Adaptive Mac protocols for wireless sensor networks

Zhao, Yizhi

2012

Zhao, Y. (2012). Adaptive Mac protocols for wireless sensor networks. Doctoral thesis, Nanyang Technological University, Singapore.

<https://hdl.handle.net/10356/50780>

<https://doi.org/10.32657/10356/50780>

NANYANG
TECHNOLOGICAL
UNIVERSITY

**ADAPTIVE MAC PROTOCOLS
FOR WIRELESS SENSOR NETWORKS**

YIZHI ZHAO

SCHOOL OF COMPUTER ENGINEERING

2012

**ADAPTIVE MAC PROTOCOLS
FOR WIRELESS SENSOR NETWORKS**

YIZHI ZHAO

YIZHI ZHAO

School of Computer Engineering

A thesis submitted to the Nanyang Technological University
in partial fulfilment of the requirement for the degree of
Doctor of Philosophy

2012

DEDICATION

To my family, near and far...

With Love...

Acknowledgements

During my study towards the Ph.D. degree in the School of Computer Engineering at the Nanyang Technological University (NTU), I have received tremendous encouragements, help and support from professors, friends and my family, without whom the successful completion of this dissertation would be impossible. I would like to take this opportunity to express my heartfelt thanks to them.

First, I would like to express my sincere gratitude to my supervisor Associate Prof. Miao Chunyan for her valuable advice and constant encouragement. Her initial encouragement invoked my courage of pursuing the Ph.D. degree. I have benefited a lot from her comprehensive research methodology and rich research experience. Thanks for her giving me the freedom to explore. Her guidance in viewing the research issues from different angles raises my graduate study to a Ph.D. level.

I am deeply grateful to Associate Prof. Ma Maode for his constant guidance, valuable suggestions and insightful opinions throughout my Ph.D. study. The graduate course “Wireless Networking” taught by Associate Prof. Ma is a starting point leading me to the research area of Wireless Sensor Networks. The contents taught in the course and a series of study on the tutorial and research papers laid the foundation of my research direction. I had a series of discussions with Associate Prof. Ma and received valuable suggestions on the research direction and insights into the experimental results, which are very helpful to my research paper publication and towards final formation of the thesis.

I must sincerely express my gratitude to Dr. Zhang Jing Bing for his generous support and stimulating suggestions in the preliminary stage of my research work. The discussions feedbacks and insightful opinions from Dr. Zhang firmly provided valuable inputs to form up the projection on the future researches.

My thanks also go to Dr. Shen Zhiqi for providing me generous opinions on my research work and kind assistant in papers publication. Dr. Huang Dong, as Associate Prof. Miao’s another student and a friend of mine, provided me open and sincere opinions on my research direction and approach, which was a good experience sharing and worthy of my sincere appreciation. I would also like to thank Jessie Nguyen, Chen

You, Li Shiwei, Henry Ng and other NTU students for allowing me to supervise them to carry out their final year projects and industry attachment projects in the research areas that are very close to my Ph.D. research work. I had lots of discussions with them on algorithms design, implementation and simulation results, which I think are beneficial to my Ph.D. research results.

I am indebted to my father Zhao Qi and my mother Cui Gongrong. They gave me a healthy body and mind full of curiosity. These are my most valuable assets. Throughout my life, my father has always encouraged me to pursue the scholarly highest achievements; and unconditionally parenting and love from parents are always motivating me to move forward to the higher professional target. My deepest gratitude goes to my wife, Wang Jian, for her affectionate understanding, patience and support throughout my Ph.D. study. Except for her own work, she spent a great deal of her time taking care of our son academically and in life, which allows me to focus on my Ph.D. study. I would also like to thank my son, Chunkai, for being a good boy during my busiest time. My sisters and parents-in-law also gave me lots of spirit support and encouragement. I dedicate this dissertation to all of them.

My Ph.D. study was carried out under a Scientific Staffs Development Scheme (SSDS) of the Agency for Science, Technology and Research (A*STAR) of Singapore, and sponsored by Singapore Institute of Manufacturing Technology (SIMTech) of A*STAR. I sincerely appreciate the sponsorship and all the support provided by SIMTech of A*STAR. The views and conclusions contained here are those of the author and should not be interpreted as necessarily representing the official policies or endorsement, either expressed or implied, of sponsor.

Table of Contents

Dedication	II
Acknowledgements	III
Table of Contents	V
List of Figures.....	XI
List of Tables	XV
List of Acronyms	XVI
Abstract	XVIII
1 INTRODUCTION	1
1.1 SENSOR NETWORKS	1
1.2 POTENTIALS AND APPLICATIONS OF SENSOR NETWORKS	1
1.3 THE NEED FOR ADAPTABLE AND APPLICATION-AWARE WIRELESS SENSOR NETWORKS.....	4
1.4 ADAPTIVE MAC PROTOCOLS FOR APPLICATION-AWARE WSNS.....	5
1.5 THESIS STATEMENT.....	6
1.6 THESIS CONTRIBUTIONS.....	7
1.6.1 Adaptive Scheduling MAC Protocol – AS-MAC.....	7
1.6.2 Self-adaptive MAC Protocol – SEA-MAC.....	8
1.6.3 Characterization and Rendering of Application-Specific WSN.....	9
1.6.4 Projections on Future Research of MAC Scheme	9
1.7 THESIS ORGANIZATION	10
2 RESEARCH ISSUES IN WSNS	13
2.1 CHARACTERISTICS AND CHALLENGES OF WIRELESS SENSOR NETWORKS	13
2.1.1 Constrained Resource Challenge: Energy-Efficiency.....	13
2.1.2 Large Scale Challenge: Scalability and Robustness	13
2.1.3 Data Centric Network Challenge: Data-centric Routing and in-network Processing	14
2.1.4 Ad hoc Network Challenge: Network Discovery and Self-organization Capability.....	14
2.1.5 Node Failure Challenge: Fault Tolerance	14

2.2	RESEARCH ISSUES IN MAC LAYER PROTOCOL	14
2.2.1	<i>Protocols Stack – A Research Foundation</i>	14
2.2.2	<i>Wireless MAC Issues</i>	16
2.2.3	<i>Focused Research Topics</i>	20
2.3	RESEARCH METHODOLOGIES	21
2.3.1	<i>Development Environment</i>	21
2.3.2	<i>Benchmarking Evaluation</i>	22
3	RELATED WORK	25
3.1	CLASSIFICATION AND ANALYSIS OF WSN MAC PROTOCOLS.....	25
3.2	CONTENTION-BASED MAC PROTOCOLS	27
3.2.1	<i>PAMAS</i>	28
3.2.2	<i>S-MAC and S-MAC-AL</i>	29
3.2.3	<i>AC-MAC</i>	32
3.2.4	<i>T-MAC</i>	33
3.2.5	<i>TEA-MAC</i>	35
3.2.6	<i>asTEA-MAC</i>	37
3.2.7	<i>Sift and CSMA/p*</i>	38
3.2.8	<i>Summary of Contention-Based MAC Protocols</i>	39
3.3	SCHEDULING-BASED MAC PROTOCOLS	41
3.3.1	<i>TRAMA</i>	42
3.3.2	<i>FLAMA</i>	44
3.3.3	<i>SMACS</i>	46
3.3.4	<i>SRSA</i>	47
3.3.5	<i>R-MAC</i>	47
3.3.6	<i>RMAC</i>	48
3.3.7	<i>DW-MAC</i>	49
3.3.8	<i>Summary of Schedule-Based MAC Protocols</i>	51
3.4	HYBRID MAC PROTOCOLS	52

3.4.1	<i>Z-MAC</i>	53
3.4.2	<i>HyMAC</i>	54
3.4.3	<i>Funnelling-MAC</i>	55
3.4.4	<i>Crankshaft</i>	56
3.4.5	<i>Summary of Hybrid MAC Protocols</i>	57
3.5	PREAMBLE SAMPLING BASED MAC PROTOCOLS	58
3.5.1	<i>WiseMAC</i>	58
3.5.2	<i>CSMA-MPS</i>	60
3.5.3	<i>X-MAC</i>	61
3.5.4	<i>B-MAC</i>	61
3.5.5	<i>SCP-MAC</i>	62
3.5.6	<i>Summary of Preamble Sampling MAC Protocols</i>	63
3.6	CHALLENGES TO MAC TECHNOLOGY FOR WSNS	64
3.6.1	<i>Adaptability</i>	64
3.6.2	<i>Resource-Efficiency through Cross-Layer Design</i>	65
3.6.3	<i>Decentralized</i>	65
3.6.4	<i>Real-time and Quality-of-Service-aware</i>	65
3.6.5	<i>Balance among Multiple Metrics</i>	66
3.7	SUMMARY	66
4	AN ENERGY-EFFICIENT AND LOW-LATENCY MAC PROTOCOL WITH ADAPTIVE-SCHEDULING FOR MULTI-HOP WSNS	67
4.1	DIRECT MOTIVATION OF PROPOSED SCHEME	67
4.2	EXECUTIVE SUMMARY OF PROPOSED SCHEME	68
4.3	DIRECT RELATED WORK	69
4.4	SYSTEM BACKGROUND	70
4.5	THE PROPOSED AS-MAC SCHEME	71
4.5.1	<i>DW-MAC Scheme Operation Procedure and Highlights</i>	71
4.5.2	<i>Our Proposed AS-MAC Scheme</i>	73

4.5.3	<i>Enhancement of AS-MAC for Broadcast Traffic</i>	83
4.5.4	<i>Implementation Complexity and Cost</i>	87
4.6	PERFORMANCE EVALUATION	87
4.6.1	<i>Performance Evaluation under Unicast Traffic</i>	87
4.6.2	<i>Performance Evaluation under Broadcast Traffic</i>	93
4.7	SUMMARY	97
5	PERFORMANCE EVALUATION OF AS-MAC PROTOCOL WITH DIFFERENT AS-PERIOD IN MULTI-HOP NETWORKS	99
5.1	PROTOCOL CONFIGURATION	99
5.2	SIMULATION RESULTS AND DISCUSSIONS	100
5.2.1	<i>Energy Efficiency</i>	100
5.2.2	<i>Packet Delivery Ratio (PDR)</i>	103
5.2.3	<i>End-to-End Delay</i>	106
5.3	SUMMARY	112
6	PERFORMANCE EVALUATION OF AS-MAC PROTOCOL WITH DIFFERENT TIMEOUT IN MULTI-HOP NETWORKS	113
6.1	DUTY CYCLE AND NETWORK PARAMETERS CONFIGURATION	113
6.2	SIMULATION RESULTS AND DISCUSSIONS	114
6.2.1	<i>Energy Efficiency</i>	114
6.2.2	<i>Packet Delivery Ratio (PDR)</i>	116
6.2.3	<i>End-to-End Delay</i>	120
7	AN ENERGY-EFFICIENT AND SELF-ADAPTIVE DUTY CYCLE MAC PROTOCOL FOR TRAFFIC DYNAMIC WSNS	125
7.1	MOTIVATION OF PROPOSED SCHEME	125
7.2	EXECUTIVE SUMMARY OF PROPOSED SCHEME	125
7.3	DIRECT RELATED WORK	126
7.3.1	<i>Related Work to Basic Scheme</i>	126
7.3.2	<i>Related Work to Self-Adaptive Duty Cycle</i>	127

7.4	SYSTEM BACKGROUND	128
7.5	THE PROPOSED SEA-MAC SCHEME.....	129
7.5.1	<i>Dynamically Altering Operation Cycle Time</i>	<i>130</i>
7.5.2	<i>Duty Cycle Efficiency based Duty Cycle Turning</i>	<i>132</i>
7.5.3	<i>Implementation Complexity and Cost.....</i>	<i>136</i>
7.6	PERFORMANCE EVALUATION.....	136
7.6.1	<i>SEA-MAC Performance Evaluation and Comparison with AS-MAC</i>	<i>136</i>
7.6.2	<i>SEA-MAC Performance Evaluation and Comparison with RI-MAC.....</i>	<i>143</i>
7.7	SUMMARY	149
8	CHARACTERIZATION AND RENDERING OF APPLICATION-SPECIFIC WSNS	151
8.1	TRAFFIC-AWARE CHARACTERIZATION AND APPLICATION-AWARE RENDERING	151
8.1.1	<i>Traffic-aware Characterization.....</i>	<i>152</i>
8.1.2	<i>Application-aware Rendering</i>	<i>153</i>
8.2	RENDERING REALIZATION OF TRAFFIC-AWARENESS AND REQUIREMENT-AWARENESS	154
8.2.1	<i>Rendering Realization of Application Traffic-Awareness</i>	<i>154</i>
8.2.2	<i>Rendering Realization of Application Requirement-Awareness.....</i>	<i>155</i>
8.3	DEPLOYMENT OF AS-MAC OVER TINYOS/NESC	157
8.3.1	<i>Migration of AS-MAC from OMNet++ to TinyOS/nesc.....</i>	<i>158</i>
8.3.2	<i>Limitation of Physical Capability of Wireless Sensors</i>	<i>159</i>
9	CONCLUSIONS.....	161
9.1	SUMMARY OF CONTRIBUTIONS	161
9.2	PROJECTIONS ON FUTURE RESEARCH OF MAC PROTOCOLS FOR WSNS	162
9.2.1	<i>Hybrid and Self-adaptable Protocols Are Needed</i>	<i>162</i>
9.2.2	<i>Cross-Layer Interaction and Design Are Needed</i>	<i>166</i>
9.3	CONCLUDING REMARKS.....	168

APPENDIX A: LOGIC FLOWCHARTS OF AS-MAC IN TINYOS/NESC	169
APPENDIX B: TIME INTERFACE	171
AUTHOR’S PUBLICATIONS	172
BIBLIOGRAPHY	173

List of Figures

Figure 2-1: The protocols stack and research issues of a WSN	15
Figure 2-2: Location-dependent sensing: hidden nodes, exposed nodes, and capture.	17
Figure 3-1: IEEE 802.11 frame	28
Figure 3-2: S-MAC basic scheme	30
Figure 3-3: Adaptive Listening in S-MAC-AL	31
Figure 3-4: Who should sleep when node A is transmitting to B?.....	31
Figure 3-5: The periodic listen/sleep cycle for AC-MAC. For reason of possible high traffic, AC-MAC provides 2 chances of communication within corresponding one cycle interval of S-MAC.	32
Figure 3-6: The basic T-MAC protocol scheme, with adaptive active times.....	34
Figure 3-7: IEEE 802.11 frame	35
Figure 3-8: Sand-clock (Intra a node) scheme: If a sensor has no activities when awake, its timer decrements wake counter while incrementing the sleep counter.....	36
Figure 3-9: wake-time sharing (Inter-nodes) scheme.....	37
Figure 3-10: Counters are updated in communication	37
Figure 3-11: Sleep-time adjustment (intra-node) scheme	38
Figure 3-12: Sleep counter is adjusted before sleep.....	38
Figure 3-13: RMAC Overview	49
Figure 3-14: DW-MAC Overview	50
Figure 3-15: Funneling effect in sensor networks [54]	55
Figure 3-16: Contention and message exchange in the Crankshaft protocol [101].....	57
Figure 3-17: WiseMAC scheme.....	60
Figure 3-18: Sender and receiver synchronization schemes [32].....	63
Figure 3-19: Adaptive channel polling and multi-hop streaming [32].....	63
Figure 4-1: One-to-one mapping function in DW-MAC	72
Figure 4-2: Scheduling frame exchange (a) CSMA/CA, (b) DW-MAC.....	72

Figure 4-3: Adaptive change of resilient active time	74
Figure 4-4: Forwarding SCH (a) DW-MAC, (b) AS-MAC	76
Figure 4-5: DW-MAC one-to-one proportional mapping function with new notation	77
Figure 4-6: Condition for Collision-Free Data Transmission	77
Figure 4-7: Scheduling data transmission when no SCH (CTS) is received.....	80
Figure 4-8: Breadth First Search Algorithm.....	84
Figure 4-9: A broadcast tree based on the Breath First Search Algorithm	85
Figure 4-10: Optimized Multi-hop Forwarding under Broadcast	86
Figure 4-11: Average and Maximum End-to-End Delay(s) for EI=200s.....	90
Figure 4-12: Average and Maximum End-to-End Delay(s) of AS- MAC for EI=150s, 100s, 50s, 30s, and 20s	90
Figure 4-13: Average Energy Consumption under Unicast Traffic with EI=200s	92
Figure 4-14: Average Energy Consumption of AS-MAC under Unicast Traffic with EI=150s, 100s, 50s, 30s, and 20s	92
Figure 4-15: Packet Delivery Ratio under Unicast Traffic with EI=200s.....	93
Figure 4-16: Packet Delivery Ratio of AS-MAC under Unicast Traffic with EI=150s, 100s, 50s, 30s, and 20s	93
Figure 4-17: Average End-to-End Delay versus Network Size	95
Figure 4-18: Packet Delivery Ratio versus Network Size.....	95
Figure 4-19: Power Consumption versus Network Size	96
Figure 5-1: Average Power Consumption (mW) for Different EIs from 200s to 5s.....	101
Figure 5-2: Average Power Consumption Difference (mW)	101
Figure 5-3: Packet Delivery Ratio (PDR) (%) for Different EIs from 200s to 20s.....	103
Figure 5-4: Packet Delivery Ratio (PDR) (%) for Different EIs from 10s to 5s.....	103
Figure 5-5: Packet Delivery Ratio (PDR) Difference (%) (Tas800-Tas400).....	104
Figure 5-6: Packet Delivery Ratio (PDR) (%) for Different EIs from 200s to 20s.....	105
Figure 5-7: Packet Delivery Ratio (PDR) (%) for Different EIs from 10s to 5s.....	105
Figure 5-8: Packet Delivery Ratio (PDR) Difference (%) (Tas528-Tas400).....	105

Figure 5-9: Average End-to-End Delay of Different EIs	107
Figure 5-10: Average End-to-End Delay Difference of Different EIs (Tas800-Tas400).....	107
Figure 5-11: Max End-to-End Delay of Different EIs	108
Figure 5-12: Max End-to-End Delay Difference (s) (Tas800-Tas400).....	108
Figure 5-13: Average End-to-End Delay of Different EIs	109
Figure 5-14: Average End-to-End Delay Difference of Different EIs (Tas528-Tas400).....	109
Figure 5-15: Max End-to-End Delay of Different EIs	110
Figure 5-16: Max End-to-End Delay Difference (s) (Tas528-Tas400).....	110
Figure 6-1: Average Power Consumption (mW) for Different EIs from 200s to 5s.....	115
Figure 6-2: Average Power Consumption Difference (mW)	115
Figure 6-3: Packet Delivery Ratio (PDR) (%) for Different EIs from 200s to 20s.....	117
Figure 6-4: Packet Delivery Ratio (PDR) (%) for Different EIs from 10s to 5s.....	117
Figure 6-5: Packet Delivery Ratio (PDR) Difference (%) (T_{080ms} - T_{060ms}).....	117
Figure 6-6: Packet Delivery Ratio (PDR) (%) for Different EIs from 200s to 20s.....	118
Figure 6-7: Packet Delivery Ratio (PDR) (%) for Different EIs from 10s to 5s.....	119
Figure 6-8: Packet Delivery Ratio (PDR) Difference (%) (T_{060ms} - T_{045ms}).....	119
Figure 6-9: Average End-to-End Delay of Different EIs	120
Figure 6-10: Average End-to-End Delay Difference of Different EIs (T_{080ms} - T_{060ms}).....	120
Figure 6-11: Max End-to-End Delay for different EIs.....	121
Figure 6-12: Max End-to-End Delay Difference (s) (T_{080ms} - T_{060ms}).....	121
Figure 6-13: Average End-to-End Delay of Different EIs	122
Figure 6-14: Average End-to-End Delay Difference of Different EIs (T_{060ms} - T_{045ms})	123
Figure 6-15: Max End-to-End Delay for different EIs.....	123
Figure 6-16: Max End-to-End Delay Difference (s) (T_{060ms} - T_{045ms}).....	124
Figure 7-1: Three different Duty Cycle operations	135
Figure 7-2: Flow Chart of Duty Cycle Efficiency based Duty Cycle tuning	135
Figure 7-3: Outperformance of SEA-MAC over AS-MAC on average end-to-end delay.....	138

Figure 7-4: Performance comparison between SEA-MAC and AS-MAC on average power consumption per successful transmission	141
Figure 7-5: Performance comparison between SEA-MAC and AS-MAC on PDR for different EIs	142
Figure 7-6: Average and maximum end-to-end delay of RI-MAC and SEA-MAC with different EIs versus sensing ranges	145
Figure 7-7: End-to-end delay difference between RI-MAC and SEA-MAC versus EIs and sensing ranges	145
Figure 7-8: End-to-end delay improvement SEA-MAC over RI-MAC for EIs and sensing ranges	147
Figure 7-9: Average Duty Cycle of RI-MAC and SEA-MAC versus sensing ranges and EIs .	147
Figure 7-10: Average Duty Cycle Difference between RI-MAC and SEA-MAC (RI-SEA) versus EIs and sensing ranges	148
Figure 8-1: Traffic-aware characterization and application-aware rendering	151
Figure 8-2: Block level architecture of the programming concepts in TinyOS/nesc	158
Figure 9-1: Indicative performance of MAC protocol	164
Figure 9-2: Dominating performance metrics gain when dual-behaviour protocol switches between SB protocol and CB protocol	166
Figure A-1: Initialization of the AS-MAC	169
Figure A-2: Logic flow during the Sync period	169
Figure A-3: Logic flow during the AS-period	170
Figure A-4: Logic flow during the Sleep period	170

List of Tables

Table 2-1: Development Environment	22
Table 2-2: Individual Evaluations	23
Table 3-1: A Summary of Contention-based MAC Protocols for WSNs	40
Table 3-2: A Summary of Scheduling-based MAC Protocols for WSNs	51
Table 3-3: A Summary of Hybrid MAC Protocols for WSNs	57
Table 3-4: A Summary of Preamble Sampling MAC Protocols for WSNs	63
Table 4-1: SCH frame format	75
Table 4-2: Networking Parameters	88
Table 4-3: Operational Cycle Configuration	88
Table 4-4: Average Number of Packets Generated per Event (EI=200s)	89
Table 5-1: Duty Cycle Configuration	99
Table 5-2: Networking Parameters	99
Table 6-1: Duty Cycle Configuration	113
Table 6-2: Networking Parameters	113
Table 7-1: Operational cycle configuration	131
Table 7-2: Protocol parameters used in comparison with AS-MAC	137
Table 7-3: Average Performance Difference between End-to-End Delay of SEA-MAC and AS-MAC for Different Traffic Loads	139
Table 7-4: Protocol parameters used in comparison with RI-MAC	143
Table 8-1: Traffic load zones with varying Event Interval and sensing range	156
Table 8-2: Realization of the rendering application specific requirements for the different traffic loads in the AS-MAC protocol	156
Table B-1: TimeSyncService and Wiring to the TimeSyncService	171
Table B-2: TinyOS/nesc Compilation Messages	171

List of Acronyms

ACK	ACKnowledgegment
ADC	Analog-Digital Converter
APP Layer	APPLication Layer
AS-MAC	Adaptive Scheduling MAC
CDMA	Code Division Multiple Access
CDS	Connected Dominating Set
CLD	Cross-Layer Design
CPU	Central Processing Unit
CSMA/CA	Carrier Sense Multiple Access with Collision Avoidance
CSMA/CD	Carrier Sense Multiple Access with Collision Detection
CTS	Clear To Send
CW	Contention Window
DIFS	Distributed Inter-Frame Spacing
DLL Layer	Data Link Layer
DSP	Digital Signal Processing
EI	Event Interval
FDMA	Frequency Division Multiple Access
FRTS	Future Request-To-Send
HTL	High Traffic Load
LTL	Low Traffic Load
MEMS	Micro-Electro-Mechanical Systems
MAC	Medium Access Control
MANET	Mobile Ad hoc NETwork
NAV	Network Allocation Vector
NET Layer	NETwork Layer
NS-2	Network Simulation version 2
OMNet++	Simulation software OMNet++
QoS	Quality of Service
PDR	Packet Delivery Ratio
PHY Layer	PHYsical Layer
RAT	Resilient Active Time

RTS	Ready To Send (or Request To Send)
SEA-MAC	SElf Adaptive MAC
SIFS	Short Inter Frame Spacing
TCP	Transport Control Protocol
TDMA	Time Division Multiple Access
UDP	User Datagram Protocol
WLAN	Wireless Local Area Network
WSN	Wireless Sensor Network

Abstract

Recent advancements in wireless communications and sensor technologies have enabled the development of low-cost sensor networks for diverse applications, such as intelligent battlefields, earthquake response systems, military objects tracking, reconnaissance surveillance, hazardous environment exploration monitoring, industry monitoring, and smart hospitals, etc. Different applications have their own characteristics and different requirements for the wireless sensor networks (WSNs). A set of performance metrics, including energy-efficiency, self-organization ability, scalability, robustness, latency, throughput, fairness, etc., are not only used to measure the performance of protocols, but also the applicability of a wireless sensor network to a certain application. Adaptability and application-awareness of a wireless sensor network directly determines the applicability of a wireless sensor network. Due to the diversity of applications, the adaptability and application-awareness of a wireless sensor network are still open research issues.

Medium access control (MAC) layer protocols play a critical role in fulfilling the performance metrics of wireless sensor networks. In this dissertation, we propose to fulfill the adaptability and application-awareness of wireless sensor networks with energy-efficient adaptive MAC protocols as well as effective characterization and modelling of the application-specific requirements. Such characterization and modelling can be rendered to the adaptive MAC protocols, so that the wireless sensor networks enabled with the adaptive MAC protocols are able to be adaptable to application-specific requirements.

Firstly, we investigate characteristics of wireless sensor networks and identify related research challenges. We report the literature studies on the MAC layer protocols – an essential layer to enable the adaptability and application-awareness of wireless sensor networks. Secondly, we describe design principles and evaluate performance of two novel MAC protocols: An energy-efficient, low-latency and high packet delivery ratio Adaptive Scheduling MAC protocol for multi-hop wireless sensor networks – AS-MAC, and a Self-Adaptive and open duty cycle MAC protocol for multi-hop wireless sensor networks – SEA-MAC.

The AS-MAC introduces an Adaptive Scheduling (AS) period to a duty cycle to realize Resilient Active Time (RAT) for the dynamic traffic loads. The AS-MAC introduces dual-role SCH frame with tracked retry number, uses enhanced proportional mapping and adjustable timeout mechanism; and innovatively combines timeout-based Resilient Active Time and enhanced proportional mapping in the protocol design to realise waking up on demand during the Sleep period of an operation cycle to transmit and receive data packets and prevent data collision. The SEA-MAC further adjusts the duty cycle dynamically based on the duty cycle efficiency, so as to mitigate inefficient duty cycle problem in the circumstances of low traffic load, and reduce end-to-end delay in circumstances of high traffic load. The SEA-MAC is realized through rendering application-specific characteristic of traffic flow onto the AS-MAC. Experiment results show that the proposed protocols are a great advancement of some existing protocols on performance. For example, for high unicast traffic load, AS-MAC superior to DW-MAC (a Demand-Wakeup MAC protocol) in end-to-end delay, energy consumption, and keeps the same Packet Delivery Ratio (PDR) as DW-MAC. The SEA-MAC further outperforms AS-MAC and RI-MAC (a Receiver-Initiated MAC protocol) in end-to-end delay and duty cycle, and keeps PDR up to 100%. The AS-MAC and SEA-MAC are of adaptability to the dynamic traffic loads.

1 Introduction

In this chapter, we firstly provide a brief background on wireless sensor networks (WSNs) and list the potential applications of WSNs. Secondly we highlight the needs for adaptive and application-aware WSNs in the diverse applications, and pinpoint the focal point of this research. Finally we present our contributions and organization of the thesis.

1.1 Sensor Networks

Rapid advancements in sensor technologies have enabled the development of low-cost, low-power, multi-functional sensor nodes. Recent advancements in wireless communication technologies have enabled the collaboration of a large number of self-organized sensing nodes, hence forming a wireless sensor network.

A wireless sensor network is a collection of nodes organized into a cooperative network [84]. Each node consists of processing capability (one or more microcontrollers, CPUs or DSP chips), may contain multiple types of memory (program, data and flash memories), have a Radio Frequency (RF) transceiver (usually with a single omnidirectional antenna), have a power source (e.g., batteries and solar cells), and accommodate various sensors and actuators. The nodes communicate wirelessly and often self-organize after being deployed in an ad hoc fashion.

1.2 Potentials and Applications of Sensor Networks

Deployment of sensor networks can be standalone single sensor or massive networked sensors, depending on application requirements. The networked sensors deployment has great potential advantages over a standalone single sensor deployment. The following five aspects of advantages are highlighted: redundancy, complementarity, self-organization, timeliness and cost [86].

Redundancy: In general, neighboring sensors in a sensor network can capture the same set of data or partial set of full data of an environment with different fidelity. Thus the sensor network as a whole contains redundant data about the environment. However, due

to the energy constraints and application requirement, the redundant data among multiple sensors will be processed with data fusion and in-network processing techniques to reduce the overall uncertainty before presenting to application layer, thus improving the accuracy. On the other hand, the high level of redundancy allows better fault tolerance of the system. Though a dead node may lose the data it captured, its neighboring node(s) holding redundant data still can transmit the data to sink.

Complementarity: A sensor network employing multiple sensor types with multiple modalities and feature sensing capabilities can compensate the limitations of individual sensing modality and coverage. This complementary sensor network may give a representation of data over a large spatial area through the union of many small sensor coverage areas, or provide several aspects of the same phenomenon that can be used together for studying one event which is otherwise impossible to perceive using individual sensors [85], [86].

Self-location awareness and self-organization: A sensor network is an ad-hoc network. Its network discovery algorithms and relative positioning algorithms make its nodes be able to position itself in the network and form network structure automatically.

Timeliness: Since energy consumption used in data processing is only one tenth of energy consumption used in transmission/receiving the same amount of data, parallel data fusion algorithm and in-network processing technique can be used in processing the data from multiple sensors in order to achieve higher processing speed.

Cost: The advances in sensor technology and MEMS fabrication have contributed to better, cheaper, and smaller sensors, which in turn have made it economically feasible to deploy large numbers of sensors in sensor networks.

The above potential advantages of sensor networks open diversified application spectrum, from military, industrial to environmental and civilian monitoring.

Military Applications

Military applications of sensor networks drive a great number of wireless sensor networks technologies' development and advancement. The rapid deployment, self-organization and fault tolerance characteristics make sensor networks a promising technique for military command, control, communications, computing, intelligence,

surveillance, reconnaissance, and targeting (C4ISRT) systems [2]. A massive number of sensors with long-range acoustic/seismic sensing capability and short-range multiple-modality can be launched in the battle field in advance by military aircraft for placement by personnel or unmanned aerial vehicles (UAVs). They form a wireless linked network to monitor the status of a friendly force, equipment and ammunition, to obtain information on the composition and movements of opposing forces, to survey critical terrains, approach routes and paths, and to detect nuclear, biological and chemical weapons launched by enemies.

Environmental and Habitat Monitoring

Environmental and habitat monitoring is to monitor phenomena that usually distribute over a large area, such as to monitor the air [87], soil, and water [88] of critical terrains. Sensor nodes equipped with chemical sensors can be used to detect biological or chemical toxins in air or drinking water and provide early warning of attacks. In agriculture, wireless sensor networks are deployed to monitor both short-term changes in fertilizer and pesticide levels and long-term trend of vegetation responses to climate and diseases [92]. Sensor networks are also used in ecosystem monitoring which aims at understanding the response of wild populations to habitats over time [89] [90]. Other environmental monitoring applications include forest fire detection, flood detection, large-scale Earth monitoring and planetary exploration, and biocomplexity mapping of the environment [2].

Industrial Applications

Flexibility and low cost features have been attracting industry to employ WSNs as a means of improving machine performance and maintainability and lowering cost. Specifically, sensors have long been placed on or in machines to monitor their health through wear, heat, lubrication levels, or similar information [91]. Recently, with the advances in sensor technology, distributed computing and wireless communication technique, sensors begin to play an even more important role in manufacturing and monitoring effort. For example, in the semiconductor industry, the sensors can be instrumented to report machine and product status in real time for process control feedback and condition-based equipment maintenance. In the automobile industry, sensor networks are used to detect the pollution levels in an exhaust system when the engine runs, and to sense the vehicle deceleration for airbag deployment triggers [92].

Another important WSNs application in industry is structural health monitoring (SHM) where sensors are placed directly into the frame of buildings to detect structural damage and are connected wirelessly to exchange information. The typical sensors that are used in SHM normally include strain gauges, accelerometers, and velocity sensors. These embedded sensors report on the structural integrity and strength of the building and send out alarm when the detected changes exceed normal ranges [105].

Civilian Applications

The most significant application of sensor networks in civilian arena is tele-medical care of mobile patients. Portable sensors can be carried by patients to provide continual reading on vital signs, including blood pressure, respiratory rate, pulse, temperature and other physiological data and to determine medication levels in real time [110]. In addition, sensor networks have been used to monitor transportation patterns and perform traffic control in urban areas [95] [96]. Sensor networks are also used in building automation, home automation, energy utilization monitoring and control in departmental stores, etc [105]. In fact, due to the pervasive nature of sensors, sensor networks have the potential to revolutionize the very way we understand and construct complex physical systems [97].

1.3 The Need for Adaptable and Application-Aware Wireless Sensor Networks

From the above discussion, we may see that different applications have different requirements for the WSNs. An environment monitoring or military surveillance system requires rapid and high-fidelity transmission of the captured data to base station for further processing and assessment. Throughput requirement might or might not be a concern in this case, subject to nature and amount of the captured data. Highly-secured data transmission of the battlefield objects is a high priority requirement in a military surveillance system. While energy-efficiency is an expected requirement in a military surveillance system, but might not be as rigid as in environment monitoring that is expecting sensor nodes to function in an unattended environment for years of lifetime. Scalability and adaptability also become highly expected features because battlefield is a rough environment that may likely cause some nodes dying out. Thus ability to accommodate such network changes, i.e. reorganization capability of the network is a vital feature of this kind of network. But they may not be a major concern in an

industrial monitoring application where the sensor nodes are most likely stationarily deployed onto or surrounding the object to be monitored. Even in a multi-hop scenario, the relay nodes are often well positioned in accordance to a planned topology in order to maximize resource utilization of the sensor network in full spectrum. Hence, adaptability and application-awareness of a WSN determine the suitability of the WSN to an application.

Furthermore, we argue that even within the same application, there are needs for handling dynamic traffic loads. Existing duty cycle MAC protocols, including synchronous and asynchronous ones, are mainly optimized for light traffic loads (usually Low Traffic Load's message interval $> 10s$; and High Traffic Load's message interval $< 10s$). A WSN, however, could often experience bursty and high traffic loads. For example, either broadcast or converge-cast traffic could suddenly increase channel contention in a local neighborhood [81] [94]. Thus, existing protocols become less efficient in energy efficiency, latency and packet delivery ratio as traffic load increases. A highly adaptive protocol is required for WSNs to handle dynamic traffic loads and wide range of traffic loads change.

1.4 Adaptive MAC Protocols for Application-Aware WSNs

A wireless sensor network is composed of sensor nodes that are often deployed with ad hoc topology and densely inside or close to the objects or environment of interests. These tiny sensor nodes are mainly composed of three components: sensing unit (sensor and analog-digital converter (ADC)) for sensing, processing unit (processor and memory) for data processing, and transceiver for wireless communications.

All these components rely on a power unit for source of energy which is battery powered for most of the cases. For applications of WSNs, it is generally true that it is either too difficult or impractical to charge or replace the exhausted batteries, or it is cost-prohibitive to lay power lines for the sensor nodes. Future sensor nodes therefore are envisioned as disposable, which demands technologies that can prolong battery lifetime as much as possible. On the other hand, miniaturization requirement for the sensor nodes enforces such devices with a small memory, constrained energy supply, less processing power and communication bandwidth. Moreover, topologies of the sensor networks are constantly changing due to a high node failure rate, occasional

shutdown and abrupt communication interferences. All these limitations impose unavoidable requirements for the protocols designed for the WSNs. These requirements, once fulfilled, will become the features and performance metrics of the WSNs. They include, but not limited to:

- Energy efficiency; Low latency and delay; High throughput and packet delivery ratio; Self-organization ability; Scalability; Adaptability; and Fairness.

A stack of protocols of WSNs are designated in the different communication layers, including physical (PHY) layer protocol, data link layer (DLL), medium access control (MAC) protocol, network (NET) layer, transport layer, application (APP) layer. MAC protocol is the first protocol layer above the PHY layer and consequently MAC protocol heavily influences the performance of a WSN. The fundamental task of a MAC protocol is to regulate the access of a number of nodes to a shared medium in such a way that certain application-dependent performance requirements are satisfied [68]. In particular, MAC protocol coordinates the time and provides channel access control mechanisms when a number of nodes access a shared communication medium. While a synergetic capability through cross-layer design (CLD) amongst the different layers of protocols is important to enhancement of the performance metrics, we argue that MAC layer protocol plays the most significant role for a WSN, directly and indirectly. Even in the cross-layer design, MAC layer protocol is a vital component to the overall performance of a WSN eventually. The adaptability of MAC protocol to the data pattern and particulars of an application is a vital factor for a WSN towards application-awareness.

1.5 Thesis Statement

Based on definition of sensor networks, their potential in diverse applications, application-specific requirements for the WSNs, and their constraints outlined in previous sections, we argue that adaptive MAC protocols are significantly important towards application-aware WSNs and effectively and efficiently enable adaptability to traffic loads dynamics. Thus, the statement of this thesis is as following:

Enabling application-aware Wireless Sensor Networks with adaptive MAC Protocols.

Therefore, in this thesis, we present Adaptive Scheduling MAC protocol, called AS-MAC, in order to achieve adaptability to the dynamic traffic loads and various applications. While AS-MAC achieves energy-efficiency, low-latency and high packet delivery ratio (PDR) for multi-hop WSNs, there is a limitation in its adaptability due to use of a fixed duty cycle. Furthermore, we present a Self-Adaptive MAC protocol, called SEA-MAC, in order to enhance adaptability to dynamic traffic loads WSNs applications. SEA-MAC employs dual-adaptable mechanisms in protocol design to improve the adaptability to dynamic traffic loading requirement.

1.6 Thesis Contributions

This thesis addresses the technical challenges to enable the adaptable wireless sensor networks from the following two-folds, and provides projections on the future research and development of MAC protocols in third-fold. Hence the contributions of this thesis are embodied in the following three-folds and briefly described in the following three sub-sections below:

- 1) First-fold: MAC protocols development which is described in sub-sections [1.6.1](#) and [1.6.2](#).
- 2) Second-fold: characterization, modeling and rendering of application-specific requirements which are described in sub- section [1.6.3](#).
- 3) Third-fold: projections on future research of MAC scheme which is described in sub-section [1.6.4](#).

1.6.1 Adaptive Scheduling MAC Protocol – AS-MAC

First, we propose an energy-efficient, low-latency and high Packet Delivery Ratio (PDR) MAC protocol for multi-hop WSNs – AS-MAC. This AS-MAC protocol uses a novel Adaptive Scheduling (AS) technique to handle the data traffic. AS-MAC protocol originally combines contention-based approach controlled by a timeout mechanism with scheduling-based approach into a hybrid MAC protocol. AS-MAC protocol employs timeout mechanism to create a Resilient Active Time (RAT) within the AS-period, so as to allow nodes to schedule more data transmission within one operational cycle when the traffic load is high and go to sleep early when the traffic load is low. Concretely, the contributions of this part of my thesis can be summarized as follows:

- An Adaptive Scheduling period has been introduced to make the active time period adaptive to traffic load, which reduces idle listening thus saving energy.
- Two new parameters have been introduced in the scheduling frame (SCH) to prevent duplicate data transmission and delay to the scheduled data transmission.
- An enhanced mapping function has been proposed to prevent the collisions between data and acknowledgement (ACK) messages in most of scenarios.
- A progressive timeout mechanism has been proposed to prevent the node far away from the sink from early-sleep problem.

1.6.2 Self-adaptive MAC Protocol – SEA-MAC

While AS-MAC achieves energy-efficiency, low-latency and high PDR for multi-hop WSNs, its adaptability to the varying traffic is limited due to its fixed duty cycle. The fixed duty cycle, especially low duty cycle scheme has three drawbacks.

- 1) It will cause sleep latency if the duty cycle level is not enough to handle traffic flow promptly. The sleep latency increases proportionally with increasing number of hops.
- 2) The fixed duty cycle cannot adapt to the traffic variation. A fixed low duty cycle is designated for low traffic load to save energy, but it will result in high end-to-end delay and low packet delivery ratio (PDR) for the high traffic load. In the circumstance of very low traffic load, it will result in inefficient duty cycle problem. Vice versa, a fixed high duty cycle might help reduce end-to-end delay and increase PDR, but it will result in significant energy wastage and inefficient duty cycle problem when traffic becomes low. Thus, it is desirable to adopt an adaptive duty cycle for varying traffic loads in MAC protocol design.
- 3) A fixed synchronous duty cycle may increase the possibility of collision. If neighbouring nodes sense signals and turn to active mode at the same time, then all may contend for the channel, making a collision very likely.

Therefore, we propose a Self-Adaptive MAC protocol, called SEA-MAC, for traffic-dynamic WSNs. This MAC protocol employs dual-adaptable mechanisms in protocol design to enhance the adaptability to traffic loading dynamic requirement. SEA-MAC not only has a resilient active time (RAT) in AS-period, but also has a

resilient T_{as}/T_{sleep} ratio (RR), so as to be more adaptive to the traffic loads. SEA-MAC further adjusts the duty cycle dynamically based on the traffic load, so as to mitigate inefficient duty cycle problem in the circumstances of low traffic load, or reduce end-to-end delay in circumstances of high traffic load. SEA-MAC provides nodes with just enough active time for sending and receiving packets in the provided duty cycle.

The contributions of this part of my thesis can be summarized as follows:

- Inefficient duty cycle has been conceptualized first time in MAC protocol design and identified as a problem that impacts the MAC protocol performance.
- Inefficient duty cycle has been quantified so as to become calculable factor.
- An adjustable duty cycle scheme has been realized to enable the self-adaptive duty cycle MAC protocol.

1.6.3 Characterization and Rendering of Application-Specific WSN

In the inspiration of the need for a generic MAC protocol which does not assume any prior knowledge of application requirements but can tune its duty cycle automatically according to traffic conditions, as second-fold of contribution, we characterized and modelled application-specific requirements, and rendered them in the generic form for the proposed suite of MAC protocols, so that the WSN built with this model and render has good performance and adaptability to the different applications. We generalize the traffic-aware characterization and application-aware rendering with parameters and make the protocol adaptable to application-specific performance criteria.

1.6.4 Projections on Future Research of MAC Scheme

As the third-fold contribution, we provided projections on the future research of MAC scheme by presenting a dual-behaviour MAC scheme to the peers working in this area. The dual-behaviour MAC scheme combines contention-based protocol with scheduling-based protocol as a hybrid MAC scheme, and leverages the advantages of both types of protocol in different traffic loads and patterns by switching the behaviour of MAC protocols between the two types of protocols according to the traffic patterns and application-specific performance requirements. We provided qualitative performance analysis of the two types of protocols, illustrated how to determine the bounds for

switching, identified the possible factors affecting the bounds, and presented the performance of post-turning-points.

1.7 Thesis Organization

Chapter 2 identifies research issues of wireless sensor networks. The Chapter investigates the characteristics of WSNs, identifies research issues in MAC layer protocol, spells out the focused research topic of this dissertation, and illustrates the research methodology adopted in the thesis.

Chapter 3 reports the literature studies on the MAC layer protocols for WSNs – an essential layer to enable the adaptability and application-awareness. The literature study includes four types of MAC layer protocol: Contention-based MAC protocol, Scheduling-based MAC protocol, Hybrid MAC protocol and Preamble sampling based MAC protocol. This Chapter further analyzes the challenges of MAC technology for WSNs.

Chapter 4 describes the design principles for a novel energy-efficient and low-latency MAC protocol for multi-hop WSNs – Adaptive Scheduling MAC (AS-MAC), and exhibits the performance comparison between AS-MAC and DW-MAC. This novel MAC protocol, as an initial solution towards adaptable and application-aware WSNs, is targeting at industrial monitoring applications which require long lifetime and minimum delay in data relay.

Chapter 5 evaluates the performance of AS-MAC protocol with different AS-period in multi-hop networks. AS-period is one of the critical parameters in AS-MAC protocol. Performance comparisons will present performance characteristics of AS-MAC protocol with different AS-period, so as to determine the optimal parameter for AS-MAC protocol under varying traffic loads.

Chapter 6 evaluates the performance of AS-MAC protocol with different Timeout values in multi-hop networks. Timeout value is one of the critical parameters in AS-MAC protocol. Performance comparisons will present performance characteristics of AS-MAC protocol with different Timeout values, so as to determine the optimal parameter for AS-MAC protocol under varying traffic loads.

Chapter 7 describes the design principles of another novel MAC protocol for multi-hop WSNs – SEA-MAC Scheme Operation, and exhibits the performance comparison between SEA-MAC, AS-MAC, DW-MAC and RI-MAC. This SEA-MAC further adjusts the duty cycle dynamically based on the traffic load, so as to mitigate inefficient duty cycle problem in fixed duty cycle scheme.

In Chapter 8, based on the need for a generic MAC protocol which does not assume any prior knowledge of application requirements but can tune its duty cycle automatically according to traffic conditions, we describe characterization and rendering of the application-specific requirements of WSNs in the generic form.

Chapter 9 finally concludes this dissertation with a summary of contributions. It provides projections for future research work towards the adaptive and application-aware Wireless Sensor Networks and concluding remarks.

2 Research Issues in WSNs

In this chapter, we present the characteristics and challenges of WSNs, research issues in MAC layer protocol and wireless MAC issues. Then we depict the focused research topics in more details, and describe the research methodology and benchmarking evaluation approaches adopted in the following chapters.

2.1 Characteristics and Challenges of Wireless Sensor Networks

A sensor network is a deployment of massive numbers of small, inexpensive, self-powered devices that can sense, compute, and communicate with other devices for the purpose of gathering local information to make global decisions about a physical environment [69]. The nature of such a network imposes the following intrinsic characteristics that distinguish it from traditional networks:

2.1.1 Constrained Resource Challenge: Energy-Efficiency

The low-cost deployment is one acclaimed advantage of sensor networks, which implies the resources to individual nodes are severely limited. Low processing speed, limited memory, constrained energy supply and bandwidth make the designs that are unable to efficiently utilize these resources not applicable to sensor networks. This characteristic requires a WSN to have an energy-efficient algorithm for the MAC protocol, routing protocol, data processing and communication in order to prolong the lifetimes of sensor nodes.

2.1.2 Large Scale Challenge: Scalability and Robustness

Some WSNs are composed of hundreds or thousands of sensor nodes that provide a large geographic coverage. At the same time, a high density is required to work against the high failure rate of sensor nodes, the low confidence in individual sensor readings, the limited communication range and low capability of single sensor nodes. Such as battlefields and earthquake response systems, etc. are examples of the applications. The effectiveness and performance of the sensor network should not be affected when the number of sensor nodes is increased. This characteristic requires a WSN to have

scalability, self-configuration capability, and robustness.

2.1.3 Data Centric Network Challenge: Data-centric Routing and in-network Processing

Data centric processing is an intrinsic characteristic of sensor networks. Sensor data is no longer accessed by ID or by IP as in Internet and it is more natural to address the data through content, location or constraints. The IDs of the sensor nodes are of no interest to the applications. The naming schemes in sensor networks are often data oriented. For example, an environmental monitoring system requests the temperature readings through queries such as “collect temperature readings in the region area bounded by the rectangle (x1, y1, x2, y2)”, instead of queries such as “ collect temperature readings from a set of nodes with the sensor net address x, y and z.”. This characteristic requires a WSN to provide effective and efficient data-centric routing, in-network processing to ensure the easy query and energy-efficient processing.

2.1.4 Ad hoc Network Challenge: Network Discovery and Self-organization Capability

Network structure of WSN is formed in real-time. The structure is updated periodically due to the high failure rate of sensor nodes. Hence it requires effective and efficient networking discovery algorithms, which in turn requests nodes to be self-location aware; therefore requiring relative position algorithm.

2.1.5 Node Failure Challenge: Fault Tolerance

Sensor nodes are prone to failure due to environmental interference, insufficient power, and physical damage. Sensor network's topology changes frequently due to node failure. However, sensor nodes' failure should not significantly affect the performance of the whole sensor network. This is referred to as the *reliability* or *fault tolerance* challenge [2]

2.2 Research Issues in MAC Layer Protocol

2.2.1 Protocols Stack – A Research Foundation

Architecturally similar with other networks, such as wireless local area networks (WLAN), cellular networks, mobile ad-hoc networks (MANET), in WSN the individual

device operation for all nodes within a communications network is divided into layers, with each layer governed by its own rules or protocol. Each protocol layer is standardized to all other devices in the network, allowing layer-to-layer communication among all nodes. Within the device's protocol stack (hierarchical grouping of layers), one layer can communicate only with the layers immediately above or below itself. Importantly, the layers are not concerned with the other protocols running in the stack, only its own, in addition to the information required to be shared in the adjoining layers. This allows the substitution of protocols at various layers without affecting the others. The most common protocol stack found in wireless communications is composed of (lowest to highest) the physical layer, the data link layer, the network layer, the transport layer, and the application layer. The protocol stack of a WSN can be presented as **Figure 2-1** below. This thesis will focus on the MAC layer, which is a subset of the DLL layer, and the NET layer [2][70].

Figure 2-1: The protocols stack and research issues of a WSN

Since this research work focuses on the MAC protocol of DLL, we only elaborate the key research issues for the MAC layer, and brief the adjoining PYH layer and NET layer.

The Physical layer is responsible for frequency selection, carrier frequency generation, signal detection, modulation and data encryption [2]. Frequency generation and signal detection have more to do with the underlying and transceiver design and hence are beyond the scope of this research work.

The Network layer ensures that all packets are properly routed from the transmitter to the receiver. In point to point networks, like ad-hoc networks, this problem become complex, as the protocol must be able to choose routes for packets based on such factors

as (a) net hop count from sender to receiver, (b) overall distance traveled by the packet, (c) local network congestion avoidance, and (d) available resources at the nodes on the potential route [70]. The problems addressed in this layer are relevant but not the focus of the proposed research. The issues will be considered and elaborated further once needed.

2.2.2 Wireless MAC Issues

The *MAC sub-layer* addresses the problem of controlling access of the wireless devices to the transmission medium in a broadcast network. In situations where several devices communicate by sharing the same medium, the MAC protocol will schedule and designate access channels to avoid simultaneous transmissions (collisions) on the same channel [76]. The unique properties of the wireless medium make the design of MAC protocols very different from, and more challenging than, wireline networks. The unique properties of wireless systems and their medium are:

- **Half-Duplex Operation:** In wireless systems, there is node *self-interference* dilemma. When a node is transmitting data, a large fraction of the signal energy leaks into the receive path, causing receiver very difficult to receive data. The transmitted and received power levels can differ by orders of magnitude. The leakage signal typically has much higher power than the received signal, which makes it impossible to detect a received signal while transmitting data. Hence, collision detection is not possible while sending data, so Ethernet-like protocols cannot be used. Due to the half-duplex mode of operation, the uplink and downlink need to be multiplexed in time (TDD) or frequency (FDD). As collisions cannot be detected by the sender, all proposed protocols attempt to decrease the probability of a collision using collision avoidance principles [7].
- **Location-Dependent Carrier Sensing:** It is well known that in free space, signal strength decays with the square of the path length [71]. As a result carrier sensing is a function of the position of the receiver relative to the transmitter. In the wireless medium, because of *multipath propagation* (The signal received by a node is a superposition of time-shifted and attenuated versions of the transmitted signal. As a result, the received signal power varies as a function of time), signal strength decays according to a power law with distance. Only nodes within a specific radius of the

transmitter can detect the carrier on the channel [7]. This location-dependent carrier sensing results in three types of nodes in protocols that use carrier sensing.

- **Hidden Nodes Problem:** A hidden node is one that is within the range of the intended destination but out of range of the sender [72]. Consider the case shown in **Figure 2-2**. Node A is transmitting to node B. Node C cannot hear the transmission from A. During this transmission when C senses the channel, it falsely thinks that the channel is idle. If node C starts a transmission, it interferes with the data reception at B. In this case node C is a hidden node to node A. Hence, hidden nodes can cause collisions on data transmission.
- **Exposed Nodes Problem:** An exposed node is one that is within the range of the sender but out of range of the destination [72]. In **Figure 2-2**, consider the case that node B is attempting to transmit to A. Node C can hear the transmission from B. When it senses the channel, it thinks that the channel is busy; therefore restrain itself from attempting to transmit any data. However, any transmission by node C does not reach node A, and hence does not interfere with data reception at node A. In theory, C can therefore have a parallel conversation with another terminal out of range of B and in range of C. In this case, node C is an exposed node to node B. If the exposed nodes are not minimized, the bandwidth is underutilized.
- **Capture Problem:** Capture occurs when a receiver can cleanly receive a transmission from one of two simultaneous transmissions, both within its range [72]. In **Figure 2-2**, when nodes A and D transmit simultaneously to B, the signal strength received from D is much higher than that from A, and D's transmission can be decoded without errors in the presence of transmission from A. Capture can improve protocol performance, but it results in unfair sharing of bandwidth with preference given to nodes closer to the BS.

Figure 2-2: Location-dependent sensing: hidden nodes, exposed nodes, and capture.

We have to consider all these constraints imposed by the physical nature of the medium in the design of protocols. In addition to the need to overcome these constraints, MAC protocols have to address a set of performance metrics that are very closely related to the application requirements; therefore they are used to compare the performance of the protocols. Consequentially a MAC protocol's awareness to different aspects of the metrics determines the suitability of the protocol to an application. The performance metrics are described as follows:

▪ ***Energy Efficiency***

Energy is a primary concern in sensor networks since most nodes are battery powered. Hence energy efficiency of MAC protocol is the most important requirement for fulfillment. There are four major sources of energy waste identified in [10], [12] as described briefly as follows:

- *Collision*: When a transmitted packet is corrupted it has to be discarded, and the follow-on retransmissions increase energy consumption. Collision increases latency as well.
- *Overhearing*: meaning that a node picks up packets that are destined to other nodes.
- *Control packet overhead*. Sending and receiving control packets consumes energy too. Contention-based protocols introduce contention-packet overhead; scheduling-based protocols introduce synchronization-packet overhead.
- *Idle listening*: This is major source of inefficiency. *i.e.*, listening to receive possible traffic that is not sent. Some measurements have shown that idle listening consumes 50–100% of the energy required for receiving. For example, Stemm and Katz measure that the idle:receive:send ratios are 1:1.05:1.4 [73], while the Digita 2Mbps Wireless LAN module (IEEE 802.11/2Mbps) specification shows idle:receive:send ratios is 1:2:2.5 [74].

▪ ***Collision Avoidance***

This is a basic task of any MAC protocols that attempt to increase efficiency of almost all the aspects of metrics, including energy-efficiency, end-to-end delay,

throughput, channel utilization, etc. It determines when and how a node can access the medium and send its data. Collisions are not always completely avoided in the regular operation. Some levels of collisions are acceptable in contention-based MAC protocols.

- ***Scalability and Adaptability***

Refer to capability to adapt to the changes in network size, node density and topology. Some nodes may die over time; some new nodes may join later; some nodes may move to different locations. A good MAC protocol should gracefully accommodate such network changes.

- ***Delay***

Delay is defined as the average time spent by a packet in the MAC queue, i.e., from the instant it is enqueued till its transmission is complete. Delay is a function of protocol and traffic characteristics. Therefore, when comparing protocols, it is necessary to compare them based on the same traffic parameters.

- ***Throughput***

The amount of data transferred from sender to receiver in unit time. Throughput is the fraction of the channel capacity used for data transmission. A MAC protocol's objective is to maximize the throughput while minimizing the access delay. If the average message size is P bits, the average time to transfer a single packet is T secs, and C b/s is the capacity of the channel, then the throughput η is given by $\eta = P/TC$.

- ***Channel Utilization***

How well is the channel used? It is also called bandwidth utilization.

- ***Fairness***

A MAC protocol is fair if it does not exhibit preference to any single node when multiple nodes are trying to access the channel. This results in fair sharing of the bandwidth [75]. This definition can be biased when traffic with different priorities is handled. When multimedia traffic is supported, fairness is defined as being able to distribute bandwidth in proportion to their allocation.

2.2.3 Focused Research Topics

Different applications stress different aspects of the performance metrics described above. An environmental monitoring and military surveillance system require rapid and high-fidelity transmission of the captured data to base station for further processing and assessment. Hence a low-latency and high energy-efficient MAC protocol is expected in such applications. The throughput requirement might or might not be a problem, subject to the nature and amount of the captured data. Highly-secured data transmission of the battlefield objects is a high priority requirement in a military surveillance system. While energy-efficiency is an expected requirement in a military surveillance system, but might not be as rigid as in environment monitoring that is expecting sensor nodes functioning in an unattended environment for years of lifetime. Scalability and adaptability also become highly expected features because battlefield is a rough environment that may likely cause some nodes dying out. Thus ability to accommodate such network changes, i.e. reorganization capability of the network is a vital feature of the network. But they may not be a major concern in an industrial monitoring application where the sensor nodes are most likely stationarily deployed onto or surrounding the object to be monitored. Even in multi-hop scenario, the relay nodes are often well positioned in accordance to a planned topology in order to maximize resource utilization of the sensor network in full spectrum.

To date, while sheer number of research works address energy-efficiency and other related aspect of performance metrics [15], [10], [12], [16], [17], they are targeting at only one specific application, and are not suitable for wide range of applications requirements. For example, when traffic patterns change, for instance from low-traffic load to high-traffic load, some protocols cannot exhibit their original performance, likely resulting in a shortened lifetime, long latency, decreased delivery ratio, etc. To address the problems, we focus our research work in the following areas:

- Research and development of energy-efficient and adaptive MAC protocols:

In whatever application, energy-efficiency is a top priority requirement in MAC protocol design for WSNs. It directly determines the lifetime of a WSN. Thus, it is the first priority for us to consider in the design of the protocol. Adaptability and application-awareness of MAC protocols refers to the capability of protocols to adapt to

application-specific requirements without sacrificing energy-efficiency. Hence the research and development shall devote to a suite of protocols stressing different aspects of performance metrics, such as energy-efficient and low-latency MAC protocol; energy-efficient and delivery-guaranteed MAC protocol; energy-efficient and high-throughput MAC protocol; energy-efficient, low-latency and high-throughput MAC protocol; etc. These combinations of performance metrics compose multimodality of WSNs; therefore become the cornerstones towards adaptability and application-awareness of WSNs.

- Characterization, modeling and rendering of application-specific requirements for a WSN:

Development of specific application-awareness of MAC protocol is the first step towards adaptability and application-awareness of a WSN. How to characterize and model application-specific requirements, how to present the multimodality in a way that can be rendered to the MAC protocols are to be investigated. Successful characterization modelling and model-rendering make the WSNs enabled by the multimodal adaptive MAC protocols to be able to present energy-efficiency and adaptability and application-awareness to the diverse applications.

2.3 Research Methodologies

2.3.1 Development Environment

The large-scale simulation is commonly regarded as a viable approach to understand and evaluate the performance of the target systems. After a survey on the wireless simulation platforms available, we adopt OMNet++ simulator to support our research. OMNet++ is an open source, component-based, modular and open architecture environment for discrete event simulation [OMNeT++]. OMNet++ is suitable not only for modeling but also for simulation. OMNet++ models the different elements of the modelled network through a component-based architecture and a hierarchical organisation of the network. These compound modules allow simulation of large-scale networks. The programmable component-based aspect of OMNeT++ allows the user to describe the network topology in a very flexible way. OMNet++ provides a Graphic Network Editor (**GNED**) that allows exporting the generated network topology files into XML format and importing XML files respecting the network topology definition

to create topologies. Since XML is a common way to exchange structured information between applications, this provides a standard means to interface OMNeT++ with other systems. The OMNeT++ simulator includes a set of user interfaces and tools that may be used to perform state estimations of the networks and also to visualise the simulation results. OMNeT++ provides a *Tkenv*-based user interface, which supports interactive execution of the simulation, tracing and debugging.

We shall use Crossbow Professional Kits with Add-on Sensor Nodes as our test-bed. However, for very large-scale sensor networks simulated in the simulation environment, it is nearly impossible to make a full evaluation on a physical test-bed. Therefore, we only use it to provide proof-of-concept evaluations of our designs for such a network scenario.

Table 2-1: Development Environment

<i>Items</i>	<i>Description</i>
Simulation	OMNeT++
Test-Bed	Crossbow Professional Kits (WSN-PRO2110CA) with Add-on Sensor Nodes (SN21140CA)
OS	Linux, other Unix-like systems and on Windows (XP, Win2K)
Language	C++
Integral Development Environment	Make, Visual .NET, GNED editor, Tkenv/Cmdenv user interface
Statistical tool	Excel, <i>Plove</i> tool for visualisation of output vector files, <i>Scalars</i> tool for visualisation of output scalar files.
Documentation	Word, Visio, Excel, PowerPoint.

2.3.2 Benchmarking Evaluation

Since our research will focus on two aspects of topics as stated in section 1.6, the evaluation will consist of two phases: evaluation of individual scheme towards energy-efficient and adaptive MAC protocol, and evaluation of integration of adaptive MAC protocol and application-awareness modeling. Table 2-2 shows the partial plan of individual evaluations. The first row illustrates the meaning of the individual evaluation with the proposed AS-MAC protocol as example, i.e. the proposed AS-MAC protocol compares with the baseline protocol in terms of performance metrics listed at right column. The second and rest of rows list the MAC protocols to be designed with

consideration of a set of performance metrics that meet application-specific requirements.

Table 2-2: Individual Evaluations

<i>Protocols</i>	<i>Baselines</i>	<i>Performance Metrics</i>
AS-MAC	S-MAC [12], S-MAC-AL [10], DW-MAC [24]	Energy-efficiency, End-to-end delay, Packet delivery ratio
SEA-MAC	AS- MAC, RI-MAC [112]	Energy-efficiency, Duty Cycle, End-to-end delay, Packet delivery ratio
YY-MAC	YY-Baseline-MAC	Energy-efficiency, delivery ratio, fairness
...	...	...

3 Related Work

We have carried out an extensive literature survey on current research works for WSNs. A number of survey papers on WSNs and MAC protocol [1][2][3][4][5][6] [7] have been studied; a great number of different type of MAC protocols [9][10][11] [12][13][14][15][16][17][18][19][20][21][22][23][24][25][26][27][28][29][30][31][32][33][34][35][36][37][38] have been individually and deeply researched, especially on energy conservation mechanisms [51][52][53][54], traffic-aware mechanisms [55][56][57][58][59] [60] [61] [62] [63] [64] [65] [66] [67], etc. Due to the space limitation, we cannot include all of the works. In this section, we only focus on those works that are directly related to our research.

3.1 Classification and Analysis of WSN MAC Protocols

The MAC protocols designed for WSNs are required to accommodate the unique features of WSNs. Important design factors include *self-organization ability*, *Power Consumption*, *Fairness*, *Delay*, *Scalability*, *Adaptability* and *Throughput*. The detailed description can be found in [6].

MAC protocols can be classified based on the underlying mechanism of collision avoidance and the organization approach of sensor nodes. Conventional wireless MAC protocols may be classified based on architecture (distributed or centralized) first, then further classified based on mode of operation (random access, guaranteed access, and hybrid access) [6]. For wireless ad hoc networks, MAC protocols may be classified as contention-based and contention-free [1]. Contention-based MAC protocol may be further classified as *sender-initiated* vs. *receiver-initiated*, *single-channel* vs. *multiple-channel*, *power-aware*, *directional antenna based*, *unidirectional link based* and *QoS aware schemes*; contention-free protocol may be further classified as *time-division multiple-access* (TDMA), *frequency-division multiple-access* (FDMA), and *code-division multiple-access* (CDMA) techniques.

While wireless sensor networks are different from wireless ad hoc networks, the MAC design for wireless sensor networks can be similarly categorized. [98] classifies the MAC protocols into random access, slotted access, frame-based and hybrid

protocols based on how nodes organize access to the shared medium. [99] broadly classifies the MAC protocols into contention-based protocols and schedule-based protocols. [100] similarly classifies the MAC protocols into unscheduled (contention-based) and scheduled protocols, and also provides one more layer of detailed classification under the two broad categories. These classifications are based on the specific techniques used in the MAC protocols. [111] provides a thematic taxonomy and classifies MAC protocols according to the problems dealt with: scheduled protocols, protocols with common active period, preamble sampling protocols and hybrid protocols. We broadly classify the MAC protocols for WSNs into *Contention-based protocols*, *Contention-free (Scheduled-based) protocols*, *Hybrid protocols* and *Preamble Sampling protocols* based on our analysis and summary of the MAC protocols for WSNs.

Contention-based MAC protocol operates on demand basis, where sensor nodes access the shared channel in asynchronous manner, and they compete for channel whenever there are packets in their buffers ready to be sent [8]. By nature, the contention-based approach provides good scalability needed by WSNs to support the node changes. Unfortunately, the downside of this approach is its serious energy inefficiency due to idle listening, collision, overhearing, and control packet overhead [9].

Contention-free MAC protocol, also known as *scheduling-based protocol*, operates based on scheduling. Scheduled channel-access schemes can involve time slots, frequency bands, or spread-spectrum codes. The greatest advantages of the scheduling-based MAC protocol are the absence of collision and control-packet overhead and few control-packets overhead which are necessary for maintaining the synchronization among the nodes. Nevertheless, it has some disadvantages that negatively affect the network scalability and adaptability. Some scheduling-based protocols employ numbering system to identify the nodes and establish network routing. Any addition of new nodes or deletion of the existing nodes require re-establishing network routing and re-start the network, which affect the network scalability. Some scheduling-based protocols use fixed time-slot in scheduling of control/data packets exchange, which restrict the network's adaptability to the varying traffic loads. When the fixed time-slot is long, but traffic load is low, energy-efficiency will be reduced. On the other hand,

when the fixed time-slot is short, but traffic load is heavy, the end-to-end delay will be prolonged.

Having realized advantages and disadvantages of the abovementioned two distinct types of MAC protocol, more people have developed *hybrid MAC protocol* that aims to leverage advantages and mitigate disadvantages of these two types of protocol. Hybrid MAC protocol might be designed to switch its behavior between contention-based protocol and scheduled-based protocol according to the traffic loads and patterns. Thus it can be further classified as *Dual-behaviour hybrid MAC protocol*.

Another type of protocol uses preamble sampling or channel polling for medium access control. The two terms refer to the same procedure of periodically polling the wireless channel to test for traffic. It is classified as *Preamble Sampling based MAC protocol*.

It is not possible to include all the MAC protocols in the literature study due to the space limits and newly developed MAC protocols published continually. However, many representative MAC protocols in the abovementioned categories have been included and discussed in this literature study.

3.2 Contention-Based MAC Protocols

Due to half-duplex characteristic of operation in wireless networks, almost all MAC protocols are based on carrier sensing and/or collision avoidance mechanisms to perform contention to access medium channel. *Carrier sensing* refers to listening to the physical medium to detect any ongoing transmissions. *Collision avoidance* can be implemented by using additional signaling control messages, such as *in-band handshaking* [12] and *out-of-band signaling* [12].

The standardized IEEE 802.11 distributed coordination function (DCF) is a typical example of the contention-based protocol. **Figure 3-1** shows the IEEE 802.11 frame. It is widely used in ad hoc wireless networks because of its simplicity and robustness to the hidden node problem. However, study shown that its energy consumption is very high when nodes are in idle mode due to the idle listening. In addition, If IEEE 802.11 MAC frame is used in broadcast, the addresses of the sender and receivers are not needed because every node is the recipient of the transmission. The shaded address portion in

Figure 3-1 creates unnecessary overhead which in turn increases energy consumption. Power-Aware Medium-Access with Signaling (PAMAS) and Sensor-MAC (S-MAC) made improvement on energy saving over the standardized IEEE 802.11 in different ways.

Figure 3-1: IEEE 802.11 frame

3.2.1 PAMAS

The Power-Aware Medium-Access with Signaling (PAMAS) [11] is one of the first power-aware MAC protocols proposed for multi-hop wireless networks. The PAMAS extended Medium Access with Collision Avoidance (MACA) technique by carrying out the control signaling (RTS, CTS, busy tones) on a separate radio control channel from data channel. PAMAS avoids overhearing by putting nodes into a sleep mode whenever their neighbors are in transmission. When a node wakes up, it sends a probe signal via control channel to identify any possible ongoing transmissions and their duration. If it receives answers from any neighbor, the node will sleep again for the specified duration.

The advantages of PAMAS is that control signals exchanged via the control channel among the neighboring nodes do not interfere with neighbors' transmissions in the data channel, which provides an efficient way for control signal exchanging, especially in the case of low traffic load.

The main disadvantages of PAMAS, however, is that having two separate channels makes PAMAS more difficult to deploy as two channels require two radios and additional complex channel allocation, which increases the energy consumption and the device cost. Another major drawback is that a node that can be reached in the control channel may not be reachable on the data channel in actual deployment due to the possibility of having different characteristics on each channel. In addition, PAMAS does not address the issue of reducing idle listening.

Since the message size in most WSNs is small, the benefits of separating data and control transmission is not brought into play. If the design can reduce the cost of the

second transceiver, PAMAS might be applicable to the sensor networks with large size of data whereby the above benefits can be made good use of.

3.2.2 S-MAC and S-MAC-AL

Sensor-MAC (S-MAC) [12] was proposed for such an application scenario: sensor networks are deployed in an ad hoc fashion, with nodes remaining mostly inactive for a long time, but becoming suddenly active when something is detected. For such applications, energy conservation and self-configuration are primary goals, while per-node fairness and latency are less important.

Unlike PAMAS, S-MAC does not use any out-of-channel signaling. S-MAC uses several novel techniques to reduce energy consumption and support self-configuration. (1) It enables *low-duty-cycle* operation allowing nodes to choose their own listen/sleep schedule in a multi-hop network to reduce energy consumption. (2) It adopts *virtual clusters* where the neighboring nodes have common sleep schedules to reduce control overhead and enable traffic-adaptive wake-up. (3) It uses *in-band signaling* and *Network Allocation Vector* (NAV) to avoid energy waste on overhearing unnecessary traffic. (4) It applies *message passing* to reduce contention latency for applications that requires in-network data processing. The experiment results show that S-MAC obtains significant energy savings compared to a random access MAC protocol without sleeping. **Figure 3-2** depicts basic scheme of S-MAC where a complete operation cycle consists of SYNC period, DATA period and SLEEP period. All nodes perform synchronization in SYNC period. Nodes with data to send contend channel and exchange control packets RTS/CTS with the receiver in DATA period to appoint wakeup time and duration in SLEEP period. Those nodes overhearing the RTS/CTS packets use random back-off and NAV (Network Allocation Vector) mechanisms to back-off the appointed duration. The node wins the contention and its receiver wakeup at the appointed time and for the appointed duration in SLEEP period to transmit and receive data packets, and then go to sleep.

Figure 3-2: S-MAC basic scheme

S-MAC was significantly extended to S-MAC-AL [10] (we use different abbreviations purposely to distinguish these two versions and reveal evolution of the protocol) by adding in a new *Adaptive Listening* technique to further reduce sleep latency caused by the periodic sleep of each node. Since S-MAC-AL [10] systematically handles energy efficiency issue from several key aspects of power consumption, and its original work [12] appeared in year 2002, S-MAC-AL probably is the most studied MAC protocol for WSNs. For the sake of this reason, we put more detailed analysis on its key design points and resultant of them as follows.

(1). *Choosing, Maintaining and Synchronizing Schedules*: S-MAC-AL uses *Periodic Listen and Sleep* scheme as basis in its protocol design. Before each node starts its periodic listen and sleep, it needs to choose a schedule, maintain it in its *schedule table* (storing the schedule of all its known neighbors) and exchange it with its neighbors at its next scheduled listen time. Neighboring nodes form a virtual cluster so as to set up a common sleep schedule. If two neighboring nodes reside in two different virtual clusters, they wake up at the listen periods of both clusters.

(2). *Adaptive Listening*: Periodic sleep results in high latency, especially in multi-hop networks. The latency caused by the periodic sleep is called *sleep latency* [10]. To reduce sleep latency, S-MAC-AL lets node who overhears its neighbor's transmissions wake up for a short period of time at the end of the transmission. Hence, if the node is the next-hop node, its neighbor can immediately pass the data to it instead of waiting for its scheduled listen time, which is illustrated in Figure 3-3.

Figure 3-3: Adaptive Listening in S-MAC-AL

(3). *Overhearing Avoidance*: S-MAC-AL uses *Network Allocation Vector* (NAV) to realize virtual carrier sense, and uses both physical carrier sense and virtual carrier sense to determine if the medium is free. As a result, each node overhears many packets that are not directed to it; hence a significant waste of energy, especially when node density is high and traffic load is heavy. **Figure 3-4** elaborates the means of overhearing avoidance used in S-MAC. Supposed node A is currently transmitting a data packet to B. S-MAC lets all immediate neighbors of both sender and receiver go to sleep after they hear an RTS or CTS packet until the current transmission is over, as indicated by “X” in **Figure 3-4**.

Figure 3-4: Who should sleep when node A is transmitting to B?

(4). *Message Passing*: S-MAC-AL uses message passing, in which long message are divided into frames and sent in burst, to minimizing communication overhead.

The advantages of S-MAC-AL are in several folds. The low-duty-cycle scheme significantly reduces the energy waste caused by idle listening. Adaptive listening significantly improves the sleep latency caused by the periodic sleep of each node in a multi-hop network and increases applicability of the scheme in its targeted applications. The virtual cluster reduces synchronization overhead, but does not require coordination from cluster heads. Instead, nodes can communicate directly with peers, and only need to coordinate with neighbors, which make it more robust to topology change than cluster-based approaches and more scalable.

The main disadvantages of S-MAC-AL are that nodes exchange their schedules by periodically broadcasting an SYNC packet to their immediate neighbors, which increase collision probability, when topology changes and before network becomes stable. S-MAC-AL actually relies on the SYNC packets for neighborhood discovery. Since the SYNC packets are transmitted unreliably due to possible collisions and interference, S-MAC-AL will experience a long warm-up period for neighborhood information discovery. Sleep and listen periods are predefined as a constant, which decreases the efficiency of the algorithm and sensitivity to varying traffic loads. Sensor nodes may have several schedules to follow, which multiplies the duty cycle of the nodes and causes these nodes to die faster. While message passing minimizes communication overhead, it is obtained at the expense of unfairness in medium access. Due to S-MAC-AL's clear goal as mentioned earlier, such per-node unfairness is a traded-off for a better energy conservation overall and contributes to good scalability and collision avoidance. Since S-MAC and S-MAC-AL adopt fixed duty-cycle, they are not adaptive to dynamic traffic loads.

3.2.3 AC-MAC

In some real-time applications over sensor networks, such as target-tracking, the amount of instantaneous traffic load comes from the speed of the moving target. Such dynamic traffic needs a traffic-adaptive protocol to handle. To accommodate traffic dynamics, Adaptive Coordinated MAC (AC-MAC) [13], taking S-MAC as a basis, designed a *traffic-adaptive* mechanism to further improve latency and throughput while remaining as energy-efficient as S-MAC. Dynamically coordinating the duty cycles is the key design point of AC-MAC. AC-MAC uses the number of packets queued at the MAC layer as an indication of the traffic load. Based on the traffic information, AC-MAC can provide an adaptive number of chances for communication within one basic cycle time for S-MAC. **Figure 3-5** shows the scheme of the adaptive duty cycle of AC-MAC.

Figure 3-5: The periodic listen/sleep cycle for AC-MAC. For reason of possible high traffic, AC-MAC provides 2 chances of communication within corresponding one cycle interval of S-MAC.

The number of packets queued at MAC layer for node i , say N_i , can be mapped into a right value, $R_i = f(N_i)$. Where $f(N_i)$ is an application-specific function which expresses the desire of a node for transmission. Each new reduced duty cycle composes of a fixed RTS/CTS interval and a varying sleep interval depending on R_i . With increase of the value R_i , the sleep interval in a reduced cycle time is decreasing. To avoid collision between the data transmission and the next following RTS/CTS interval, the shortest sleep interval should cover the maximum time for data transmission. The R_i value is determined only at the beginning of the every basic cycle time by every node.

The advantages of AC-MAC are that it introduces traffic-aware mechanism to adjust duty-cycle, thus improve the latency and throughput without sacrificing energy-efficiency. Sensor nodes only adopt one schedule per SYNC period, which simplifies the protocol, mitigates overhead of the SYNC packets and decrease the chance of collision of SYNC packets in S-MAC scheme. Since S-MAC already takes low-duty-cycle, the latency and throughput improvement from traffic-aware mechanism is not significant in high-traffic-load scenario, and keep almost the same as S-MAC in low-traffic-load scenario.

3.2.4 T-MAC

Targeting at the applications with traffic load variations in time and location, Timeout-MAC (T-MAC) [14] introduces an *adaptive* duty cycle in different way – by dynamically ending the active period of the cycle through a timeout mechanism as shown in [Figure 3-6](#). It can reduce the amount of energy wasted on idle listening, in which nodes wait for potentially incoming messages, while still maintaining a reasonable throughput (in homogeneous local unicast). An *early sleeping* problem, as a side effect of overhearing avoidance in S-MAC, has been observed and attempted to be resolved in T-MAC through two mechanisms. The first is *future request-to-send* (FRTS) packet that allows sensor nodes that have data to send to inform the next hop of a future message transfer time that they want as so to awake the destined node just-in-time and improve latency. The second is a *full-buffer priority with threshold control* that allows sensor nodes whose buffer is almost full to take priority to control the channel by immediately sending an RTS even just after receiving an RTS from another node, so that the full-buffer node won't sleep early which might cause buffer overflow and lose useful data.

Figure 3-6: The basic T-MAC protocol scheme, with adaptive active times

By head-to-head comparison with classical CSMA without duty cycle and the S-MAC with fixed duty cycle through extensive simulations in [14], it can be found that under homogeneous message arrival rate, T-MAC and S-MAC achieve similar reductions in energy consumption as up to 98 % compared to CSMA, and in a sample scenario with variable message arrival rate. While T-MAC outperforms S-MAC in energy saving by a factor of 5, such a gain is actually at the cost of reduction of throughput and additional latency [14] [15], especially in the high-load situation. In the worst case, T-MAC has only about 70% of the throughput achieved by S-MAC when message arrival rate fluctuates.

T-MAC has several advantages. Its adaptive duty-cycle makes it to be adaptive to varying traffic loads and save energy. It can optionally be deployed with different features and mechanisms, such as overhearing avoidance, full-buffer priority, and FRTS. This capability offers possibility to leverage different mechanisms in different applications with various data pattern. However, which features and mechanisms will be opted in depends on the communication pattern, such as Local Unicast / Broadcast, Nodes-to-Sink reporting, etc. But it is a challenge to distinguish the communication pattern of a live WSN.

Advantages – T-MAC protocol can optionally be deployed with different features and mechanisms, such as overhearing avoidance, full-buffer priority, and FRTS. This capability offers possibility to leverage different mechanisms in different applications with various data patterns.

Disadvantages – However, which features and mechanisms will be opted in depend on the communication pattern, such as local unicast / broadcast, nodes-to-sink reporting, etc. It is hard, however, to distinguish the communication pattern of a live WSN. Another downside is that the timeout mechanism that makes T-MAC adaptable to varying traffic loads can also bring about the early sleeping problem. A shorter timeout

implies an aggressive power-down policy with more energy saving, but making nodes go to sleep too early, which in turn results in longer latency.

3.2.5 TEA-MAC

Throughput and Energy-Aware MAC (TEA-MAC) is another contention-based MAC protocol. It focuses on high utilization of WSNs while ensuring low consumption of energy. TEA-MAC's vision is that energy saving in WSNs can be obtained without sacrificing throughput and suffering any additional delays [15]. Two main contributions of TEA-MAC are: (1) a sleeping sensor should wake up as soon as a communication is requested because such just-in-time wake-ups can reduce latency, thus increase the throughput. (2) Waking sensor should sleep as soon as it becomes idle since instant sleeps can reduce non-productive consumption of energy.

TEA-MAC addressed energy inefficiency issue from three aspects: (1) idle listening, (2) control-packet overhead, and (3) overhearing. It employs frequent sleeps to significantly reduce idle listening. It proposes a simplified structure of IEEE 802.11 packet frames to reduce the overhead of control packets, thus improve latency. It exploits overhearing intelligently and uses short sleeps and productive overhearing to achieve high throughput.

TEA -MAC has some key points in its protocol design:

(1). *Simplified Structure of IEEE 802.11 MAC Frames*: Having considered that broadcast is the prevalent communication method for WSN, it has been realized that the address portion (shaded portions) for both sender and receiver in IEEE 802.11 MAC packet frame (Figure 3-7) are not needed because every node is the recipient of the transmission. Shaded address portion creates unnecessary overhead. Similarly, the sender/receiver's address in RTS, CTS and ACK can be eliminated from the frame.

Figure 3-7: IEEE 802.11 frame

(2). TEA-MAC achieved energy efficiency through *Two-State/Two-Counter/Two-Scheme*. Each sensor node has two states (*Awake* and *Sleep*), two counters (*Wake counter* and *Sleep counter*) and two operational schemes (*Sand-clock scheme* and *Wake-*

time sharing scheme). By nature, a sensor node is either in the wake-up state or in the sleep state. The two counters are to control wake-up time and sleep time. The two schemes (*Sand-clock (Intra node) scheme* and *Wake-time sharing scheme (inter nodes)*) are to control the continuous transition between the two states. **Awake state** refers to either receive or transmit mode of its radio. **Sleep state** refers to the sleep mode of its radio. The two counters are non-negative integers to control wake-up time and sleep time. **Wake counter** determines the waking duration, whose value is positive when the sensor awakes but becomes zero when the sensor goes to sleep. **Sleep counter** determines the sleep duration, whose value is positive when the sensor sleeps but becomes zero when the sensor wakes up or start to communicate (i.e., transmits or receives). The two schemes are ruling how the two states transition occurs continuously. ***Sand-clock (Intra node) scheme***: if a sensor has no activities when awake, its timer decrements *wake counter* while incrementing the *sleep counter*. **Figure 3-8** depicts the basic activities of a sensor node during its waking and sleeping time (*Sand-clock scheme*).

Figure 3-8: Sand-clock (Intra a node) scheme: If a sensor has no activities when awake, its timer decrements wake counter while incrementing the sleep counter.

Wake-time sharing scheme (inter nodes): a communication (e.g., transmission or reception) occurring between the sender and receiver will trigger the updating of all counters. When a sensor node transmits or receives in the awake state, its wake counter is reset to the value of the associated packet duration plus the processing delay duration, and its sleep counter is reset to zero. **Figure 3-9** depicts the wake-time sharing scheme. By sharing the same values in their counters, neighboring nodes can wake up in the same time. **Figure 3-10** shows how the counters are updated in communications.

Figure 3-9: wake-time sharing (Inter-nodes) scheme.

Figure 3-10: Counters are updated in communication

3.2.6 asTEA-MAC

TEA-MAC has been enhanced by TEA-MAC with adjustable sleeps (asTEA-MAC) [9] through introducing 3rd counter – *Message counter* and 3rd scheme – *sleep-time adjustment scheme*.

Message counter: records the number of messages sent and received during a *waking* period, whose value is reset to zero when the sensor wakes up.

Sleep-time adjustment scheme: Before going to sleep, the *sleep counter* will be adjusted based on the *message counter* to reflect the traffic load, reduce energy consumption in a low-traffic scenario and increase throughput in a high-traffic scenario.

Figure 3-11 and Figure 3-12 detail the *sleep-time adjustment scheme*.

Figure 3-11: Sleep-time adjustment (intra-node) scheme

Figure 3-12: Sleep counter is adjusted before sleep

asTEA-MAC realized reduction of energy consumption without sacrificing throughput by eliminating non-productive energy consumption in idle listening and overhearing and establishing virtual synchronization without overhead. Simulation results show that in high-traffic and low-traffic situations, asTEA-MAC has a low energy consumption and high throughput [9], which are very desirable features for WSNs' applications which often present unpredictable varied traffic load. Since the communication schedule in asTEA-MAC does not rely on the identity and location of a node's neighbors, it is independent of the arrival of a new sensor node and departure or replacement of an existing sensor node. asTEA-MAC therefore achieves good scalability and adaptability.

3.2.7 Sift and CSMA/p*

Targeting at event-driven applications over WSNs, Sift [16] uses a randomized CSMA protocol. The motivation to produce Sift is that when an event is detected, the first R of N potential reports are the most crucial part of messaging and have to be relayed with low latency. To reduce the latency for the delivery of event reports, Sift employs a fixed-size contention window and a carefully-chosen, non-uniform probability distribution of transmitting in each slot within the window. If no node starts to transmit in the first time slot in the window, then each node increases its transmission probability exponentially for the next time slot.

Sift achieves very low latency for many traffic patterns compared to the 802.11 MAC protocol. However, there are some main drawbacks in Sift. Firstly, idle listening is increased due to listening to all time slots before sending. Secondly, overhearing is

increased. When there is an ongoing transmission, nodes must listen until the end in order to contend for the next transmission, which causes overhearing. Thirdly, it needs a system wide time synchronization in time slotted contention windows, which increases implementation complexity.

Similarly, in [17], a non-persistent CSMA protocol (CSMA/ p^*) has been proposed with a carefully chosen non-uniform probability distribution p^* by which nodes randomly select contention time slots. It has been shown that CSMA/ p^* is optimal in the sense that p^* is the unique probability distribution that minimizes collisions between contending stations. While CSMA/ p^* provides optimal performance when the number of nodes for contention, N is known, it can achieve sub-optimal performance when N is unknown. The optimal probability distribution works only when nodes always have data to transmit and they are synchronized for the channel access. When data arrival to a node is highly random and nodes cannot sense for data transmission among each other in multi-hop scenario, its performance degrades to that of CSMA scheme.

3.2.8 Summary of Contention-Based MAC Protocols

In summary, contention-based MAC protocols operate on demand basis. Sensor nodes that want to communicate with peers compete for the access to the medium. By nature, the contention-based protocols inherit good scalability and adaptability necessary to support topology change. Unfortunately, its major drawback is energy inefficiency due to idle listening, collision, overhearing, and control-packet overhead. Nevertheless, it has also been observed based on aforementioned studies and analysis that being adaptive is the key to achieve efficient usage of energy. Two folds of adaptability can be considered: traffic adaptive and topology adaptive. *Traffic adaptive*, held by AC-MAC, requires the communication channel to be available only when there is traffic, and channel bandwidth allocated for a node is proportional to the traffic amount it has. While *topology adaptive* employs the minimum number of nodes and power levels to construct the communication system. These adaptabilities are very much related with characteristics and requirements of applications.

By adding an *adaptive* duty cycle, AC-MAC further improves latency and throughput while remaining as energy-efficient as S-MAC. By using *adaptive* duty cycle, i.e. varying active window adapting to different traffic loads, T-MAC mitigates

two problems in S-MAC – the high latency and the low sensitivity to varying traffic loads. TEA-MAC is augmented by asTEA-MAC through introducing two *adaptive* factors – *Message counter* and *adjustable sleep-time* which reflects number of messages during waking period, and adapts to varying traffic loads, respectively; hence reducing energy waste and increasing throughput.

Key design points and advantages and disadvantages of the contention-based MAC protocols surveyed have been summarized in [Table 3-1](#).

Table 3-1: A Summary of Contention-based MAC Protocols for WSNs

WSN MAC Protocol	Targeting Application	Contention Resolution/ Scheduling Algorithm or Key Design Points	Advantages	Disadvantages and Room for Improvement
PAMAS [11]	• Multi-hop	<ul style="list-style-type: none"> • Use separate channel for control and data transmission • Use probe signal to avoid overhearing 	<ul style="list-style-type: none"> • Separate control and data channel do not interfere neighbors' data transmission 	<ul style="list-style-type: none"> • Complex channel allocation increases the energy consumption and the device cost • Not address the issue of reducing idle listening
S-MAC [12]	<ul style="list-style-type: none"> • Bursty event • Multi-hop 	<ul style="list-style-type: none"> • Fixed low duty cycle • Choose, maintain and synchronize schedules over virtual clusters • Use physical/virtual carrier sense with randomized carrier sense time, RTS/CTS exchange and NAV to avoid overhearing • Message Passing 	<ul style="list-style-type: none"> • Low duty cycle to save energy • Virtual clusters to support scalability and self-configuration • Overhearing avoidance to save energy • Message passing to reduce contention latency 	<ul style="list-style-type: none"> • Sleep latency due to periodic sleep • Fixed duty cycle not adaptive to dynamic traffic loads
S-MAC-AL [10]	<ul style="list-style-type: none"> • Bursty event • Multi-hop 	<ul style="list-style-type: none"> • Fixed low duty cycle • Choose, maintain and synchronize schedules over virtual clusters • Use physical/virtual carrier sense with randomized carrier sense time, RTS/CTS exchange and NAV to avoid overhearing • Message Passing • Adaptive Listening 	<ul style="list-style-type: none"> • Low duty cycle to save energy • Virtual clusters to support scalability and self-configuration • Overhearing avoidance to save energy • Message passing to reduce contention latency • Adaptive listening to improve latency in multi-hop message relay 	<ul style="list-style-type: none"> • Adaptive listening increase energy consumption and lower delivery ratio • Fixed duty cycle not adaptive to dynamic traffic loads
AC-MAC [13]	<ul style="list-style-type: none"> • Target-tracking with dynamic traffic loads • Multi-hop 	<ul style="list-style-type: none"> • <i>Adaptive duty cycle</i> (ADC) with $R_i = f(N_i)$ determining cycle time. $f(.)$ is application-specific function 	<ul style="list-style-type: none"> • Adaptive to dynamic traffic loads and improve end-to-end latency in multi-hop WSN 	<ul style="list-style-type: none"> • Sleep latency due to periodic sleep • It is hard to precisely determine the $f(.)$
T-MAC [14]	• Dynamic traffic loads	<ul style="list-style-type: none"> • <i>Adaptive duty cycle</i> (ADC) with Timeout mechanism 	<ul style="list-style-type: none"> • Adaptive to dynamic traffic loads to save 	<ul style="list-style-type: none"> • ADC causes reduced throughput and

	<ul style="list-style-type: none"> in time and location • Multi-hop 	<ul style="list-style-type: none"> dynamically ending the active part • <i>Future request-to-send</i> (FRTS) • Full-buffer priority with threshold control 	<ul style="list-style-type: none"> more energy 	<ul style="list-style-type: none"> increased latency • It is hard to distinguish the communication pattern of a live WSN
TEA-MAC [15]	<ul style="list-style-type: none"> • Event-driven • HTL • LTL 	<ul style="list-style-type: none"> • Simplified Structure of IEEE 802.11 MAC Frames • Wake counter/Sleep counter • Sand-clock scheme / Wake-time sharing scheme 	<ul style="list-style-type: none"> • Energy conservation obtained without sacrificing throughput and latency • Balanced performance • Easy-to-implement algorithm 	<ul style="list-style-type: none"> • Sand-clock type of counters requires strict timing, thus frequent time synchronization needed to eliminate impacts of clock drift
asTEA-MAC [9]	<ul style="list-style-type: none"> • Event-driven • HTL • LTL 	<ul style="list-style-type: none"> • Simplified Structure of IEEE 802.11 MAC Frames • Wake counter/Sleep counter/Message counter • Sand-clock scheme / Wake-time sharing scheme / Sleep-time adjustment scheme 	<ul style="list-style-type: none"> • Scalable to node change • Adaptive to dynamic traffic loads 	<ul style="list-style-type: none"> • Sand-clock type of counters requires strict timing, thus frequent time synchronization needed to eliminate impacts of clock drift
Sift [16]	<ul style="list-style-type: none"> • Event-driven • High node-density • Multi-hop 	<ul style="list-style-type: none"> • Increasing, non-uniform probability distribution within a fixed contention window 	<ul style="list-style-type: none"> • Scalable to the network size • Very low latency for backlogged messages from high node-density area 	<ul style="list-style-type: none"> • Idle listening and overhearing increased, energy increased slightly • Implementation complexity increased due to the need of system wide time synchronization
CSMA/p * [17]	<ul style="list-style-type: none"> • Event-driven • High node-density • Multi-hop 	<ul style="list-style-type: none"> • Optimal backoff distribution p^* for random selection of contention time slots 	<ul style="list-style-type: none"> • Very low latency for backlogged messages from high node-density area • Collision minimized 	<ul style="list-style-type: none"> • Energy increased • Implementation complexity increased

Legend:

CB: Contention-Based MAC protocol

SB: Scheduling-Based MAC protocol

LTL: Low Traffic Load (message interval > 10s)

HTL: High Traffic Load (message interval < 10s)

-: Not available

3.3 Scheduling-Based MAC Protocols

Collision of control or data packets is one of major reasons of energy waste. When a node receives more than one packet at the same time, there are two scenarios: (1) the two packets' signal strength is at the similar level, making the node unable to cleanly receive either of the packets. These two packets are called collided. (2) The signal strength of one of packets is stronger than that of another packet, making the node able to capture the stronger signal and cleanly receive the packet with the stronger signal. This node is called captured node. All packets that cause the *collision* and packet from

the captured node have to be discarded and retransmissions of these packets are required, which increase the energy consumption [4]. The probability of collisions of control or data packets in any contention-based scheme increases with the traffic load, resulting in degraded channel utilization and reduced battery life. This motivates the need for establishing transmission schedules statically or dynamically to allow nodes to transmit data packets without collisions [18]. Scheduling-based MAC protocols and schemes can be implemented based on *time-division multiple-access* (TDMA), *frequency-division multiple-access* (FDMA), and *code-division multiple-access* (CDMA) techniques, respectively [9]. Most of the scheduling schemes that have been proposed and studied in the context of WSNs have employed TDMA techniques. Time slots assignment can be performed in two approaches: centralized approach and decentralized (distributed) approach. Therefore, scheduling-based MAC protocols can be further classified into *centralized scheduling-based MAC protocol* and *distributed scheduling-based MAC protocol*. In WSNs, the distributed scheduling schemes have been the predominant design for MAC protocols.

3.3.1 TRAMA

Traffic-Adaptive MAC (TRAMA) [18] is a distributed scheduling MAC protocol based on TDMA technology. It allows for flexible and dynamic scheduling of time slots based on the traffic load. It employs a traffic adaptive distributed election scheme to select receivers based on the schedules announced by transmitters. Nodes exchange their two-hop neighborhood information and the transmission schedules with specification on which nodes are the intended receivers of their traffic in chronological order, and then select the nodes that should transmit and receive during each time slot [18].

TRAMA has some key points in its protocol design:

(1) TRAMA assumes a single, time-slotted channel for both data and signalling transmissions. Time is organized as random-access periods (*signalling slots*) and scheduled-access periods (*transmission slots*). During random-access periods, nodes perform contention-based channel acquisition and thus signalling packets are prone to collision. During scheduled-access periods, nodes perform collision-free data exchange and schedule propagation. (2) *Neighbor protocol (NP)*: Node exchange one-hop neighboring information during signalling slots. The signalling slots are sufficiently

long to ensure that this information is reliably transmitted and all nodes have consistent two-hop neighborhood information. (3) *Schedule exchange protocol (SEP)*: each node publishes its schedule during its last winning slot in each epoch. The schedule uses bitmaps to efficiently denote the intended recipients (unicast or broadcast) for each of its future packets in the next epoch. If the number of packets that a node wishes to send is less than the number of winning slots in an epoch, it gives up the remaining slots (which are packed at the end together between the last used winning slot (called changeover slot) and the final schedule propagation slot). The changeover slot is used for synchronization, which requires all neighbors to wake-up to listen to it. The node sleeps at all times when it is not required to transmit or receive. (4) *Adaptive election algorithm (AEA)*: it uses a hash function based on node IDs and time to ensure that there is unique ordering of node priority among all nodes within any two-hop region at each round. In the simplest version of this algorithm, a node transmits if and only if it has the highest priority among all nodes within its two-hop neighborhood. However, to prevent the waste of a slot, if there is no data to send, TRAMA implements a sophisticated AEA algorithm to allow for slot reuse.

Advantages – Significant energy savings can be achieved due to higher percentage of sleep time, up to 87% of sleep time subject to the offered load [18]. Higher throughput can also be achieved, around 40% over S-MAC and CSMA and around 20% over 802.11, compared to contention-based protocols because it avoids collisions due to hidden nodes. Since the intended receivers are indicated by bitmaps, each of whose bits corresponds to one particular receiver, less communication is needed for the multi-cast and broadcast communication.

Disadvantages – Delay is higher than random-selection protocols due to the scheduling overhead. It is observed that TRAMA algorithm is quite sophisticated, hence consuming processing and memory resources nontrivially, consequentially, consuming precious power of sensor nodes, which are inevitable in order to keep the protocol working. The length of a transmission slot is fixed based on the channel bandwidth and data size, and transmission slots are seven times longer than signalling slots; therefore signalling slots occupy at least 1/8 of operation cycle. TRAMA makes all nodes active in signalling slots to obtain consistent two-hop topology information across all nodes

and perform contention-based channel acquisition. Thus the duty cycle is at least $1/8$, i.e. 12.5%, which is a considerably high value.

The above factor of the transmission period over signalling period is one of important protocol parameters that have impacts on the duty cycle. It is observed that it is relative to the number of nodes in one-hop neighbourhood and two-hop neighbourhood. It is highly expected to have an analytical model to quantify the relationship between the factor of transmission slots over signalling slots and the number of nodes in one-hop/two-hop neighbourhood. In addition, based on the above analysis, it can be seen that TRAMA is well suited for applications that are not delay sensitive but require high delivery guarantees and energy efficiency. A typical example of such applications is one for periodic data collection and monitoring over WSNs. In addition, TRAMA differs from S-MAC (which also provides explicit energy conservation mechanisms) in two fundamental ways: (1) TRAMA is inherently collision-free as its MAC mechanism is schedule-based as opposed to S-MAC's which is contention-based; and (2) TRAMA uses an adaptive, dynamic approach based on current traffic patterns to switch nodes to low power mode, while S-MAC's scheme is static based on a pre-defined duty cycle.

3.3.2 FLAMA

Schedule-based protocols exhibit inherently higher delivery delays compared to contention-based approaches. In TRAMA, this is exacerbated by the need to propagate schedule information. FLow-Aware MAC (FLAMA) [19] avoids explicit information exchange and employs a much simpler election algorithm than that of TRAMA.

FLAMA is a distributed scheduling MAC protocol based on TDMA that leverages traffic predictability in applications over WSNs. It has the following major features: (1) the distributed maintenance of energy efficient, collision-free transmission schedules based on two-hop neighbourhood information and implicit traffic information; (2) low transmission delays with limited processing and storage requirements; and (3) robust operation that accommodate topology changes. FLAMA achieves significant smaller delays compared to TRAMA with significant improvement in energy savings and reliability.

FLAMA has some key points in its protocol design:

FLAMA keeps some of basic points as the same as TRAMA in protocol design. Differentiating from TRAMA, (1) It assumes a single, time-slotted channel for both data and signalling transmissions. (2) Time is organized as random-access periods (*signalling slots*) and scheduled-access periods (*transmission slots*). (3) During random-access periods, nodes contend for channel, perform time synchronization, and neighbour discovery but using implicit traffic information exchange. (4) During scheduled-access periods, nodes perform collision-free data exchange. Differentially, (5) FLAMA does not require explicit schedule announcements during the scheduled-access periods. Alternatively, it exchanges application-specific traffic information among nodes during random access periods to reflect application-specific traffic patterns. (6) FLAMA realizes a proportional time slots assignment in accordance with the amount of traffic generated/forwarded by a node.

It is believed that traffic information can be determined by having the application explicitly specify its traffic characteristics, or by using traffic prediction techniques at each node. [19]. For example, in environmental monitoring, periodic data gathering generates data streams over a collection tree rooted at the information sink and spanning all relevant nodes. When sending data, each node transmits to the upstream next-hop towards the sink. This information could be used. e.g.: FLAMA uses the concept of flows to characterize application traffic patterns. Flows represent one-hop traffic information and specify the transmitter, the receiver(s), and the rate at which packets are sent. FLAMA uses flow-based traffic information to determine transmission schedules.

Advantages – Simulation results show that FLAMA significantly outperforms S-MAC in terms of delivery ratio, drop rate, and energy efficiency [19]. FLAMA achieves better delivery ratio than TRAMA as well. This is due to the factor that FLAMA performs traffic-adaptive scheduling without incurring much overhead. Also it does not exchange any control packet during the schedule-access period and hence the channel contention probability is less.

Disadvantages – It is noted that FLAMA experiences longer delay than S-MAC. The level of the delay depends on the number of two-hop nodes.

However, the above disadvantage is compensated by much higher reliability. Hence end-to-end applications perceive the delay much less in FLAMA than in S-MAC. When

application-specific characteristics and requirements are considered in designing MAC protocols, the performance of the designed protocol often outperforms that of the protocols designed for common purposes.

3.3.3 SMACS

Self-organizing Medium Access Control for Sensor Network (SMACS) [20] is another distributed scheduling MAC protocol based on TDMA. SMACS develops four algorithms to tackle different aspects of issue. SMACS provides scalability with its self-organization capability, and make a sensor network to be able to operate under very dynamic condition, such as during startup, steady state and failure, as well as unattended scenarios. SMACS uses TDMA scheme to establish transmission/reception communication schedules. At the same time, it employs a superframe structure and time synchronization to discover neighbors. Combining neighborhood discovery and TDMA schedule assignment, SMACS is able to form a connected network without the need for any local or global master node. A communication link consists of a pair of time slots on a randomly selected, but fixed frequency or frequency hopping sequence. It is a feasible design in WSNs, since the available bandwidth can be much higher than the maximum data rate for an individual sensor node. Therefore, it can avoid the necessity for network-wide synchronization. SMACS also achieves energy conservation by using a random wake-up schedule in the connection phase and by turning the radio off during idle time slots.

Eavesdrop-And-Register (EAR) protocol [20], developed in the same suit of algorithm SMACS, attempts to offer continuous service to the sensor nodes that join stationary WSNs. The EAR protocol is transparent to the SMACS, so that the SMACS is functional until the introduction of mobile nodes into the network. Sequential Assignment Routing (SAR) supports multi-hop routing that is robust to failure. Single-Winner-Election (SWE), Multi-Winner-Election (MWE) and Central Node Election (CNE) handle signal and data transfer.

This set of algorithms separately addresses scalability, mobility, robust routing in multi-hop scenario and cooperative signal processing, which are important issues in certain applications. However, how to collectively employ these algorithms to reach different prospects of goal without sacrificing energy-efficiency is a great challenge

3.3.4 SRSA

Self-Reorganizing Slot Allocation (SRSA) [21] scheme also adopts self-organizing approach in the MAC protocol design, and targets at the scenarios with low traffic load such as monitoring applications. To take the advantages of clustering technique, SRSA is designed to support the operation of cluster topology over WSNs. It, at the same time, offers mechanisms for reducing inter-cluster TDMA interference without having to use CDMA or FDMA. By the SRSA, the above objective can be achieved through a self-reorganization evolution process consisting of the following main steps. (1) It initiates cluster nodes' communication with an initial random TDMA allocation. (2) It finds out carrier sense (CS) and hidden node (HN) collisions through carrier sensing affected cluster heads. (3) It adaptively changes the slot allocation schedule locally based on the feedback received from the collisions experienced by the local nodes within a cluster so that the number of CS and HN collisions can be reduced in the subsequent TDMA frames and eventually minimized to a stable range. (4) The TDMA schedule derived from this self-learning is then downloaded by the cluster head to its cluster members at the beginning of the next frame. This iterative process is very similar to that of the self-learning in artificial neural network, through which system obtains an adaptive capacity.

Advantages – SRSA leverages the advantages of node clustering to reduce the overall network-wide transmission energy expenditure by routing the aggregated data through the local cluster heads, at the same time, controls inter-cluster interference without using CDMA or FDMA schemes across clusters which would incur additional costs in terms of spectrum usage and network interface complexity for low-cost sensor nodes.

Disadvantages – SRSA's feedback based self-adaptive slot reorganization algorithm converges in slow speed. When only active detection turns on, it takes 1225 seconds to converge. Even both active and passive detection turn on, it still take 98 seconds to converge. Its throughput and latency perform fairly. Such characteristics confine its usage within non-dynamic and larger delivery latency tolerable monitoring applications.

3.3.5 R-MAC

Among the causes of energy waste identified in [10], three main causes are addressed in Reservation-MAC (R-MAC) [22]: packets collision, idle listening and unnecessary

overhearing of packets. R-MAC uses two separate periods during communication process. In the first period, nodes compete for time slots reservation for the future transmission. In the second period, each node follows the reserved time slots to wake up and carry out the data transmission and reception, and go back to the sleep mode immediately after the reserved time slots. R-MAC claims it outperforms S-MAC and T-MAC in term of energy saving, but actually only for the case of high traffic load (above 15 packets/s). In the traffic load below it, S-MAC and T-MAC are more in favourite of energy saving. It is not a surprise that R-MAC outperforms S-MAC and T-MAC in high traffic load situation since S-MAC and T-MAC are designated for low traffic load rather than high traffic load sensor networks.

3.3.6 RMAC

Although S-MAC-AL has achieved better energy conservation and significantly improved the sleep latency in a multi-hop network, two problems remain: firstly it still needs multiple operational cycles to complete multi-hop data delivery, and such an end-to-end delivery latency may increase substantially with increasing of the hop number. Secondly, cluster nodes need to periodically synchronize their common sleep schedule. Such synchronization frequency will increase as traffic load increases, and communication demands may suddenly increase in a burst in a small area such as in event monitoring applications. The probability of network contention thus increases significantly. Routing-enhanced MAC (RMAC) [23] protocol resolves the above problems without sacrificing the energy efficiency.

Specifically, RMAC uses cross-layer routing information to enable a data packet to deliver across multiple hops in a single operational cycle. RMAC uses pioneer frame (PION) to serve for both RTS and CTS frame. A PION control frame contains various routing-aware information, such as current node's address, the next-hop address, the final destination address, the number of hops that the PION has travelled and the duration of the transmission. This dual function allows a node to send a CTS to its upstream node and inform the node of its readiness to receive data; simultaneously, send a RTS to its downstream node and inform it of its willingness to send new data. This dual function makes the multi-hop relaying of PIONs very efficient. **Figure 3-13** shows an overview of the operation procedure that data sender S sends DATA to destination C 3-hop away.

Figure 3-13: RMAC Overview

An operational cycle of a sensor node is divided into three stages: SYNC, DATA and SLEEP. In SYNC period, sensor nodes synchronize the clocks. The DATA period is used only to send and receive PION frames, setting up the schedule for the actual data transmission. Data frames are transmitted and received only during the SLEEP period. Upon the schedule is set up, only senders and receivers will wakeup just-in-time (JIT) for the scheduled data transfer in SLEEP period. All other nodes in the multi-hop path go to sleep mode to save energy. After finishing data transfer and receiving an ACK from receiver, sender immediately goes to sleep mode to save energy; while receiver, after a short SIFS, immediately relays the data frame to its next hop. Similarly, after relaying the data and receiving an ACK from the next receiver, this receiver immediately goes to sleep mode to save energy. This entire process is repeated until the final destination is reached.

3.3.7 DW-MAC

Demand wakeup MAC (DW-MAC) [24] is a further effort of RMAC [23] to improve delivery latency, increase packet delivery ratio (fraction of data packets successfully delivered) and increase energy efficiency, hence contributing to the prolonging of sensor networks' lifetime.

Similar to RMAC, DW-MAC divides its operational cycle into three stages as well: SYNC, DATA and SLEEP, and they serve as the same function as that in RMAC. In a Data period, nodes with pending data contend for channel access using CSMA/CA protocol. DW-MAC also replaces RTS/CTS with a single control frame called *scheduling frame* (SCH). Differentiating from PION frame in RMAC, however, SCH uniquely reserves the proportional interval of time in the following Sleep period for

transmitting and receiving the pending data packets. In RMAC, if there are two hidden resources succeeding in scheduling through PIONs in a DATA period, they will collide at the beginning of the next SLEEP period. In order to avoid such collisions, DW-MAC sets up one-to-one mapping between a DATA period and the following SLEEP period according to proportional mapping function expressed in equation (3-1):

$$T_i^S = T_i^D \cdot \frac{T_{Sleep}}{T_{Data}} \quad (3-1)$$

Where T_i^D denotes the time difference between a specific time instance t_i in a DATA period and the beginning of that DATA period, and T_i^S denotes the time difference between the start of the subsequent SLEEP period and the corresponding mapped time instance during the SLEEP period. With this mapping function, a sender and its intended receiver(s) can uniquely determine the starting point for data packet transmission in a SLEEP period. Figure 3-14 depicts an overview of scheduling in DW-MAC based on this one-to-one mapping, where sender node A sends DATA to destination node C.

Figure 3-14: DW-MAC Overview

DW-MAC benefits from the following aspects: (1) DW-MAC schedules data transmissions without exchanging any time information, which eliminates schedule overhead; (2) Just-in-time (JIT) wake-up and go-to-sleep in Sleep period eliminate unnecessary listening for senders and receivers, thus saving energy. (3) One-to-one proportional mapping for a node's Data period and Sleep period ensures that data

transmissions by nodes that wakeup during the Sleep period do not collide at their intended receivers, which increases channel efficiency and packet delivery ratio, as well as energy efficiency. However, it is worthy to point out that DW-MAC adopts fixed duty-cycle in the scheme design, which makes it not to be adaptive to the varying traffic loads.

3.3.8 Summary of Schedule-Based MAC Protocols

In summary, scheduling-based MAC protocol may be based on TDMA, FDMA and CDMA. Currently, most of the scheduling-based MAC schemes use TDMA's time slots reservation and scheduling. The greatest advantage of scheduling-based MAC protocols is energy efficiency due to its features in built-in duty cycle, collision-free and absence of control-packet overhead. Nevertheless, it has obvious disadvantage in scalability and adaptability. Since a scheduling-based MAC scheme needs to rigidly follow the assigned communication time slots, it requires frame synchronization which involves the complicated tasks of slot allocation and schedule maintenance. These tasks create additional latency that decreases the throughput and increases the non-productive consumption of energy. When an existing node of a WSN leaves the network or a new node joins the network, the frame size and time slot assignment have to be revised in order for the network to adapt to the change. Key design points and performance features of the schedule-based MAC protocols surveyed have been summarized in [Table 3-2](#).

Table 3-2: A Summary of Scheduling-based MAC Protocols for WSNs

WSN MAC Protocol	Target Application	Contention Resolution/ Scheduling Algorithm or Key Design Points	Advantages	Disadvantages and Room for Improvement
TRAMA [18]	<ul style="list-style-type: none"> •Event-detection and monitoring •Multi-hop 	<ul style="list-style-type: none"> •Neighbor Protocol (NP) •Schedule Exchange Protocol (SEP) •Adaptive Election Algorithm (AEA) 	<ul style="list-style-type: none"> •Collision-free for data transmission •Higher percentage of sleep time •High throughput •Less communication needed for unicast and broadcast 	<ul style="list-style-type: none"> •Complexity of election algorithms •Scheduling overhead causes higher delay
FLAMA [19]	<ul style="list-style-type: none"> •Event-detection and monitoring •Data gathering in multi-hop tree structure 	<ul style="list-style-type: none"> •Distributed Election Algorithm (DEA) 	<ul style="list-style-type: none"> •Collision-free for data transmission •Higher percentage of sleep time •High throughput •Simplified election algorithms •Higher end-to-end 	<ul style="list-style-type: none"> •Hard-coded routing information lacks of scalability and self-configurability •Long initial network formations process •Delay depends on the number of two-

			reliability • Improved latency	hop nodes
SMACS [20]	<ul style="list-style-type: none"> • Dynamic condition with failure nodes • Multi-hop 	<ul style="list-style-type: none"> • Combining neighborhood discovery and TDMA schedule assignment • Using a random wake-up schedule • Four algorithms separately address scalability, mobility, multi-hop routing and cooperative signal processing issues 	<ul style="list-style-type: none"> • Scalability with self-organization • Slow mobility in a subset of nodes • Robust routing for the multi-hop networks 	<ul style="list-style-type: none"> • How to jointly address scalability, mobility, robust routing for multi-hop network together with energy-efficiency is a challenge
SRSA [21]	<ul style="list-style-type: none"> • Monitoring application with low traffic loads • Multi-hop 	<ul style="list-style-type: none"> • Self-reorganization evolution process • Mechanisms for reducing inter-cluster TDMA interference without having to use CDMA or FDMA 	<ul style="list-style-type: none"> • Does not require any global or neighborhood synchronization • Low energy overhead • Low packet drop rate 	<ul style="list-style-type: none"> • Increased latency
R-MAC [22]	<ul style="list-style-type: none"> • Monitoring application • Multi-hop 	<ul style="list-style-type: none"> • Avoids data collisions through reservation of time slots • Adjusts the duration of the sleep and active periods according to the traffic loads 	<ul style="list-style-type: none"> • Perform more energy-efficiently in high traffic loads • Keep low data collision rate in both low traffic loads and high traffic loads 	<ul style="list-style-type: none"> • Proposed GTS packets likely cause sleep latency issue in the neighboring nodes, which is not evaluated
RMAC [23]	<ul style="list-style-type: none"> • Monitoring application with burst events • Multi-hop 	<ul style="list-style-type: none"> • Cross-layer routing information • Dual function control frame PION to replace RTS/CTS 	<ul style="list-style-type: none"> • The schedule fully dynamically formed-up by demand • Increased throughput, reduced latency and energy consumption 	<ul style="list-style-type: none"> • Only neighboring-node wide synchronization may hinder the latency reduction • Fixed duty-cycle is not adaptive to varying traffic loads
DW-MAC [24]	<ul style="list-style-type: none"> • Monitoring application with burst events • Multi-hop 	<ul style="list-style-type: none"> • Cross-layer routing information • Dual function control frame SCH to replace RTS/CTS • One-to-one proportional mapping 	<ul style="list-style-type: none"> • Schedules data transmissions without exchanging any time information, thus eliminating schedule overhead • Just-in-time (JIT) wakeup and go-to-sleep eliminate idle listening 	<ul style="list-style-type: none"> • Fixed duty-cycle is not adaptive to varying traffic loads

Legend:

CB: Contention-Based MAC protocol

SB: Scheduling-Based MAC protocol

LTL: Low Traffic Load (message interval > 10s)

HTL: High Traffic Load (message interval < 10s)

-: Not available

3.4 Hybrid MAC Protocols

Previous two types of MAC protocols have their advantages and disadvantages as described. Contention-based MAC protocols fundamentally rely on CSMA with some

control signals support, thus presenting simplicity, flexibility, and robustness. Scheduling-based MAC protocols, for most of cases, use TDMA, thus exhibit particularities such as collision-free, high channel utilization, low latency, etc. Hybrid MAC protocol is to combine the strengths of the controlled access approaches and the random access techniques while offsetting their disadvantages, so as to behave as dual-characteristic protocol and achieve better performance.

3.4.1 Z-MAC

Zebra MAC (Z-MAC) [25] is a hybrid MAC scheme, which is suitable for the applications where expected data rate and two-hop contention are medium or high. The main feature of Z-MAC is its adaptability to the level of contention in the network so that under low traffic load with less contention, it behaves like a CSMA protocol while under heavier traffic load with more contentions, it works like a TDMA based scheduling protocol. Z-MAC uses the CSMA scheme as the baseline MAC protocol and TDMA scheme as a “hint” to enhance contention resolution.

Z-MAC has some key points in the protocol design. (1) Z-MAC uses a Distributed Randomized TDMA Scheduling algorithm (DRAND) [26], an efficient scalable channel scheduling algorithm, for time slots assignment on the channel. (2) After the time slot assignment, each node reuses its assigned slot periodically in each frame. A node with an assigned time slot is called *owner* of that time slot, and others are called *non-owners* of that slot. (3) Before transmission, nodes always perform carrier sensing and transmit a packet when the channel is clear. However, an owner of that time slot always has higher priority over its non-owners in accessing the channel in that time slot. (4) In addition, four key procedures need to be gone through once in setup phase: (a) neighbor discovery, (b) slot assignment, (c) local frame exchange and (d) global time synchronization.

Z-MAC has some key advantages: By mixing the TDMA and the CSMA scheme, Z-MAC becomes more robust to timing failures, time-varying channel conditions, slot assignment failures and topology changes than a standalone TDMA. Z-MAC may easily and rapidly adapt to dynamic traffic conditions by implicitly switching between CSMA and TDMA schemes depending on the level of contention. Z-MAC adopts the preamble sampling technique, which is a major technique to lower duty cycle, during CSMA

portion of the protocol to save energy. At the same time, Z-MAC can save more energy by approximating a CSMA protocol under light traffic load and a TDMA protocol under high traffic load. CSMA is contention-based protocol that is suitable for light traffic load. Based on CSMA, a node will wake-up to contend for channel and send data only when it has data; when node does not transmit any data, it sleeps to save energy. TDMA is a scheduling-based protocol that is more suitable for high traffic load. The pre-scheduled periodical time slots in TDMA can help disseminate high traffic load more efficiently without much overhead; therefore increasing throughput. TDMA protocol also eliminates collision incurred in contention-based protocol; therefore saving energy. Z-MAC can achieve a better performance in terms of throughput, fairness, energy efficiency. The main drawback of Z-MAC is that it has no protection from hidden node problem since it uses CSMA scheme as its baseline MAC.

3.4.2 HyMAC

Unlike previously analysed hybrid MAC protocols that switch between TDMA based scheduling protocol and CSMA based contention protocol based on the level of contention in the network, such as Z-MAC, HyMAC [27] is a hybrid MAC layer protocol that combines the strengths of both TDMA and FDMA scheme in the protocol design. HyMAC takes advantages of multiple frequencies provided in the radio component of recent sensor node platforms. Strictly speaking, HyMAC should be categorized into scheduling-based protocol since it uses TDMA and FDMA scheme in its scheduling algorithm.

The algorithm performs a breadth first search (BFS) to the tree-based nodes structure, and assigns a different time slot and different frequency for the one-hop and two-hop sibling neighbours, and the same time slot but different frequency for the one-hop and two-hop non-sibling neighbours. Such assignments eliminate conflicting neighbours and maximize the utilization of time and frequency resources. Feasibility of HyMAC in practice depends on the maximum number of frequencies required for the operation in dense networks. While the author of [27] has observed that the number of frequencies needed is well below the number of channels provided in practical WSN radio components such as CC2420, no performance evaluation and comparison has been done against the typical MAC protocols, such as S-MAC, T-MAC, Z-MAC or B-MAC. A minor comparison done separately against MMSN [28] and RT-Link [29] is

limited only to the number of potential conflicts and number of required time slots. More important performance metric, such as energy consumption, throughput, packet delivery ratio, latency, etc. are not reported.

3.4.3 Funnelling-MAC

Funnelling-MAC [54] is a hybrid MAC protocol that aims to tackle the funnelling effect in sink-oriented many-to-one multi-hop WSNs. **Figure 3-15** depicts the funnelling effect in sensor networks where the region far away from sink (entry region of the funnelling) presents lesser node density and therefore lighter traffic load; the region closer to the sink (intensity region of the funnelling) presents higher node density and therefore heavy traffic load. When the traffics flow from entry region to intensity region and reach sink, the significant packet collision, congestion and loss will occur, this is referred to *funnelling effect*. Funnelling-MAC uses pure CSMA scheme to handle lighter traffics in the entry region, and uses hybrid TDMA/CSMA scheme to tackle heavier traffics in the intensity region. Funnelling-MAC lets sink provide light-weight clock synchronization for TDMA scheduling in the intensity region, dynamically adjusts the depth of TDMA operating in the intensity region by controlling the transmission power of the broadcast beacon. All sensor nodes perform CSMA by default unless they receive a beacon. Upon receiving a beacon, they consider themselves to be in the intensity region and then deemed *funnelling-nodes (f-nodes)*. These nodes can perform TDMA while the nodes that do not receive the beacon (e.g., those nodes outside the intensity region) perform CSMA.

Figure 3-15: Funnelling effect in sensor networks [54]

Experimental results show that Funnelling-MAC significantly outperforms other typical protocols such as Z-MAC and B-MAC in terms of throughput and packet loss. In addition, the energy-efficiency is also slightly better than that of Z-MAC and B-MAC. However it is worth of noting that Funnelling-MAC achieves the better performances under an assumption that it has a very powerful sink. The sink has to manage TDMA scheduling of sensor events in the intensity region, compute and maintain the depth of the intensity region, which require more computational capacity and energy than simple sensors. This might limit the scope of application of the protocol.

3.4.4 Crankshaft

Crankshaft [101] is another hybrid MAC protocol that targeted at dense wireless sensor networks. Crankshaft protocol divides time into frames, which is further divided into two types of slots: broadcast slots (Contention-window) and unicast slots (Message Exchange Window) as shown in [Figure 3-16](#). During a broadcast slot all nodes wake up to listen for an incoming message. Any node with a broadcast message to send contends with all other nodes in the contention-window. Once a node wins the contention, it uses destination node's MAC address to determine wake up slot of the destination node and starts to send the message to the destination node. The receiver polls the preamble sent by sender at the fixed offsets from the start of a frame and gets the rendezvous of the message transmission. Both sender and receiver base on the rendezvous to transmit and receive the message in message exchange window. Crankshaft uses a Data/Ack sequence for unicast messages.

The channel polling mechanism used in Crankshaft is very similar with the scheduled channel polling mechanism in SCP-MAC [32] where only contending senders are awake during the contention window and only the winner of the contention is awake for receiver's poll. This is a very energy-efficient contention resolution.

The simulations results show that the Crankshaft protocol can provide good convergecast performance at low energy consumption in dense networks therefore being suitable for long-lived monitoring applications, where energy efficiency is a key. The trade-off is that broadcast bandwidth is significantly reduced, which means that to use the Crankshaft protocol effectively, broadcast flooding must be minimised. Another

downside is that Crankshaft protocol as proposed has a very rigid structure, which limits the applicability of the protocol.

Figure 3-16: Contention and message exchange in the Crankshaft protocol [101]

3.4.5 Summary of Hybrid MAC Protocols

In summary, hybrid MAC protocols harmonize the contention-based and scheduling-based schemes while compensating their disadvantages each other under different traffic load conditions. Contention-based schemes possess simplicity, flexibility, robustness, and low latency, hence enabling nodes to be adaptable to network topology change with support of some control signals. On the other hand, scheduling-based schemes present collision-free, high channel utilization and throughput, which are suitable for high traffic load scenarios. A switching mechanism can be designed so that a hybrid MAC protocol can switch itself between the contention-based and schedule-based protocols, and adapts its behavior to the level of traffic loads which can be presented in different forms of the sensed traffic intensity on real-time. Though hybrid MAC protocols seem an ideal approach to achieve optimal performance of the MAC protocol design, this advantage comes at the cost of protocol complexity wherein an accurate mapping from traffic loads to the switching point might not be achieved easily.

Key design points and performance features of the hybrid MAC protocols surveyed have been summarized in [Table 3-3](#).

Table 3-3: A Summary of Hybrid MAC Protocols for WSNs

WSN MAC Protocol	Target Application	Contention Resolution/ Scheduling Algorithm or Key Design Points	Advantages	Disadvantages and Room for Improvement
Z-MAC [25]	•Medium to high data rate with dynamic	•CSMA as baseline MAC and TDMA as “hint” •Owner and non-owners of slot	•Exhibits adaptability to varying traffic loads •Robust to time-varying channel conditions,	•Slot assignment and synchronization needed for Z-MAC may lead high costs

	<ul style="list-style-type: none"> • traffic loads • Multi-hop 		<ul style="list-style-type: none"> • slot assignment failures and topology changes 	<ul style="list-style-type: none"> • especially when significant network changes occur frequently
HYMA C [27]	<ul style="list-style-type: none"> • Data gathering with tree-based nodes structure 	<ul style="list-style-type: none"> • Combine TDMS and FDMS • Breadth First Search (BFS) for time slot and frequency assignment 	<ul style="list-style-type: none"> • Scheduling algorithm for time slot and frequency assignment is easy to implement 	<ul style="list-style-type: none"> • No performance evaluation and comparison has been done against the typical MAC protocols
Funnelli ng-MAC [54]	<ul style="list-style-type: none"> • Sink-oriented many-to-one data gathering • Multi-hop 	<ul style="list-style-type: none"> • CSMA scheme to handle lighter traffics • Hybrid TDMA/CSMA scheme to tackle heavier traffics • Sink dynamically adjusts the boundary of the intensity region by controlling the transmission power of the broadcast beacon 	<ul style="list-style-type: none"> • Achieve high throughput and low packet loss in a fair good energy consumption 	<ul style="list-style-type: none"> • Require a very powerful sink that need to manage TDMA scheduling, compute and maintain the boundary of the intensity region, etc. • Limit the scope of application of the protocol
Crankshaft [101]	<ul style="list-style-type: none"> • Long-lived monitoring application with dense WSN 	<ul style="list-style-type: none"> • Hybrid CSMA and channel polling 	<ul style="list-style-type: none"> • Very low energy consumption 	<ul style="list-style-type: none"> • Broadcast bandwidth is significantly reduced

Legend:

CB: Contention-Based MAC protocol

SB: Scheduling-Based MAC protocol

LTL: Low Traffic Load (message interval > 10s)

HTL: High Traffic Load (message interval < 10s)

-: Not available

3.5 Preamble Sampling Based MAC Protocols

Idle listening has been found in 802.11 protocols to consume substantial energy, hence becomes a dominant reason to cause energy waste in most WSNs applications [30], [31], [32]. Lowering the radio duty cycle can reduce energy consumption. *Low Power Listening* (LPL), also called *preamble sampling*, is one of major techniques to lower duty cycle. In LPL, a sensor with data to send and its receiver usually do not know the sleeping schedule of each other, they rely on probe message and sampling to establish data transmission rendezvous. When a sender has data, it transmits a preamble that is at least as long as the Sleep period of the receiver. The receiver will wake up according to its duty cycle, detect the preamble, and stay awake to receive the data. This type of protocols is called preamble sampling or channel polling MAC protocols.

3.5.1 WiseMAC

Hoiydi proposed the “Spatial TDMA and CSMA with Preamble Sampling” protocol [33] in order to mitigate the idle listening. In this protocol, all sensor nodes have two

communication channels: data channel is accessed with TDMA method for transport of regular and frequent traffic; control channel is accessed with CSMA method for transport of sporadic signaling traffic. Such TDMA (Schedule-based) /CSMA (contention-based) hybrid approach brings about idle listening reduction and energy conservation. Wireless Sensor MAC (WiseMAC) [34] further minimizes the idle listening and prolongs the nodes' lifetime bases on its prior work [33]. WiseMAC mitigates idle listening by using non-persistent CSMA (np-CSMA) with preamble sampling. This allows low power communication without the need for explicit synchronization between the nodes. The receiver only wakes for a short time to sample the medium, thereby limiting idle listening.

However, predetermined fixed-length preamble would increase power consumption. To further avoid the power wastage due to idle listening, WiseMAC minimizes the length of the wake-up preamble by letting sender know the sampling schedule of receiver so that the sender can begin the preamble with minimized duration and data transmission only a short time before receiver will wake up, taking possible clock drifts into account. The nodes may learn and refresh their neighbors' sampling schedule during every data exchange as part of the Acknowledgment message. WiseMAC schedules transmissions so that the receiver's sampling time corresponds to the middle of the sender's preamble, as depicted in **Figure 3-17**. This reduces the energy consumed when sending a long preamble. In addition, for low traffic loads where the data frame is shorter than the preamble, WiseMAC repeats the data frame in place of the extended preamble. Receivers process this data frame in the way that if the node is not the intended recipient it returns to sleep; if the node is the recipient, it remains awake until the end of the transmission and sends an acknowledgment.

A key advantage of LPL technique is that the sender and receiver can be completely decoupled in their duty cycles yet be able to energy-efficiently find a rendezvous for data transmission. The simplicity of this design removes the need for schedules synchronization, and eliminates the overhead of synchronization. While LPL is simple, asynchronous, and energy-efficient, the long preamble has several inherent disadvantages: (1) it is suboptimal in terms of energy consumption at both the sender and receiver. The receiver has to wait the full period of the preamble before the data transmission can start, even if the receiver has woken up at the beginning of the

preamble, which wastes energy at both the sender and receiver. (2) It suffers from big overhearing problem, where all non-target receivers also wake up during the long preamble and stay awake for the period of the preamble to find out if the data packet is destined for them. This wastes energy of all non-target receivers for the transmission period of the sender. (3) The per-hop latency from the long preamble will be accumulated over multi-hop path and multi-hop latency will become quite significant.

Figure 3-17: WiseMAC scheme

3.5.2 CSMA-MPS

A CSMA-based Minimum Preamble Sampling MAC [101] was proposed to reduce the fairly long preamble incurred in CSMA-PS [103] and WiseMAC [34][104]. CSMA-MPS also combines the alternating transmitting and receiving mechanism of STEM [114] with the more efficient wake-ups of WiseMAC to increase the efficiency of wakeup mechanism and further optimize the preamble time to reduce the overhead at the transmitter and receiver. Compared to WiseMAC, the CSMA-MPS protocol requires only half preamble length of WiseMAC on average because the preamble is transmitted only as long as the sender node hits the receive window of the receiver, thus reducing the overhearing. In addition, periodic synchronization and network wide synchronization are not required by the nodes, which reduce the overhead of the protocol. At most each node keeps synchronization information from all its direct neighbours. While CSMA-MPS shortens the preamble, reduce the overhearing and save the energy consumption, but the latency and throughput performances are not good enough.

3.5.3 X-MAC

A Short Preamble MAC (X-MAC) [35], having realized the drawbacks of WiseMAC the abovementioned, proposes an approach based on CSMA-MPS to improve the WiseMAC in terms of energy consumption and latency reduction. (1) X-MAC embeds address information of the target receiver in the short preamble so that non-target receivers can quickly go back to sleep, which mitigates overhearing problem. (2) X-MAC uses a *strobed preamble* to allow the target receiver to interrupt the long preamble as soon as it wakes up and determines that it is the target receiver. This reduces the time and energy wasted when waiting for the entire preamble to complete. Experimental evaluation in a testbed shows that X-MAC achieves significant energy savings and per-hop latency reduction. X-MAC also can optionally use an automated algorithm for adapting the duty cycle of the nodes to best accommodate the traffic load in the network. Nevertheless, a major drawback of X-MAC is its unsuitability for delay sensitive networks.

3.5.4 B-MAC

MAC's re-configurability is essential for dynamic systems such as WSNs [36], where a lightweight and configurable MAC protocol has been proposed named as Berkeley-MAC (B-MAC), which is used as the default MAC protocol for Mica2. It provides a set of primitive operations through a flexible control interface which allows an application to implement its own MAC protocols via a well-defined interface. Several key mechanisms employed in B-MAC are worth mentioning. (1) B-MAC uses Clear Channel Assessment (CCA) algorithm and packet backoffs mechanism for channel arbitration and link layer acknowledgements for reliability. (2) B-MAC adopts an adaptive preamble sampling scheme in Low Power Listening (LPL) to reduce duty cycle, minimize idle listening and improve channel utilization. (3) B-MAC proposes an adaptive bidirectional interface that allows middleware services to reconfigure a set of parameters of an MAC protocol based on the current traffic load and protocol policy for WSN applications. Enabling or disabling CCA allows the MAC protocol to incline to a contention-based protocol or schedule-based MAC protocol. Configuring the backoffs scheme allows the transmission services to change the fairness and available throughput. Adjustable preamble length and check interval have direct impact on nodes' duty cycle and an optimal duty cycle for specific dense network can be easily achieved by setting a

suitable check interval in the protocol. By exploring the set of configurable parameters, the B-MAC protocol can make local policy decisions to optimize power consumption, latency, throughput, fairness or reliability.

It is shown in [36] that B-MAC scheme has higher throughput and better energy efficiency than those of S-MAC and T-MAC schemes in a wide array of network conditions, but does not trade-off fairness in the process. The B-MAC scheme, unlike Z-MAC and SCP-MAC, does not rely on the contention level to synergy the pair of contention-based and schedule-based approaches. It adopts a more proactive way to regulate and control the effects that the designed network intends to achieve through multiple parameters setting in the MAC protocol.

3.5.5 SCP-MAC

Inspired by LPL technique, Scheduled Channel Polling (SCP-MAC) [32] introduces a new approach to synchronize the polling time of all neighboring nodes to reduce idle listening and overhearing as well as transmitter's costs. The wakeup tone duration in SCP in **Figure 3-18** (b) is much shorter than that of the preamble, which costs much when traffic varies, before the data frame in LPL in **Figure 3-18** (a). The SCP-MAC works based on its prior work S-MAC [10] – a contention-based MAC and efficiently extends the way of broadcasting its schedule by discovering an optimal synchronization period and wakeup tone length that will minimize the overall energy consumption. To reduce the latency of the bursty traffic in multi-hop scenarios, SCP-MAC develops an adaptive channel polling mechanism by dynamically inserting additional, high-frequency polling slots to the nodes on the path as depicted in **Figure 3-19**, which allows them to stream packets quickly over multiple hops. Since SCP-MAC synergies its scheduled channel polling mechanism with contention-based MAC protocol, it can be categorized as a hybrid MAC protocol.

Figure 3-18: Sender and receiver synchronization schemes [32]

Figure 3-19: Adaptive channel polling and multi-hop streaming [32]

3.5.6 Summary of Preamble Sampling MAC Protocols

Preamble sampling based MAC protocols completely decouple the duty cycle of sender from that of receiver and yet able to link them with a data transmission rendezvous. This feature simplifies the design of the preamble-based protocol and removes the need for schedules synchronization, thus eliminating the protocol overhead. The energy wastes due to the overhearing problem and prolonged preamble can be mitigated using dedicated embedded address and strobed short preamble. However, it can be observed that when duty cycle of sensor nodes is very low, the number of wasted strobed short preamble will increase. Hence the strobed short preamble is not a complete solution to make the protocol adaptive to the traffic loads. Key design points and performance features of the preamble sampling MAC protocols surveyed have been summarized in [Table 3-4](#).

Table 3-4: A Summary of Preamble Sampling MAC Protocols for WSNs

WSN MAC Protocol	Target Application	Contention Resolution/ Scheduling Algorithm or Key Design Points	Advantages	Disadvantages and Room for Improvement
WiseMAC [34]	<ul style="list-style-type: none"> • Low to medium data rate • Multi-hop 	<ul style="list-style-type: none"> • Spatial TDMA and np-CSMA with preamble sampling • Sampling schedule exchange among the neighbors 	<ul style="list-style-type: none"> • Decoupling sender and receiver removes synchronization overhead • Sampling schedule exchange enables just-in-time preamble and data transmission 	<ul style="list-style-type: none"> • Overhearing problem in non-target receivers • End-to-end latency over multi-hop path
CSMA-MPS [101]	<ul style="list-style-type: none"> • High bit-rate transceivers with small transmit and receive turnaround times • Low duty cycle 	<ul style="list-style-type: none"> • Alternating transmitting and receiving mechanism from STEM • Wakeup mechanism from WiseMAC 	<ul style="list-style-type: none"> • Decoupling sender and receiver removes synchronization overhead 	<ul style="list-style-type: none"> • Not applicable to medium and lower bit-rate transceivers • High probability of collision and high energy consumption in non-high bit-rate application

X-MAC [34]	<ul style="list-style-type: none"> • Event monitoring with dynamic traffic loads • Multi-hop 	<ul style="list-style-type: none"> • Address information embedding in short preamble for target receiver • Strobed preamble to enforce the immediate data transmission • Adaptive duty-cycle to dynamic traffic loads 	<ul style="list-style-type: none"> • Overhearing problem mitigated significantly • Truncating the preamble saves energy at both the transmitter and receiver and allows for lower latency • Adaptive to dynamic traffic loads 	<ul style="list-style-type: none"> • Hidden node problem still exist due to using np-CSMA protocol • Adaptability to different traffic patterns is to be evaluated further
B-MAC [35]	<ul style="list-style-type: none"> • Event monitoring with a wide range of network conditions 	<ul style="list-style-type: none"> • Clear Channel Assessment (CCA) • Adaptive Preamble sampling scheme • A set of reconfigurable parameters of MAC protocol • A well-defined flexible interface 	<ul style="list-style-type: none"> • High throughput and energy efficiency • Allow to reconfigure a set of parameters of an MAC protocol based on the current traffic load 	<ul style="list-style-type: none"> • Long preamble may introduce additional latency • No protection mechanism against the hidden terminal problem
SCP-MAC [32]	<ul style="list-style-type: none"> • Variable traffic loads with burst events • Multi-hop 	<ul style="list-style-type: none"> • Synchronized Channel Polling • Adaptive Channel Polling (ACP) 	<ul style="list-style-type: none"> • Adaptive to variable traffic loads • ACP reduces multi-hop latency • Significantly improve throughput 	<ul style="list-style-type: none"> • Improvement in energy conservation is very limited

Legend:

CB: Contention-Based

SB: Scheduling-Based

LTL: Low Traffic Load (message interval > 10s)

HTL: High Traffic Load (message interval < 10s)

3.6 Challenges to MAC Technology for WSNs

Different communication infrastructures, networking protocols and algorithms have been proposed for traditional wireless networks as well as for WSNs. The differences come not only from their physical characteristics, but also from application-specific requirements. As a result, the requirements for MAC protocol in WSNs are noticeably different from those for traditional wireless networks. Of which, the following requirements are the most relevant to WSNs, thus envisioned as the most challenging to MAC technology for WSNs.

3.6.1 Adaptability

WSNs are often application-specific. While there are typical applications over WSNs, different applications still exhibit their own peculiarities on the usage pattern of networks. As a result, the MAC layer protocols for WSNs need to be adaptable enough to accommodate a variety of traffic patterns - rate-based or bursty, reliable or best effort,

and so on [37]. This needs the MAC layer protocols to coordinate with the higher layers of the protocols stack (Network and Application layers) to adapt radio usage to application communication pattern and achieve higher energy savings [38].

3.6.2 Resource-Efficiency through Cross-Layer Design

Majority of the existing protocols and schemes are designed based on classical layered approach and are individually developed for different layers in the protocol stack of the networks. While these protocols may achieve very high performance in terms of the metrics related to each of these individual layers, they have not been globally optimized to maximize the overall network performance while minimizing the energy expenditure [39]. This is particularly true when the WSNs characterize strong dynamics from various aspects. The challenge is to find a compromise that can maximize the overall energy efficiency. An analysis that synergically considers all the mutual interactions among the different layers is deemed to be extremely complicated. This complexity is exacerbated due to the difference in order of magnitude of the time constants at which each layer operates [40] [41].

3.6.3 Decentralized

Most algorithms working in WSNs need to be decentralized. This is due to both the large scale of the network and the intrinsic unreliability of any single node in the network. Consequently, the MAC layer protocols need to be designed and developed with decentralized algorithms [37].

3.6.4 Real-time and Quality-of-Service-aware

While the issue of energy-efficiency has dominated most of the research in WSNs, the increasing interests in real-time applications along with the demanding on streaming images and video traffic have posed additional challenges. Throughput, latency and delay jitter become major performance metrics to measure quality of service (QoS) of WSNs. For object tracking and monitoring applications over WSNs, once sensor nodes are deployed, they are required to establish communication infrastructure in real-time, collaboratively collect data pertinent to the objects in real-time, and send them back to the base station within time constraints. Thus real-time network discovery algorithms, data processing and information delivery capabilities are indispensable requirements in such applications. Collecting sensed imaging and audio/video data requires both energy-

efficiency and QoS-aware network protocols including MAC layer protocols to support real-time and QoS-aware features [42]. The paper of [43] provides a survey on the existing literature on the issue of ad-hoc network management from the QoS service perspective, and identifies that the policy-based management scheme is a promising approach for QoS provisioning and management. It has also described the crucial components for effective functioning of the policy-based QoS service framework.

3.6.5 Balance among Multiple Metrics

The MAC protocols designed for WSNs should accomplish balancing among the performance metrics in order to fulfil the common goal of the WSNs. Although an unbalanced design may produce a very smart scheme to enhance one of metrics, without balanced considerations on other metrics, the smart scheme alone cannot improve the overall performance of the networks measured by various metrics. For example, an MAC protocol may be designed well enough to achieve significant energy saving by turning off the radio transceivers at a node too frequently. It may hinder the reliability of packet delivery and cause data loss, hence requiring retransmissions which results in greater power consumption.

3.7 Summary

MAC layer protocols for WSNs have attracted a lot of attention in recent years. Their design poses challenges that are absent in traditional wireless networks. In this chapter, we reviewed recent progress in the development of MAC protocols for WSNs. The protocols were grouped into four main categories: *Contention-based*, *Contention-free (Schedule-based)*, *Hybrid* and *Preamble Sampling protocols*. The protocols were classified based on their key features and the techniques they employ. For each protocol, the background, main features, operation procedures and key design considerations were described. We discussed the advantages and disadvantages of the protocols, as well as challenging issues in the design of MAC protocols for WSNs.

4 An Energy-Efficient and Low-Latency MAC Protocol with Adaptive-Scheduling for Multi-hop WSNs

In this chapter, we describe the design of a novel energy-efficient and low-latency MAC protocol for multi-hop wireless sensor networks – adaptive scheduling MAC (AS-MAC), and demonstrate its performance as well as comparisons with peers’ research works in this area. This novel MAC protocol, as a tentative solution to the identified problems, is targeting at industrial monitoring applications which require long lifetime and minimum delay in data relay.

4.1 Direct Motivation of Proposed Scheme

Several studies [77] [74] [12] show that idle listening is one of the largest sources of energy consumption at wireless nodes. Many solutions have been proposed to reduce the idle listening. Some solutions work based on synchronized *duty cycle* [12], [10], [14], [23], [24]. Some solutions employ *preamble sampling* or *low power listening* (LPL) technique [36], [34], [35], [32], [101]. By the synchronized *duty cycle* approach, sensor nodes periodically alternate their operations between an active mode and a sleep mode in a complete duty cycle. In the active mode, a sensor node is able to transmit or receive data, while in the sleep mode the sensor node completely turns off its radio to save energy. In order to transmit data from one node to another, the radios of both nodes must be switched on during the transmission which needs synchronizations amongst operational cycles of different nodes. Hence, the synchronized *duty cycle* approach strives toward low duty cycle, i.e. being in short active mode and long sleep mode, to mitigate idle listening and save energy. By the *preamble sampling* approach, when a sender has data to send, it first transmits a preamble that is at least as long as the Sleep period of the receiver. The receiver wakes up at regular intervals to detect the preamble, and once detecting the preamble, the receiver will stay awake to receive the data. This allows low power communication without the need for explicit synchronization between the nodes. The receiver only wakes for a short time to sample the medium, thereby limiting idle listening.

One example of the protocols with synchronized duty cycles is S-MAC [12]. While S-MAC obtains significant energy savings compared to a random access MAC protocol without sleeping, it experiences significant latency due to inefficient and low duty cycle. Forwarding a packet over multiple hops often requires multiple operational cycles since nodes have to wait for next cycle to forward data. This latency caused by periodic sleep is called *sleep latency*. Different approaches [10] [14] [23] [24] have been proposed to reduce sleep latency. Some of them [10] [14] [23] are mainly effective under light traffic load while DW-MAC proposed in [24] can perform well under high traffic load. However, its sleep latency has not been fully minimized. Under low traffic load, its idle listening still exists due to its fixed data period.

4.2 Executive Summary of Proposed Scheme

In this chapter, we present a new MAC protocol, called Adaptive Scheduling MAC (AS-MAC) protocol that evolves from DW-MAC. Similar to DW-MAC, the AS-MAC protocol allows sensor nodes to wake up on demand during the Sleep period of a duty cycle to transmit or receive packets with a mapping function to prevent data collision. AS-MAC differs from DW-MAC as AS-MAC has replaced the Data period with Adaptive Scheduling (AS) period within which it adaptively changes the resilient active time period, which is the time duration when a node stays awake, so as to allow nodes to schedule more data transmission within one operational cycle when the traffic load is high and go to sleep early when the traffic load is low. The contributions of this work can be summarized as follows:

- An Adaptive Scheduling period has been introduced to make the active time period adaptive to traffic load, which reduces idle listening thus saving energy.
- Two new parameters have been introduced in the scheduling frame (SCH) to prevent duplicate data transmission and delay to the scheduled data transmission.
- An enhanced mapping function has been proposed to prevent the collisions between data and acknowledgement (ACK) messages in most of scenarios.
- A progressive timeout mechanism has been proposed to prevent the node far away from the sink from early-sleep problem.

4.3 Direct Related Work

T-MAC [14] is one typical example of the efforts to reduce sleep latency, which uses timeout to dynamically determine the length of active period under variable load. That is, the active period ends when no activation event has occurred for a period of time, TA . Consequently, TA determines the maximum amount of idle listening in the active period. While T-MAC outperforms S-MAC, the performance improvement in term of energy saving is minor, only by 5%, which is achieved with cost of a decreased maximum throughput due to its early sleep problem.

S-MAC with Adaptive Listening (S-MAC-AL) [10] has introduced a new Adaptive Listening scheme into S-MAC to reduce sleep latency. With adaptive listening, a node that overhears its neighbor's transmissions wakes up for a short period of time at the end of the transmission. Hence, if the node is the next-hop node, its neighbor can immediately pass data to it instead of waiting for its scheduled listen period, by which sleep latency will be reduced. S-MAC-AL can reduce end-to-end delay by over 50% compared to S-MAC. And more delay reduction can be achieved with the increase of sensing range. Unfortunately, this achievement comes with much more energy consumption and lower packet delivery ratio with increasing of sensing range [24] because many neighboring nodes may overhear the RTS or CTS and wakeup, whereas only one of them is the next-hop node. By S-MAC-AL, a packet can be delivered up to two hops, e.g. from A to B to C, in each operational cycle. But it, generally, cannot go beyond two hops within the cycle since the next hop after C is unlikely to be awake due to the limited sensing range.

Demand Wakeup MAC (DW-MAC) [24] has introduced a new low-overhead scheduling algorithm to address the high end-to-end latency problem in multi-hop forwarding and the high energy-consumption problem suffered by S-MAC, T-MAC and S-MAC-AL. DW-MAC uses one-to-one mapping function to prevent collisions in data transmission. By shortening Data period, DW-MAC can achieve shorter end-to-end latency. However, in the DW-MAC, the durations of Data period and Sleep period are fixed. When traffic load increases, Data period might not be long enough for the node that has a burst traffic load to exchange required number of SCH frames. It makes the node have to wait for another cycle, which will increase the end-to-end delay significantly because each cycle is quite long. On the other hand, the fixed Data period

can lead to idle listening when the traffic load is light. Since the DW-MAC scheme cannot adapt to the dynamic change of traffic load, idle listening and latency cannot be reduced much under variable traffic load. With aim to improve the drawbacks in the DW-MAC scheme, our proposed AS-MAC scheme introduces a new idea to have an adaptive scheduling and some assistant mechanisms to adapt to different amount of traffic load so that it can achieve much less energy consumption and much lower packet transmission delay.

4.4 System Background

The network architecture under the study of this dissertation is a multi-hop sensor network. Assume there are N nodes distributed randomly within a cover area of $A \text{ m}^2$. The nodes can communicate with each other or send data to the sink using their Omni-directional antennas. Each sensor is able to do multicasting, which means a sensor can send data to several nodes concurrently but not necessarily broadcast. The desired receivers are able to receive the desired packets while other receivers will filter them out at the physical layer [78].

Each node has a fixed transmission radius r and a fixed sensing range R . A sensor receives a message if and only if it is within the transmission radius of the transmission sensor. Each sensor can be in two main modes: sleep mode and active mode. During active mode, sensors can transmit, receive data or stay idle. All sensors have the same power consumption parameters.

The size of each data packet is fixed to M bits. Packets with larger size will be divided into small size packets and transmitted independently. At each sensor node, there is a queue of packets for the transmission. Forwarding packets have higher priority than packets generated by the node itself. When there is a packet available to be sent, a node will first send a control packet to all of its neighbours. Only the node with the same network address as the address in the control frame will reply. Nodes will then schedule the time for packet transmission and send the packet in a scheduled time slot. The packet reaches the receiver after one round trip propagation delay. The receiver at the destination should have been tuned to the same channel to receive the packet. The size of ACK packet is of L bits. It will only be sent once the packet is successfully received. If the sending node does not receive acknowledgement after a certain time, it

will retransmit the packet. It will keep retransmitting the packet until the transmission is acknowledged successfully or the retry limit is exceeded.

4.5 The Proposed AS-MAC Scheme

Since AS-MAC evolves from DW-MAC, it is worth highlighting DW-MAC scheme's operation procedure and key design points before devoting to the AS-MAC scheme.

4.5.1 DW-MAC Scheme Operation Procedure and Highlights

DW-MAC is a synchronized duty cycle protocol with each complete cycle divided into three periods:

- Synchronization period with duration T_{SYNC} : During this period, all sensor nodes are synchronized by any synchronization protocol such as the one in [79].
- Data period with duration T_{DATA} : Sensor nodes with pending data contend for channel access using a scheduling frame (SCH) which serves as the function as either or both of RTS /CTS in the CSMA/CA in IEEE802.11.
- Sleep period with duration T_{SLEEP} : Most of sensor nodes sleep in this period, and only those having scheduled data transmission wake up to send or receive data and ACK message.

In the operation of the DW-MAC, two design features play decisive roles in the operation of the protocol as discussed as follows.

4.5.1.1 Proportional Mapping Function

DW-MAC differs from prior work in the way it schedules nodes to wake up during the Sleep period in a cycle. It employs a one-to-one proportional mapping function to determine wake-up and data transmission time of a pair of nodes in the Sleep period. In a Data period, the time interval during which the transmission of an access control frame automatically reserves a proportional time interval in the following Sleep period for the transmission and reception of a data packet. Such a proportional mapping function is illustrated in **Figure 4-1**.

Figure 4-1: One-to-one mapping function in DW-MAC

4.5.1.2 Scheduling Frame (SCH)

DW-MAC replaces RTS/CTS used in CSMA/CA with a single scheduling frame SCH which serves as either or both of RTS and CTS. In this way, DW-MAC minimizes scheduling overhead.

4.5.1.3 Optimized Multi-hop Forwarding

When a node has a packet to send, it will send a SCH frame in the Data period to schedule the data transmission in the Sleep period. A node can broadcast one single SCH frame to serve as CTS to the node in the previous hop and as RTS for the node in the next hop. By a SCH frame, the number of access control messages needed for scheduling data transmission can be reduced by half. **Figure 4-2** Shows that the number of SCHs needed to schedule x -hop data transmission is only $x+1$ while it needs $2x$ frames in CSMA/CA. The scheduling overheads and the possibility of collisions can be much reduced.

Figure 4-2: Scheduling frame exchange (a) CSMA/CA, (b) DW-MAC

4.5.2 Our Proposed AS-MAC Scheme

Our proposed AS-MAC aims to enhance the DW-MAC scheme. The AS-MAC is similar to DW-MAC in several aspects of the design. 1) It follows Sync-Data-Sleep 3-phase operational cycle for its nodes' operation and 2) It adopts an enhanced proportional mapping function for scheduling data transmission in the Sleep period. However, the proposed AS-MAC differentiates itself from DW-MAC in the following aspects which make AS-MAC superior to DW-MAC in terms of energy-efficiency, latency and packet delivery ratio.

4.5.2.1 Adaptive Scheduling Period

The major disadvantage of DW-MAC is that the durations of the Data period and Sleep period are fixed [24], which results in that the system cannot dynamically adapt to different traffic loads.

To solve this problem, AS-MAC designs a flexible Resilient Active Time (RAT) (defined as the time duration for a node to stay awake) within the fixed Adaptive Scheduling (AS) period, so that nodes can change the length of the RAT in each operational cycle and be adaptive to the variable traffic load. To make it feasible, a timer is employed to decide the moment the nodes should sleep. The timer expires when there is no transmission of SCH frames or transmission/collision in its neighbourhood for a period of time, T_0 . **Figure 4-3** illustrates adaptive change of the RAT in the AS-period for three scenarios respectively. The numbers of events used here are just for indicative purpose.

- In 1st operation cycle (average traffic load case): after AS-period starts, two events occur. After the 2nd event, there is no new event within timeout T_0 period. Thus node enters Sleep period after two events plus T_0 time, which is earlier than anticipated starting time of Sleep period. For this case, $T_0 < \text{RAT} < \text{preset AS-period}$.
- In 2nd operation cycle (high traffic load case): node keeps sensing 5 events in succession, and the interval of any two events in succession is shorter than the timeout T_0 period. Thus node keeps Active mode till its full AS-period. For this case, $\text{RAT} = \text{preset AS-period}$; i.e. RAT reaches its maximum duration - the AS-period.

- In 3rd operation cycle (low traffic load case): after AS-period starts, there is no single event. Thus after timeout T_0 period, node immediately enters Sleep mode. For this case, $RAT = T_0$.

Thus under high traffic load, the RAT will be prolonged to enable nodes to exchange more SCH frames, consequently transmitting more data in the Sleep period, which is the case in the 2nd operation cycle, where the RAT reaches its maximum duration - the AS-period. Under low traffic load, the RAT will be shortened to enable nodes to sleep early, even before the Sleep period starts, which is the case in the 1st and 3rd operation cycle.

Figure 4-3: Adaptive change of resilient active time

T_0 indicates maximum continuing amount of idle listening in the AS-period. The adaptive RAT can help to effectively reduce energy consumption due to less idle listening when traffic is low. In the last operation cycle in [Figure 4-3](#), by DW-MAC, a node must stay awake for the time $T_{SYNC} + T_{DATA}$ while by AS-MAC, if the node has not sensed event happened in T_0 , it will go to sleep, staying awake only for the period of $T_{SYNC} + T_0$. Since $T_0 < T_{DATA}$, the node can save much more energy. To avoid the node sleeping early when there is still transmission ongoing, we can set a high value of T_0 .

Although the AS-period and Sleep period are fixed in AS-MAC, the nodes, with help of a timer, can resiliently adjust the RAT according to the traffic load. Such self-adjustability dynamically changes the ratio of the length of the actual active time period over that of the Sleep period to adapt to the variable traffic load as much as possible to achieve lower energy consumption while maintaining high packet delivery ratio.

4.5.2.2 SCH Frame with Tracked Retry Numbers

Similar to DW-MAC, AS-MAC employs a single scheduling frame SCH to serve as either or both of RTS and CTS. The SCH includes all the fields of RTS/CTS such as

sender and receiver addresses and duration of transmission. In addition, a SCH also includes some cross-layer information. For broadcast message, a SCH frame includes the network layer address of its source, for unicast message, it includes the network layer address of its final destination. This cross-layer information enables an intermediate node to set up schedules to the next hop node even before receiving the actual data packet.

But AS-MAC has a different design from DW-MAC. AS-MAC includes two parameters in the header of the SCH. Each parameter has 3 bits to indicate not only the role of the SCH but also the number of times the SCH has been sent as RTS or/and CTS. The two parameters are *RTS_RetryNo* and *CTS_RetryNo*, respectively. In addition, SCH frame includes the unique sequence number for each packet and its copies to avoid duplicate data reception. The SCH frame size is set as 15 bytes, which is 1 byte (8 bits) more than that in DW-MAC even though SCH frame in AS-MAC is only 6 bits more than that in DW-MAC. **Table 4-1** shows the SCH frame format:

Table 4-1: SCH frame format

Frame Control	Duration	Sender Address	Receiver Address	Sequence Number	Destination/Source Addr	CTS_Retry No	RTS_Retry No	CRC
2bytes	2bytes	2bytes	2bytes	2bytes	2bytes	3bits	3bits	2bytes

The purpose to introduce the two parameters, *RTS_RetryNo* and *CTS_RetryNo*, is to avoid duplicate data transmission, thus reducing latency and energy consumption. AS-MAC uses the unique sequence number, *RTS_RetryNo* and *CTS_RetryNo* to differentiate distinct successful SCH transmissions. When a node sends an SCH (RTS), the *RTS_RetryNo* will be set to the number of times this SCH has been tried to send. When the SCH (RTS) successfully reaches the receiver, the receiver will record the *RTS_RetryNo* and send a SCH (CTS) to the source with *CTS_RetryNo* set as same as the *RTS_RetryNo* received. Assume A needs to send a data packet to D through B and C as shown in **Figure 4-4** (a). A sends an SCH (RTS) successfully to B and sets a timer to wait for SCH (CTS) from B. Upon receiving the SCH (RTS) from A, B sends an SCH (CTS) to A which at the same time serves as SCH (RTS) to C. However, A might lost the SCH (CTS) from B before the SCH (CTS) timer timeout and interprets that its SCH (RTS) transmission failed. Before the retry limit has been reached, A sends another SCH (RTS) to B. In the mean time, after receiving the SCH (RTS) from B, C

will continue the forwarding to D. If the 2nd SCH (RTS) succeeds and A also receives SCH (CTS) from B, the DW-MAC uses the 2nd time SCH (RTS) to schedule the data transmission. Differentially in AS-MAC, since B's CTS_RetryNo returning to A has recorded the 1st successful SCH (RTS), A will use the sending time of this 1st successful SCH (RTS) to schedule the data transmission, and B does not change the schedule to send data to C, C needs not to reschedule the data transmission to D as shown in **Figure 4-4 (b)**.

Figure 4-4: Forwarding SCH (a) DW-MAC, (b) AS-MAC

The advantages of SCH frame with tracked retry numbers are in three-folds: (1) end-to-end delivery latency will be reduced as the 1st successful RTS is used for mapping the data transmission; (2) energy consumption can be reduced due to the avoidance of unnecessary SCH frames transmission; (3) Improvement of the delivery latency and energy conservation will not introduce perceivable overhead.

4.5.2.3 Enhanced Proportional Mapping Function

To represent DW-MAC's proportional mapping function in more generic notations, **Figure 4-1** is represented as the following **Figure 4-5** with revised notation. The notation mapping from DW-MAC to AS-MAC and rationality are tabulated in Appendix A. The remaining sections of the chapter will use the new notations pertaining to the proportional mapping function.

Figure 4-5: DW-MAC one-to-one proportional mapping function with new notation

DW-MAC's proportional mapping function shown in equation (4-1) ensures the collision-free transmission between data and data packets and collision-free transmission between data and ACK packets in most scenarios. AS-MAC enhances DW-MAC's proportional mapping function by taking the ACK packet transmission into the consideration of proportional mapping as shown in equation (4-2).

$$\frac{T_i}{T_i'} = \frac{T_{SCH}}{T_{Data}} = \frac{T_{AS}}{T_{SLEEP}} \quad (4-1)$$

$$\frac{T_{AS}}{T_{SLEEP}} = \frac{T_i}{T_i'} \leq \frac{T_{SCH}}{T_{Data} + T_{ACK}} \quad (4-2)$$

We would say that this proportional mapping function further mitigates the probability of collision between data and ACK packet in multi-hop scenario and prevent data-ACK collision in two-hop scenario. To prove it, let's consider the worst case scenario when n SCHs are successfully transmitted from A to B in a sequence. The start time of i^{th} SCH is at the moment of T_i right after the AS-period begins as in Figure 4-6.

Figure 4-6: Condition for Collision-Free Data Transmission

In order for two successive SCHs to be transmitted successfully, they cannot overlap with each other, which needs: $T_{i+1} \geq T_i + T_{SCH}$. In the worst case scenario,

$$T_i = (i-1)T_{SCH} \quad (4-3)$$

The i^{th} SCH corresponds to one successful Data transmission from A with one successful ACK transmission from B, which start at T_i' and require $T_{Data} + T_{ACK}$ to complete in the Sleep period, if the propagation delay is ignored. To prevent collisions, the successive transmission of a packet and an ACK cannot overlap as $T_{i+1}' \geq T_i' + T_{Data} + T_{ACK}$. Hence, we have:

$$T_i' \geq (i-1)(T_{Data} + T_{ACK}) \quad (4-4)$$

From equations (4-3) and (4-4):

$$\frac{T_i}{T_i'} \leq \frac{(i-1)T_{SCH}}{(i-1)(T_{Data} + T_{ACK})} = \frac{T_{SCH}}{T_{Data} + T_{ACK}} \quad (4-5)$$

The enhanced proportional mapping represented in (2) indicates that the ratio of $\frac{T_{AS}}{T_{SLEEP}}$ and $\frac{T_i}{T_i'}$ should be lesser than or equal to a constant value which is determined by three constant values at the right side of the equation (4-2). It also indicates that T_{Data} in equation (4-1) ($=T_4$ in Figure 4-1) becomes too small to secure a collision-free data-ACK transmission in DW-MAC. AS-MAC, however, gets away this shortcoming by adding in T_{ACK} in the ratio calculation. It is equally saying that T_i' should be long enough in order to secure a collision-free data-ACK transmission. The resultant ratio from equation (4-2) will be used as guideline to set parameters T_{AS} and T_{SLEEP} .

With this mapping function plus MAC layer virtual sensing (RTS/CTS mechanism), AS-MAC can schedule data transmission with data-ACK collision-free in the Sleep period in two-hop scenario. For the multi-hop scenario, the data-ACK collisions will be mitigated with T_{ACK} being taken into the consideration in proportional mapping function.

Under the broadcast traffic, since each data transmission does not require an ACK, the function can be simplified as that of the DW-MAC.

From this mapping function, the maximum number of SCHs that can be issued in an AS-period can be determined by $N_{SCH} = T_{SLEEP}/(T_{Data}+T_{ACK})$, where N_{SCH} is the maximum number of SCHs in an AS-period, T_{SLEEP} is the duration of the Sleep period, which is a fixed parameter.

4.5.2.4 Data Transmission Scheduling without Compulsory SCH (CTS)

In DW-MAC, a contending node will not send a data packet during SLEEP period unless it receives SCH (CTS) to confirm the scheduling is success before AS-period is over. However, there is a high chance that node does not receive SCH (CTS) but there was a successful transmission of SCH (RTS). The reason can be that the SCH (CTS) is collided or has not been sent yet due to busy medium. If that is the case, the destination node will still wake up at the scheduled time for receiving data but the source node will not send anything since it does not receive the SCH (CTS). As the result, the transmission is wasted and the receiving node will consume more energy due to idle listening. Moreover, the data transmission will have to be scheduled again in next duty cycle, which increases unnecessary traffic and delivery latency.

To make sure no successful scheduling transmission is wasted, for AS-MAC, if no SCH (CTS) is received, the node still sends the data. Since the node cannot know which transmission of SCH (RTS) is the first successful one without receiving SCH (CTS), it will keep sending the data at the time slots scheduled by *all* SCH (RTS) transmissions it has made *until* it receive the ACK message from the destination node indicating it has received data successfully.

An example is shown in [Figure 4-7](#) where node A has sent 3 SCH (RTS)s to B to schedule a data transmission. The second transmission is successful; however the SCH (CTS) is collided. For DW-MAC, A will have to wait until next cycle to reschedule all over again. For AS-MAC, A will still schedule to send data at T'_{1A} , T'_{2A} . Since A receives an acknowledgement message after second transmission, it cancels the transmission at T'_{3A} .

The scheduling of data transmission without confirmation of success transmission of SCH (RTS) in AS-MAC prevents a node from postponing data transmission until the next cycle due to failed acknowledgement of successful scheduling process, thus, provides *shorter delay for data transmission*.

Figure 4-7: Scheduling data transmission when no SCH (CTS) is received

Even though the sending node may consume more energy due to sending duplicate data, since the receiving node only wakes up at scheduled time for *successful* data transmission, it will not wake up to receive duplicate data. Say in another way, receiving node will fully utilize the energy it consume during data period, which offsets the slightly increase of energy consumption for sending node.

Proposition: *Whether the average energy wasted due to sending duplicate data by source node in AS-MAC is greater or less than the average energy wasted due to destination node waking up waiting for data while it is not coming in DW-MAC depends on the probability that a SCH (RTS) transmission is successful.*

Proof:

We use the number of wasted wake up time (node wakes up without any data transmission is successfully committed) to evaluate the ability of saving energy of two algorithms. We use the scenario as mentioned in the above example to prove our proposition.

We denote

- Maximum number of SCH (RTS) retransmission allowed in one duty cycle as n , this value depends on the delay between two retransmissions, which in turn depends on the contention window and transmission time of each SCH frame.
- The possibility that one transmission is successful as a ($a < 1$). This value depends mostly on the traffic load and environment condition (signal's fading problem)
- $P(x)$ is the possibility that there is x wasted wake up time

For DW-MAC:

Since A will not send any data unless it receives SCH (CTS), A will postpone until next period to schedule data. It will not use the Sleep period to send data, thus, no transmission is wasted for A. As for B, it will schedule to wake up in Sleep period to receive data when it successfully receives SCH (RTS) from A. Since B does not know that A cancels the data transmission, it will wake up during the Sleep period and result in one wasted wake up time. If there is no SCH (RTS) successfully reach B, it will not wake up in Sleep period, thus, result in no wasted wake up time.

The possibility that in n retransmission, no SCH (RTS) transmission reach B is:

$$P(0) = (1 - a)^n$$

The possibility that in n retransmission, there is at least one transmission reaches B is:

$$P(1) = 1 - (1 - a)^n$$

Hence, the average number of wasted wake up time is:

$$\bar{x}_{DW} = \frac{\sum P(x).x}{\sum P(x)} = \frac{(1-a)^n.0 + (1-(1-a)^n).1}{1} = 1 - (1 - a)^n \quad (4-6)$$

For AS-MAC:

For AS-MAC, B will never have to waste any wake up time since A will not postpone the data transmission. The number of wasted wake up time is thus solely depends on the retransmission of SCH (RTS) from A.

If the first SCH (RTS) transmission is successful, there will be no wasted wake up time since after the first data transmission is successful, A will not retransmission the data any more. The probability that the first SCH (RTS) transmission is successful is:

$$P(0) = a^n$$

If the k^{th} retransmission is the first successful SCH (RTS), A will wake up k time in Sleep period to send data and thus, wastes $(k-1)$ wake up times. The probability that the k^{th} retransmission is the first successful SCH (RTS) is:

$$P(k-1) = (1-a)^{k-1} \cdot a$$

If all the SCH (RTS) transmissions are unsuccessful, A will wake up n times without successfully transmit the data, thus, will waste all n wake up time. The probability that all n retransmission are unsuccessful is:

$$P(n) = (1-a)^n$$

Hence, the average number of wasted wake up time is:

$$\begin{aligned} \bar{x}_{AS} &= \frac{\sum P(x) \cdot x}{\sum P(x)} = \frac{a \cdot 0 + (1-a) \cdot a \cdot 1 + \dots + (1-a)^{n-1} \cdot a \cdot (n-1) + (1-a)^n \cdot n}{1} = \frac{a \sum_{i=0}^{n-1} (1-a)^i \cdot i + (1-a)^n \cdot n}{1} = \\ &= \frac{1-a}{a} [1 - (1-a)^n] \end{aligned} \quad (4-7)$$

From equation (4-6) and (4-7), we have

$$\begin{aligned} \bar{x}_{DW} > \bar{x}_{AS} &\leftrightarrow a < \frac{1}{2} \\ \bar{x}_{DW} \leq \bar{x}_{AS} &\leftrightarrow a \geq \frac{1}{2} \end{aligned}$$

In conclusion, if the possibility that a SCH (RTS) is successful is high, the proposed method by AS-MAC will save more energy than DW-MAC and vice versa. Hence, it is more useful to apply it when traffic load is low. However, since by using proposed method, node does not have to use another duty cycle to schedule data transmission, it will also eliminate the extra transmissions that the node uses in next duty cycle. As the result, we decide to use the proposed method for our simulation of AS-MAC under various traffic loads.

4.5.3 Enhancement of AS-MAC for Broadcast Traffic

With broadcast traffic, a source node sends a packet to all the other nodes in the network. The operation of AS-MAC for broadcast traffic is slightly different from that for unicast traffic as follows. SCH frames only function as a RTS message to schedule data transmission. It won't function as a CTS message for broadcast traffic. A source node broadcasts SCH frame to all of its neighbour nodes exactly once without retransmission. The source node will automatically schedule time to wake up for data transmission regardless of the transmission is successful or not. Similarly, in Sleep period, data is broadcasted without acknowledgement. Thus, it is more difficult for broadcast packet to reach all nodes in the network. A broadcast is considered successful if all nodes in the network receive the broadcast packet. End-to-end delay is calculated as the time taken by the packet to reach all the nodes in the network.

4.5.3.1 Major Problem under Broadcast Traffic

Early sleeping is the situation when one node transmits an SCH frame to a sleeping node, which results in a failed transmission. The impact of early sleeping problem on the performance of AS-MAC in unicast traffic case is minor. But in the broadcast traffic scenario, every node only broadcasts SCH (RTS) exactly once and there is no SCH (CTS) to confirm the successful transmission. If a node does not stay awake long enough to wait for the SCH frame, it will not have a chance to wake up in the next Sleep period to accept the data packet. It is obvious that the node which is farther from the sink has higher chance to miss a SCH in the AS-period and, consequently, missing the data transmission in Sleep period due to early sleeping since the SCH frame has to travel through more intermediate hops to reach it.

4.5.3.2 Proposed Solution to Early Sleeping

To solve the early sleeping problem, on one hand, we employ the Connected Dominating Set (CDS) mechanism proposed by Gandhi et al. in [83] to minimize the number of retransmissions, and use the Optimized Multi-Hop Forwarding Algorithm proposed in [24] to optimize Multi-Hop Forwarding. There are two reasons to use CDS: (1) CDS helps streamline the number of forwarding nodes for broadcast traffic, thus improving the efficiency of forwarding the broadcast traffic; (2) DW-MAC used CDS in its broadcast traffic evaluation. To have a fair comparison, it has also been used in our scheme in the broadcast traffic evaluation. On the other hand, we propose an

approach to adjust timeout duration to prevent the farther nodes from early sleeping. Hence a combined solution with three steps to enhance the proposed AS-MAC under broadcast traffic is described below in details.

1) Minimize the Number of Retransmissions

Retransmissions consume valuable resources of the network such as power and bandwidth. Hence it is important to choose the intermediate nodes carefully to avoid redundancy of retransmissions. To minimize the number of retransmissions as much as possible, it is necessary to construct the broadcast tree. A broadcast tree is a spanning tree rooted at the sink where if nodes u and v both belong to the broadcast tree and u is a parent of v , then u is responsible for transmitting the message to v without any collision at v . At the initialization step, the broadcast tree will be generated by using Breadth First Search Algorithm, which is described in **Figure 4-8**. **Figure 4-9** depicts an example of broadcast tree constructed based on the algorithm.

Figure 4-8: Breadth First Search Algorithm

Figure 4-9: A broadcast tree based on the Breath First Search Algorithm

Based on the broadcast tree, a set of Connected Dominating Set (CDS) can be constructed which consists of all nodes that are internal (non-leaf) in the broadcast tree. In the above example, CDS consists of node 1, 3, 6, 2 and 12. Once the CDS is determined, the broadcast itself can be performed based on it. The CDS nodes that have received a broadcast message will rebroadcast it exactly once. All the other nodes, which are leaves of the broadcast tree (node 5, 7, 4, 9, 10, 8, 11), will not rebroadcast the message. By this way, the least number of rebroadcasts can be achieved, and the transmitted data packets can reach all nodes in the network with reduced contention, collision and redundant retransmissions.

2) Optimized Multi-Hop Forwarding

To make data packets to reach the farthest node as soon as possible, AS-MAC adopts the Optimized Multi-Hop Forwarding Algorithm proposed in [24], where a source node selects an immediate forwarder and indicates it in the SCH frame. The forwarder rebroadcasts the SCH frame SIFS time after receiving the SCH. Any other nodes will back off before rebroadcasting SCH. The immediate forwarder will be selected as the one with the highest height or the most number of edges on the longest downward path to a leaf in the broadcast tree.

By setting intermediate forwarder, we can prioritize the transmission of the broadcast packets along some paths that can reach the farthest nodes rather than having all rebroadcasting nodes compete for the medium equally. Determining the intermediate forwarder may base on different criteria such as location, number of children nodes of its neighbours. The index of the intermediate forwarder will be indicated inside the SCH frame, so that rebroadcast nodes know whether it is an intermediate forwarder or not. In

this way, the delay involved in broadcasting from the sink to the farthest node could be shortened, which in turn will reduce the probability of early sleeping.

Figure 4-10: Optimized Multi-hop Forwarding under Broadcast

3) Adjusting Timeout Duration

Since nodes farther away from sink node have to wait longer for the message to arrive, a nature approach to prevent farther nodes from early sleeping is to adjust timeout duration T_0 for nodes at different locations. The policy taken by the AS-MAC is that farther nodes have longer timeout durations to wait for coming packets. The timeout duration for a node can be determined by the number of hops along the path from the sink to the node. The more hops a node is away from the sink, the longer timeout duration will be set for the node. The index of the intermediate forwarder indicated inside the SCH frame can be used to determine its distance (in number of hops) from the sink. Let's assume that one hop transmission is the transmission from the sink to its direct neighbouring nodes, which are within the radio range of the sink. If T_0 is the timeout of the 1st-hop node, T_n denotes the timeout of the n^{th} -hop node, and ΔT denotes the average prolonged time for each hop which can be determined by the transmission time of a SCH, then $T_n = T_0 + (n-1) \Delta T$.

The timeout duration has impacts on energy conservation and end-to-end latency. A shorter timeout is benefit to energy conservation but affects end-to-end latency, while a longer timeout will improve end-to-end latency but consume more energy due to the longer awake period for low traffic loads. Hence, an appropriate timeout value should be able to achieve a better trade-off between energy conservation and end-to-end latency.

4.5.4 Implementation Complexity and Cost

It can be seen from the following analysis that the added implementation complexity and cost in AS-MAC is ignorable compared to the DW-MAC.

Addition of the timeout mechanism, RTS_RetryNo and CTS_RetryNo, the enhanced proportional mapping function, variable timeout value for different nodes in the AS-MAC design do not change the complexity of the scheme design of DW-MAC; and the calculations of these values are linear and can be easily implemented through simple programming. Thus we may say that the added implementation complexity is ignorable compared to the DW-MAC. The computation cost incurred for timeout checking and variable timeout value, as well as the extra energy for sending/receiving the six more bits for RTS_RetryNo and CTS_RetryNo in the SCH control frame (see [Table 4-1](#) for details) are very minor. Hence we may ignore the added implementation complexity and extra cost incurred while compared with DW-MAC.

4.6 Performance Evaluation

In this section, we present the numerical results from the simulation experiments to evaluate the performance of the proposed AS-MAC under both unicast and broadcast traffic. For the result comparisons, since DW-MAC used NS-2 as simulation software, while we have employed OMNET++ for the simulation of the AS-MAC, the presented DW-MAC data are precisely derived from the original DW-MAC paper.

4.6.1 Performance Evaluation under Unicast Traffic

4.6.1.1 Simulation Design under Unicast Traffic

Similar to the DW-MAC, we also assume that the network under the study is a 7x7 grid sensor network with 49 nodes. Each node is 200 meters away from its neighbours and the sink at the center is the only destination node of the unicast traffic. All data packets generated at different nodes will be sent to the sink node, which results in the path that each data packet traverses, ranging from 1 to 6 hops.

We compared the performance of the AS-MAC with that of the DW-MAC in [\[24\]](#) which has been proven outperforming other protocols such as S-MAC [\[12\]](#), S-MAC-AL [\[10\]](#), and RMAC [\[23\]](#). In our simulation experiment design, the same network parameters as in DW-MAC have been set for AS-MAC as shown in [Table 4-2](#), except

for two differences: (1) the size of SCH is 15 bytes, which one byte longer than that in DW-MAC, (2) the duty cycle is not fixed since the RAT is not fixed with parameter duty cycle replaced by timer T_0 . To avoid early sleeping problem, T_0 is set equal to 80ms, which is greater than the time required for an SCH frame to traverse 6 hops. If $T_{SCH,Hn}$ denotes the time required for an SCH frame to traverse n hops, then

$T_{SCH,Hn} = (T_{SCH} + T_{SIFS}) * n = (SizeofSCH / Bandwidth + T_{SIFS}) * n = (15bytes / (20kbps/8) + 5ms) * n = (6ms + 5ms) * n = 11n (ms)$. Thus $T_{SCH,H6} = 66ms$. T_0 is much smaller than the duration of DATA period in DW-MAC. When traffic load is low, nodes will go to sleep earlier in AS-MAC compared to DW-MAC.

Table 4-2: Networking Parameters

Bandwidth	20Kbps	Ch. Encoding Ratio	2
Tx Power	31.2mW	Tx Range	250m
Rx Power	22.2mW	Carrier Sensing Range	550m
Idle Power	22.2mW	Contention Window (CW)	64ms
Sleep Power	3μW	Size of ACK	10B
SIFS	5ms	Size of SCH	15B
DIFS	10ms	Size of Data	100B
Retry Limit	5	State Transition Power	31.2mW
T₀	80ms	State Transition Time	2.47ms

The propagation delay is neglected. In order to compare fairly, we set the length of the operational cycle similar to that in the DW-MAC with the duration of Sleep period and AS-period adjustable. The duration of Sleep period must satisfy condition (2) and at the same time the AS-period should be as long as possible to adapt to high traffic load. Since transmission time is proportional to the size of data packet in transmission, we have:

$$\frac{T_{AS}}{T_{SLEEP}} \leq \frac{T_{SCH}}{T_{Data} + T_{ACK}} = \frac{SizeofSCH}{SizeofData + SizeofACK} = \frac{15}{100 + 10} \approx 0.136 \quad (4-8)$$

Durations of synchronization, AS-period for AS-MAC or Data Period for DW-MAC and Sleep period are shown in [Table 4-3](#).

Table 4-3: Operational Cycle Configuration

	T _{SYNC} (ms)	T _{AS} or T _{Data} (ms)	T _{SLEEP} (ms)	T _{Cycle} (ms)
DW-MAC	55.2	168.0	4241.8	4465.0
AS-MAC	55.2	400	4009.8	4465.0

The size of data packet is fixed as 100 bytes in AS-MAC. The shortest path routing function has been used for routing decision as it is available in OMNET++ library [80], which is the simulation software employed in this work. There are three main sources of power consumption of a WSN: sensing, computing and communication. Among the three sources, we focus to evaluate the issue of the power consumption in the communication including the energy consumed by idle listening in order to simplify the scenario and have a fair comparison with DW-MAC. It is assumed that all nodes are synchronized with each other.

Event Interval (EI) has been used to express interval of the events arrival, whose values can estimate the traffic loads. The larger the EI value is, the lower the traffic load is. And the smaller the EI value is, the higher the traffic loads is. To have a fair comparison with DW-MAC, we let EI start with 200s, i.e. there is an observed event happening every 200 seconds. And this event will happen at a random location with coordinates being generated following uniform random distribution.

To investigate the performance of AS-MAC in a wider range of event interval, we have tested the AS-MAC scheme with more frequent EI values of 150s, 100s, 50s, 30s, and 20s. For each EI value, any node with its sensing range covering the location of event happening will collect the information, produce a data packet and send it back to the sink. The sensing range for each node is set from 100m to 500m with interval of 50m as DW-MAC did. By increasing the sensing range, the traffic load increases since more sensors can sense the observed event and generate more data packets. The simulation has been taken for 10 random runs, each of which lasted for certain time or 500 events triggered. **Table 4-4** shows the average number of packets generated for each event under different sensing ranges for scenario EI=200s.

Table 4-4: Average Number of Packets Generated per Event (EI=200s)

Range (m)	100	150	200	250	300	350	400	450	500
P a c k e t s	0.8	1.7	3.1	4.6	6.4	8.4	10.6	12.9	15.2

4.6.1.2 Results and Discussions

We show the performance results of AS-MAC in two sets. One set with EI=200s is to have counterpart comparison with DW-MAC as shown in **Figure 4-11**. Another set with smaller EIs is to show the performance of AS-MAC for higher traffic loads as shown in

Figure 4-12. For EI=200s, **Figure 4-11** shows the average and maximum end-to-end packet delay of DW-MAC and AS-MAC. Packets experience longer delay by DW-MAC scheme than by AS-MAC scheme when the sensing range increases. More specifically, for sensing range 100m~500m, the average end-to-end packet delay of AS-MAC can achieve around 36%~65% lesser than that of DW-MAC, respectively, which indicates that AS-MAC outperforms DW-MAC with traffic load increasing. When the sensing range is small, a complete data transmission can be completed within one or two cycles. Since the AS-period of AS-MAC (400ms) is much longer than the Data period of DW-MAC (168ms) shown in **Table 4-3**, AS-MAC allows more scheduled transmissions per cycle if a node has enough data packets to send, which makes the number of operational cycles required for a packet to transmit from a source node to the sink is much lesser in AS-MAC than in DW-MAC.

Figure 4-11: Average and Maximum End-to-End Delay(s) for EI=200s

Figure 4-12: Average and Maximum End-to-End Delay(s) of AS-MAC for EI=150s, 100s, 50s, 30s, and 20s

For EI=150s, 100s, 50s, 30s, and 20s, **Figure 4-12** shows the average (square pillar) and maximum (round pillar) end-to-end packets delay for these higher traffic loads scenarios, respectively. For example, when EI=20s, for sensing range 100m~500m, the average end-to-end packet delay of AS-MAC can still keep on 36%~65% lesser than that of DW-MAC, respectively, though this is not a counterpart comparison and the comparison is not in favour of AS-MAC protocol. Hence, the long AS-period in AS-MAC significantly contributes to the improvement of the average end-to-end packet delay.

Figure 4-13 shows the average power consumptions of DW-MAC and AS-MAC for EI=200s. Since AS-MAC allows nodes to sleep early when there is no data to transmit, it reduces idle listening, hence resulting in the lesser energy consumption than that of DW-MAC by approximately 40%~35% for sensing range 100m~ 500m, respectively. **Figure 4-14** shows the average power consumptions of AS-MAC for EI=150s, 100s, 50s, 30s, and 20s for different sensing ranges. Although it is not a counterpart comparison between **Figure 4-13** and **Figure 4-14** which is again not in favour of AS-MAC, it can be seen that AS-MAC still outperforms DW-MAC in terms of energy consumption. **Figure 4-13** and **Figure 4-14** show that AS-MAC consume approximately 40%~34% less power than DW-MAC for sensing range 100m~500m when EI=150s; 40%~21% for sensing range 100m~500m when EI=50s; and 41%~1% for sensing range 100m~450m when EI=20s. Only when EI=20s and sensing range=500m which represents a much higher traffic load than EI=200s, the energy consumption of AS-MAC exceeds that of DW-MAC by 6%.

These figures indicate that when there is less data to transmit, AS-MAC can save more energy due to the adaptive resilient active time within the AS-period. With traffic load increasing, AS-MAC still exhibits very competitive performance in energy consumption compared with DW-MAC. We may conclude from the above discussion that innovatively combining the adaptive resilient active time with a long AS-period can reduce the energy consumption, at the same time, can significantly shorten the average end-to-end packet delay.

Figure 4-13: Average Energy Consumption under Unicast Traffic with EI=200s

Figure 4-14: Average Energy Consumption of AS-MAC under Unicast Traffic with EI=150s, 100s, 50s, 30s, and 20s

Figure 4-15 shows that both AS-MAC and DW-MAC can keep the packet delivery ratio (PDR) of 100% until the sensing range reaches 400m. After that, the packet delivery ratios of both schemes start to drop. But the packet delivery ratio of AS-MAC drops much more slowly than that of DW-MAC. The difference is perceivable because AS-MAC uses the unique sequence number, RTS_RetryNo and CTS_RetryNo to differentiate distinct successful SCH transmissions, hence reducing wasted transmissions. Thus, the number of times the nodes give up transmission due to exceeding retry limit is much lesser in AS-MAC than that in DW-MAC. **Figure 4-16** exhibits the packet delivery ratio for the higher traffic loads scenarios with EI=150s, 100s, 50s, 30s, and 20s. Although AS-MAC's PDRs start to drop below 100% for these higher traffic loads earlier than DW-MAC, they still keep on very high PDR which are above 99.88%.

Figure 4-15: Packet Delivery Ratio under Unicast Traffic with EI=200s

Figure 4-16: Packet Delivery Ratio of AS-MAC under Unicast Traffic with EI=150s, 100s, 50s, 30s, and 20s

4.6.2 Performance Evaluation under Broadcast Traffic

4.6.2.1 Simulation Design under Broadcast Traffic

For the broadcast case, we evaluate the performance of AS-MAC in grid networks with size varying from 3x3 with 9 nodes to 11x11 with 121 nodes. In the networks, each node is 200 meters away from its neighbors and the sink node at the centre is the only source node of the broadcasting traffic. The sink node generates a broadcast packet once every 100 seconds so that transmission for one packet complete before the next packet is generated. Similarly as under unicast traffic, traffic load can be adjusted by varying sensors' sensing range.

The value of ΔT mentioned in previous section is important as it will affect the performance of AS-MAC under broadcast traffic. If ΔT is small, the energy consumption will be smaller but in trade-off network reliability. In the simulation, two different values of ΔT have been used, which are 0.02s and 0.015s. T_0 of the sink node is 0.08s. In the simulation, only the AS-MAC with Connected Dominating Set (CDS) based flooding is evaluated, indicated by CDS. However, the DW-MAC with simple flooding, indicated by ALL, the DW-MAC with CDS, and the DW-MAC with CDS with Multi-hop forwarding algorithm, indicated by CDS-MH, will be evaluated.

4.6.2.2 Results and Discussions

In [Figure 4-17](#), the end-to-end delay is evaluated as the time it takes for the last node to receive a given broadcast packet. It is clear that AS-MAC provides much lower end-to-end latency compared to those in three DW-MAC cases. It proves that by using adaptive scheduling period with resilient active duration and adjustable timeout duration T_0 , AS-MAC can effectively adapt to the variation of traffic load. When traffic is high, nodes can extend its active period, in this case, to 0.4s per period of 4.465s, which is much longer than the fixed Data period of 0.168s in DW-MAC. Thus it can allow more contending nodes to finish their transmission in one cycle. It is observable from [Figure 4-17](#) that end-to-end delay changes from within one cycle up to slightly more than one cycle time when the size of the grid network changes from 3x3 to 11x11 by the AS-MAC, while by the DW-MAC the delay is at least more than one cycle for 3x3 network and more than 5 cycles for 11x11 network. In addition, the end-to-end delay for scenarios AS-MAC CDS MH-0.02 and AS-MAC CDS MH-0.015 with network size ≤ 7 has been achieved under condition of almost 100% packet delivery ratio. The end-to-end delay for scenario AS-MAC CDS MH-0.015 with network size > 7 has been achieved under condition of packet delivery ratio below 100% as shown in [Figure 4-15](#).

Figure 4-17: Average End-to-End Delay versus Network Size

Figure 4-18: Packet Delivery Ratio versus Network Size

Figure 4-18 shows the packet delivery ratio of the two protocols. Packet delivery ratio is calculated as the percentage of broadcast packets that are successfully received by all nodes in a network.

It can be observed that since CDS-MH algorithm reduce the redundancy of the transmission, compared to the simple flooding, it has the advantages to allow data packets to reach farther nodes quickly to avoid early sleeping and reducing the number of packets transmitted. As the result, the packet delivery ratio of AS-MAC when $\Delta T=0.015$ decreases dramatically when the grid size increases. When network size is small, the performances of AS-MAC with $\Delta T=0.015$ s and $\Delta T=0.02$ s are same. However, when network size increases, $\Delta T=0.015$ s is not long enough to overcome early sleeping problem, resulting in the farther nodes going to sleep early. When ΔT is set to 0.02s, the packet delivery ratio can be kept at the high-end.

We may derive from the analysis above that a variable ΔT with a non-linearity subject to node's distance n (see section 4.5.3.2 for details) might be more suitable in practice for avoiding severely degradation in packet delivery ratio.

Figure 4-19 shows the average power consumption of DW-MAC and AS-MAC. It can be seen that for DW-MAC, energy consumption is fairly consistent with increase of the network size because the duration of Data period is fixed and the active time per cycle is fixed regardless whether there is data transmission. However, for AS-MAC, the average power consumption is linearly increasing because the duration of active time is changing according to the distance of a node away from the sink node. With in broadcast traffic, the farther the node is away from the sink node, the longer it will have to stay awake (T_0) to wait for data transmission. When T_0 of a node is longer than the duration of Data period in DW-MAC, which is 0.168s, the power consumption of that node will be higher than that of each node by the DW-MAC. Since the value of T_0 can be flexible and the maximum value is the duration of the AS-period, which is 0.4s, when the network size is large, T_0 value for farther nodes can increase up to 0.4s, which makes average power consumption increasing dramatically. For a smaller network, the energy consumption is much lower since T_0 of each node is less than 0.168s. From **Figure 4-19**, it is clear that the acceleration of energy consumption by AS-MAC increases when ΔT increases.

Figure 4-19: Power Consumption versus Network Size

It is observable from the analysis above that introduction of the new parameters into the AS-MAC protocol, on the one hand, enables it to achieve better performance, such

as End-to-End delay, PDR and energy consumption; on the other hand, makes the AS-MAC more sensitive to the parameters values like T_0 , ΔT , etc. Since DW-MAC does not have these parameters, its power consumption presents a more stable performance for the different network sizes. This is a trade-off between comprehensive performance improvement and robustness of the protocol. A scheme that is more robust to the parameters newly introduced, at the same time presents comprehensive performance improvement is yet to be developed.

4.7 Summary

In this chapter, a MAC protocol with an adaptive scheduling scheme has been proposed. By the adaptive scheduling, AS-MAC allows nodes to schedule more data transmissions within one operational cycle when traffic load is heavy and go to sleep early when traffic load is light. We have compared the performance of AS-MAC and DW-MAC by simulation experiments. And the results have shown that the AS-MAC outperforms DW-MAC in energy efficiency, average end-to-end delay as well as packet delivery ratio for the unicast traffic scenarios. Under unicast traffic load with Event Interval (EI) = 200s, AS-MAC can reduce end-to-end delay by 65%, lower energy consumption by 40% compared to DW-MAC, while keeping nearly 100% packet delivery ratio. Under higher unicast traffic loads with EI = 150s, 100s, 50s, 30s and 20s, AS-MAC can still presents competitive performance advantages over DW-MAC with EI=200s in the above three aspects of performance. Under broadcast traffic, the results have shown that the AS-MAC could perform better than DW-MAC in a smaller size network. The end-to-end delay is much lower than that of DW-MAC. The packet delivery ratio and energy consumption, especially in a large network, could be controlled by adjusting some protocol parameters, such as a non-linear ΔT subject to node's distance from the sink.

In the future, more simulation work would be conducted to test the performance of AS-MAC for different layouts and scales of networks and under broadcast scenario. Experimental testing of the AS-MAC performance on real sensor network will also be conducted to evidence the applicability of the AS-MAC to the real world problem. The wireless sensor networks built with the features of high energy-efficiency, high delivery ratio, and low-latency will have great potential applicability in mid to large scale of

WSN applications, such as environmental monitoring, remote critical equipment monitoring and control in manufacturing.

5 Performance Evaluation of AS-MAC Protocol with Different AS-Period in Multi-hop Networks

In this chapter, we evaluate the performance of AS-MAC protocol with different AS-period in multi-hop network.

5.1 Protocol Configuration

Since the Resilient Active Time (RAT) mechanism in AS-period plays a critical role to achieve good performance and adaptability to the varying traffic loads, and AS-period and timeout value are two decisive components for the effect of RAT mechanism, and Event Interval (EI) is one of key factors for the traffic load, we conducted simulations for AS-period 400ms ($T_{as}=400ms$), 528ms ($T_{as}=528ms$), and 800ms ($T_{as}=800ms$) under varying traffic loads ranging from EI=200s to EI=5s to compare the performance. **Table 5-1** shows the duty cycle configuration for $T_{as}=400ms$, $T_{as}=528ms$ and $T_{as}=800ms$. The AS-MAC protocol presented in Chapter 4 has proven that the ratio of T_{AS}/T_{SLEEP} should keep ≤ 0.136 in order to ensure that there is no data-ACK collision in two-hop nodes and reduced possibility of data-ACK collision for multi-hop transmissions. The ratios 0.1 of T_{AS}/T_{SLEEP} for $T_{as}=400ms$, 0.136 for $T_{as}=528ms$, and 0.136 for $T_{as}=800ms$ are purposely taken in order to test AS-MAC performance with different ratios.

Table 5-1: Duty Cycle Configuration

AS-MAC	T_{SYNC} (ms)	T_{AS} (ms)	T_{SLEEP} (ms)	T_{CYCLE} (ms)	Ratio of T_{AS}/T_{SLEEP}
$T_{as}=400ms$	55.2	400	4009.8	4465.0	0.1
$T_{as}=528ms$	55.2	528	3881.8	4465.0	0.136
$T_{as}=800ms$	55.2	800	5882.8	6738.0	0.136

We used the same network parameters as we used in performance evaluation between AS-MAC and DW-MAC for our simulation of AS-MAC with different AS-periods. **Table 5-2** shows the networking parameters.

Table 5-2: Networking Parameters

Bandwidth	20Kbps	Ch. Encoding Ratio	2
Tx Power	31.2mW	Tx Range	250m
Rx Power	22.2mW	Carrier Sensing Range	550m
Idle Power	22.2mW	Contention Window (CW)	64ms
Sleep Power	3 μ W	Size of ACK	10B
SIFS	5ms	Size of SCH	15B
DIFS	10ms	Size of Data	100B

Retry Limit	5	State Transition Power	31.2mW
T_0	80ms	State Transition Time	2.47ms

We set data packet size of 100 bytes for AS-MAC. The longer packet size can be handled by increasing SCH frame size. We use the shortest path routing function available in OMNET++ library. In evaluating power efficiency, we focus on power consumption by communication and ignore power consumption by other components. The propagation delay is neglected. It is assumed that all nodes are synchronized with each other. The power consumption for SYNC period is assumed to include the average of Tx Power, Rx Power and Idle Power, which is 25.2mW.

The network layout is a 49-node (7x7) grid network with each node is 200 meters from its neighbors and the sink is at the center. All data packets generated are sent to the sink node. The length of paths that each data packet traverses ranges from 1 to 6 hops. To avoid early sleeping, T_0 is set equal to 80ms, which is greater than the time required for a SCH frame to traverse 6 hops.

Every EI (from 200s to 5s) interval, there is an event triggered at a random location with coordinates being generated using uniform random distribution function. Any node with its sensing area covering the location will generate a data packet and send it back to the node. To obtain comparable data for AS-MAC with different T_{as} , we set the sensing range for each node to be from 100m to 500m. By increasing the sensing range, the traffic load increases since more sensors can sense the event and generate more data packets. The results were taken as the average of 10 random runs, each of which has 500 events triggered.

5.2 Simulation Results and Discussions

5.2.1 Energy Efficiency

Figure 5-1 (a), (b) and (c) show the average power consumption of AS-MAC protocol with $T_{as}=400ms$, $T_{as}=800ms$ and $T_{as}=528ms$ respectively for different EIs ranging from 200s to 5s. **Figure 5-2** (a) shows the difference between the average power consumption of AS-MAC with $T_{as}=400ms$ and $T_{as}=800ms$, and **Figure 5-2** (b) shows the difference between $T_{as}=400ms$ and $T_{as}=528ms$ for different EIs ranging from 200s to 5s. The difference of average power consumption in **Figure 5-2** (a) is obtained by deducting average power consumption of $T_{as}=400ms$ from that of $T_{as}=800ms$

(represented by Tas800-Tas400). Thus those areas below zero indicate that AS-MAC with Tas=800ms consumes less energy than that with Tas=400ms. Vice versa, those areas above zero indicate that AS-MAC with Tas=800ms consumes more energy than that with Tas=400ms. **Figure 5-2** (b) is obtained in the same way for the difference between Tas=400ms and Tas=528ms.

(a) Average Power Consumption (Tas=400ms)

(b) Average Power Consumption (Tas=800ms)

(c) Average Power Consumption (Tas=528ms)

Figure 5-1: Average Power Consumption (mW) for Different EIs from 200s to 5s

(a) Ave Power Consumption Difference (mW) (Tas800-Tas400)

(b) Ave Power Consumption Difference (mW) (Tas528-Tas400)

Figure 5-2: Average Power Consumption Difference (mW)

From **Figure 5-1** (a), (b) and **Figure 5-2** (a), we may have the following observations for the power consumptions when AS-period is doubled from 400ms to 800ms:

- 1) For majority of traffic loads, the power consumptions reduce about 0.1 ~ 0.3mW further. This is because (1) when T_{as} increases, T_{sleep} also increases proportionally but by the ratio of $0.136/1 = 7.353$ times, and thus increment of T_{sleep} is much larger than that of T_{as} in order to keep up the ratio 0.136 of T_{as} over T_{sleep} . The much larger increased T_{sleep} contributes to the more energy conservation. (2) The timeout-based RAT controls the nodes' actual active time in AS-period, which leads to under-utilization of allocated duty cycle and more energy conservation.
- 2) Only under very high traffic load scenario ($EI=5s$, Sensing Range=450m and 500m), the power consumption of $T_{as}=800ms$ exceeds that of $T_{as}=400ms$ about 0.1mW, but still very competitive. This is because the nodes' actual active time in AS-period increases in order to handle the high traffic load, which is necessary and leads to high-utilization of allocated duty cycle and the increased energy consumption.

From **Figure 5-1** (a), (c) and **Figure 5-2** (b), we may have the following observations for the power consumptions when AS-period is changed from 400ms to 528ms:

- 1) For majority of traffic load, the power consumptions increase: for low to medium traffic load the power consumptions increase up to 0.1mW; for high traffic load the power consumptions increase up to 0.3mW. This is because (1) when T_{as} increases, T_{sleep} has to decrease in order to keep the same T_{cycle} , therefore the actual active time for $T_{as}=528ms$ increases, especially with the increased traffic load. (2) Although the timeout-based RAT controls the nodes' actual active time in AS-period, the increased traffic load increases the utilization of allocated duty cycle which in turn increases energy consumptions.
- 2) Only in some scenarios of low traffic loads (see light green areas), the power consumptions of $T_{as}=528ms$ decrease up to 0.1mW. This is because the timeout-based RAT controls the nodes' actual active time in AS-period, which leads to under-utilization of allocated duty cycle and energy conservation.

5.2.2 Packet Delivery Ratio (PDR)

5.2.2.1 PDR Comparison between Tas400 and Tas800

The **Figure 5-3** (a) and (b) displays the PDR under low-medium Unicast traffic load with EI from 200s to 20s respectively for Tas=400ms and Tas=800ms. The **Figure 5-4** (a) and (b) displays the PDR under high Unicast traffic with EI from 10s to 5s respectively for Tas=400ms and Tas=800ms. **Figure 5-5** shows the PDR difference between Tas=400ms and Tas=800ms. The PDR difference in **Figure 5-5** is obtained by deducting PDR of Tas=400ms from that of Tas=800ms (represented by Tas800-Tas400). Thus those areas above zero indicate that AS-MAC with Tas=800ms has the higher PDR than that with Tas=400ms. Vice verse, those areas below zero indicate that AS-MAC with Tas=800ms has the lower PDR than that with Tas=400ms.

(a) PDR under Low-medium Unicast Traffic (Tas=400ms) (b) PDR under Low-medium Unicast Traffic (Tas=800ms)

Figure 5-3: Packet Delivery Ratio (PDR) (%) for Different EIs from 200s to 20s

(a) PDR under High Unicast Traffic (Tas=400ms)

(b) PDR under High Unicast Traffic (Tas=800ms)

Figure 5-4: Packet Delivery Ratio (PDR) (%) for Different EIs from 10s to 5s

Figure 5-5: Packet Delivery Ratio (PDR) Difference (%) (Tas800-Tas400)

From **Figure 5-5**, we may have the following observations for the PDR when AS-period is doubled from 400ms to 800ms:

- 1) For majority of the traffic loads, the PDR improve. **Figure 5-5** shows that such improvement is about from 1% to 12%. When traffic load is in high-end, the PDR improvement is more obvious and can increase from 4% up to more than 12%.
- 2) Only for certain medium to high traffic load (light blue portion), the PDR drop a bit, by about 0~2%.

The observed points indicate that a long operation cycle is beneficial for PDR in high traffic load, but not much in medium traffic load. This is because the long operation cycle drags the end-to-end delay (see discussion in section 5.2.3) for the medium traffic load, which indirectly affects the PDR for the medium traffic load. However, the protocol can leverage the long operation cycle to tackle the high traffic load, hence being beneficial for PDR as well.

5.2.2.2 PDR Comparison between Tas400 and Tas528

The **Figure 5-6** (a) and (b) displays the PDR under low-medium Unicast traffic load with EI from 200s to 20s respectively for Tas=400ms and Tas=528ms. The **Figure 5-7** (a) and (b) displays the PDR under high Unicast traffic load with EI from 10s to 5s respectively for Tas=400ms and Tas=528ms. **Figure 5-8** shows the PDR difference between Tas=400ms and Tas=528ms. The PDR difference in **Figure 5-8** is obtained by deducting PDR of Tas=400ms from that of Tas=528ms (represented by Tas528 -Tas400).

(a) PDR under Low-medium Unicast Traffic ($Tas=400ms$) (b) PDR under Low-medium Unicast Traffic ($Tas=528ms$)

Figure 5-6: Packet Delivery Ratio (PDR) (%) for Different EIs from 200s to 20s

(a) PDR under High Unicast Traffic ($Tas=400ms$)

(b) PDR under High Unicast Traffic ($Tas=528ms$)

Figure 5-7: Packet Delivery Ratio (PDR) (%) for Different EIs from 10s to 5s

Figure 5-8: Packet Delivery Ratio (PDR) Difference (%) ($Tas528-Tas400$)

From **Figure 5-8**, we may have the following observations for the PDR when AS-period is changed from 400ms to 528ms:

- 1) For almost all the traffic loads, the PDR improve. **Figure 5-8** shows that such improvement is in the range from 1% to 14%. When traffic load is in high-end, the PDR improvement is more obvious and can increase from 4% up to more than 14%.
- 2) Only for a few medium to high traffic load scenarios (light blue portion), the PDR drop a bit, by about 0~2%. But the PDR with $T_{as}=528\text{ms}$ improves than the PDR with $T_{as}=800\text{ms}$ as much less spot areas are in the range of -2~0%.

From PDR comparison between $T_{as}=400\text{ms}$ and $T_{as}=800\text{ms}$ and PDR comparison between $T_{as}=400\text{ms}$ and $T_{as}=528\text{ms}$, it can be summarized that (1) the longer operation cycle do beneficial to the PDR for high traffic load, but not much for medium traffic loads. (2) In the circumstances of the same operation cycle, a proper long AS-period (for this case $T_{as}=528\text{ms}$ with ratio 0.136 for T_{as}/T_{sleep}) is more beneficial to the PDR.

5.2.3 End-to-End Delay

5.2.3.1 End-to-End Comparison between $T_{as}400$ and $T_{as}800$

The **Figure 5-9** (a) and (b) displays the average end-to-end delay under all Unicast traffic load with EI from 200s to 5s respectively for $T_{as}=400\text{ms}$ and $T_{as}=800\text{ms}$. **Figure 5-10** shows the average end-to-end delay difference between $T_{as}=400\text{ms}$ and $T_{as}=800\text{ms}$. The average end-to-end delay difference is obtained by deducting the average end-to-end delay of $T_{as}=400\text{ms}$ from that of $T_{as}=800\text{ms}$ (represented by $T_{as}800-T_{as}400$). For the average end-to-end delay, the smaller, the better. Thus those areas above zero indicate that AS-MAC with $T_{as}=800\text{ms}$ has the longer average end-to-end delay than that with $T_{as}=400\text{ms}$. Vice verse, those areas below zero indicate that AS-MAC with $T_{as}=800\text{ms}$ has the shorter average end-to-end delay than that with $T_{as}=400\text{ms}$.

(a) Ave. End-to-End Delay of Different EIs (Tas=400ms) (b) Ave. End-to-End Delay of Different EIs (Tas=800ms)

Figure 5-9: Average End-to-End Delay of Different EIs

Figure 5-10: Average End-to-End Delay Difference of Different EIs (Tas800-Tas400)

The **Figure 5-11** (a) and (b) displays the max end-to-end delay under all Unicast traffic load with EI from 200s to 5s respectively for Tas=400ms and Tas=800ms. **Figure 5-12** shows the max end-to-end delay difference between Tas=400ms and Tas=800ms. The max end-to-end delay difference is obtained by deducting max end-to-end delay of Tas=400ms from that of Tas=800ms (represented by Tas800-Tas400). For the max end-to-end delay, the smaller, the better. Thus those areas above zero indicate that AS-MAC with Tas=800ms has the longer max end-to-end delay than that with Tas=400ms. Vice versa, those areas below zero indicate that AS-MAC with Tas=800ms has the shorter max end-to-end delay than that with Tas=400ms.

(a) Max End-to-End Delay of Different EIs (Tas=400ms) (b) Max End-to-End Delay of Different EIs (Tas=800ms)

Figure 5-11: Max End-to-End Delay of Different EIs

Figure 5-12: Max End-to-End Delay Difference (s) (Tas800-Tas400)

From **Figure 5-10** and **Figure 5-12**, we may have the following observations for the end-to-end delay when AS-period is doubled from 400ms to 800ms:

- 1) The average end-to-end delay and max end-to-end delay both increase for majority of traffic loads. The average end-to-end delay increases up to 5.5s; and max end-to-end delay increases up to 64s. This is because the operational cycle increases, for this case, from 4465ms to 6738ms, which cause the increased end-to-end delay.
- 2) Only when traffic load is very high (Sensing Range=450m and 500m, EI=6s and 5s), the end-to-end delay are reduced significantly for both average end-to-end delay and max end-to-end delay. This is because AS-MAC with a longer AS-period can handle the higher traffic load that cannot be tackled well by the AS-MAC with a shorter AS-period. The average end-to-end delay is shortened up to 16s for the higher traffic load; the max end-to-end delay is shortened up to 60s (actual max end-to-end delay is shortened up to 1735s which is not shown in the figure due to the space constraint).

5.2.3.2 End-to-End Comparison between Tas400 and Tas528

The **Figure 5-13** (a) and (b) displays the average end-to-end delay under all Unicast traffic load with EI from 200s to 5s respectively for Tas=400ms and Tas=528ms. **Figure 5-14** shows the average end-to-end delay difference between Tas=400ms and Tas=528ms. The average end-to-end delay difference is obtained by deducting the average end-to-end delay of Tas=400ms from that of Tas=528ms (represented by Tas528-Tas400).

(a) Ave. End-to-End Delay of Different EIs (Tas=400ms) (b) Ave. End-to-End Delay of Different EIs (Tas=528ms)

Figure 5-13: Average End-to-End Delay of Different EIs

Figure 5-14: Average End-to-End Delay Difference of Different EIs (Tas528-Tas400)

The **Figure 5-15** (a) and (b) displays the max end-to-end delay under all Unicast traffic load with EI from 200s to 5s respectively for Tas=400ms and Tas=528ms. **Figure 5-16** shows the max end-to-end delay difference between Tas=400ms and Tas=528ms. The max end-to-end delay difference is obtained by deducting max end-to-end delay of Tas=400ms from that of Tas=528ms (represented by Tas538-Tas400).

(a) Max End-to-End Delay of Different EIs (Tas=400ms) (b) Max End-to-End Delay of Different EIs (Tas=528ms)

Figure 5-15: Max End-to-End Delay of Different EIs

Figure 5-16: Max End-to-End Delay Difference (s) (Tas528-Tas400)

From **Figure 5-14** and **Figure 5-16**, we may have the following observations for the end-to-end delay when AS-period is changed from 400ms to 528ms:

- 1) The average end-to-end delay and max end-to-end delay both decrease for majority of traffic loads. The average end-to-end delay decreases up to 31s; and max end-to-end delay decreases up to 160s (actual max end-to-end delay decreases up to 2012s which is not shown in the figure due to the space constraint). This is because the operational cycle keeps the same, but Tas increases from 400ms to 528ms, which enables AS-MAC to be able to handle traffic more promptly, thus reducing the end-to-end delay.
- 2) When comparing the end-to-end delay between Tas=800ms and Tas=528ms, we may see that for the same ratio of Tas over Tsleep, the protocol with shorter operation cycle is more beneficial to the end-to-end delay than the longer operation cycle.

It is observable from comparison figures [Figure 5-2](#), [Figure 5-5](#), [Figure 5-8](#), [Figure 5-10](#), [Figure 5-12](#), [Figure 5-14](#) and [Figure 5-16](#) that change in the AS-period (T_{as}) and operation cycle (T_{cycle}) has the following impacts for the different aspects of AS-MAC performance:

- The timeout-based RAT mechanism is very beneficial to the energy conservation for different traffic load. It ensures the energy-efficient use of the allocated duty-cycle regardless the traffic load scenarios.
- In the circumstance of the same T_{cycle} , the AS-MAC with a shorter T_{as} can have more energy conservation than that with a longer T_{as} (see [Figure 5-2](#) (b)). However, when T_{cycle} increases, T_{as} and T_{sleep} increase accordingly, and T_{sleep} increases in the proportion of multiple times than T_{as} does. Hence the longer T_{sleep} leads to more energy conservations. On the other hands, although T_{as} also increases in the longer T_{cycle} scenario, AS-MAC can effectively control the actual active time with the RAT mechanism, therefore won't lead to the energy-inefficient use of the allocated duty-cycle (see [Figure 5-2](#) (a)).
- The longer T_{cycle} is indeed beneficial to the PDR for high traffic load, but not much for medium traffic loads (see [Figure 5-5](#)). In the circumstances of the same operation cycle, a proper longer T_{as} (for this case $T_{as}=528\text{ms}$ with ratio 0.136 for T_{as}/T_{sleep}) is more beneficial to the PDR, especially for medium to high traffic load scenarios (see [Figure 5-8](#)), as the proper longer T_{as} provides a plentiful time for the protocol to transmit the control frames and deliver data packets within lesser operation cycles, thus increasing the PDR. And a proper longer T_{cycle} and T_{as} do not have much negative impacts on the end-to-end delay (see [Figure 5-14](#) and [Figure 5-16](#)).
- The longer T_{cycle} causes the increased average end-to-end delay and max end-to-end for low to medium traffic loads. This is because the longer operation cycle unavoidably results in the increased network latency especially for low to medium traffic loads as the allocated active times are not utilized efficiently. Only when traffic load is very high, the allocated AS-period and Sleep period are fully made use of for disseminating the control frames and data packets, which leads to significant reduction in both the average and max end-to-end delay (see [Figure 5-10](#) and [Figure 5-12](#)). In the circumstances of the same operation cycle, a proper longer T_{as} (for this case $T_{as}=528\text{ms}$ with ratio 0.136 for T_{as}/T_{sleep}) can shorten the end-

to-end delay more significantly, not only for high traffic load but for low to medium to traffic load as well (see [Figure 5-14](#) and [Figure 5-16](#)), as the proper longer T_{as} provides a plentiful time for the protocol to transmit the control frames and deliver data packets yet within the shorter T_{cycle} and lesser operation cycles, thus shortening the end-to-end delay.

5.3 Summary

It is nontrivial to achieve good performance in full spectrum of performance metrics at the same time. Energy efficiency, end-to-end delay reduction and packet delivery ratio (PDR) are very important facets of performance metrics. In this chapter, the performance of an Adaptive Scheduling MAC (AS-MAC) protocol with different Adaptive Scheduling period (AS-period) and operation cycle time for multi-hop networks have been presented in these important facets. AS-MAC achieves very good performance in energy-efficiency, end-to-end delay reduction and PDR, especially for the high traffic loads scenarios, and good performance trade-off in these facets. AS-MAC also exhibits adaptability to the different traffic loads with an adjustable RAT. Performance comparisons between $T_{as}400ms$ and $T_{as}800ms$ and between $T_{as}400ms$ and $T_{as}528ms$ show that AS-MAC protocol with a moderate long AS-period and operation cycle time will be beneficial to the energy efficiency, PDR and end-to-end delay for most of traffic loads.

Furthermore, when energy conservation is the most important concern, a longer operation cycle and thus a longer AS-period with the effective RAT mechanism can be employed in the protocol; while when PDR and end-to-end delay are major concerns of network performance, a shorter operation cycle time with a proper longer AS-period should be employed in the protocol in order to increase the PDR and shorten the end-to-end delay. Since the T_{as} and T_{cycle} can be easily parameterized in the algorithm implementation, we claim that AS-MAC protocol with key parameters adjustable on the fly through parameterization, such as T_{cycle} , T_{as} , T_{as}/T_{sleep} ratio, timeout value, etc. it is adaptable to the varying traffic scenarios and application requirements for multi-hop networks.

6 Performance Evaluation of AS-MAC Protocol with Different Timeout in Multi-hop Networks

As mentioned in the earlier chapter, AS-period T_{as} and timeout value T_0 in AS-MAC protocol are two important factors affecting the RAT and performance of the protocol and adaptability to the varying traffic loads. We have evaluated the impacts of AS-period on the performance in last chapter. In this chapter, we evaluate the performance of AS-MAC protocol with different timeout value T_0 in multi-hop network.

6.1 Duty Cycle and Network Parameters Configuration

Timeout value T_0 is the most direct factor determining the Resilient Active Time (RAT), we conducted simulations for Timeout value $T_0=0.08s$, $T_0=0.06s$ and $T_0=0.045s$ (25% shorter than the compared timeout) under varying traffic loads ranging from $EI=200s$ to $EI=5s$ to compare the performance. **Table 6-1** shows the duty cycle configuration for $T_0=0.08s$, $T_0=0.06s$ and $T_0=0.045s$.

The network layout is a 49-node (7x7) grid network with each node is 200 meters from its neighbors and the sink is at the center. All data packets generated are sent to the sink node. The length of paths that each data packet traverses ranges from 1 to 6 hops.

Table 6-1: Duty Cycle Configuration

T_{sync} (ms)	T_{AS} (ms)	T_{SLEEP} (ms)	T_{CYCLE} (ms)	Ratio of T_{AS}/T_{SLEEP}	Timeout (T_0)(ms)
55.2	528	3881.8	4465.0	0.136	80ms
55.2	528	3881.8	4465.0	0.136	60ms
55.2	528	3881.8	4465.0	0.136	45ms

We used the same network parameters as we used in performance evaluation between AS-MAC and DW-MAC for our simulation of AS-MAC with different timeout. **Table 6-2** shows the networking parameters.

Table 6-2: Networking Parameters

Bandwidth	20Kbps	Ch. Encoding Ratio	2
Tx Power	31.2mW	Tx Range	250m
Rx Power	22.2mW	Carrier Sensing Range	550m
Idle Power	22.2mW	Contention Window (CW)	64ms
Sleep Power	3 μ W	Size of ACK	10B
SIFS	5ms	Size of SCH	15B
DIFS	10ms	Size of Data	100B
Retry Limit	5	State Transition Power	31.2mW

Timeout	T_0	State Transition Time	2.47ms
---------	-------	-----------------------	--------

Every EI (from 200s to 5s) interval, there is an event triggered at a random location with coordinates being generated using uniform random distribution function. Any node with its sensing area covering the location will generate a data packet and send it back to the node. To obtain comparable data for AS-MAC with different Timeout value T_0 , we set the sensing range for each node to be from 100m to 500m. By increasing the sensing range, the traffic load increases since more sensors can sense the event and generate more data packets. The results were taken as the average of 10 random runs, each of which has 500 events triggered.

6.2 Simulation Results and Discussions

6.2.1 Energy Efficiency

Figure 6-1 (a), (b) and (c) show the average power consumption of AS-MAC protocol with $T_0=0.08s$, $T_0=0.06s$ and $T_0=0.045s$ respectively for different EIs ranging from 200s to 5s. **Figure 6-2** (a) shows the difference between the average power consumption of AS-MAC with $T_0=0.08s$ and $T_0=0.06s$. **Figure 6-2** (b) shows the difference between the average power consumption of AS-MAC with $T_0=0.06s$ and $T_0=0.045s$. The difference of average power consumption in **Figure 6-2** (a) is obtained by deducting average power consumption of $T_0=0.06s$ from that of $T_0=0.08s$ (represented by $T_{080ms}-T_{060ms}$). Thus those areas above zero indicate that AS-MAC with $T_0=0.08s$ consumes more energy than that with $T_0=0.06s$. Vice versa, those areas below zero indicate that AS-MAC with $T_0=0.08s$ consumes less energy than that with $T_0=0.06s$. Similarly, the difference of average power consumption in **Figure 6-2** (b) is obtained by deducting average power consumption of $T_0=0.045s$ from that of $T_0=0.06s$ (represented by $T_{060ms}-T_{045ms}$). Thus those areas above zero indicate that AS-MAC with $T_0=0.06s$ consumes more energy than that with $T_0=0.045s$. Vice versa, those areas below zero indicate that AS-MAC with $T_0=0.06s$ consumes less energy than that with $T_0=0.045s$.

Figure 6-1: Average Power Consumption (mW) for Different EIs from 200s to 5s

Figure 6-2: Average Power Consumption Difference (mW)

From **Figure 6-2** (a) and (b), we may have the following observations for the scenario that the Timeout T_0 changes from 80ms to 60ms (25% reduction) and that the Timeout T_0 changes from 60ms to 45ms (25% reduction):

- 1) For all the traffic loads within the scope of study, the power consumptions reduce about 0.1 ~ 0.25mW when Timeout T_0 changes from 80ms to 60ms (25%

reduction), and it further reduces about 0.075 ~ 0.3mW when Timeout T_0 changes from 60ms to 45ms (25% reduction). This is because when T_0 decreases, the resilient active time decreases too as the timeout-based RAT controls the nodes' actual active time in AS-period, and thus T_{sleep} will increase accordingly, which contribute to the energy conservation.

- 2) When timeout T_0 is shortened, there will be more energy conservation in the nodes with the shorter timeout than that with the longer timeout. Since the traffic loads for $T_0=80ms$, $T_0=60ms$ and $T_0=45ms$ are the same, the same traffic load will make the nodes with the longer timeout value to sleep later than the nodes with the shorter timeout value, which result in the shorter sleep time for the nodes with $T_0=80ms$ than for the nodes with $T_0=60ms$, which in turn have the shorter sleep time than the nodes with $T_0=45ms$. Thus there will be more energy saving for the shorter timeout value with increased traffic load.
- 3) However, this energy saving will be paid off by the increased end-to-end delay, which can be seen in the following session.

6.2.2 Packet Delivery Ratio (PDR)

6.2.2.1 PDR Comparison between $T_0=80ms$ and $T_0=60ms$

The **Figure 6-3** (a) and (b) displays the PDR under low-medium Unicast traffic load with EI from 200s to 20s respectively for $T_0=80ms$ and $T_0=60ms$. The **Figure 6-4** (a) and (b) displays the PDR under high Unicast traffic with EI from 10s to 5s respectively for $T_0=80ms$ and $T_0=60ms$. **Figure 6-5** shows the difference of the PDR between $T_0=80ms$ and $T_0=60ms$. The PDR difference in **Figure 6-5** is obtained by deducting PDR of $T_0=60ms$ from that of $T_0=80ms$ (represented by $(T_0=80ms - T_0=60ms)$). Thus those areas above zero indicate that AS-MAC with $T_0=80ms$ has the higher PDR than that with $T_0=60ms$. Vice verse, those areas below zero indicate that AS-MAC with $T_0=80ms$ has the lower PDR than that with $T_0=60ms$.

Figure 6-3: Packet Delivery Ratio (PDR) (%) for Different EIs from 200s to 20s

Figure 6-4: Packet Delivery Ratio (PDR) (%) for Different EIs from 10s to 5s

Figure 6-5: Packet Delivery Ratio (PDR) Difference (%) (T0=80ms-T0=60ms)

From Figure 6-5, we may have the following observations for the PDR when timeout is changed from T0=80ms and T0=60ms:

For the majority of traffic load scenarios, the PDR of $T_0=80\text{ms}$ is about 1% higher than that of $T_0=60\text{ms}$; for the high traffic load scenario, the PDR of $T_0=80\text{ms}$ is about 1%~2% higher than that of $T_0=60\text{ms}$; Only for a few cases the PDR of $T_0=80\text{ms}$ is about 1% lower than that of $T_0=60\text{ms}$. This is reasonable results, after all the shorter timeout value bring about a shorter RAT, which slow down the nodes' response to the traffic, thus affect the success package delivery, especially with the increased traffic loads.

6.2.2.2 PDR Comparison between $T_0=60\text{ms}$ and $T_0=45\text{ms}$

The **Figure 6-6** (a) and (b) displays the PDR under low-medium Unicast traffic load with EI from 200s to 20s respectively for $T_0=60\text{ms}$ and $T_0=45\text{ms}$. The **Figure 6-7** (a) and (b) displays the PDR under high Unicast traffic with EI from 10s to 5s respectively for $T_0=60\text{ms}$ and $T_0=45\text{ms}$. **Figure 6-8** shows the difference of PDR between $T_0=60\text{ms}$ and $T_0=45\text{ms}$. Similarly, the PDR difference in **Figure 6-8** is obtained by deducting PDR of $T_0=45\text{ms}$ from that of $T_0=60\text{ms}$ (represented by $(T_0=60\text{ms}-T_0=45\text{ms})$). Thus those areas above zero indicate that AS-MAC with $T_0=60\text{ms}$ has the higher PDR than that with $T_0=45\text{ms}$. Vice verse, those areas below zero indicate that AS-MAC with $T_0=60\text{ms}$ has the lower PDR than that with $T_0=45\text{ms}$.

(a) PDR under low-medium Unicast Traffic
($T_{as}=528\text{ms}$, $T_0=60\text{ms}$)

(b) PDR under low-medium Unicast Traffic
($T_{as}=528\text{ms}$, $T_0=45\text{ms}$)

Figure 6-6: Packet Delivery Ratio (PDR) (%) for Different EIs from 200s to 20s

Figure 6-7: Packet Delivery Ratio (PDR) (%) for Different EIs from 10s to 5s

Figure 6-8: Packet Delivery Ratio (PDR) Difference (%) (T0=60ms-T0=45ms)

From **Figure 6-8**, we may have the following observations for the PDR when timeout is changed from $T_0=60\text{ms}$ and $T_0=45\text{ms}$:

For the majority of traffic load scenarios, the PDR of $T_0=60\text{ms}$ is about 1% higher than that of $T_0=45\text{ms}$; for the high traffic load scenario, the PDR of $T_0=60\text{ms}$ is about 1%~4% higher than that of $T_0=45\text{ms}$; Only for a few cases the PDR of $T_0=60\text{ms}$ is about 1% lower than that of $T_0=45\text{ms}$. This is the evidence again that the shorter timeout value slow down the nodes' response to the traffic, thus further affect the success package delivery. The negative impact becomes more serious with the increased traffic loads.

6.2.3 End-to-End Delay

6.2.3.1 End-to-End Comparison between $T_0=80\text{ms}$ and $T_0=60\text{ms}$

The Figure 6-9 (a) and (b) displays the average end-to-end delay under all Unicast traffic load with EI from 200s to 5s and $T_{as}=528\text{ms}$ respectively for $T_0=80\text{ms}$ and $T_0=60\text{ms}$. Figure 5-10 shows the difference of average end-to-end delay between $T_0=80\text{ms}$ and $T_0=60\text{ms}$. The average end-to-end delay difference is obtained by deducting the average end-to-end delay of $T_0=60\text{ms}$ from that of $T_0=80\text{ms}$ (represented by $(T_{080\text{ms}}-T_{060\text{ms}})$). For the average end-to-end delay, the smaller, the better. Thus those areas above zero indicate that AS-MAC with $T_0=80\text{ms}$ has the longer average end-to-end delay than that with $T_0=60\text{ms}$. Vice versa, those areas below zero indicate that AS-MAC with $T_0=80\text{ms}$ has the shorter average end-to-end delay than that with $T_0=60\text{ms}$.

Figure 6-9: Average End-to-End Delay of Different EIs

Figure 6-10: Average End-to-End Delay Difference of Different EIs ($T_{080\text{ms}}-T_{060\text{ms}}$)

From Figure 5-10, we may have the following observations for the End-to-End Delay when timeout is changed from $T_0=80\text{ms}$ and $T_0=60\text{ms}$: For majority of the traffic loads, the average end-to-end delay increase a bit by up to 0.5s. For a few cases in high traffic load, the average end-to-end delay increase from 1s to 2.5s. This is a payoff of gain in the energy saving for the high traffic load scenario.

The **Figure 6-11** (a) and (b) displays the max end-to-end delay under all Unicast traffic load with EI from 200s to 5s respectively for $T_0=80\text{ms}$ and $T_0=60\text{ms}$. **Figure 6-12** shows the difference of max end-to-end delay between $T_0=80\text{ms}$ and $T_0=60\text{ms}$. Similarly, the max end-to-end delay difference is obtained by deducting max end-to-end delay of $T_0=60\text{ms}$ from that of $T_0=80\text{ms}$ (represented by $(T_0=80\text{ms}-T_0=60\text{ms})$). For the max end-to-end delay, the smaller, the better. Thus those areas above zero indicate that AS-MAC with $T_0=80\text{ms}$ has the longer max end-to-end delay than that with $T_0=60\text{ms}$. Vice verse, those areas below zero indicate that AS-MAC with $T_0=80\text{ms}$ has the shorter max end-to-end delay than that with $T_0=60\text{ms}$.

Figure 6-11: Max End-to-End Delay for different EIs

Figure 6-12: Max End-to-End Delay Difference (s) ($T_0=80\text{ms}-T_0=60\text{ms}$)

From **Figure 6-12**, we may have the following observations for the Max End-to-End Delay when timeout is changed from $T_0=80\text{ms}$ and $T_0=60\text{ms}$: For a few cases, the maximum end-to-end delay decrease up to 20s; for the majority of the low to medium traffic loads, the maximum end-to-end delay increase up to 70s. For some cases of high traffic load, the maximum end-to-end delay increase from 70s to 250s.

6.2.3.2 End-to-End Comparison between $T_0=60\text{ms}$ and $T_0=45\text{ms}$

The **Figure 6-13** (a) and (b) displays the average end-to-end delay under all Unicast traffic load with EI from 200s to 5s and $T_{as}=528\text{ms}$ respectively for $T_0=60\text{ms}$ and $T_0=45\text{ms}$. **Figure 6-14** shows the average end-to-end delay difference between $T_0=60\text{ms}$ and $T_0=45\text{ms}$. The average end-to-end delay difference is obtained by deducting the average end-to-end delay of $T_0=45\text{ms}$ from that of $T_0=60\text{ms}$ (represented by $(T_0=60\text{ms}-T_0=45\text{ms})$). For the average end-to-end delay, the smaller, the better. Thus those areas above zero indicate that AS-MAC with $T_0=60\text{ms}$ has the longer average end-to-end delay than that with $T_0=45\text{ms}$. Vice versa, those areas below zero indicate that AS-MAC with $T_0=60\text{ms}$ has the shorter average end-to-end delay than that with $T_0=45\text{ms}$.

From **Figure 6-14**, we may have the following observations for the End-to-End Delay when timeout is changed from $T_0=80\text{ms}$ and $T_0=60\text{ms}$: For majority of the traffic loads, the average end-to-end delay increase a bit by up to 1s. For the high traffic load, the average end-to-end delay increase from 1s to 8s. This is a payoff of gain in the energy saving for the high traffic load scenario.

Figure 6-13: Average End-to-End Delay of Different EIs

Figure 6-14: Average End-to-End Delay Difference of Different EIs ($T_0=60\text{ms}-T_0=45\text{ms}$)

The **Figure 6-15** (a) and (b) displays the max end-to-end delay under all Unicast traffic load with EI from 200s to 5s respectively for $T_0=60\text{ms}$ and $T_0=45\text{ms}$. **Figure 6-16** shows the max end-to-end delay difference between $T_0=60\text{ms}$ and $T_0=45\text{ms}$. The max end-to-end delay difference is obtained by deducting max end-to-end delay of $T_0=45\text{ms}$ from that of $T_0=60\text{ms}$ (represented by $T_0=60\text{ms}-T_0=45\text{ms}$). For the max end-to-end delay, the smaller, the better. Thus those areas above zero indicate that AS-MAC with $T_0=60\text{ms}$ has the longer max end-to-end delay than that with $T_0=45\text{ms}$. Vice versa, those areas below zero indicate that AS-MAC with $T_0=60\text{ms}$ has the shorter max end-to-end delay than that with $T_0=45\text{ms}$.

From **Figure 6-16**, we may have the following observations for the end-to-end delay when timeout changes from 60ms to 45ms: For some low traffic load cases, the maximum end-to-end delay decrease up to 20s; for the majority of the medium to high traffic loads, the maximum end-to-end delay increase up to 70s. For some cases of high traffic load, the maximum end-to-end delay increase from 70s to 250s.

(a) Max End-to-End Delay for different EI ($T_{as}=528\text{ms}$, $T_0=60\text{ms}$)

(b) Max End-to-End Delay for different EI ($T_{as}=528\text{ms}$, $T_0=45\text{ms}$)

Figure 6-15: Max End-to-End Delay for different EIs

Figure 6-16: Max End-to-End Delay Difference (s) ($T_0 60\text{ms} - T_0 45\text{ms}$)

It is observable from **Figure 6-10** and **Figure 6-12**, **Figure 6-14** and **Figure 6-16** that when the timeout T_0 is shortened:

- For the same traffic load, the shorter the timeout is, the more energy conservation nodes can achieve. However, the energy saving will be paid-off by the increased end-to-end delay.
- Shortening timeout value does not help to improve the PDR; in contrast, if it is too short it will affect the PDR. This is because if the timeout value in AS-period is too short, the node won't have enough time for nodes to schedule rendezvous with their downstream nodes and handle the traffic load, especially for the high traffic load scenarios, which will make some packets not able to send through and affect PDR.
- A short timeout value for AS-period definitely increases the average end-to-end delay and max end-to-end delay, especially for the high traffic load. This is again because the short timeout value in AS-period does not give the node enough time to handle the traffic load, especially for the high traffic load scenarios, which result in the longer end-to-end delay.

7 An Energy-Efficient and Self-Adaptive Duty Cycle MAC Protocol for Traffic Dynamic WSNs

7.1 Motivation of Proposed Scheme

While the AS-MAC presented in Chapter 4 has significantly improved the performance of DW-MAC from all aspects – power consumption, end-to-end delay and packet delivery ratio (PDR), we think AS-MAC can be further improved because of its fixed duty cycle approach adopted in its implementation.

The fixed duty cycle, especially low duty cycle scheme has three drawbacks. First, it will cause sleep latency if the duty cycle level is not enough to handle traffic flow promptly. The sleep latency increases proportionally with increasing number of hops. Second, the fixed duty cycle cannot adapt to the traffic variation. A fixed low duty cycle is designated for low traffic load to save energy, but it will result in high end-to-end delay and low packet delivery ratio (PDR) for the high traffic load. In the circumstance of very low traffic load, it will result in inefficient duty cycle problem. Vice versa, a fixed high duty cycle might help reduce end-to-end delay and increase PDR, but it will result in significant energy wastage and inefficient duty cycle problem when traffic becomes low. Thus, it is desirable to adopt an adaptive duty cycle for variant traffic load in MAC protocol design. Third, a fixed synchronous duty cycle may increase the possibility of collision. If neighbouring nodes sense signals and turn to active mode at the same time, then all may contend for the channel, making a collision very likely.

7.2 Executive Summary of Proposed Scheme

In this Chapter, we present a new MAC protocol, called Self-Adaptive Duty Cycle MAC (SEA-MAC) protocol that derives from AS-MAC. Similar to AS-MAC, the SEA-MAC protocol adopts basic scheme of AS-MAC: (1) Contending for time slots during the Adaptive Scheduling (AS) period of a duty cycle to schedule data transmission for Sleep period and using timeout to realize Resilient Active Time (RAT) for AS-period. (2) Waking up on demand during the Sleep period of an operation cycle to transmit or receive data packets and employing a proportional mapping function to prevent data collision. This basic scheme enable SEA-MAC to adaptively prolong the RAT length so

as to allow nodes to schedule more data transmission within one operational cycle when the traffic load is increasing, and shorten the RAT length and go to sleep early when the traffic load is decreasing. Differentially from AS-MAC, (3) SEA-MAC further adjusts the duty cycle dynamically based on the traffic load, so as to mitigate inefficient duty cycle problem in the circumstances of low traffic load, or reduce end-to-end delay in circumstances of high traffic load. SEA-MAC provides nodes with just enough active time for sending and receiving packets in the provided duty cycle. Experimental results show that SEA-MAC is superior not only to its former protocol AS-MAC for majority of traffic scenarios, but also to a representative MAC protocol of another type, RI-MAC (Receiver-Initiated MAC) for majority of traffic scenarios.

The significance of this work can be summarized as follows:

- Inefficient duty cycle has been conceptualized first time in MAC protocol design and identified as a problem that impacts the MAC protocol performance.
- Inefficient duty cycle has been quantified so as to become calculable factor.
- An adjustable duty cycle scheme has been realized to enable the self-adaptive duty cycle MAC protocol.

7.3 Direct Related Work

7.3.1 Related Work to Basic Scheme

The basic scheme of SEA-MAC is based on the AS-MAC which is related to the works of T-MAC, S-MAC, S-MAC-AL and DW-MAC. They all improved the performance of the protocols to which they compared. With the aim to further improve the drawbacks in the DW-MAC scheme, the AS-MAC scheme has introduced an Adaptive Scheduling period (AS-period) to make protocol adaptive to the varying traffic load so that it can achieve much less energy consumption and much lower end-to-end delay. AS-MAC employs timeout-based Resilient Active Time (RAT) mechanism, dual-role SCH frame with tracked retry number, enhanced proportional mapping and adjustable timeout mechanism; and innovatively combines timeout-based RAT and enhanced proportional mapping in the protocol design, so that it can effectively and efficiently handle both unicast and broadcast traffic.

7.3.2 Related Work to Self-Adaptive Duty Cycle

Although the above MAC protocols' performances improve with their schemes' evolution over and over, one of inherent common features is that they all base on the fixed duty cycle which limits the further improvement of protocol performance. Adjustable duty cycle might be the last resort to make further improvement of protocol performance. Although it has been stated in [10] that duty cycle can be manually changed according to application requirements, we believe that there is need for a generic MAC protocol which does not assume any prior knowledge of application requirements but can tune its duty cycle automatically according to traffic conditions. Several solutions using adjustable duty cycle have been proposed [106] [107] [108] [109]. These solutions either achieved tradeoff performance compared to S-MAC, i.e. gain in one aspect of performance but compromise another aspect of it [106] [108] or achieved minor performance improvement [107] [109].

Dynamic Sensor-MAC (DSMAC) [106] extends S-MAC by adding dynamic duty cycle feature in order to reduce the latency for delay-sensitive applications. In DSMAC, all nodes share their one-hop latency values (time between the reception of a packet into the queue and its transmission) within the SYNC period. When a receiver node notices that average one-hop latency reaches a high value, which indicates that the traffic from senders is increasing and its duty cycle is not enough to handle the traffic load, it may shorten its sleep time and announces it within the SYNC period. Accordingly, after a sender node receives this announcement, it checks its queue for packets destined to that receiver node. If there is one, the sender shall double its duty cycle so as to align its schedule with that of the receiver provided its battery level is above a specified threshold and its duty cycle has not reached maximum allowable duty cycle. DSMAC achieves tradeoff performance compared to S-MAC, i.e. reduced latency but fair average power consumption per packet as compared to S-MAC.

Adaptive sensor Medium Access Control (AMAC) [108] also bases on S-MAC but realizes adjustable duty cycle in order to reduce energy consumption. AMAC adjusts duty cycle by simply doubling or halving the duty cycle of nodes with the change of traffic load. While achieving reduced energy consumption, AMAC doubles the latency, which makes it not suitable for delay-sensitive applications.

Utilization Based Duty Cycle Tuning MAC (U-MAC) [107] also realizes adaptive duty cycle, but based on the node's utilization to determine its duty cycle. U-MAC adopts a post-transmission selective sleep method to further reduce energy consumption, especially for the low traffic load scenario. U-MAC achieves the better performance on both energy consumption and latency compared to S-MAC and DSMAC.

Optimized MAC [109] similarly bases on the S-MAC and uses adaptive duty cycle according to the traffic load. Optimized MAC determines the traffic load based on the number of messages in the queue. When a node receives a message, it increases the messages counter, and when it sends a message it decreases the counter. If the message counter is greater than a threshold (COUNT_{thres}), then it increases its duty cycle and informs the neighboring sensor of the changed duty cycle using the SYNCrts packet. The neighboring sensor upon the reception of SYNCrts packet checks its message counter. If it is also greater than the COUNT_{thres}, then it also increases its duty cycle. If not, it keeps with the original duty cycle. When the traffic load is low and message counter is less than COUNT_{thres}, the duty cycle is decreased. Optimized MAC slightly improves the energy consumption as compared to S-MAC, but slightly higher than T-MAC. Optimized MAC achieves reduced latency than both S-MAC and T-MAC protocols because of the adaptive duty cycle.

7.4 System Background

To have a fair comparison with AS-MAC and RI-MAC protocols, the network architecture adopted in this protocol is the same as AS-MAC and RI-MAC, i.e. it is a multi-hop sensor network. Assume there are N nodes distributed randomly within a cover area of $A \text{ m}^2$. The nodes can communicate with each other or send data to the sink using their Omni-directional antennas. Each sensor is able to do multicasting, which means a sensor can send data to several nodes concurrently but not necessarily broadcast. The desired receivers are able to receive the desired packets while other receivers will filter them out at the physical layer [78].

Each node has a fixed transmission radius r and a fixed sensing range SR . A sensor receives a message if and only if it is within the transmission radius of the transmission sensor. Each sensor can be in two main modes: sleep mode and active mode. During active mode, sensors can transmit, receive data or stay idle. All sensors have the same

power consumption parameters.

The size of each data packet is fixed to M bits. Packets with larger size will be divided into small size packets and transmitted independently. At each sensor node, there is a queue of packets for the transmission. Forwarding packets have higher priority than packets generated by the node itself. When there is a packet available to be sent, a node will first send a control packet to all of its neighbors. Only the node with the same network address as the address in the control frame will reply. Nodes will then schedule the time for packet transmission and send the packet in a scheduled time slot. The packet reaches the receiver after one round trip propagation delay. The receiver at the destination should have been tuned on to receive the packet. The size of ACK packet is of L bits. It will only be sent once the packet is successfully received. If the sending node does not receive acknowledgement after a certain time, it will retransmit the packet. It will keep retransmitting the packet until the transmission is acknowledged successfully or the retry limit is exceeded.

7.5 The Proposed SEA-MAC Scheme

Our proposed SEA-MAC aims to enhance the AS-MAC scheme. The SEA-MAC is similar to AS-MAC in several aspects of the design. 1) It follows SYNC-AS-SLEEP 3-phase operational cycle for its nodes' operation. 2) It inherits the timeout-based Resilient Active Time (RAT) mechanism for AS-period. 3) It adopts the enhanced proportional mapping function for scheduling data transmission for the Sleep period. 4) It keeps using dual-role SCH frame with tracked retry numbers for efficient multi-hop forwarding. However, the proposed SEA-MAC differentiates itself from AS-MAC in the following aspects which make SEA-MAC superior to AS-MAC and RI-MAC in terms of energy-efficiency, latency and packet delivery ratio.

In WSNs, the traffic load in general mainly depends on two factors: *Event Interval* (EI) and *Sensing Range* of sensor nodes. In order to design a MAC protocol that is adaptive to the varying traffic load, these two factors are necessarily considered separately.

7.5.1 Dynamically Altering Operation Cycle Time

The *Sensing Range* of sensor nodes refers to how far a sensor device can detect signal. The sensing range influences the number of SCH packets generated per event; therefore the larger the sensing range, more SCH packets and traffic will be generated, and more energy will be consumed. Once an event happens, sensor nodes are expected to be in awaked mode for an adequate period of time to transmit or receive the generated packets. The high traffic load requires sensor nodes staying in the active mode long enough so that no excessive delay will be caused due to insufficiency of awaked time. However, the active mode period should not be unnecessary long considering energy constraints. Therefore, a just-enough-period of awaked time should be deduced based on the number of packets generated per event which is almost fixed once the sensing range is determined.

To find out the just-enough-period of active mode based on the number of packets generated per event, the duration to transmit a SCH (RTS) packet to and receive the corresponding SCH (CTS) from the next hop, excluding retransmissions, should be determined first. Let T_{TxRx} denotes the time needed to transmit a SCH (RTS) packet and receive the corresponding SCH (CTS). It is calculated in the following equation (1):

$$T_{TxRx} = 2 \times (T_{SCH}) + T_{SIFS} + T_{DIFS} \quad (7-1)$$

To have a fair comparison with AS-MAC, we use the same network parameters as in AS-MAC which is shown in **Table 7-1**. Thus $T_{SIFS} = 5\text{ms}$, $T_{DIFS} = 10\text{ms}$. $T_{SCH} = \text{Size of SCH} / \text{Bandwidth} = 15\text{bytes} / (20\text{kbps} / 8) = 6\text{ms}$. Thus, $T_{TxRx} = 2 \times 6 + 5 + 10 = 27\text{ms}$.

Since the $T_{AS} = 400\text{ms}$ in original AS-MAC, the maximum number of SCH packets that can be transmitted is about $400 / 27 = 14.8$. In accordance to the simulation, the numbers of packets generated per event for different sensing ranges are tabulated in **Table 7-1**. In view of the number of packets generated per event in **Table 7-1**, for sensing range $> 500\text{m}$, $T_{AS} (= 400\text{ms})$ is not enough to schedule more than 14.8 control packets within an AS-period; therefore the T_{AS} needs to be prolonged proportionally in order to be able to handle the number of packets generated per event.

Let N_{SCH} denotes the number of packets generated per event, and T_{AS} denotes AS-period of time required to complete N_{SCH} SCH packets. Then T_{AS} can be derived from Equation (2):

$$T_{AS} = P (N_{SCH} \times T_{TxRx}) \quad (7-2)$$

where P is the allowable persistent number of contention for each SCH packet. Since the SCH packets generated by each event are from different nodes in most of cases, and likely from non-adjacent nodes, these SCH packets may not be transmitted all in sequentially. Thus when $P > 1$, $P (N_{SCH} \times T_{TxRx})$ would provide redundant AS duration. Thus we use $P=1$ in the following calculation.

Once AS-period is known, the total operation cycle time T_{Cycle} can be determined by equation (3):

$$T_{Cycle} = T_{SYNC} + T_{AS} + T_{SLEEP} \quad (7-3)$$

T_{SYNC} represents the SYNC period, To have a fair comparison with AS-MAC protocol, we remains unchanged ($=55.2ms$). T_{SLEEP} represents the SLEEP period. We have proven in our previous work that T_{AS}/T_{SLEEP} should keep a ratio of 0.136 in order to ensure collision-free Data-ACK packet transmission within two-hop; therefore, $T_{SLEEP} = \frac{T_{AS}}{0.136}$.

$$T_{Cycle} = T_{SYNC} + T_{AS} + \frac{T_{AS}}{0.136} \quad (7-4)$$

According to equation (7-1), (7-2), (7-4) and value of first two rows in Table 7-1, we can calculate T_{AS} and T_{Cycle} under various traffic loads as shown in middle portion of Table 7-1,

Table 7-1: Operational cycle configuration

Sensing range (m)	100	150	200	250	300	350	400	450	500	
No. of packets generated per event	0.8	1.7	3.1	4.6	6.4	8.4	10.6	12.9	15.2	$T_{AS_lb} =$
SYNC period (ms)	55.2	55.2	55.2	55.2	55.2	55.2	55.2	55.2	55.2	304.00
AS period (ms)	25.60	54.40	99.20	147.20	204.80	268.80	339.20	412.80	486.40	
SLEEP period (ms)	188.24	400.00	729.41	1082.35	1505.88	1976.47	2494.12	3035.29	3576.47	
Total Operation Cycle Time (ms)	269.04	509.60	883.81	1284.75	1765.88	2300.47	2888.52	3503.29	4118.07	
(Corrected) AS period (ms)	304.00	304.00	304.00	304.00	304.00	304.00	339.20	412.80	486.40	
(Corrected) SLEEP period (ms)	2235.29	2235.29	2235.29	2235.29	2235.29	2235.29	2494.12	3035.29	3576.47	
Total Operation Cycle Time (ms)	2594.49	2594.49	2594.49	2594.49	2594.49	2594.49	2888.52	3503.29	4118.07	

However, AS-period cannot be too short; otherwise the SYNC period will become dominant portion of total active time within an operation cycle. Therefore, a low boundary of AS-period T_{AS_LB} needs to be pre-determined before the actual sensor network setup. Experimental results showed that the performance of a sensing range between 350m and 400m is a turning point, thus the low boundary is in between 350m and 400m. We take the average of 268.8ms for sensing range 350m and 339.2ms for 400m; the result is divided by T_{SYNC} . Consequently, a factor that reflects the supposed ratio of T_{AS} over T_{SYNC} is calculated as $(268.8+339.2)/2/55.2=5.51$. This ratio indicates that the SYNC period will become a dominant factor of total active time once it exceeds $1/5.51$ or 0.1815 times of AS-period. Thus the low boundary T_{AS_LB} is determined by multiplying this factor by 55.2, which approximately equals to 304ms. Any values of AS-period lower than 304ms in [Table 7-1](#) will be forced to 304ms to prevent the duration of AS-period from becoming too short.

Thus the last three rows in [Table 7-1](#) show the corrected AS-period and the corresponding SLEEP period and operation cycle time that provide just-enough-period for the sensor nodes to transmit scheduling packets SCH for each happened event, and relay the data packets to the destination without excessive delay. The SEA-MAC algorithm will base on traffic load to dynamically alter the T_{AS} , and the corresponding T_{SLEEP} , and T_{Cycle} to mitigate energy consumption and excessive end-to-end delay.

7.5.2 Duty Cycle Efficiency based Duty Cycle Turning

While the just-enough-period provides adequate T_{AS} , T_{SLEEP} , and T_{Cycle} to schedule and transmit data, it is calculated based on number of packets generated per event. When *Event Interval* (EI), which indicates how often an event will occur, becomes shorter or longer, the just-enough-period will become inadequate. If EI is long enough such that there won't be any event in the current operation cycle, indicating a very low traffic load, then the sensor node can hibernate in the succeeding operation cycle to save more energy. While if EI is very short, which creates too much traffic for sensor nodes to accommodate in just-enough-period, then the sensor node should shorten its operation cycle and increase its active time in order to disseminate the busy traffic load quickly. Thus an adequate duty cycle depends on the traffic load which in turn is reflected by duty cycle efficiency.

We know that *duty cycle* can be defined as ratio of total active time over total operation cycle time. Total utilized time is a summary of total transmitting time T_{Tx} and total receiving time T_{Rx} , while total operation cycle time is a summary of utilized time and total idle time T_{Idle} during a complete operation cycle. Thus, the duty cycle can be defined as

$$DC = \frac{T_{Tx} + T_{Rx}}{T_{Tx} + T_{Rx} + T_{Idle}} \quad (7-5)$$

Since the RAT in SEA-MAC is effective active period used in scheduling the data transmission, the RAT in AS-period can be taken as measurement of traffic load. Since we base on proportional mapping function to map transceiver's wake-up time and period, the ratio of RAT over T_{AS} reflects the effective duty cycle in SEA-MAC protocol. Hence we define *duty cycle efficiency* as the ratio of RAT over T_{AS} .

$$E_{DC} = \frac{RAT}{T_{AS}(SR)} \quad (7-6)$$

Where RAT is the Resilient Active Time that can be accumulated for each node; $T_{AS}(SR)$ is a function that returns a T_{AS} period value according to the input sensing range SR based on **Table 7-1**. There are three typical duty cycle efficiencies. First, When $T_0 < RAT < T_{AS}$, equation (7-6) represents an ordinary middle duty cycle efficiency. Second, when $RAT = T_{AS}$, the node is in the saturated loading and the duty cycle efficiency reaches the highest value which is denoted as $E_{DC-H}=1$. Third, when $RAT = T_0$, the duty cycle efficiency is the lowest which is denoted as $E_{DC-L} (<<T_{AS})$. Since the timeout T_0 , which is the longest possible idle time within an AS-period, is a necessary time duration making active time resilient, we include it in the effective active period.

SEA-MAC is designated with the following two key points. (1) During SYNC period, each node's duty cycle efficiency is exchanged among adjacent sensor nodes, so that each node is able to be aware of the duty cycle efficiency of its adjacent nodes at the beginning of operation cycle. (2) Three options of operation cycle will be provided for a sensor node at the end of every operation cycle.

Let E_{DC}^0 denotes the duty cycle efficiency of node itself, E_{DC}^N denotes the duty cycle efficiency of its neighbor nodes, $N=1\sim n$. The first option is when $E_{DC-L} < E_{DC}^0 < E_{DC-H}$, (E_{DC}^0 denotes the duty cycle efficiency of node itself), indicating that the node

itself is in the adequate efficiency level. If all E_{DC}^N (E_{DC}^N stands for duty cycle efficiency of its neighboring nodes, $N=1\sim n$) are less than E_{DC-H} , then the duty cycle of the succeeding operation cycle should be kept the same as the preceding one, as shown in Middle DC Operation scenario in **Figure 7-1**. If any $E_{DC}^N = E_{DC-H}$, then the duty cycle of the succeeding operation cycle for this node should be increased.

The second option is when $E_{DC}^0 = E_{DC-L}$, indicating the duty cycle efficiency of node itself has reached the lowest efficiency. If all $E_{DC}^N = E_{DC-L}$ ($N=1\sim n$), then the duty cycle of the succeeding operation cycle for this node should be reduced in order to comparatively increase the duty cycle efficiency. A hibernation cycle which contains SYNC period, a $T_0/2$ and HIB period is introduced for the second option. A HIB period equals to the sum of AS-period and SLEEP period minus a $T_0/2$, i.e. $T_{HIB} = T_{AS} + T_{SLEEP} - T_0/2$. We leave $T_0/2$ active time in order for node to keep a non-zero RAT and return to middle DC operation when the traffic load recovers. Thus a node sleeps most of time in the hibernation cycle, as shown in Lower DC Operation scenario.

The third option is when $E_{DC}^0 = E_{DC-H}$, indicating the node has reached the highest efficiency and AS-period might not be long enough to handle the traffic load, the duty cycle of the succeeding operation cycle should be increased in order to disseminate potential high traffic load and mitigate unnecessary excessive end-to-end delay, as shown in Higher DC Operation scenario. **Figure 7-1** shows the three options of the operation cycle.

The duty cycle will be increased or decreased by a certain ratio R . Each sensor node contains two counters C_{SCH}^P and C_{SCH}^C which count the total SCH packets numbers of this node for previous and current operation cycle respectively. The counter C_{SCH}^C consists of three parts: the number of residual SCH from the previous cycle, the number of SCH generated in the current cycle, and the number of SCH received from adjacent nodes in the current cycle. The similar components for counter C_{SCH}^P : the number of residual SCH from the previous of previous cycle, the number of SCH generated in the previous cycle, and the number of SCH received from adjacent nodes in the previous cycle. The ratio $R = C_{SCH}^C / C_{SCH}^P$ decides the adjusted amount of duty cycle in next operation cycle, or the pace of adaptation to the traffic change. Since the SEA-MAC is the synchronous duty cycle protocol, we take half or double of previous duty cycle as

duty cycle in the next operation cycle. For example: when $0.5 < R < 1$, next operation duty cycle will be half of current duty cycle; when $1 < R \leq 2$, next operation duty cycle will be twice of current duty cycle; when $2 < R \leq 3$, next operation duty cycle will be thrice of current duty cycle; etc.

The checking and decision is made in SYNC period of every operation cycle, as shown in Figure 7-2. The adjustable operation cycle, thus adaptive duty cycle, enable the sensor nodes to operate more in active time if EI is short and more in inactive time if EI is long.

Figure 7-1: Three different Duty Cycle operations

Figure 7-2: Flow Chart of Duty Cycle Efficiency based Duty Cycle tuning

7.5.3 Implementation Complexity and Cost

It can be seen from the following analysis that the added implementation complexity and cost in SEA-MAC is ignorable compared to the AS-MAC.

The added computation for obtaining the dynamical altering operation cycle time (equation (7-1) ~ (7-4)) and duty cycle efficiency based duty cycle turning (equation (7-5) and (7-6)) in the SEA-MAC design do not change the linearity of the scheme design of AS-MAC; and the calculations of these values are linear and can be easily implemented through simple programming. Thus we may say that the added implementation complexity in SEA-MAC is ignorable compared to the AS-MAC. The cost incurred for computations of these values are very minor. Hence we may ignore the added implementation complexity and extra cost incurred in SEA_MAC while compared with AS-MAC.

7.6 Performance Evaluation

In this section, we compare SEA-MAC with AS-MAC and RI-MAC in the same topology of network and the same traffic model of network. The network topology is a 49-node (7 x 7) grid sensor network where the neighboring nodes are 200 meters apart and the sink node locates in the center of the network. Data packets generated in sensor nodes will be sent to sink node through 1 to 6 hops. The traffic model follows the Random Correlated-Event (RCE) traffic model [24], where there is impulse traffic triggered by spatially correlated random events. RCE picks a random (x,y) location for each event. If every node has a sensing range SR , only nodes that are within the circle centered at (x,y) with radius SR generate packets to report this event. We adjust the sensing range SR to simulate different degrees of traffic load in a network. We employed OMNET++ as the simulation platform.

7.6.1 SEA-MAC Performance Evaluation and Comparison with AS-MAC

In this section, we present the numerical results from the simulation experiments and evaluate and compare the performance of the SEA-MAC with AS-MAC under unicast traffic.

7.6.1.1 Simulation Design

As described early, the simulation model used in SEA-MAC is similar to that used in AS-MAC, a 49-node grid network. To have a fair comparison, SEA-MAC uses the same network parameters as in AS-MAC as shown in **Table 7-2**. Since it has been proven that AS-MAC is superior to DW-MAC [24], S-MAC [12] and RMAC [23], we only need to do comparisons between SEA-MAC and AS-MAC.

Table 7-2: Protocol parameters used in comparison with AS-MAC

Bandwidth	20Kbps	Ch. Encoding Ratio	2
Tx Power	31.2mW	Tx Range	250m
Rx Power	22.2mW	Carrier Sensing Range	550m
Idle Power	22.2mW	Contention Window (CW)	64ms
Sleep Power	3 μ W	Size of ACK	10B
SIFS	5ms	Size of SCH	15B
DIFS	10ms	Size of Data	100B
Retry Limit	5	State Transition Power	31.2mW
T ₀	80ms	State Transition Time	2.47ms

The size of data packet is fixed as 100 bytes in AS-MAC. The shortest path routing function has been used for routing decision as it is available in OMNET++ library [80], which is the simulation software employed in this work. To have a fair comparison with AS-MAC, it is assumed that all nodes are synchronized with each other.

Event Interval (EI) has been used to express interval of the events arrival, whose values can estimate the traffic loads. The larger the EI value is, the lower the traffic load is. And the smaller the EI value is, the higher the traffic load is. And these events will happen at a random location with coordinates being generated following a uniform random distribution.

We have investigated the performance of SEA-MAC protocol in the same range of event intervals, i.e. with EI values of 10s, 9s, 8s, 7s, 6s and 5s. For each EI value, any node with its sensing range covering the location of event happening will collect the information, produce a data packet and send it back to the sink. The sensing range for each node is set from 100m to 500m with interval of 50m as AS-MAC did. By increasing the sensing range, the traffic load increases since more sensors can sense the observed event and generate more data packets. The simulation has been taken for 10 random runs, each of which completes when 500 events triggered.

7.6.1.2 Simulation Results and Discussions

The simulation results are illustrated in this section in terms of average end-to-end delay and energy consumption per successful transmission. All the figures are plotted according to the amount of improvement of SEA-MAC protocol over AS-MAC.

7.6.1.2.1 Average End-to-end Delay

Figure 7-3 (a) and (b) display the average end-to-end delay (s) of different EIs respectively for AS-MAC and SEA-MAC. **Figure 7-3** (c) presents the amount of improvement of SEA-MAC over AS-MAC on average end-to-end delays in seconds (s), which are obtained by subtracting the average end-to-end delays in SEA-MAC from that in AS-MAC. Therefore, a point with a y-value greater than zero indicates that the average end-to-end delay in SEA-MAC is less than the respective values in AS-MAC, at the traffic condition specified by this point. Vice versa, a point with a y-value less than zero indicates that the average end-to-end delay in SEA-MAC is larger than the respective values in AS-MAC. **Figure 7-3** (d) presents the amount of improvement of SEA-MAC over AS-MAC on end-to-end delays in percentage (%).

Figure 7-3: Outperformance of SEA-MAC over AS-MAC on average end-to-end delay

To facilitate analysis and discussion, we partition the traffic load into 6 zones: L1, L2, M1, M2, H1 and H2 stands for traffic load respectively with [sensing range, EI] value of [(100,150,200), (10, 9, 8)], [(100,150,200), (7, 6, 5)], [(250,300,350), (10, 9, 8)], [(250,300,350), (7, 6, 5)], [(400,450,500), (10, 9, 8)], [(400,450,500), (7, 6, 5)], as indicated in **Table 7-3** (a) and (b). It can be seen from **Figure 7-3** and **Table 7-3** that when sensing range $\leq 200\text{m}$, the SEA-MAC's end-to-end delay is not as good as AS-MAC. When sensing range $> 200\text{m}$, SEA-MAC's end-to-end delay is superior to AS-MAC, and the amount of improvements becomes larger and larger as the event arrival rate increased or EI decreased. This trend is obtained since the AS-period is dynamically adjusted based on the number of packets generated per event. As the event arrival rate increased in AS-MAC, more packets will be delayed and transmitted in the next operation cycle due to insufficient active time of the current cycle. However, in SEA-MAC, the active time will be adjusted according to the current traffic conditions. In this way, fewer packets will need to be transmitted in the next operation cycle, compared with AS-MAC. More specifically, when traffic load is in low-end zones L1 and L2, SEA-MAC's average of average end-to-end delay increases 1.99s and 1.16s respectively. When traffic load is in middle-range zones M1 and M2, SEA-MAC's average of average end-to-end delay reduces 0.86s and 2.60s respectively. When traffic load is in high-end zones H1 and H2, SEA-MAC's average of average end-to-end delay reduces further by 2.31s and 11.99s respectively. Similarly the averages of max end-to-end delay for different traffic load zones are shown in yellow in **Table 7-3** (a). **Table 7-3** (b) shows the end-to-end delay performance difference in term of percentage.

Table 7-3: Average Performance Difference between End-to-End Delay of SEA-MAC and AS-MAC for Different Traffic Loads

(a) End-to-end delay performance difference in second(s)									
Sensing Range (m)	100	150	200	250	300	350	400	450	500
EI=10 (s) (Ave delay)	Traffic load zone L1			Traffic load zone M1			Traffic load zone H1		
EI=10 (s) (Max delay)	Ave of Ave Delay (s) of L1			Ave of Ave Delay (s) of M1			Ave of Ave Delay (s) of H1		
EI=9 (s) (Ave delay)		-1.99 (s)			0.86 (s)			2.31 (s)	
EI=9 (s) (Max delay)	Ave of Max Delay (s) of L1			Ave of Max Delay (s) of M1			Ave of Max Delay (s) of H1		
EI=8 (s) (Ave delay)		-4.96 (s)			8.29 (s)			26.36 (s)	
EI=8 (s) (Max delay)									
EI=7 (s) (Ave delay)	Traffic load zone L2			Traffic load zone M2			Traffic load zone H2		
EI=7 (s) (Max delay)	Ave of Ave Delay (s) of L2			Ave of Ave Delay (s) of M2			Ave of Ave Delay (s) of H2		
EI=6 (s) (Ave delay)		-1.16 (s)			2.60 (s)			11.99 (s)	
EI=6 (s) (Max delay)	Ave of Max Delay (s) of L2			Ave of Max Delay (s) of M2			Ave of Max Delay (s) of H2		
EI=5 (s) (Ave delay)		-2.62 (s)			22.84 (s)			519.58 (s)	
EI=5 (s) (Max delay)									

(b) End-to-end delay performance difference in percentage (%)

Sensing Range (m)	100	150	200	250	300	350	400	450	500
El=10 (s) (Ave delay)	Traffic load zone L1			Traffic load zone M1			Traffic load zone H1		
El=10 (s) (Max delay)	Ave of Ave Delay (%) of L1			Ave of Ave Delay (%) of M1			Ave of Ave Delay (%) of H1		
El=9 (s) (Ave delay)	-59.69 (%)			14.01 (%)			27.22 (%)		
El=9 (s) (Max delay)	Ave of Max Delay (%) of L1			Ave of Max Delay (%) of M1			Ave of Max Delay (%) of H1		
El=8 (s) (Ave delay)	-64.86 (%)			30.50 (%)			42.14 (%)		
El=8 (s) (Max delay)									
El=7 (s) (Ave delay)	Traffic load zone L2			Traffic load zone M2			Traffic load zone H2		
El=7 (s) (Max delay)	Ave of Ave Delay (%) of L2			Ave of Ave Delay (%) of M2			Ave of Ave Delay (%) of H2		
El=6 (s) (Ave delay)	-39.57 (%)			34.92 (%)			50.90 (%)		
El=6 (s) (Max delay)	Ave of Max Delay (%) of L2			Ave of Max Delay (%) of M2			Ave of Max Delay (%) of H2		
El=5 (s) (Ave delay)	-68.44 (%)			44.70 (%)			68.20 (%)		
El=5 (s) (Max delay)									

The results show that when traffic load is in low-end zones, SEA-MAC presents a longer end-to-end delay than AS-MAC. The reason for it is that nodes might enter hibernation cycle due to the low traffic load, and during the hibernation period, the nodes sense some events and attempt to send SCH packets. But they have to wait for the next operation cycle to react to the newly sensed events, which results in the increased end-to-end delay. When traffic load is in mid-range and high-end zones, SEA-MAC superior to AS-MAC in end-to-end delay with the largest improvement of 50.90% in average of average end-to-end delay and 68.20% in average of max end-to-end delay. This is because the nodes might operate in double duty-cycle when traffic load is in mid to high-end zones, and increased duty-cycle contributes to the reduced end-to-end delay.

7.6.1.2.2 Average Power Consumption

Figure 7-4 (a) and (b) show the average energy consumption per successful transmission of AS-MAC and SEA-MAC respectively. While they present stable trend, the trending levels are different in different sensing ranges. **Figure 7-4** (c) shows the percentage amount of outperformance of SEA-MAC over AS-MAC on energy conservation per successful transmission, i.e. the difference of these two trends, which probably present the fluctuation. In similar way, these values are obtained by subtracting the energy consumption by SEA-MAC from that by AS-MAC. Therefore, a point with a y-value greater than zero indicates that the energy consumption by SEA-MAC is less than the respective values in AS-MAC, at the traffic condition specified by this point. Vice verse, a point with a y-value less than zero indicates that the energy consumption by SEA-MAC is larger than the respective values by AS-MAC.

(a) AS-MAC ave. energy consumption per successful transmission

(b) SEA-MAC ave. energy consumption per successful transmission

(c) Outperformance SEA-MAC over AS-MAC on energy consumption per successful transmission (%)

Figure 7-4: Performance comparison between SEA-MAC and AS-MAC on average power consumption per successful transmission

For some of cases in traffic load zone L1 and L2, SEA-MAC consumes more energy than AS-MAC. This is because SEA-MAC adopts the shorter AS-period, SLEEP-period and total operation cycle time as indicated [Table 7-1](#) in order to gain the reduced end-to-end delay. But the SYNC-period is the same in both protocols; therefore the ratio of $T_{\text{SYNC}} + T_{\text{AS}} / T_{\text{Cycle}}$ which reflects the active time of the nodes in SEA-MAC is greater than that in AS-MAC. Since the performance of energy consumption is based on per successful transmission, energy consumption of SEA-MAC is comparatively higher than AS-MAC for the low traffic zone where there are fewer packets. For the mid-range and high-end traffic load zones M1, M2, H1 and H2, the number of packets increases; furthermore the ratio of $T_{\text{SYNC}} + T_{\text{AS}} / T_{\text{Cycle}}$ decreases due to the longer T_{Cycle} and number of packets increases in zones H1 and H2, so energy consumption of SEA-MAC is comparatively lower than AS-MAC for the high traffic zone. More specifically, for sensing range 250m~500m, the energy saving range from 1.88% (for sensing range 250m and EI=10s) to 16.28% (for sensing range 250m and EI=6s). Average of average

energy saving for M1, M2, H1 and H2 zone are 10.08, 11.03, 10.48 and 6.61 respectively. So the SEA-MAC saves more energy in traffic load zone M2.

7.6.1.2.3 Packet Delivery Ratio (PDR)

Figure 7-5 (a) and (b) show the PDR (%) for different EIs of AS-MAC and SEA-MAC respectively. **Figure 7-5** (c) shows the outperformance of SEA-MAC over AS-MAC on PDR for different EIs. It can be seen that SEA-MAC achieve a great improvement on PDR for different EIs compared with AS-MAC; and the higher the traffic load is (with increased sensing range and shorter EI), the larger amount of PDR SEA-MAC superior to AS-MAC becomes. The greatest improvement 16% increasing in PDR occurred for the highest traffic load scenario with sensing range of 500m and EI of 5s. This improvement is expected to achieve as the average energy consumption per successful transmission has been enhanced by SEA-MAC, as mentioned in earlier Section, and consequently the total number of successful transmission must have been increased.

(a) AS-MAC PDR (%) for different EIs

(b) SEA-MAC PDR (%) for different EIs

(c) Outperformance SEA-MAC over AS-MAC on PDR for different EIs

Figure 7-5: Performance comparison between SEA-MAC and AS-MAC on PDR for different EIs

7.6.2 SEA-MAC Performance Evaluation and Comparison with RI-MAC

In this section, we present the performance comparison between SEA-MAC and RI-MAC [112] based on simulation experiments results of the protocols for the 49-node (7×7) grid network under unicast traffic.

7.6.2.1 Simulation Design

To have a fair comparison, we adopt the same parameters as RI-MAC in simulation experiments. **Table 7-4** summarizes the key protocol parameters that the two MAC protocols used in the simulation. It consists of two portions: the parameters common to the two protocols and the parameters that are protocol-specific.

Table 7-4: Protocol parameters used in comparison with RI-MAC

Protocol Common Parameters:			
Bandwidth	250 Kbps	Size of Hardware Preamble	6 B
SIFS	192 μ s	TX Range	250 m
Slot time	320 μ s	Carrier Sensing Range	550 m
Retry limit	5		
Protocol Specific Parameters:			
	SEA-MAC	RI-MAC	
DIFS	960 μ s	N/A	
Special Frame/Size	SCH/15 B	Beacon/6-9B	
Size of ACK	10 B	5 B	
Backoff Window	0-63	0-255	
Dwell Time	N/A	Variable	
Sync Period (T_{sync})	0.0552s	N/A	
AS Period (T_{as})	0.2s	N/A	
Sleep Period (T_{sleep})	2.4s	N/A	
Timeout (T_o)	0.022s	N/A	

We set DIFS as 960 μ s which is 5 times as much as the value of SIFS to follow the DCF scheme. SCH size is set as 15 bytes which is 9-6 bytes more than RI-MAC's Beacon. Out of 15 bytes, 4 bytes store sender address and receiver address, 1 byte store CTS_Retry_No and RTS_Retry_No. We use 32 as the initial backoff window and 64 as maximum backoff window.

In addition, we adjusted the values of some parameters in SEA-MAC in order to achieve competitive performance. Study shows that nodes synchronization may cause a lot of schedule overhead, especially for the low traffic load or no traffic load, synchronization period results in more than half of the total duty cycle. But we found

that nodes do not necessarily synchronize with each other in every operation cycle. According to the study in PW-MAC [113], clock only drifts 5ms in every 500 seconds time. In our simulation one operation cycle time without sync is set as 2.6 seconds. Thus clock drifts $(2.6 \times 5) / 500 = 0.026 \text{ ms} = 26 \text{ } \mu\text{s}$ within one operation cycle. It would be a waste to synchronize the clock in every operation cycle. In our implementation, we set the nodes to synchronize with each other every 6 operation cycles. Clock drifts in this time interval should be $(2.6 \times 6 + 0.0552) \times 5 / 500 = 0.156552 \text{ ms} = 156.552 \text{ } \mu\text{s}$ which is half of slot time. Since each slot time is 320 μs , we set the receiver nodes wake up 1 slot time earlier than the transmitter nodes in the Sleep period to ensure that one synchronization every 6 operation cycles does not affect protocol's performance. We take assumption in the simulation that we do not consider hardware latency.

We have conducted the performance testing of SEA-MAC protocol not only for the traffic scenario that RI-MAC presented (EI=60s, sensing range from 100m to 500m for 100 events), but also for other low traffic load and high traffic load scenarios, i.e. with EI values of 200s, 100s, 30s, 20s, 10s, 9s, 8s, 7s, 6s and 5s and sensing range from 100m to 500m with 50m in each step for 100 events in order to present a wider spectrum of performance comparison. For each EI value, any node with its sensing range covering the location of event happening will collect the information, produce a data packet and send it back to the sink. By increasing the sensing range, the traffic load increases since more sensors can sense the observed event and generate more data packets. The simulation has been taken for 10 random runs, each of which ends once 100 events triggered.

7.6.2.2 Simulation Results and Discussions

The simulation results are illustrated in this section in terms of average and maximum end-to-end delay and duty cycle. All the figures are plotted according to the amount of improvement of SEA-MAC protocol over RI-MAC.

7.6.2.2.1 End-to-end Delay

Figure 7-6 shows the average and maximum end-to-end delay of RI-MAC and SEA-MAC with different EIs covering low, medium and high traffic load scenarios and different sensing ranges from 100m to 500m. The diagrams show the performance only for the selected EIs.

Figure 7-6: Average and maximum end-to-end delay of RI-MAC and SEA-MAC with different EIs versus sensing ranges

(a) Average end-to-end delay difference (s)

(b) Maximum end-to-end delay difference (s)

Figure 7-7: End-to-end delay difference between RI-MAC and SEA-MAC versus EIs and sensing ranges

Figure 7-7 (a) presents the amount of improvement of SEA-MAC over RI-MAC on average end-to-end delays in seconds (s) for various traffic load scenarios, which are obtained by subtracting the average end-to-end delays of SEA-MAC from that of RI-MAC for different EIs (i.e. $(E2E\ Delay_{RI-MAC} - E2E\ Delay_{SEA-MAC})$ represented by RI-

SEA). Therefore, a point with a y-value greater than zero indicates that the average end-to-end delay of RI-MAC is greater than the respective values in SEA-MAC, at the traffic condition specified by this point. Vice versa, a point with a y-value lesser than zero indicates that the average end-to-end delay in RI-MAC is lesser than the respective values in SEA-MAC. **Figure 7-7 (b)** presents the amount of improvement of SEA-MAC over RI-MAC on maximum end-to-end delays in seconds (s) for various traffic load scenarios, which are obtained in the similar way as in **Figure 7-7 (a)**.

It can be seen that average end-to-end delay of SEA-MAC is smaller than that of RI-MAC in all traffic conditions in question. SEA-MAC average end-to-end delay smoothly increases from half operation cycle time to one operation cycle times with the increasing traffic load. That indicates, in average, data packet can be transferred from source to sink within one operation cycle. RI-MAC shows more explicit climb in average end-to-end delay as traffic load increase. Maximum end-to-end delay difference show the same trend as average end-to-end delay does, and the difference presents more drastic climb. This is because contention and collision become more and more serious with the increasing traffic load for the contention-based protocol and become the significant source of delay.

Figure 7-8 (a) presents the amount of improvement of SEA-MAC over RI-MAC on average end-to-end delays in percentage (%) for various traffic load scenarios, which are obtained according to the equation $\frac{E2E\ Delay_{RI-MAC} - E2E\ Delay_{SEA-MAC}}{E2E\ Delay_{RI-MAC}} \%$ for different EIs (represented by RI-SEA). The improvement on average end-to-end delays ranges from 1% up to 80%. **Figure 7-8 (b)** presents the amount of improvement of SEA-MAC over RI-MAC on maximum end-to-end delays in percentage (%) for various traffic load scenarios, which are obtained in the similar way as **Figure 7-8 (a)**. The improvement on maximum end-to-end delays ranges from 30% up to 90%.

- (a) Average end-to-end delay improvement (%) (b) Maximum end-to-end delay improvement (%)

Figure 7-8: End-to-end delay improvement SEA-MAC over RI-MAC for EIs and sensing ranges

7.6.2.2.2 Average Duty Cycle

Figure 7-9 shows average duty cycle of RI-MAC and SEA-MAC for different EIs and sensing range from 100 m to 500 m. The diagrams show the performance only for the selected EIs.

Figure 7-9: Average Duty Cycle of RI-MAC and SEA-MAC versus sensing ranges and EIs

Figure 7-10 shows average duty cycle difference between RI-MAC and SEA-MAC for all the traffic loads in question, which are obtained by subtracting the average duty cycle of SEA-MAC from that of RI-MAC for different EIs (i.e. $(\text{DutyCycle}_{\text{RI-MAC}} - \text{DutyCycle}_{\text{SEA-MAC}})$ represented by RI-SEA). Therefore, a point with a y-value greater than zero indicates that the average duty cycle of RI-MAC is greater than the respective

values in SEA-MAC, at the traffic condition specified by this point. Vice versa, a point with a y-value lesser than zero indicates that the average duty cycle in RI-MAC is less than the respective values in SEA-MAC.

Figure 7-10: Average Duty Cycle Difference between RI-MAC and SEA-MAC (RI-SEA) versus EIs and sensing ranges

It can be observed from **Figure 7-9** and **Figure 7-10** that RI-MAC requires lower duty cycle than SEA-MAC under light traffic load, but SEA-MAC outperforms RI-MAC in average duty cycle under medium and heavy traffic load. This is because (1) RI-MAC does not need to synchronize clock of each node, which eliminates a lot of schedule overhead incurred in SEA-MAC. (2) when there is no traffic, all the nodes only wake up for a short instance and transmit a base beacon. Hence the overall duty cycle of nodes under no traffic load can be as small as 0.075% according to the RI-MAC simulation; while the duty cycle of SEA-MAC under no traffic load is about $(6 \cdot T_o + T_{sync}) / (6 \cdot T_{sleep} + 6 \cdot T_a + T_{sync}) = 1.196\%$ which is about 16 times larger than that of RI-MAC under light traffic load.

However, when the traffic load increases, the duty cycle of SEA-MAC appear to be much more stable than that of RI-MAC. Especially when EI reaches 5 s, the average duty cycle of SEA-MAC is lower than that of RI-MAC for almost all the sensing ranges. There are a few main reasons for it: (1) RI-MC is an asynchronous contention-based duty cycle protocol. The contention and collision become more and more serious with the increasing traffic load for the asynchronous contention-based protocol. (2) The overhearing incurred significantly increases the duty cycle of RI-MAC as well. In RI-MAC, the sender remains active and waits silently until the receiver explicitly signifies

when to start data transmission by sending a short beacon frame. When a receiver node wake up and send a base beacon to request data, the probability that more than one node response to the request and transmit packet, which will cause collision, will increases with the increasing traffic load. For example, when the sensing range becomes 400 m, the average number of packets generated from each event is 10.6. The larger this number is, the higher probability of collision incur. Once collision happen, resending the beacon frames and data packets without any backoff results in the increasing duty cycle, more retransmission and overhearing and boost up overall duty cycle.

However, in SEA-MAC, when a packet is first generated, the node will first choose a backoff from 0-31 before transmitting. This design prevents the packets from collision with each other at the first time. Secondly, SEA-MAC uses Dual-Role SCH frame with Tracked Retry Numbers for multi-hop forwarding and optimizes high traffic relay for multi-hop transmission.

For the Packet Delivery Ratio (PDR), since the PDR of SEA-MAC and RI-MAC under different scenarios are all 100%, we leave out the PDR comparison.

7.7 Summary

In this chapter, a Self-Adaptive MAC protocol (SEA-MAC) has been proposed to enhance AS-MAC protocol and address inefficient duty cycle issue in fixed duty cycle scheme. SEA-MAC inherits basic scheme of AS-MAC which enables SEA-MAC to adaptively prolong the RAT length so as to allow nodes to schedule more data transmission within one operational cycle when the traffic load is increasing, and shorten the RAT length and go to sleep early when the traffic load is decreasing. Differentially from AS-MAC, SEA-MAC further adjusts the duty cycle dynamically based on the traffic load, so as to mitigate inefficient duty cycle problem in fixed duty cycle scheme. SEA-MAC realizes the adaptive duty cycle in two aspects: first in micro level adjusting active time based on Event Interval (EI); second in macro level adjusting duration of operation cycle time according to sensing range.

We have compared the performance of SEA-MAC to AS-MAC and RI-MAC by simulation experiments. And the results have shown that the SEA-MAC outperforms AS-MAC in energy efficiency, average and max end-to-end delay as well as packet

delivery ratio under the mid-range and high-end unicast traffic scenarios. Under the mid-range and high-end traffic, SEA-MAC superiors to AS-MAC with the largest improvement of 50.90% in average of average end-to-end delay and 68.20% in average of max end-to-end delay, lowers energy consumption by 1.88% ~ 16.28% compared to AS-MAC, and improves PDR further 0%~16% over the AS-MAC. Under low-end traffic zones, the results have shown that the SEA-MAC performs not as well as AS-MAC in end-to-end delay and energy saving, especially for sensing range 100m and 200m. When compared to RI-MAC, the results show that end-to-end delay of SEA-MAC superiors to RI-MAC by 1% to 80% for average end-to-end delay and 30% to 90% for maximum end-to-end delay. For average duty cycle, RI-MAC requires lower duty cycle than SEA-MAC under light traffic load, but SEA-MAC outperforms RI-MAC in average duty cycle under medium and heavy traffic load. These results imply that SEA-MAC presents better adaptability to the high traffic load, which leaves the room to further improve the adaptability of SEA-MAC for full spectrum of traffic loads in the future.

8 Characterization and Rendering of Application-Specific WSNs

We developed an AS-MAC protocol which adapt to the changing traffic load, but in micro-level. That is it only can adjust its Resilient Active Time (RAT) within the range of Adaptive Scheduling period (AS-period) and operation cycle time. In order to increase the adaptability to the dynamic traffic load, we developed a Self-Adaptive MAC (SEA-MAC) to further adjust its RAT in macro-level, i.e. in duty cycle level. Experimental results have shown that different AS-period T_{AS} in AS-MAC, hence different ratio value of T_{AS}/T_{SLEEP} , result in different network performance. In view of this fact plus the inspiration by the need for a generic MAC protocol which does not assume any prior knowledge of application requirements but can tune its duty cycle automatically according to traffic conditions, we believe that the traffic pattern can be characterized and application-specific requirements can be modelled and rendered in the generic form, so that the WSN built with this model has good performance and adaptability to the different applications.

8.1 Traffic-aware Characterization and Application-aware Rendering

To realize the above benefits, we characterize and model the adaptive protocol from two aspects: traffic-aware characterization and application-aware rendering. A high level traffic-aware characterization and application-specific requirements rendering can be depicted in the following **Figure 8-1**:

Figure 8-1: Traffic-aware characterization and application-aware rendering

If we use P_{Net} denote the performance of the network protocol, then

$$P_{\text{Net}} = f_1 (\text{App-specific Traffic Pattern}) \cdot S_s \cdot f_2 (\text{App-specific Requirements}) \quad (8-1)$$

Where f_1 is traffic-aware characterization, f_2 is application-aware rendering, and S_s refers to the sensors' setting. The application-specific traffic pattern and some parameters' setting of sensor nodes are taken as input to traffic-aware characterization. The application-specific requirements are taken as input to application-aware rendering. These inputs are fed to the protocol and rendered as a set of parameters of the protocol, such as $P_k^1, P_k^2, \dots, P_k^M$. M is the number of the parameters of the protocol. This set of parameters is the result of traffic-aware characterization and application-aware rendering. P_k^i ($i=1 \sim M, k=1 \sim T$) are parameters mapped to that particular traffic pattern and application requirements and will be used to drive the execution of the protocol. If there are some special needs for WSNs to take, the blue dotted line triangles in **Figure 8-1** present alternative sets of parameters that are expected to be set (thus as input as well) into the model before or during protocol execution. Different aspects of performance metric $PM_1, PM_2, \dots, PM_n$ are the output of the protocol.

8.1.1 Traffic-aware Characterization

The traffic-aware characterization is for the protocol to sense the traffics of the application, so that the dynamic changes of traffics can be fed to the protocol so as for protocol to adjust its parameters and adapt to the changed traffics accordingly. The factors that most significantly affect traffic are event interval (λ), size of data (S_d) and sensing range radius (ρ). Event interval and size of data are protocol-independent factors. MAC protocols are able to keep track of traffic characteristics like event interval and size of data with the sensor network and share the traffic information among sensor nodes. Protocol algorithm is able to detect the event interval λ through counters of how many events occur per unit period. Sensing range is a re-configurable factor and can be pre-set before running the protocol. And protocol algorithm should be informed of the sensors' various setting, including sensing range radius, etc. These factors also affect the performance of MAC protocols. The other factors affecting performance of the protocols are protocol-specific parameters. For the AS-MAC protocol, for example, the $T_{\text{AS}}/T_{\text{SLEEP}}$ ratio and timeout value T_0 are the most significant factors. For the other protocols, the influential factors might be different. Thus, the

traffic-aware characterization will be carried out in the two stages: (1) Protocol-Independent Factors Sensing and Setting; (2) Traffic-Aware Rendering. The following equation expresses the Protocol-Independent Factors Sensing and Setting.

$$P_{\text{Net}} = f_1(\lambda, S_d, \rho) \cdot S_s \cdot f_2(\dots) \quad (8-2)$$

Where: f_1 : Performance function regarding to the influential factors λ , S_d , and ρ .

λ : Event interval.

S_d : Size of the data.

ρ : Sensor range radius.

Traffic-Aware Rendering is discussed together with application-aware rendering below.

8.1.2 Application-aware Rendering

The application-aware rendering is for protocols to take application-specific requirements, such as energy-efficiency, latency, packet delivery ratio (PDR), fairness, etc., into the account, so that the protocols can adapt its behavior to the specific needs of the application. These performance metrics carrying the application-specific requirements can be assigned with different weightage in order to reflect the partiality of the needs to the facet of performance metric. A performance metric with the highest weightage should be considered in the first priority. Similarly, a performance metric with the lowest weightage should be considered in the last priority. The f_2 portion in equation (8-3) presents the application-aware rendering in generic form:

$$P_{\text{Net}} = f_1(\dots) \cdot S_s \cdot f_2\left(\prod_{j=1}^N \frac{\sum_{i=1}^M W_i \cdot P_i^j}{M}\right) \quad (8-3)$$

Where: P_i^j : j^{th} parameter relevant to the protocol for i^{th} performance metrics/factor.

N : the number of parameters of the protocol to be considered.

M : the number of performance metrics/factors to be considered.

W_i : the weightage of an application-specific performance metrics/factor to be considered. It may refer to energy efficiency, latency, packet delivery ratio, fairness, etc. The weightage should be between 0 and 1. Adjusting W_i can reflect the application-

specific requirements in terms of which aspect of network performance metrics is more important to the application. Setting W_i to zero indicates that an application-specific performance metric is not to be taken into account.

As for the AS-MAC protocol, P_i^1 is ratio of T_{AS}/T_{SLEEP} named after R_i and P_i^2 is timeout value T_i . Thus we have

$$P_{Net} = f_1 (\dots) \cdot S_s \cdot f_2 \left(\frac{\sum_{i=1}^M W_i \cdot R_i}{M}, \frac{\sum_{i=1}^M W_i \cdot T_i}{M} \right)$$

R_i is the T_{AS}/T_{SLEEP} ratio obtained for i^{th} factor through experiments. R_i and T_i in turn depend on λ , ρ and application-specific factors. Thus $R_i(\lambda, \rho)$ and $T_i(\lambda, \rho)$ express a traffic-aware rendering, and the following equation expresses application-aware rendering:

$$P_{Net} = f_1 (\dots) \cdot S_s \cdot f_2 \left(\frac{\sum_{i=1}^M W_i \cdot R_i(\lambda, \rho)}{M}, \frac{\sum_{i=1}^M W_i \cdot T_i(\lambda, \rho)}{M} \right) \quad (8-4)$$

If we consider three factors: Latency, Package Delivery Ratio and Energy Consumption, then $M=3$ and $R_1(\lambda, \rho)$, $R_2(\lambda, \rho)$ and $R_3(\lambda, \rho)$ are the T_{AS}/T_{SLEEP} ratio of these three factors.

If we consider different timeout values for different hop node, then we have:

$$P_{Net} = f_1 (\dots) \cdot S_s \cdot f_2 \left(\frac{\sum_{i=1}^M W_i \cdot R_i(\lambda, \rho)}{M}, \frac{\sum_{i=1}^M W_i \cdot T_i^n(\lambda, \rho)}{M} \right) \quad (8-5)$$

Where $T_i^n(\lambda, \rho) = T_i^0(\lambda, \rho) + (n-1) \Delta T$. ($n \geq 1$).

T_i^n is timeout value of the n^{th} -hop node. T_i^0 is the timeout of the 1^{st} -hop node which depends on λ and ρ . ΔT denotes the average prolonged time for each hop which can be determined by the transmission time of a SCH package.

8.2 Rendering Realization of Traffic-Awareness and Requirement-Awareness

8.2.1 Rendering Realization of Application Traffic-Awareness

MAC protocols are able to keep track of traffic characteristics like event interval and size of data with the sensor network and share the traffic information among sensor

nodes. Such rendering of application traffic-awareness can be realized through the following parameters.

- T_d : A counter to record the traffic density in term of how many events occur per unit period. Event interval $\lambda=1/ T_d$.
- S_d : Size of the data. We adopted the Data Size field in IEEE802.11 frame in our protocol design, thus may easily obtain the data size value of the application traffic.
- ρ : Sensing range is a re-configurable factor and can be pre-set before running the protocol. The protocol algorithm should be informed of the sensors' various setting, including sensing range radius, etc.

The characteristics of application traffics can be reflected by these parameters and fed as inputs to the protocol for further application requirement-aware rendering.

8.2.2 Rendering Realization of Application Requirement-Awareness

We use three performance metrics – Energy Consumption, Package Delivery Ratio (PDR) and End-to-End Delay (Latency) in AS-MAC protocol, to elaborate how to realize the rendering of application requirement-awareness. From equation (8-3), we have $M=3$ for the three performance metrics.

For AS-MAC protocol, there are two critical parameters:

- R_i : ratio of T_{AS}/T_{SLEEP} for the i^{th} factor. In AS-MAC protocol, the ratio R_i depends on the event interval λ and sensing range ρ . These two parameters have been obtained through rendering of application traffic-awareness described above. If we use W_1 , W_2 and W_3 to denote the weights of the three performance metrics, thus reflecting the application-specific requirements to the protocol, then we have average T_{AS}/T_{SLEEP} ratio $\bar{R}$:

$$\bar{R} = \frac{W_1 \cdot R_1(\lambda, \rho) + W_2 \cdot R_2(\lambda, \rho) + W_3 \cdot R_3(\lambda, \rho)}{3} \quad (8-6)$$

- T_i^n : is timeout value of the n^{th} -hop node. $T_i^n(\lambda, R) = T_i^0(\lambda, R) + (n-1) \Delta T$.

Similarly, we have average timeout value based on the three performance metrics weighted with W_1 , W_2 and W_3 :

$$\bar{T}^n = \frac{W_1 \cdot T_1^n(\lambda, \rho) + W_2 \cdot T_2^n(\lambda, \rho) + W_3 \cdot T_3^n(\lambda, \rho)}{3} \quad (8-7)$$

Equations (8-6) and (8-7) represent the rendering of application traffic-awareness and application requirement-awareness respectively through parameterization of the traffic loading (λ , ρ) and weightages of performance metrics (W_1 , W_2 , W_3) in AS-MAC. In order to realize the rendering in the protocol program, we tabulate the traffic loads with the following generic table **Table 8-1** which partitions the traffic loads into nine (9) traffic load zones and represents them with a pair of (λ , ρ). Thus (λ_i , ρ_j) ($i, j=1, 2, 3$) represent different traffic loads. The traffic load zones partitioned by the gross dotted lines more accurately present the different traffic loads; and thinner dotted lines partition the traffic load zones in the finer granularity.

Table 8-1: Traffic load zones with varying Event Interval and sensing range

Sensing Range ρ (m) Event Interval λ (s)	100 150 200	250 300 350	400 450 500
El=100 (s) El=80 (s) El=60 (s) El=40 (s) El=20 (s)	(λ_1, ρ_1)	(λ_1, ρ_2)	(λ_1, ρ_3)
El=10 (s) El=9 (s) El=8 (s) El=7 (s) El=6 (s)	(λ_2, ρ_1)	(λ_2, ρ_2)	(λ_2, ρ_3)
El=5 (s) El=4 (s) El=3 (s) El=2 (s) El=1 (s)	(λ_3, ρ_1)	(λ_3, ρ_2)	(λ_3, ρ_3)

The following matrix table **Table 8-2** clearly and concisely present the realization of the rendering the traffic-awareness and requirement-awareness for the above different traffic loads in the AS-MAC protocol implementation. For illustration purpose, we categorize

(λ_1, ρ_1) , (λ_2, ρ_1) , (λ_1, ρ_2) as Low Traffic Load (LTL)

(λ_3, ρ_1) , (λ_2, ρ_2) , (λ_1, ρ_3) as Medium Traffic Load (MTC)

(λ_3, ρ_2) , (λ_2, ρ_3) , (λ_3, ρ_3) as High Traffic Load (HTL)

In the real algorithm implementation, we adopted the finer granularity traffic load zones as depicted by the thinner dotted lines.

Table 8-2: Realization of the rendering application specific requirements for the different traffic loads in the AS-MAC protocol

Application Traffic	Application Requirements	
	Prioritized Sequence [Prioritized Number (Weightage)]	
	1(W ₁), 2(W ₂), 3(W ₃)	1(W ₃), 2(W ₂), 3(W ₁)
LTL (λ_1, ρ_1) (λ_2, ρ_1) (λ_1, ρ_2)	Long T _{as} , Short T ₀	Short T _{as} , Long T ₀
MTL (λ_3, ρ_1) (λ_2, ρ_2) (λ_1, ρ_3)	Long T _{as} , Short T ₀	Short or Medium T _{as} , Long T ₀
HTL (λ_3, ρ_2) (λ_2, ρ_3) (λ_3, ρ_3)	Short or Long T _{as} , Short T ₀	Medium T _{as} , Long T ₀

Where: W₁, W₂ and W₃ denote the weights of the three performance metrics respectively.

An example of reading one of traffic load zones is described as follows. 1(W₁), 2(W₂), 3(W₃) indicates the prioritized sequence: energy consumption first, PDR second and end-to-end delay third. Thus, when traffic load is in the HTL zone and application requirements are represented by 1(W₁), 2(W₂), 3(W₃), then a Short or Long T_{as} and a Short T₀ should be adopted by sensor nodes. Similarly, when traffic load is in the MTL zone and application requirements are represented by 1(W₃), 2(W₂), 3(W₁), then a Short or Medium T_{as} and a Long T₀ should be adopted by sensor nodes.

With the above realization of rendering application specific requirements for the different traffic loads, the AS-MAC protocol is made adaptable to the dynamic traffic loads. i.e. when traffic load changes, the protocol may adapt its behaviors to suit to the changed traffic load and delivery a better performance. The application specific requirements can also be adjusted by feeding a set of new requirements through remote actuation, as depicted by the dotted triangles as inputs to the box “Traffic-aware and App-aware rendering”.

8.3 Deployment of AS-MAC over TinyOS/nesc

Deployment of the developed MAC protocols onto a WSN Kit serves two purposes: 1) to test the applicability of the protocols in the real application; 2) to achieve the application-aware rendering as discussed in the previous two sections. Such an attempt was made on the AS-MAC protocol using Crossbow WSN Kit.

8.3.1 Migration of AS-MAC from OMNet++ to TinyOS/nesc

The AS-MAC was developed in C++ in the OMNeT++ simulation software. Crossbow provides the TinyOS/nesc programming platform to write applications for the sensor nodes. A great effort was spent in migrating the AS-MAC protocol codes from OMNet++ environment over to the TinyOS/nesc programming platform and all the necessary components, including Client Tire, Server Tire and Mote Tire, and functional blocks were developed and wired up. **Figure 8-2** depicts the block level architecture of the programming concepts in TinyOS/nesc.

To acquire live data readings from the environment, the light sensor module on the MTS400 sensor-board of the IRIS mote was interfaced with the program application. Note that the sensor-board contains several other sensors (such as temperature sensor, barometer etc) and they can be similarly interfaced with the AS-MAC program to obtain more types of sensor readings when needed.

Figure 8-2: Block level architecture of the programming concepts in TinyOS/nesc

The full implementation of the AS-MAC has been successfully converted from the OMNeT++ environment into the TinyOS/nesc platform. Tests were run to assess the functionality of various components in the AS-MAC and the sending/receiving/intercepting abilities were determined to be operational. The flowcharts in Appendix A depict of logic implementation flow of the program components of AS-MAC in TinyOS/nesc.

8.3.2 Limitation of Physical Capability of Wireless Sensors

8.3.2.1 Non-functional Time Synchronization Service

The Time Synchronization Service provided by the Crossbow WSN Kit is used to establish a global network time stamp and to schedule network communication. The purpose of performing time synchronization is to synchronize all the nodes in the network to the same time. This is accomplished by declaring the time on the base station mote as the global time, then synchronizing all the nodes within the mesh network to the global time. To obtain the current global time from the Time Synchronization Service, the AS-MAC program must be wired to the *TimeSyncService* module using the *Time* interface as described in Appendix B-1. However the compilation pointed out the two critical errors in *TimeSyncM* module which is inside of *TimeSyncService* as shown in Appendix B-2. Since the *TimeSyncService* component provided by the Crossbow WSN Kit is not functional, other components providing the *Time* and *TimeSet* interfaces were explored. The *LogicalTime* and *SimpleTimeM* components were verified to provide the *Time* interface. Though they successfully compile with the AS-MAC program, their functionality in acquiring time was tested and determined to be defective as well. As a result, more components providing serviceable *Time* and *TimeSet* interfaces would have to be sourced.

We may see the limitation of the proposed protocols from the actual constraints of physical sensors. Regardless the type of the proposed protocols: synchronized or hybrid MAC protocol, as long as they involve in time synchronization of all the nodes, a reliable Time Synchronization Service provided by the physical sensors becomes vital. The overhead incurred to overcome the obstacle should be considered as well.

9 Conclusions

Recent advancements in low-cost, low-power, and multi-functional sensor nodes as well as in wireless communications technologies have led to emerging research in the field of wireless sensor networks. A set of applications, such as intelligent battlefields, earthquake response systems, military object tracking, reconnaissance surveillance, hazardous environment exploration monitoring, industry monitoring, and smart hospitals, become the key motivations of the recent rise in research efforts in this area. In such diverse application scenarios, it is a big challenge and cost prohibitive to address different application-specific requirements through retrofitting the deployed protocol. Hence, researching and developing a suite of energy-efficient, adaptive and application-aware MAC protocols adaptable to the diverse applications are motivating our research.

9.1 Summary of Contributions

To our best knowledge, few research results are found to address application-awareness issue through adaptive MAC protocols. The contributions of this thesis are summarized in the following three-folds:

First-fold, we have proposed two MAC protocols that are suitable to application-specific requirements. First, we proposed an energy-efficient, low-latency and high packet delivery ratio MAC protocol for multi-hop WSNs – AS-MAC. This AS-MAC protocol uses a novel Adaptive Scheduling (AS) technique to handle the data traffic. AS-MAC protocol originally combines contention-based approach controlled by a timeout mechanism with scheduling-based approach into a hybrid MAC protocol. AS-MAC protocol employs timeout mechanism to create a resilient active time (RAT) within the AS-period, so as to allow nodes to schedule more data transmission within one operational cycle when the traffic load is high and go to sleep early when the traffic load is low. Second, we proposed an energy-efficient, self-adaptive MAC protocol for traffic-dynamic WSNs – SEA-MAC. This SEA-MAC protocol employs dual-adaptable mechanisms in protocol design to fulfill the adaptability to traffic loading dynamic requirement. SEA-MAC not only has a resilient active time (RAT) in AS-period, but also has a resilient T_{as}/T_{sleep} ratio (RR), so as to be further adaptive to the traffic loads.

Experimental results have shown that both protocols outperform the compared MAC protocols in terms of energy efficiency, average end-to-end delay as well as packet delivery ratio.

Second-fold, we modeled application-specific requirements from two aspects: traffic-aware characterization and application-aware rendering for the proposed suite of MAC protocols. We generalized the traffic-aware characterization and application-aware rendering with parameters and make the protocol adaptable to application-specific performance criteria.

Third-fold, we provide projections on the future research of MAC scheme by presenting a dual-behaviour MAC scheme and the qualitative analysis of performance of the two types of protocols in the following section. Since the dual-behaviour MAC scheme combines contention-based protocol with scheduling-based protocol as a hybrid MAC scheme, and leverages the advantages of both types of protocol in different traffic loads and patterns, it will open the new opportunities for the peers working in this area to research and develop more adaptive and agile MAC protocols.

9.2 Projections on Future Research of MAC Protocols for WSNs

Based on the literature studies, discussions and realization of two novel MAC protocols as well as the performance comparisons amongst the protocols, we recognize that adaptability is crucial to achieve energy efficiency, low latency, packet delivery ratio and other performance metrics. Also bearing in mind those challenges posed on the MAC layer protocol design for WSNs as aforementioned, we foresee the future research on the design and development of MAC layer protocols for WSNs as follows.

9.2.1 Hybrid and Self-adaptable Protocols Are Needed

The effective approach to provide balance among multiple metrics, in the author's opinion, is to use dual-behaviour hybrid scheme that integrates the advantages of two types of existing MAC layer protocol solutions into a Performance-Driven Self-Adaptable MAC (PDSEA-MAC) protocol. The following figures and descriptions illustrate the concepts and effect of this scheme. They are qualitative and indicative only.

The performance of contention-based and scheduling-based MAC protocol are qualitatively shown in **Figure 9-1 (a)** and **(b)** respectively. Each curve qualitatively represents the moving trend of different performance metrics in the MAC protocol discussed; the vertical bar qualitatively represents performance threshold level beyond which the performance of that type of protocol is unacceptable to a certain application. For instance, the curve labelled with Energy-Saving (CB) in **Figure 9-1 (a)** represents the performance trend of energy saving of contention-based MAC protocol. The contention-based MAC protocol will get very good performance in energy-saving when the network is in low traffic load. The energy-saving of the protocol will drop to a medium level with the increasing traffic load; and it will reach to a level that is unsustainable in term of network lifetime when the network is in high traffic load. With the increasing traffic load, on the other hand, the collisions are increasing due to heavy traffic load and end-to-end delay will be prolonged due to the queuing-up traffics, thus the Collision-Density (CB) and Latency (CB) curves are moving up and eventually reach to a level that the serious collisions make the energy consumption worse, and the latency becomes unacceptable to the application. When a critical one or majority of the performance metrics of an application reaches to the unacceptable threshold level (say TL1), we say that the protocol reaches the bound of that type of protocol (say CB-Bound) for the application. The similar elaborations are applicable to **Figure 9-1 (b)**.

(a): Qualitative performance of contention-based MAC protocol

(b): Qualitative performance of schedule-based MAC protocol

Figure 9-1: Indicative performance of MAC protocol

One possible approach is to design a bound (CB-Bound) for contention-based protocol that is composed of specific application-oriented performance metrics, such as energy-saving, throughput, latency, etc. While some metrics might not be directly taken as performance metrics, they are critical factors to determine the bound, such as collision density (number of the collisions occurring within unit interval), effective-bandwidth (the bandwidth only used for data transmission). Different weights may be assigned to each metric to reflect the specifics of the application requirements as discussed in Chapter 8. Once measurement of the metrics is above the bound, such as its collision density and latency go up, throughput or even effective-bandwidth downturn to an unacceptable level, say from point TL1, it indicates that the WSN is experiencing a high traffic load that result in unacceptable performance. From this moment, the network can auto-switch to a re-configurable scheduling-based MAC protocol so as for the network to leverage the advantages of collision-free protocol. The small solid dots in Figure 9-2 represent the turning points of the performance due to switching of the protocol. The performance curves re-route from dot lines to solid lines, thus dominating performance metrics gain. For instance, the collision density changes to collision-free; the throughput changes from downgrading to upgrading though in slow rate; the energy-saving jumps to a high level with a slow degrading rate. Re-configurability might include parameters pertinent to the current traffic load, number of one-hop and two-hop neighbours, etc. The parameters can be fed into the scheduling-based MAC protocol so as for it to assign time slots to each node for scheduling transmission more rationally.

Vice versa, a bound for scheduling-based protocol (SB-Bound) can be composed of another set of performance metrics, such as throughput, latency, etc. Different weights

apply to the bound here too. Once measurement of the metrics is beyond the bound, such as it might start to experience unnecessary latency and decreased throughput, say from point TL2, it indicates that the WSN is under a low traffic load. While it is in favour of energy-saving, but unnecessary latency and decreased throughput might be unacceptable for the application. Although low throughput is sometimes because of low traffic load, when traffic load is unbalanced, nodes with more data to transmit will experience unnecessary high latency and low throughput. When such a low-throughput and high-latency scenario are developing to a level unacceptable from performance point of view, the network can auto-switch back to the contention-based protocol so as for the network to leverage the advantages of contention-based protocol. From the turning points in **Figure 9-2**, the performance curves re-route to solid lines, and dominating performance metrics for CB protocol gain, but with cost of sacrificing minor metrics. For example, the throughput jumps to a high level with a slow descending rate; Latency is reduced to a lower level. But energy-saving degrades to a low level; and collision-free changes to non-collision-free. CB protocol is also in favour of high scalability and adaptability.

(a): Dominating performance metrics gain when SB protocol takes over in high traffic load

(b) Dominating performance metrics gain when CB protocol takes over in low traffic load

Figure 9-2: Dominating performance metrics gain when dual-behaviour protocol switches between SB protocol and CB protocol

Two types of MAC protocol can be fused to a hybrid scheme by providing a pair of bound threshold to control the auto-switching as described above. Once a performance metric is defined, the bounds can be determined. The optimal algorithms purely for scheduling-based and contention-based MAC protocols can be concisely designed respectively. Thus complexity of the algorithms remains low, but it might need a bit more memory at each of the sensor nodes to store this dual-behaviour protocol, which won't be a big concern in the near future due to the fast advancement of sensor technologies.

9.2.2 Cross-Layer Interaction and Design Are Needed

While Energy-efficient MAC layer protocols and other layers of the protocol stack for WSNs are and will continuously be attracting researchers' attention, as aforementioned, they are not jointly optimized in order to maximize the overall network performance while minimizing the energy expenditure. In order to offer satisfactory network performance, the MAC layer, routing at the network layer and the transport layer protocols need to interactively and synergically work together with the PHY layer. For example, the packet loss rate in the MAC layer can be reported to the transport layer so that a TCP protocol is able to differentiate congestion from packet loss [44]. Cross-layer design can be performed in two approaches. The first is the *cross-layer interaction*, where the traditional layered structure is preserved, but each layer is informed of the conditions of other layers in order to improve overall performance. The second

approach is to merge layered protocols into one unified component by a *cross-layer module* (XLM) design. In general, the first approach keeps the transparency between protocol layers, thus preserve the flexibility of each layer while the second approach can achieve much better performance through tighter interaction between layers.

An increasing number of recent researches have exhibited promising advantages of cross-layer interaction and design of WSNs protocols [8] [39] [41] [45] [46] [47] [48]. The authors in [8] have introduced protocol stack tree (PST) to facilitate loosely-coupled cross-layer interaction so as to meet requirements of a given class of applications. In [39], a unified cross-layer protocol has been designed and developed, which replaces the entire traditional layered protocol architecture that has been used so far in WSNs in order to achieve resource-efficiency. To date, a few research results have been reported on unified cross-layer protocols to efficiently address various communication issues spanning entire protocol stack including transport, routing, MAC with physical layer efforts for WSNs [39]. In [41], two sets of cross-layer design had been developed. One is an MAC+Routing cross-layer design that integrates dynamic scheduling at the data link layer with diversity routing at the network-layer. Another is an MAC+PHY cross-layer design that enables data link layer protocol to provide detailed control of radio states at the physical layer. The authors of [45] had developed a set of tightly integrated networking protocols (MAC+Routing) to achieve high energy-efficiency and prolong the network lifetime, especially when nodes are mobile. The authors of [46] illustrated, through a non-linear optimization model, a joint optimal design of PHY, MAC and routing layers in order to maximize the lifetime of energy-constrained WSNs. The authors of [47] had used forward error correlation (FEC) coding with considerations of the characteristics of both PHY layer and MAC layer to determine awake/sleep periods for narrowband WSNs to maximize overall energy-efficiency. In [48], authors have proposed a priority-based scheduler at MAC layer for multiple connections with diverse QoS requirements. Each connection employs adaptive modulation coding (AMC) scheme at PHY layer over wireless fading channels. Priority-function is updated dynamically depending on the channel quality, QoS satisfaction and service priority across layers. The literatures [49] [50] have provided more detailed reviews on cross-layer design techniques and methodologies, proposed taxonomy of cross-layer design, described open research issues, and discussed possible shortcomings of cross-layer design for WSNs.

Although cross-layer design creates interactions, some intended, some unintended, we believe that this technique, if used in a cautionary approach, can bring about significant improvement in terms of overall energy conservation in WSNs and fulfil the various communication requirements in WSNs.

9.3 Concluding Remarks

Moving forward, we believe that adaptive MAC protocols can effectively and efficiently enable adaptability to dynamic traffic loads, thus they are significantly important towards application-aware wireless sensor networks. The wireless sensor networks built with adaptive mechanisms in MAC layer protocol or through cross-layer design can be adaptable to the diversified applications, and ease the development and deployment of wireless sensor networks for the applications.

We believe that modelling of application-specific wireless sensor networks through traffic-aware characterization and application-aware rendering provides a generic framework for realizing the adaptive MAC protocols to deal with the dynamic traffic loading and application-specific requirements. Realization of traffic-aware characterization and application-aware rendering provides a proactive adaptive wireless sensor network. We also believe that dual-behaviour hybrid schemes and cross-layer interaction and design are promising approaches which are well-suited for the unique challenges of wireless sensor networks. Dual-behaviour hybrid schemes provide a reactive adaptive wireless sensor network. Cross-layer interaction and design open alternative opportunities towards the adaptive solutions for wireless sensor networks.

Appendix A: Logic Flowcharts of AS-MAC in TinyOS/nescC

Figure A-1: Initialization of the AS-MAC

Figure A-2: Logic flow during the Sync period

Figure A-3: Logic flow during the AS-period

Figure A-4: Logic flow during the Sleep period

Appendix B: Time Interface

Table B-1: TimeSyncService and Wiring to the TimeSyncService

```

interface Time {
    // get current time. Return it in tos_time_t structure
    async command tos_time_t get();
    // get the high 32 bits of current time
    async command uint32_t getHigh32();
    // get the lower 32 bits of current time
    async command uint32_t getLow32();
    // get clock phase offset since last clock tick in unit of micro-seconds.
    async command uint16_t getUs();
}

```

The wiring to the TimeSyncService is performed via:

```

uses {
    interface StdControl as TimeStart;
    interface Time;
}
...
Implementation {
    components MyAppM, TimeSyncService;
    MyAppM.TimeStart -> TimeSyncService;
    MyAppM.Time -> TimeSyncService;
}
...
}

```

Table B-2: TinyOS/nesC Compilation Messages

```

In file included from
C:/Crossbow/cygwin/opt/MoteWorks/tos/lib/TimeSync/TimeSyncService.nc:26, from
MyApp.nc:21:
In component `TimeSyncM':
C:/Crossbow/cygwin/opt/MoteWorks/tos/lib/TimeSync/TimeSyncM.nc: In function
`Time.get':
C:/Crossbow/cygwin/opt/MoteWorks/tos/lib/TimeSync/TimeSyncM.nc:119: interface has
no command or event named `ticksLeft'
C:/Crossbow/cygwin/opt/MoteWorks/tos/lib/TimeSync/TimeSyncM.nc: In function
`Time.getLow32':
C:/Crossbow/cygwin/opt/MoteWorks/tos/lib/TimeSync/TimeSyncM.nc:138: interface has
no command or event named `ticksLeft'
C:/Crossbow/cygwin/opt/MoteWorks/tos/lib/TimeSync/TimeSyncM.nc: In function
`ReceiveMsg.receive':
C:/Crossbow/cygwin/opt/MoteWorks/tos/lib/TimeSync/TimeSyncM.nc:311:
`TRANS_TIME' undeclared (first use in this function)
make: *** [exe0] Error 1

```


Author's Publications

1. Journal

- [1] **Yi Zhi Zhao**, Chunyan Miao, Maode Ma, Jing Bing Zhang and Cyril Leung, "A Survey and Projection on Medium Access Control Protocols for Wireless Sensor Networks", (accepted by journal "ACM Computing Survey" and to appear in print in vol. 45, issue # 1 (Impact Factor: 12.7))
- [2] **Yi Zhi Zhao**, Maode Ma, Chunyan Miao, Thuy Ngoc Nguyen, "An Energy-Efficient and Low-Latency MAC Protocol with Adaptive Scheduling for Multi-hop Wireless Sensor Networks", *Computer Communications*, 33 (2010) 1452–146, <http://dx.doi.org/10.1016/j.comcom.2010.03.025>.
- [3] **Yi Zhi Zhao**, Maode Ma, "Rationality of Critical Parameters Determination in AS-MAC Protocol for Multi-hop Networks", *International Journal of Information Technology*, Vol. 17 No. 1 2011, 1 – 15.
- [4] **Yi Zhi Zhao**, Chunyan Miao, Maode Ma, You Cheng, "A Self-Adaptable Energy-Efficient Medium Access Control Protocol for Wireless Sensor Networks", *Wireless Personal Communications*, published online on 14 Jan. 2012, DOI [10.1007/s11277-012-0508-7](https://doi.org/10.1007/s11277-012-0508-7).
- [5] **Yi Zhi Zhao**, Chunyan Miao, Maode Ma, Li Shiwei, "A Performance-driven Dual-behavior SELF-Adaptable Medium Access Control Protocol (PDSEA-MAC) for Multi-hop Wireless Sensor Networks", to be submitted to *Computer Communications*.

2. Conference

- [1] **Yi Zhi Zhao**, Thuy Ngoc Nguyen, Maode Ma, Chunyan Miao, "An Energy-Efficient MAC Protocol with Adaptive Scheduling for Wireless Sensor Networks", *IEEE International Conference on Industrial Informatics (INDIN2009)*, 23-26 June 2009, Cardiff, UK, vol. 1, pp. 446-451. DOI: [10.1109/INDIN.2009.5195845](https://doi.org/10.1109/INDIN.2009.5195845).
- [2] **Yi Zhi Zhao**, Chunyan Miao and Maode Ma, "Performance of Adaptive Scheduling MAC (AS-MAC) Protocol with Different AS-Period in Multi-hop Networks", *The 6th IEEE Conference on Industrial Electronics and Applications (ICIEA2011)*, 21-23 June 2011, Beijing, China. pp. 1881 – 1886. DOI: [10.1109/ICIEA.2011.5975898](https://doi.org/10.1109/ICIEA.2011.5975898).

Bibliography

- [1] S. Kumar, V. S. Raghavan, J. Deng, “Medium Access Control protocols for ad hoc wireless networks: a survey”, *Ad Hoc Networks*, 2006, pp. 326–358. DOI= <http://dx.doi.org/10.1016/j.adhoc.2004.10.001>. Available online 2 November 2004
- [2] Ian F. Akyildiz, W. Su, Y. Sankarasubramaniam, Y., and E. Cayirci, “Wireless Sensor Networks: A Survey”, *Computer Networks (Elsevier) Journal*, Vol. 38, No. 4, March 2002. pp. 393-422. DOI= [http://dx.doi.org/10.1016/S1389-1286\(01\)00302-4](http://dx.doi.org/10.1016/S1389-1286(01)00302-4).
- [3] Pawel Piotr Czapski, “A Survey: MAC Protocols For Applications of Wireless Sensor Networks”, *IEEE TENCON 2006 Conference*, November 2006. pp. 1– 4. DOI= <http://doi.acm.org/10.1145/1160143.1160161>.
- [4] Demirkol, I.; Ersoy, C.; Alagoz, F. “MAC protocols for wireless sensor networks: a survey”, *IEEE Communications Magazine*, Volume 44, Issue 4, April 2006, pp. 115 – 121.
- [5] Piyush Naik and Krishna M. Sivalingam, “A Survey of MAC Protocols for Sensor Networks”, In: C. S. Raghavendra, Krishna M. Sivalingam and Taieb Znati (Eds), “Wireless Sensor Networks”, Kluwer Academic Publishers, ISBN:1-4020-7883-8. pp. 93-107.
- [6] Jennifer Yick, Biswanath Mukherjee, Dipak Ghosal, “Wireless sensor network survey”, *Computer Networks* 52 (2008) 2292–2330.
- [7] Ajay Chandra, V. Gummalla and John O. Limb, “Wireless Medium Access Control Protocols”, *IEEE Communications*, Second Quarter 2000. pp. 2-15.
- [8] Arjan Duresij, “Architectures for Heterogeneous Wireless Sensor Networks” (Invited Paper), 2005 IEEE 16th International Symposium on Personal, Indoor and Mobile Radio Communications, *PIMRC'2005*. Berlin, Germany, September 11-14, 2005, pp.189-1296.
- [9] C.K. Nguyen and A. Kumar, “Medium Access Control with Adjustable Sleeps for Wireless Sensor Networks”, *Proceedings of the IEEE Symposium on Computers and Communications (ISCC 2006)*, 2006, pp. 270–276. DOI= <http://dx.doi.org/10.1109/ISCC.2006.96>.
- [10] Wei Ye, John Heidemann, and D. Estrin, “Medium Access Control with Coordinated Adaptive Sleeping for Wireless Sensor Networks,” *IEEE/ACM Transactions on*

- Networking*, Vol. 12, No. 3, June 2004, pp. 493–506. DOI= <http://dx.doi.org/10.1109/TNET.2004.828953>.
- [11] Suresh Singh and C. S. Raghavendra, “PAMAS: Power Aware Multi-Access Protocol with Signaling for Ad Hoc Networks,” *ACM Computer Communication. Review*, Vol. 8, No. 3, July 1998, pp. 5–26
- [12] Wei Ye, John S. Heidemann, and Deborah Estrin. “An Energy-Efficient MAC Protocol for Wireless Sensor Networks”, In *Proceedings of the 21st Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM 2002)*, pp.1567–1576, June 2002.
- [13] Jing Ai, Jingfei Kong and Damla Turgut, “An Adaptive Coordinated Medium Access Control for Wireless Sensor Networks”, *Proceedings of Ninth International Symposium on Computers and Communications, (ISCC 2004)*, pp. 214–219.
- [14] Tijs van Dam and Koen Langendoen "An Adaptive Energy-Efficient MAC Protocol for Wireless Sensor Networks". The First ACM Conference on Embedded Networked Sensor Systems (SenSys 2003), Los Angeles CA. November 2003. SenSys '03. ACM, New York, NY, 171-180. DOI= <http://doi.acm.org/10.1145/958491.958512>.
- [15] C.K. Nguyen and A. Kumar, “An Energy-Aware Medium Access Control Protocol with Frequent Sleeps for Wireless Sensor Networks”, *Proceedings of the 10th IEEE Symposium on Computers and Communications (ISCC 2005)*, June 2005, pp. 386–391. DOI= <http://dx.doi.org/10.1109/ISCC.2005.26>.
- [16] K. Jamieson, H. Balakrishnan, and Y. C. Tay. (2003, May), “Sift: A MAC protocol for event-driven wireless sensor networks”. MIT Lab. Computer Science Technology. Rep. 894. [Online]. Available: <http://www.lcs.mit.edu/publications/pubs/pdf/MIT-LCS-TR-894.pdf>
- [17] Y. C. Tay, K. Jamieson, H. Balakrishnan. “Collision-Minimizing CSMA and Its Applications to Wireless Sensor Networks”, *IEEE Journal on Selected Areas in Communications*, vol. 22, No. 6, August 2004, pp. 1048 – 1057.
- [18] V. Rajendran, K. Obraczka, and J. J. Garcia-Luna-Aceves, “Energy-Efficient, Collision-Free Medium Access Control for Wireless Sensor Networks,” 1st published in *Proceedings of the first international conference on Embedded networked sensor systems*. ACM Press, 2003, pp. 181–192. 2nd published in *Wireless Networks* Vol.12, No.2, February 2006, pp. 63-78. DOI= <http://dx.doi.org/10.1007/s11276-006-6151-z>.
- [19] V. Rajendran, J. J. Garcia-Luna-Aceves, and K. Obraczka, “Energy-Efficient, Application-Aware Medium Access for Sensor Networks”, *2005 International*

- Conference on Mobile Adhoc and Sensor Systems Conference (MASS 2005)*, November 2005.
- [20] K. Sobrabi. J. Gao. V. Ailawadhi. G.J. Potlie. "Protocols for self-organization of a wireless sensor network". *IEEE Personal Communications*, October 2000. Pages 16-27.
- [21] Tao Wu and Subir Bisws, "A Self-Reorganizing Slot Allocation Protocol for Multi-cluster Sensor Networks". pp. 309-315, *IEEE*, 2005.
- [22] Samira Yessad, Farid Nat-Abdesselam, Tarik Taleb and Brahim Bensaou, "R-MAC: Reservation Medium Access Control Protocol for Wireless Sensor Networks", *IEEE Computer Society*, 32nd IEEE Conference on Local Computer Networks, pp. 719–724, 2007. DOI= <http://dx.doi.org/10.1109/LCN.2007.134>.
- [23] Shu Du, Amit Kumar Saha, and David B. Johnson. "RMAC: A Routing-Enhanced Duty-Cycle MAC Protocol for Wireless Sensor Networks". In *Proceedings of the 26th Annual IEEE Conference on Computer Communications (INFOCOM 2007)*, pp. 1478–1486, May 2007.
- [24] Yanjun Sun, Shu Du, Omer Gurewitz, David B. Johnson, "DW-MAC: A Low Latency, Energy Efficient Demand Wakeup MAC Protocol for Wireless Sensor Networks", In *Proceedings of The Ninth ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc'08)*, pp. 53-62, May 26-30, 2008. DOI= <http://doi.acm.org/10.1145/1374618.1374627>.
- [25] I. Rhee, A. Warriar, M. Aia, and J. Min. "Z-MAC: a hybrid MAC for wireless sensor networks". In *SenSys '05: Proceedings of the 3rd International Conference on Embedded Networked Sensor Systems*, pp. 90–101, New York, NY, USA, 2005. ACM Press. DOI= <http://doi.acm.org/10.1145/1098918.1098929>.
- [26] I. Rhee, A. Warriar, J. Min, and Lisong Xu, "DRAND: Distributed Randomized TDMA Scheduling for Wireless Ad-hoc Networks", *Proceedings of the seventh ACM international symposium on Mobile ad hoc networking and computing (MobiHoc 2006)*, May 2006. pp. 190-201. DOI= <http://doi.acm.org/10.1145/1132905.1132927>.
- [27] Mastrooreh Salajegheh, Hamed Soroush, and Antonis Kalis, "HYMAC: Hybrid TDMA/FDMA Medium Access Control Protocol for Wireless Sensor Networks", *The 18th Annual IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC'07)*, pp. 1-5, 3-7 Sept. 2007.
- [28] G. Zhou, C. Huang, T. Yan, T. He, J.A. Stankovic and T.F. Abdelzaher, "MMSN: Multi-Frequency Media Access Control for Wireless Sensor Networks," In *Proceedings of the 25th IEEE International Conference on Computer Communications*.

- (INFOCOM2006), pp. 1-13, 2006.
- [29] A. Rowe, R. Mangharam and R. Rajkumar, "RT-Link: A Time Synchronized Link Protocol for Energy Constrained Multi-hop Wireless Networks," The 3rd Annual IEEE Communications Society on Sensor and Ad Hoc Communications and Networks, 2006. SECON '06. pp. 402-411. 8 Sept. 2006.
- [30] FEENEY L. AND NILSSON M. 2001. Investigating the energy consumption of a wireless network interface in an ad hoc networking environment. *IEEE INFOCOM* 2001. Anchorage, Alaska, 3; pp. 1548–1557. DOI= [10.1109/INFCOM.2001.916651](https://doi.org/10.1109/INFCOM.2001.916651)
- [31] SHIH, E., BAHL, P., AND SINCLAIR, M. J. 2002. Wake on wireless: an event driven energy saving strategy for battery operated devices. In *Proceedings of the 8th Annual international Conference on Mobile Computing and Networking* (Atlanta, Georgia, USA, September 23 - 28, 2002). MobiCom '02. ACM, New York, NY, 160-171. DOI= <http://doi.acm.org/10.1145/570645.570666>.
- [32] Wei Ye. Fabio Silva and John Heidemann, "Ultra-Low Duty Cycle MAC with Scheduled Channel Polling", In *SenSys '06: Proceedings of the 4th International Conference on Embedded Networked Sensor Systems*, pp. 321-334, November 1-3, 2006, Boulder, Colorado, USA. ACM Press. DOI= <http://doi.acm.org/10.1145/1182807.1182839>.
- [33] EL-HOIYDI, A. 2002. Spatial TDMA and CSMA with Preamble Sampling for Low Power Ad Hoc Wireless Sensor Networks. In *Proceedings of the Seventh international Symposium on Computers and Communications (Iscc'02)* (July 01 - 04, 2002). ISCC. IEEE Computer Society, Washington, DC, 685.
- [34] EL-HOIYDI A. AND DECOTIGNIE J. 2004. WiseMAC: An Ultra Low Power MAC Protocols for the downlink of infrastructure wireless sensor networks. In *Proceedings of the Ninth IEEE Symposium on Computers and Communication (ISCC2004)*. pp. 244–251. DOI= <http://dx.doi.org/10.1109/ISCC.2004.1358412>.
- [35] BUETTNER, M., YEE, G. V., ANDERSON, E., AND HAN, R. 2006. X-MAC: a short preamble MAC protocol for duty-cycled wireless sensor networks. In *Proceedings of the 4th international Conference on Embedded Networked Sensor Systems* (Boulder, Colorado, USA, October 31 - November 03, 2006). SenSys '06. ACM, New York, NY, 307-320. DOI= <http://doi.acm.org/10.1145/1182807.1182838>.
- [36] Joseph Polastre, Jason Hill, and David Culler. "Versatile low power media access for wireless sensor networks". In *Proceedings of the 2nd ACM SenSys Conference*, pp. 95–107, Baltimore, MD, USA, November 2004. ACM.

DOI= <http://doi.acm.org/10.1145/1031495.1031508>.

- [37] John A. STANKOVIC, Tarek F. ABDELZAHER, Chenyang LU, Lui Sha, and Jennifer C. HOU, "Real-Time Communication and Coordination in Embedded Sensor Networks", *PROCEEDINGS OF THE IEEE*, Volume 91, Issue 7, July 2003, pp. 1002 – 1022.
- [38] Paolo Barontib, c, Prashant Pillaia, Vince W.C. Chooka, Stefano Chessab, c, , Alberto Gottab and Y. Fun Hua, "Wireless sensor networks: A survey on the state of the art and the 802.15.4 and ZigBee standards", *Computer Communications* Volume 30, Issue 7, 26 May 2007, pp. 1655-1695.
- [39] Ian F. Akyildiz, Mehmet C. Vuran and Ozgur B. Akan, "A Cross-Layer Protocol for Wireless Sensor Networks", In *Proceedings of 2006 40th Annual Conference on Information Sciences and Systems*, March 2006 pp.1102 – 1107.
- [40] V. Kawadia, and P. R. Kumar, "A Cautionary Perspective on Cross-Layer Design," *IEEE Wireless Communications*, February 2005, pp. 3-11.
- [41] Brian DeCleene, Victor Firoiu, Matthew Dorsch, Steve Zabele, "Cross-layer Protocols for Energy-Efficient Wireless Sensor Networking". *IEEE Military Communications Conference*. 3 pp. 1477–1484.
- [42] M. Younis. K. Akkaya. M. Eltoweissy, .A. Wadaa, "On Handling QoS Traffic in Wireless Sensor Networks", in the *Proceedings of the International Conference HAUAI International Conference on System Sciences (HICSS-37)*, Big Island, Hawaii, January 2004.
- [43] Kaustubh S. Phanse, Luiz A. DaSilva, "Addressing the requirements of QoS management for wireless ad hoc networks", *Computer Communications* 26 (2003), pp. 1263–1273.
- [44] Ian F. Akyildiz, Xuedong Wang, "A Survey on Wireless Mesh Networks", *IEEE Radio Communications*, September 2005, pp. s23-s30.
- [45] L. van Hoesel, T. Nieberg, J. Wu, and P. J. M. Havinga, "Prolonging the lifetime of wireless sensor networks by cross-layer interaction," *IEEE Wireless Communications*, Vol. 11, No. 6, December 2004, pp. 78 – 86.
- [46] Ritesh Madan, Shuguang Cui, Sanjay Lall, Andrea Goldsmith, "Cross-Layer Design for Lifetime Maximization in Interference-Limited Wireless Sensor Networks", *IEEE Transactions on Wireless Communication*, Vol. 5, No. 11, November 2006, pp. 3142-3152. DOI= [10.1109/TWC.2006.04770](http://doi.acm.org/10.1109/TWC.2006.04770)

- [47] Heikki Karvonen, Carlos Pomalaza-Ráez, “A Cross Layer Design of Coding and Awake/Sleep Periods in WSNs”. The 17th Annual IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC'06), 2006.
- [48] Qingwen Liu, Xin Wang, and Georgios B. Giannakis, “A Cross-Layer Scheduling Algorithm with QoS Support in Wireless Networks”, IEEE Transactions on Vehicular Technology, Vol. 55, No. 3, May 2006, pp. 839-847.
- [49] Tommaso Melodia, Mehmet C. Vuran, and Dario Pompili, “The State of the Art in Cross-Layer Design for Wireless Sensor Networks”, Wireless Syst. /Network Architect. 2005, LNCS 3883, pp. 78–92, 2006.
- [50] Vijay T. Raisinghani, Sridhar Iyer, “Cross-layer design optimizations in wireless protocol stacks”, Computer Communications 27 (2004), pp. 720–724.
- [51] Giuseppe Anastasi, Marco Conti, Mario Di Francesco, Andrea Passarella, “Energy conservation in wireless sensor networks: A survey”, Ad Hoc Networks 7 (2009) 537–568.
- [52] Thanos Stathopoulos, Rahul Kapur, Deborah Estrin, John Heidemann, Lixia Zhang, “Application-Based Collision Avoidance in Wireless Sensor Networks”, Proceedings of the 29th Annual IEEE International Conference on Local Computer Networks (LCN'04),
- [53] Bashir Yahya and Jalel Ben-Othman, “Towards a classification of energy aware MAC protocols for wireless sensor networks”, WIRELESS COMMUNICATIONS AND MOBILE COMPUTING. [DOI: 10.1002/wcm.743](https://doi.org/10.1002/wcm.743)
- [54] Christopher K. Nguyen and Anup Kumar, “A Wireless-Sensor-Network Energy-Efficient Medium-Access-Control Protocol with Overhearing Avoidance”, The Proceedings IEEE CCNC 2006, pp. 773-777.
- [55] Junzhao Du and Weisong Shi, App-MAC: An Application-Aware Event-Oriented MAC Protocol for Multimodality Wireless Sensor Networks, IEEE Transactions on Vehicular Technology, Vol. 57, No. 6, November 2008.
- [56] Seung Hyong Rhee; Kwangsue Chung; Yongsuk Kim; Wonyong Yoon; Ki Soo Chang, “An application-aware MAC scheme for IEEE 802.15.3 high-rate WPAN”, Volume 2, 21-25 March 2004 Page(s):1018 - 1023 Vol.2. IEEE Transactions on Vehicular Technology.
- [57] Yu Wei*, Liu Xingjie*, Cai Yu+, Zhou Zucheng, “An Application-aware Delayed-ACK for Video Streaming over IEEE 802.15.3 WPANs”, IEEE 63rd Vehicular

- Technology Conference, 2006. VTC 2006-Spring.
- [58] Chigo Okonkwo; Martin, R.; Ferrera, M.P.; Guild, K.; O'Mahony, M.; Everett, J., "Demonstration of an Application-Aware Resilience Mechanism for Dynamic Heterogeneous Networks", Page(s): 20-23, The proceedings of ICTON2007.
- [59] Bai, Y.; Ito, M.R, "Application-aware buffer management: new metrics and techniques", Volume 51, Issue 1, March 2005 Page(s):114 – 121, IEEE Transactions on Broadcasting.
- [60] de Amorim, M.D.; Duarte, O.C.M.B.; Pujolle, G., " Application-aware multicast ", Page(s): 2506-2510, IEEE GLOBECOM '01. IEEE.
- [61] Xi Chen, Yang Xiao, Yu Cai, Jianhua Lu, Zucheng Zhou, "An energy diffserv and application-aware MAC scheduling for VBR streaming video in the IEEE 802.15.3 high-rate wireless personal area networks", Computer Communications 29 (2006) 3516–3526.
- [62] M. Ghannad Rezaie, V. Shah Mansouri, M. R. Pakravan, "Traffic Aware Dynamic Node Scheduling for Power Efficient Sensor Networks", The Proceedings of ISSNIP2004, pp. 37-42.
- [63] Gavin Holland, Nitin Vaidya, Paramvir Bahl, "A Rate-Adaptive MAC Protocol for Multi-Hop Wireless Networks", ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom2001), Rome, Italy, July 2001.
- [64] S. Chatterjea, L.F.W. van Hoesel and P. Havinga, "An Application-Tailored MAC Protocol for Wireless Sensor Networks".
- [65] Ashish Sharma, Elizabeth M. Belding, "A Case for Application Aware Channel Access in Wireless Networks", *HotMobile* 2009, February 23-24, 2009, Santa Cruz, CA, USA.
- [66] Bhaskar Krihnamachar and John Heridemann, "Application-Specific Modelling of Information Routing in Wireless Sensor Networks", IEEE 2004.
- [67] Shouyi Yin, Yongqiang Xiong, Qian Zhang, Xiaokang Lin. "Traffic-aware routing for real-time communications in wireless multi-hop networks", Volume 6 Issue 6, Pages 825 – 843, [Wireless Communications and Mobile Computing](#), 29 Aug 2006.
- [68] Holger Karl and Andreas Willig: "Protocols and Architectures for Wireless Sensor Networks", 2005.
- [69] <http://www.xbow.com>, <http://robotics.eecs.berkeley.edu/~pister/SmartDust/>
- [70] L. Litwin, "The Medium Access Control Sublayer," *IEEE Potentials*, vol. 20, no. 4, Oct.-Nov. 2001, p. 30-4.

- [71] G. L. Stuber, *Principles of Mobile Communications*, Kluwer Academic Press, 1997
- [72] V. Bharghavan, "A New Protocol for Medium Access in Wireless Packet Networks," publication of the Timely group, 1997, <http://timely.crhc.uiuc.edu/publications.html>
- [73] Mark Stemm and Randy H Katz, "Measuring and reducing energy consumption of network interfaces in hand-held devices," *IEICE Transactions on Communications*, vol. E80-B, no. 8, pp. 1125–1131, Aug. 1997.
- [74] Oliver Kasten, *Energy Consumption*, http://www.inf.ethz.ch/~kasten/research/bathtub/energy_consumption.html, Eidgenössische Technische Hochschule Zurich.
- [75] S. Lu, V. Bharghavan, and R. Srikant, "Fair Scheduling in Wireless Packet Networks," *IEEE/ACM Trans. Net.*, vol.7, no.4, Aug. 1999, <http://timely.crhc.uiuc.edu/publications.html>, pp. 473–89.
- [76] Vishal Ailawadhi, Ph.D thesis, "Mobility Issues in Hybrid Ad-Hoc Wireless Sensor Networks", UNIVERSITY OF CALIFORNIA, Los Angeles, 2002.
- [77] Energy consumption in wireless nodes (2006): <http://www.cwc.oulu.fi>
- [78] [Stojmenovic, M.](#); Nayak, A.; Localized routing with guaranteed delivery and a realistic physical layer in wireless sensor networks; in *Computer Communications*, v 29, n 13-14, 21 Aug. 2006, p 2550-5
- [79] Saurabh Ganeriwal, Ram Kumar, and Mani B. Srivastava., "Timing-Sync Protocol for Sensor Networks", in *Proceedings of the First International Conference On Embedded Networked Sensor Systems (SenSys 2003)*, pages 138-149, Nov. 2003.
- [80] Discrete Event Simulation System (2008): <http://www.omnetpp.org/>.
- [81] Sze-Yao Ni, Yu-Chee Tseng, Yuh-Shyan Chen, and Jang-Ping Sheu. The Broadcast Storm Problem in a Mobile Ad Hoc Network. In *Proceedings of the Fifth Annual International Conference on Mobile Computing and Networking (MobiCom 1999)* pages 151-162, August 19.
- [82] G. P. Halkes, T. Van Dam, and K.G. Langendoen, "Comparing energy-saving MAC protocols for wireless sensor networks", *Mobile Networks and Applications*, 10(5):783–791, October 2005.
- [83] Rajiv Gandhi, Srinivasan Parthasarathy, and Arunesh Mishra, Minimizing Broadcast Latency and Redundancy in Ad Hoc Networks. In *Proceedings of the Fourth ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc 2003)*, Jun. 2003, pp. 222–232. doi>[10.1145/778415.778442](https://doi.org/10.1145/778415.778442)

- [84] J. Hill, R. Szewczyk, A. Woo, S. Hollar, D. Culler, and K. Pister, System Architecture Directions for Networked Sensors, ASPLOS, November 2000.
- [85] R. R. Brooks and S. S. Iyengar. *Multi-sensor fusion: fundamentals and applications with software*. Prentice Hall, Inc, New Jersey, 1997.
- [86] R. C. Luo and M. G. Kay. Multisensor integration and fusion in intelligent systems. *IEEE Transactions on Systems, Man, and Cybernetics*, 19(5):901–931, September/October 1989.
- [87] K. A. Delin and S. P. Jackson. Sensor web for in situ exploration of gaseous biosignatures. In *Proceedings of 2000 IEEE Aerospace Conference*, Big Sky, MT, March 2000.
- [88] X. Yang, K. G. Ong, W. R. Dreschel, K. Zeng, C. S. Mungle, and C. A. Grimes. Design of a wireless sensor network for long-term, in-situ monitoring of an aqueous environment. *Sensors*, 2(7):455–472, 2002.
- [89] A. Cerpa, J. Elson, M. Hamilton, and J. Zhao. Habitat monitoring: application driver for wireless communications technology. In *2001 ACM SIGCOMM Workshop on Data Communications in Latin America and the Caribbean*, April 2001. DOI=[10.1145/371626.371720](https://doi.org/10.1145/371626.371720).
- [90] B.W. West, P. G. Flikkema, T. Sisk, and G.W. Koch. Wireless sensor networks for dense spatio-temporal monitoring of the environment: a case for integrated circuit, system, and network design. In *Proceedings of 2001 IEEE CAS Workshop on Wireless Communications and Networking*, August 2001.
- [91] C. Chong and S. P. Kumar. Sensor networks: evolution, opportunities, and challenges. *Proceedings of the IEEE*, 91(8):1247–1256, August 2003.
- [92] D. Shepherd. Networked microsensors and the end of the world as we know it. *IEEE Technology and Society Magazine*, pages 16–22, Spring 2003.
- [93] J. M. Caicedo, E. Clayton, S. J. Dyke, and M. Abe. Structural health monitoring for large structures using ambient vibrations. In *Proceedings of the ICANCEER Conference*, Hong Kong, August 2002.
- [94] Hongwei Zhang, Anish Arora, Young-ri Choi, and Mohamed G. Gouda. Reliable Bursty Convergecast in Wireless Sensor Networks. In *Proceedings of the Sixth ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc 2005)*, pages 266–276, May 2005.
- [95] T. T. Hsieh. Using sensor networks for highway and traffic applications. *IEEE*

- Potentials*, pages 13–16, April/May 2004.
- [96] A. N. Knaian. A wireless sensor network for smart roadbeds and intelligent transportation systems. Master's thesis, Massachusetts Institute of Technology, June 2000.
 - [97] D. Estrin, R. Govindan, J. Heidemann, and S. Kumar. Next century challenges: scalable coordination in sensor networks. In *Mobile Computing and Networking*, pages 263–270, 1999.
 - [98] LANGENDOEN K 2008, Medium Access Control in Wireless Networks. Book chapter: Energy-Efficient Medium Access Control, pp. 535–560. 2008.
 - [99] KRISHNAMACHARI, B. 2005. Networking Wireless Sensors. Book chapter: Medium access and sleep scheduling, pp. 82–101. 2006.
 - [100] KREDO, K. II. AND MOHAPATRA, P., 2007. Medium Access Control in Wireless Sensor Networks, *Computer Networks*, 961–994, 2007.
 - [101] HALKES, G. P. AND LANGENDOEN, K. G. 2007. Crankshaft: An Energy-Efficient MAC-Protocol For Dense Wireless Sensor Networks. In *EWSN*, 2007.
 - [102] MAHLKNECHT S. AND BOECK M. 2004, CSMA-MPS: A Minimum Preamble Sampling MAC Protocol for Low Power Wireless Sensor Networks, in: *Proceedings 2004 IEEE International Workshop on Factory Communication Systems*. 2004, pp. 73 – 80, DOI: [10.1109/WFCS.2004.1377683](https://doi.org/10.1109/WFCS.2004.1377683).
 - [103] EL-HOIYDI, A 2002. Aloha with Preamble Sampling for Sporadic Traffic in Ad Hoc Wireless Sensor Networks“, *IEEE International Conference on Communications (ICC)*, Volume 5, pp. 3418 -3423.
 - [104] EL-HOIYDI, A., DECOTIGNIE J.-D., 2003, WiseMAC, An Ultra Low Power MAC Protocol for the WiseNET Wireless sensor Network”, *ACM SenSys*, Los Angeles, 5-7 Nov 2003.
 - [105] Frost & Sullivan, Technical Insights. “Smart Sensors and Sensor Networks: Opportunities for Networked Intelligent Wireless Sensors”. D628 © 2006 Frost & Sullivan 2006.
 - [106] P. Lin, C. Qiao and X. Wang, Medium access control with a dynamic duty cycle for sensor networks, *IEEE Wireless Communications and Networking Conference*, vol. 3, 1534–1539, 21–25 March 2004.

- [107] S.-H. Yang, *et al*, Utilization Based Duty Cycle Tuning MAC Protocol for Wireless Sensor Networks, in: Proceedings of Global Telecommunication Conference 2005, IEEE GLOBECOM 2005, pp. 3258 – 3262, DOI: [10.1109/GLOCOM.2005.1578377](https://doi.org/10.1109/GLOCOM.2005.1578377).
- [108] T. K. Justin, *et al*, An Adaptable Energy-Efficient Medium Access Control Protocol for Wireless Sensor Networks, in *System Sciences, 2007. HICSS 2007. 40th Annual Hawaii International Conference on*, 2007, pp. 292a-292a. DOI=[10.1109/HICSS.2007.59](https://doi.org/10.1109/HICSS.2007.59).
- [109] R. Yadav, S. Varma, N. Malaviya, Optimized medium access control for wireless sensor network, *International Journal of Computer Science and Network Security (IJCSNS)*, 2008; 8(2): 334–338. >.
- [110] Abhiman Hande*, Todd Polk, William Walker and Dinesh Bhatia, Self-Powered Wireless Sensor Networks for Remote Patient Monitoring in Hospitals, *Sensors* 2006, 6, 1102-1117.
- [111] Bachir, A., Dohler M., Watteyne, T., and Leung, K.K. 2010. MAC Essentials for Wireless Sensor Networks. *IEEE Communications Surveys & Tutorials*, VOL. 12, NO. 2, Second Quarter 2010. 222-248. DOI=[10.1109/SURV.2010.020510.00058](https://doi.org/10.1109/SURV.2010.020510.00058).
- [112] Sun, Y., Du, S., Gurewitz, O.& Johnson, David B. (2008). RI-MAC: A Receiver-Initiated Asynchronous Duty Cycle MAC Protocol for Dynamic Traffic Loads in Wireless Sensor Networks. *SenSys'08*, November 5–7, 2008, Raleigh, North Carolina, USA.
- [113] Tang, L., Sun, Y. J., Gurewitz, O. and Johnson, David B (2011). PW-MAC: An Energy-Efficient Predictive-Wakeup MAC Protocol for Wireless Sensor Networks. In *Proceedings of the 30th IEEE International Conference on Computer Communications (INFOCOM 2011)*, pages 1305–1313, April 2011.
- [114] Schurgers, C., Tsiatsis, V., Ganeriwal, S. and Srivastava, M. (2002). Optimizing sensor networks in the energy-latency-density design space. *IEEE Transactions on Mobile Computing*, Vol. 1 No. 1 Jan 2002, DOI: [10.1109/TMC.2002.1011060](https://doi.org/10.1109/TMC.2002.1011060).