in many UANs, all the nodes have similar communication capabilities.

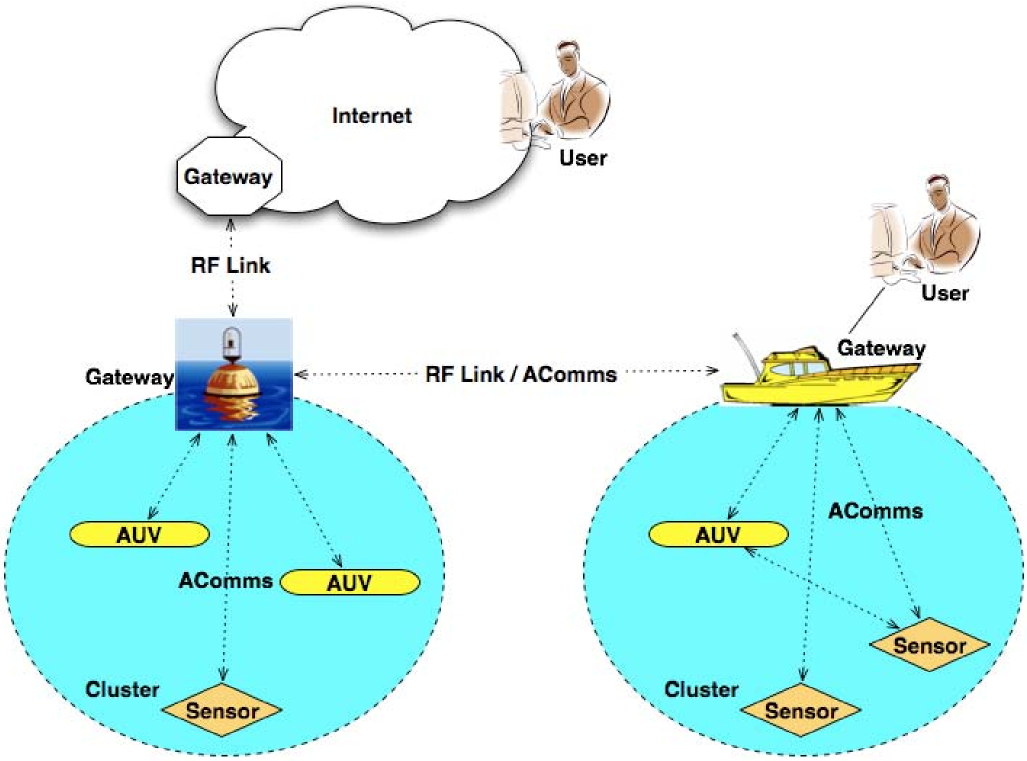


Figure 3.1: An example UAN architecture with two cells, one using a centralized topology while the other using a distributed topology (Chitre et al., 2008)

3.1.2 Spatial re-use, channelization and allocation

In a centralized topology, the total available channel capacity has to be divided spatially between cells for re-use purposes. If the network spans a geographic area larger than single hop range of a single node, channels can and need to be re-used due to limited channel capacity. This is commonly done in cellular networks. Re-use patterns are designed to allow for maximum permissible co-channel interference. Re-use patterns and inter-cell channelization can be achieved dynamically or statically. In 802.11, 11 fixed frequency bands are allocated for use in cells (only three are actually orthogonal) and a given AP uses a fixed frequency band. Thus, there is a static frequency band based channelization between cells. In certain cellular networks, there are provisions for dynamic inter-cell channelization through concepts such as channel borrowing between base stations supervised by a mobile switching centre (MSC) (Rappaport, 2002). In distributed topology, there are no explicit "cells" but the network has to effectively do some form of channelization and spatial reuse of channels over large areas and this is further discussed in Section 3.2.5.

Within a cell, the channel capacity needs to be divided for multiple access. In many networks, the channelization within a cell is essentially static. Cellular wireless systems such as GSM, allocate pre-defined TDMA time slots and frequency bands to users upon request and hence can be considered static channelization within a cell. In cellular systems, multiple channels also can be allocated to the same user. For example, HSCSD mode in GSM and can be considered a form of dynamic channelization within a cell. Note that the static channels created by the channelization are dynamically allocated to different users, so the channel allocation is dynamic. In distributed topology protocols such as 802.11 DCF, the length of data can be varied. This in effect varies the allocated time slot and can be considered as a form of dynamic TDMA channelization within a cell.

In static channel allocation protocols, nodes are permanently allocated predetermined channels (also referred to as scheduled or deterministic protocols). In dynamic channel allocation protocols, one option is to use pure random access (such as ALOHA), i.e. with no explicit channel reservation. Another option is to use random access only for the control channel; the *control channel* is a shared channel that all nodes can send and listen to, and use to make reservations for *data channels*. For example, in dynamic distributed topologies, nodes can request for data channel using a random access protocols (on the control channel) and are assigned a data channel by the recipient. As another example in dynamic centralized topologies, nodes can send requests for data channel to the central controller using a random access control channel. In dynamic centralized networks, a third option is to have no contention even in the control channel, for example the central controller can poll the clients over a control channel. As discussed above, the allocated data channel can be statically channelized or dynamically channelized (as in HSCSD in GSM or 802.11 DCF).

General UANs also require bi-directional communications. In a distributed topology, there is bi-directional symmetry and no additional considerations are required. In centralized topology, the options are duplexing in time, code or frequency. However, most existing UAN physical layers are half duplex due to practical considerations. This has a significant impact on design choices in UANs and is discussed further in Section 3.2.2.1.

3.1.3 Need for dynamic channelization and allocation in UANs

Channelization of any kind implies division of some allowed maximum channel capacity. Maximum usable frequency bandwidth and hence the channel capacity could be restricted by regulation and/or other physical layer requirements. In inter-cell channelization, if base stations use only one fixed channel as in 802.11, then it implies usage of only a portion of available capacity by a certain cell. If there are no neighbouring cells, such a situation is not good for capacity starved

UANs. Therefore, inter-cell channelization needs to be dynamic.

In simple UANs, where a fixed number of nodes are deployed and there are sufficient channels, one can use static protocols regardless of utilization level in each channel. Static TDMA, FDMA or CDMA protocols can easily be used for small static networks based on the capabilities of the underlying physical layer.

In case of a large number of nodes, a channel per node requirement could yield very low data rate channels. If each node only transmits occasionally, the total capacity could be used more efficiently by having a smaller number of high data rate channels. If there are more nodes than there are channels, once all the channels are allocated, further requests are blocked. Channels need to be provided to all users without blocking. Protocols therefore need to dynamically channelize the intra-cell channel capacity also and allocate channels on demand. Capacity of a single channel needs to be adaptively varied according to number of nodes. Channels should be allocated only for short periods of time (this is a form of channelization in time) and nodes need to contend for them repeatedly.

Another reason to consider dynamic allocation in MAC protocols is the need to allocate channels on demand in response to arrivals and departures of nodes. This is the case of cellular mobile networks with client nodes wanting to make calls. In UANs, such cases could arise in AUV networks.

3.2 Selection of MAC Protocols for UANs

Based on the concepts discussed in the previous section, the selection of an effective MAC protocol for use in UANs is outlined below. Dynamic channelization, channel allocation and the selection of appropriate topologies are all important issues to be addressed. But firstly, the appropriate channelization – code, frequency or time division, needs to be selected. There has been much discussion about the relative merits of CDMA, FDMA, TDMA and protocols such as MACA (Proakis et al., 2001). Shahabudeen and Chitre (2005) had earlier given a strong indication that MACA-based protocols were good for UAN MAC, but this needed a more fundamental insight. These channelization options are compared to illustrate their relative merits.

3.2.1 The general equivalence of static TDMA, FDMA and CDMA

Static FDMA, TDMA and CDMA effectively provide same unidirectional data rate performance in an ideal case. To make comparisons on equal terms, assume that the system has a fixed maximum bandwidth b. The average power consumption over time must be the same for all systems for fair comparison and to that effect maximum transmission power P_T per unit bandwidth is assumed. Let P_R be the received power per unit bandwidth. Another assumption is Gaussian ambient noise with power N_0 per unit bandwidth. The data is transmitted in a time window of length t. Let C be the number of user channels and R be the data rate. Let ς be the received signal-to-noise ratio. The received signal to noise ratio over the bandwidth b is the ratio of the received power $P_R b$ and the ambient noise power $N_0 b$. For simplicity, it is assumed that all links experience equal frequency-independent path-loss ratio β and hence $P_R = \beta P_T$ remains a constant.

The theoretical capacity for a given additive white Gaussian noise (AWGN) communication channel is given by Shannon's law as

$$R = b\log_2(1+\varsigma) \tag{3.1}$$

This law is used to illustrate the equivalence of R from an information theoretic standpoint. First let's compare TDMA with FDMA. For a TDMA system, each user time slot is t/C time units long and uses the entire bandwidth b at maximum power Pt. Therefore the effective data rate per user over time t is

$$R_{\mathrm{TDMA}} = b\log_2\left(1+\frac{P_R b}{N_0 b}\right)\frac{t}{C}\frac{1}{t} = \frac{b}{C}\log_2\left(1+\frac{P_R}{N_0}\right) \tag{3.2}$$

For a FDMA system, each user time slot is t time units long and uses b/C units of bandwidth at maximum power P_T. The effective data rate over time t is

$$R_{\mathrm{FDMA}} = \frac{b}{C}\log_2\left(1+\frac{P_R\,(b/C)}{N_0\,(b/C)}\right) = \frac{b}{C}\log_2\left(1+\frac{P_R}{N_0}\right) \tag{3.3}$$

This is the same as R_{TDMA} and is the same equivalence embodied in the familiar concept of fixed time-bandwidth product. This implies that in a fixed amount of time t, C users using C bands of b/C get the same average data rate as C users dividing t into C slots of t/C and each using the entire bandwidth b.

Now consider the ideal CDMA utilizing a total bandwidth b after spreading. The spreading factor κ expands the bandwidth by κ. To be able to compare on equal terms, a fully loaded orthogonal CDMA system with $\kappa = C$ is used. The bandwidth per user is b/κ units before spreading. Considering orthogonality of CDMA codes, i.e. multiple access interference (MAI) is zero (requires perfect time synchronization), Shannon's law for the de-spread channel can be re-written as

$$R_{\mathrm{CDMA}} = \frac{b}{\kappa}\log_2\left(1+\frac{P_R\,(b/\kappa)}{N_0\,(b/\kappa)+MAI}\right) = \frac{b}{\kappa}\log_2\left(1+\frac{P_R}{N_0}\right) \tag{3.4}$$

If $\kappa = C$ for a fully loaded synchronous orthogonal CDMA,

$$R_{\text{CDMA}} = \frac{b}{C} \log_2 \left(1 + \frac{P_R}{N_0}\right) \tag{3.5}$$

Thus, this is exactly the same as the FDMA and TDMA ideal cases. Such equivalences have been discussed in prior publications (Ipatov, 2005).

It is interesting to note that channelization using CDMA or FDMA also requires division in time as all transmission frames are finite in time. In other words, TDMA can be considered as a fundamental division mechanism for all channels. FDMA and CDMA could be viewed as further orthogonal divisions in frequency or code space.

3.2.2 General strengths and weaknesses of CDMA, FDMA and TDMA

The above comparison between TDMA, FDMA and CDMA are true under some ideal assumptions. TDMA, for example, has inefficiencies arising due to required guard periods between slots to compensate for imperfect clock synchronization between nodes, clock drift and propagation delay. FDMA requires guard bands between bands, since infinitely sharp cut-off filters are physically impossible to realize.

In CDMA, the above result holds only when all chips are synchronized and mutually orthogonal. It works well in downlink synchronous transmissions in terrestrial radio wireless networks. In high propagation delay UANs, however, this is not easy to achieve. There will be interference between channels due to non-orthogonality of codes arising mainly from latencies. Typical CDMA analysis also assumes perfect power control. The classical near-far problem will otherwise create large MAI from near-by sources and performance will degrade.

Considering MAI from M other users (based on published analysis (Rappaport, 2002)), for asynchronous CDMA, Shannon's law for the de-spread channel can be re-written as

$$\begin{aligned} R_{\mathrm{CDMA}} &= \frac{b}{\kappa}\log_2\left(1+\frac{P_R\,(b/\kappa)}{N_0\,(b/\kappa)+M\,(P_R/\kappa)\,(b/\kappa)}\right) \\ &= \frac{b}{\kappa}\log_2\left(1+\frac{P_R}{N_0+M\,(P_R/\kappa)}\right) \end{aligned} \tag{3.6}$$

If $k = C$ for a fully loaded CDMA,

$$R_{\mathrm{CDMA}} = \frac{b}{C}\log_2\left(1+\frac{P_R}{N_0+M\,(P_R/C)}\right) \tag{3.7}$$

M can be at maximum $C-1$ in an C-node fully connected scenario. As M increases, MAI increases and performance degrades to less than the equivalent FDMA or TDMA protocol. Such degradation has been shown to make asynchronous CDMA (as would be the case in distributed topology underwater networks) less attractive than FDMA or TDMA in a fully connected system of C nodes (Ipatov, 2005).

An advantage of TDMA based protocols is that they provide flexibility in terms of implementation over any physical layer technology. As long as the MAC layer has access to transmit and receive behaviours of the physical layer, any underlying physical layer such as OFDM, FH-BFSK, etc may be used.

3.2.2.1 Full duplex requirement for CDMA and FDMA

In the case of cellular wireless networks using static FDMA or CDMA, the clients communicate with a multi-band or multi-code base station. Thus, all users can communicate in parallel with the base station, and vice versa. The general equivalence in Section 3.2.1 is thus valid.

Consider the case of C isolated nodes communicating with each other in a distributed topology. In TDMA, since only one node is active at a given time, it can transmit to any one or more of the $C-1$ neighbours. For FDMA or CDMA, one needs to assume that they are all transmitting simultaneously in orthogonal frequency or code bands to get same or similar performance as the TDMA. So unless all nodes can receive on all the channels (frequency bands or CDMA codes) while transmitting, there can be no receivers! This implies that all nodes are capable of receiving all bands or codes (except the one in which it transmits) in parallel and at the same time transmit i.e. each node is full duplex.

Thus, the key difference from TDMA is that nodes need to be able to receive while it is transmitting (full duplex). In CDMA this could amount to a near-far problem and reduce the performance. In FDMA extremely good inter-band filtering would be required to minimize bandwidth wastage due to guard bands. Typically packets use a detection preamble followed by data. So unless the preamble is also code or frequency band tunable, there will be interference from the preamble in code division and frequency division data channel methods. Smaller bandwidth for preambles reduces detection performance. Thus, such multi-band full duplex receivers are typically not used and most underwater modems available today are half-duplex. There has nevertheless been some work on full duplex systems in UANs (Jarvis et al., 1997).

3.2.2.2 CDMA performs better in terrestrial cellular networks?

As discussed earlier, CDMA can be shown to be inferior when used in a single cell or among fully connected nodes (Ipatov, 2005). However, CDMA is often reported to have performance gains in cellular networks with re-use considerations (Ipatov, 2005). In standard TDMA or FDMA based systems, a re-use pattern of 7 is commonly used. In CDMA, frequency re-use is not necessary (although sometimes used), and when using all CDMA codes in a single cell and considering co-channel interference from neighbouring cells, capacity gains over TDMA or FDMA of up to 5 times have been shown (Ipatov, 2005).

However, we take a closer look at some of this analysis to see if comparisons were made on equal terms and are appropriate for UANs. Ipatov (2005) uses a voice activity factor of about 3/8 in asynchronous CDMA capacity analysis, but do not apply it to to GSM or TDMA analysis. With b as the total bandwidth, and t as time window as defined in 3.2.1 (bt is the time-bandwidth product), and if K_{CDMA} is the number of channels available (and used) in a cell (for CDMA), using the analysis without the voice activity factor, the signal-to-interference-noise ratio (SINR) Ψ can be written as (Ipatov, 2005)

$$\Psi = \frac{2bt}{1.5K_{\text{CDMA}} - 1} \tag{3.8}$$

The above considers interference from 6 neighbouring cells. This gives an upper bound to the number of channels for a given SINR as

$$K_{\text{CDMA}} \leq \frac{4bt}{3\Psi} + \frac{2}{3} \tag{3.9}$$

If Ψ is 5 (i.e. 7 dB) (Ipatov, 2005)

$$K_{\text{CDMA}} \leq \frac{4bt}{15} + \frac{2}{3} \tag{3.10}$$

A comparison for an FDMA based system uses a re-use factor of 7, and the number of channels in a cell K_{GSM} is (Ipatov, 2005)

$$K_{\text{GSM}} = \frac{bt}{7} \tag{3.11}$$

Thus, with the voice activity factor discounted, K_{CDMA} is shown to be about 1.75 times K_{GSM} (assuming $bt >> 2/3$). It is true that CDMA has an advantage in being able to use the voice activity factor for capacity increase in voice traffic, as this cannot directly be done in FDMA/TDMA based systems. For data communications, the benefit of such a factor will depend on the burstiness of traffic.

However, in standard hexagonal based geometry analysis used for FDMA or TDMA based systems, a cluster size of 7 gives an SINR of 18dB (Rappaport, 2002)! Using 18 dB ($\Psi = 63$) as the criterion in the CDMA equation (3.9),

$$K_{\text{CDMA}} \leq \frac{4bt}{180} + \frac{2}{3} \tag{3.12}$$

This shows that under similar SINR requirements, the TDMA or FDMA system has much better performance than asynchronous CDMA. Perfect power control is also assumed in such typical CDMA analysis (Ipatov, 2005). CDMA performance significantly varies with power control errors also (Muratore and Romano, 1996).

For fourth generation (4G) OFDM based systems, TDMA has been considered a better multiple access option compared to FDMA and CDMA (Bisaglia et al., 2005). WiMAX, a popular new technology, uses orthogonal frequency division multiple access (OFDMA), where OFDM sub-carriers are assigned as channels to users, and can be interpreted as form of dynamic FDMA. OFDM has been shown to be better than CDMA in multi-path environments (Martoyo et al., 2002) and now seems to be the choice for 4G communications (Bisaglia et al., 2005). UAN modems using OFDM technology have also been developed (Chitre et al., 2005).

The above discussion is primarily to say that even in terrestrial cellular systems, CDMA's superiority is arguable. Each system has its own difficulties. CDMA eases re-use planning, but there are real world issues such as power control and code offsets between cells to separate users' pseudo-random number (PN) sequences (Muratore and Romano, 1996). Moreover, the primary focus of this chapter is on distributed topology and the full-duplex and multi-channel requirements in CDMA and FDMA make TDMA a better choice of channelization for distributed topology UANs.

3.2.3 Dynamic allocation protocols

Having established that TDMA fundamentally is a good choice for UANs in distributed topology, a dynamic variant (henceforth referred to as dynamic TDMA or D-TDMA) in order to address the dynamic channelization and allocation requirements that were set out at the beginning, can now be considered. This can be modelled using a frame with multiple slots, with a contention period between frames as shown in Figure 3.2(a). Nodes use random access during the contention slots (control channel) to request data slot (data channel) allocations in the frame. A channel is allocated for a certain number of frames (a system parameter). Once all the channels are allocated, further requests are not satisfied until channels

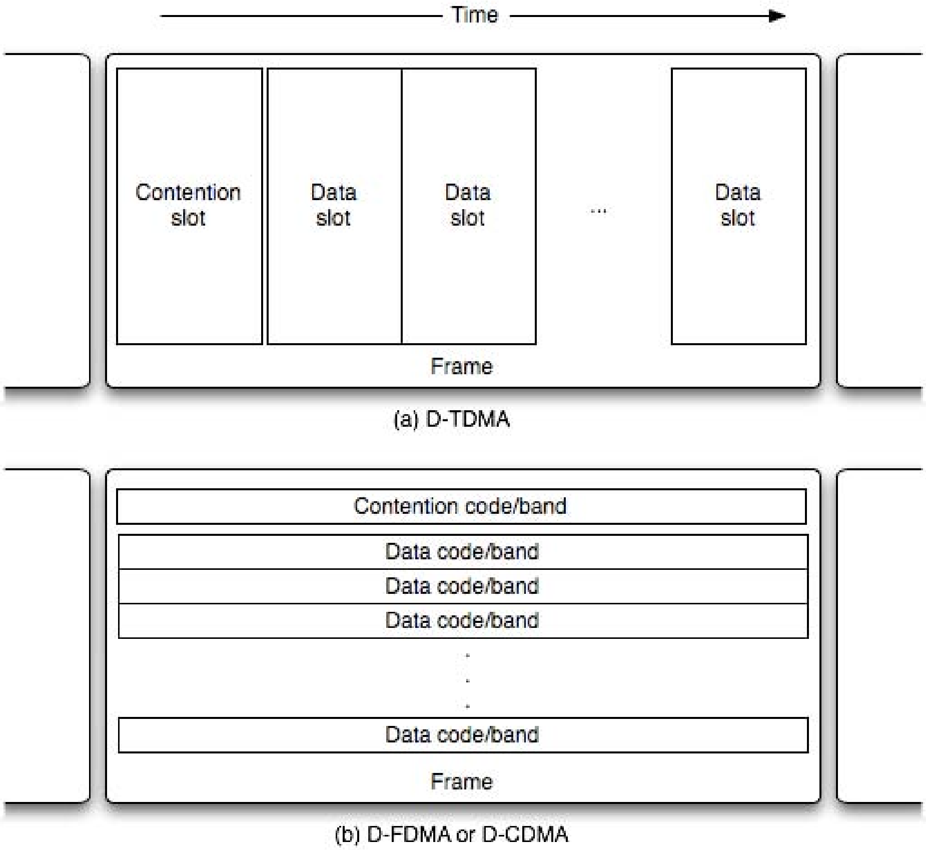


Figure 3.2: A schematic representation of D-TDMA, D-FDMA and D-CDMA.

are relinquished. Multiple successful contentions can happen within a single contention slot. Similar models for dynamic TDMA are used in many terrestrial networks such as Hiperlan/2 which was a parallel development to 802.11 standards by the European Telecommunications Standards Institute (ETSI) (Doufexi et al., 2002).

For comparison, dynamic FDMA or CDMA equivalents are shown in Figure 3.2(b). The control channel for contention can be a band or code by itself (although the requests in a frame may be for allocation of channel in subsequent frames). The same equivalence as described in Section 3.2.1 holds except for the additional shared contention-based control channel. Just like the contention frequency band in FDMA or control channel code in CDMA takes up available channel capacity, the contention period in the dynamic TDMA takes up a certain portion of total channel capacity. In terms of capacity, this control channel is equivalent irrespective of the type of channelization used.

Any of the above protocols can easily be implemented in centralized networks (they are used in cellular wireless networks). In distributed topology networks, if a node receives a RTS, it can allocate one or more of the unused receive channels to the requestor. Nodes need to be aware of what channels are being used in the neighbourhood to avoid collisions. Note that all receivers have to be full duplex multi-band or multi-code capable for this to work equally well in distributed FDMA or CDMA systems. As discussed earlier in Section 3.2.2.1, this is not typically preferred for UAN systems. The D-TDMA protocol works well without such requirements.

D-TDMA has a contention slot unlike pure TDMA that takes up bandwidth and reduces efficiency. This is the primary cost for the ad hoc capability that D-TDMA offers.

3.2.4 Dynamic TDMA protocol and MACA based protocols

It can be shown how the MACA protocol relates to the D-TDMA protocol described above. When the term MACA is used, it refers to a family of closely associated protocols which uses essentially the same principles of handshaking etc as MACA. The transmitter sends a RTS to the receiver and the receiver responds with a CTS. Upon reception of the CTS, the transmitter sends the DATA packets

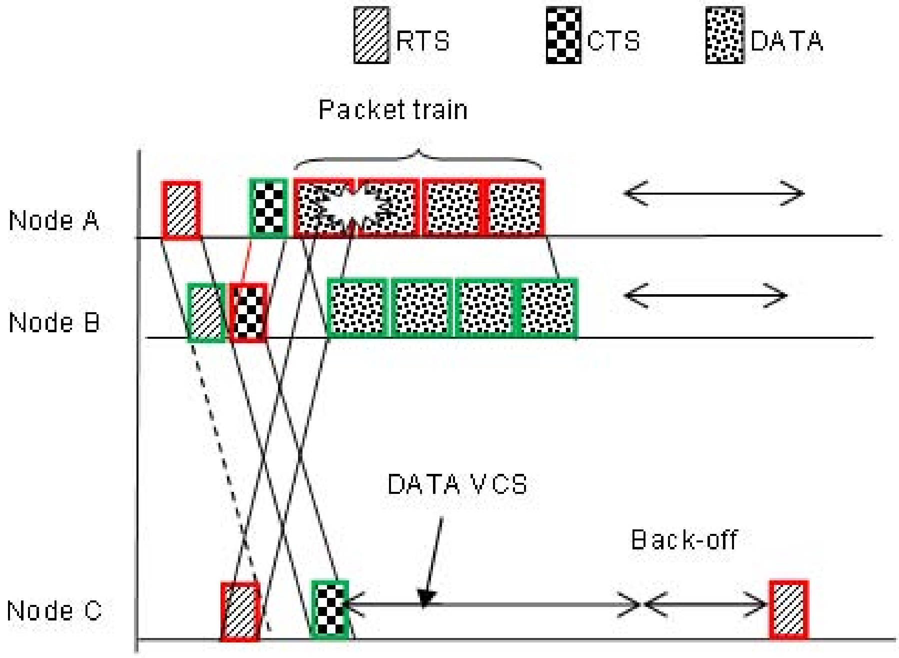


Figure 3.3: MACA protocol model with RTS/CTS/PACKET-TRAIN. Node A sends an RTS to Node B and Node B sends a CTS back to Node A. Node A then sends a DATA batch to Node B. Reception of CTS at another node C is shown which then performs a VCS to avoid interference with Node A's transmission. A potential collision from Node C is shown. How back-off starts after completion of one batch transmission is also indicated.

in a batch (referred to as DATA-TRAIN) and number of packets in a batch is variable and is specified in RTS. If CTS is not received, the transmitter does a random back-off and repeats the process. Once the receiver successfully receives some part of the data train, it sends an optional acknowledgement (ACK). Figure 3.3 illustrates this. More protocol details and performance analysis for MACA are given in Chapter 4.

Next, MACA is related to the D-TDMA protocol. In D-TDMA, let all the slots in a frame be allocated to one successful requestor at a time and let there be variable number of slots in a frame. Let each slot be viewed as a DATA packet and a frame as a DATA-TRAIN as used in the MACA-based protocol. Let the contention slot be also of variable duration as determined by the completion of one successful RTS/CTS exchange. This is the same model as MACA discussed above! Thus, MACA can be viewed a special case of a D-TDMA scheme and inherits many of the stated advantages. The most important advantage is the requirement of only a half-duplex physical layer. D-TDMA is very similar to many of the efficient dynamic protocols used in cellular networks. These protocols employ random access for the control channel only. Usually MACA is viewed as a different class of protocol, a type of random access protocol. Through the above illustration, it is evident that D-TDMA and MACA are closely related, and that the two apparent classes of protocols are not so different after all. In fact, it is interesting to note that the almost parallel development of Hiperlan/2 protocol by ETSI and 802.11 by IEEE used D-TDMA and MACA based protocols respectively (Doufexi et al., 2002). Of the two, IEEE 802.11 is the more commonly used wireless LAN protocol today.

By allowing variable data length, MACA based protocols can allow variable number of nodes to communicate within the same time frame and thus effectively meets the requirement to have dynamic channelization. There is no strict upper limit on the number of users it can support as the duration of the data frame can be varied. Performance gracefully degrades with number of users. MACA also eliminates the critical difficulty with clock synchronization required by D-TDMA as there are no repeated frames with multiple time slots for nodes to deal with. Other technical difficulties with FDMA, CDMA outlined in Section 3.2.2 are also absent. Thus, MACA provides the basis for one of the most flexible robust protocols for a dynamic channelization and allocation that works well in distributed topology and can be extended to centralized topologies as in 802.11 PCF.

The benefit from the use of packet trains needs a quick evaluation. The classic hidden node collision problem of MACA (Molins and Stojanovic, 2006) is shown in Figure 3.3. If instead of the packet train, if a single large DATA packet is used, in such a RTS collision, the entire DATA packet is more easily lost. And for re-transmission, the complete RTS, CTS, DATA, ACK exchange has to be repeated needlessly wasting channel capacity. Arbitrarily long duration single coded packets might also not be feasible due to physical layer memory and processing limitations. When packet trains are used, the RTS collisions only affect some of the packets in the train and the ACK will indicate this. By using fairly large number of packets in the train, throughput efficiency can greatly be improved as the results demonstrate in Chapter 4. The idea that packet trains improve performance of protocols such as MACA can be found in other papers (Molins and Stojanovic, 2006; Garcia-Luna-Aceves and Fullmer, 1998).

3.2.5 Re-use, topology selection

In a distributed topology, when using MACA based protocols (as used in 802.11 DCF ad hoc mode), when links (two nodes communicating with each other) are separated sufficiently in space, some links can operate simultaneously in time. In other words, time domain channel-reuse is inherent. Re-use happens automatically and no extra mechanisms are needed. Thus, for UANs where contiguous multi-hop nodes are present, it is best to use distributed protocols along the principles of 802.11 DCF, and not use pure centralized topology (note that the centralized 802.11 PCF operates within the DCF framework). Centralized topology is best only for situations operating as single collision domain or in the case of spatially separate collision domains connected via surface radio gateways. This is the case in terrestrial cellular wireless networks with the BS connected to each other through high speed wired network. If centralized topology protocols follow the approach of 802.11 PCF, since it essentially rides on DCF, the same automatic re-use mechanisms are present and can also be used.

3.2.6 Propagation delay and its impact

How does propagation delay affect the behaviour of a MACA-based protocol? The basic impact of propagation delay is the loss of channel utilization and efficiency in the transmission delays between RTS, CTS, DATA and ACK packets. Since many commercial acoustic modems with a useful data rate has only a few kilometres of range, many practical UANs have a few kilometres as a single collision domain at maximum. Thus, RTS/CTS based protocols are not heavily impaired in most cases. As shown in Chapter 4, the use of appropriate batch sizes in data packet train can help counter this loss of efficiency.

Such loss of efficiency is also a fundamental limitation of dynamic ad hoc networks that satisfies the requirement of dynamic channelization and allocation as outlined at the beginning of the chapter. Only static networks can avoid this loss of efficiency. Thus, it is not a problem with MACA per se, but a limitation arising directly from network requirements and underwater characteristics. In case of very high propagation delay networks that span tens of kilometres as a single collision domain, ALOHA-based variants with no handshaking could be explored.

MACA in its original form does not have physical carrier sense (PCS) . However, if PCS were used as in 802.11, its effectiveness would be undermined by high underwater propagation delay. PCS works on the premise that when a node transmits, all the other nodes hear it instantaneously. In UANs this is not true. The authors' own simulation studies have shown that carrier sensing makes only negligible difference in performance. There are published variants of MACA that uses PCS (e.g. Slotted-FAMA (Molins and Stojanovic, 2006)). However, the efficacy of PCS in UANs has not been conclusively shown. Some of these are still open research problems.

3.3 Conclusion

This chapter compared TDMA, FDMA and CDMA at a fundamental level for use in a distributed topology UAN and showed that TDMA based schemes are perhaps best suited for the purpose. Dynamic TDMA is a natural extension that addresses the requirement of dynamic channel allocation (ad hoc networks). MACA is seen as a further extension of dynamic TDMA and inherits many of its advantages. It offers even greater robustness as it does not require precise time synchronization, which is often difficult to achieve in many underwater networks. An 802.11 style protocol based on MACA with suitable modifications is perhaps the best choice for general purpose distributed topology UANs, and has good experimental validation unlike many other proposals for underwater networks. There were a number of ongoing research projects around the world during the same period as this work, exploring suitable variations to the MACA protocol for UANs to improve its performance. In going from pure TDMA, to D-TDMA and to MACA, there are trade-offs such as decreased efficiency and increased delay, that come together with benefits such as ad hoc capability, scalability, not needing time synchronization, and not having to do re-use planning. The system designer has to choose the trade-offs based on the exact requirements of the network being built.

Chapter 4

A high performance MAC protocol for underwater acoustic networks: MACA-EA

After establishing a sound basis for the utility of MACA-based protocols for UAN MAC in Chapter 3, our attention can now be turned to a comprehensive evaluation of a novel MACA-based protocol called MACA-EA for UANs, using simulations, mathematical modelling and field experiment validation. A part of this work has been published (Shahabudeen and Motani, 2009) and a new draft has been submitted (Shahabudeen et al., 2011).

4.1 Review

Chitre et al. (2008) presented a review of MAC protocols for UANs. Small static UANs are best served by pure reservation protocols such as TDMA. If nodes are capable of communicating over multiple channels, a static cellular model where channels are spatially divided and allocated to cells according to a channel re-use pattern, is also a good option (Stojanovic, 2007). However, pure reservation based protocols are unable to meet the ad hoc demands of many underwater applications. For such applications, dynamic and ad hoc MAC protocols are necessary. One of the simplest dynamic and ad hoc protocol is Multiple Access with Collision Avoidance (MACA). The choice of MACA for UANs has been explored in detail in Chapter 3. Variants of MACA are used in terrestrial networks such as 802.11 (Bianchi, 2000). Protocols such as FAMA (Fullmer and Garcia-Luna-Aceves, 1995), DACAP (Peleato and Stojanovic, 2006) also are closely related to MACA. Petrioli et al. (2008) presented a simulation-driven comparative study of some of these protocols. Peng and Cui (2006) present another evaluation of MACA-like protocols for UANs.

Several papers present analysis of MACA or its variants. Garcia-Luna-Aceves and Fullmer (1998) presented an analysis of FAMA, a close variant of MACA. The authors assume that packet collisions are the only source of error, and ignore any loss due to noise (failure to detect or decode a packet due to noise). A three-way handshake with no acknowledgement is assumed. The paper lacks a saturated load analysis, service time distribution and a complete queuing analysis. Bianchi (2000) analyzed the IEEE 802.11 DCF (which is also a variant of MACA). This paper also ignores packet detection and decoding losses. The analysis assumes a freezing back-off algorithm, rather than an optimal back-off window that is used in this chapter. The service time equation derived is not expressed in closed form. Molins and Stojanovic (2006) present a UAN oriented analysis of FAMA. The protocol used in the simulations seems to use only a single DATA packet, the length which is 30 times that of the RTS/CTS packet. The batch mode protocol proposed in that paper is inefficient as it acknowledges each data packet independently. The reliable transmission throughput performance is low as compared to the results presented in this chapter. The expression for throughput does not capture the impact of batch size or back-off window size. The paper also lacks a saturated load throughput analysis similar to (Bianchi, 2000) and a complete queueing analysis. A more detailed comparison of the S-FAMA protocol with the work in this thesis is presented in Section 4.3.5.

The above papers do not analyze the queuing behaviour of MACA based protocols. Park et al. (2006) present a queuing analysis of the 802.11 MAC. However, the analysis does not allow for detection and decoding losses, reliable delivery of packets across a link or transmission of a batch of data packets to minimize the effect of long propagation delays. The analysis is specific to 802.11 and its freezing-back-off model, and therefore not directly applicable to the optimal back-off window based protocol that is proposed in this chapter.

As seen, most of the analysis published to date focus on terrestrial networks and therefore ignore packet loss due to noise, and the long propagation delays encountered in underwater networks. The higher packet loss calls for link level acknowledgments. The long propagation delays call for sending a batch of packets for every RTS/CTS exchange (Request to Send/Clear to Send). The papers that address these issues only present limited analysis and focus on simulation results. A detailed analytical model and appropriate queuing analysis is missing from these papers. In this chapter, a novel MACA-based protocol is presented for use in UANs along with a detailed mathematical analysis of its performance. Performance bounds on the protocol are derived through the use of realistic models and parameters, and validate them through a combination of analysis, simulation and experiment.

The analysis presented in this chapter was influenced by some key ideas that have appeared in other papers. Specifically, the saturated load analysis was motivated by Bianchi (2000). The Markov chain analysis to first find the expected service time for successful transmission was influenced by Chatzimisios et al. (2003). The service time distribution analysis by comparison with standard distributions was motivated by the analysis of IEEE 802.11 presented by Foh et al. (2007). The idea of dummy states used in the service time distribution analysis was adapted from the analysis by Gupta and Kumar (2004).

Through a novel analysis, this chapter presents new insights on the well-studied MACA protocol family, with primary focus towards UANs. Section 4.2 outlines the system model, the MACA-EA protocol and the performance measures. Service model related measures such as expected service time and throughput are derived in Section 4.3 and the service time probability distribution is derived in Section 4.4. The analysis of queuing and total delay behaviour is presented in Section 4.5. A discussion on pre-emptive contention is provided in Section 4.7.1. This is followed by further analysis of the optimum back-off window in Section 4.7.2. In Section 4.6, a recently developed enhancement named MACA-SEA is presented.

The MACA-EA analysis presented in this chapter may be applied to terrestrial wireless networks. However, our primary objective is to analyze the underwater MAC problem and therefore adaptation of the analysis to more general networks is not presented in this chapter.

4.2 System Model

In this section the system model used in the chapter is discussed, including the arrival and departure models, key model parameters, the protocol model being analyzed and the performance measures.

4.2.1 Input-output models

In the arrival (input) model, each data packet from network layer, or other layers above the data link layer (DLL), fits within a single DLL/physical layer packet. Both saturated load and Poisson arrivals are considered. If the higher layer data size requires the usage of multiple DLL/Physical layer packets, fragmentation and reassembly may be needed. However, fragmentation/reassembly is not modelled in this chapter.

For the service model, a packet train model is adopted where the DATA is sent in batches of size B with re-transmissions at the DLL (infinite retry model). For the retry mechanism, some novel enhancements are proposed, as detailed later in the chapter.

4.2.2 Packet detection, error and collision model

The packet model has fixed length detection preamble at the start. Detection probability P_d is dependent on the nature of the preamble. Packet decoding probability P is determined by the bit error rate (BER) of the physical layer, the number of bits in the packet and the coding scheme. The probability that a packet is detected and decoded correctly k, is:

$$k = P_d P \tag{4.1}$$

Control and data packets may use different modulation, coding and packet length, as robustness is of key importance to control packets while data rate is of importance in data packets. To model this, the decoding probability P_D of data packets is allowed to differ from that of control packets. Therefore the overall data packet success probability k_D is:

$$k_D = P_d P_D \tag{4.2}$$

Let the time duration (in seconds) of a control channel packet be L while that of a data packet be L_D . The maximum propagation delay is D. The number of nodes in the collision domain is N; it is assumed that there are no hidden nodes (Garcia-Luna-Aceves and Fullmer, 1998). In a single collision domain scenario, the number of nodes N is the total number of nodes. In a multiple collision domain (multi-hop) scenario, N is best viewed as the number of neighboring nodes that each node effectively contends with. Section 4.7.7 briefly outlines a rudimentary analysis for multi-hop networks.

4.2.3 MACA-based protocol model

The protocol is based on MACA using RTS/CTS (request to send/clear to send) exchange (Karn, 1990). The basic model used is RTS-CTS-DATATRAIN-ACK. The transmitter sends RTS and the receiver sends back CTS. The transmitter then sends a batch of DATA packets (DATA-TRAIN). The receiver then sends a single acknowledgement (ACK) which indicates failed packets in the batch. Similar protocols with packet trains that employ ACKs after every packet (RTS-CTS-DATA-ACK-DATA-ACK…) are not efficient for UANs due to the two-way propagation delay overhead and thus only a single ACK is used at the end. The protocol is elaborated further below.

In the RTS contention algorithm, a node starts with a uniform probability distributed back-off in a contention window W. When the back-off timer expires, a RTS is sent. Timer t_A starts when RTS is sent. If the timer expires before reception of CTS, RTS back-off procedure starts again. Once CTS is received, DATA-TRAIN is sent followed by wait for ACK. If ACK is not received, the RTS cycle repeats. Reception of RTS/CTS packets and a possible DATA frame while waiting to send RTS triggers Virtual Carrier Sense (VCS). Successful DATA transmission for any one node restarts RTS contention cycle for all. Note that 802.11 uses freezing back-off which is described in (Bianchi, 2000) whereas a constant window is used here. This protocol also does not use Physical Carrier Sense (PCS) whereas it is used in 802.11 (See 4.7.3 for comments on PCS usage). All nodes use the same contention window W at any given time.

The timers used to wait for CTS and ACK (t_A) are related to D and control packet time duration L to give enough time for the round trip delay as

$$t_A = 2D + 2L \tag{4.3}$$

To make reliable transfer more efficient, two variations are proposed with regards to acknowledgments and retransmissions to handle failed data packets. After a batch of DATA is received, an ACK is sent by the receiver. In typically used retry models, if an ACK fails to reach the transmitter, the RTS/CTS based contention cycle and batch DATA transmission processes repeat. Two enhancements to this retry process are introduced. Firstly, instead of sending one ACK packet, i ACK packets are sent, a feature termed Multi-ACK. The second enhancement is as follows. When the sender of the DATA train does not receive the ACK, RTS is repeated with the same UID (unique identification number, incremented only for a RTS for a new packet train). The receiver sends back an ACK instead of CTS for the repeated RTS. Together with the Multi-ACK feature, this is called the Early-Multi-ACK model. The retry mechanism uses constant back-off with infinite retries (other options include exponential increase exponential decrease, maximum retries capped, etc.). This protocol shall be referred henceforth as MACA-EA (for MACA-Early-ACK, omitting the term “Multi” for simplicity).

4.2.4 Performance measures

Queuing theory is commonly used in the modelling and analysis of wireless networks. Typically, the arrival process is modelled as Poisson distributed and the service time as exponentially distributed. A Markov chain analysis is then used to study the behaviour of the system. Important common metrics derived are service time distribution and its expected value, throughput efficiency, expected steady state queue length and expected total waiting time.

The mean packet service time s_p is defined as the expected delay from the time a packet is intended for transmission (RTS contention starts) until it is successfully delivered, i.e., until the ACK (with retries) shows successful reception of the specific packet. Mean batch service time s_b is defined as the average delay from the time a batch is intended for transmission (RTS contention starts) until it is successfully transmitted, i.e., until the first ACK is received for the batch. The above different definitions of s_p and s_b are important for the queuing analysis in Section 4.5. In Section 4.3, expected service times (s_p and s_b) are related to the network parameters as follows

$$s_b = f(N, D, L, L_D, B, k, k_D, W, t_A) \quad s_p = g(N, D, L, L_D, B, k, k_D, W, t_A) \tag{4.4}$$

Another important performance metric for reliable transfer is throughput, which is analyzed in Section 4.3.3. In some papers on similar protocols in radio networks, this is termed as "saturation throughput" – the throughput of the network when the queue is saturated or always has data to transmit (Bianchi, 2000). Such a measure is valid for file transfer applications. This is also a measure of efficiency or channel utilization. Normalized throughput T is defined as the number of packets successfully transferred per unit time normalized by the system capacity $(1/L_D)$. B packets are sent as a batch in time s_b by definition, and of these, only k_D succeed due to decoding and detection losses. Thus, the normalized throughput T per node is

$$T = \frac{k_D B / s_b}{(1/L_D)} \tag{4.5}$$

In Section 4.4, simulations and numerical analysis examine the service time probability distribution. Once the service behaviour is characterized with mean service times s_p and s_b and the service time CDF, other queuing metrics such as waiting time and queue length under non-saturated conditions (Poisson arrivals etc) can be derived using queuing analysis (Section 4.5). The total waiting time W_T includes the waiting time in the queue W_Q and the mean service time s_p per input packet, i.e., $W_T = W_Q + s_p$.

4.2.5 A brief note on simulations

The simulator used for this study is described in detail in Chapter 8. In the simulation model, all nodes have data to send and each node sends data to one other recipient node. The simulator accurately models collision, decoding and detection errors, propagation delays, etc. It's based on Omnet++ (Omnet++, URL), an established discrete event simulation system. The nodes are randomly spread in an area whose dimensions are chosen to match the required maximum propagation delay D. At each sender, the number of acknowledged packets received during the duration of the test is recorded and added at the end of the simulation run to get the overall network total. The normalized reliable throughput is this total normalized using factor t_{SIM}/L_D, which is the theoretical maximum possible number of data packets sendable during the total simulation time t_{SIM}. The complete algorithm for the protocol discussed here has been implemented for both simulations and the sea-trials. The software system allows the same algorithm code to be used for simulations as well as sea-trials, i.e., there is no code porting required for sea-trials. This ensures that the simulation results and sea-trial results will have no artifacts due to possible porting differences and errors. Sea-trial related details are discussed in Section 4.3.6. For sea-trials, key parameters are estimated and then plugged into the corresponding simulations and analysis models. For example, the probability of detection and decoding is estimated by sending a series of packets and recording the number of detected and successfully decoded packets.

4.3 Analysis of Service Time and Throughput

In this section, expressions for the expected service time and the saturated throughput are derived. These closed form expressions closely match simulations and can be used for estimating protocol behaviour and considerably reduce the need for simulations. The expected service time metrics are then used in Section 4.5 for queuing analysis.

4.3.1 Markov chain model for the protocol excluding retries

In the analysis of the Slotted-FAMA protocol, Molins and Stojanovic (2006) show that the performance of RTS/CTS based protocols improves with slotting. In case of the protocol at low-to-medium range, from simulations it was observed that there may be a marginal difference with time slotting. For the analysis here, the slotted model is used and in that sense the protocol used here is closely related to Slotted-FAMA. In line with the definition used by Molins and Stojanovic (2006), during RTS contention phase, the slot duration l is defined as

$$l = L + D \tag{4.6}$$

This allows for collisions to be contained within the slot boundaries. For $D \leq L$, packets transmitted in the same slot will at least partially collide. For example, for the NUS/ARL modem (Chapter 8), highly robust control packets have duration $L = 0.6s$. Thus, the model is very effective for D of the order of 0.6s (up to 900meters range). For $D \gg L$, packets in the same slot might not collide and the analysis is expected to give a conservative bound. For $D \gg L$, time slotting (as defined above) as a protocol feature might also prove to be ineffective, and the un-slotted version could potentially outperform the slotted version.

The protocol model is as described earlier in Section 4.2.3. A node starts with a uniformly selected back-off time slot in the integer range $[1, W]$. The actual contention window time period is Wl. For simplicity of analysis, it is assumed that no collisions happen during the CTS period, assuming VCS starts due to RTS reception (results showed that this simplification did not have significant impact on the analytical predictions). So in our analysis model, CTS loss will only be due to decoding and packet detection probability. If the transmitter does not get CTS, it restarts the contention window for RTS. Any other node which had received the RTS does a VCS for CTS. It resets and restarts contention if CTS does not arrive. Thus, until one node gets a CTS and DATA transmission starts, this process will continue. In order to handle the case of some nodes missing the winning CTS and interfering with the DATA phase, all nodes monitor for DATA packets and DATA packets contain information of how many packets remain in the batch. This helps

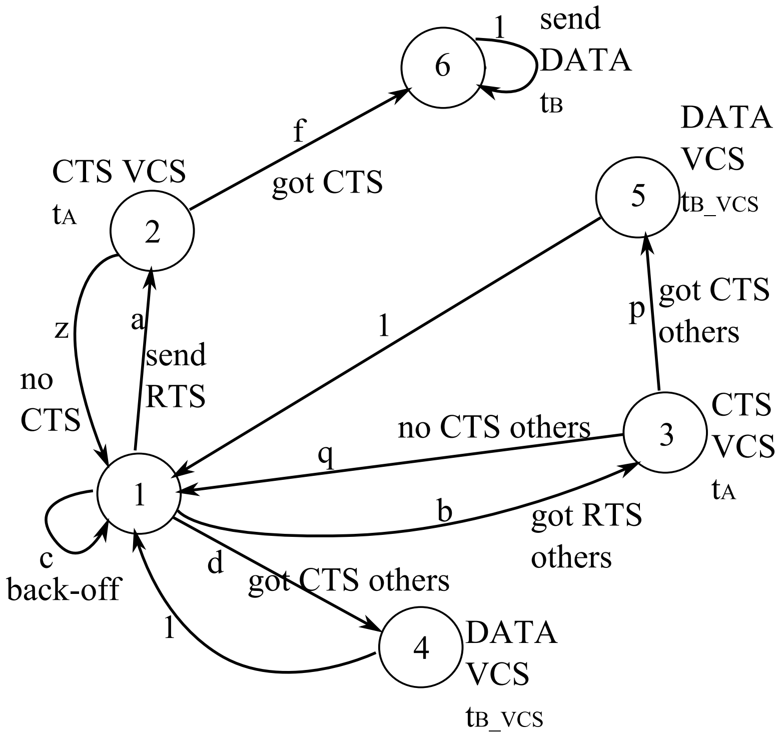


Figure 4.1: Main Markov chain for computing expected service time

nodes that missed RTS/CTS to regain VCS with a probability close to 1 after a few DATA packets are sent. This can be seen by noting that the probability of getting at least one packet after n packets are sent is $1-(1-k_D)^n$ which rapidly tends to 1 as n increases. Thus, the contention cycle synchronization is maintained. This is similar to the NAV (Network Allocation Vector) concept used in 802.11.

RTS packet transmissions are scheduled at the start of a time slot only; other response control packets are transmitted immediately to allow immediate VCS. DATA packets do not use slotting and there are no gaps between the DATA packets in a batch.

The protocol is represented using the main model in Figure 4.1 and a supplementary model in Figure 4.2. The main model accounts for the RTS/CTS process until a batch of DATA is transmitted. The supplementary model accounts for the

ACK process. The absorbing state 6 in Figure 4.1 and state 7 in Figure 4.2 are for mathematical convenience and the protocol in actual operation does go back to state 1 and repeats the whole process after one cycle is complete. Circles with enclosed numbers are states. Transition probabilities are shown along the arrows. The duration spend in state 1 is 1, and for others states is t_B, $t_{\mathrm{B\_VCS}}$(described later in (4.9), (4.10)) or t_A as indicated. In the analysis, state transitions will be represented as a pair such as (g, h) for a transition from state g to h. State transition probability will be represented as $P(g, h)$.

The start of RTS contention cycle is at state 1. The probability of a node sending a RTS at the start of a new slot is modelled as $P(1, 2) = a = 2/(W + 1)$. This is because the expected value of the uniformly distributed contention window is $W' = (W + 1)/2$, and that is used as the expected value of a geometric process for transition (1,2) to satisfy Markov Chain requirements. Once a RTS is sent, node is in state 2, waiting for t_A time slots for CTS to arrive. If CTS arrives, it goes to state 6 and transmits a batch of duration t_B (this is described later in (4.9)).

For notational convenience, define $\omega = 1/W'$. The probability that the RTS transmitted in a given slot has no collision from any other node is $(1-\omega)^{N-1}$, i.e., no other node transmits a RTS in that slot. CTS will be successfully received if apart from having no collisions, RTS is received at the receiver (probability k) and the CTS in turn is received at the transmitter (probability k) with a combined probability of k^2. This is shown in Figure 4.1 as $P(2, 6) = f = k^2 (1 - \omega)^{N-1}$. If CTS is not successfully received, transition (2,1) happens as shown with probability $z = 1 - f$.

If a RTS is not sent (probability $1 - \omega$), the current node counts down the RTS timer by one slot. During this back-off period, the probability that one of the $N-1$ neighbors has a successful RTS transmission is $y = (N-1)\omega(1-\omega)^{N-2}$ using same arguments as in last paragraph. And k being the RTS detection probability, the current node could receive a RTS from another node with probability ky . Thus, the transition (1,3) with $P(1,3) = b = (1 - \omega)\, ky$ occurs as shown.

In state 3, it awaits CTS for time t_A. Thereafter if CTS is successful (probability $p = k^2$, since both RTS needs to be independently received by recipient and CTS received by current node), it goes to state 5 for batch VCS (for time $t_{\mathrm{B\_VCS}}$, described later in (4.10)), following which it goes back to state 1 with probability 1 immediately. CTS failure in state 3 with probability $q = 1 - k^2$ takes the system back to state 1.

If during back-off as described in last paragraph (probability $1 - \omega$), CTS is received directly, transition (1,4) occurs. As before, the probability that at least one of the $N - 1$ neighbors has a successful RTS transmission is $y = (N - 1)\omega(1-\omega)^{N-2}$. But this RTS was missed (probability $1-k$) but CTS was received (RTS received at other node and CTS received by node under consideration with probability k^2). Thus, $P(1,4) = d = (1 - \omega)\,(1 - k)k^2 y$ as shown. In state 4, VCS for batch reception (for time $t_{\mathrm{B\_VCS}}$, described later in (4.10)) occurs and goes back to state 1 thereafter with probability 1. Note that for transitions (3,5) and (1,4), its possible that the intended recipient of the CTS may not receive it, but the neighbors who overheard any CTS honours VCS for batch transmission.

This is a design choice for this protocol. If system is backing off and either RTS or CTS from others is not received as stated above, it goes back to state 1 as shown with $P(1,1) = c = 1 - a - b - d$.

Note that for states except state 1, where more than 1 unit of time is spent, it could be modelled using an expanded Markov chain as used later on in Section 4.4. But the above simplified Markov chain gives correct results if time in each state is accounted for as done in (4.11) for example.

A Markov matrix $\mathbf{M}$ (Gross and Harris, 1998) represents this as follows using $P(a,b)$ as shown in Figure 4.1. $\mathbf{Q}$ is the transient state matrix.

$$\mathbf{M} = \begin{bmatrix} c & a & b & d & 0 & 0 \\ z & 0 & 0 & 0 & 0 & f \\ q & 0 & 0 & 0 & p & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \mathbf{Q} = \begin{bmatrix} c & a & b & d & 0 \\ z & 0 & 0 & 0 & 0 \\ q & 0 & 0 & 0 & p \\ 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{4.7}$$

The fundamental matrix $\mathbf{F}$ (Gross and Harris, 1998) is the $\mathbf{F} = (\mathbf{I} - \mathbf{Q})^{-1}$. Let $E(m,n)$ be the expected number of times the system is in state n after starting from state m. $E(1,n)$ is the expected number times the state n will be visited if the chain starts in state 1. Using standard Markov Chain theory (Gross and Harris, 1998):

$$\begin{aligned} E(1,n) = F_{1,n}; \quad E(1,1) &= \frac{1}{k^2} \frac{W'}{\left(\frac{W'-1}{W'}\right)^{N-1}} \\ E(1,2) = \frac{1}{k^2} \frac{1}{\left(\frac{W'-1}{W'}\right)^{N-1}}; \quad E(1,3) &= \frac{N-1}{k} \\ E(1,4) = (1-k)(N-1); \quad E(1,5) &= k(N-1) \end{aligned} \tag{4.8}$$

4.3.2 Enhanced retry mechanism

Handling failed data packets through ACK and re-transmission is analyzed now as briefly discussed in Section 4.2.3 earlier. After the process from state 1 to 6 in Figure 4.1, ACK is sent by the receiver. In typically used retry models, if ACK fails to reach, the stages from state 1 to 6 repeat until ACK is successfully received. Two enhancements to this retry process are introduced. Firstly, instead of sending one ACK packet, i ACK packets are sent. The probability of correctly receiving at least one of them is $1-(1-k)^i$. The additional time required for multiple ACKS is $(i-1)L$ and round trip time t_A is allowed for ACKs to be delivered. Thus, batch transmission time t_B is:

$$t_B = BL_D + t_A + (i-1)L \tag{4.9}$$

In the second enhancement as mentioned in Section 4.2.3, when the sender of DATA train does not receive the ACK, RTS is repeated with the same UID (unique identification number, incremented only for a RTS for a new packet train). Receiver sends back ACK instead of CTS for such repeated RTS. This is shown in Figure 4.2. The VCS delay $t_{\text{B\_VCS}}$ spent in states 4 and 5 in Figure 4.1 is computed as follows. When using the Early-ACK feature, instead of receiving CTS, some of it could be Early-ACKs.

Though there is no real batch transmission, let's for clarity use t_E as the time associated for a pseudo batch transmission in state 6 of such an Early-ACK cycle ($t_E = 0$ of course). If n batch transmissions of t_B are considered over a long period of time among other nodes, there are thus $n(1-k)^i$ occurrences of t_E. The expected VCS time in states 4 and 5 in Figure 4.1 is thus $t_{\text{B\_VCS}} =$

$(nt_B + n(1-k)^i t_E)/(n + n(1-k)^i)$. Since $t_E = 0$,

$$t_{\text{B\_VCS}} = \frac{t_B}{(1 + (1-k)^i)} \tag{4.10}$$

Let the time till successful reception of CTS from state 1 to state 6 of Figure 4.1 be s_{CTS} (excluding the batch transmission time in state 6). This gives

$$s_{\text{CTS}} = (l)E(1,1) + t_A E(1,2) + t_A E(1,3) + t_{\text{B\_VCS}} E(1,4) + t_{\text{B\_VCS}} E(1,5) \tag{4.11}$$

Using (4.8), s_{CTS} can be simplified as $s_{\text{CTS}} = \gamma + (N-1)t_{\text{B\_VCS}}$ where γ is

$$\gamma = \frac{l}{k^2 W'} \left(\frac{W'}{W'-1} \right)^N (W'^2 + W' - 2) + \frac{2(N-1)l}{k} \tag{4.12}$$

s_{CTS} is also the time taken to get an Early-ACK if the initial ACK fails after a batch transmission from a given node, since it uses the same process from state 1 to state 6 (excluding the batch transmission delay in state 6) as shown in Figure 4.2. Based on Figure 4.2, the total batch service time s_b from state 1 to state 7 (from RTS until Early-Multi-ACK) can now be computed using expected time from state 1 to 6 (including batch transmission) and additional time s_{CTS} for Early-Multi-ACK as follows

$$s_b = s_{\text{CTS}} + t_B + (1-k)^i s_{\text{CTS}} = \left(1 + (1-k)^i\right)\gamma + N t_B \tag{4.13}$$

Equation (4.13) shows the average time for batch transmission from the perspective of a single node in a group of N nodes. The factor Nt_B has an intuitive appeal, since on average each node should get a turn to send DATA after $N-1$ batch transmissions by other nodes. A particular batch comprises of packets to be resend (those in the previous batch that did not get across), and new packets. In the Early-ACK model, such a specific batch (with a specific UID) is transmitted only

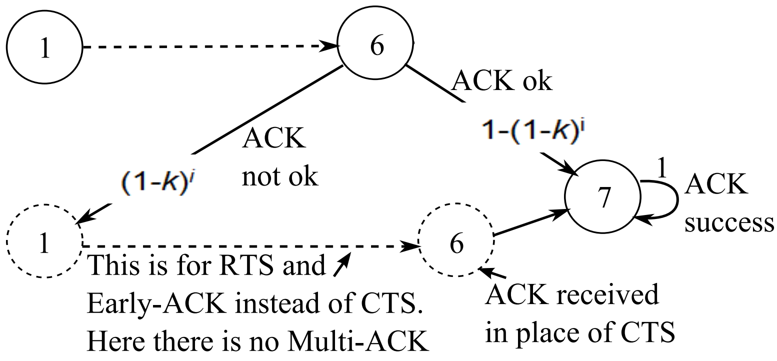


Figure 4.2: Markov chain for Early-Multi-ACK

once and the service time s_b refers to the time taken to send that. If the ACK is lost, the specific batch never gets resent, as an Early-ACK will be sent in response to the repeat RTS. If the ACK is not lost, the next batch will be formed with a new UID and a RTS will be send for that. This new batch may even comprise entirely of old DATA packets if the ACK indicated that no packets made it previously, and even then it will still be considered as a new batch, with a new UID and another service time s_b applies to it.

An important observation is that in this protocol the contention process and the variable time associated with it (not including batch data transmission) depends only on control packet duration and its probability of success. It does not depend on individual DATA packet duration or its success probability. Total data packet transmission duration is captured though the term t_B in (4.13) and it depends on the product BL_D.

The following are some key comments regarding ACKs and the ARQ mechanism used in the protocol variants in this chapter:

- ACK will be sent if any one packet in the train gets through. For simplicity, it can be assumed that the receiver will get at least one packet and hence it will send back an ACK with probability 1 (probability of getting at least one packet is $1-(1-k_D)^B$. As an example, for $P_d = P_D = 0.9$ and $B = 5$, P (get at least one packet) $= 0.9998$).

- During the Early-ACK phase, there is no Multi-ACK, since in this phase RTS/CTS exchanges are potentially in progress among other nodes and it will be a protocol violation. Only one ACK packet can be sent, which is of the same size and coding type as a CTS packet, for correct reception during this contention phase.
- The above reliability mechanisms are related to general ARQ strategies. The analysis here can be considered an ideal form of retry based reliability. ACK is assumed to convey complete information on lost packets and re-transmissions happen only on lost packets. Other standard and practical forms of ARQ such as the Selective Repeat ARQ in TCP/IP etc may have limitations on amount of information that can be contained in ACK. They would be less efficient since re-transmissions may potentially happen for packets previously received. Thus, the results here can be viewed as an upper bound on reliable performance using batch mode service as described in this chapter.
- Kalscheuer (2004) presented an ARQ mechanism for an UAN MACA-based protocol. It uses a different model from the Early-ACK model in MACA-EA and after each batch of DATA, the receiver initiates an ARQ request for corrupt packets and the process continues until all the packets in the current batch are received successfully. In MACA-EA, after one batch transmission,

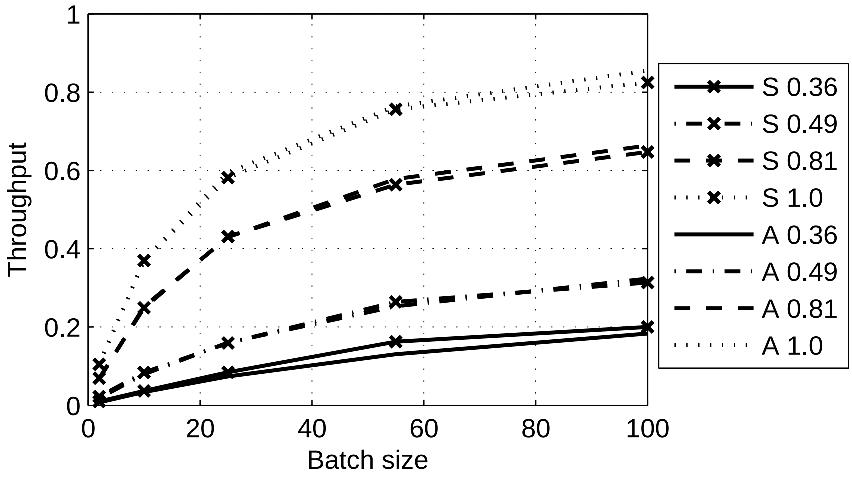


Figure 4.3: Throughput vs. batch size for $k = k_D = 0.36$, 0.49, 0.81 and 1.0, simulations (S), analysis (A). Parameters: $L = L_D = 0.5s, N = 3, D = 0.5s, W = 4, i = 3$.

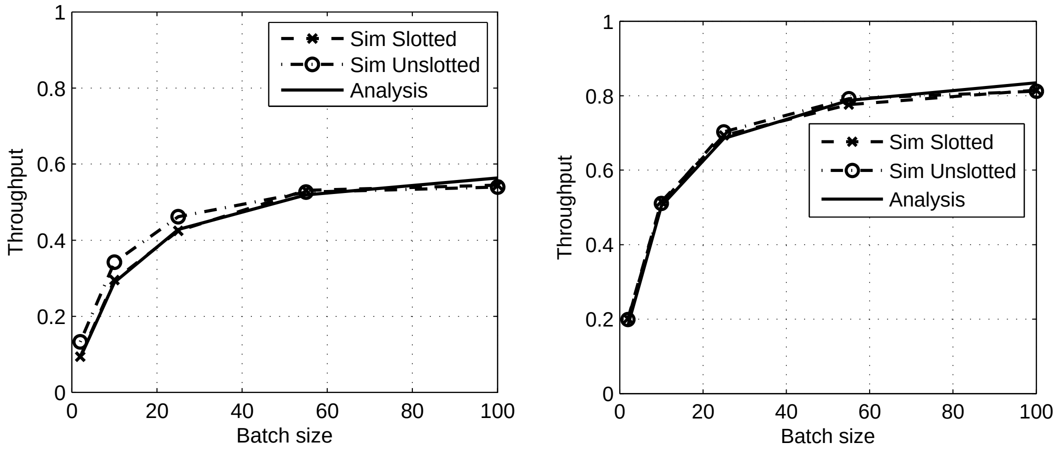
all nodes proceed to contend and a subsequent batch contains both previously corrupt packets and new DATA as described earlier. In the ARQ protocol in (Kalscheuer, 2004), it is also not clear how failed SRQ packets (they occur in the same context as ACK packets in MACA-EA) are handled.

- The optimum number of ACKs i can be found as discussed in Section 4.7.5.

4.3.3 Expected throughput for reliable transfer

Using (4.5) and (4.13), the throughput is shown in Figure 4.3 for one scenario from both analysis and simulations (this is the network throughput, i.e., NT). The parameters used are as follows unless otherwise mentioned: $L = L_D = 0.5, N = 3, D = 0.5, W = 4, i = 3$ (i.e., three Early-Multi-ACK). In Figure 4.4, some cases where $L \neq L_D$ and $k \neq k_D$ is explored. The analytical results match the simulation results well.

In Figure 4.4, simulation results for the unslotted model have also been



(a) $L_D = 1.0s, k_D = 0.63$ (b) $L_D = 1.5s, k_D = 0.9$

Figure 4.4: Throughput vs. batch size. Parameters: $[L = 0.5s, k = 0.81, N = 3, D = 0.5s, W = 9, i = 3]$

shown to ascertain the effectiveness of slotting as discussed in Section 4.3.1. As seen, there is little difference from the slotted model for the various simulations undertaken in this study. Based on this finding, the sea-trials discussed in Section 4.3.6 do not use slotting.

The throughput has a saturation type behaviour at higher batch sizes. Using (4.5) and (4.13), saturation throughput behaviour is $\lim_{B\to\infty} T = \frac{k_D}{N}$, which is the theoretical upper bound for one-way transmissions (when the combined detection and decoding success probability is k_D). To compare, the performance of the standard ACK model (without Early-Multi-ACK) is $\bar{s_b} = \frac{1}{k}(\gamma + N\bar{t_B})$ (see Appendix A.1 for details). The corresponding asymptotic throughput is $\lim_{B\to\infty} T = \frac{kk_D}{N}$. This performance is lower by a fraction k of that of the enhanced ARQ model proposed in this chapter.

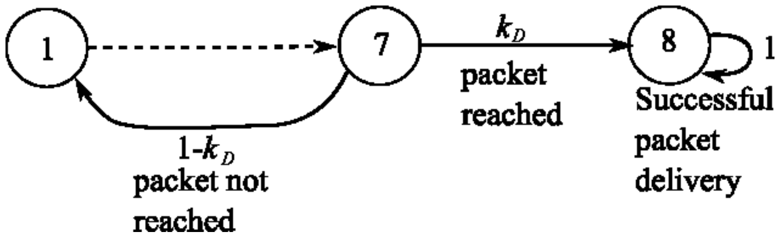


Figure 4.5: Markov chain to compute s_p

4.3.4 Expected packet service time s_p

The average reliable delivery service time s_p is measured on a single packet within a batch. ACK indicating error packet determines if any re-transmissions happen in the next round as shown in Figure 4.5. It gives s_p as

$$s_p = \frac{1}{k_D} s_b \tag{4.14}$$

4.3.5 Comparison with previous analyses

For $k = k_D = 1$ in Figure 4.3, the collision-limited performance of the protocol variants can be seen. This is the scenario used in most papers on UANs (e.g., (Guo et al., 2009)), i.e., only collisions are assumed to cause losses. Based on Section 4.3.3, T will converge asymptotically to $1/N$ using the enhanced retry mechanism if $k = k_D = 1$. This throughput performance of the batch-mode MACA based protocol used in this chapter is as good as or better than that of most of the previously proposed ad hoc protocols for UANs such as PCAP (Guo et al., 2009). Most previous analysis and simulation that do not consider detection (P_d) and decoding (P) based packet losses, are ignoring very important loss factors in UANs. By taking into account both P_d and P in our analysis, the results here provide a stronger basis for MACA-based protocol performance analysis for underwater applications. It should be highlighted that this protocol performance analysis includes retry based reliability. Actual file transfers are done in sea trials used to measure protocol performance, as mentioned in Section 4.3.6.

Some key differences with S-FAMA (Molins and Stojanovic, 2006) are noted here, as it is a very similar protocol. Features such as Early-ACK, Multi-ACK are absent in S-FAMA. S-FAMA sends all packets including DATA only at the start of a slot, whereas MACA-EA sends only RTS at the beginning of a slot. Longer contiguous DATA transmission in S-FAMA is typically achieved through the use of single long DATA packet (simulations use DATA packets that are thirty times longer than the control packets), where as packet trains with short DATA packets are used in MACA-EA. DATA packet trains are also considered in S-FAMA, but the acknowledgement mechanism uses one ACK per DATA packet, where as MACA-EA uses a single ACK at the end of a batch. The best average throughput per node is shown to be about 0.022 at a range of about 1.9 km, where the average number of neighbors is shown to be 4. It is low as compared to the best per node throughput performance of MACA-EA (shown to approach $1/N$ in the absence of detection errors, so predicted to be about 0.25 for 4 neighbors). However, the results they have presented are not for saturated traffic and therefore may not be directly compared with the results here.

4.3.6 Sea trial results

In this section the key findings from acoustic modem sea trial results are summarized. Trial details can be found in Chapter 8. It used un-slotted protocol implementation and a saturated load (file transfer) application. Multiple tests were performed at each parameter setting as permitted by allocated time during the sea-trial. Note that the same MAC protocol C code is used in the modem and the simulator through a unified simulator and modem software interface. Number of nodes $N = 3$, a maximum delay D of 0.4 seconds is used to emulate trial distances, contention time window $Wl = 10$ seconds and batch sizes $B = 5$, 10 and 40. A saturated traffic model is used. In the sea trials, estimated $P_d = 1$ and $P = P_D = 0.9$. This trial also used i=1, i.e., only one ACK. The modem used for the trial uses a Power Amplifier (PA) for transmission which takes about 300 ms to settle after being turned on. This delay is termed as t_{PA}. When PA is turned on, no reception is possible. For batch transmission, PA is turned on/off only at the start and end of the batch. This PA behaviour is captured in the simulations. Thus, for the analysis, even though the actual packet duration (highly robust control or data packets in the NUS/ARL modem) is 0.6 seconds, an effective time period of 0.9 seconds has to be considered in the contention part, i.e., $L = 0.9$. During batch transmission, since PA is turned on/off only at the beginning and end, a different data packet length $L_d = 0.6$ is defined. Thus, (4.9) is modified as $t_B = BL_d + t_A + (i-1)L + t_{\text{PA}}$. The results are shown in Figure 4.6. The analysis closely matches with simulation results as well as the sea trial results. The sea trials thus give strong validation for both simulation and the analysis. It should be noted that the performance at sea was slightly worse than the predictions, due to some of the simplifying assumptions made in the analysis and simulation models. During the sea trials, image files (see Figure 4.6) were transferred using this

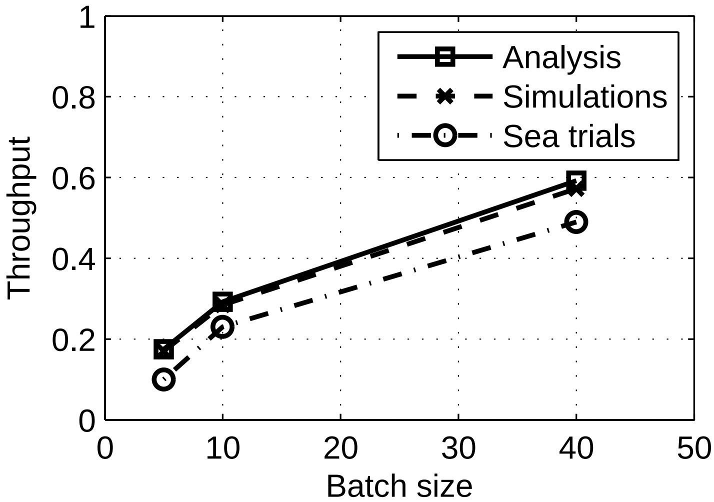


Figure 4.6: Analysis comparison with sea trials and simulations. Parameters: $N = 3, D = 0.4s, W = 10, P_d = 1, P = 0.9, i = 1, L = 0.9s, L_d = 0.6s$. A sample image that was transferred between modems in a recent sea trial is also shown. This file transfer used the MACA based protocol with the Early-ACK retry mechanism.

protocol with Early-ACK mechanism to retransmit failed packets.

In the system used for the experiments, control packets (RTS, CTS, ACK) were designed to have a small payload capacity to carry PHY tuning parameters, power control, and other network related information. Although many of these features were not used in the experiments, they were implemented for future use. Each of the DATA packets were also of the same modulation, coding and length, and used this payload capacity to carry application data. Section 4.7.4 discusses the use of longer DATA packets.

Tight time synchronization was not used in experiments due to practical limitations, and hence the implementation should be considered as un-slotted. As discussed in Section 4.3.3, the difference between slotted and un-slotted models was found to be marginal (see Figure 4.4). Thus, the slotted analysis used here still presents a reasonable approximation. It is certainly possible to add slotting to

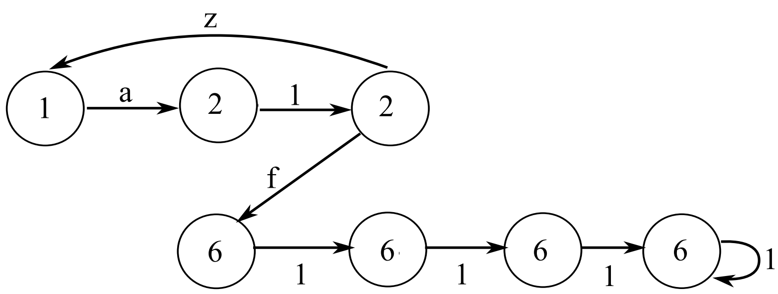


Figure 4.7: Part of Markov chain with dummy states for computing service time distribution

UAN experiments through the use of accurate synchronized clocks, but the added complexity may not be warranted as the performance gain may be marginal.

4.4 Service Time Distribution Analysis

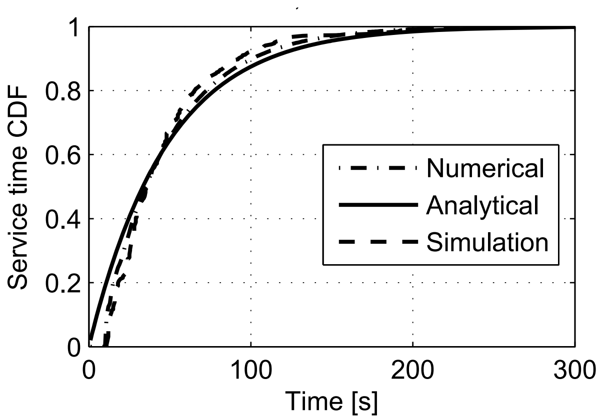
In this section, the service time distribution is computed analytically and numerically. The probability distribution is important to ascertain the validity of the assumptions used in queuing analysis in the next section.

Dummy states are introduced for the non-unity delay states with time delays t_A (states 2 and 3) and t_B or $t_{\text{B\_VCS}}$(states 4, 5, 6) in Figure 4.1. To illustrate this concept, a part of Figure 4.1 is shown in Figure 4.7, where transition probabilities between dummy states are 1. The example shows t_A of 2s in state 2 and t_B of 4s in state 6. The packet can reach the final absorbing state (complete transmission) through many paths. Each path will have a finite number of steps and the number of steps directly relates to the time (or the number of slots) taken to reach the absorbing state. This is the service time s_b for that path. Using Markov chain properties, the expected time over all paths to reach final state can be found.

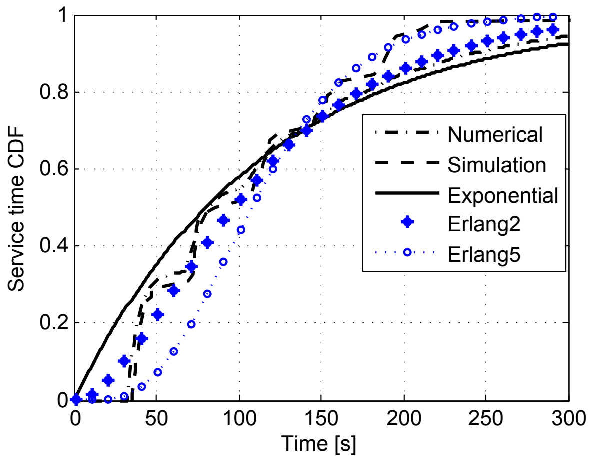
The Markov chain shown in Figure 4.1 is implemented in Matlab using dummy states as mentioned above. For illustration, one example and its transition matrix $\mathbf{M}$ is shown in (A.2). Note that this matrix is only one realization for a certain parameter set. For the experiments below where batch size B is varied, which effectively varies t_B, the number of total states and hence the transition matrix $\mathbf{M}$ is varying. This is also the case for variations that cause t_A etc to change. These matrices are dynamically formed by a Matlab program. Following that, $\mathbf{M}$ is used together with the additional transitions in Figure 4.2 for the retry scheme to form the complete Markov chain $\mathbf{M_R}$ as shown in Appendix A.2.

The matrix $\mathbf{M_R}$ is in the canonical form of an absorbing Markov chain with one absorbing state. Using standard Markov chain theory, the probability of being in the final state after starting from the initial state in n-steps can be computed. ${\mathbf{M_R}}_{i,j}^{(n)}$ of the matrix $\mathbf{M}^{(n)}$ gives the probability that the Markov chain, starting in state s_i, will be in state s_j after n steps. The CDF for a particular scenario is shown in Figure 4.8(a) with batch size $B = 10$ (In Appendix A.2, a smaller batchsize is used to keep the illustrative matrix small).

In queuing theory, exponentially distributed service time is commonly used. To compare, exponential distribution curves are also shown which uses the analytical mean service time from (4.13). The numerically computed CDF shows close similarity to an exponential distribution behaviour. Simulation is also used to estimate the distribution nature (results are shown in Figure 4.8) and it also shows a closely matched exponential behaviour. Based on such a comparative study, exponential service time distribution can be assumed as an approximation for further analysis. Note, that for these protocols, there is a finite minimum



(a) $B = 10$, Mean $= 48$s



(b) $B = 55$, Mean$= 116$s

Figure 4.8: Service time CDF. Parameters: $L = L_D = 0.5s, N = 3, D = 0.5s, W = 4, i = 3, k = k_D = 0.81$. "Analytical" curve uses Exponential fit.

service time due to the handshaking till an ACK is received even if the sequence RTS/CTS/DATA/ACK succeeds in one round. For larger batch sizes, the CDF deviates more from exponential but still presents a good approximation as shown. Even for large batch sizes, the numerical computation (based on the analytical model in this chapter) matches the simulation results extremely well.

Simulation is also used to estimate the distribution nature and it shows excellent match to the numerical analysis. In queuing theory, exponentially distributed service time is commonly used. To compare, exponential distribution curve is shown, which uses the analytical mean service time from (4.13). The numerically computed (and the simulated) CDF shows close similarity to an exponential distribution behavior.

Note that these protocols have a finite minimum service time due to the handshaking till an ACK is received even if the sequence RTS/CTS/DATA/ACK succeeds in one round. Such a larger batch size example ($B = 55$) is shown in Figure 4.8(b). The numerical computation (based on the analytical model in this chapter) matches the simulation results reasonably well. Although the numerically derived CDF somewhat deviates from an exponential distribution, it provides a reasonable approximation. A brief investigation was also made on Erlang distributions which offer a more general class of distributions that offer a better fit in some cases. The Erlang-K distribution is formulated as a sum of K exponential distributions. The maximum likelihood estimate (MLE) for the Erlang-K is the K that minimizes $K + Klog(A/M) - (K+1)log(K) + log(K!)$, where A is the arithmetic mean of the data, and M is the geometric mean of the data (Erlang, URL). Using this, $K = 2$ was found to be optimal in multiple sets of service time data with different batch sizes, and a plot is shown in Figure 4.8(b). An Erlang-K distribution with $K = 5$ also shown for comparison. Based on this comparative study, we can use exponential service time distribution as a basic approximation for further analysis, and use Erlang-2 ($K = 2$) as a slightly better approximation.

4.5 Queuing Analysis

The extremely low data rates of UANs are a bottleneck in many co-operative underwater missions and the data generated may be in excess of the capacity, thereby treatable as saturated traffic. File transfer applications also provide an example of saturated traffic scenarios. Metrics such as throughput in such cases can be analyzed as discussed in Section 4.3. In some scenarios, data generation rates could be lower than system capacity and can be treated as unsaturated. In this section this latter case is addressed.

Poisson arrivals with rate λ is considered as in standard queuing theory analysis and the service time distribution is modelled as exponential or Erlang-K with $K = 2$ (as illustrated in the previous section). Thus, the system is modeled as either M/M^B/1 queue (Gross and Harris, 1998) with exponential batch service or M/E_2^B/1 queue (Chaudhry and Templeton, 1984) with Erlang-2 batch service for queuing delay analysis.

4.5.1 Unsaturated queuing analysis

All nodes contend after each batch transmission as discussed in Section 4.3.1. The long durations between batch transmission opportunities for UAN nodes make it inefficient for a node not to contend until it has DATA to send. So in this protocol, a node will restart contention immediately after the current cycle even if there are no packets in the buffer. The expected time till it gets a successful CTS according to (4.13) is after $(N-1)$ other batch transmissions and contention cycles. During this period a node gets to accumulate DATA from Poisson arrivals in the queue.

When CTS arrives, a node starts sending DATA in the queue, while remaining open to new arrivals until the allowed batch transmission time is complete, i.e., new arrivals immediately enter service up to limit B and finish with others. In addition, there are failed DATA packets to be retried. Thus, in our model, the number of contending nodes is not dependent on the arrival rate at each node. The process consists of contention cycles followed by batch transmission of DATA packets. The validity of this pre-emptive model is evaluated in detail in Section 4.7.1.

The contention process consists of RTS and CTS exchanges between N nodes as described earlier and its average duration or its statistical distribution is independent of the ensuing DATA batch transmission. The DATA batch transmission is for a fixed duration and the total service time is the sum of the variable contention period and the fixed DATA transmission period. The definition of mean batch service time s_b is the average delay from the time a batch is intended for transmission (RTS contention starts) until it is considered successfully transmitted, i.e., until the first ACK is received for the batch. Even if no DATA is available to be sent when CTS is received, the transmitter and receiver as well as all the neighbors diligently execute the batch transmission phase. The transmitting node continues to remain open to arrivals, and transmits the packets if they arrive within the allocated batch transmission period. ACK will be sent by the receiver indicating the number of packets received (even if no packets are successfully received) after the batch transmission period is over. Thus, as long as the number of nodes in the contention phase remains the same (apart from environmental and system factors), the batch service time s_b does not change. This is the base metric for much of the analysis (including throughput) as seen earlier. This implies that the service time distribution is independent of DATA content in the batch transmission phase (i.e., there may or may not be sufficient packets in queue to fill the batch size B). The exponential (or Erlang-2) nature is fundamentally a result of the contention process. And hence it can be used as the basis for the queueing analysis for the non-saturated case. The exponential (Erlang-2) service time remains valid as long as all nodes contend during each contention cycle, which is the protocol model used. In the queueing system, the server keeps serving, whether or not there are packets in the queue, and is unaffected by arrivals or the contents of the batch transmission phase itself.

4.5.2 Expected waiting time

Expected values for steady state queuing length and waiting times can now be derived using queuing theory for batch service. First exponential service time is considered. The probability generating function of the steady state probability distribution $P(z)$ for M/M^B/1 (can be inferred from results by Gross and Harris (1998)) is $P(z) = \frac{1-r_0}{1-r_0 z}$. Service rate μ_b is defined as $1/s_b$. The constant r_0 is the root of the characteristic equation of M/M^B/1 as shown below.

$$\left(\mu_b z^{B+1} - (\lambda + \mu_b)z + \lambda\right) p_n = 0; n \geq 0 \tag{4.15}$$

The root r_o needs to be found numerically except for small values of B. When $B = 1$, (4.15) becomes quadratic. Roots of (4.15) can analytically be found up to $B = 4$ using Matlab's symbolic math tool box, but expressions are cumbersome.

For $B >= 5$, analytical roots are not always obtainable as is generally known in polynomial theory.

Required moments such as expected queue length and waiting time can be computed as follows. Expected system total length L_T (includes the packets in service)

$$L_T = P'(z)|_{z=1} = \frac{r_0}{1 - r_0} \tag{4.16}$$

For Erlang-K, the queue length can be found as (Chaudhry and Templeton, 1984)

$$L_T = \sum_{B}^{B+K-1} (z_j - 1)^{-1} \tag{4.17}$$

where z_j are the roots of

$$z^B - \left(1 + \frac{\lambda}{\mu_b}(1 - z)\right)^{-K} = 0 \tag{4.18}$$

Using Little's Law (Gross and Harris, 1998), expected waiting time W_Q (excluding service) can be found as follows for either Exponential or Erlang-K (s_b from (4.13))

$$W_Q = \frac{L_T}{\lambda} - s_b \tag{4.19}$$

Total waiting time W_T (the sum of queuing time W_Q and service time of one packet s_p is then (s_p from (4.14))

$$W_T = \frac{L_T}{\lambda} - s_b + s_p \tag{4.20}$$

The average intake batch size is reduced to Bk_D, since retries will occupy $(1-k_D)B$ on average for every batch, this modified batch size should be used in (4.15) and (4.18).

4.5.3 Waiting time variation with batch size

One example of how W_T varies with batch size is shown in Figure 4.9, with the same base parameters as in Section 4.3.3 and $\lambda = 0.05$. Simulations match the analytical results well and validate the accuracy of the analytical model. As seen, in this case the exponential and the Erlang-2 results are very similar (a small difference seen at $B < 10$). Another example is shown in Figure 4.10 with a very different set of parameters (such as $L \neq L_D$, $k \neq k_D$, larger $N = 10$, large $W = 23$s) and the analysis gives a good indication of expected behaviour.

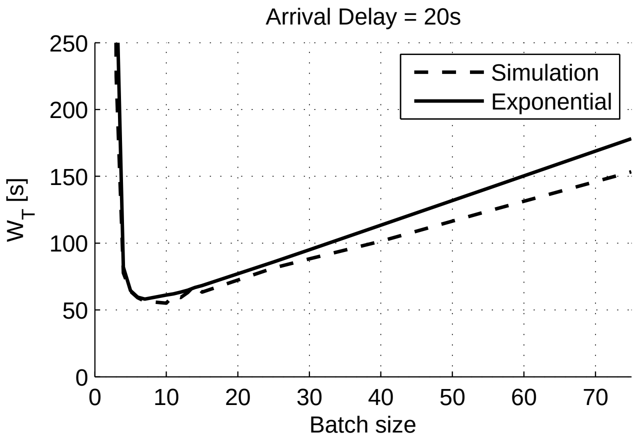
This shows that there is potentially an optimum batch size that minimizes the total waiting time. Saturated throughput analysis in Section 4.3.3 showed that by increasing the batch size arbitrarily, high levels of throughput can be achieved, but it does not give the cost in terms of total delay encountered for Poisson arrivals. Thus, for Poisson arrivals, it is best to optimize the batch size to minimize the waiting time. This is a key observation in this chapter. Using the analytical model, how different arrival rates and the number of nodes impact the optimum batch size in Figure 4.11 can be seen. The optimum batch size increases with decreased arrival delay (increasing λ) and also with the number of nodes.

Analytical insights into the optimum batch size using the exponential model are presented below. Define α and β as follows

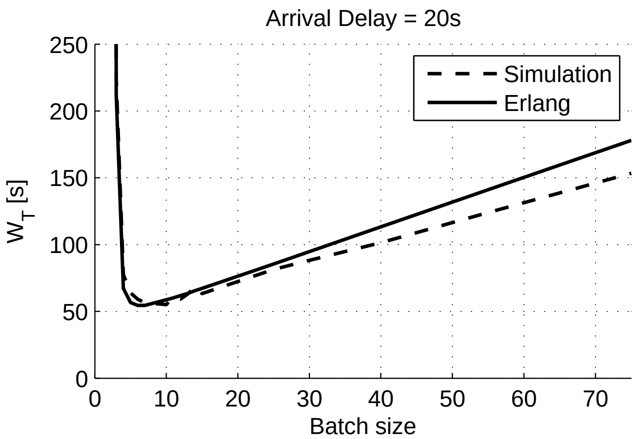
$$\alpha = \left(1 + (1-k)^i\right)\gamma + N\left(t_A + (i-1)l\right); \quad \beta = NL_D \tag{4.21}$$

Using α and β, (4.13) can be simplified as

$$s_b = \alpha + \beta B \tag{4.22}$$



(a) Exponential Fit



(b) Erlang Fit

Figure 4.9: W_T (in seconds) vs. Batch size (B), Parameters: $L = L_D = 0.5s, N = 3, D = 0.5s, W = 4, i = 3,\ \lambda = 0.05,\ k = k_D = 0.81$

Since on average $k_D B$ packets are transmitted in time s_b, for a stable system, the arrival rate must satisfy $\lambda < \frac{k_D B}{s_b}$. Therefore, we get $\lambda_{\max} = \max(\lambda)$ for given B as follows

$$\lambda < \frac{k_D B}{\alpha + \beta B} = \lambda_{\max} \tag{4.23}$$

and $B_{\min} = \min(B)$ for a given λ as follows (shown along with the optimum in

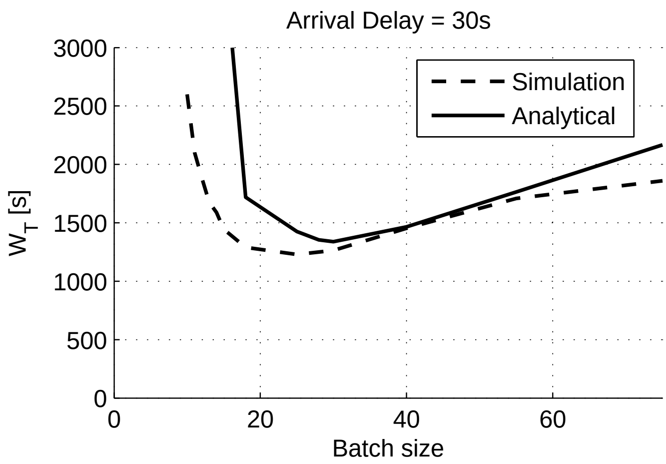


Figure 4.10: W_T (in seconds) vs. Batch size (B), $N = 10, D = 1.0s, L = 0.5s, L_D = 1.5s, k = 0.81, k_D = 0.9, W = 23, i = 3, \lambda = 0.033$

Figure 4.11)

$$B > \frac{\alpha}{k_D/\lambda - \beta} = B_{\min} \tag{4.24}$$

From (4.23), $B_{\min} = 1$ if

$$\lambda < \frac{k_D}{\alpha + \beta} \tag{4.25}$$

For such λ, from (4.15) for $B = 1$ (non-batch mode use of the protocol), using (4.22) we obtain $r_0 = \frac{\lambda}{\mu_b} = \lambda(\alpha + \beta)$ and the corresponding waiting time relationship to λ can be shown to be as follows based on (4.20) (s_p from (4.14))

$$W_T = \frac{1}{k_D}(\alpha + \beta) + \frac{\lambda(\alpha + \beta)^2}{1 - \lambda(\alpha + \beta)} \tag{4.26}$$

Similarly, for $B \gg 1$ in (4.15), an approximation $r_o = \frac{\lambda}{\lambda + \mu_b}$ can be used and the corresponding waiting time from (4.20) can be shown to be as follows, which is independent of λ

$$W_{T\_A} = \frac{1}{k_D}(\alpha + \beta B) \tag{4.27}$$

The independence of above waiting time from λ occurs for such batch sizes, because every arrival gets served in the very next batch transmission for very high B. The

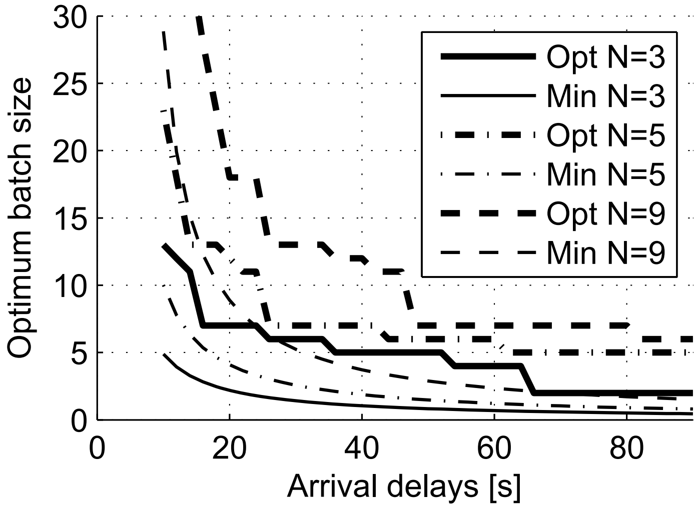


Figure 4.11: Variation of optimum ('Opt') and minimum ('Min') batch size with the number of nodes N and arrival delays. Parameters: $L = L_D = 0.5s, N = 3, D = 0.5s, W = 4, i = 3,\ k = k_D = 0.81$.

behaviour is illustrated in Figure 4.12 where $B_{\min}$ (4.24) and the asymptotic lower bound (4.27) is indicated (the latter as a dash-dot line). This general behaviour was observed in all numerical and simulation results. If W_T for $B = 2$ (based on (4.27)) is higher than W_T for $B = 1$ (based on (4.26)), it can be inferred that for $\lambda < \frac{\beta}{k_D(\alpha+\beta)^2+\beta(\alpha+\beta)}$, there will be no such optimum behavior (presence of a minimum for W_T as illustrated in Figure 4.12 or the specific example in Figure 4.9) and W_T monotonically increases with B. For such small λ, $B = 1$ is the optimum batch size. For larger λ and still within the upper limit specified by (4.25), there may be an optimum batch size behavior similar to that seen in Figure 4.9. A simple engineering approximation of this optimum batch size $\overline{B_{\text{opt}}}$ can be obtained as follows since an analytical solution is intractable. A near-optimum batch size is obtained from the simple graphical construction shown in Figure 4.12; $\overline{B_{\text{opt}}}$ chosen as mid-point of $B_{\min}$ and B_x as shown.

$$\overline{B_{\text{opt}}} = 1 + \frac{k_D\lambda(\alpha+\beta)^2}{2\beta\left(1 - \lambda(\alpha+\beta)\right)} \tag{4.28}$$

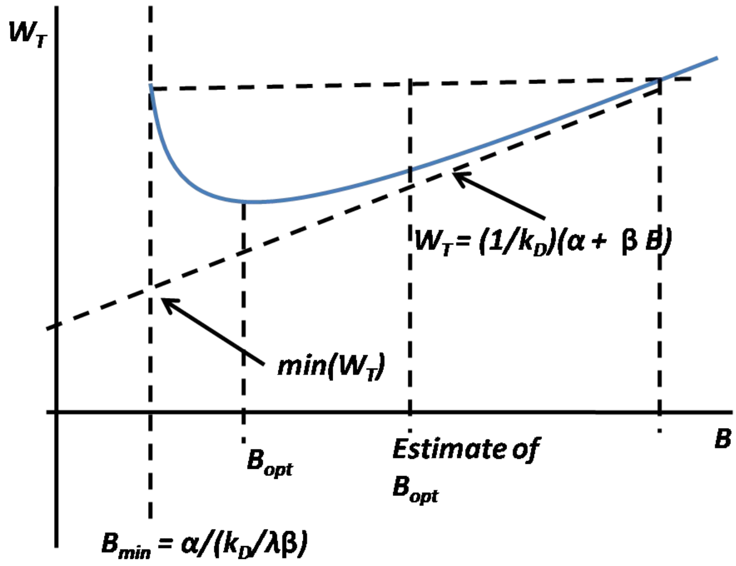


Figure 4.12: Waiting time behaviour illustration – the solid curve is W_T. Other key characteristics are as indicated.

For $\frac{k_D}{\alpha+\beta} < \lambda < \frac{k_D B}{\alpha+\beta B}$, $B_{\min} > 1$ and the optimum batch size needs to be estimated using numerical computation as described in previous section. A general lower bound $\underline{W_T}$ can be found using $B_{\min}$ (4.24) in the asymptotic relationship of (4.27) as indicated in Figure 4.12.

$$\underline{W_T} = \frac{\alpha}{k_D} + \frac{\beta}{k_D}\left(\frac{\alpha\lambda}{k_D - \beta\lambda}\right) \tag{4.29}$$

4.6 Protocol Enhancement: MACA-SEA

Motivated by TDMA's superior performance, through some modifications to the MACA-EA algorithm, a certain degree of self sequencing or a TDMA-style "taking turns" behaviour can be achieved. Much of the loss of bandwidth in MACA-EA happens because of contention. If a more fair contention process is used, where after a node's batch transmission, it gives way to other nodes to transmit before restarting contention, performance can be improved. Such a less greedy algorithm was successfully tested out and it has been termed MACA-SEA, (for MACA-Sequenced-Early-ACK). Results have shown that performance improves up to 20% over MACA-EA.

4.6.1 Algorithm outline

In a realistic deployment, each node may be required to communicate with at least one other node, whose identity (ID) is known. A node may not know the total number or the IDs of all the other nodes in its neighbourhood. Let's assume that unique MAC identifier numbers are ensured. The core idea behind the algorithm can be summarized as follows: after a node successfully transmits, it abstains from contention for $N - 1$ "turns", where number of nodes N is a local estimate based on the recent history of overheard communications. "Turns" refer to successful transmissions by other nodes. It essentially makes the algorithm less greedy and introduces cooperation and fairness into the contention process. After a node successfully completes its turn to transmit DATA packets, it sets a maximum timeout (TIMEOUT) to allow other nodes to complete their turns, after which it re-enters contention. This takes care of any deadlock situation, such as a node not wanting to utilize its turn. The key components of the algorithm are shown in Figure 4.13. An important feature of this algorithm is that contention window $W = 2$ can be used regardless of N unlike MACA-EA, since when the "taking turns" behaviour is effective, there is no contention with other nodes. The larger contention window in MACA-EA is required to reduce the collision probability for N-node contention. The simulations showed that the sequence does get broken

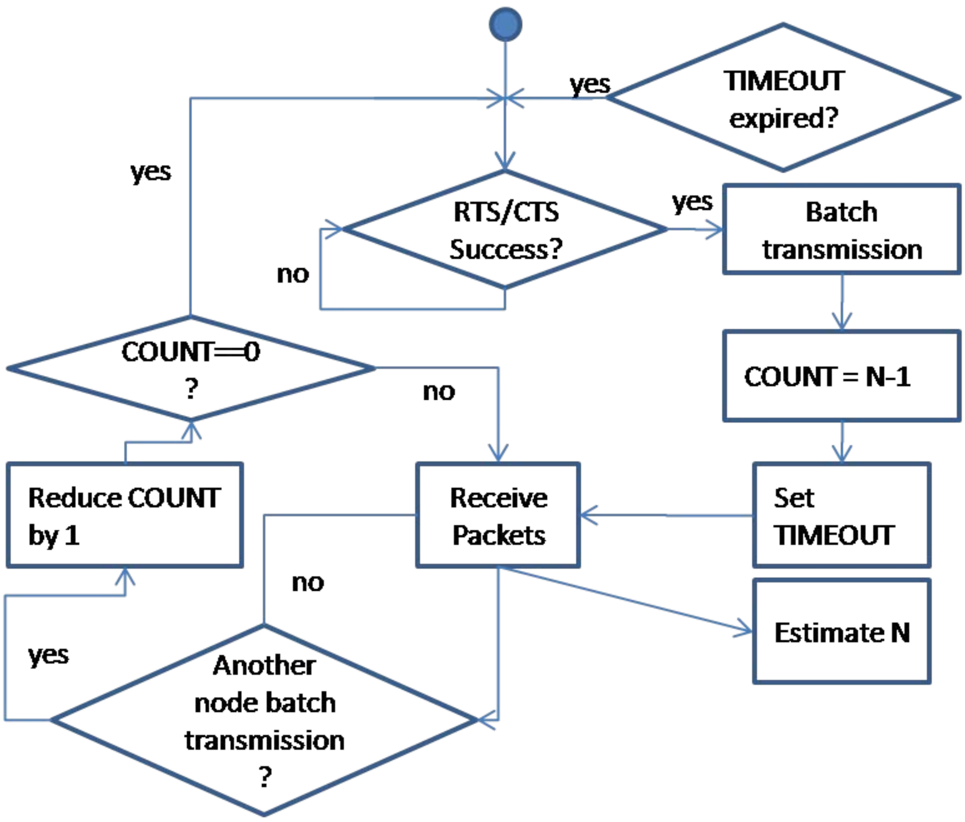


Figure 4.13: MACA-SEA flowchart

occasionally due to packet loss and errors, but on average $W = 2$ gives the best result.

4.6.2 MACA-SEA performance

Here a snapshot of the results for throughput T is provided. In the simulations for the following results, the following parameters were used: combined probability of detection and decoding $k = 0.64$, number of nodes $N = 5$, simulation time $t = 4000$ seconds. Contention Window W is as indicated in the legend. As seen in Figure 4.14, there is an improvement for MACA-SEA of up to 20% over the previous algorithm. MACA-SEA can automatically adapt towards a pseudo-TDMA sequenced behaviour, providing very high throughput for a MACA-based

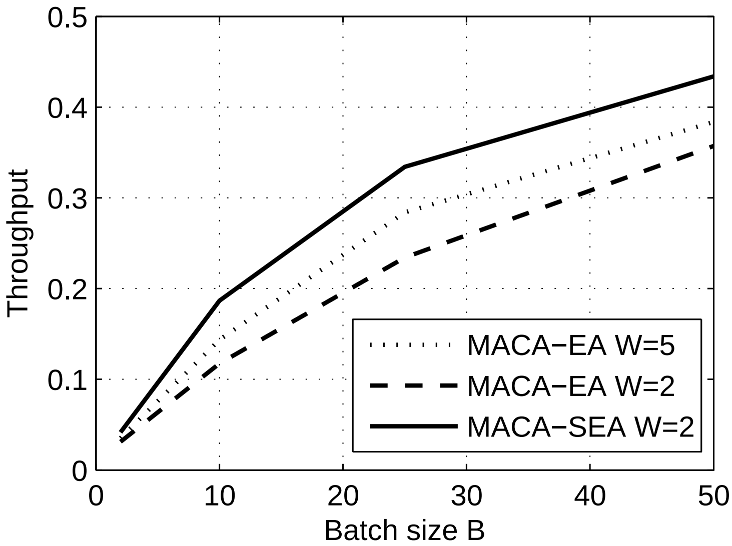


Figure 4.14: Performance of MACA-SEA and MACA-EA. Parameters: $L = 0.5s, N = 5, D = 0.5s, i = 3, k = 0.64$. Contention window W as indicated.

ad hoc protocol.

4.7 Discussion

4.7.1 Pre-emptive contention

If a certain node is inactive and has no DATA to send for extended periods of time, MACA-EA protocol does not advocate that it keep contending. In our analysis, that node will not be part of the equation. Only active nodes are considered in counting N, and by definition they are active only if they have DATA to send according to the specified arrival rate, assumed same for all nodes in this analysis. Since it takes a finite time t after a batch transmission before the next successful contention, the average queue size will be λt during that period alone. The extreme case of having no packets may occur with a small probability in this Poisson arrival model. Even if there is one packet to transmit, the server attempts to be fair to it by successfully contending in order to transmit it. In UANs, the opportunity to transmit may be very infrequent due to losses and low data rate, and hence for very active missions, it was considered desirable to proceed with contention even if there were no packets at the start of the contention. The expectation is that by the time contention is completed, there may be non-zero packets to deliver. Adaptive batch sizes can be useful in such scenarios for a protocol in real applications. Adaptive batch size analysis is not undertaken in this chapter. One of the potential scenarios is an AUV mission where there are a small number of nodes in a single collision domain, all actively exchanging data to each other. TDMA could be used for such a scenario. The motivation in moving from TDMA to MACA is to eliminate the extreme dependence on time synchronization and thereby increasing robustness, and to add ad hoc capability (nodes can arrive depart easily). In MACA-EA, unlike TDMA, bandwidth usage can be adaptively stopped if there is nothing to send for extended periods of time.

It should be noted that after successful contention (CTS received), if there is still no DATA, the batch transmission period will be honored by all nodes based on VCS principle. ACK will not be sent. Note that such occurrences will be related to packet arrival rate, and for lower rates, the optimum batch size also will be low, and $B = 1$ for λ below a threshold as discussed in Section 4.5.3. So the wasted batch transmission time is not significant. Nevertheless, it may seem that the pre-emptive contention used by MACA-EA is not ideal. Therefore to test its merit, MACA-EA is compared with MACA-EA-WAIT, a variant in which the protocol waits for B packets to be in the buffer before contention starts. Two

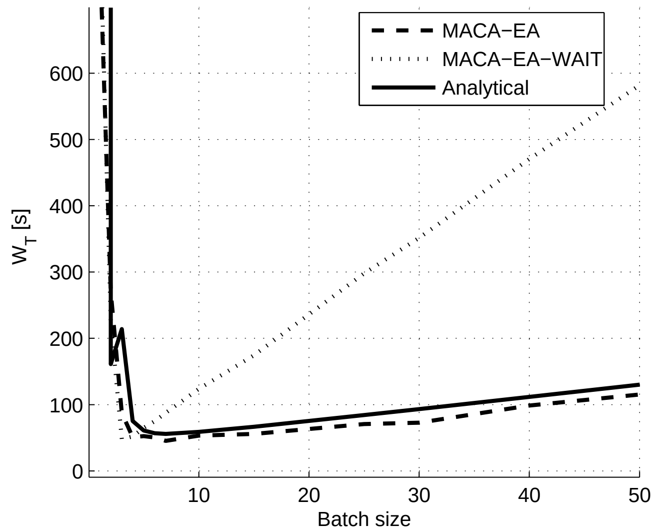


Figure 4.15: Comparison with MACA-EA-WAIT, arrival delay = 20s. Parameters: $L = 0.5s, N = 3, D = 0.5s, W = 6, i = 3,\ k = 0.81$.

sample results are shown in Figure 4.15 and Figure 4.16. Parameters: Number of nodes $N = 3, k = 0.81, i = 3$.

The results indicate that the lowest waiting time for both MACA-EA and MACA-EA-WAIT are comparable. MACA-EA-WAIT has a slightly lower optimum batch size at which the minimum occurs. For high arrival rate (saturation throughput), MACA-EA-WAIT waiting time increases sharply with increasing batch size. As larger batch sizes give better throughput efficiency, the MACA-EA protocol can give reasonable throughput efficiency for saturated load and at the same time, reasonable waiting times for Poisson arrivals. Thus, the pre-emptive contention model has an overall positive effect on the proposed protocol.

The reasons for the behaviour of MACA-EA-WAIT are as follows:

- When contention starts only after B pkts are in queue, new packets that arrive between contention start and successful batch transmission need to wait till next turn, which will be on average $N - 1$ turns later.

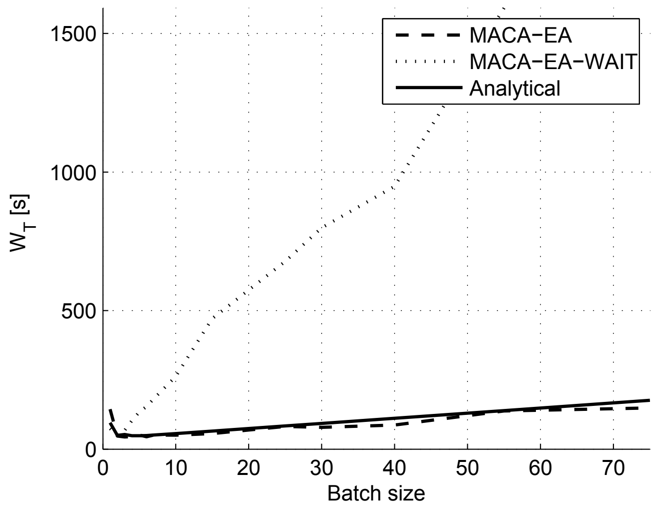


Figure 4.16: Comparison with MACA-EA-WAIT, arrival delay = 50s. Parameters: $L = 0.5s, N = 3, D = 0.5s, W = 6, i = 3,\ k = 0.81$.

- Packets that need to be retried cannot be tried immediately after the current transmission, but need to wait for buffer to get full.

- For higher batch sizes, the time to wait for buffer to get filled increases proportionally, and the combination of the above two factors seem to outweigh some of the gains that is possible by abstaining from contention.

- Contention and associated collisions is only one of the loss mechanisms. Packet detection and decoding losses also contribute greatly to the behaviours. In UANs, the latter could even dominate collision losses in small networks. (Note that analysis for 802.11 etc typically focuses only on collisions, and this factor is ignored in typical UAN MAC analysis).

4.7.2 Optimum RTS window

The optimum RTS window for throughput maximization can be derived as follows. Using X and U to represent terms which has no W' in the service time equation

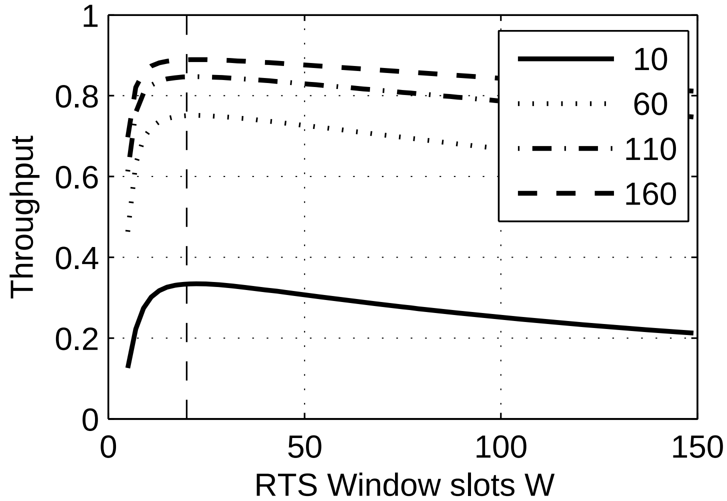


Figure 4.17: Variation of network throughput with W (legend shows batch size B). Parameters: $L = 0.5s, N = 10, D = 0.5s, W = 4, i = 3,\ k = 0.81$.

(4.13), $s_b = X \frac{1}{W'} \frac{W'}{W'-1}^N (W'^2 + W' - 2) + U$. From (4.5), T can be written as $T = \frac{kBL}{X \frac{1}{W'} \frac{W'}{W'-1}^N (W'^2+W'-2)+U}$. Using $\partial T/\partial W' = 0$, the optimum for positive W' can be obtained as

$$W' = \frac{1}{2}\left(N + \sqrt{N^2 + 8N - 8}\right) \tag{4.30}$$

The approximation $W' \approx N + 2$ can be used for $N \gg 1$. The optimum contention window size is thus $W = 2W' - 1 \approx 2N + 3$. It can be observed how throughput changes with W in Figure 4.17. Different lines represent different batch sizes. For $N = 10$ and other parameters as in Section 4.3.3, Figure 4.17 shows the throughput using the analytical formula and the optimum $W \approx 2N + 3$ can be seen. It can be seen that varying batch size has no effect on optimum RTS window and it depends only on the number of nodes N, i.e., on neighbourhood density. This result helps the MAC to dynamically select an optimum back-off window based on its estimate of the number of neighbouring nodes (estimated based on the received packets over time since each packet has the sender identity) and eliminates the need for freezing back-off as employed in 802.11.

4.7.3 Physical Carrier Sense

A variation where PCS is used at the end of RTS phase is possible as used in the model by Bianchi (2000). It can be assumed that a transceiver is in receiving mode unless a transmission is made. If the back-off counter expires at a time when a reception was in progress, it can be considered as a PCS that prompts a back-off. Simulations show that with the protocol variations and the parameters used here, PCS just before sending RTS provides only a small gain on top of VCS based on RTS, CTS and DATA. Hence PCS was not modelled.

4.7.4 Single Long DATA Instead of Batches

In many prior analyses, single long data packets were used instead of a batch of short DATA packets. Many commercial modems can transmit longer DATA packets compared to control packets. Forward error correction (FEC) codes are used to make such DATA packets robust. Control-DATA collisions do not necessarily destroy the DATA packet as a whole, but introduce errors that are recoverable through FEC. The analysis model in this chapter can be adapted by setting $B = 1$ and choosing appropriate k_D and L_D.

The following is a brief qualitiative comparison between batch of short DATA and single long DATA packets. Using short DATA packets in a batch gives a smooth degradation in performance in terms of losses due to control-DATA packet collisions, since at least some of the packets in the batch may be successfully delivered. If single long DATA packets are used instead of a batch of shorter ones, beyond a certain error threshold the entire large packet will be lost and the loss behavior will be abrupt. On the other hand, longer DATA could allow FEC coding to be spread across a larger packet and therefore be more robust to errors. However, this involves higher processing power also as FEC codec complexity in general is an increasing function of packet duration.

Long single DATA also cannot allow features such as DATA-VCS. Packet train based DATA-VCS is a technique used in terrestrial radio wireless. DATA-VCS helps in achieving VCS even if both RTS and CTS from neighbors are missed, by listening in to any of the DATA packets that follow. If there are many short DATA packets, the probability of receiving some of them at least is high.

It may be good to have "synchronization portions" in such long DATA packets, to avoid the wastage of losing the packet as a whole. For example, the equivalent single long packet for a batch size $B = 50$ with $L_D = 0.5$, is a 25second packet and it will not be prudent to lose it fully, just because the detection preamble was missed (probability P_d). In other words, it is good to keep a batch model even for longer contiguous data transmission, and keep L_D not too large. A study on optimum L_D is not undertaken in this thesis.

4.7.5 Optimum number of ACKs

Equation (4.13) can be re-written as

$$s_b = \left(1 + (1-k)^i\right) V' + Y' + (i-1)Z = q^i V + Y + (i-1)Z \qquad (4.31)$$

where $q < 1$, V', V, Y', Y, Z are factors used for convenience. This form for s_b clearly shows that for a certain $i >= 1$, there will be a minimum. The exact value depends on the parameters such as k, W, N, l, B, D etc. It was found that for some of the parameter combinations used in the simulations, $i = 3$ gave the minimum (e.g., $k = 0.7, L = 0.5, D = 0.5$ $(l = 1, t_A = 2), N = 3, W = 9, B = 5$). In some cases $i = 2$ or $i = 4$ or other values may be better. It is thus possible to set optimal i according to network conditions, though a fixed i is used in this chapter for simplicity. The optimum i can be derived as follows,

$$\begin{aligned} \frac{ds_b}{di} &= q^i \log_e(q)V + Z = 0 \\ i &= \left\lceil \frac{\log_e \frac{-Z}{\log_e(q)V}}{\log_e(q)} \right\rceil \end{aligned} \tag{4.32}$$

4.7.6 Forward Error Correction (FEC) and Power Control

FEC and power control have a significant role in achieving optimum performance in UAN MAC protocols. RTS/CTS exchanges can carry information to do both dynamic FEC and power control. Packets typically have two parts - a detection preamble and an information part. The information signal could be control packets using a predetermined modulation and Forward Error Correction (FEC) scheme or user DATA in variable modulation and FEC schemes. The decoding of control packet RTS could help estimate the BER and the CTS can specify the FEC scheme to use. Power control may be needed to adapt the range required for routing and node connectivity. Power control can also be done effectively via the RTS/CTS exchange, by letting the RTS use maximum power and the CTS specify the required power. These two features are important advantages offered by the

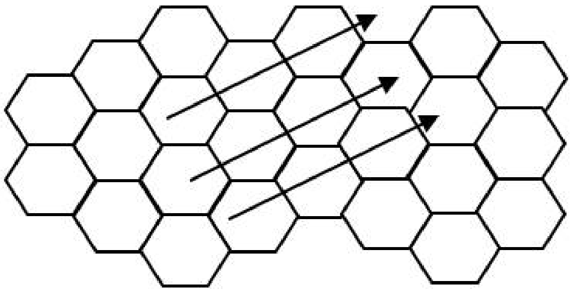


Figure 4.18: Hexagonal cell model and sample traffic pattern

protocols adapted from MACA, which may more than compensate for the loss in round trip delay in the handshake, in highly time varying UAN environments.

4.7.7 Multi-hop and hidden nodes

Although multi-hop networks are not the focus of this thesis, some thoughts on how the work in this chapter could be extended to multi-hop networks in the future are outlined below. The primary network UAN architecture considered in this thesis is as described in Figure 3.1, where the acoustic networks are viewed as separate single collision domains connected to others via surface gateways. Although multi-hop is possible, the applications and implementations being pursued as part of this thesis focused on single-hop networks. The analysis presented here is thus meant primarily for such isolated single collision domains and network architecture. Nevertheless, for large multi-hop acoustic networks, the results here could be used as an approximation in a N-node one-hop neighborhood. The following is a rudimenary outline on how this may be attempted for shallow water acoustic networks where 2D topology is reasonable. An in-depth analysis is out of scope for this thesis and may be undertaken in the future. A standard cellular hexagonal geometry may be used as shown in Figure 4.18. Let us assume that the packet arrival rate at a node is Poisson with rate λ. Assume a traffic flow pattern as shown. Let H be the number of hops to the intended recipient of each transmission. Consider two options, one where each node takes one hop at a time ($h = 1$) and another where a single hop is made across H nodes ($h = H$). In the traffic pattern shown, the aggregate arrival rate at each node can be modelled as $\lambda H/h$ considering routed traffic. In order to perform non-saturated queuing analysis we require $\lambda H < \mu$. The number of neighbors N in an effective collision domain is related to the number of hops h as shown below.

$$N = 6(1 + 2 + ... + h) = 3h(h + 1) \tag{4.33}$$

It is commonly stated that nearest neighbor connectivity is the best for UANs (for example, to minimize power (Dolc and Stojanovic, 2007)) and hence $h = 1$ and therefore $N = 6$ and the aggregate traffic is λH. Thus, by using the effective number of nodes in the one-hop neighborhood and aggregate traffic ($N = 6$, traffic is λH in the example), approximate estimates of MAC performance in multi-hop networks using the equations in this chapter could be obtained.

4.8 Conclusion

Enhancements for MACA-based protocols for UANs such as monitoring of DATA packets during contention phase (to aid VCS) and Early-Multi-ACK ARQ model were proposed in MACA-EA. A good analytical model that relates the parameters N, D, L, L_D, B, k, k_D, W, t_A to expected service time and throughput was derived and the impact of parameters batch size (B) and detection and decoding probability (k, k_D) on throughput was illustrated. Service time distribution was shown to be exponential with an analytically known mean. Queuing analysis for Poisson arrivals was done and the effect of batch size B was shown. The throughput of the MACA protocol increases with batch size, but so does the waiting time and the existence of an optimum batch size was shown. One of the important novel contributions is the formulation of a model that helps compute the total queuing delay for the retry based protocol. Optimum value of back-off counter W based on the number of nodes N has been derived. The protocols were implemented in acoustic modems, and medium range field trials corroborate the simulations and analysis well. System analysis is now possible without needing to resort to extensive simulations. It should be noted that for very long propagation delay regimes, the slotted version of the protocol could be less effective, and the analysis results less accurate. A variant called MACA-SEA was also proposed, which further improves upon MACA-EA by using a less greedy contention algorithm. It tries to achieve a pseudo-TDMA "taking-turns" behaviour for the protocol.

In future, other MACA-based protocol proposals can be analyzed through suitable modifications to the models used here. MACA-SEA also could be characterized analytically and requires more extensive testing to study its robustness and limitations. The use of PCS could be further investigated. Freezing back-off variation may be investigated for highly dense networks (Bianchi, 2000).

For large ad hoc UANs, where TDMA time frame assignment is cumbersome, supporting ad hoc function is nearly impossible and time synchronization issues lead to poor robustness, MACA based protocols with packet trains and suitable variations such as Early-Multi-ACK is an excellent choice. Accurate analytical models of the protocol greatly helps in understanding its relationship to environmental and system parameters. Knowing the system performance beforehand can help to pre-determine possible throughput and waiting time so as to control data generation rate and plan realistic communication strategies for ad hoc UANs.

Chapter 5

Adaptive Multi-mode Medium Access Control for Underwater Acoustic Networks

There is a need to work towards MAC protocol standardization as set out in the objectives of this thesis. Though the MACA-EA protocol presented in the previous chapter has good performance in its class of protocols, in order to satisfy the requirements of a general heterogeneous UAN, it may need to work in conjunction with other time domain protocol modes. It is also important to further verify its performance in relation to other time domain protocols such as TDMA and centralized topology MACA-based schemes and incorporate it as part of a wider set of time domain protocols operating as a “suite” of protocols, that can adapt to different environments and requirements. This work has been submitted for publication (Shahabudeen et al., 2011a).

5.1 Introduction

Underwater modems currently utilize diverse physical layer standards and medium access control (MAC) protocols in overlapping frequency bands and cannot co-exist or communicate with each other over standardized protocols. Physical and MAC Layer protocol standardization attempts are in progress, such as the JANUS initiative at NURC (McCoy, 2009)[1]. As part of that initiative, a suitable candidate for an adaptive MAC protocol suite for underwater acoustic networks (UAN) was previously proposed (Shahabudeen et al., 2009). This chapter presents important enhancements and an in depth analysis of this adaptive protocol suite, which will henceforth be referred to as Adaptive Multi-mode MAC, or MAC-AMM for short.

In a universal UAN scenario, a single MAC protocol and a single physical layer type cannot cater to differing deployment and traffic requirements. The first problem is that there exists numerous underwater acoustic modem implementations around the world today that are incompatible with each other. The first problem is how can we provide adaptation mechanisms to allow them to co-exist and communicate, if deployed in the same geographical area. The answer lies in standardization. The second problem, quite independent of the first, is how MAC protocols can autonomously switch between centralized topology and distributed topology. For example, in a centralized topology network, the MAC Controller (MC) could malfunction. Can the client nodes communicate with the other nodes

[1]The author was part of an invited group formed by NATO Undersea Research Centre (NURC) looking into standardization of MAC protocols for UANs. More details at http://www.januswiki.com/

in the absence of the MC? Or in a network currently operating in distributed mode, if a gateway buoy is introduced to operate as an MC, can the MC take charge of the MAC co-ordination seamlessly? Related to this, there is the question of how the centralized topology performance compares with the distributed mode. The third problem is related to traffic intensity. As we shall see later on, the relative performance of protocols is dependent on traffic intensity. As traffic varies, can modems adaptively switch between protocol modes to provide the best performance? Requirements for a robust and flexible heterogeneous UAN are no longer uncommon and the above three MAC problems require a unified solution.

In this chapter, the proposed solutions for the first and second problems will be outlined briefly. The bulk of the chapter focuses on the characterization of the different MAC modes, both theoretically and analytically, to ascertain their relative performance and to provide insights into adaptation mechanisms for the third problem. Further enhancements to the protocol modes being used, also need to be pursued in the continued interests of pushing performance boundaries.

It is necessary for a universal robust UAN MAC to be able to work without time synchronization but at the same time, be able to utilize time sync if it is available to improve performance. The need for ad hoc functionality in the network, i.e., nodes joining and departing is very important. The issue of geographical scalability and robustness also cannot be ignored. Thus, in the hunt for a basic protocol model, MACA-based schemes stand out as a good choice from among the many alternatives. This has been discussed at length in Chapter 3.

The MAC-AMM protocol has two levels of operation, Level-1 is to achieve co-existence and Level-2 is to achieve communications among heterogeneous assets. In this adaptive protocol, nodes dynamically adapt their physical layer and MAC protocol modes based on node capability, network scenario and traffic intensity. Level-1 MAC has a single mode – DATA-ACK. Level-2 MAC has three modes – a distributed MACA-based mode (MACA-EA), a centrally controlled polled mode (MACA-C) and a low traffic DATA-ACK mode. The main focus in this chapter is on Level-2 MAC that allows full-fledged communications among all nodes in a network.

First, the different modes as outlined above are described and how the dynamic adaptation works in Section 5.2. Following that throughput performance analysis is presented in Section 5.3. Then mode adaptation based on traffic intensity is illustrated in Section 5.4, where waiting time is used as the metric. A state dependent variation of the DATA-ACK mode and a self-sequencing variation of the MACA-EA protocol that provide improved performance are presented in Section 5.5.

5.2 Protocol Modes in MAC-AMM

5.2.1 Level-1 compliance

Co-existence is defined as the ability to perform communications among modems of the same physical layer type, while operating in a region where there are modems using different physical layers. For the co-existence mode, all modems need to adopt the same detection preamble for a given frequency band. If that is not

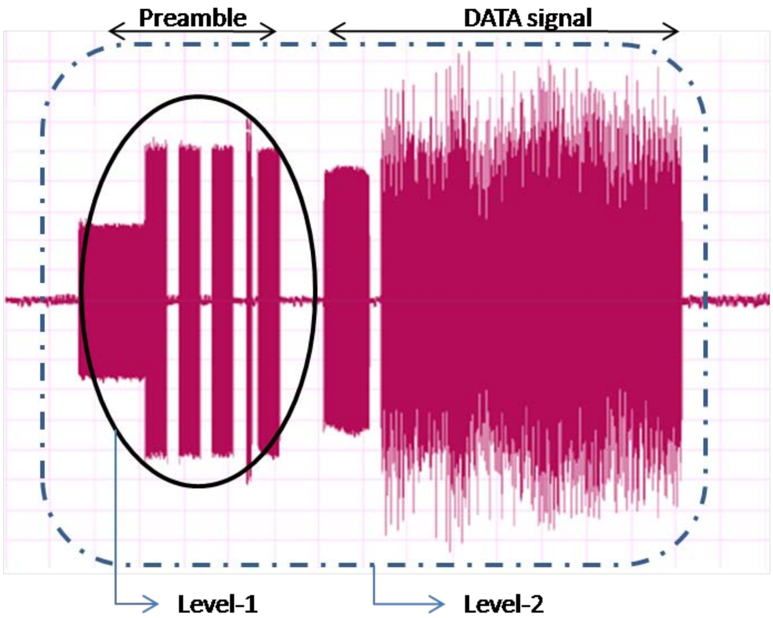


Figure 5.1: A sample physical layer packet structure shows the preamble and data signal portion. Physical layer compliance levels are as indicated.

possible, modems can implement an alien signal detection feature, i.e., to be able to detect a signal in its frequency band that is not of its physical layer type. This could be based on energy detection, for example. There could also be wakeup tones that are also present as part of the preamble structure, such as those being proposed in the JANUS initiative. Energy detection can also be used alongside the detection preamble to monitor the signal following the preamble, to determine end of the packet, for example. Such a minimal compliance at the physical layer is termed Level-1. This concept is illustrated in Figure 5.1.

With such a minimal compliance at the physical layer, it is possible to use a DATA-ACK protocol to communicate among modems of the same type, while operating in an environment consisting of alien nodes. In this mode, nodes use a random back-off before transmitting a DATA packet. The details of the back-off procedure are the same as in Section 5.2.3 for a RTS packet. In fact this mode can be viewed as similar to the Basic Access Scheme in 802.11 (Bianchi, 2000). At high traffic intensity, this protocol will have lower performance compared to other options as we shall see later on, but at this level of minimal physical layer compliance, it offers perhaps one of the best solutions to perform communications and handle interference between alien modem types.

5.2.2 Level-2 compliance

To enable communications among heterogeneous assets operating in the same geographical area, modems will be required to implement a standardized physical layer (along with their proprietary physical layer or as the only system and on top of the alien signal detector or detection preamble). Such suitable physical layer standards are currently being addressed under the JANUS initiative at NURC. This is termed Level-2 compliance, as illustrated in Figure 5.1.

Level-2 MAC has a distributed MACA-based mode (MACA-EA), a centrally controlled polled mode (MACA-C) and a low traffic DATA-ACK mode that nodes dynamically adapt based on deployment, node configuration and traffic as explained in Section 5.2.6. In the centralized mode, a cell is defined to consist of a MAC Controller (MC) and the nodes within its control. Many of the centrally controlled MAC protocols use a polling scheme, where the MC polls the client nodes (Kebkal et al., 2005). Some of the distributed protocols are ALOHA, CSMA, MACA (Karn, 1990), FAMA (Garcia-Luna-Aceves and Fullmer, 1998) etc. Among distributed protocols, some protocols such as MACA and FAMA involve handshaking using control packets before data transmission. Centrally controlled modes typically perform better than distributed modes by eliminating contention. However, in a generic network environment with heterogeneous nodes, a centrally controlled protocol alone might not be usable and distributed modes may be needed. Prior work addressing such large scale ad hoc dynamic underwater networks includes the Seaweb project (Rice et al., 2000). The terrestrial IEEE 802.11 family of protocols also use such combination protocols in the form of Point Coordination Function (PCF) and Distributed Coordination Function (DCF). Chapter 3 had discussed in depth on the choice of suitable MAC protocols, how time domain protocols are perhaps best suited for UANs and how MACA based protocols serve well in adhoc UANs, especially since time synchronization is not always available..

5.2.3 Distributed mode of Level-2 MAC: MACA-EA

The MACA-EA protocol discussed in the previous chapter forms the basis of the distributed mode in the Level-2 MAC protocol suite. The protocol is based on MACA using RTS/CTS (request to send/clear to send) exchange (Karn, 1990). The basic model used is RTS/CTS/DATA-TRAIN/ACK. The transmitter sends RTS and the receiver sends back CTS. The transmitter then sends a batch of DATA packets (DATA-TRAIN). The receiver then sends a single acknowledgement (ACK) which indicates failed packets in the batch. Similar protocols with packet trains that employ ACKs after every packet (RTS/CTS/DATA/ACK/DATA/ACK. . .) are not efficient for UANs due to the two-way propagation delay overhead and thus ACK is used only at the end.

The RTS contention algorithm and other details were discussed in the previous chapter.

5.2.4 Centralized mode of Level-2 MAC: MACA-C

In this mode of the protocol suite, an MC controller controls the collision domain or "cell". An RTR (Request-to-receive) initiates all communication sequences for the uplink (towards MC or between nodes in the same cell). All nodes monitor for MC control packets to detect presence of a controlling MC and then switch to the controlled MAC mode. Channels not mentioned in RTR can be assumed to be uncontrolled by the MC and nodes may make use of them as they wish (e.g. using MACA-EA). The nodes that operate as MC may be software pre-configured (e.g. radio buoys). For uplink, MACA-C operates in few modes as follows:

- RTR-DATA-ACK: The intended node responds with DATA in control channel modulation if it is meant for the MC and uses power control information inferred from RTR's received power, assuming bi-directional validity of power information. MC then closes with ACK. Multiple ACKs may be used to increase receive probability. ACK may include earliest next RTR timing and help reduce uncertainty.

- RTR-RTS-CTS-DATA-ACK: If the destination is another node (not the MC), or if a node wishes to use another FEC scheme, a node sends out RTS once RTR is received. That is followed by CTS-DATA-ACK just as in MACA-EA. This mode has some similarities to a previously published protocol (Kebkal et al., 2005). CTS indicates FEC and power control information as described earlier.

- RTS-CTS-DATA-ACK: For downlink, MC starts with RTS and uses this sequence just as in MACA-EA.

5.2.5 Level-2 distributed mode with no handshaking: DATA-ACK

A DATA-ACK reliable data transfer mode without RTS/CTS exchange is also necessary for the distributed mode. As will be discussed in Section 5.4, for low traffic intensity, RTS/CTS handshaking is not necessary and increases waiting time. The DATA is sent using the same back-off process as RTS in MACA-EA as described earlier. The receiver sends back an ACK when it receives a DATA packet.

5.2.6 Adaptive multi-mode MAC

In MAC-AMM, a modem assesses its neighbourhood and traffic intensity and switches to an appropriate MAC mode from the above choices. When modems do not sense dissimilar modems, they could use any physical layer and MAC protocols. This allows the usage of proprietary technologies and protocols in isolated environments. For modems that have only Level-1 compliance, if they detect alien signals (hear a standardized preamble followed by indecipherable packet or based on energy detection) they should automatically switch to Level-1 DATA-ACK protocol that uses random back-off.

Level-2 adaptation is possible in two ways – modems implement only the standardized physical layer or they implement the standardized physical layer alongside any proprietary scheme and have mechanisms to switch between them. For nodes using the compliant physical layer and MAC protocol, there is no change in behaviour required. For nodes using compliant physical layer and non-compliant MAC protocol when in isolation, if they hear packets belonging to the standardized MAC protocol (as identified by the type field), they should switch to the standardized Level-2 MAC protocol for compliance. Nodes with multiple physical layers (one of which is compliant), operating in non-standard physical layer in isolation, need to switch to compliant mode upon detecting alien signals.

In level-2 MAC, there are three modes MACA-EA, MACA-C and DATA-ACK. The nodes determine the presence of an MC through RTR messages. If they do hear RTR, they use MACA-C. If they do not hear RTR messages, they use MACA-EA. In other words, the switching between MACA-C and MACA-EA is decided by the network deployment and is not automatic. The network automatically switches to DATA-ACK mode if the network is operating in very low traffic intensity (sporadic DATA), if it is currently operating in either MACA-C or MACA-EA mode. In MACA-C and MACA-EA modes, batch size B also needs to be adapted based on traffic intensity. Figure 5.2 captures this process for Level-2 compliance.

More details on the behaviour of the protocol modes with respect to traffic intensity are presented in Section 5.4. The modes discussed here are all for reliable communications, i.e., with retries for failed packets. A brief discussion on broadcasts and unreliable transmissions is provided in Section 5.6.

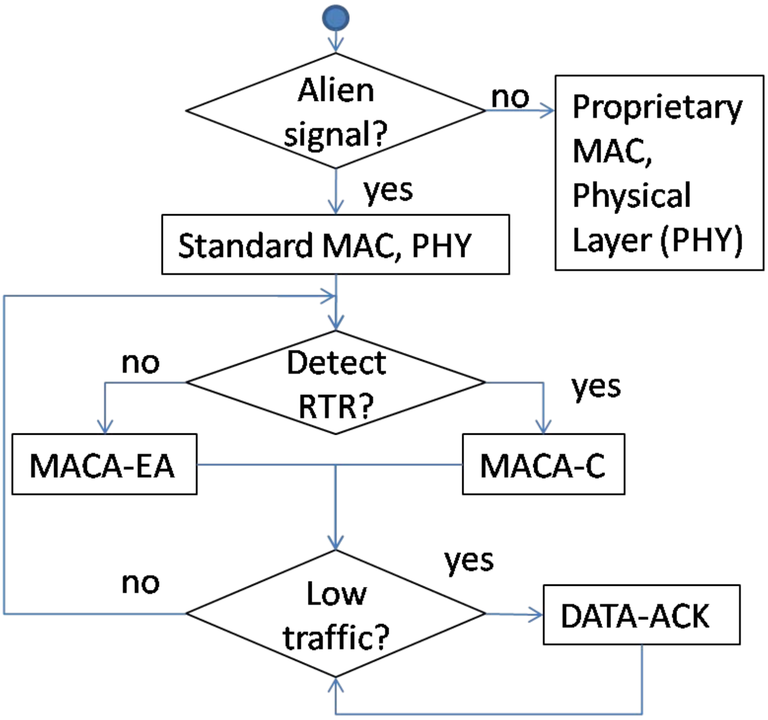


Figure 5.2: MAC-AMM adaptation

5.3 Throughput Analysis of LEVEL-2 MAC

Here the throughput performance of DATA-ACK, MACA-EA and MACA-C for saturated traffic with respect to system and environment parameters (such as batch size B for MACA-EA and MACA-C) is presented. The performance measures were described in the previous chapter. Apart from that, for the DATA-ACK protocol,

$$T = \frac{1/s_p}{(1/L)} \tag{5.1}$$

Simulations and mathematical analysis are undertaken for all the three modes. Although the analysis uses some simplifying assumptions, the analysis and simulation results match reasonably well. Therefore the mathematical analysis can be used for comparative studies in the future. Impact of parameter variations other that those presented in this chapter can also easily be studied using the analytical models.

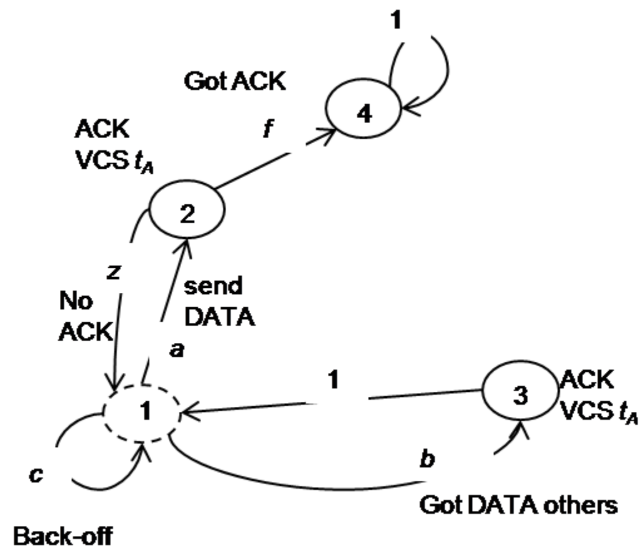


Figure 5.3: Markov Chain for computing Expected Service Time for DATA-ACK protocol

5.3.1 DATA-ACK

The analysis used here follows the same technique as the MACA-EA analysis in Chapter 4. Note that this mode has queueing of incoming data instead of discarding packets if the server is in the process of sending DATA or waiting for an ACK. The standard ALOHA based analysis for such protocols does not model queueing, and uses a model where DATA is transmitted as soon as it arrives and does not model the ACK from the receiver. There is also no retry at the MAC level for lost DATA in such models. The model here is for reliable transmission, and no DATA packet is discarded. The protocol can be represented using the model in Figure 5.3.

The details of this analysis are in Appendix B.1, where the service time is shown to be

$$s_p = \left(\frac{lW' + t_A}{k^2}\right)\left(\frac{W'}{W'-1}\right)^{(N-1)} + \frac{t_A(N-1)}{k} \tag{5.2}$$

As discussed in Chapter 4, $l = L + D$ (4.6), $W' = (W+1)/2$, W is the contention window, and t_A is the timer for ACK (or CTS) reception. The throughput can be computed using (5.1), multiplied by N for network throughput. This analytical result matches simulations well as shown in Figure 5.4. Although there is no batch size in the DATA-ACK protocol, the same throughput is shown across all values of B for ease of representation.

5.3.2 MACA-EA

Mathematical analysis and simulation results for throughput has been shown earlier in Chapter 4 for MACA-EA protocol. The batch service time is (4.13)

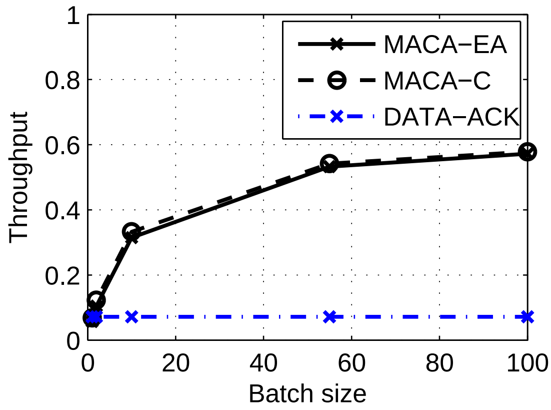
$$s_b = \left(1 + (1-k)^i\right)\left(\frac{l}{k^2 W'}\left(\frac{W'}{W'-1}\right)^N (W'^2 + W' - 2) + \frac{2(N-1)l}{k}\right) + N t_B \quad (5.3)$$

The throughput can be computed using (4.5), multiplied by N for network throughput.

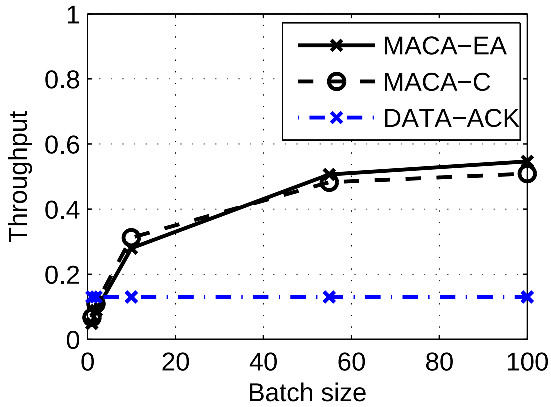
Results are shown in Figure 5.4 and depicts how the performance of MACA-EA improves with packet train length. For high batch sizes, the reliable delivery throughput for the network approaches k_D as shown in Chapter 4.

5.3.3 MACA-C

Here the performance of the distributed data transfer topology RTR-RTS-CTS-DATA-ACK mode of the MACA-C protocol is presented, for a fair comparison with MACA-EA and the DATA-ACK protocols. The MC can be considered as



(a) Simulation



(b) Analysis

Figure 5.4: Network throughput of MACA-EA, MACA-C and DATA-ACK (Packet duration $L = 0.5s, L_D = 1.0s$, detection and decoding probability $k = 0.81, k_D = 0.63$, one-way propagation delay $D = 0.5s$, number of nodes $N = 7$, contention window $W = 17$, Multi-ACK $i = 3$).

acting as the channel access arbitrator for N nodes, which need to communicate to any other node.

The MACA-C protocol can be analyzed as shown in Figure 5.6. The process starts with the MC sending an RTR in state s1. If RTR is received successfully it goes to state s3, else to state s2. In s2, the response timeout t_{RTR} needs to be set such that it allows sufficient time to detect post-RTR packets such as RTS, CTS or DATA to ensure that the RTR was received, and at the same time not too long

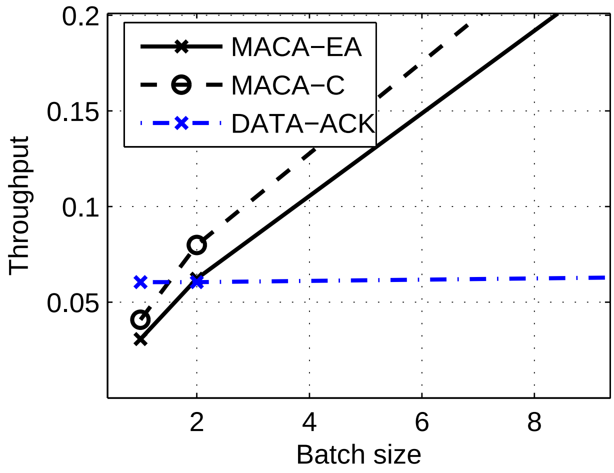


Figure 5.5: Network throughput of MACA-EA, MACA-C and DATA-ACK: behaviour at low batch size, showing that DATA-ACK is better than RTS/CTS protocols with B=1. Parameters: $L = L_D = 0.5s, k = k_D = 0.81, N = 4, D = 0.5s, W = 11, i = 3$.

so as not to waste bandwidth when the RTR is lost. The analysis and simulation used $t_{\mathrm{RTR}} = 2L + BL + 2D$ seconds or $2L + 12L + 2D$ seconds, whichever is lower. This allows for the packets in a batch, the RTS, the CTS and the propagation delay, capped at $B = 12$. If RTR is received, the node will proceed from s3 to an Early-ACK mode in state s4 if the previous ACK was not received or directly to state s5 otherwise. In s4, if the RTS and Early-ACK is received with a combined probability of k^2, it proceeds to state s5. In s5, if RTS is received by recipient and CTS by the sender with a combined probability of k^2, it proceeds to send DATA and wait for Multi-ACK in state s6. This sequence repeats for N nodes. From the Markov chain it can be ascertained that the expected service time to send a batch is

$$s_b = N\left(l + \left(\frac{1}{k} - 1\right)t_{\mathrm{RTR}} + (1-k)^i \frac{1}{k^2}t_A + \frac{1}{k^2}t_A + t_B\right) \qquad (5.4)$$

In (5.4), the first l (slot length $l = L+D$) is for the RTR packet itself. $\left(\frac{1}{k} - 1\right)t_{\mathrm{RTR}}$ is the time to go from state s1 to s3, since expected passes through s2 is (1/k-1)

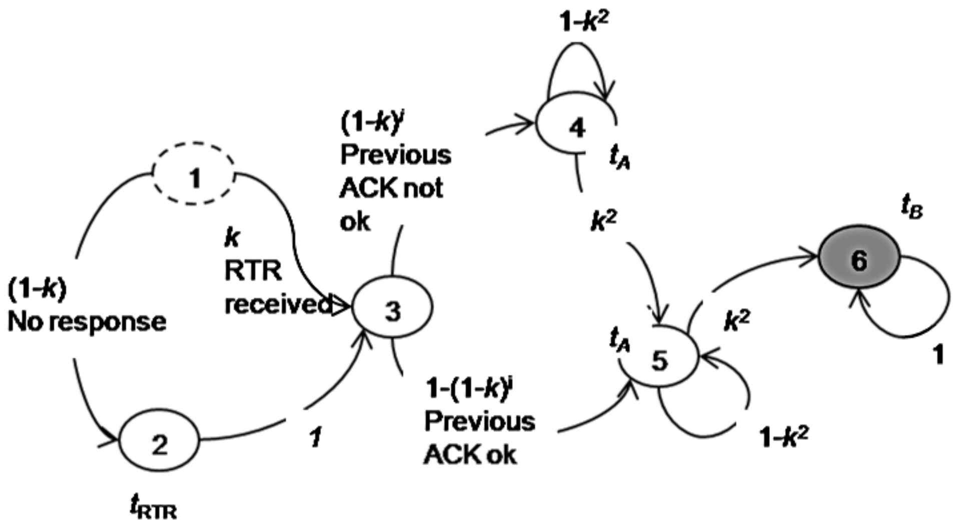


Figure 5.6: Markov chain for the MACA-C protocol

(expected passes through s1 is $1/k$, since probability s1$\rightarrow$s3 is k). $(1-k)^i(1/k)^2t_A$ is the expected time from s3 to s5 (path s3$\rightarrow$s4$\rightarrow$s5 only, since s4$\rightarrow$s5 has no delay), $(1-k)^i(1/k)^2t_A$ is the delay in the RTS/CTS process in s5, and finally the batch transmission delay in s6 is t_B.

Note that the batch service time s_b is defined as the time from the start of the sending process till reception of the ACK. In the model used here, if the ACK after the batch transmission is not received (assumed to be sent by receiver always with probability 1 (as discussed in Section 4.3.2), the ACK may be received as an Early-ACK in the next cycle in state s4 as shown. Even so, the result in (5.4) holds. The throughput can be computed using (4.5), multiplied by N for network throughput.

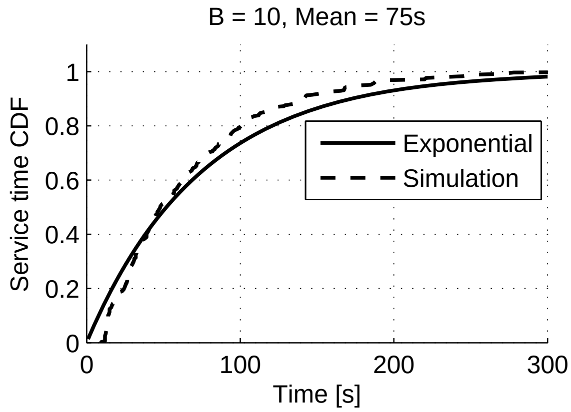
Results are shown in Figure 5.4. It shows that MACA-C performance can be better than MACA-EA. But the difference may not be substantial. The MACA-C protocol eliminates collisions, but there are still overheads in arbitrating through a MAC Controller (MC), and the throughput is restricted. Another cause of the similarity between MACA-C and MACA-EA throughput performance is that for higher batch sizes, the performance is dominated by the batch sending, and the contention period in MACA-EA or the RTR process in MACA-C becomes a less significant factor in the overall service time. Thus, one would expect the greatest percentage difference between MACA-C and MACA-EA at low batch sizes, especially $B = 1$. MACA-C service time is clearly lower for $B = 1$ or similar low batch sizes. However, as elaborated further in the following discussion, there is little utility in using MACA-C or MACA-EA in very low batch sizes, compared to the simpler DATA-ACK mode. It can be noted that DATA-ACK mode is better than RTS/CTS protocols such as MACA-EA or MACA-C when $B = 1$ as shown in Figure 5.5.

5.4 Mode Adaptation Based on Traffic Intensity

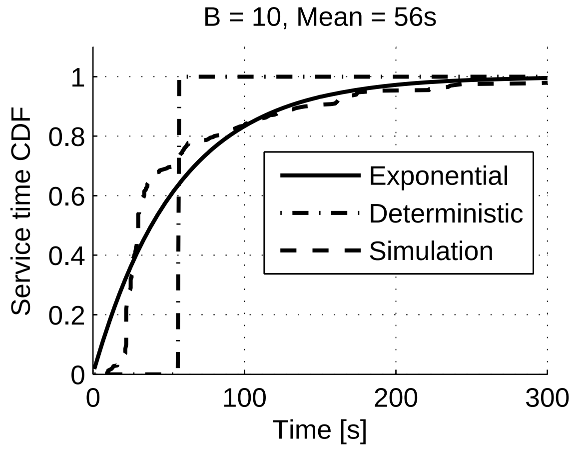
In this section, the unsaturated traffic scenario is considered where packet arrivals are modeled as Poisson distributed. Choice of protocol to use is also related to the traffic intensity $\rho = \lambda/\mu$ where λ is the arrival rate (let $\delta = 1/\lambda$ be the arrival delay) and μ is the average service rate of the MAC protocol. In general, $\lambda < \mu$ for a stable system. Each protocol variant has an upper limit to the service rate and hence a maximum permissible λ. Waiting time in the system is used as the metric for comparing the performance of the different modes. Here a reference TDMA mode is presented as well, to see how it compares with the three modes of MAC-AMM and to provide an upper bound for performance (lowest possible waiting time). As mentioned earlier, apart from mode switching based on traffic intensity, batch size B also needs to be automatically adapted for both MACA-C and MACA-EA. For the results presented in this section, all arrival delays are given per node. The average network delay is a fraction $1/N$ of this.

5.4.1 Service time distribution

In order to do a waiting time analysis, the service time distributions of the modes need to be characterized. The service time distribution of the MACA-EA protocol has been shown to be near exponential in Chapter 4. A plot of the MACA-EA service time distribution for $N = 4$ is shown in Figure 5.7(a) and it shows the analytical exponential fit (the plot in Chapter 4 was for $N = 3$). The service time distribution of the DATA-ACK protocol has been obtained via simulations and is exponential and a sample is shown in Figure 5.8. The service time behaviour of the MACA-C protocol has been obtained through simulations and is plotted in Figure 5.7(b). The exponential fit for the MACA-C service time is poorer. Other typically used distributions do not give a good fit either. In the waiting time analysis that follows, the exponential and deterministic service time distributions are considered, as the actual distribution lies somewhere in between. We can see that while exponential approximation yields conservative results while the deterministic service model offers optimistic results. In the waiting time queueing analysis for the MACA-EA and DATA-ACK modes, a Markov model is used as an approximation and it yields a good match with simulation results. Note that the mean of the DATA-ACK protocol is for the service time to send one packet, while that for the MACA-EA or the MACA-C protocol is for a batch of packets B



(a) MACA-EA



(b) MACA-C

Figure 5.7: Service time distributions of MACA-EA and MACA-C. Parameters: $L = L_D = 0.5s, N = 4, D = 0.5s, i = 3, k = k_D = 0.81$.

as indicated. TDMA-REF service time is nearly deterministic. Normal TDMA is deterministic, but the retry-based reliability process introduces a small degree of non-determinism as discussed later in this section. Batch service time is defined as time till ACK is received and hence retries have to be included in its computation.

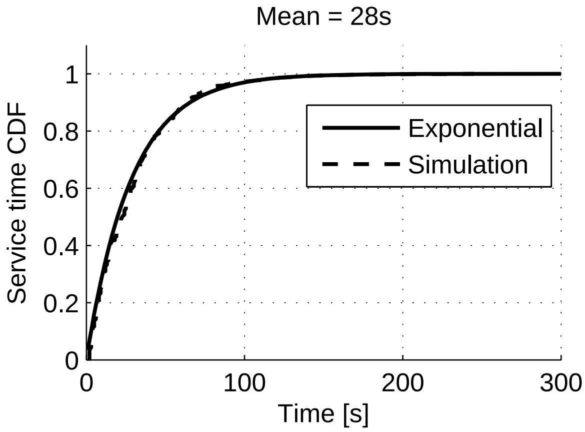


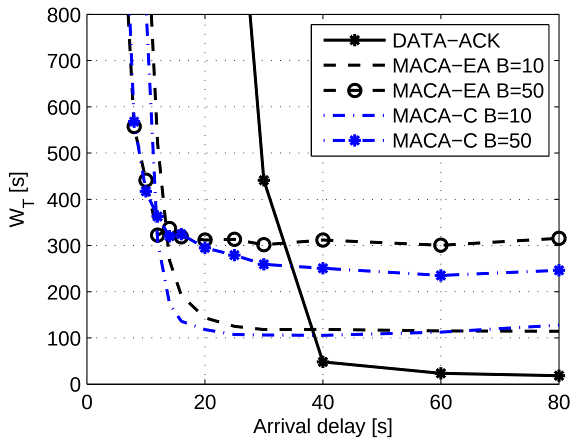
Figure 5.8: Service time distribution of DATA-ACK. Parameters: $L = L_D = 0.5s, N = 4, D = 0.5s, k = k_D = 0.81$.

5.4.2 MACA-EA

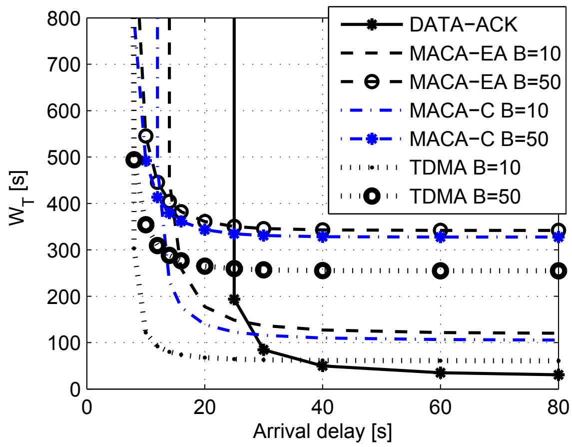
Since the service time distribution is nearly exponential, an M/M^B/1 model is used. The waiting time analysis of the MACA-EA mode uses the analysis model discussed in Chapter 4. Simulation results for waiting time are shown in Figure 5.9(a). The analytical results are shown in Figure 5.9(b).

5.4.3 MACA-C

To analyze MACA-C, the M/M^B/1 and the M/D^B/1 models are used as approximations. The M/M^B/1 analysis is the same as presented in Section 4.5.2 while the M/D^B/1 analysis is shown in Appendix B.2. Simulation results are shown in Figure 5.9(a) and the corresponding analytical results are shown in Figures 5.9(b) and 5.10. The results compare quite well. As noted earlier, the M/M^B/1 analysis gives a higher waiting time estimate than the simulations for larger batch sizes. The M/D^B/1 analysis gives a lower waiting time estimate than the simulations for lower delay regime. Nevertheless the overall trend and the predictions given



(a) Simulation



(b) Analysis (MACA-C exponential service)

Figure 5.9: Waiting time for the different modes. Parameters: $L = 0.5s, L_D = 1.0s, N = 4, D = 0.5s, W = 11, i = 3, k = 0.81, k_D = 0.72$.

by both the models are very useful. It can be seen that the waiting time performance of this implementation of MACA-C is comparable to that of MACA-EA, just as in the case of saturated throughput. The important observation is that for an equivalent distributed topology operation, MACA-C may not offer any significant advantage compared to MACA-EA. However, MACA-C may bring advantage to cases where the uplink mode RTR-DATA-ACK and the downlink mode RTS-CTS-DATA-ACK can offer better performance in a star topology communication

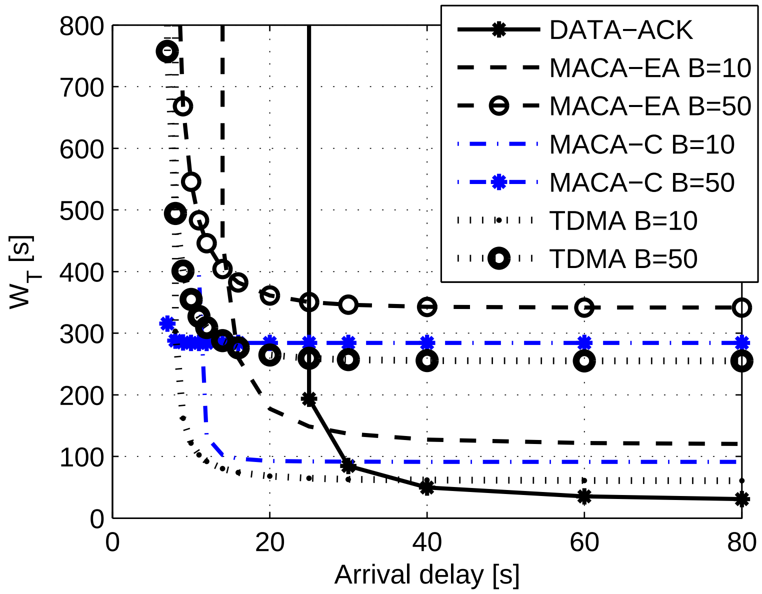


Figure 5.10: Waiting time for the different modes. Analysis (MACA-C deterministic service). Parameters: $L = 0.5s, L_D = 1.0s, N = 4, D = 0.5s, W = 11, i = 3, k = 0.81, k_D = 0.72$.

between clients and MC. The choice of MACA-C over MACA-EA is thus primarily dependent on the traffic pattern demand of the application. MAC-AMM easily allows the switch between the centralized and distributed modes, by introducing an MC in the network neighborhood. All the nodes can automatically switch modes based on RTR detection as discussed earlier in Section 5.2.6.

5.4.4 DATA-ACK

For the DATA-ACK mode, a M/M/1 model is used. Using $\mu = 1/s_p$, the waiting time is (Gross and Harris, 1998),

$$W_T = \frac{1}{\mu - \lambda} \tag{5.5}$$

Simulation results are shown in Figure 5.9(a) and they match reasonably well with the analytical results shown in Figure 5.9(b), and yield reasonably correct relative performance comparison with the other modes. It is noted that the rapid rise below 40 seconds (this is close to the instability region) in the analysis is different from simulations, but the overall performance is reasonably well predicted by analysis. It can be seen that at high arrival delay it can outperform MACA-EA, MACA-C and even TDMA-REF depending on the batch sizes chosen.

5.4.5 TDMA-REF

Here we try to benchmark the three modes of MAC-AMM with a reference TDMA-based protocol termed TDMA-REF . The TDMA-REF protocol has no contention process. Just as in any static TDMA-based protocol model, transmission frames are statically assigned to all nodes in a given sequence. In each frame, the assigned node transmits a batch of packets. In order to have a comparable scheme that allows for retry-based reliability, ACK from the receiver is sent within each frame. In order to enhance performance, the Multi-ACK technique from MACA-EA is used.

Accounting for the two way propagation delay, the service time for N nodes is

$$s_b = \frac{1}{1-(1-k)^i} N(BL_D + D + 3L + D) \tag{5.6}$$

where the factor $1/(1-(1-k)^i)$ accounts for the expected retries when ACK is lost. If the ACK is lost and the whole batch is retried, the packet service time s_p is

$$s_p = 1/(1-(1-k_D)^{(1/k_D)})s_b \tag{5.7}$$

The reason for resending an entire batch when the ACK is lost is that in such a TDMA-based protocol, there is no RTS/CTS contention process and the associated Early-ACK provision to resend the ACK. Hence each node has no option but to retransmit the entire batch in the next frame when an ACK is not received.

Since the whole batch is re-transmitted each time an ACK fails, the receiver can receive a particular packet in any one of the batches. Thus, the factor $1 - (1 - k_D)^{(1/k_D)}$ accounts for this success probability and the consequent retries. The TDMA-REF protocol used here is sophisticated enough to provide a good benchmark.

Analysis for the general service model M/G/1 or M/G^B/1 show that waiting time measures are related primarily to the mean and variance of the service time distribution (Gross and Harris, 1998). Since the variance is zero for the deterministic service model, it provides the best case queueing performance for a given mean. Thus, we can use a M/D^B/1 model for the TDMA-REF protocol to provide as with required benchmark, though, due to the possibility of retries when ACK is lost, the batch service time is not strictly deterministic. The M/D^B/1 analysis is described in Appendix B.2. The results are shown in Figure 5.9(b). To assess how the queueing performance for TDMA-REF will vary with a difference service time distribution model, we can also look at how the Markov model M/M^B/1 compares with the deterministic M/D^B/1 model in Figure 5.11. The analysis method for M/M^B/1 queueing model for TDMA-REF is the same as that presented in Section 4.5.2. The results are plotted in Figure 5.11. It shows that the Markov model produces higher waiting time at arrival delays close to saturation as expected and illustrates how the deterministic case produces the best case for TDMA-REF and gives us a good benchmark.

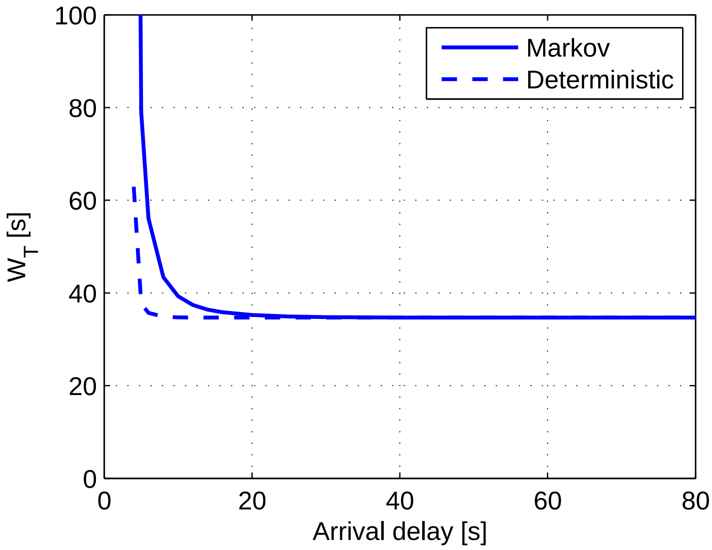


Figure 5.11: Comparison of deterministic and Markov models for TDMA-REF analysis. Parameters: $L = 0.5s, N = 4, D = 0.5s, W = 11, i = 3, k = 0.81$.

And we can see from Figure 5.9(b) that TDMA-REF can be better for the same batch size as expected. Using the analysis models presented here we can assess the performance cost in not using TDMA-REF. However, the choice of TDMA-REF vs. other modes is also dependent on the other requirements of ad hoc functionality, scalability and robustness (especially to time synchronization). The analytical models presented above can help compute the waiting time performance of any mode in MAC-AMM easily without requiring simulations, and help determine the best choice of protocol for a certain application based on its priorities.

5.4.6 Effect of traffic intensity

There are two key points related to traffic intensity. The first is that a given mode has a maximum service rate towards which it will have rapidly increasing waiting time. The second is that each protocol has a lower bound for waiting time as δ

increases, at a fixed set of environment and protocol parameters.

There is another perspective to the results. We can define the QOS (Quality of Service) requirement as a maximum allowable waiting time for a given arrival delay. It is possible that different modes can satisfy the QOS requirement. To see what this means, if we say that for an arrival delay of 40 seconds, we require the waiting time to be no more than 100 seconds for reliable transfer, Figure 5.9(b) shows that DATA-ACK, TDMA-REF, MACA-EA and MACA-C all can satisfy it with a suitable batch size. When arrival delays become lower, the lower performing protocols will no longer be able to satisfy this combined QOS requirement. For example, at arrival delay 20 seconds with a 100 seconds maximum waiting time, the DATA-ACK mode can no longer support it (for this particular set of parameters). This perspective is very important in assessing the relative merits.

Except for high arrival delay, TDMA-REF can provide the lowest waiting time for a given batch size as seen. It can also support the lowest arrival delay (or the highest arrival rate). However, it should be noted that in TDMA-REF, the batch size B has to be fixed at deployment and it needs to be tuned to a certain minimum arrival delay. Higher B will give better throughput efficiency, and support a lower delay, but also suffer increased waiting time. However, in TDMA-REF, it is not easily possible to adaptively tune the batch size, especially if required frequently. Both MACA-C and MACA-EA do not have this drawback, and can vary the batch size at every transmission as required. Dynamic node joining and departures (ad hoc capability) also cannot easily be supported in TDMA-REF. It is also critically dependent on time synchronization, which may not always be available in a general UAN scenario.

MACA-C has both ad hoc capability and the ability to adaptively vary B to suit arrival delays. Since it can support additional uplink and downlink modes also, it may be a good choice in some UAN scenarios. However, as discussed in Appendix B.3, MACA-C will have to contend with scalability issues and interference between neighbouring "cells". If it is a small network in isolation, such issues may not be present and MACA-C will be a good choice. Without elaborating on the scenarios where such protocols are useful, the purpose of the analysis in this chapter is to understand their performance bounds and most importantly the transition traffic intensity to change to and from the DATA-ACK protocol mode.

Comparing MACA-EA and DATA-ACK, the latter is best for high arrival delay region as can be seen. Beyond a certain arrival delay, DATA-ACK will have a lower waiting time. But at lower arrival delay, DATA-ACK cannot support the traffic intensity and only MACA-EA can provide stable service and a reasonable waiting time. It should be noted that the possibility of RTS/CTS based protocols and a random access protocol being dynamically adapted based on traffic intensity had been previously mentioned (Peng and Cui, 2006). But the paper does not contain a detailed simulation and mathematical analysis as we have presented here, using metrics such as normalized throughput or waiting time for reliable transfer of DATA.

This cross over arrival delay ($\delta = 1/\lambda$) can be judged as follows. It has been shown in Chapter 4 that for $1/\lambda > (k_D(\alpha+\beta)^2 + \beta(\alpha+\beta))/\beta$, $B = 1$ is the optimum batch size, where α and β are as defined in Chapter 4. And since at $B = 1$, DATA-ACK protocol performs better than MACA-EA, we can use $1/\lambda$ the cross-over arrival delay. In other words for

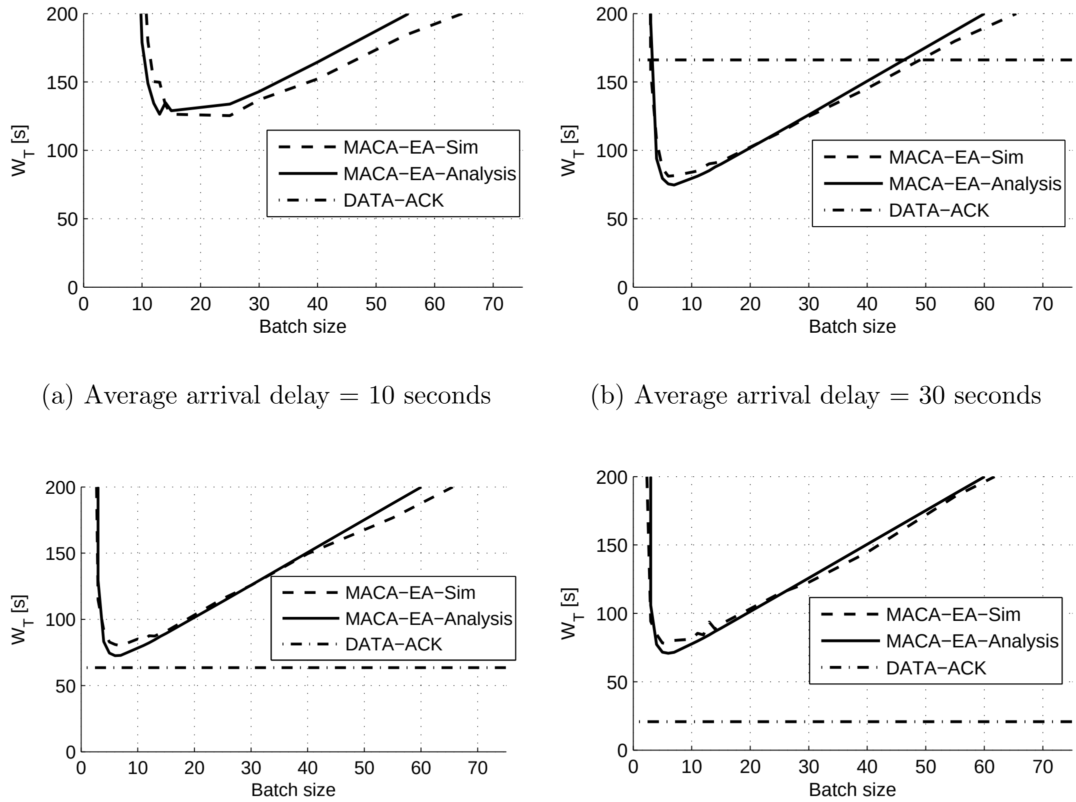
$$\delta > \frac{k_D(\alpha+\beta)^2 + \beta(\alpha+\beta)}{\beta} \tag{5.8}$$

the protocol suite must switch to the DATA-ACK mode. A key factor is batch size B adaptation with respect to δ when MACA-EA is in operation. Optimum batch size has been estimated for MACA-EA in Chapter 4. Higher batch size is suited for lower δ to ensure stability and for minimizing waiting time. With such a combined batch size and protocol mode adaptation mechanism, this suite can deliver optimum performance at any traffic intensity for an ad hoc UAN.

5.4.7 Adaptation algorithm

Here we present an algorithm that can dynamically adapt the batch size of the MACA-EA protocol and enable the switch to DATA-ACK mode based on traffic intensity. The same algorithm can easily be adapted to switch between MACA-C and DATA-ACK as well.

Figure 5.12 shows how DATA-ACK compares with MACA-EA in terms of waiting time for a range of batch sizes. It shows how between 30 seconds and 40 seconds arrival delay, the DATA-ACK protocol gets better than MACA-EA in terms of waiting time, for the set of parameters used here ($N = 4, L = L_D = 0.5, D = 0.5, k = k_D = 0.81, i = 3$). Figure 5.12(c) shows the approximate delay where the two match. The match between simulations and analysis for MACA-EA waiting time is also confirmed in Figure 5.12, similar to that presented in Chapter 4. Figure 5.13 shows how DATA-ACK performance compares with $B = 5$



(a) Average arrival delay = 10 seconds

(b) Average arrival delay = 30 seconds

(c) Average arrival delay = 35 seconds

(d) Average arrival delay = 40 seconds

Figure 5.12: Varying batch size B, DATA-ACK (simulated) and MACA-EA at a given arrival delay. Parameters: $L = L_D = 0.5s, N = 4, D = 0.5s, W = 11, i = 3, k = k_D = 0.81$.

and $B = 2$ for MACA-EA. It also shows roughly 30 seconds arrival delay as the transition point. Note that the small difference between the DATA-ACK modes in Figures 5.12 and 5.13 is due to the former being obtained from the mathematical model, and the latter from simulations.

The adaptation mechanism is as follows. *Use the average batch size occupancy to decide batch size. Average the batch size occupancy over multiple batch transmissions (e.g. 3 transmissions). If the batch occupancy is greater than* 80% *of B, increase B by S. The step S is set to 10% of the batch size. If the batch occupancy falls below 50% of B, then decrease batch size by S. Simulations showed*

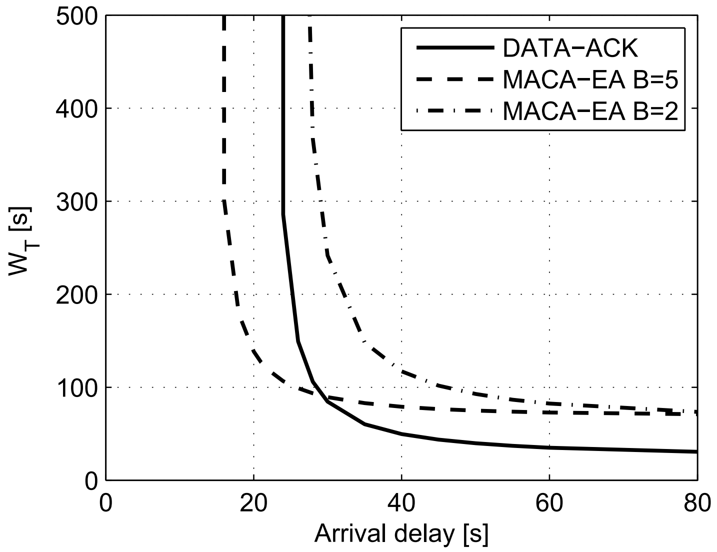


Figure 5.13: Comparing $B = 5$ and $B = 2$ MACA-EA with DATA-ACK (all analytical) at different arrival delays. Parameters: $L = L_D = 0.5s, N = 4, D = 0.5s, W = 11, i = 3, k = k_D = 0.81$.

that this helps adapt the MACA-EA protocol batch size to its optimum value as discussed earlier. If the adapted optimum B in MACA-EA drops to less than $B = B_{\min}$*, (*$B_{\min} = 5$*) then switch to DATA-ACK mode. In DATA-ACK mode, monitor the queue size, if queue size exceeds a certain threshold, switch back to MACA-EA with* $B = B_{\min}$.

Simulation of the MACA-EA adaptation described above has been done. For an arrival delay of 10s, it gave an adapted batch size of $B \approx 15$ with $W_T = 148$ seconds and for an arrival delay of 35s it gave an adapted batch size of $B = 5$ (which is $B_{\min}$, to switch over to DATA-ACK mode) with $W_T = 85$ seconds. In both cases, the simulations were started off with $B = 50$. The values are in reasonable agreement with the optimum batch size seen in Figure 5.12(a) and Figure 5.12(c) respectively (the approximate batch size at the lowest point in the curves). The switch over from MACA-EA to DATA-ACK should be done once adapted batch size reaches $B_{\min}$ as mentioned above. Some of the parameters of the above algorithm are currently heuristics and require further analytical investigation.

If data arrives in a burst in distributed mode, MACA-EA should be used. If the system is in DATA-ACK mode at the time, it switches to MACA-EA mode. If operating in centralized mode, the protocol can directly take care of burst data. The DATA-ACK mode is suitable for situations with low sporadic load. The case of low load could arise in many situations such as underwater sensor network deployments which need to report sporadic information. The DATA-ACK mode gives the least possible latency in this case and makes information and consequent potential actions more timely at the receiver.

The adaptation mechanism is a decentralized decision process but the MAC operations for all nodes proceed without disruption. The operation of the MACA-EA protocol at a given node is quite robust to the choice of batch sizes used by neighbouring nodes. The batch size is indicated in the RTS and hence regardless of the optimality of its choice, all neighbouring nodes take appropriate protocol decisions based on the received RTS. The DATA-ACK mode is not any different from the RTS-CTS contention process, since it is only a type flag in the control packet that changes indicating the content. The back-off and other mechanisms for the DATA packet in the DATA-ACK protocol is the same for RTS, and the ACK follows the logic of CTS transmission and hence there is no problem even if there is a mix of RTS-CTS and DATA-ACK going on at the same time in a network. And if the traffic situation is the same in all nodes, after a transient period, all nodes may end up using similar choice of batch size and protocol mode.

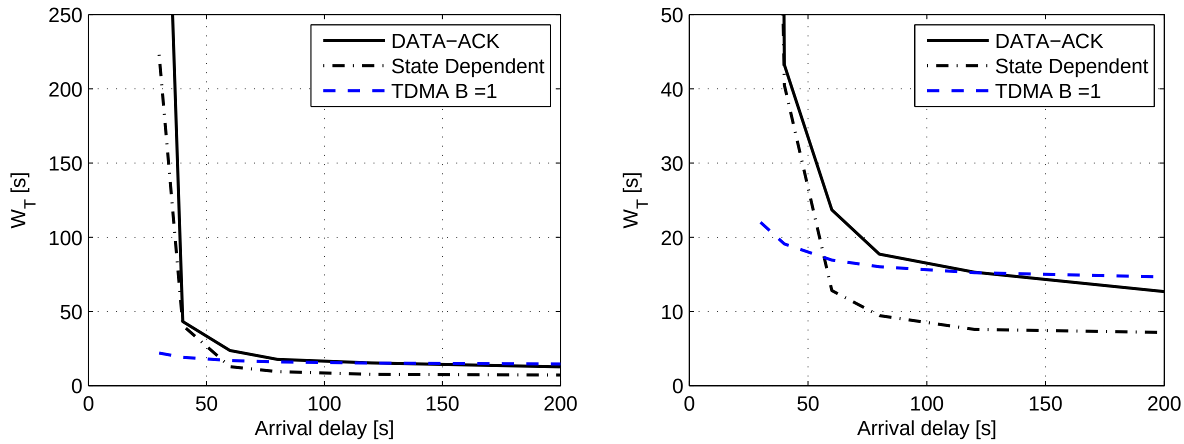
5.5 State dependent DATA-ACK protocol

A further novel enhancement to the DATA-ACK protocol which proves to be effective in low traffic intensity is presented here. In this algorithm, when the queue size is 0, the back-off window is reduced to 1, i.e., the node transmits as soon as a packet arrives. If the queue size is greater than 1, then the same contention process as earlier is used. The results are shown in Figure 5.14 (simulation results for both modes of DATA-ACK and analytical for TDMA-REF). The waiting time is indeed up to 50% lower for very high arrival delays. In fact, it is better than TDMA-REF at batch size B=1 (the lowest possible) at high arrival delays for reliable transfer. Thus, for low rates, the protocol suite will switch to the state dependent DATA-ACK mode. It is important to note that ALOHA-style non state dependent "transmit immediately upon arrival" model is not feasible for a queued scheme with retry based reliability. Thus, this result is not to be treated as the same as an ALOHA scheme.

5.6 Discussion

Here we discuss a few important points on MAC-AMM.

Unreliable messaging and broadcasts can be easily acheived in MAC-AMM. In the Level-1 MAC, the random back-off method applies to all packets equally, including unreliable messages and broadcasts. In Level-2 communications mode,



(a) (b) Magnified view

Figure 5.14: State dependent variation for the DATA-ACK protocol mode. Parameters: $L = 0.5s, N = 4, D = 0.5s, W = 11, k = 0.81$. DATA-ACK modes simulated, TDMA-REF analytical.

if unreliable short messaging (no ACKs) is required, the same contention logic as RTS can be used to send single short DATA packets using control packet FEC. This is essentially the DATA-ACK protocol without the ACK. In MACA-C uplink, RTR-DATA format can be used for unreliable short messaging (no ACKs). Broadcasts are done via these unreliable modes. Unreliable broadcast mode can be used for beacons such as those proposed in JANUS. JANUS beacon and similar concepts attempt to allow nodes to broadcast useful information about itself to neighbouring nodes. Such broadcast packets need to come under the control of a MAC protocol to avoid interference in a given acoustic frequency band. It's easy to accomplish such a beacon in the proposal as mentioned above using the general purpose broadcasts.

It is important to take note of possible inter-MC interference in MACA-C. If there are multiple cells in the neighbourhood controlled by different MCs, there can be inter-cell and inter-MC interference. Mechanisms will be required to address this. We do not address this issue in this thesis but provide a brief discussion on possible mechanisms in Appendix B.3. This is a general challenge for the centralized topology as discussed in Chapter 3.

A discussion on FEC and power control as well as the use of long DATA packets instead of batches of short packets was provided in the previous chapter. Apart from FEC codes, other physical layer parameters also may require tuning (Shankar et al., 2010) and RTS/CTS exchange can also facilitate that.

The work here is primarily aimed at single-hop networks. Multi-hop extensions should be done as part of future work. A brief discussion on possible methodology for multi-hop analysis was outlined in Section 4.7.7. As mentioned in Section 4.2.2, in a multi-hop scenario, N can be viewed as the number of neighboring nodes that each node effectively contends with. As mentioned in Section 5.4.6, when using MACA-C in multi-hop networks, inter-MC interference may be experienced, if there are multiple cells in the neighborhood controlled by different MCs. Mechanisms will be required to address this and it is briefly discussed in Appendix B.3. This is a general challenge for the centralized topology as discussed in Chapter 3.

5.7 Conclusion

This chapter presents a comprehensive MAC protocol suite to address a diverse and heterogeneous underwater network with multiple levels of compliance which also allows for proprietary protocols to be used in isolation. The MAC protocol in Level-2 operations (communications among heterogeneous assets) of the suite has both distributed and centralized operating modes. A novel adaptation scheme chooses between the modes based on deployment, environment and system parameters. Analytical throughput performance results were shown which has been verified through simulations. Traffic intensity is shown to be a key parameter, especially to determine the transition to the low traffic DATA-ACK mode. An algorithm to dynamically adapt the batch size in MACA-C and MACA-EA, as well as to enable the automatic switch to the DATA-ACK mode was presented. A novel state dependent variation of the DATA-ACK protocol was also presented. The MACA-SEA protocol needs to be analytically characterized and validated through sea-trials. The key vision is a self-organizing network, with nodes able to dynamically adapt to any scenario through environment discovery. A key utility of this work is the accurate analytical characterization of the relative performance of the different modes in this protocol suite. Mission planners planning to use such networks can also understand the performance boundaries and ensure that communication requirements are within feasible limits.

Chapter 6

Twin-MAC Protocols

In the previous chapters, the focus has been on the design and analysis of suitable ad hoc protocols for UANs where time synchronization is not necessarily available. As the results showed in Chapter 5, TDMA performs the best compared to MACA-EA, MACA-C etc, except at very low packet arrival rates compared to the state dependent DATA-ACK protocol. At the same time, also as noted before, TDMA has many limitations in scalable ad hoc UANs. Nevertheless, TDMA remains a benchmark for comparison of time domain protocols and an investigation has been made recently on TDMA performance in high propagation delay UANs at a fundamental level (Chitre et al., 2011). This chapter discusses the fascinating possibility of utilizing the propagation delay in networks more constructively than traditionally done, that evolved out of this research. This work lays important foundations for further research into how random access networks may potentially benefit from propagation delay. Section 6.1 provides an illustrative overview of a new concept that can be termed “Super TDMA”. Section 6.2 shows how to utilize this concept in a few novel MAC protocols. The content in Section 6.1 is part of a paper accepted for publication (Chitre et al., 2011). The material in Section 6.2 has also been accepted for publication (Shahabudeen et al., 2011b).

6.1 Introduction

First the “Super TDMA” concept is presented, a form of TDMA that can utilize propagation delay unlike traditionally done.

6.1.1 Throughput greater than 1?

The key performance metric used in the thesis has been throughput. Equation 4.5 defined throughput. In the ideal case, probability of detection and decoding k is 1. To transmit B packets, the batch service time $s_b = BL$ in the ideal case with no delay. Therefore,

$$T = \frac{kB/s_b}{(1/L)} = \frac{1\ B/BL}{(1/L)} = 1 \tag{6.1}$$

Thus, the definition of throughput as used in the thesis earlier and in the general literature on UANs is such that it can at maximum be 1. As another example, the standard simplified equation (considering only propagation delay and discounting effects such as clock drift) for TDMA throughput could be written as

$$T = \frac{L}{L+D} = \frac{L}{S} \tag{6.2}$$

where L is the TDMA packet duration, D is the one way propagation delay and $S = L+D$ the slot or frame duration. As D increases, the throughput T decreases from the ideal maximum of 1. Commonly used analysis considers propagation

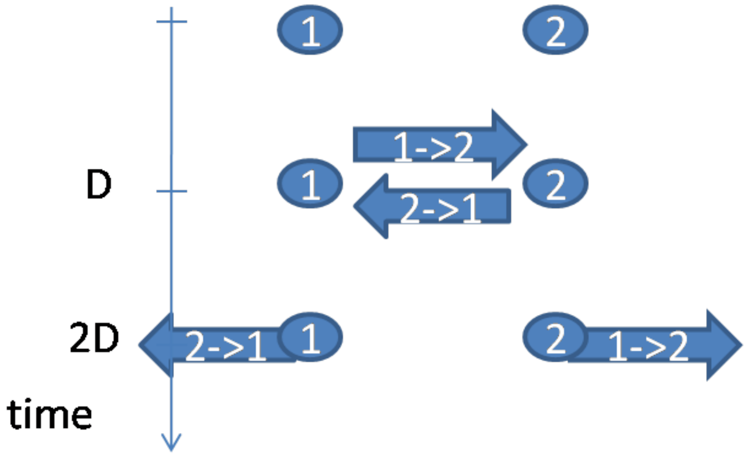


Figure 6.1: Simultaneous transmissions in a two-node network. It shows exchange of packets of duration equal to the propagation delay (i.e., $L = D$) between node 1 and node 2. At time $t = 0$, the transmissions start. At $t = D$, the packets have fully left the transmitters and are just reaching the receivers. At $t = 2D$, the packets receptions are complete and the process repeats.

delay as a loss factor that reduces the throughput to less than 1. Therefore, the question whether throughput can go beyond 1 in the case of networks with propagation delay is rarely asked.

But the question has a very interesting answer as was discovered recently by Chitre et al. (2011)[1]. The throughput can indeed exceed 1. In fact throughput has an upper bound of $N/2$ where N is the number of nodes. The following sections will illustrate this fascinating result with examples and some key concepts.

6.1.2 2-node network

Consider a 2-node network with propagation delay D between the nodes. The idea of allowing nodes to transmit simultaneously and letting their packets "cross in flight" has been previously considered (Hou et al., 1999). When the duration of the

[1]The following material in this section is derived from the contributions of the author to this paper.

packet L is set equal to the propagation delay D, the simultaneous transmissions occur as shown in Figure 6.1. Since in time $2D$, 2 packet exchanges of duration $L = D$ complete, it gives a throughput of 1, the maximum possible throughput for a network with only two nodes. Note that for the schemes discussed in this chapter, the necessary condition is $L \leq D$. The technique of packets crossing in flight is possible due to the well established linearity of the underwater acoustic channel.

6.1.3 4-node regular tetrahedron network

Consider a 3-dimensional network with 4 nodes placed at the vertices of a regular tetrahedron as shown in Figure 6.2(a). If only slotted transmissions are allowed in this network, the transmission schedule can easily be represented in terms of a matrix χ (symbol used by Chitre et al. (2011) is $\mathbf{Q}$) with rows representing the nodes and columns representing the time slots. If the schedule is periodic, the matrix has to only have sufficient columns to represent one period of the schedule. Positive entries in the matrix represent a transmission from the node given by the row to the node number specified in the entry. For example, if the entry at location (3,1) is 2, it means that node 3 transmits to node 2 during time slot 1. Negative entries represent receptions; an entry -1 at location (2,3) denotes that node 2 receives a packet from node 1 during time slot 3. Setting the duration of the time slots in the schedule to be equal to the propagation delay D between any two nodes in this network, it can be ensured that every transmission starting at a slot boundary is received at a slot boundary. Therefore the entire slot duration can

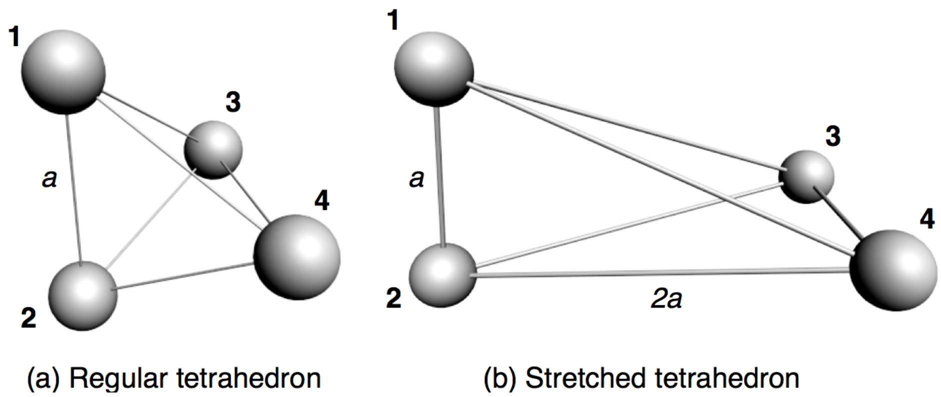


Figure 6.2: Regular tetrahedron and stretched tetrahedron networks

be used for transmission, and therefore set the packet duration $L = D$. By using the following periodic transmission schedule χ for the network, a throughput of 2 can be achieved! Note that the equation assumes steady-state, where all nodes have started receiving packets as per the schedule.

$$\chi = \begin{bmatrix} 2 & -2 \\ 1 & -1 \\ -4 & 4 \\ -3 & 3 \end{bmatrix} \tag{6.3}$$

It can be observed that in this example, nodes 1 and 2 send packets to each other simultaneously in slot 1 and receive them simultaneously in the second slot. Nodes 3 and 4 send packets to each other simultaneously while nodes 1 and 2 are receiving. This causes no collisions since the transmissions arrive at nodes 1 and 2 only in the next time slot when they are transmitting (see Figure 6.3). The same reasoning applies to transmissions from 1 and 2 not resulting in collisions on nodes 3 and 4. If considered separately, pair (1,2) and pair (3,4) behave exactly as discussed in the isolated two node case of Figure 6.1. Over two time slots, all four

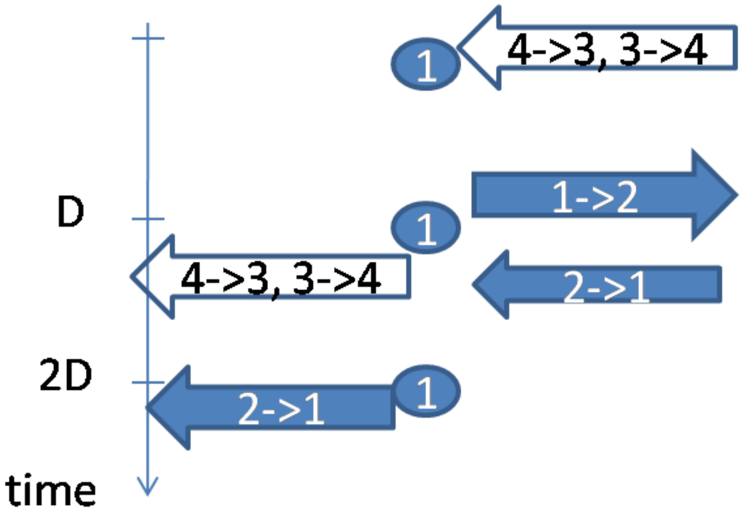


Figure 6.3: Simultaneous transmissions in a regular tetrahedron network. This shows a snapshot of the cyclic process viewed at node 1 in steady state. By symmetry, it is the same for all nodes. At time $t = 0$, interfering receptions arrive at node 1 from node 3 and 4 and node 1 starts transmitting to node 2. By $t = D$, node 1 has finished transmitting to node 2, and the expected packet from its peer node 2 has just arrived. The interfering receptions from node 4 and node 3 are also over. At $t = 2D$, the reception of the valid packet from node 2 is complete and the process repeats.

nodes transmit and receive a packet each successfully; the network throughput is therefore $4/2 = 2$.

6.1.4 4-node stretched tetrahedron network

If the length of four edges of the regular tetrahedron is doubled as shown in Figure 6.2(b), it gives a 4-node "stretched" tetrahedron network. Setting the time slot duration equal to the propagation delay along the unstretched edge, the following transmission schedule χ for the network provides a throughput of 2.

$$\chi = \begin{bmatrix} 2 & -2 \\ 1 & -1 \\ 4 & -4 \\ 3 & -3 \end{bmatrix} \tag{6.4}$$

The explanation is very similar to that of the previous example. The difference is that all nodes transmit in slot 1 simultaneously and all four receive in the next slot (nodes 1 and 2 transmit to each other, and so do nodes 3 and 4). This process can repeat ad infinitum. Here also since there are a total of four successful transmission/receptions in a 2 slot period, the throughput is 2 by definition.

In fact this example gives a fundamental schedule (all nodes transmit in slot 1 simultaneously and all receive in the next slot, Appendix C.1) that can be applied to any network where there are pairs with unit distances between them, and there are even distances between pairs. If such network topology exists, in each slot all nodes will transmit and in the next slot all will receive and all interferences will align such that no reception is hampered (packet interferences at a node occur when it is transmitting). The throughput in that case would be $N/2$ quite obviously! This idea is explored below.

6.1.5 N-node networks

For any even number of nodes $N \geq 4$, it is possible to construct a 2-dimensional network geometry that achieves the $N/2$ upper bound as closely as desired, provided the girth of the network is allowed to grow arbitrarily. To construct such a network, the N nodes are placed pair wise as shown in Figure 6.4. Let delay

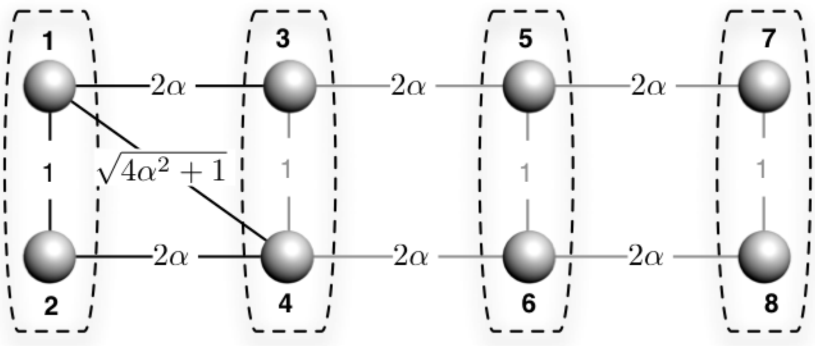


Figure 6.4: A 2-dimensional N-node network for even N

matrix $\mathbf{D}$ represent the inter-node time delays, i.e., $D_{i,j}$ be the propagation delay between nodes i and j (Chitre et al., 2011). Each pair is placed such that the delay between the nodes of the pair is 1, i.e., $D_{j,j+1} = 1$ for odd j, and the delay between adjacent pairs $D_{j,j+2} = 2\alpha \; \forall \; j, \; \alpha \in Z^+$. Thus, the delay $D_{ij} = \alpha \, |i - j|$ is even for odd i and j, or even i and j. If i is odd and j is even,

$$D_{ij} = \sqrt{D_{i+1,j}^2 + 1} \tag{6.5}$$

When the difference between i and j is large, $D_{i+1,j}$ is large and $D_{ij} \approx D_{i+1,j} = \alpha \, |i - j + 1|$ which is even. The largest deviation δ from an even delay occurs when $i - j + 1 = 2$:

$$\delta = \sqrt{4\alpha^2 + 1} - 2\alpha \tag{6.6}$$

As mentioned in Section 6.1.4, the prototype schedule in Appendix C.1 can be used for this network, as it can be viewed as a network that consists of unit distance pairs, and each pair separated from each other by an even distance. But as shown above, all distances between pairs are not integral. The concept of ρ-schedule (Appendix C.2) can be used to handle scheduling in such situations, where $\rho^- = 0$ and $\rho^+ = \delta$ (i.e., the packet size is $1 - \rho^+ = 1 - \delta$).

As mentioned in Section 6.1.4, if it were a "perfect" topology with even distance separated pairs (and unit distance between pairs), the throughput would be $N/2$. Since this network has a packet size of $1-\delta$ in a unit time slot, the throughput quite obviously reduces by that factor to give

$$T = (1-\delta)\,\frac{N}{2} = \frac{N}{2}\left(1-\sqrt{4\alpha^2+1}+2\alpha\right) \tag{6.7}$$

Even for $\alpha = 1$, the throughput $T = N(3-\sqrt{5})/2$, which is 76% of the $N/2$ upper bound. As α increases, the throughput approaches the upper bound. This gives a network geometry that approaches the $N/2$ upper bound asymptotically for any even number of nodes N.

6.1.6 Bounded network geometries

Since the 2-dimensional N-node network in Section 6.1.5 consists of $N/2$ pairs of nodes with distance 2α between each pair, the girth G of the network is given by

$$G = 2\alpha\left(\frac{N}{2}-1\right) \tag{6.8}$$

As α becomes large, this network can achieve a throughput arbitrarily close to the $N/2$ upper bound. However, the girth G also increases without bound with α. It is then natural to ask what throughput can be achieved by a network whose girth is bounded. For a fixed maximum value of girth G, what is the throughput T that can be achieved?

The girth G of a network is defined in terms of the slot length S. For any finite physical space, an arbitrarily high throughput can be achieved by letting the slot length $S \to 0$. Typically the minimum message duration that can be supported by a real system sets a lower limit on slot length and an upper limit on network girth.

Consider the 2-dimensional N-node network from Section 6.1.5 with girth G given by (6.8). Using the ρ-schedule from Section 6.1.5 with this network, a throughput T given by (6.7) is obtained. Writing T in terms of G:

$$T = \frac{N}{2}\left(1 + \frac{G}{N/2-1} - \sqrt{1 + \left(\frac{G}{N/2-1}\right)^2}\right) \tag{6.9}$$

By selecting $\alpha = 1$ and girth as

$$G = 2\left\lfloor \frac{G_{\max}}{2} \right\rfloor \leq G_{\max} \tag{6.10}$$

Substituting in (6.8) gives $N = G + 2$. Substituting N and G in (6.9),

$$T = \left(3 - \sqrt{5}\right)\left(1 + \left\lfloor \frac{G_{\max}}{2} \right\rfloor\right) \tag{6.11}$$

Thus, for a network of maximum girth $G_{\max} \geq 4$, there exists at least one network geometry and schedule that achieves a throughput of

$$T \geq \left(3 - \sqrt{5}\right)\left(1 + \left\lfloor \frac{G_{\max}}{2} \right\rfloor\right) \tag{6.12}$$

where $\lfloor x \rfloor$ is the largest integer not greater than x.

Equation (6.9) directly relates three important parameters in the network — the number of nodes N, the girth of the network G and the throughput T. For different number of nodes N, the relationship between girth G and throughput T is shown in Figure 6.5. For a given number of nodes, increasing girth will allow throughput to be increased towards the $N/2$ bound. For a given girth, a throughput according to (6.12) for an appropriate number of nodes ($N = G+2$ in this geometry) can be achieved. Note that girth is defined in terms of cS, where c is the signal propagation speed (Chitre et al., 2011).

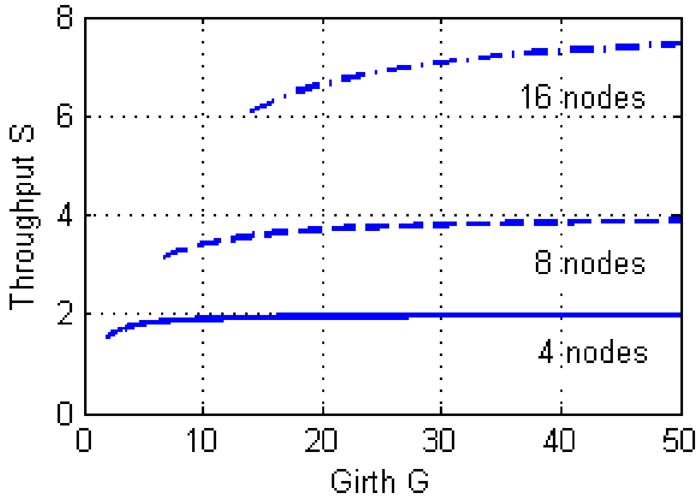


Figure 6.5: Throughput trade-off with delay

6.1.7 Remarks

The topologies used to illustrate the possibility of the $N/2$ throughput are very restricted indeed. Nevertheless this sets a new target of $N/2$ for network performance maximization instead of 1 as typically done[2]. This may not be achievable in most cases, but the awareness of what is possible and the insights into interference alignment may lead to new protocols. Research on random access protocols with such a goal offers a great challenge for the future.

6.2 Pair-wise Transmission Protocols

This section presents a theoretical exploration of some novel MAC protocols for randomly placed nodes with arbitrary link patterns that use the above concept of propagation delay utilization. Conventional MAC protocol design for such net-

[2] For more details on this concept, use (Chitre et al., 2011) and references therein

works focuses on mitigation of the impact of propagation delay (Peyravi, 1999; Peleato and Stojanovic, 2007b; Syed et al., 2007; Guo et al., 2009). In networks with negligible propagation delays, MAC protocols endeavour to avoid simultaneous transmissions in a collision domain. When networks have significant propagation delays, simultaneous transmissions do not cause any harm as long as they do not collide at an intended receiver. As discussed in the previous section, the presence of large propagation delays opens up the possibility of designing transmission schedules that allow much greater network throughput than networks of the same size with no propagation delay. In order to benefit from large propagation delays, simultaneous transmissions must be utilized while avoiding collisions at the receiver. Although this may be easy to do in network topologies with special geometrical constraints, our interest here is to design MAC protocols that are applicable in general networks with minimal constraints. In this chapter, three protocols with utility in large propagation delay underwater networks are presented. The first protocol is a static TDMA based protocol called Twin-TDMA. The second protocol is a dynamic variant that is termed Twin-DTDMA. The third protocol is an ALOHA based protocol called Twin-ALOHA. All three protocols allow pairs of nodes in the network to transmit simultaneously without colliding.

The earlier examples clearly illustrate the possibility of high throughput by using the concept of simultaneous transmissions in pairs of nodes. A larger network is broken down into two node sub-units, and each sub-unit can utilize simultaneous transmissions while avoiding interference with other sub-units. This concept will be referred to as “Twin-TX” in our discussion. Although algorithms to find schedules that harness the high throughput potential of networks with large propagation delay have been developed (Chitre et al., 2011), random access protocols that can benefit from this potential remain elusive. The protocols presented in this chapter take a step towards tapping some of the potential, though they cannot always achieve the optimal throughput for networks with arbitrary geometries.

In order to keep the protocols and analysis in this chapter simple, time synchronization among nodes is assumed and the effect of packet loss due to bit errors is ignored. The only packet loss modelled is due to collisions.

6.2.1 Twin-TDMA

6.2.1.1 Throughput

A key performance metric used in the study of MAC protocols is throughput. A simple expression (considering only propagation delay and discounting effects such as clock drift) for TDMA throughput is given by

$$T = \frac{BL}{BL + D_{\max}} = \frac{BL}{S} \tag{6.13}$$

where L is the TDMA packet duration, B is the number of packets transmitted in a burst (back-to-back in a single slot), $D_{\max}$ is the maximum one way propagation delay in the network and $S = BL + D_{\max}$ is the slot duration. For a network without propagation delay, $D_{\max} = 0$ and $T = 1$. However, as $D_{\max}$ increases, the throughput T decreases.

In a two-node network, the nodes can exchange data packets simultaneously (provided the packet length is equal to the propagation delay) to get the maximum throughput as discussed earlier. In Twin-TDMA, this concept is extended to an arbitrary sized network. In TDMA, in one slot, only one node transmits. In Twin-TDMA, in each slot, a pair of nodes exchanges bursts of data packets simultaneously. Just like traditional TDMA, the slot duration $S = BL + D_{\max}$. There is an additional constraint that $BL \leq D_{\min}$, where $D_{\min}$ is the minimum delay among any pair of nodes. Since there are $2B$ packets transmitted in one slot, Twin-TDMA has a throughput given by

$$T = \frac{2BL}{BL + D_{\max}} \tag{6.14}$$

This exchange of packets in one pair of nodes in the network is illustrated in Figure 6.6. It is assumed that $B = 1$ and that the two nodes shown are separated by a delay $D_i \geq L$ and $D_i \leq D_{\max}$. The sequence illustrates how after time L, the packets have been completely transmitted, after time $L + D_i$ receptions start, and after time $D_i + L$ receptions complete. After time $D_{\max} + L$, all nodes in the network will be clear of the transmissions and the next pair starts the Twin-TX transmissions.

Hou et al. (1999) used such a simultaneous transmission concept, but restricted to only two nodes and not extended to a general N-node TDMA. In another related work called STUMP, a set of scheduling constraints are imposed, and the solution yields a schedule (Kredo et al., 2009). The performance of TDMA, STUMP and Twin-TDMA is compared using an example network presented by Kredo et al. (2009) with 12 nodes and a sink. Since that network is a centralized topology, a set of expressions for the throughput performance of TDMA and Twin-TDMA in a centralized topology (shown in Figure 6.7) is presented. In

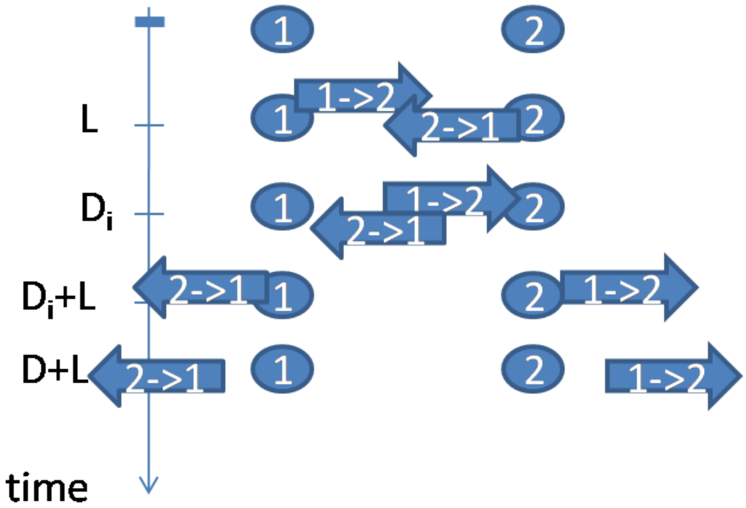


Figure 6.6: Twin-TDMA

this topology, $D_{\min}$ and $D_{\max}$ refer to the minimum and maximum distances from the central node (MC) to any other node. For TDMA, the throughput

$$\begin{aligned} T &= \frac{(B_1 + B_2)L}{B_1 L + D_{\max} + B_2 L + D_{\max}} \\ &= \frac{(B_1 + B_2)L}{(B_1 + B_2)L + 2D_{\max}} \end{aligned} \tag{6.15}$$

where B_1 is the number of packets the client sends to the MC in one slot and B_2 is the number of packets the MC sends to the client in the same slot. For Twin-TDMA, the number of packets from MC to a node and from the node to MC in a single slot are equal, i.e., $B_1 = B_2$. The throughput is given by

$$T = \frac{2B_1 L}{B_1 L + D_{\max}} \tag{6.16}$$

For the example network by Kredo et al. (2009), the round trip propagation delay $2D_{\max} \approx 6$ seconds, the packet duration (called slot duration) $L = 0.4$ seconds and $(B_1 + B_2) = 11$ as 10 packets were used by a client and 1 by the sink. This gives a throughput $T = 0.42$ for TDMA. The STUMP schedule was shown to improve upon this to about $T = 0.56$. A minimum propagation delay $D_{\min} = 2$ sec-

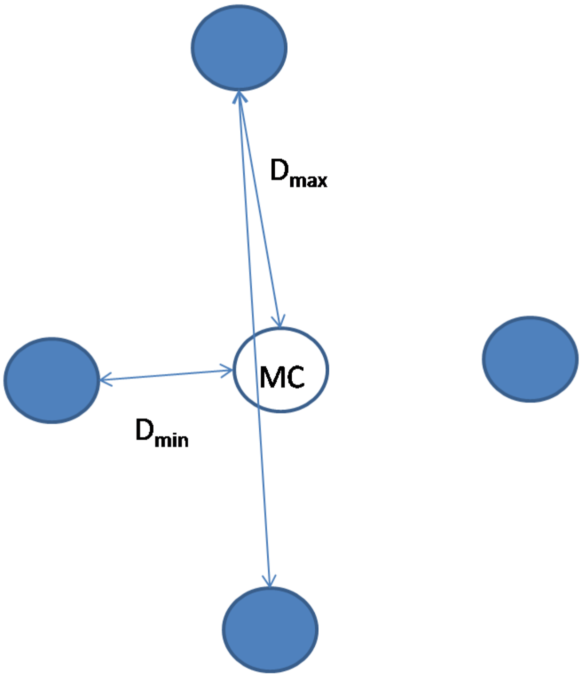


Figure 6.7: Centralized topology

onds is assumed (the minimum separation for the example network is of the order of 3500 m, i.e., 2 seconds). Since Twin-TX requires $B_1 L \leq D_{\min}$, a batch size of $B_1 = B_2 = 5$ is used. The Twin-TDMA throughput therefore is $T = 0.8$, significantly better than that of STUMP in this example.

6.2.1.2 Queuing delay

Although throughput is an important metric, another equally important metric is network propagation delay or equivalently the queuing delay in a single hop network[3]. If the queuing delay was not a concern, choosing arbitrarily large TDMA slot duration (large B) would increase throughput in TDMA, allowing the throughput to be arbitrarily close to 1 irrespective of the $D_{\max}$ of the network. For the example presented by Kredo et al. (2009), one can increase the B_1 and the cor-

[3]In this chapter, the focus is only on single hop networks where all nodes are within a single collision domain.

responding throughput would increase even without the STUMP schedules. For example, setting $B_1 = 49$ would give a throughput $T = 0.77$. So what prevents us from using a high batch size to get high throughput? With a fixed rate Poisson arrival model at each node, larger frames mean greater total queuing delay. Since we do not wish to have arbitrary large queuing delays, we cannot select arbitrarily high batch size and therefore limit the throughput. Clearly, there is a trade-off between throughput and queuing delay.

For an $M/M/1$ system, the total queueing delay W_T is

$$W_T = \frac{1}{\mu - \lambda} \tag{6.17}$$

where μ is the poisson service rate and λ is the arrival rate. $\mu = 1/s$, where s is the service time. From (6.17), we can see that for $\lambda \to 0$, $W_T \to s$. For λ much lesser than the saturation limit (μ), we can use s as a lower bound of W_T for a quick insight. For batch transmission system such as the TDMA system considered here, models such as $M/D^B/1$ (Poisson arrival, deterministic batch service) should be used for accurate analysis. In such systems, the service time s may be used as a lower bound of W_T for low λ.

In a batch transmission deterministic system, as discussed above, assuming no other overheads or losses, the service time s for N nodes is

$$s = N((B_1 + B_2)L + 2D_{\max}) \tag{6.18}$$

Then the waiting time W_T is

$$\min W_T \approx N((B_1 + B_2)L + 2D_{\max}) \tag{6.19}$$

Thus, it can be seen that as $B_1 + B_2$ increases, throughput increases and at the same time waiting time also increases. A middle ground with reasonable $B_1 + B_2$ leads to an acceptable queuing delay and throughput. This is of course the criteria used in some papers implicitly (Kredo et al., 2009). The waiting time for Twin-TDMA W_T is

$$\min W_T \approx s = N(B_1 L + D_{\max}) \tag{6.20}$$

Apart from the improved throughput of Twin-TDMA with respect to STUMP in the previous example, there is also a benefit in terms of waiting time. For STUMP, $\min W_T \approx N(11L + 2D) = 124.8$ seconds. For Twin-TDMA, $\min W_T \approx N(5L + D) = 60$ seconds. For a fairer comparison, the total traffic supported should be identical and 10 slots in total should be used for client and sink in the STUMP example. That gives us $\min W_T \approx N(10L + 2D) = 120$ seconds, which is about about 50% worse than the queuing delay of Twin-TDMA.

It is important to note that the topology in the STUMP example is centralized, i.e., all transmissions occur between clients and a sink. In the Twin-TDMA example, there is a difference that the sink and clients get equal slots of 5 each, whereas the STUMP example has asymmetric 10 slots for clients and 1 for the sink. Although the centralized topology was illustrated for Twin-TDMA for comparison with STUMP, distributed topology pair-wise transmissions are also possible in Twin-TDMA.

6.2.2 Dynamic Twin-TDMA

For ad hoc networks, static TDMA is not suitable. Here, an extension to Twin-TDMA to provide dynamic slot allocation is outlined. In an ad hoc network, nodes

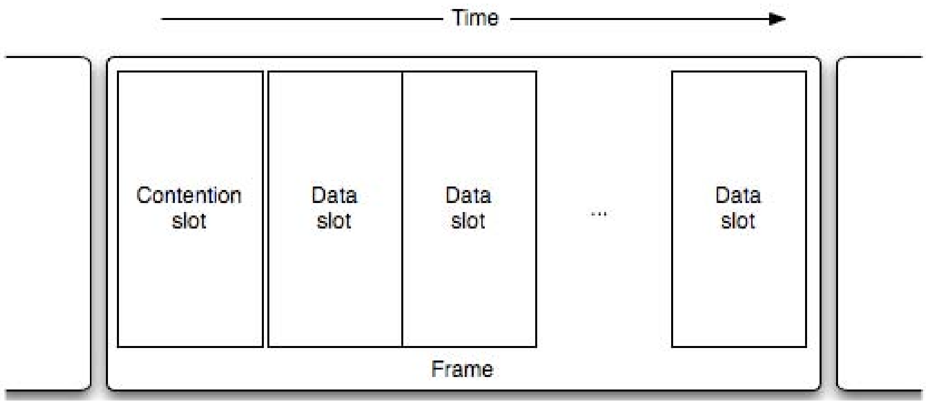


Figure 6.8: Dynamic TDMA

may join and depart from the network and may need different bandwidths. As discussed in Chapter 3, assuming there is time synchronization, a dynamic form of TDMA can be used. The scheme can be represented by the general model shown earlier in Figure 3.2(a) (repeated in Figure 6.8 for convenience), where time is slotted and the slots are assigned as contention or data slots. Nodes contend for data slot allocation during the contention slots.

6.2.2.1 Centralized dynamic Twin-TDMA

The dynamic Twin-TDMA can be used in a centralized mode where the Twin-TX can occur in data exchanges between a client and the sink (termed MAC Controller – MC) or in peer-to-peer mode, between two clients. The centralized topology and data exchanges between clients and the MC are considered in this section. The MC is responsible for dynamic slot allocation. In the contention slot, nodes use random access with a uniform window back-off and send a RTS (Request for Slot) packet to the MC. The MC assigns a slot (or multiple slots) per frame for M frames. After M frames, those slots are no longer reserved and can be re-assigned to another node. The start of the contention slot may be indicated by the MC through a beacon packet. Upon receiving the beacon packet, the clients can start a back-off for sending the RTS. Initially there will be collisions between client RTS packets. But as clients get assigned to their slots, the contention will decrease.

Such centralized topology dynamic TDMA schemes are not by themselves novel and terrestrial radio wireless systems have employed them (DTDMA, URL). But the novelty introduced here is the simultaneous transmission by the MC and the client in the assigned slots, which is only possible because of the large propagation delays in UANs.

6.2.2.2 Performance

Let us first ignore the contention process and take a look at the performance related to the assigned TDMA slots. The analysis is similar to the static case. Nevertheless, another example is presented, with the number of nodes (excluding the MC) $N = 10$, the packet length $L = 0.4$ seconds, the propagation delays $D_{\min} = 2$ seconds (3 km) and $D_{\max} = 4$ seconds (6 km). In dynamic TDMA, 5 slots for uplink and 5 for downlink imply $B_1 + B_2 = 10$. Thus, the throughput $T = 0.33$ and the waiting time $\min W_T \approx 120$ seconds. For dynamic Twin-TDMA, $B_1 = B_2 = 5$ for the same uplink and downlink capacity. Therefore $T = 0.67$ and the waiting time $\min W_T \approx 60$ seconds. In both respects the new protocol performs better than the dynamic TDMA.

When contention is taken into account, the performance depends on the exact contention model used. In a model where the slot allocations are changed infrequently, it is reasonable to expect no more than one node to contend during most contention slots. Since the round trip time for the RTS/CTS exchange during the contention slot is $t_A \leq 2L + 2D_{\max}$, the duration of the contention slot can be set as $C = 2L + 2D_{\max}$. The effective throughput will drop due to the contention process. If there are n_s slots in each frame, the throughput for dynamic Twin-TDMA is

$$\begin{aligned} T &= \frac{2B_1 L n_s}{n_s(B_1 L + D_{max}) + C} \\ &= \frac{2B_1 L n_s}{n_s(B_1 L + D_{\max}) + 2L + 2D_{\max}} \end{aligned} \tag{6.21}$$

Typically, a frame would be designed to have sufficient slots for all nodes in the network to have a chance to transmit, i.e., $n_s \sim N$. The effect of contention on the waiting time is then to increase its expected value over one frame by C. If s', $s' = s + C$ where s is given by (6.20). The waiting time $\min W'_T \approx s'$, i.e.,

$$\min W'_T \approx N(B_1 L + D_{\max}) + 2L + 2D_{\max} \tag{6.22}$$

Taking the same example analyzed earlier in this section, but taking contention into account, for Twin-TDMA we get a slightly reduced throughput $T = 0.58$ and a waiting time $\min W_{T'} \approx 69$ seconds.

6.2.3 Twin-ALOHA

This section looks at how the Twin-TX concept can be utilized in an ALOHA-like protocol. Consider a scenario in which a pair of nodes are always deployed together and require sporadic communication among themselves. During some deployments, these nodes are in a geographical area where they have to co-exist with other network nodes and therefore need a MAC protocol. Due to the sporadic transmission needs, a simple protocol such as ALOHA is perhaps well suited to

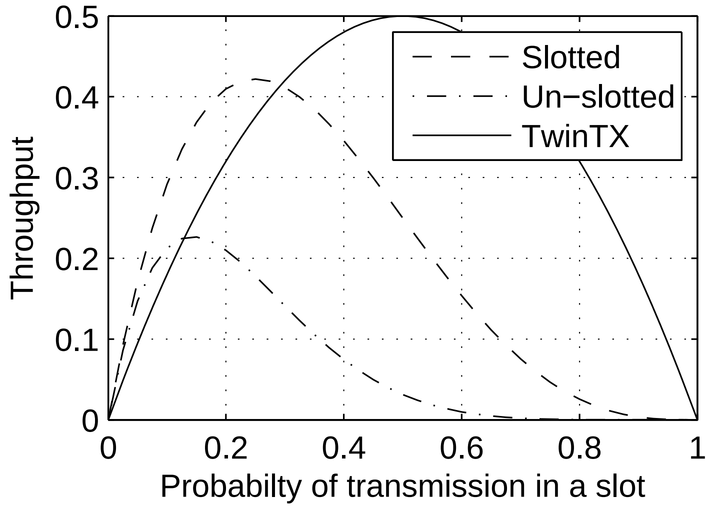


Figure 6.9: Throughput for slotted Aloha, un-slotted Aloha and twin-Aloha in a four node network

the application. However, the Twin-TX concept can enhance the performance of ALOHA in this situation.

ALOHA typically uses randomly chosen transmission times at each node. Instead of one node choosing a transmission time independently, if pairs can simultaneously transmit, then throughput can be improved. Take an example network with 4 nodes arranged as a tetrahedron. In slotted-ALOHA, each node will transmit with a probability $p = 1/N$, N is the number of nodes, for optimal throughput. The maximum throughput is about 0.42 as shown in Figure 6.9. The throughput T can be expressed as follows where p is the transmission probability:

$$T = Np\left((1-p)^{N-1}\right) \tag{6.23}$$

In our example scenario, the data exchange is between pairs and they know about each other. Assuming that the nodes are time synchronized, both nodes in each pair can be started off on the same pseudo-random number generation seed. Based on this they chose the same random slot for transmission. Each pair also knows

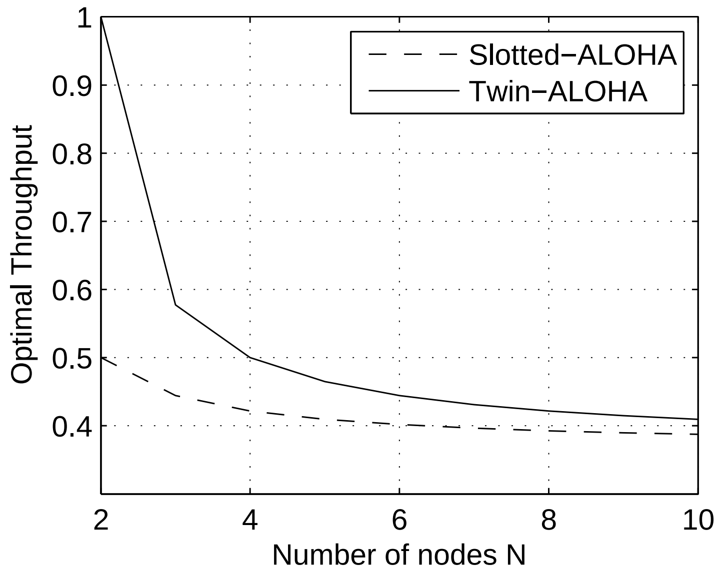


Figure 6.10: Optimal throughput for Aloha and Twin-Aloha

not to transmit in the subsequent reception slot. So there is no *self-collision*. As long as the other pairs do not transmit in the next slot, the transmission will be successful. The throughput is then:

$$T = \frac{N}{2} p \left((1-p)^{N/2-1} \right) \tag{6.24}$$

This is shown in Figure 6.9. Essentially, the performance is that of a network with half the number of nodes. If pairs are not equal in separation and or placed such that transmissions cross slot boundaries, we need to use guard periods. As a first approximation, the effect of such guard periods can be ignored and look at what the optimal performance of such a scheme would be for N nodes as compared to standard slotted ALOHA using (6.23) and (6.24) in Figure 6.10. As seen, for a small number of nodes, there is a clear improvement.

6.3 Conclusion

This chapter discussed the "SuperTDMA" concept and illustrated how throughput is upper bounded by $N/2$ in networks with significant propagation delay, instead of 1 for near zero propagation delay networks such as most terrestrial radio wireless networks. This chapter then presented three Twin-TX (simultaneous pairwise transmission concept) variants of standard protocols (TDMA, dynamic TDMA and ALOHA) that endeavour to harness this high throughput potential of large propagation delay networks. Although the throughput benefits from these protocols is modest as compared to the upper bound of $N/2$ for a N-node network, this work lays important foundations for further research into how random access networks may potentially benefit from propagation delay. Although the $N/2$ upper bound may not be achievable in many network geometries, an awareness of the potential and the insights into schedules that support a high throughput, may lead to the design of novel MAC protocols that are able to benefit from large propagation delays. The protocols presented in this chapter utilize simple pairwise simultaneous transmission, and are only the first steps in this line of research.

Chapter 7

A Multi-channel MAC Protocol for AUV Networks

A key aim throughout this thesis is the enhancement of MACA-based protocols for UANs. As we have seen in the previous chapters, MACA concept can be used as a good basis to develop very effective protocols for UAN MAC for many applications. This chapter describes the work done on multi-channel extensions to a basic MACA-based protocol. It presents a novel attempt to adapt MACA for use over multiple modems optimized for different ranges, primarily aimed at AUV networks. This work has been published (Shahabudeen et al., 2007).

7.1 Introduction

Communications in an underwater network comprising of Autonomous Underwater Vehicles (AUV), fixed sensor nodes and control vessels are commonly implemented using acoustic links. Typical acoustic modems used to establish these links operate at low data rates and ranges up to a few kilometres. At much shorter ranges of tens to hundreds of meters, higher performance communication links can be established using high frequency acoustics. If more than one of the nodes is at the surface, radio links in air also become feasible. As AUVs move around during a collaborative mission, the inter-node separations may vary from few tens of meters to several kilometres. In this chapter the possibility of effectively using such multiple communication channels in an AUV network is explored. The key objective is to maximize throughput and overall data rate per node using multiple channels.

Here the use of multiple communication channels using MACA-based protocols is explored. Freitag et al. (2005) discussed the general idea of utilizing multiple communication channels or bands simultaneously in AUVs. However the MAC protocols for the different channels were not integrated in this work. The modems were meant for different applications and no unification at the MAC level was described. Single transceiver, multi-channel adaptive clustered communications are quite popular in terrestrial networks (Lin and Gerla, 1997). Orthogonal channels using CDMA, FDMA or TDMA are employed to perform adaptive and self-optimized clustered communications.

The term channel is used here to represent very different capability modems, and thus this study is different from the common context of multi-channel communications, where a single transceiver has the option to choose between multiple channels as in FDMA or CDMA. The channels in this study refer to multiple modems or transceivers with varying data rates and range capability that can be used simultaneously. The words modem, transceiver and channel are used interchangeably here.

Here a MACA-based protocol is used on multiple communication channels in an AUV network. An efficient data packet train and position information exchange is used to enhance the protocols performance. End-to-end acknowledged data delivery is measured to characterize true end-to-end throughput. Realistic modem characteristics in terms of Bit Error Rate (BER) and packet detection probabilities, as well as reasonable AUV motion models are used in the simulations. This gives a good indicative measure of the systems performance. A data link layer (DLL) architecture based on queuing is used to interface to multiple physical layers and provide a seamless standard interface to the network layer. The protocol described here shall be referred to as MACA-MCP.

In Section 7.2, we look at how to model the different range modems. The network architecture and the basic algorithm is presented in Section 7.3. The simulation setup, including the modelling of AUV motion, is presented in Section 7.4. In Section 7.5 we present the simulation results and discussion.

7.2 Multi-channel Modelling

7.2.1 BER performance modelling

Medium range acoustic underwater modems typically have an optimal performance range and performance reduces at both shorter and longer distances. For example, an OFDM modem developed at the ARL and tailored for shallow water applications, exhibits a U-shaped BER curve with an optimal range at about 800 meters where the BER is at its lowest (Chitre, 2006). At shorter ranges below 400 meters the performance reduces due to lengthening of the multi-path channel while at longer ranges above 1.2 km the performance reduces due to reducing signal-to-noise ratio (SNR). This will be referred to as the medium range modem or the MR-channel here.

Short range acoustic modems that operate up to a few hundred meters range but at much higher data rates are also possible as part of an AUV communication system. Such modems typically operate at much higher frequencies and larger bandwidths compared to the medium range modems. Such a modem will be referred to as the short range modem or the SR-channel. The SR modem has also been assumed to demonstrate a similar behaviour as the MR modem, i.e. its performance drops slightly at very short ranges.

Similarly, a long range (as compared to the MR modem range) modem is also used, which operates up to about 2km and assumed to be on a lower frequency band orthogonal to MR and SR. A much lower data rate than the MR modem is chosen for this model. This will be referred to as LR channel.

The behaviour of three such modems is captured by the BER curves in Figure 7.1.

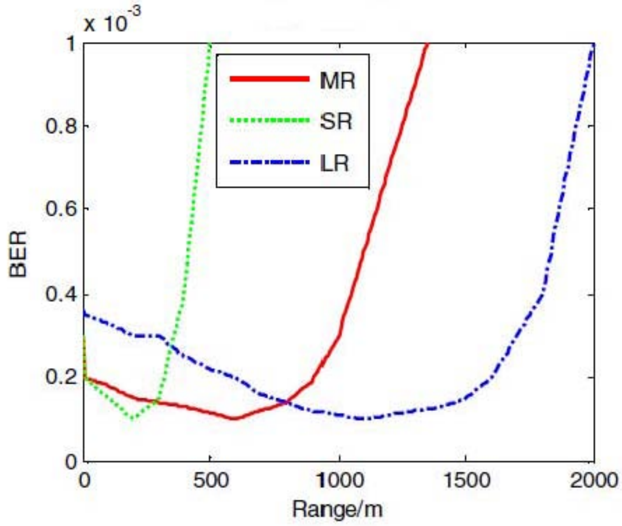


Figure 7.1: Coded BER curves for all channels. The BER is modelled as having a rapid increase after their maximum range in this study.

Detection Preamble	Packet 1	Packet 2	…	Packet N

Figure 7.2: Packet train

7.2.2 The packet train model and packet loss ratio

The data packet train model is shown in Figure 7.2. Each packet train is preceded by a detection preamble used to detect the packet[1]. P_d is the probability of detection of a packet. For simplicity, the time duration of the detection preamble is ignored in the simulations. Each packet has equal probability of success once the train preamble is detected.

In a packet train with B packets and a detection preamble, the Packet Loss Ratio denoted by R_{PL} is as defined in (7.1), where n is the packet size in bits and

[1] Note that the model in the more recent work in Chapter 4 has separate detection preambles for each packet in the train.

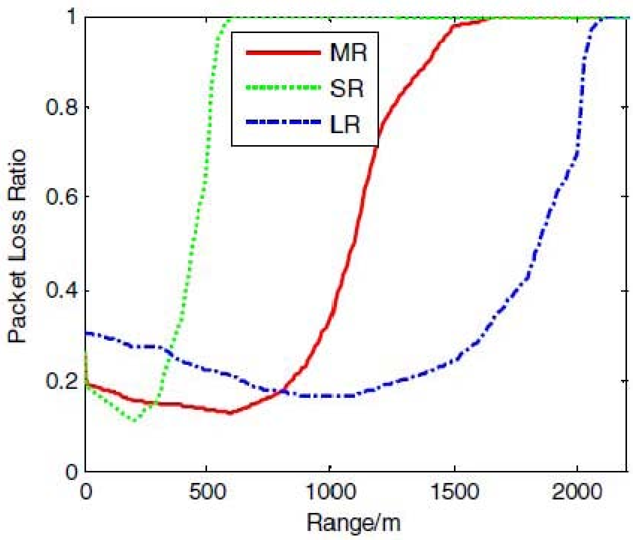


Figure 7.3: PLR curves for all channels for 1000 bits per packet. At the optimum range, the packet loss is about 15%.

E is the bit error rate.

$$R_{\mathrm{PL}} = 1 - P_d(1 - E)^n \tag{7.1}$$

The PLR-curve in Figure 7.3 uses (7.1) and shows the variations in overall packet loss with range. The final values for packet loss shown here are consistent with the order of packet loss ratios that have experimentally observed by ARL in underwater modems at sea. It should be noted that the definitions of SR and MR channels used here in terms of their BER and performance ranges are quite arbitrary. For applications with different overall mission area, multiple modems with different optimal regions can be chosen. The techniques and algorithms used here can then be applied.

7.3 Network Architecture and Algorithms

7.3.1 The physical layer

The half-duplex physical layer used is similar to that described by Shahabudeen and Chitre (2005) and is shown in Figure 8.4. However, no queuing is done to packets coming from the DLL. Queuing at the physical layer makes it harder for the DLL to have good control over the protocol in terms of determining when to send a packet. The physical layer does not accept the DLL packet if transmission is in progress, but accepts and transmits if a reception is in progress, i.e. aborts the reception in favour of the transmission. However, it informs the DLL about ongoing receptions (carrier sense) and the DLL may use that information to avoid transmissions while a reception is in progress.

7.3.2 Network layer and data transmission model

The network layer used in this study is currently not involved in routing, but acts as a data generator at each node. When routing is added to the protocol stack, this layer will implement it and a separate application layer will be used. At each node, data is generated at a rate sufficient to fully load the DLL queuing system and send packets to randomly selected nodes. It is assumed that there is data to be sent to all other nodes at all times, i.e. the system is fully loaded. This assumption is consistent with delay tolerant applications such as file transfers between all nodes.

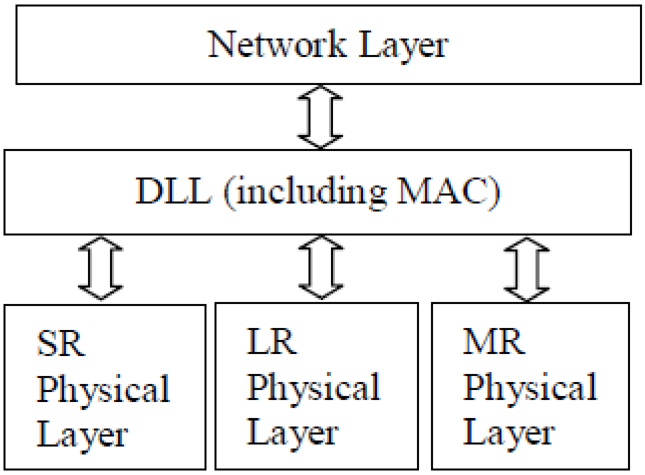


Figure 7.4: Network stack architecture

7.3.3 The overall architecture

A single network layer is connected to three physical layers via a unifying DLL as shown in Figure 7.4.

7.3.4 The basic DLL algorithm

For each of the physical layers (channel), the DLL-MAC maintains a separate state machine. For each channel, the basic MAC algorithm used is MACA-MCP, adapted from MACA and FAMA. It uses RTS/CTS/DATA/ACK exchange with increasing back-off and virtual carrier sense similar to MACA as well as physical carrier sense as used in FAMA. However, it does not follow the restrictions on RTS and CTS time durations imposed by FAMA (Garcia-Luna-Aceves and Fullmer, 1998). Here RTS and CTS packets are shorter than the transmission latencies and round-trip delays. In underwater networks its not practical to put the same requirements on RTS and CTS durations due to the high latencies in transmission (Molins and Stojanovic, 2006). Short RTS and CTS are also needed to minimize collision. In MACA-MCP, the use of data packet trains greatly improves the performance of the hand-shake protocol (examples of this idea has been presented by Molins and Stojanovic (2006) and Garcia-Luna-Aceves and Fullmer (1998), and was discussed earlier in chapters 3 and 4). The RTS contains information on how many data packets are intended for transmission as a single packet train. Once the transmitter receives the CTS, it sends the data packet train. At the receiver, an efficient ARQ method of sending an ACK only at the end of the train indicating all received packets is used and transmitter re-transmits lost data. Increasing back-off timers are used on failed RTS to reduce further collisions. This ACK concept used is different from those proposed by Molins and Stojanovic (2006) and Garcia-Luna-Aceves and Fullmer (1998), where each data packet indicates whether there is subsequent data and an ACK follows every data packet.

Apart from the MACA-based protocol operation as discussed above, position information exchange is used to synergize the multiple channels. Each RTS and CTS contains the position information of the transmitter as well as the latest available position information for its neighbours. In this study, the number of neighbour positions send was set to two. All nodes listen to all packets irrespective of destination and extract the position information contained in all the RTS and CTS packets. Each node maintains a table of position information on all its peers including the time of last update. There is also a timeout set for validity of position information, since nodes are assumed to be mobile. This position information is gathered and shared by the multiple channels.

In the MAC state machine for each channel, when the channel is free based on physical and virtual carrier sense, a recipient node is chosen to send data to, based on whether it is within the particular modem's ideal range. It also uses position information to identify nodes that are certainly out of range and not choose them. If the position information is outdated for all nodes, then the node is chosen in a round robin fashion to ensure fairness. Though the traffic pattern used (mentioned in Section 7.3.2) is a fully loaded model, the algorithm implementation skips a chosen node if there is no data to be sent to it during a given attempt. Thus, after having chosen an appropriate node based on data availability and range, the above MACA-based protocol is initiated. Once the ACK is received, the process repeats. Each of the three state-machines operate independently for the different channels, except for sharing the position information obtained in the process.

7.4 Simulation Setup

All packet lengths are in given in bits. The data rates used are 2400 bps for MR modems, 7200 bps for SR modem and 720 bps for LR modem. Each simulation run is for 4000 seconds and ten runs are averaged for the results in Section 7.5. More details on the simulator can be found in Chapter 8.

7.4.1 Modelling of AUV node motion

A simplified scenario is used for the study a group of eight AUVs on a collaborative mission in a 2km x 2km search area and moving about independently using a

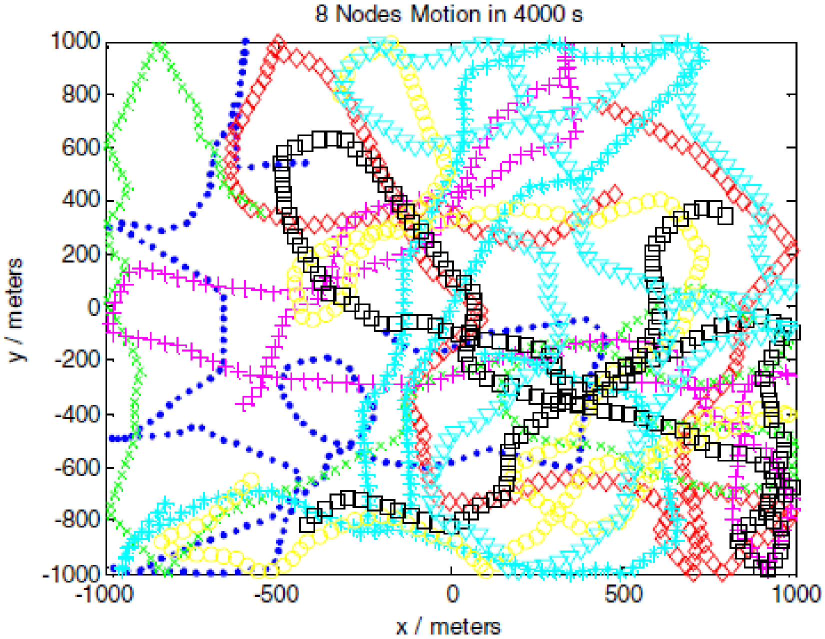


Figure 7.5: AUV nodes motion during one realization of the simulation

variant of the random direction mobility model with reflecting boundaries (Nain et al., 2005) as shown in Figure 7.5. The nodes move in a straight line at a constant velocity between updates. Updates are done at regular time steps (1 second) and during each update, each node has a small probability of random direction change limited to ±45 degrees. Motion is also bounded within the square mission area of 2km x 2 km. At the boundaries, nodes alter course as if it were reflected off. The inter-node distances can vary with time from a few meters to more than a kilometre. A speed of 2 m/s has been used.

Each modem has been tested at their optimum range regimes and with no motion. This was to confirm the correct behaviour and performance of each modem independently when node separations are on average, within their range limits. In their range regime, the modems do work properly using the same protocol. As expected, simulations did show that motion degrades the performance of a single modem when used independently. The degree of degradation depends on the exact motion ranges used and the motion model and is not quantified here. This can be attributed to the fact that in the middle of a RTS/CTS/DATA/ACK exchange, the nodes could go out range and this becomes an additional error contributing factor.

7.4.2 Some factors affecting performance

The factors that impact performance mainly are AUV motion, BER, packet detection probability and collisions. This study uses a static BER and packet detection probability behaviour. Collisions are related to timeouts and retry timers used in the protocol. RTS and CTS packet lengths are quite critical to each channels performance in MACA-MCP protocol. These control collision performance to a great extent. Reducing these control packet sizes, along with better tuning of back-off timers etc are needed to minimize collisions. Reasonable values were chosen for these parameters in the simulations, but they were not fully optimized. Another factor is the data packet size. An optimum packet size needs to be chosen for the best performance. The number of packets in a train is another very important factor. The following studies show the effects of train size. Packet size is fixed at 1000bits.

7.5 Simulation Results

Results from a numerical simulation of MACA-MCP protocol in a small AUV network are presented below.

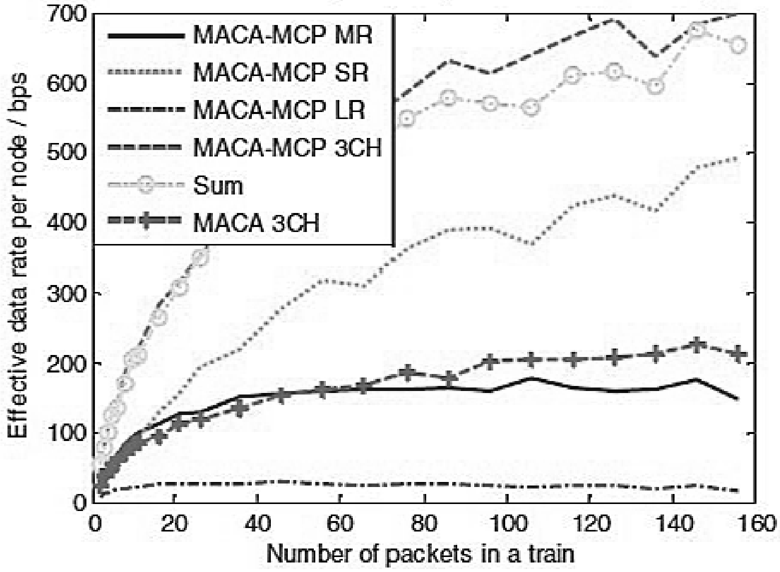


Figure 7.6: Effective data rates for the different protocols as a function of packet train size. Parameters: bits/packet = 1000, baud rates (SR/MR/LR)= 7200/2400/720 bits/s, propagation delay is time varying based on motion model, number of nodes $N = 8$.

7.5.1 MACA-MCP effective data rate performance

Figure 7.6 shows the effective data rate per node for the system in different scenarios. Data rate per node is computed by the traffic generating network layer using (7.2). Average Acknowledged means those data bits for which the RTS/CTS/DATA/ACK was completed.

$$Data\ rate\ per\ node = \frac{Average\ Acknowledged\ Data\ bits}{Simulation\ Time} \tag{7.2}$$

Three scenarios use MACA-MCP protocol on a single channel (labelled MACA-MCP MR, MACA-MCP SR, MACA-MCP LR) and one uses all three channels simultaneously (labelled MACA-MCP 3CH). The Sum is the arithmetic sum of the three channels using MACA-MCP individually. For comparison, we implemented three independent state machines for a basic MACA-based protocol together with the packet train enhancement for each of the channels, i.e, RTS-CTS-

DATA_TRAIN-ACK. There is no position information exchange and the receiver is chosen for sending data in a round robin fashion. MACA 3CH shows the data rate when all channels are simultaneously used using this MACA protocol.

By comparing the MACA 3CH and MACA-MCP 3CH it can be seen that the MACA-MCP is able to increase overall data rate per node by about 300%. This is the key result. Using position information effectively to divide the traffic across the multiple channels helps bring about this significant improvement.

Another interesting result is that the SR modem data rate is much higher than that of the MR or LR modem. Of course, due to the fact that the SR modem has three times the data rate of the MR modem, one expects higher data rate when the modems are operated separately in their own ideal ranges. Here, despite the fact that the test range for the AUVs are four times longer than the maximum range of the SR modem, on average it is about to communicate effectively to immediate neighbours within its range and achieve a high effective bit rate per node. Thus, in a similar scenario if one needed to design a single modem system, an SR modem could be the choice. However, it should be noted that this performance characteristic is a function of the test range, the number of nodes used and the nature of clustering behaviour in a given scenario as discussed in Section 7.5.3 below.

By comparing MACA-MCP 3CH with Sum it can be seen that when all three channels are used simultaneously, overall throughput is better than the sum of the individual throughputs (Sum) of three channels using MACA-MCP individually. The gain is of the order of 10% in the scenario used here. This synergy comes about through the position information gathering and sharing mechanism across the modems. Each channels performance improves by getting cross-channel position information.

7.5.2 MACA-MCP throughput performance

The following discussion looks at the throughput performance of MACA-MCP protocol using one modem only, all three channels simultaneously and the MACA protocol without position information using three channels. This is shown in Figure 7.7. Throughput calculation is done using (7.3) where Data rate per node is as defined in (7.2). Total bit rate used refers to the bit rate of a single modem in the case of protocol using single modem and the combined bit rate of all modems used for multichannel protocol. This is the throughput for the network as a whole.

$$Throughput = \frac{Data\ rate\ per\ node}{Total\ bitrate\ used} \times Number\ of\ nodes \tag{7.3}$$

It can be seen that MACA-MCP has up to about 60% throughput when used with a single channel in MR and SR channels. In the LR channel only mode, the maximum throughput seems to be about 30% only for the simulation scenario that has been used here, in terms of number of nodes and operating range. Data in the single modem mode can only be sent to some peers that are within range, but as the AUVs move around, a node should be able to reach almost any other node over a period of time. This will affect the average transmission latencies.

In the complete 3 channel mode, the MACA-MCP protocol can achieve above 50% efficiency for optimal packet train size. Destinations should usually be available on at least the LR modem and thus there is greater connectivity in the

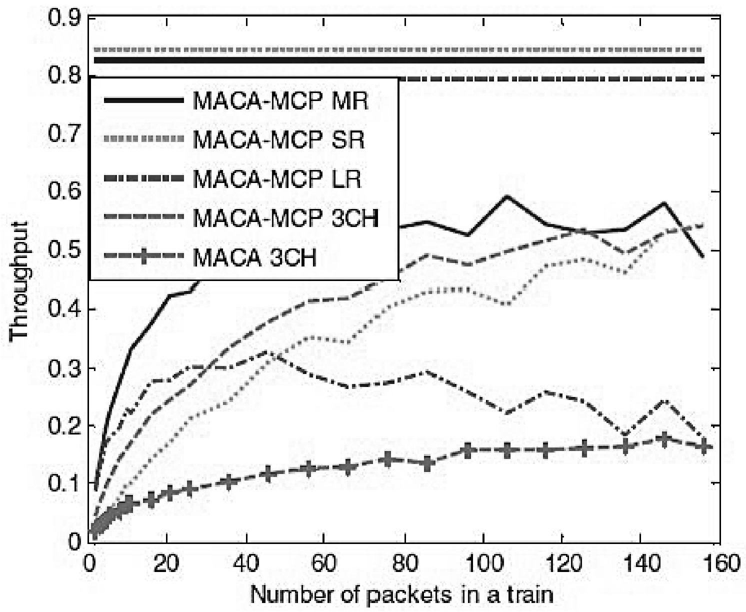


Figure 7.7: Network throughput normalized to individual BAUD. Parameters: bits/packet = 1000, baud rates (SR/MR/LR)= 7200/2400/720 bits/s, propagation delay is time varying based on motion model, number of nodes $N = 8$.

3 channel mode at any given time. MACA without position info over 3 channels has at best about 17% throughput only.

The flat lines on top show averaged theoretical throughput for each modem considering only packet detection probability and BER, in their ideal range. In other words, in a point-to-point one-way transmission, in each of the modems ideal range regime, this is the throughput performance one expects. There are no losses due to collisions and nodes going out of range. There are no losses due to protocol overheads like handshaking, retry timers etc.

Another perspective is obtained on the same result when this is normalized to the theoretical averaged point to point data rate shown as flat lines in Figure 7.7. This normalization excludes the effects of BER and packet detection probability and helps look at the pure network effects on the throughput. These effects now include motion, collisions and protocol overheads like handshaking and timeout

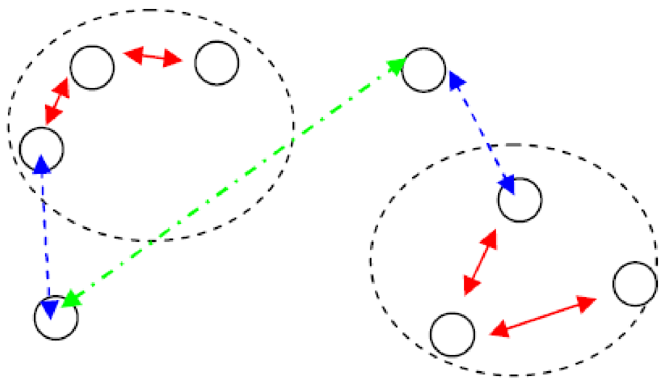


Figure 7.8: Node clustering behaviour

losses. It can be seen that the loss due to the network effects is about 30% and the MACA-MCP protocol is quite efficient.

7.5.3 Adaptive clustering behaviour

The MACA-MCP protocol effectively achieves a node clustering behaviour as shown in Figure 7.8. Dashed lines show possible MR channel connections, solid lines show possible MR connections and dash-dot line shows possible LR connections. As each node preferentially chooses to transmit to nodes on the appropriate channel depending on inter-node distances, in steady state with sufficient position information at each node, such a highly clustered scenario should arise in the network. The key difference as compared to (Lin and Gerla, 1997) is that parallel and simultaneous communication is possible using multiple modems as opposed to using a single channel at any node at a given time. The scheme can also be viewed as having a rudimentary level of inherent automatic power control at each node since the multiple modems have different ranges and modems are chosen based on receiver range.

7.6 Conclusion

This chapter presented a study on using multiple modems optimized for different ranges simultaneously in an AUV network. A unified DLL algorithm has been developed that allows synergistically using multiple physical layers. A variation of the MACA protocol called MACA-MCP has been developed with performance enhancements through packet trains and position information exchange. This achieves a self-organizing clustered network behaviour that leads to good efficiency and data rates per node in the underwater AUV network as shown through simulations.

High speed SR modems should definitely be considered in AUVs to augment the MR or LR modem as they can improve the overall throughput using the MACA-MCP protocol. The basic concept is also scalable in the sense that the definitions of operating range used here for SR, MR and LR modems etc are arbitrary and the system designer can design point-point modems to cater to different range regimes and suit the multi-channel concepts here to very different overall mission ranges. The network layer will need to use novel and efficient routing techniques to exploit this DLL algorithm to achieve full connectivity. This could be looked into as part of future work on AUV networks.

Chapter 8

Unified Simulation and Implementation Software Framework

As mentioned in chapter 1, there is a need for simulations to be cross-verified with sea-trials to ensure validity of the results. This chapter describes the software framework used for the development and testing of the MAC protocols presented in this thesis. It enables seamless simulations and sea-trials. This work has been published (Shahabudeen et al., 2009).

8.1 Introduction

There are many commercial and university based acoustic modems in use around the world, and they use many forms of MAC protocols - both scheduled (such as TDMA) and random access. Improved underwater MAC protocols for varying deployment scenarios are constantly being investigated in many research institutes and commercial entities. The primary mode of investigation is through simulations using network simulators such as NS2 etc, with proprietary modifications to model the underwater channel. Simulations are used since the design/debug/validate cycle is less costly. Some of the promising protocols can then be implemented in modems for sea trials. This typically involves porting of simulation code into the appropriate programming language and software environment in a modem. There are many challenges associated with such a porting process. One of them is to maintain exact algorithmic match in the modem implementation and the simulation code since minor variations in the protocol logic can have significant effects on the protocol behaviour. If coded separately based on simulation code from another environment, there could be significant costs in terms of time and effort to develop it.

Modem hardware could have additional complexities absent in the simulator and the original simulation code might need significant modifications to cater to such differences. This in turn can alter protocol behaviour and performance as seen in simulations and comparison of sea trial results with simulation results could become misleading. To give an example, the ARL acoustic OFDM modem has a High Power Amplifier (HPA) that needs to be turned on before transmissions and turned off for receptions. By design, the control of this feature is delegated to the data link layer (DLL) together with MAC functionality, for optimum use. A standard simulation program for a MAC protocol might not take into account such factors as the HPA control, and during the porting process this could lead to difficulties in maintaining algorithmic integrity.

A software framework was developed to address this situation. It is built upon an earlier proposal for a unified framework called Underwater Network Architecture (UNA) for a structured inter-layer communications (Chitre et al., 2006). In this framework, identical C code for the MAC protocol runs in both the simulator and the modem. The simulator captures the essential behaviour of the modem and uses the same software interfaces as the modem. The MAC code thus developed will run on any UNA compliant modem.

Being able to use the same MAC code in simulator and in the modem is of immense value. Algorithm debugging is possible within the simulation environment itself which can then be run in the modem with virtually no further issues. Since the same code is used without changes, performance comparisons between simulations and actual sea trials become meaningful. The simulator also becomes a stronger and trustable tool, once there is such an identical code linkage with the modem.

Carlson et al. (2004) presented a simulator for AUV communications. A very closely related work, Shusta et al. (2008) looks at unifying interfaces to physical layer etc to facilitate portable MAC development, and have some overlaps with the ideas in this chapter. But the concept of identical code for simulation and implementation is not discussed.

The main contribution here are the presentation of this unified software framework for MAC protocol development for underwater networks, and a case study where one MAC protocol is simulated and also operated in the modem.

The authors will make this framework publicly available, whereby other researchers can develop realistic MAC protocols that will run in any UNA compliant modem without requiring modification. This could help make MAC protocol results from different groups around the world easily comparable on equal terms. Typically, simulations are performed to test out MAC protocol ideas and get some indicative results. Very rarely are these translated into real acoustic modem trials and compared with simulation results. This framework makes a modem trial an easy step after simulation study. The need for such a globally accepted framework with standardized interfaces for MAC simulation study was recently emphasized by Otnes et al. (2009) and it is hoped that the work presented here is a step toward such a standard. It should be noted that there are some other projects currently working towards similar goals (Guerra et al., 2009). Petroccia et al. (2011) presents one of the latest results in this area, where a ns2 based simulation system makes it easy for developers to take protocol implementations to sea trials.

In Section 8.2, we present the FAPI and UNA standards used as the basis of the framework. After a brief look at the ARL modem in Section 8.3, we look at some of the details of the simulator in Section 8.4. We illustrate how MAC code is implemented in Section 8.5. In Section 8.6 we look at a sea-trial that used this framework.

8.2 FAPI and UNA

This software framework is based on the proposal for a unified framework called Underwater Network Architecture (UNA) published earlier (Chitre et al., 2006).

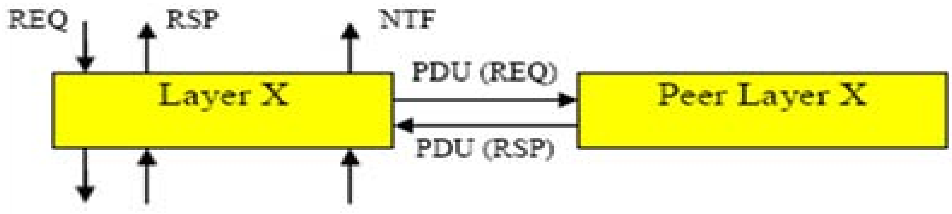


Figure 8.1: Message nomenclature in UNA (adapted from (Chitre et al., 2006))

In that proposal, a structured inter-layer message passing mechanism was defined as shown in Figure 8.1. It uses a request (REQ), response (RSP), and notification (NTF) paradigm for inter-layer communications. The higher layer sends a REQ to which it expects a RSP. NTFs are unsolicited messages from the lower layer. This simple model suffices for all the interactions between layers in the acoustic network stack.

The data link layer is defined to provide single-hop data transmission within which the main sub function is Media Access Control (MAC). DLL can also provide reliability optionally (through retransmissions). The UNA based system expects mainly the following REQ, RSP and NTFs from DLL: “Send Packet REQ”, “Packet Transmitted RSP”, “Error RSP” and “Incoming Packet NTF”.

The UNA based physical layer is defined to provide half duplex transmission and reception of data. It provides modulation and error correction capability and orthogonal channels may be selected via parameters. It uses the following REQ, RSP and NTFs: “Send Packet REQ”, “Packet Transmitted RSP”, “Error RSP” and “Incoming Packet NTF”. Other extensions to provide functionalities such as carrier sense will be added in the next version of UNA.

Though the above REQ/RSP/NTF model for physical layer is the same in nomenclature as the DLL, the functionality is different. For example, when the

DLL operates in reliable transmission mode, "Packet Transmitted RSP" indicates that a packet has been reliably delivered through re-transmissions, whereas in the physical layer, it indicates an un-acknowledged unilateral transmission. Additionally, both DLL and physical layer use parameter setting and retrieval messages "Set Parameter REQ" and "Get Parameter REQ" to set various layer related parameters.

A Framework API (FAPI) was also previously proposed to provide many essential services for a network stack (Chitre et al., 2006). It is a C API to abstract hardware and OS functionality and handles layer registration, message queues for inter-layer communications and timers. Implementation of FAPI requires a single threaded model, though multi-threaded implementations are possible. The API is shown in Appendix D.

DLL implementations adhering to FAPI can easily be ported from one system to another as long as the light weight FAPI interface is supported. For systems that do not have FAPI, the interface can easily be added. This framework was identified recently as a potential framework for MAC standardization by an independent group (Otnes et al., 2009).

8.3 ARL Modem

The following is a brief look at the acoustic modem where this framework has been implemented. This modem is used for the sea trials described later on in this chapter.

The ARL modem employs OFDM modulation and operates at 31.25 kHz

Figure 8.2: The ARL modem used in the tests

centre frequency with 12.5 kHz bandwidth (ARL, URL). It has a maximum range of 3 km and has up to 7.5 kbps data rate. It supports repetition, convolutional, Golay and LDPC coding. OFDM packets consist of two parts, a detection preamble and a data modulated signal. The detection preamble is a DSSS sequence to provide reliable detections and accurate frame synchronization. The modem physical layer implements the FAPI interface. DLL code can be developed to use this FAPI interface to send and receive data. Various parameters also can be set using the FAPI API.

8.4 The Simulator

The ARL modem simulator was developed using the discrete event simulation package Omnet. It's written in C++ and provides flexible and fast simulation of discrete-event simulations. Each of the layers in the protocol stack is implemented as an Omnet module. The underwater acoustic channel is thus implemented as an Omnet module. For MAC simulation study, three modules – Network, data link and physical layers form a logical node module (represents a single modem). Each node is linked to the channel module.

Accuracy of the simulator have been verified through analysis of timing logs of packets send and received, graphical representation of protocol behaviour as well as numerical accounting of packets send, received and lost due to various means like Bit Errors (BER) and collisions. This simulator has been in use for a number of years and was used in numerous random access protocol studies at ARL.

8.4.1 The unified simulator and modem software model

FAPI was designed to be a C language API for the modem so that it is easily implemented in any hardware system. The C++ based Omnet system for the simulator was independently chosen for being freely available for academic use and its simplicity and ease of use. This is linked to a FAPI framework as shown in Figure 8.3.

Both simulator and modem have FAPI implementations. The simulator implementation of FAPI uses a call-back mechanism to interact with Omnet discrete event API through dummy Omnet layer. The same DLL/MAC module, written using FAPI interface, is shown to exist for both modem and simulator. The FAPI based DLL/MAC module in the simulator does not interact with Omnet API directly. This also shows a possible mechanism that can be used by other existing underwater simulators to adopt the proposal here to provide a common FAPI interface for the DLL/MAC.

In the simulation, the Omnet network layer acts as the data generator for

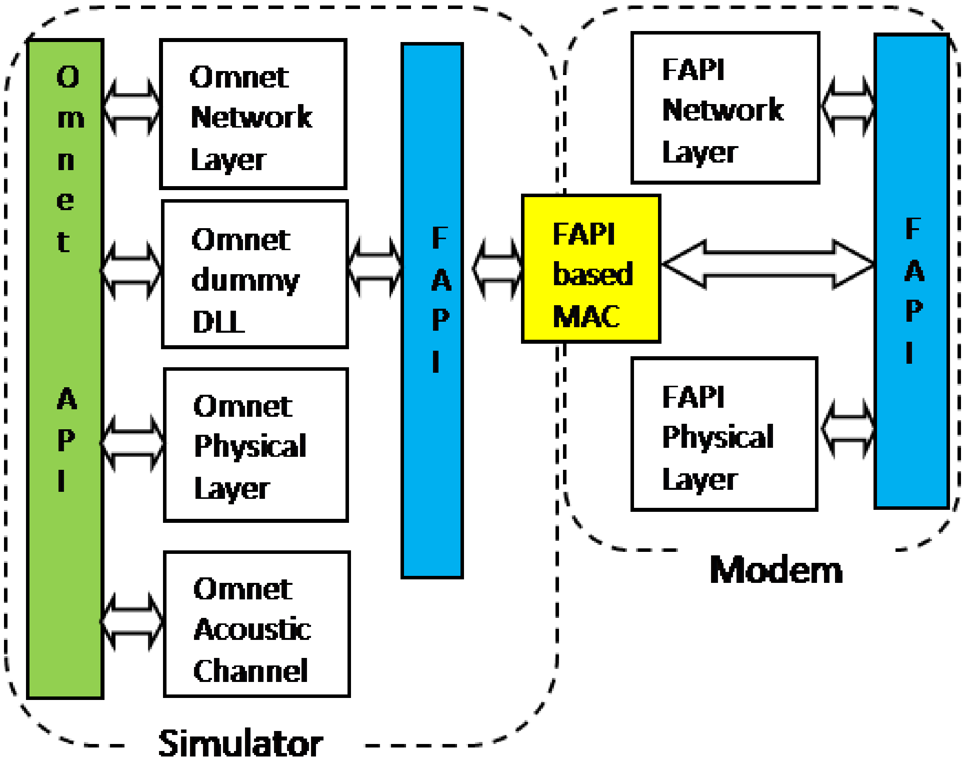


Figure 8.3: Simulator and modem software model

the data link layer below. The data for the physical layer from DLL/MAC layer goes to the Omnet physical layer which emulates the modem, which in turn is send across the channel module that models collisions and BER.

8.4.2 Simulator physical layer details

The physical Layer is assumed to be a half duplex system as is usually the case in most commercial acoustic modems. BER and collisions are simulated at the physical layer. A simplified state diagram for the physical layer is shown in Figure 8.4.

RX refers to reception of packets from other nodes and TX refers to transmission. Receive collisions that occur during transmissions do not interrupt the transmission, but the incoming packets will be lost due to the half duplex model. All collided RX packets are considered lost. In a modem operating at sea, there

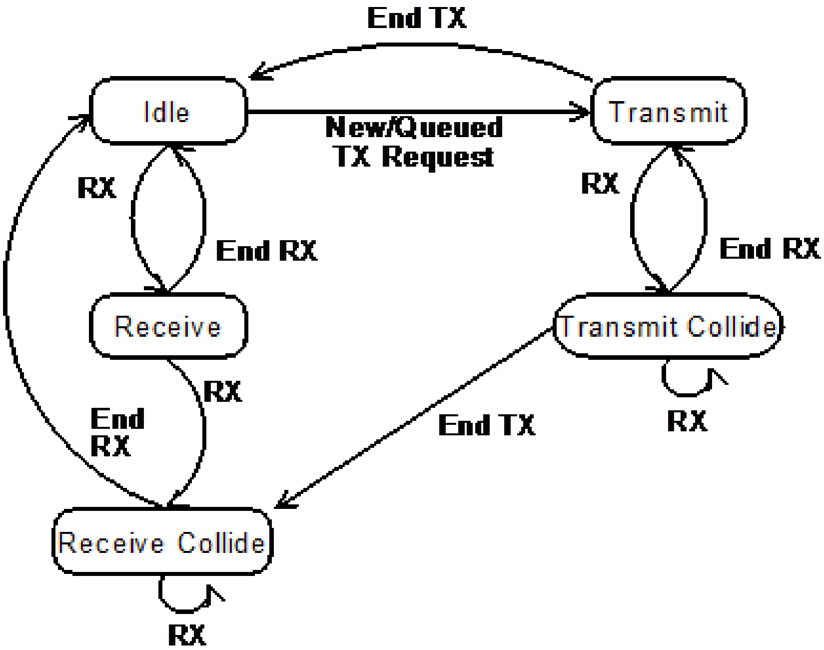


Figure 8.4: Simplified physical layer state diagram (adapted from (Shahabudeen and Chitre, 2005))

is a possibility that partial collisions could be tolerated and hence in this regard the simulator will give more conservative results in terms of collisions.

In the ARL modem simulator, detection and BER based errors are handled by the simulator's physical layer module. All packets are modelled as consisting of two parts, a detection preamble and a data modulated portion following that. Detection is characterized by detection probability P_d and decoding success probability by P. Packet loss probability P_{loss} is then

$$P_{\text{loss}} = 1 - P_d P \tag{8.1}$$

The simplest option to model detection and decoding is to assign fixed values P_d and P to match experimental conditions (see Section 8.6.1). For greater realism, it is possible to use SNR based BER and P calculations in the simulator. The SNR is computed as follows where N_0 represents the ambient noise power spectral density.

$$SNR = \frac{Received\ Power}{N_0 \times Bandwidth} \tag{8.2}$$

The E_b/N_0 ratio is then computed as

$$\frac{E_b}{N_0} = SNR \times \frac{Bit\ Rate}{Bandwidth} \tag{8.3}$$

Associated BER and corresponding P can then be computed based on a given modulation type such as QPSK using analytical or empirical models. The overall packet loss probability can be then computed as in (8.1). Other methods can also be used such as range dependent BER profile of particular modems as used in the simulations for moving AUVs as shown earlier in Figure 7.1.

In many published simulation results on underwater networks, detection and decoding errors are not considered (only collisions), which would undermine their validity in field trials, since in most modem sea trials, packet BER and detection losses can be a significant factor.

8.4.3 Channel model details

A simple channel model is used. The channel propagates all packets to all nodes except the sender and accounts for the propagation delay and path loss. The path loss model used is the spherical spreading model:

$$path\ loss\ in\ dB = 20 \log(range) \tag{8.4}$$

More complicated models (e.g. (Raysin et al., 1999)) could be implemented, but the above is deemed sufficient as a first order approximation for the scope of most simulations. In simulations where fixed P and P_d are chosen (as mentioned in

Section 8.4.2) to match experimental conditions, no path loss is modelled at the channel, only propagation delay is modelled.

Typically, additive white Gaussian noise (AWGN) is used, although channels vary in noise characteristics such as the warm shallow water channels with impulsive noise. Node motion etc also can be modelled as described earlier in Chapter 7.

8.4.4 Simulator limitations

A simulator needs to capture essential behaviour of the underwater acoustic channel. To test a MAC protocol, the physical layer emulation has to capture essential behaviour of a modem to a reasonable accuracy. For a protocol such as the MACA based protocol illustrated in an earlier chapter, important characteristics required from the combination of physical layer and the underlying channel are packet detection and loss emulation (due to various factors such as path loss, ambient noise, packet collisions) as well as propagation delays. These are currently captured accurately by the simulator.

However, the simulator does not currently model reverberations and multipath associated with real sea environments. Such characteristics were not deemed necessary to evaluate the performance of the MAC protocol being evaluated at ARL to be used over the ARL OFDM modem. One of the reasons is that OFDM handles multipath effectively by the use of cyclic prefix in each of the packets. The higher MAC layer does not see multi-path effects directly. Such multi-path effects would translate into packet detection and decoding losses at the physical layer and this is indeed captured by the current simulator.

However for some type of MAC protocols, such characteristics cannot be ignored in the simulator. For example T-LOHI protocol relies on tone transmissions for channel capture (Syed et al., 2008). A tone by itself cannot carry source node or other information and the protocol relies on detection of tones to perform channel capture. If there are reverberations, a node will be unable to distinguish other nodes' tones from its own, and thereby the main mechanism in the protocol will fail. If reverberations are not modelled in the simulator used for testing such a protocol, such problems will not arise and results will not meaningfully predict performance of such a protocol operating in a modem at sea.

Thus, it should be highlighted that there are simplifying assumptions in all MAC simulators and none can capture the effects of a modem operating at sea completely. There is no validation as important as testing the protocol at sea using a modem. This is the main motivation behind the concept of unifying simulation and implementation framework, which provides MAC protocol developers a way to translate simulation studies to sea trials with little added effort.

8.5 Writing MAC Code

The MAC is a sub-function of data link layer. MAC function is responsible for deciding when packets are sent and received. The data link layer has other functions such as link adaptation (e.g., choosing and setting physical layer parameters for optimal performance). This link adaptation could include power control also, since transmission power setting can be viewed as a link parameter. There is

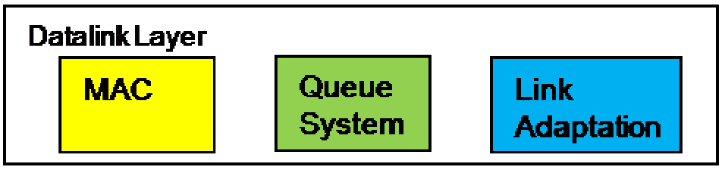


Figure 8.5: Key sub-functions of data link layer

inter-play between these sub-functions, for example to do power control, the MAC control packets might carry transmit power and recommended power information etc. There could also be a queuing sub-system that decides the data to be sent according to priorities, models such as FIFO, lifetime and expiry etc. These functions are represented in Figure 8.5.

The following discussion is on the MAC function and how it is implemented in the framework. Appendix D outlines the UNA messages, FAPI interfaces and some utility functions provided by the DLL to the MAC function. A sample MAC implementation for a simplistic ALOHA-like protocol is presented in Appendix D.4. This code sample shows that it is very simple and similar to the way most MAC protocols are written in a simulation program. Since the interfaces are standardized across the simulator and the modem, the code runs on both with no changes and avoids porting issues and effort. Correct porting is important in order to achieve the performance gains seen in simulations and to help cross verify simulations with experiments. In general, simulator and modem software interfaces could have differences that make porting difficult.

The framework also ensures that the simulation code for the MAC protocol handles all the required complexities in a modem. For example when a MAC protocol requests the physical layer to send a packet, the physical layer in a modem might be unable to send it at that point in time and choose to return an error. A simple simulation program might not take into account such situations and would require modifications during modem implementation, and it could have potential impact on the algorithm. How the framework handles this is shown in Appendix D. There could be other important peculiarities in a modem. For example, the ARL modem uses a high power amplifier (HPA) for transmission and it takes about 300ms to settle after turn on. And when HPA is turned on, no receptions are possible. If in a certain MAC protocol, it needs to transmit a control packet right away, it would realize that it has to wait about 300ms before it can actually transmit. For example, in position based collision avoidance protocols, it could significantly affect the main protocol mechanisms as 300ms translates to roughly 500meters. So the HPA might need to be switched on 300ms before the intended transmission to avoid such problems. But once HPA is switched on, receptions are not possible during that period, and this again can have repercussions. Therefore, all such modem behaviour and limitations should ideally be captured into simulations in order to develop realizable protocols for a modem implementation and to get accurate assessment of protocol performance through simulations. How the framework handles the HPA delay is shown in Appendix D.

8.6 Modem Trials and Results

In this case study, the MACA-EA protocol is implemented and run on both the simulator and in the ARL acoustic modem. The protocol used here is described in detail in chapter 4 and it uses RTS/CTS handshake followed by a batch of

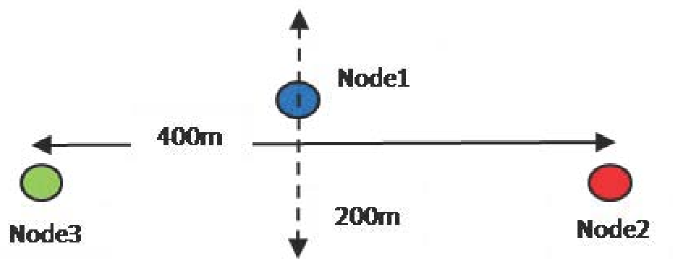


Figure 8.6: Node deployment geometry

DATA packets and an optional ACK packet that indicates the number of packets successfully received. During the modem sea trials, packet detection and decoding probability is captured. This is in turn used in the simulations using the simple model of using constant P_d and P mentioned in Section 8.4.2.

8.6.1 Sea trials

Here some further details of the medium range trial in Singapore Coastal waters (south of Pulau Ubin Island) conducted in June 2009 is described (briefly discussed in Chapter 4). The modems were deployed as shown in Figure 8.6 and Figure 8.7. Node 2 is in boat "West Squadron", Node 3 is in boat "Dolphin". West Squadron and Dolphin were separated by about 400 meters on both days. Node 1 is in a chase boat that moved about in between the main boats as shown in Figure 8.6.

The separation between Node 2 and 3 was of the order of 400m. The depths were 6 to 12 meters. The modems used were ARL modems. The traffic pattern was Node 1 to Node 2, Node 2 to Node 3 and Node 3 to Node 2. The modems estimated $P_d = 1$ and $P = 0.9$ to 0.95 for the tests shown here. Thus, the primary loss factor is contention and collisions. The packet duration in the modem is 0.6seconds. It can also be noted that there is a limited PCS present

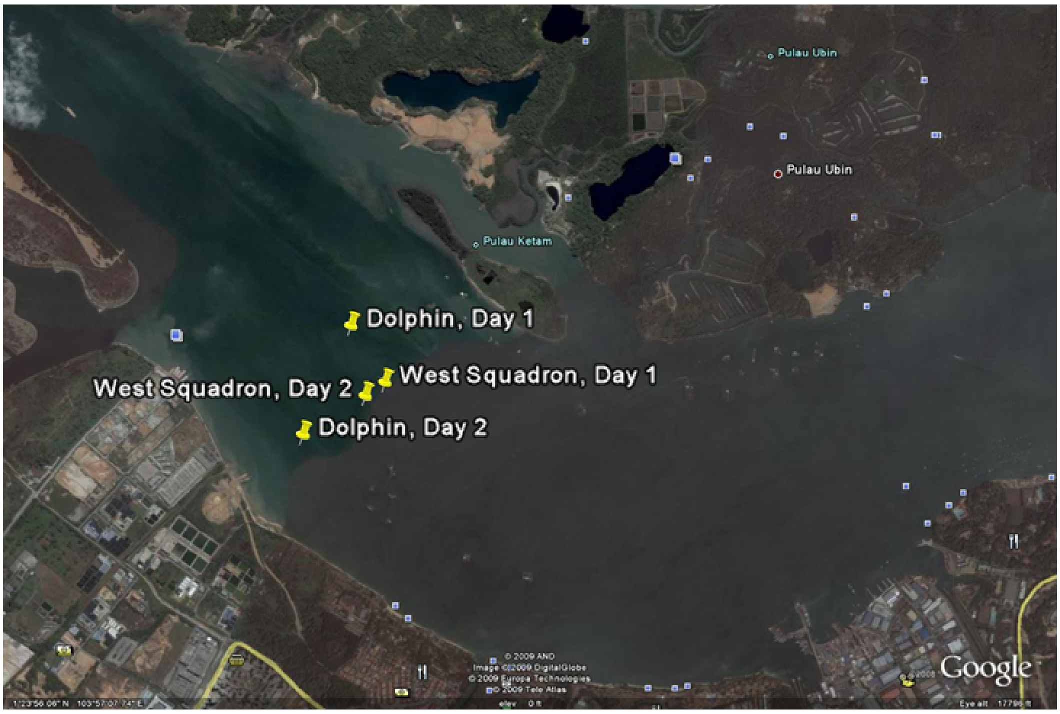


Figure 8.7: Sea trial location off the north of Singapore and below Pulau Ubin island. Boats were separated by about 400 to 500 meters during trials. Water depth was 6 to 12 meters.

in the modem since the modem hardware does not allow transmission to start if an ongoing reception is in progress. For batch transmission, HPA is turned on/off only at the start and end of the batch only for efficiency. The results were shown in Figure 4.6. Simulations give a reasonably accurate match. Throughput is seen to improve with batch size, as proven in simulations. As mentioned above, in the high success probability scenario used here, collisions are the most significant source of losses.

Trial results such as above provide a good validation to the simulator's accuracy in modelling propagation delays, packets losses due to BER, collisions etc, and confirm implicitly that it captures the behaviour of the modem physical layer and the channel that is relevant to the MAC protocol being investigated. Once such as close match with sea-trials is established, the simulator can be relied upon for further investigations for similar protocols.

8.7 Conclusion

Simulations are used for MAC protocol development since the design-debug-validate cycle is less costly. Being able to use the same code in simulator and in a modem through a published structured software interface is of immense value. Simulation results can meaningfully predict modem performance and aids greatly in MAC protocol development. There is no further porting to be done after a simulation study to conduct a modem trial. Sea trials have proven the utility of the framework.

MAC protocol standardization for underwater networks is being attempted by more than one group currently. MAC protocol candidates need to be evaluated on a common platform to provide comparative results (Otnes et al., 2009). One key aim for the framework proposed in this chapter, is to provide such a platform that has been validated through modem field tests.

There is certainly more work to be done in collaboration with interested researchers who would like to contribute to such initiatives. As mentioned earlier, there are other very closely related projects that are looking at unifying interfaces to modem physical layer etc to facilitate portable MAC software development (Shusta et al., 2008; Guerra et al., 2009). In future, such parallel initiatives should be linked to develop a globally acceptable framework.

Chapter 9

Conclusion

Time domain protocols are well suited for UANs. MACA-based protocols can be seen as an extension of dynamic TDMA and inherit many of its advantages. They offer even greater robustness as they do not require precise time synchronization, which is often difficult to achieve in many underwater networks. They also offer ad hoc functionality and scalability, both of which are not easy to achieve with regular TDMA.

A very efficient MACA-based protocol called MACA-EA has been developed. Enhancements such as monitoring of DATA packets during contention phase (to aid VCS) and Early-Multi-ACK ARQ method for batch DATA transmission in the MACA-EA protocol give better throughput performance than most reported results for similar protocols for reliable transfer. The analytical model for MACA-EA can help compute the impact of batch size, detection and decoding probability and other parameters on normalized throughput. Service time distribution of the protocol is nearly exponential with an analytically known mean. We obtained a new model formulation that can help compute the total queuing delay for the retry based MACA-EA protocol. The queueing analysis shows that for Poisson arrivals, while the throughput increases with batch size, so does the waiting time, and that there exists an optimum batch size. Another useful result is that the optimum value of the back-off counter is a linear function of the number of nodes. The protocols were implemented in acoustic modems, and medium range field trials corroborate the simulations and analysis well. System analysis is now possible without needing to resort to extensive simulations.

In order to address the larger problem of a heterogenous UAN, a comprehensive MAC protocol suite – MAC-AMM was developed. The MAC protocol suite has both distributed (MACA-EA, DATA-ACK) and centralized (MACA-C) operating modes. A detailed analytical comparison of the performance of all the modes has been presented along with TDMA as a benchmark. Optimum performance can be achieved by selecting the mode based on traffic-intensity. A new state dependent DATA-ACK protocol is also presented, and it offers the best performance for low arrival rates. The novel sequencing feature in MACA-SEA also improves performance of MACA-EA further. In MAC-AMM protocol suite, nodes self regulate the operation modes and topology according to self capability, capabilities of neighbouring nodes and traffic intensity.

MACA-based protocols can also be successfully used on multiple modems optimized for different ranges in an AUV network. The unified DLL algorithm MACA-MCP, synergistically uses multiple physical layers and uses position information exchange. The network layer will need to use novel and efficient routing techniques to exploit this DLL algorithm to achieve full connectivity. The novel protocols that perform simultaneous transmissions, Twin-TDMA, Dynamic Twin-TDMA and Twin-Aloha utilize propagation delay constructively to provide better waiting time and throughput.

The software framework developed as part of this work, for seamless MAC simulation and implementation, allows simulation results to meaningfully predict modem performance and aids greatly in MAC protocol development. There is no further porting to be done after a simulation study to conduct a modem trial. Many sea trial results showed the utility of the framework. This can also help the standardization process, to evaluate MAC protocol candidates on a common platform.

The work in this thesis provides reliable results based on three modes of investigation: simulations, mathematical analysis and sea-trials. The simulations were carried out on an software platform that allows seamless operation of the code in acoustic modems. The core protocols have been tested extensively, and have been used successfully in real sea-trials to exchange data. Simulations, field experimentation and mathematical analysis have been combined to provide accurate results on the performance of MACA-based protocols in UANs. A key contribution of this work is thus the accuracy and validity of the results, along with the novelty in protocol variations.

The central focus of the thesis has been the use of MACA-based protocols in UANs and this thesis provides some new and useful insights for underwater network designers. Mission planners planning to use such networks can understand the performance boundaries and ensure that communications requirements match what is feasible. Underwater networks remain challenged in terms of data rates and delays involved, and it is important to fully understand its potential and limitations in real applications.

9.1 Future Work

There are many avenues that remain to be explored as a natural continuation of the work in this thesis. The following are some of the key areas for further investigation.

- Long packets vs. batch mode DATA: The model for data transmission used in this thesis is multiple DATA packets sent as a batch. It should be compared more extensively to the alternative of using single long DATA packets as discussed in Section 4.7.4.

- Link tuning and power control: As discussed in Section 4.7.6, more study is required to investigate link tuning (which includes FEC) and power control using RTS/CTS handshake. The analytical models could incorporate this.

- Further protocols based on "SuperTDMA": It is possible that many novel protocols are awaiting to be discovered that can utilize the "SuperTDMA" concept even more effectively than the Twin-MAC protocols of Chapter 6. Furthermore, it will be useful to have sea-trials that can demonstrate such protocols.

- Starvation: The concept of starvation is well known in terrestrial radio wireless networks. It refers to the situation where some nodes are "starved" in terms of getting transmission opportunities. When using RTS/CTS based contention resolution, it is possible that due to various reasons some nodes dominate over others. Starvation prevention and minimization techniques need to be explored for UAN MAC based on MACA.

- Standardization and MAC-AMM: MAC protocol standards for UANs are long overdue. Concepts such as MAC-AMM will be useful in this regard. More collaborative effort on a global scale is required to make this happen. A common software platform for simulation evaluation of MAC protocols should be established, so that performance comparisons become direct and meaningful.

- MACA-MCP and similar multi-channel concepts: This involves development of high speed short range modems and coupling them with longer range modems through MACA-MCP in sea trials with AUVs. More detailed mathematical analysis and use of other mobility models and traffic patterns for MACA-MCP simulations.

- Sea-trials to do further performance evaluations of UAN MAC protocols.

- Comprehensive extension of performance analysis to random multi-hop UANs.

Bibliography

AKYILDIZ, I. F., POMPILI, D., AND MELODIA, T. 2006. State-of-the-art in protocol research for underwater acoustic sensor networks. In *ACM International Workshop on Underwater Networks (WUWNet)*, Los Angeles, USA.

ARL Acoustic Research Laboratory, NUS, Underwater Acoustic Communications. http://arl.nus.edu.sg/twiki/bin/view/ARL/UNET.

BIANCHI, G. 2000. Performance analysis of the IEEE 802.11 distributed coordination function. *IEEE Journal on Selected Areas in Communications 18,* 3, 535–547.

BISAGLIA, P., BOCCARDI, F., D'AMICO, V., MORETTI, M., SCANAVINO, B., AND VERONESI, D. 2005. On the capacity comparison of multi-user access techniques for fourth generation cellular TDD OFDM-based systems. In *Vehicular Technology Conference. VTC 2005-Spring*, Volume 5, pp. 3077–3081 Vol. 5.

CARLSON, E. A., BEAUJEAN, P. P., AND AN, E. 2004. Simulating communication during multiple AUV operations. In *IEEE/OES Autonomous Underwater Vehicles*, pp. 76–82.

CASARI, P., MARELLA, S., AND ZORZI, M. 2007. A comparison of multiple access techniques in clustered underwater acoustic networks. In *IEEE Oceans' 07*, Aberdeen, Scotland.

CHAN, C. Y. AND MOTANI, M. 2007. An integrated energy efficient data retrieval protocol for underwater delay tolerant networks. In *IEEE Oceans' 07*, Aberdeen, Scotland.

CHATZIMISIOS, P., BOUCOUVALAS, A. C., AND VITSAS, V. 2003. Packet delay analysis of IEEE 802.11 MAC protocol. *Electronics Letters 39,* 18, 1358–1359.

CHAUDHRY, M. L. AND TEMPLETON, J. G. C. 1984. *A first course in bulk queues.* John Wiley & Sons.

CHITRE, M. 2006. Underwater acoustic communications in warm shallow water channels. Ph. D. thesis, National Univeristy of Singapore.

CHITRE, M., FREITAG, L., SOZER, E., SHAHABUDEEN, S., STOJANOVIC, M., AND POTTER, J. 2006. An architecture for underwater networks. In *IEEE OCEANS'06 Asia Pacific*, Singapore.

CHITRE, M., MOTANI, M., AND SHAHABUDEEN, S. 2010. A scheduling algorithm for wireless networks with large propagation delays. In *OCEANS 2010*, Sydney.

CHITRE, M., MOTANI, M., AND SHAHABUDEEN, S. 2011. Throughput of wireless networks with large propagation delays. *Under review, Transactions on Networking*.

Chitre, M., Ong, S. H., and Potter, J. 2005. Performance of coded OFDM in very shallow water channels and snapping shrimp noise. In *MTS/IEEE OCEANS 2005*, pp. 996–1001 Vol. 2.

Chitre, M., Shahabudeen, S., and Stojanovic, M. 2008. Underwater acoustic communications and networking: Recent advances and future challenges. *Marine Technology Society Journal - "The State of Technology in 2008" 42,* 1.

Dolc, A. P. and Stojanovic, M. 2007. Optimizing the transmission range in an acoustic underwater network. In *OCEANS'07*, Vancouver, Canada.

Doufexi, A., Armour, S., Butler, M., Nix, A., Bull, D., McGeehan, J., and Karlsson, P. 2002. A comparison of the HIPERLAN/2 and IEEE 802.11a wireless LAN standards. *IEEE Communications Magazine 40,* 5, 172–180.

Doukkali, H. and Nuaymi, L. 2005. Analysis of MAC protocols for underwater acoustic data networks. In *IEEE 61st Vehicular Technology Conference, VTC 2005-Spring*, Volume 2, pp. 1307–1311.

Doukkali, H., Nuaymi, L., and Houcke, S. 2006. Distributed MAC protocols for underwater acoustic data networks. In *IEEE 64th Vehicular Technology Conference, VTC-2006 Fall.*, pp. 1–5.

DTDMA Time Division Multiple Access: Dynamic TDMA. http://en.wikipedia.org/wiki/Dynamic_TDMA.

Erlang MLE for Erlang distribution. http://sci.tech-archive.net/Archive/sci.stat.math/2010-03/msg00037.html.

Foh, C., Zukerman, M., and Tantra, J. 2007. A markovian framework for performance evaluation of IEEE 802.11. *IEEE Transactions on Wireless Communications 6,* 4, 1276–1265.

Freitag, L., Grund, M., Singh, S., Partan, J., Koski, P., and Ball, K. 2005. The WHOI micro-modem: an acoustic communications and navigation system for multiple platforms. In *OCEANS, 2005. Proceedings of MTS/IEEE*, pp. 1086–1092 Vol. 2.

Freitag, L. and Singh, S. 2000. Multi-user frequency hopping underwater acoustic communication protocol. Technical report, WHOI.

Freitag, L. E., Grund, M., Partan, J., Ball, K., Singh, S., and Koski, P. 2005. Multi-band acoustic modem for the communications and navigation aid AUV. In *MTS/IEEE OCEANS*, pp. 1080–1085 Vol. 2.

Fullmer, C. L. and Garcia-Luna-Aceves, J. J. 1995. Floor acquisition multiple access (FAMA) for packet-radio networks. In *SIGCOMM 95*, Cambridge, MA USA.

Garcia-Luna-Aceves, J. J. and Fullmer, C. L. 1998. Performance of floor acquisition multiple access in ad-hoc networks. In *Third IEEE Symposium on Computers and Communications, ISCC '98*, pp. 63–68.

Gross, D. and Harris, C. M. 1998. *Fundamentals of Queueing Theory.* Wiley.

Guerra, F., Casari, P., and Zorzi, M. 2009. World Ocean Simulation System (WOSS): A simulation tool for underwater networks with realistic propagation modeling. In *ACM International Workshop on Underwater Networks (WUWNet)*, Berkeley, USA.

Guo, X., Frater, M. R., and Ryan, M. J. 2006. A propagation-delay-tolerant collision avoidance protocol for underwater acoustic sensor networks. In *MTS/IEEE OCEANS06*, Singapore.

Guo, X., Frater, M. R., and Ryan, M. J. 2009. Design of a propagation-delay-tolerant MAC protocol for underwater acoustic sensor networks. *IEEE Journal of Oceanic Engineering 34,* 2.

Gupta, N. and Kumar, P. R. 2004. A performance analysis of the 802.11 wireless LAN medium access control. *Communications in Information and Systems 3,* 4, 279–304.

Heidemann, J., Wei, Y., Wills, J., Syed, A., and Yuan, L. 2006. Research challenges and applications for underwater sensor networking. In *IEEE Wireless Communications and Networking Conference, WCNC*, Volume 1, pp. 228–235.

Hou, B., Hinton, O., Adams, A., and Sharif, B. 1999. An time-domain-oriented multiple access protocol for underwater acoustic network communications. In *OCEANS '99 MTS/IEEE. Riding the Crest into the 21st Century*, Volume 2, pp. 585–589 vol.2.

Ipatov, V. P. 2005. *Spread Spectrum and CDMA, Principles and Applications.* John Wiley & Sons Ltd.

Janssen, A. and Leeuwaarden, J. v. 2005. Analytic computation schemes for the discrete-time bulk service queue. *Queueing Systems: Theory and Applications 50,* 2-3, 141–163.

Jarvis, S., Janiesch, R., Fitzpatrick, K., and Morrissey, R. 1997. Results from recent sea trials of the underwater digital acoustic telemetry system. In *OCEANS '97. MTS/IEEE Conference Proceedings*, Volume 1, pp. 702–708 vol.1.

Jiejun, K., Jun-hong, C., Dapeng, W., and Gerla, M. 2005. Building underwater ad-hoc networks and sensor networks for large scale real-time aquatic applications. In *IEEE Military Communications Conference, MILCOM*, pp. 1535–1541 Vol. 3.

Jun-Hong, C., Jiejun, K., Gerla, M., and Shengli, Z. 2006. The challenges of building mobile underwater wireless networks for aquatic applications. *Network, IEEE 20,* 3, 12–18.

Kalscheuer, J. M. 2004. A selective automatic repeat request protocol for undersea acoustic links. Master's thesis, Naval Postgraduate School.

Karn, P. 1990. MACA - a new channel access method for packet radio. In *ARRL/CRRL Amateur Radio 9th computer Networking Conference*, pp. 134–140.

Kebkal, A., Kebkal, K., and Komar, M. 2005. Data-link protocol for underwater acoustic networks. In *Oceans 2005 - Europe*, Volume 2, pp. 1174–1180 Vol. 2.

KREDO, K., DJUKIC, P., AND MOHAPATRA, P. 2009. STUMP: Exploiting position diversity in the staggered TDMA underwater MAC protocol. In *INFOCOM 2009, IEEE*, pp. 2961–2965.

LIN, C. R. AND GERLA, M. 1997. Adaptive clustering for mobile wireless networks. *Selected Areas in Communications, IEEE Journal on 15,* 7, 1265–1275.

MARTOYO, I., SCHOBER, H., AND JONDRAL, F. 2002. CDMA versus OFDM, a performance comparison in selective fading channels. In *IEEE Seventh International Symposium on Spread Spectrum Techniques and Applications*, Volume 1, pp. 139–143 vol.1.

MCCOY, K. 2009. JANUS: From primitive signal to orthodox networks. In *Underwater Acoustic Measurements: Technologies & Results*, Nafplion, Greece.

MOLINS, M. AND STOJANOVIC, M. 2006. Slotted FAMA: a MAC protocol for underwater acoustic networks. In *MTS/IEEE OCEANS06.*

MURATORE, F. AND ROMANO, G. 1996. GSM versus CDMA: performance comparisons. In *Global Telecommunications Conference, GLOBECOM '96*, Volume 1, pp. 519–524 vol.1.

NAIN, P., TOWSLEY, D., BENYUAN, L., AND ZHEN, L. 2005. Properties of random direction models. In *INFOCOM 2005*, Volume 3, pp. 1897–1907 vol. 3.

OMNET++ Discrete event simulation system. http://www.omnetpp.org/.

OTNES, R., JENSERUD, T., VOLDHAUG, J. E., AND SOLBERG, C. 2009. A roadmap to ubiquitous underwater acoustic communications and networking. In

Underwater Acoustic Measurements: Technologies & Results, Nafplion, Greece, pp. 557–564.

Park, C. G., Jung, H. S., and Han, D. H. 2006. Queueing analysis of IEEE 802.11 MAC protocol in wireless LAN. pp. 139 – 139.

Peleato, B. and Stojanovic, M. 2006. A MAC protocol for ad hoc underwater acoustic sensor networks. In *WUWNet'06*, Los Angeles, USA.

Peleato, B. and Stojanovic, M. 2007a. A channel sharing scheme for underwater cellular networks. In *IEEE Oceans' 07*, Aberdeen, Scotland.

Peleato, B. and Stojanovic, M. 2007b. Distance aware collision avoidance protocol for ad-hoc underwater acoustic sensor networks. *Communications Letters, IEEE 11,* 12 (December), 1025–1027.

Peng, X. and Cui, J. H. 2006. Exploring random access and handshaking techniques in large-scale underwater wireless acoustic sensor networks. In *OCEANS 2006*, pp. 1–6.

Petrioli, C., Petroccia, R., and Stojanovic, M. 2008. A comparative performance evaluation of MAC protocols for underwater sensor networks. In *IEEE Oceans'08.*

Petroccia, R., Shusta, J., and Freitag, L. 2011. From underwater simulation to at-sea testing using the ns-2 network simulator. In *IEEE OCEANS 2011*, Spain.

PEYRAVI, H. 1999. Medium access control protocols performance in satellite communications. *IEEE Communications Magazine 37,* 3, 62–71.

PROAKIS, J. G., SOZER, E. M., RICE, J. A., AND STOJANOVIC, M. 2001. Shallow water acoustic networks. *IEEE Communications Magazine 39,* 11, 114–119.

RAPPAPORT, T. S. 2002. *Wireless Communications, Principles and Practice* (2 ed.). Prentice Hall.

RAYSIN, K., RICE, J., DORMAN, E., AND MATHENY, S. 1999. Telesonar network modeling and simulation. In *MTS/IEEE OCEANS '99*, Volume 2, pp. 747–752 vol.2.

RICE, J., CREBER, B., FLETCHER, C., BAXLEY, P., ROGERS, K., MCDONALD, K., REES, D., WOLF, M., MERRIAM, S., MEHIO, R., PROAKIS, J., SCUSSEL, K., PORTA, D., BAKER, J., HARDIMAN, J., AND GREEN, D. 2000. Evolution of Seaweb underwater acoustic networking. In *MTS/IEEE OCEANS 2000*, Volume 3, pp. 2007–2017 vol.3.

RODOPLU, V. AND MIN KYOUNG, P. 2005. An energy-efficient MAC protocol for underwater wireless acoustic networks. In *MTS/IEEE OCEANS*, pp. 1198–1203 Vol. 2.

SALVA-GARAU, F. AND STOJANOVIC, M. 2003. Multi-cluster protocol for ad hoc mobile underwater acoustic networks. In *OCEANS 2003*, Volume 1, pp. 91–98 Vol.1.

Shahabudeen, S., Chitre, M., and Motani, M. 2007. A multi-channel MAC protocol for AUV networks. In *IEEE Oceans' 07*, Aberdeen, Scotland.

Shahabudeen, S., Chitre, M., and Motani, M. 2010. *Underwater Acoustic Sensor Networks*, Chapter Dynamic TDMA and MACA based Protocols for Distributed Topology Underwater Acoustic Networks. CRC Press.

Shahabudeen, S., Chitre, M., and Motani, M. 2011a. Adaptive multi-mode medium access control for underwater acoustic networks. *Submitted to Journal of Oceanic Engineering*.

Shahabudeen, S., Chitre, M., and Motani, M. 2011b. MAC protocols that exploit propagation delay in underwater networks. In *MTS/IEEE Oceans'11*, Kona, Hawaii, USA.

Shahabudeen, S., Chitre, M., Motani, M., and Low, A. 2009. Unified simulation and implementation software framework for underwater MAC protocol development. In *MTS/IEEE Oceans'09*, Biloxi, USA.

Shahabudeen, S., Chitre, M., Potter, J., and Motani, M. 2009. Multi-mode adaptive MAC protocol suite and standardization proposal for heterogeneous underwater acoustic networks. In *Underwater Acoustic Measurements*, Nafplion, Greece.

Shahabudeen, S. and Chitre, M. A. 2005. Design of networking protocols for shallow water peer-to-peer acoustic networks. In *Oceans 2005 - Europe*, Volume 1, pp. 628–633 Vol. 1.

Shahabudeen, S. and Motani, M. 2009. Modeling and performance analysis of MACA based protocols for adhoc underwater networks. In *WUWNet'09*, Berkeley, USA.

Shahabudeen, S., Motani, S., and Chitre, M. 2011. Analysis of a high performance MAC protocol for underwater acoustic networks. *Under review, Journal of Oceanic Engineering*.

Shankar, S., Chitre, M., and Jayasuriya, M. 2010. Data driven algorithms to tune physical layer parameters of an underwater communication link. In *IEEE OCEANS'10*, Sydney, (Australia).

Shusta, J., Freitag, L., and Partan, J. 2008. A modular data link layer for underwater networks. In *OCEANS 2008*, pp. 1–5.

Smith, S. M., Park, J. C., and Neel, A. 1997. A peer-to-peer communication protocol for underwater acoustic communication. In *MTS/IEEE OCEANS '97*, Volume 1, pp. 268–272 vol.1.

Sozer, E. M., Stojanovic, M., and Proakis, J. G. 2000. Underwater acoustic networks. *IEEE Journal of Oceanic Engineering 25,* 1, 72–83.

Stojanovic, M. 2002. Frequency reuse underwater: capacity of an acoustic cellular network. In *Proceedings of the second workshop on Underwater networks*, Montreal, Quebec, Canada.

Stojanovic, M. 2005. Optimization of a data link protocol for an underwater acoustic channel. In *Oceans 2005 - Europe*, Volume 1, pp. 68–73 Vol. 1.

STOJANOVIC, M. 2007. On the design of underwater acoustic cellular systems. In *IEEE Oceans'07*, Aberdeen, Scotland.

STOJANOVIC, M., FREITAG, L., LEONARD, J., AND NEWMAN, P. 2002. A network protocol for multiple AUV localization. In *MTS/IEEE Oceans '02*, Volume 1, pp. 604–611 vol.1.

STOKEY, R. P., FREITAG, L. E., AND GRUND, M. D. 2005. A compact control language for AUV acoustic communication. In *Oceans 2005 - Europe*, Volume 2, pp. 1133–1137 Vol. 2.

SYED, A., WEI, Y., AND HEIDEMANN, J. 2008. Comparison and evaluation of the T-Lohi MAC for underwater acoustic sensor networks. *Selected Areas in Communications, IEEE Journal on 26,* 9, 1731–1743.

SYED, A. A., YE, W., HEIDEMANN, J., AND KRISHNAMACHARI, B. 2007. Understanding spatio-temporal uncertainty in medium access with ALOHA protocols. In *WuWNet '07: Proceedings of the second workshop on Underwater networks*, New York, NY, USA, pp. 41–48. ACM.

TURGAY, K. AND ERDAL, C. 2006. A MAC protocol for tactical underwater surveillance networks. In *Military Communications Conference, 2006. MILCOM 2006*, pp. 1–7.

WONG, Y. F., NGOH, L. H., WONG, W. C., AND SEAH, W. K. G. 2006. Intelligent sensor monitoring for industrial underwater applications. In *Industrial Informatics, 2006 IEEE International Conference on*, pp. 144–149.

Appendix A

MACA Analysis

A.1 The Performance of the Standard ACK Model

Since there are no Early-Multi-ACKs, t_{B_VCS} in (4.10) would be t_B (4.9), with $i = 1$. Instead of the Markov chain in Fig. 4.2 leading to (4.13), we can use Fig. A.1 to compute batch service time as

$$s_b = \frac{1}{k}\left(s_{CTS} + t_B\right) = \frac{1}{k}\left(\gamma + Nt_B\right) \tag{A.1}$$

A.2 Distribution Analysis Markov Chain

Here the distribution analysis matrix for the example described in section 4.4 is shown. The state ordering is different here compared to (4.7) as detailed below. Each column represents states s_i. $\mathbf{M}_{i,j}$ is the transition probability from state s_i to state s_j. States s2 and s3 are the dummy states that represent the time delay of 2s corresponding to delay t_A (for given parameters) in state 3 in Fig. 4.1. States s_4, s_5, s_6, s_7, and s_8 are the dummy states that represent the time delay of 5s

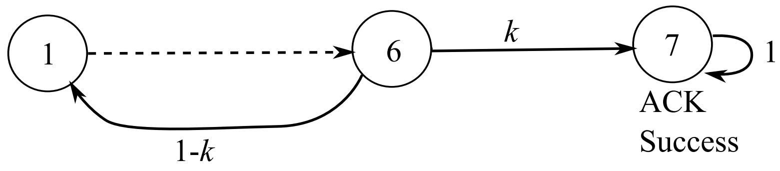


Figure A.1: Markov chain for Standard ACK model

corresponding to delay t_B (from (4.9) for given parameters $B = 4, L = 0.5, l = 1, i = 3$) in both state 4 and state 5 (in fact these states could be represented as one state) in Fig. 4.1. States s_9 and s_{10} are the dummy states that represent the time delay of 2s corresponding to delay t_A (for given parameters) in state2 in Fig. 4.1. States s_{11}, s_{12}, s_{13}, s_{14}, and s_{15} are the dummy states that represent the time delay of 5s corresponding to delay t_B (for given parameters) in state 6 in Fig. 4.1. The transition probabilities can then easily be seen by comparison with (4.7) and Fig. 4.1, with transition between dummy states having a probability of 1.

$$\mathbf{M} = \begin{bmatrix} c & b & 0 & d & 0 & 0 & 0 & 0 & a & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ q & 0 & 0 & p & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ z & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & f & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{A.2}$$

For the retry scheme in Fig. 4.2, we can use $\mathbf{M}$ to generate the complete Markov matrix $\mathbf{M_R}$ (introduced in section 4.4) as shown below, where $\eta = (1-k)^i$.

$$
\mathbf{M_R} = \begin{bmatrix} \begin{bmatrix} & & \\ & \mathbf{M} & \\ & & \end{bmatrix} & \begin{bmatrix} 0 & \dots & 0 \\ \vdots & \vdots & \vdots \\ \eta & \dots & 0 \end{bmatrix} & \begin{bmatrix} 0 \\ \vdots \\ 1-\eta \end{bmatrix} \\ \begin{bmatrix} 0 & \dots & 0 \\ \vdots & \vdots & \vdots \\ 0 & \dots & 0 \end{bmatrix} & \begin{bmatrix} & & \\ & \mathbf{M} & \\ & & \end{bmatrix} & \begin{bmatrix} 0 \\ \vdots \\ 1 \end{bmatrix} \\ \multicolumn{3}{c}{\begin{bmatrix} 0 & \dots & \dots & \dots & \dots & \dots & \dots & 1 \end{bmatrix}} \end{bmatrix} \tag{A.3}
$$

Appendix B

Analysis for MAC-AMM

B.1 DATA-ACK throughput analysis

The transmitter sends DATA and the receiver sends back ACK. The contention algorithm for sending DATA packets is the same as MACA-EA RTS sending. A node starts with a uniform probability distributed back-off in a contention window W. When the back off timer expires, a DATA is sent. Once DATA is sent, ACK timer t_A starts. The timer used to wait for ACK (t_A) (as well as CTS in MACA-EA protocol mode) is related to D and control packet time duration L to give enough time for the round trip delay as

$$t_A = 2D + 2L \tag{B.1}$$

If timer expires before reception of ACK, DATA back-off procedure starts again. If ACK is not received, the cycle repeats. Reception of DATA-ACK packets while waiting to send DATA triggers Virtual Carrier Sense (VCS). Note that 802.11 uses freezing back-off (Bianchi, 2000) whereas a constant window is used here. All nodes use the same contention window W at any given time.

In line with the definition used by Molins and Stojanovic (2006), during DATA contention phase, the slot duration l is defined as

$$l = L + D \tag{B.2}$$

The following analysis is adapted from the Markov analysis for RTS contention in Chapter 4. A node starts with a uniformly selected back-off time slot in the integer range $[1, W]$. The actual contention window time period is Wl. For simplicity of analysis, it is assumed that no collisions happen during the ACK period, assuming VCS starts due to DATA reception (results showed that this simplification did not have significant impact on the analytical predictions). So in our analysis model, ACK loss will only be due to decoding and packet detection probability. If the transmitter does not get ACK, it restarts the contention window for DATA. Any other node which had received the DATA does a VCS for ACK. It resets and restarts contention if ACK does not arrive. Thus until one node gets an ACK this process will continue.

The protocol can be represented using the model in Fig. 5.3. Circles with enclosed numbers are states. Transition probabilities are shown along the arrows. The duration spend in state 1 is 1, and for others states is t_A as indicated. In the analysis, state transitions will be represented as a pair such as (g, h) for a transition from state g to h. State transition probability will be represented as $P(g, h)$.

The start of DATA contention cycle is at state 1. The probability of a node sending a DATA at the start of a new slot is modelled as $P(1, 2) = a = 2/(W + 1)$.

This is because the expected value of the uniformly distributed contention window is $W' = (W+1)/2$, and that is used as the expected value of a geometric process for transition (1,2) to satisfy Markov Chain requirements. Once a DATA is sent, node is in state 2, waiting for t_A time slots for ACK to arrive. If ACK arrives, the process terminates.

The probability that the DATA transmitted in a given slot has no collision from any other node is $(1 - 1/W')^{N-1}$, i.e., no other node transmits a DATA in that slot. ACK will be successfully received if apart from having no collisions, DATA is received at the receiver (probability k) and the ACK in turn is received at the transmitter (probability k) with a combined probability of k^2. This is shown in Fig. 5.3 as $P(2,4) = f = k^2 \left(1 - \frac{1}{W'}\right)^{N-1}$. If ACK is not successfully received, transition (2,1) happens as shown with probability $z = 1 - f$.

If a DATA is not sent (probability $1 - 1/W'$), the current node counts down the DATA timer by one slot. During this back-off period, the probability that one of the $N-1$ neighbours has a successful DATA transmission is $y = (N-1)(1/W')(1-1/W')^{N-2}$ using same arguments as in last paragraph. And k being the DATA detection probability, the current node could receive a DATA from another node with probability ky . Thus, the transition (1,3) with $P(1,3) = b = \left(1 - \frac{1}{W'}\right) ky$ occurs as shown.

In state 3, it awaits ACK for time t_A. Thereafter it goes back to state 1 with probability 1.

If system is backing off and either DATA or ACK from others is not received as stated above, it goes back to state 1 as shown with $P(1,1) = c = 1 - a - b$.

A Markov matrix $\mathbf{M}$ (Gross and Harris, 1998) represents this as follows using $P(a, b)$ as shown in Fig. 5.3. $\mathbf{Q}$ is the transient state matrix.

$$\mathbf{M} = \begin{bmatrix} c & a & b & 0 \\ z & 0 & 0 & f \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \mathbf{Q} = \begin{bmatrix} c & a & b \\ z & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix} \tag{B.3}$$

The fundamental matrix $\mathbf{F}$ (Gross and Harris, 1998) is the $\mathbf{F} = (\mathbf{I} - \mathbf{Q})^{-1}$. Let $E(m, n)$ be the expected number of times the system is in state n after starting from state m. $E(1, n)$ is the expected number times the state n will be visited if the chain starts in state 1. Using standard Markov Chain theory (Gross and Harris, 1998):

$$\begin{aligned} E(1,n) = F_{1,n}; \quad E(1,1) &= \frac{1}{k^2} \frac{W'}{\left(\frac{W'-1}{W'}\right)^{N-1}} \\ E(1,2) = \frac{1}{k^2} \frac{1}{\left(\frac{W'-1}{W'}\right)^{N-1}}; \quad E(1,3) &= \frac{N-1}{k} \end{aligned} \tag{B.4}$$

Let the time till successful reception of ACK from state 1 to state 4 of Fig. 5.3 be s_p. This gives,

$$s_p = (l)E(1,1) + t_A E(1,2) + t_A E(1,3) \tag{B.5}$$

Using (B.4), s_p can be simplified as

$$s_p = \left(\frac{lW' + t_A}{k^2}\right) \left(\frac{W'}{W' - 1}\right)^{(N-1)} + \frac{t_A(N-1)}{k} \tag{B.6}$$

B.2 M/D^B/1 Waiting Time Analysis for MACA-C and TDMA

The characteristic equation for the M/D^B/1 system is given by (Chaudhry and Templeton, 1984)

$$z^B e^{B\rho(1-z)} - 1 = 0 \tag{B.7}$$

where $\rho = \lambda/\mu$. The roots z_i of the above equation can be found through a numerical technique (Janssen and Leeuwaarden, 2005). The total system size is (Chaudhry and Templeton, 1984)

$$L = \frac{1 - B(1-\rho)^2}{2(1-\rho)} + \sum_{1}^{B-1}(1 - z_i)^{-1}. \tag{B.8}$$

Using Little's Law (Gross and Harris, 1998), the expected waiting time W_Q (excluding service) can be found as follows (s_b from (5.6) for TDMA and from (5.4) for MACA-C)

$$W_Q = \frac{L_T}{\lambda} - s_b. \tag{B.9}$$

Total waiting time W_T (the sum of queuing time W_Q and service time of one packet s_p (from (5.7) for TDMA and from (4.14) for MACA-C) is then

$$W_T = W_Q + s_p. \tag{B.10}$$

B.3 Inter-cell or inter-MC interference

When there are multiple cells and controlling MCs in a neighbourhood using MACA-C, adjacent cell interference could take place. In scenario 1, as depicted

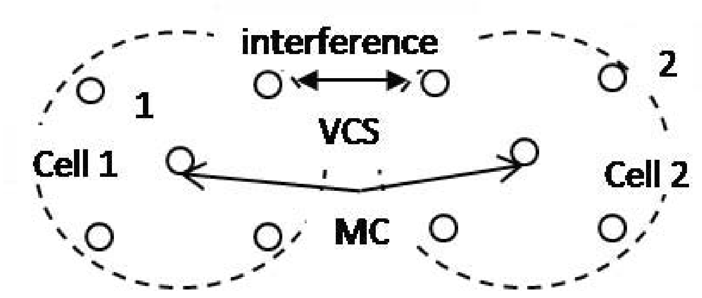


Figure B.1: Neighbouring cells Scenario 1

in Fig. B.1, the MCs are not able to hear each other or neighbouring cell nodes' transmissions. It is assumed here that all nodes only have single hop range to reach the MC using power control.

So it will continue to use MACA-C. But as can be seen, nodes from adjacent cells interfere with each other's transmission. They observe VCS as mentioned in section 5.2.3 and will not reply to MC's RTR. MC will proceed to do communications between nodes such as 1 and 2 away from such interference. Thus parallel communications takes place in neighbouring cells correctly using MACA-C in the face of inter-cell interference that does not involve the MCs.

If MCs discover that their cells are near enough to cause interference, option A is to give up RTR based control and let nodes revert to MACA-EA. Option B for MCs is to continue use MACA-C with a back-off for RTR, just as in RTS back-off in MACA-EA. The neighbour MCs can hear either the RTRs or the reply RTS, INFO or DATA packets. MCs obey VCS rules for allowing neighbours to complete one communication sequence (till ACK). In option B, the RTR contention will only be between MCs of neighbouring cells unlike in RTS contention involving all nodes in option A. It's possible that since the optimum contention window is directly proportional to participating neighbours (Bianchi, 2000), the MC based

RTR contention needs a shorter contention window and the effective contention period could be lower. Since under normal circumstances, MACA-C gives better performance than MACA-EA and RTR back-off based method could solve the problem of neighbouring cell interference, it is proposed that MCs do not relinquish their roles in favour of MACA-EA upon discovery of inter-cell interference, and instead use RTR back-off (option B). This idea needs to be further validated by simulations as part of future research.

Appendix C

Super TDMA

Note: This is reproduced with permission from a paper "Throughput of Wireless Networks with Large Propagation Delays", under review in IEEE Transactions on Networking, 2010, by M. Chitre, M. Motani, and S. Shahabudeen.

C.1 Prototype Schedule for Odd-even Distance Networks

$$\chi^{(2)} = \begin{bmatrix} 2 & -2 \\ 1 & -1 \\ 4 & -4 \\ 3 & -3 \\ 6 & -6 \\ 5 & -5 \\ \vdots & \vdots \\ N & -N \\ N-1 & -(N-1) \end{bmatrix} \tag{C.1}$$

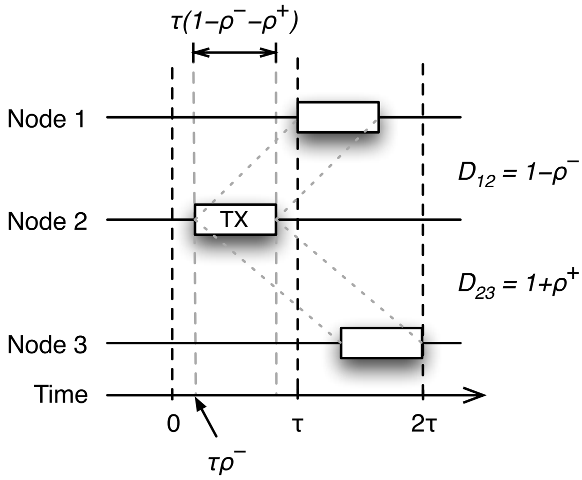


Figure C.1: An illustration of a ρ-schedule

C.2 ρ-schedule

For a network with a non-integer delay matrix, messages transmitted on time slot boundaries may be received across time slot boundaries. If the length of the message is equal to the time slot length, the message reception will span multiple time slots. For a non-integer delay matrix $\mathbf{D}$ (Chitre et al., 2010), we can round off the entries in the delay matrix to yield an integer delay matrix $\mathbf{D'}$ and define ρ^+ and ρ^- such that

$$\rho^+ = \max_{ij}(D_{ij} - D'_{ij}) \tag{C.2}$$

$$\rho^- = -\min_{ij}(D_{ij} - D'_{ij}) \tag{C.3}$$

and $\rho^+, \rho^- \leq 0.5$. If we limit the duration of each transmitted message to $\tau(1 - \rho^- - \rho^+)$ and transmit the message at time $\tau\rho^-$ after the start of the time slot, then we ensure messages are always received fully during a time slot as seen in Fig. C.1. We call a schedule with shortened messages of length $\mu = \tau(1 - \rho^- - \rho^+)$ a fraction time-slot schedule or a *ρ-schedule*.

Appendix D

UNA, FAPI, DLL Utilities and Sample MAC Code

D.1 UNA Messages

This is a short summary of UNA inter-layer messages (Chitre et al., 2006). Parameter SET and GET messages etc are omitted for brevity. Also there are general error responses: ERROR_RSP.

D.1.1 Key physical layer messages

Send a packet	PHY_SEND_PKT_REQ
Packet Transmitted Response	PHY_PKT_XMIT_RSP
Notification to DLL for incoming packet	PHY_INCOMING_PKT_NTF

D.1.2 Key data link layer messages

Send a packet	DLL_SEND_PKT_REQ
Notification to network layer for incoming packet	DLL_INCOMING_PKT_NTF
Packet Transmitted Response	DLL_PKT_XMIT_RSP

D.2 Framework API

Here some of the key API of FAPI is shown. There were some changes to the proposal by Chitre et al. (2006). In the rest of the appendix, C code or related text will be presented in text boxes for clarity. All layers implement the following message handler to receive messages from other layers.

```
/* msg: the incoming message, sender: sending layer, hX:
handle to an arbitrary object to be used by the caller
(used my multi-node simulations in the simulator) */
void (*MessageHandler)(Message* msg, int sender, UserData
hX);
```

The API for sending messages is as follows

```
/* msg is the message, me is the sender layer , destination
is the destination layer, hX is as decribed above */
int FAPI_sendMessage(void* msg, int me, int destination,
UserData hX);
```

Timer set functionality

```
/* ref is a reference for the caller to identify the timer,
timeout is the timeout in millisecs */
int FAPI_setTimer(int me, unsigned int ref, long timeout,
UserData hX);
```

Timer cancellation

```
int FAPI_cancelTimer(int me, unsigned int ref, void *hX);
```

To get the current time in milliseconds

```
long FAPI_getTime(UserData hX);
```

D.3 Data link layer Utility Functions

Here we summarize some of the important utility functions provided by the data link layer to the MAC sub-function. We omit parameter set and get utility function etc for brevity. These DLL utility functions isolate the MAC implementation from complexities such as HPA on/off control etc.

D.3.1 Reception handling

The following function switches reception off.

```
/* pStateDLL is the state information storage of the MAC
module  described further in the sample MAC code section
later on, hX is as decribed above */
void switchOffRx(stateDLL *pStateDLL, void *hX);
```

The following function switches reception on.

```
void switchOnRx(stateDLL *pStateDLL, void *hX);
```

D.3.2 HPA control

Though FAPI_sendMessage() function (see section D.2) can be used directly to send a packet to the physical layer for transmission (using UNA message PHY-SEND-PKT-REQ defined in section D.1.1), there are complexities such as HPA on/off control etc that needs to be controlled directly by the DLL. In order to isolate MAC module from such complexities, the MAC module uses the following function to transmit a packet.

```
/* mode set to 1 if HPA needs to be kept on after
transmission, for example in batch transmission.  Else
HPA will be turned off after transmission and reception
switched on */
```

```
int txPacket(stateDLL *pStateDLL, void *msg, int mode, void
*hX);
```

The above reception or transmission handling requires interaction with the physical layer. Hence response events will come from physical layer and needs to be handled, and that is done by the following function. Section D.4 on sample MAC code shows how this is done.

```
int Tx_Rx_Handler(stateDLL *pStateDLL, Message *msg, void
*hX);
```

D.3.3 PDU handlers

To create a Data link PDU, MAC utilizes the following function. This isolates the MAC protocol from the packet byte structure. The PDU length must be within the modem's allowed maximum, which is related to the level of FEC and other link parameters.

```
/* type: type of payload such as RTS or CTS, UID: unique
identifier, src and dest: node addresses, param1: an
arbitrary parameter, data: MAC data and datalen: length
of data in bytes */
DatalinkPDU *generatePDU(BYTE type, int UID, int src, int
dest, BYTE param1, BYTE *data, int datalen);
```

There are other PDU related utility functions to get and set fields, duplicate PDUs etc, which are omitted from this appendix for brevity. It's not essential for the following illustration of how to write a typical MAC module in this framework.

D.4 Sample MAC Code

Here we present a sample DLL message handler to give a flavour of MAC layer implementation using the framework. The above FAPI API and data link utility functions are used to write a typical MAC module. Here we illustrate a trivial ALOHA like simple MAC protocol. The code is not meant to compile and meant for illustration only.

D.4.1 The main handler interface

This is the main DLL message handler. All events for the data link layer come here. pStateDLL input parameter, is used to pass in the current state information of the DLL. During processing, the state can be updated and stored until next event. As mentioned earlier (section D.3.2), transmission and reception handling will generate responses from the physical layer. All events are handled first by the Tx_Rx_Handler() function as shown below to capture such responses, before an event is passed to the actual MAC code.

```
/* pStateDLL is the state information storage of the MAC
module.  Other parameters as defined earlier */
void DLL_handleMessage (stateDLL *pStateDLL, Message* msg,
int sender, void *hX) {
 //input checks
 if (msg != NULL) {
          /* handles tx/rx/HPA related behavior before doing
processing */
  if (Tx_Rx_Handler(pStateDLL, msg, hX) < 0) {
   //insert << MAIN_MAC_HANDLER >>
  }
 }
}
```

D.4.2 MAIN_MAC_HANDLER

This is the main MAC handlers overall structure (this fits into the label MAIN-MAC-HANDLER above) which handles messages from higher and lower layers as well as timers and takes appropriate action.

```
if ((sender == LAYER_NETWORK) || (sender ==
LAYER_APPLICATION)) {
 //Note the sender layer for future use
 pStateDLL->senderLayer = sender;
  if (msg->msgid == DLL_SEND_PKT_REQ) {
   //insert << DLL_SEND_PKT_REQ >>
  }
 }
 else if (sender == LAYER_PHYSICAL) {
  if (msg->msgtype == MSGTYPE_RSP) {
   if (msg->msgid == ERROR_RSP) {
     /*Handle the error as appropriate For example this could
be in response to transmission request */
   }
   else if (msg->msgid == PHY_PKT_XMIT_RSP) {
  //insert << PHY_PKT_XMIT_RSP >>
   }
  }
 else if (msg->msgtype == MSGTYPE_NTF) {
  if (msg->msgid == PHY_INCOMING_PKT_NTF) {
  //insert << PHY_INCOMING_PKT_NTF >>
   }
  }
 }
 else if (msg->msgid == FAPI_TIMER_EXPIRED_NTF) {
  //insert << FAPI_TIMER_EXPIRED_NTF >>
 }
```

D.4.3 DLL_SEND_PKT_REQ

A new packet send request from higher layer comes in here. The actions will depend on the protocol, and in this example we illustrate a random back-off before transmission.

```
DllSendPktReq* dllmsg;
dllmsg = (DllSendPktReq*)msg;
//cancel previous back-off timer
FAPI_cancelTimer(LAYER_DATALINK, ref_BACKOFF_TIMER,hX);
/*remove previous message.  No queuing in this example*/
if (pStateDLL->DATAmsg != NULL) free(pStateDLL->DATAmsg);
/* pStateDLL->DATAmsg is a pointer to type DatalinkPDU */
pStateDLL->DATAmsg = generatePDU(_DATA, 0,
pStateDLL->ownMAC, dllmsg->dstaddr,0, NULL,
SIZE_DATALINK_PDU-DatalinkHeaderLen) ;
//Do a back off
FAPI_setTimer(LAYER_DATALINK, ref_BACKOFF_TIMER,
(long)(wait), hX);
```

D.4.4 FAPI_TIMER_EXPIRED_NTF

In this example, when the timer expires, we use the txPacket() function to transmit the stored packet. We do not need to handle HPA control etc explicitly. After this action we wait for either PHY_PKT_XMIT_RSP or ERROR_RSP from physical layer.

```
if (((FapiTimerExpiredNtf *)msg)->ref == ref_BACKOFF_TIMER)
{
             /*transmit packet and switch on reception after
transmission*/
 txPacket(pStateDLL, phyreq, 0, hX);
}
```

D.4.5 PHY_PKT_XMIT_RSP

Here we handle transmission success responses from physical layer. In this example, we pass a DLL_PKT_XMIT_RSP back to the higher layer and the MAC protocol does a random back off before transmission, i.e. the packet is transmitted only when the backoff timer expires. Some transmission requests could meet with error responses which are handled as indicated earlier.

```
//inform higher layer
rsp = FAPI_createMessage(MSGTYPE_RSP, DLL_PKT_XMIT_RSP,
SIZE_MESSAGE);
FAPI_sendMessage(rsp, LAYER_DATALINK,
pStateDLL->senderLayer, hX);
```

D.4.6 PHY_INCOMING_PKT_NTF

Packets from other nodes will be handled here. First we extract the PDU from PHY_INCOMING_PKT_NTF UNA message. In this example we just pass up the incoming packet to the higher layer. In MAC protocols with control packets such as RTS, CTS, appropriate next steps will be taken here.

```
/* This MAC utility function extracts the pdu contained in
the Message struct.  This function is written by the MAC
developer and not part of the DLL utility framework */
pdu = checkPacket(pStateDLL, msg, check, 0);
if (pdu != NULL) {
 //send incoming message to the higher layer
ntf = FAPI_createMessage(MSGTYPE_NTF,
DLL_INCOMING_PKT_NTF, SIZE_MESSAGE);
   FAPI_sendMessage(ntf, LAYER_DATALINK, LAYER_NETWORK,
hX);
  free(pdu);
}
```

This Appendix is meant to provide an overview of how the framework is used to implement a MAC protocol that seamlessly works in a simulator and modem.

Appendix E

Publications From This Thesis

[1] S. Shahabudeen, M. Chitre, and M. Motani, "A multi-channel MAC protocol for AUV networks," in *IEEE Oceans' 07*, Aberdeen, Scotland, 2007.

[2] M. Chitre, S. Shahabudeen, and M. Stojanovic, "Underwater acoustic communications and networking: Recent advances and future challenges," *Marine Technology Society Journal - "The State of Technology in 2008"*, vol. 42, no. 1, 2008.

[3] S. Shahabudeen, M. Chitre, J. Potter, and M. Motani, "Multi-mode adaptive MAC protocol suite and standardization proposal for heterogeneous underwater acoustic networks," in *Underwater Acoustic Measurements*, Nafplion, Greece, 2009.

[4] S. Shahabudeen, M. Chitre, M. Motani, and A. Low, "Unified simulation and implementation software framework for underwater MAC protocol develop-

ment," in *MTS/IEEE Oceans'09*, Biloxi, USA, Oct 26-30 2009.

[5] S. Shahabudeen and M. Motani, "Modeling and performance analysis of MACA based protocols for adhoc underwater networks," in *WUWNet'09*, Berkeley, USA, Nov 3 2009.

[6] S. Shahabudeen, M. Chitre, and M. Motani, *Underwater Acoustic Sensor Networks.* CRC Press, 2010, ch. Dynamic TDMA and MACA based Protocols for Distributed Topology Underwater Acoustic Networks.

[7] S. Shahabudeen, S. Motani, and M. Chitre, "Analysis of a high performance MAC protocol for underwater acoustic networks," *Under review, Journal of Oceanic Engineering*, 2011.

[8] S. Shahabudeen, M. Chitre, and M. Motani, "MAC protocols that exploit propagation delay in underwater networks," in *MTS/IEEE Oceans'11*, Kona, Hawaii, USA, Sep 19-22 2011.

[9] M. Chitre, M. Motani, and S. Shahabudeen, "Throughput of wireless networks with large propagation delays," *Under review, Transactions on Networking*, 2011.

[10] S. Shahabudeen, M. Chitre, and M. Motani, "Adaptive multi-mode medium access control for underwater acoustic networks," *Submitted to Journal of Oceanic Engineering*, 2011.