

Advertisement-Based Energy Efficient Medium Access Protocols for Wireless Sensor Networks

by

Surjya Sarathi Ray

Submitted in Partial Fulfillment

of the

Requirements for the Degree

Doctor of Philosophy

Supervised by

Professor Wendi B. Heinzelman

Department of Electrical and Computer Engineering

Arts, Sciences and Engineering

Edmund A. Hajim School of Engineering and Applied Sciences

University of Rochester

Rochester, New York

2013

Dedication

This thesis is dedicated to my parents, Indrani and Dulal Ray and my sister,
Sreemoyee Ray.

For their endless love, support and encouragement.

Biographical Sketch

The author received a B.E. (with honors) degree in Electronics and Telecommunications Engineering from Jadavpur University, Calcutta, India in 2006 and an M.S. degree in Electrical Engineering from the University of Rochester, NY, in 2009. He was a teaching assistant in the Department of Electrical and Computer Engineering at the University of Rochester from 2006 to 2008. His research interests include the areas of wireless communications and networking, wireless ad hoc and sensor networks, and the optimization of communication networks. He interned at Toyota ITC in Mountain View, California in the summer of 2011 where he worked on the feasibility of using IEEE 802.15.4 for non critical systems in vehicles. The work is patent pending (United States Patent Application 61548002).

List of Publications:

1. Surjya S. Ray, Ilker Demirkol and Wendi Heinzelman, "Supporting Bursty Traffic in Wireless Sensor Networks through a Distributed Advertisement-based TDMA Protocol (ATMA)," published in Elsevier Ad Hoc Networks Journal.
2. Surjya S. Ray, Ilker Demirkol and Wendi Heinzelman, "ADV-MAC: Analysis and Optimization of Energy Efficiency through Data Advertisements for Wireless Sensor Networks," published in Elsevier Ad Hoc Networks Journal.

3. Surjya S. Ray, Ilker Demirkol and Wendi Heinzelman, “ATMA: Advertisement-based TDMA Protocol for Bursty Traffic in Wireless Sensor Networks,” published in proceedings of IEEE Globecom 2010, Miami, US.
4. Surjya S. Ray, Ilker Demirkol and Wendi Heinzelman, “ADV-MAC: Advertisement based MAC Protocol for Wireless Sensor Networks,” published in proceedings of IEEE 5th International conference of Mobile and Ad-hoc Networks, Wuyishan, China.
5. Surjya S. Ray, Ilker Demirkol and Wendi Heinzelman, “ADV-MAC: Advertisement based MAC Protocol for Wireless Sensor Networks,” poster session on work in progress at Mobiquitous 2009, Toronto, Canada.

Acknowledgments

In the summer of 2006, I left India and came to Rochester, NY with two things in mind. Getting a doctoral degree in Electrical Engineering from the University of Rochester was one of them. The other objective was to be ‘free’ and be myself. I got admission from several other universities across the States and I chose the University of Rochester. In retrospect, it was the best decision of my life. It helped me achieve everything I always wanted and more.

Dr. Wendi Heinzelman is a phenomenal advisor to say the least. Like all great advisors, she is insightful, provided guidance in the right direction and always made sure that I chose the direction myself. And she is much more than that. In 2007, I began the most difficult phase of my life when I came out of the closet. Dr. Heinzelman did an exceptional job helping me balance my personal and professional life. She was extremely understanding and kept encouraging me even when I saw no progress for months at a time. This dissertation would never have been possible without her. Other than my advisor, Dr. Ilker Demirkol who was a post-doctoral fellow in our lab, always encouraged me, pointed me in the right directions and helped me immensely with my research. I owe a great deal to him for my thesis.

Another factor that helped me immensely was my job as web administrator and developer at The Skalny Center here at the University of Rochester. A huge thanks to Dr. Randall Stone and Bozena Sobolewska for giving me full freedom to express my creativity in their websites including the big ones such as the annual

Polish Film Festival. This job not only gave me back my creativity but also improved my coding skills, added to my resumé, gave me a hobby, provided me with extra financial support, and reinforced my confidence.

I would like to express my sincere gratitude to my dissertation committee members, Professors Mark Bocko and Henry Kautz, for agreeing to be on my committee.

I would like to thank my wonderful fiancé Matt Vanderwerf, my dear friends Simantini Ghosh and Itender Singh who always picked up the pieces after me, time and again. Special thanks to Matt's parents, Virginia and Michael Vanderwerf who always made me feel at home. And finally, to my mom, dad and my little sister — I hope you are proud of me!

Abstract

One of the main challenges that prevents the large-scale deployment of Wireless Sensor Networks (WSNs) is providing the applications with the required quality of service (QoS) given the sensor nodes' limited energy supplies. WSNs are an important tool in supporting applications ranging from environmental and industrial monitoring, to battlefield surveillance and traffic control, among others. Most of these applications require sensors to function for long periods of time without human intervention and without battery replacement. Therefore, energy conservation is one of the main goals for protocols for WSNs. Energy conservation can be performed in different layers of the protocol stack. In particular, as the medium access control (MAC) layer can access and control the radio directly, large energy savings is possible through intelligent MAC protocol design. To maximize the network lifetime, MAC protocols for WSNs aim to minimize idle listening of the sensor nodes, packet collisions, and overhearing. Several approaches such as duty cycling and low power listening have been proposed at the MAC layer to achieve energy efficiency. In this thesis, I explore the possibility of further energy savings through the advertisement of data packets in the MAC layer.

In the first part of my research, I propose Advertisement-MAC or ADV-MAC, a new MAC protocol for WSNs that utilizes the concept of advertising for data contention. This technique lets nodes listen dynamically to any desired transmission and sleep during transmissions not of interest. This minimizes the energy lost in idle listening and overhearing while maintaining an adaptive duty cycle to han-

dle variable loads. Additionally, ADV-MAC enables energy efficient MAC-level multicasting. An analytical model for the packet delivery ratio and the energy consumption of the protocol is also proposed. The analytical model is verified with simulations and is used to choose an optimal value of the advertisement period. Simulations show that the optimized ADV-MAC provides substantial energy gains (50% to 70% less than other MAC protocols for WSNs such as T-MAC and S-MAC for the scenarios investigated) while faring as well as T-MAC in terms of packet delivery ratio and latency.

Although ADV-MAC provides substantial energy gains over S-MAC and T-MAC, it is not optimal in terms of energy savings because contention is done twice – once in the Advertisement Period and once in the Data Period. In the next part of my research, the second contention in the Data Period is eliminated and the advantages of contention-based and TDMA-based protocols are combined to form Advertisement based Time-division Multiple Access (ATMA), a distributed TDMA-based MAC protocol for WSNs. ATMA utilizes the bursty nature of the traffic to prevent energy waste through advertisements and reservations for data slots. Extensive simulations and qualitative analysis show that with bursty traffic, ATMA outperforms contention-based protocols (S-MAC, T-MAC and ADV-MAC), a TDMA based protocol (TRAMA) and hybrid protocols (Z-MAC and IEEE 802.15.4). ATMA provides energy reductions of up to 80%, while providing the best packet delivery ratio (close to 100%) and latency among all the investigated protocols.

Simulations alone cannot reflect many of the challenges faced by real implementations of MAC protocols, such as clock-drift, synchronization, imperfect physical layers, and irregular interference from other transmissions. Such issues may cripple a protocol that otherwise performs very well in software simulations. Hence, to validate my research, I conclude with a hardware implementation of the ATMA protocol on SORA (Software Radio), developed by Microsoft Research Asia. SORA

is a reprogrammable Software Defined Radio (SDR) platform that satisfies the throughput and timing requirements of modern wireless protocols while utilizing the rich general purpose PC development environment. Experimental results obtained from the hardware implementation of ATMA closely mirror the simulation results obtained for a single hop network with 4 nodes.

Contributors and Funding Sources

This work was supervised by a dissertation committee consisting of Professors Wendi Heinzelman (advisor) and Mark Bocko of the Department of Electrical and Computer Engineering and Professor Henry Kautz of the Department of Computer Science. The author conducted all of the research work from Chapter 3 to Chapter 5. Each of these chapters has been published as described below. For all of these publications, the author is the primary author, and collaborated with Dr. Ilker Demirkol who is currently a Post-doctoral Research Associate in the Department of Telematics Engineering at the University Politecnica de Catalunya in Barcelona and Dr. Wendi Heinzelman.

The work described in Chapter 3 has been published in the proceedings of the IEEE 5th International conference of Mobile and Ad-hoc Networks, Wuyishan, China, 2009. This work was supported in part by NSF CAREER Grant #CNS-0448046 and in part by a Young Investigator grant from the Office of Naval Research, #N00014-05-1-0626.

The work described in Chapter 4 has been published in the Elsevier Ad Hoc Networks Journal, 2010. This work was supported in part by NSF CAREER Grant #CNS-0448046 and in part by a Young Investigator grant from the Office of Naval Research, #N00014-05-1-0626.

The work described in Chapter 5 has been published in the Elsevier Ad Hoc Networks Journal, 2012. A preliminary version of this work was published in the

in proceedings of IEEE Globecom 2010, Miami, US. This work was supported in part by NSF CAREER Grant #CNS-0448046.

The author conducted all the experiments described in Chapter 6. This project was funded in part by Harris Corporation and CEIS, an Empire State Development-designated Center for Advanced Technology.

Table of Contents

Biographical Sketch	ii
Acknowledgments	iv
Abstract	vi
Contributors and Funding Sources	ix
List of Tables	xiv
List of Figures	xv
1 Introduction	1
1.1 Wireless Sensor Networks	1
1.2 Challenges of Wireless Sensor Networks	4
1.3 Motivations and Contributions	5
1.4 Dissertation Organization	8
2 Related Work	9
2.1 TDMA Based MAC Protocols	9
2.2 Contention Based MAC Protocols	17

2.3	Low Power Protocols	19
2.4	Hybrid MAC Protocols	20
2.5	Summary	24
3	ADV-MAC: Advertisement-based MAC Protocol for Wireless Sensor Networks	25
3.1	ADV-MAC Design Overview	26
3.2	ADV-MAC Performance Evaluation	37
3.3	Summary	47
4	ADV-MAC: Analysis and Optimization for Energy Efficiency	48
4.1	ADV-MAC Design Updates	49
4.2	Analytical Model of ADV-MAC	52
4.3	Optimized ADV-MAC Performance Evaluation	68
4.4	Summary	80
5	ATMA: Advertisement-Based TDMA Protocol for Bursty Traffic in Wireless Sensor Networks	81
5.1	ATMA Design Overview	82
5.2	Optimization of ADV Period	86
5.3	ATMA Performance Evaluation	92
5.4	Summary	106
6	Hardware Implementation of ATMA on the SORA Platform	108
6.1	Choice of Hardware Platform	109
6.2	The SORA Platform Architecture	111
6.3	Implementation of ATMA on the SORA Platform	114

6.4	Experimental Setup and Results	119
6.5	Summary	120
7	Conclusions and Future Directions	124
7.1	Conclusions	124
7.2	Future Directions	125
	Bibliography	128
	Appendices	135
A	Proofs for ADV-MAC Theorems	136
A.1	Proof of Equation (4.1)	136
A.2	Expression of $\xi(x, g)$ and Its Proof	139
A.3	Expected Value of c_i	142

List of Tables

3.1	ADV-MAC: Notations Used	34
3.2	ADV-MAC: Parameter Values	39
4.1	ADV-MAC Analytical Model: Notations Used I	54
4.2	ADV-MAC Analytical Model: Notations Used II	55
4.3	ADV-MAC Optimized Parameter Values	70
5.1	ATMA: Notations Used	89
5.2	ATMA: Simulation Parameter Values	95
6.1	Important SORA UMX APIs	115
6.2	Important Parametersfor the SORA ATMA Experiments	122

List of Figures

2.1	Clusters in LEACH.	11
2.2	Clusters in PACT.	12
2.3	MH-TRACE frame structure.	15
2.4	A snapshot of MH-TRACE clustering and medium access for a portion of an actual distribution of mobile nodes. Nodes C1 - C7 are cluster-head nodes. Reprinted with permission of the authors from [45].	16
2.5	Operation principle of LPL protocols.	20
2.6	Super-frame structure in IEEE 802.15.4	21
3.1	Examples of ADV-MAC, T-MAC and S-MAC communication. The letters after the packets indicate the destination nodes. If the over- hearing avoidance is used by T-MAC, the nodes will be in sleep mode for the hatched areas.	27
3.2	Single hop, unicast vs. data rate: Performance comparison of ADV- MAC, T-MAC and S-MAC. The extension ‘-a’ corresponds to an- alytical results.	38
3.3	Single hop, unicast vs. number of sources: Performance compari- son of ADV-MAC, T-MAC and S-MAC. Extension ‘-a’ represents analytical results.	42

3.4	Single hop, multicast: Energy consumption vs. data rate. The extension ‘-a’ corresponds to analytical results.	43
3.5	Single hop, multicast: Energy consumption vs. number of sources. The extension ‘-a’ corresponds to analytical results.	44
3.6	Multi-hop, unicast vs. data rate: Performance comparison of ADV-MAC, T-MAC and S-MAC.	45
4.1	Contention method in the ADV period. In this example, 5 nodes contend to send, two successfully transmit, two collide and one defers transmission.	50
4.2	Total average time spent by all contending nodes and the probability of collisions in both contention methods.	52
4.3	Symbols used in the analysis.	56
4.4	Example of contention in data period: the node with index number 6 was successful in both the ADV as well as the data periods. The node with index number 3 had a collision in the ADV period and hence times out. The nodes with indices 2 and 7 collide in the data period.	57
4.5	Contention in the data period with expected values of the chosen slots	61
4.6	PDR values obtained from analysis and simulation. This graph is for $N_0 = 2.36$. Confidence intervals of 95% are shown for the simulation results.	71
4.7	Average energy per node per packet obtained from analysis and simulation. This graph is for $N_0 = 2.36$. Confidence intervals of 95% are shown for the simulation results.	73
4.8	Optimal value of S_{adv} and S_{data} for a required PDR of 95% or more for $N_0 = 5$	73

4.9	Single hop: Effect of data rate on different metrics.	74
4.10	Single hop: Effect of number of sources on different metrics. . . .	75
4.11	Multi-hop: Effect of data rate on different metrics.	78
5.1	Examples of the operation of the advertisement period. The numbers in the slots indicate the IDs of nodes choosing those slots. . .	83
5.2	Example of packet corruption by simultaneous transmissions in the carrier sense range. Circles around each node denote their transmission ranges. All nodes are in the carrier sense range of each other.	86
5.3	PDR values obtained using the derived analysis and ns-2 simulations for $N_{TX} = 5$. Confidence intervals of 95% are shown for the simulation results.	91
5.4	PDR values obtained using the derived analysis for different S_{adv} and L_R values for $N_{TX} = 5$. The optimal point for energy minimization is shown for a given PDR constraint of at least 98%. . .	93
5.5	Single hop scenario. Performance as a function of the number of sources for ATMA, ADV-MAC, T-MAC, S-MAC and TRAMA. . .	98
5.6	Single hop scenario. Average energy per packet as a function of data reservation length.	100
5.7	Multi-hop scenario. Performance as a function of the number of sources for ATMA, ADV-MAC, T-MAC, S-MAC, TRAMA and Z-MAC (energy not included for Z-MAC).	102
5.8	Multi-hop scenario. Performance as a function of mean burst length for ATMA, ADV-MAC, T-MAC, S-MAC, TRAMA and Z-MAC (energy not shown for Z-MAC).	105
6.1	SORA programmable radio control board.	111

6.2	General architecture of SORA.	112
6.3	General architecture of SORA UMX.	113
6.4	General structure of the ATMA UMX Application. The figure assumes that all nodes are synchronized. Also, the figure is not drawn to scale.	117
6.5	Setup of the SORA experiments.	119
6.6	Comparison of PDR and latency values obtained from hardware experiments and software simulations.	121
A.1	Example of slot selection in ADV period: Four nodes contend of which three get to transmit as they select slots within $S_U = 9$. The three nodes select two distinct slots. Nodes 1 and 4 select the same slot and result in a collision. Node 3 selects a slot after S_U and hence has to defer.	140

1 Introduction

Wireless sensor networks (WSNs) have great potential to support monitoring and surveillance applications for environmental, industrial and military purposes. However, several challenges such as the limited energy supply, storage and computation capability of the sensors, the high cost of the sensor nodes, as well as the scalability and the need for QoS support have limited the large scale adoption of these networks. Recent advances in semiconductor and communication technology and the development of System-on-a-Chip (SoC) technology have led to the development of low cost sensors with higher storage and computational capabilities. These have mitigated some of the problems of WSNs, but the limitation on a sensor's energy resources is still a major problem. This dissertation addresses the important issue of limited energy in WSNs by developing techniques and strategies to conserve energy and increase network lifetime by using advertisements for data packets prior to their transmission.

1.1 Wireless Sensor Networks

A wireless sensor network is a network of sensor nodes deployed across an area of interest for the purpose of monitoring or surveillance. The sensor nodes are autonomous devices that are generally equipped with one or more sensors, a micro-

controller for processing gathered data, a transceiver for transmitting and receiving data, and batteries to power the device. These sensors can gather data on environmental or physical variables such as temperature, pressure, movements, etc. Data gathered by a sensor node is passed co-operatively through the network using different nodes to reach one or more sink nodes, where the data are processed and analyzed. where the data is processed and analyzed.

Development of wireless sensor networks was primarily motivated by their need for military surveillance [15]. With the availability of low cost sensors, these networks are no longer limited to military applications but are used in a wide array of applications including habitat monitoring [30], industrial process monitoring [7], traffic control [26], [53], health care [33], etc. Some of the major applications of WSNs are:

1. **Military Applications:** Military missions often involve high risk to human personnel. Thus, unmanned surveillance missions using wireless sensor networks have wide applications for military purposes such as surveillance, reconnaissance of opposing forces, targeting, damage assessment, etc. WSNs developed for military purposes should be rapidly deployed in an ad hoc fashion such as by an aircraft. They should also be energy-efficient, fault tolerant, disposable and support network dynamics. Destruction of a few nodes by enemy forces should not hamper the operation of such networks. The Sensor Information Technology (SensIT) [22] program organized by the Defense Advanced Research Projects Agency (DARPA) explored two important aspects of military WSNs – dynamic networking of sensors and information processing and extraction from such networks.
2. **Habitat Monitoring:** Monitoring plant and animal habitats on a long-term basis is widely employed by researchers in Life Sciences. However, human presence in such monitoring often causes disturbances in plant and animal

conditions, increases stress, reduces breeding successes, etc [5]. Wireless sensor networks provide a non-invasive and economical method of long-term monitoring of habitats. Such a network was used to monitor the Storm Petrel seabirds in the Great Duck Island in Maine [30]. Wireless sensors were used to measure temperature, pressure, humidity and other conditions of the birds' burrows. Another system called ZebraNet was used to track zebra and other animals in Kenya [20].

3. **Environment Monitoring:** Wireless sensor networks can be used in a wide range of environmental monitoring applications such as forest fire monitoring [52], air pollution monitoring [8], greenhouse gas monitoring [4], etc. WSNs to monitor dangerous gases such as CO, NO₂, and CH₄ have already been deployed in some cities (London and Brisbane) [8].
4. **Agriculture:** Wireless sensors may be deployed across large areas of crop fields and can monitor different parameters like moisture and fertilizer content of soil, temperature and PH level [48]. This can automate the processes of irrigation [32], application of fertilizer and pesticides, among others, minimizing human intervention and maximizing yield.
5. **Industrial Monitoring:** Industrial machineries need condition-based maintenance. Wired infrastructure for such maintenance is costly due to the cost of wiring and the inaccessible locations, such as rotating machinery. Wireless sensors are beneficial in such cases, providing greater accessibility, improved monitoring and maintenance at lower costs [14], [41]. WSNs are also widely employed in industry for product monitoring [29] as well as quality control.
6. **Health Monitoring:** Wireless personal area or body networks [33] may revolutionize the way we monitor health conditions by providing a non-invasive, inexpensive, continuous and ambulatory health monitoring. Patients wear small body sensors that monitor the patient's bio-signals such as heart rate,

and the collected data are transmitted over a hand held device. Alarms and bio-signals may be transmitted over the Internet to a health professional for real-time diagnosis [18].

1.2 Challenges of Wireless Sensor Networks

Some of the major challenges that prevent the wide spread adoption of WSNs include the following:

1. **Energy Constraint:** Wireless sensor nodes are battery-powered and often deployed in remote and inhospitable locations [50]. As such, battery replacement or any other human intervention is either not possible or extremely difficult. Therefore, these nodes are required to function for months or years at a time on the same power source to maintain the application Quality of Service (QoS). As a result, energy conservation is of the utmost importance in WSNs, and much research has been done on the development of energy efficient protocols and hardware for WSNs.
2. **Fault Tolerance:** Oftentimes a sensor node may be destroyed or stop functioning, such as when a sensor node is destroyed in a forest fire or by the enemy in a battlefield. The remaining nodes must adapt dynamically in real time and convey the data to the base stations or sinks. Thus, WSN protocols for the MAC and routing layers must have a certain level of robustness [21].
3. **Computation Capability:** Sensor nodes are small devices with very limited memory and processing power [50]. Thus, oftentimes large scale processing is not possible in sensor nodes, and the data must be transmitted to a base station to be processed. However with the advancement of semiconductor technology, this drawback has been greatly reduced.

4. Security: WSNs are lightweight networks with limits on the transmitting data rate and capacity. Thus, conventional security measures such as private keys are not readily applicable to such networks, as these may increase the network overhead and in turn decrease the network lifetime. However, security is an important requirement in applications such as surveillance. Thus, another area of research in WSNs is providing security and privacy [24], [36].
5. WSNs face some other challenges as well, like the scalability problem [12] and latency issues.

1.3 Motivations and Contributions

Most wireless sensor networks, as discussed so far, have very strict energy requirements. Much research has been done to use energy more efficiently and prolong the network lifetime. Energy conservation may be done at different layers of the protocol stack such as the application layer, the MAC layer or the network layer. However, it is easier to get direct access to the radio in the MAC layer, and thus, much of the research in energy conservation for WSNs has focused on the design of energy efficient MAC protocols.

The major cause of energy waste in conventional MAC protocols is idle listening. In MAC protocols such as IEEE 802.11 or CDMA, the nodes have to keep listening to the channel because they do not know when they might receive a message. Measurements have shown that energy consumed in idle listening is comparable to that consumed in receiving or transmitting. For example, the idle:receive:send power ratios on MicaZ motes are 1.13:1.13:1 [1] while the same ratios for Tmote Sky motes are 1.11:1.11:1 [3]. In most sensor networks, messages are sent in bursts when an event is sensed. At all other times, most nodes have

no data to send. If a conventional MAC protocol is used in sensor networks, a very large part of the energy will be wasted in idle listening.

The second cause of energy waste is collisions. If a node receives two different packets that coincide partially, then the packets are corrupted. As a result, these packets are discarded and energy is wasted in the process. The third cause is overhearing. This happens when a node receives a packet destined for another node. The last cause of energy waste is protocol overhead. Most protocols require the exchange of control packets. Since control packets contain no application data, we may consider the energy spent in exchanging these packets as wasted in overhead. However, idle listening wastes the most energy in the majority of MAC protocols.

The aim of this thesis is to understand the nature of this energy waste and to use Advertisements for available data to minimize this energy loss, optimize network lifetime, and maintain application QoS. Data advertisements are a way of informing the intended receivers that data is waiting for them before the actual data transmission begins. This allows non-receivers to turn off their radios at these times and save energy. Advertisements are different from the Request-to-Transmit packet mechanism [19], which is used to prevent collisions in the data period itself. My research investigates the use of Advertisements under both contention based and TDMA based scenarios and utilizes the bursty nature of network traffic to minimize energy consumption and improve performance. The contributions of my research include:

1. Studying the working principle and performance of energy-saving MAC protocols such as Sensor-MAC (S-MAC) [50] and Timeout-MAC (T-MAC) [46], understanding their benefits and limitations under different network conditions, and identifying places where further energy savings can be achieved without sacrificing performance.

2. Proposing Advertisement-MAC (ADV-MAC), a new MAC protocol that uses Advertisement for Data before the data period to selectively keep nodes awake or put them to sleep depending on the data availability, saving additional energy in the process. Using this approach, I develop a multicasting technique, which is absent in S-MAC and T-MAC. I compare the energy consumption, throughput and latency performances of ADV-MAC with S-MAC and T-MAC under various traffic conditions in single hop and multi-hop scenarios with extensive simulations.
3. Proposing an improved method of contention in the Advertisement period. I propose an analytical model for computing the energy and packet delivery ratio of the ADV-MAC protocol.
4. Using the analytical model to determine the optimal duration of the Advertisement period for the optimal packet delivery ratio and the lowest energy consumption. I compare the improved and optimized ADV-MAC protocol with the original ADV-MAC as well as S-MAC and T-MAC under different traffic loads in single hop and multi-hop scenarios with extensive simulations.
5. Removing the second contention of the data period in the ADV-MAC protocol and converting it into a TDMA based MAC protocol called Advertisement-based Time-division Multiple Access (ATMA) and saving additional energy under bursty traffic conditions.
6. Proposing an analytical model to determine the optimal Advertisement period of ATMA. I compare the energy consumption, packet delivery ratio and latency performances of ATMA with a similar TDMA based MAC protocol TRAMA [31], the contention based protocols ADV-MAC, S-MAC and T-MAC and hybrid protocols Z-MAC [49] and IEEE 802.15.4 [2] with extensive simulations and qualitative analysis.

7. Implementing the ATMA protocol in hardware. Simulations alone cannot reflect many of the challenges such as synchronization, clock-drift, etc., faced by real implementations of MAC protocols. Such issues may cripple a protocol that otherwise performs very well in software simulations. Hence, to validate my research, I conclude with a hardware implementation of the ATMA protocol on SORA (Software Radio) [44], developed by Microsoft Research Asia, and compare the experimental results with those obtained from simulations.

1.4 Dissertation Organization

This thesis dissertation is organized as follows: Chapter 2 presents a literature survey on related work on energy saving MAC protocols. Chapter 3 presents ADV-MAC and compares its performance with S-MAC and T-MAC. Chapter 4 presents an improved Advertisement contention method for ADV-MAC and an analytical model for the energy consumption and packet delivery ratio of ADV-MAC and uses the model to determine an optimal duration of the Advertisement period. I also compare the improved ADV-MAC with the original protocol along with S-MAC and T-MAC. Chapter 5 presents the ATMA protocol along with an analytical model to determine the optimal Advertisement period and a detailed performance comparison with TRAMA, S-MAC, T-MAC, ADV-MAC, Z-MAC and IEEE 802.15.4. Chapter 6 presents the hardware implementation of the ATMA protocol on the programmable software-defined radios called SORA (Software Radios) developed by Microsoft Asia and a comparison between the experimental and simulation results. Chapter 7 finally concludes the dissertation.

2 Related Work

Several medium access control (MAC) protocols have been proposed for wireless sensor networks (WSNs). Since energy is a big constraint for WSNs, many of these MAC protocols have been designed with the objective of minimizing energy consumption, often at the cost of other performance metrics such as latency and packet delivery ratio. In this chapter, we shall study the evolution of energy saving MAC protocols for WSNs. Most of these MAC protocols can be broadly categorized as TDMA (Time Division Multiple Access) based MAC protocols and contention-based MAC protocols.

2.1 TDMA Based MAC Protocols

TDMA (Time Division Multiple Access) protocols are based on reservations and scheduling [27], [31], [47]. In this type of protocol, every node will eventually be able to transmit its data. This is accomplished by reserving slots for the nodes. Thus, TDMA protocols prevent collisions and have bounds on end-to-end delay. These protocols are energy efficient when the network is highly loaded and all slots are used. However, TDMA-based protocols suffer from synchronization issues. The nodes in such protocols need to be tightly synchronized, forming communication clusters. It is not easy to maintain clock synchronization among

the nodes, especially in a large network. Also, when new nodes join the network or nodes leave the network, it is not easy to dynamically change the frame length and slot assignment in a cluster. Furthermore, developing an efficient schedule with a high degree of channel reuse is difficult.

2.1.1 Low-Energy Adaptive Clustering Hierarchy (LEACH)

LEACH or Low-Energy Adaptive Clustering Hierarchy [16] is a hierarchical, cluster-based, TDMA protocol. LEACH has three types of sensor nodes: base stations, cluster-heads and ordinary nodes, as shown in Fig. 2.1. The sensor nodes in LEACH organize themselves into local clusters with one of the nodes acting as the cluster-head. Only the cluster-head can communicate with the base station. Within a cluster, nodes send data to their cluster-head, which aggregates the data collected from different nodes to reduce data transmissions and save energy. The power level of transmissions between nodes and their cluster head is much lower than that between a cluster head and the base station. Cluster heads create slotted TDMA transmission schedules and broadcast them to all nodes in their respective clusters. Nodes turn their radios off other than their own time slots, saving energy. The energy consumption of cluster-heads is higher than that of normal nodes because of the data aggregation and communication to the base stations. Therefore, cluster-heads are re-elected periodically to distribute the energy load among all nodes, with each node using a stochastic algorithm to determine if it can become the cluster-head.

Although the stochastic cluster-head election method rotates the cluster-heads among the nodes, the algorithm does not take into consideration the actual energy levels of the nodes. Even if the initial network is homogeneous, this strategy may lead to uneven energy consumption, lowering network lifetime. LEACH also assumes that all nodes can communicate directly with the base station. In a large network, this may lead to cluster-heads transmitting with a very high power

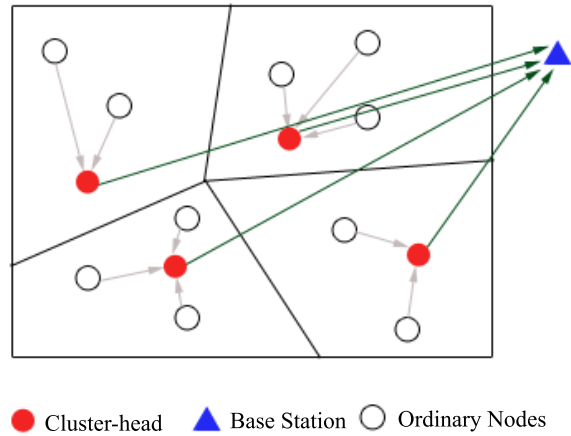


Figure 2.1: Clusters in LEACH.

level, causing them to die out soon and reduce network lifetime. Also, each node needs to know the optimum average number of clusters for the entire network. This parameter depends on global topology information and may not be readily available.

2.1.2 Power Aware Clustered TDMA (PACT)

Power Aware Clustered TDMA or PACT [35] uses passive clustering like LEACH to create a backbone network with cluster-heads and gateway nodes for communication, as shown in Fig. 2.2. However, unlike LEACH, the election of a cluster-head is based directly on a node's power level. Gateway nodes allow communication between clusters. Cluster-heads collect data from cluster members and transfer the data to the sink. Nodes that are members of two or more clusters can be elected as gateway nodes. The roles of gateways and cluster-heads are interchangeable and depend on the power levels of the nodes.

PACT uses TDMA super-frames composed of control mini-slots followed by longer data slots. In the control slots, nodes broadcast their destination addresses

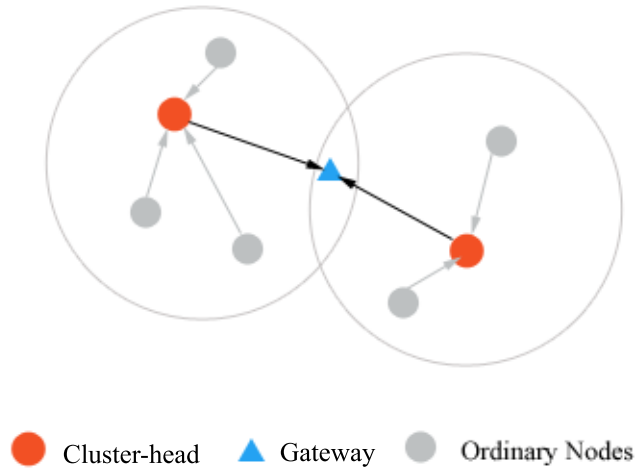


Figure 2.2: Clusters in PACT.

for the following data slots. Control messages are also used to broadcast their current cluster role status. Nodes only stay awake in their respective data slots, i.e., slots in which they send or receive data, and turn off their transceivers in other slots, saving energy.

Although nodes only stay awake in their respective data slots, all nodes must stay awake for n control slots to determine the data schedule, where n is the number of nodes in a cluster. Thus, control traffic overhead may be relatively high in larger networks. Also, the small control mini-slots require precise synchronization, which may pose problems for large and dynamic networks.

2.1.3 Self-Stabilizing TDMA (SS-TDMA)

Self-Stabilizing TDMA or SS-TDMA [16] is a TDMA protocol designed for broadcast, local gossip and convergecast applications [16]. The protocol can start from any arbitrary state and recover to states from where collision-free communication can be achieved among sensors. All traffic is scheduled in a fixed sequence of

rounds (e.g., north, south, east, west) to guarantee collision-free transmissions. SS-TDMA uses the relationship between the interference range and the communication range of nodes to get an estimate of the number of nodes within interference range that should not have the same slot number. A sensor remains active only in its allotted time slots. In the remaining slots, the sensor turns off its radio to save energy. However, application of SS-TDMA is limited because it is restricted to only grid-like topologies.

2.1.4 Node-Activation Multiple Access (NAMA)

NAMA or Node-Activation Multiple Access [9] is based on neighborhood-aware contention resolution (NCR) [9] and node activation. It is a distributed time division multiplexing scheme. In NAMA, a hash function is implemented at each node. This hash function takes a distinctive string as input and generates a random priority for each node. The distinctive input is the concatenation of a node's ID and the time slot number, and, as such, changes with different slots, giving different priorities. If a node has the highest priority in its two-hop neighborhood, it is allowed to transmit and the remaining nodes stay silent. Thus, NCR allows each node to elect winners for channel access deterministically.

Although NAMA provides contention free channel access and eliminates control overhead (except the two hop neighborhood information), it has several problems. There is no bound to the delay for channel access because of the random priority. A node may keep on generating lower priorities and never gain access to the channel. When a node wins a slot but has no data to send, the channel bandwidth is wasted along with the loss of energy as all nodes must stay awake in case data is sent. Nodes with traffic and low priorities, on the other hand, may suffer from starvation.

2.1.5 Lightweight Medium Access Protocol (L-MAC)

Lightweight Medium Access Protocol or L-MAC [17] uses a TDMA mechanism to provide a collision-free communication environment for the nodes. Due to the scheduled TDMA access, each node owns a slot in a fixed-length frame. A node may use its slot to send data to a neighbor in a frame. A node broadcasts a list of all occupied slots in its one hop neighborhood in its header. This allows new nodes to select unique collision free slots in their two hop neighborhood. The main drawback of L-MAC is that idle-listening overhead is substantial, as nodes must listen to the control sections of all slots in a frame to allow nodes to join the network or to receive or transmit data. Also, L-MAC is mostly suited for low density networks.

2.1.6 TRaffic-Adaptive Medium Access (TRAMA)

TRAMA or TRAffic-Adaptive Medium Access [31] is a distributed TDMA-based protocol. In TRAMA, time is divided into random access and scheduled access slots. Nodes gather neighborhood information by exchanging small signaling packets during the random access period and build their two-hop neighborhood. In [31], the ratio of the length of the random access and the scheduled access periods is set to 72:10000. The scheduled access period is slotted. Each slot is called a transmission slot and is owned by a node in each neighborhood. The owner of a slot is determined by a hash function that uses node IDs and the slot number as its parameter. Within the scheduled access period, every node has to send its schedule once in every *schedule interval*. In [31], the ratio of the length of one *schedule interval* and one scheduled access period is set to 1:10. At the beginning of each schedule interval, a node will calculate the slots it wins. The last slot a node wins within the schedule interval is reserved for transmitting the node's schedule for the next *schedule interval*. The rest are used for data exchange, if any.

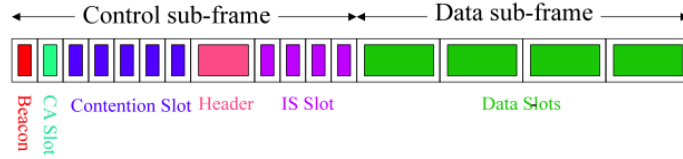


Figure 2.3: MH-TRACE frame structure.

Thus when a node transmits its schedule, all nodes in its neighborhood must be awake. Energy consumption is thus heavily dependent on the number of one-hop neighbors, i.e., the network density.

2.1.7 MH-TRACE

Multihop Time Reservation using Adaptive Control for Energy efficiency (MH-TRACE) is a cluster-based medium access control (MAC) protocol [45]. In the MH-TRACE protocol, the network is dynamically partitioned into clusters, which are maintained by cluster heads. Time is divided into super-frames consisting of several frames, and each cluster chooses a frame during which nodes in the cluster can transmit data, as shown in Fig. 2.4. Each frame consists of a contention sub-frame, an information summarization sub-frame, and a data sub-frame, as shown in Fig. 2.3. Nodes transmit their initial channel access requests to a cluster head in the contention sub-frame. As long as a node uses its reserved data slot, its channel access is renewed in subsequent frames. Nodes that are granted channel access through the transmission schedule transmitted by the cluster head transmit a summarization packet prior to actual data transmission in the information summarization (IS) sub-frame. Thus each node knows the future data transmissions in its receive range by listening to the information summarization packets. Nodes save energy by entering the sleep mode whenever they do not want to be involved with a packet transmission or reception.

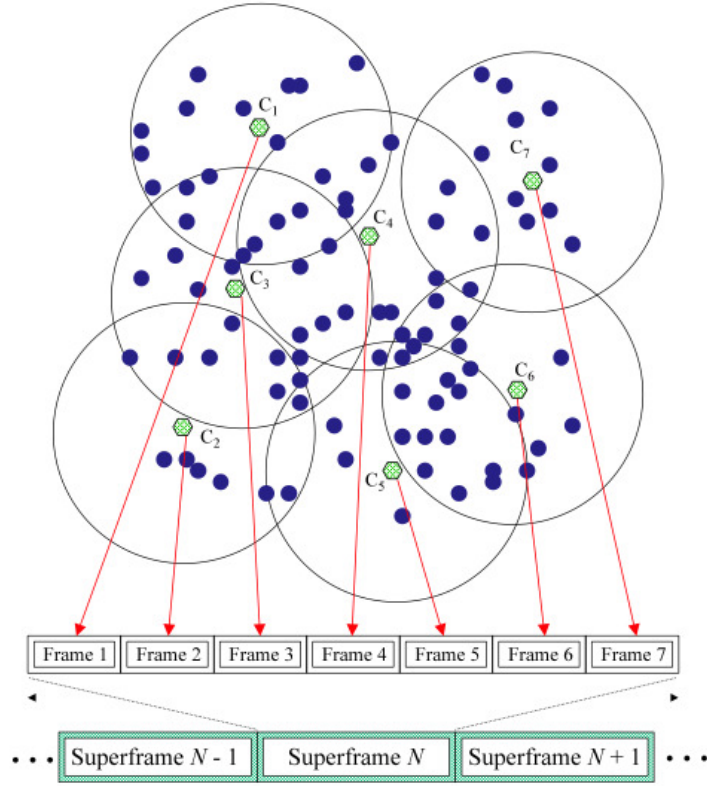


Figure 2.4: A snapshot of MH-TRACE clustering and medium access for a portion of an actual distribution of mobile nodes. Nodes C_1 - C_7 are cluster-head nodes. Reprinted with permission of the authors from [45].

2.2 Contention Based MAC Protocols

Contention-based protocols are widely employed because of their simplicity, robustness and flexibility. These protocols require little or no clock synchronization and no global topology information. The IEEE 802.11 distributed coordination function (DCF) [19] is an example of contention-based protocol. It is widely used because of its simplicity and robustness to the hidden terminal problem. However, because of idle listening, the energy consumption using this MAC is very high, as nodes are often in the idle mode. Additionally, at high loads, contention-based MAC protocols perform poorly because of the high number of collisions. PAMAS [42] made an improvement in energy savings by turning nodes off for the duration of a packet transmission if they are not the intended destination of the packet. However, PAMAS does not reduce idle listening.

2.2.1 Sensor-MAC (S-MAC)

Certain MAC protocols introduce a sleep-listen schedule into contention-based (CSMA) protocols to save the energy that is wasted in idle listening. The nodes in such protocols go to a low power sleep state whenever possible to save energy. A well known protocol in this category is Sensor-MAC (S-MAC) [50]. S-MAC was specifically designed for wireless sensor networks. S-MAC divides time into frames, each frame having an active and a sleeping part. Nodes can communicate with their neighbors using RTS/CTS/DATA/ACK sequences during the active part. During the sleeping part, nodes turn off their radios and sleep to save energy. The duty cycle, i.e., the proportion of active time within a frame, is set based on factors such as network density and message rate. Its value is decided before the network is set up and is static. Also, only one node can transmit in each frame in a neighborhood.

Although S-MAC reduces the idle listening time, a solution with a fixed duty

cycle is not optimal. S-MAC basically trades energy for throughput and latency. While a low duty cycle reduces idle listening time, it results in high latency and low throughput in medium to high traffic conditions, as only one data packet transmission can occur in each frame. On the other hand, if the duty cycle is high, the throughput and latency performances improve at the expense of reduced energy savings.

2.2.2 Improvements on S-MAC

Traffic adaptive MAC or TA-MAC [13] and Dynamic Sensor MAC or DSMAC [25] are two proposed modifications of S-MAC. TA-MAC is S-MAC with an adaptive contention window algorithm. TA-MAC uses the back-off algorithm of IEEE 802.11 and the fast collision resolution algorithm from [23] to determine the value of the contention window for the current traffic state. This improves energy consumption, delay, and packet delivery ratio.

The DSMAC protocol adjusts the duty cycle of S-MAC based on the perceived latency of nodes to cope with the high latency associated with high packet inter-arrival rates. DSMAC improves the latency of S-MAC, the trade-off being higher energy consumption.

2.2.3 Timeout-MAC (T-MAC)

S-MAC is not suitable for variable traffic loads because of its static duty cycle. Timeout-MAC or T-MAC [46] was proposed to improve the poor performance of S-MAC under variable loads. In T-MAC, the active period ends when no activation event has occurred in the channel for a time threshold T_A . An activation event can be any activity in the channel such as the firing of a periodic frame timer, the reception of any data on the radio, the sensing of a collision, etc. Nodes will keep on renewing their timeout values whenever an activation event occurs. When none

of these events occur for a duration of a timeout period, the nodes go to sleep. The TA timeouts make the active period in T-MAC adaptive to variable traffic loads.

However, whenever an activation event occurs, all nodes that hear the event renew their TA timer even if they are not a part of the transmission. As a result, nodes still end up wasting valuable energy. Optionally, nodes go to sleep after overhearing an RTS or CTS destined for another node, which is called overhearing avoidance. T-MAC also suffers from the *early sleep problem* [46]. A node may go to sleep after a timeout even if it has data to send or to receive. For example, a node having data to send may not get a response from a neighbor after sending the RTS if that neighbor happens to be in the range of an ongoing transmission. Thus, the timer at the sender will expire and it will go to sleep without transmitting data. This will increase latency and decrease packet delivery ratio.

2.3 Low Power Protocols

Another group of MAC protocols can be classified as low power listening (LPL) MAC protocols. These protocols are asynchronous and rely on preamble sampling for data transmission. Examples of such protocols are X-MAC [10] and B-MAC [37]. Fig. 2.5 shows the asynchronous operation principle of LPL protocols.

In LPL protocols, nodes have a very small duty cycle and each node maintains its own unsynchronized sleep schedule. A node having data to send first transmits an extended preamble. A node wakes up at the beginning of its duty cycle to check if any data transmission is going on. If the channel is idle, the node goes back to sleep. If a preamble is detected, the node stays awake for the remainder of the preamble to determine if it is the intended receiver. If not, after the end of the preamble, the node goes to sleep. Otherwise, it stays awake to receive the data.

Although these protocols are very energy efficient, they are mostly suited for

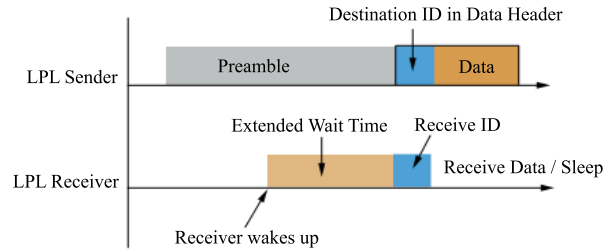


Figure 2.5: Operation principle of LPL protocols.

very low traffic loads. At high and variable traffic loads, because of the long preambles, throughput decreases and latency increases. The performance of such protocols reduces significantly when the actual neighborhood or traffic is different from the ideal model, especially when traffic rates vary greatly [51].

2.4 Hybrid MAC Protocols

Hybrid MAC protocols let nodes access the medium with a combination of contention based and contention-free access depending on the traffic load as well as protocol configuration. IEEE 802.15.4 [2] and Z-MAC [49] are two well known hybrid MAC protocols.

2.4.1 IEEE 802.15.4

IEEE 802.15.4 [2] is the IEEE standard for low power wireless communication networks. IEEE 802.15.4 is a cluster based protocol that divides the network into clusters controlled by cluster-heads. Energy saving is achieved by duty cycling, as shown in the super-frame structure of 802.15.4 (Fig. 2.6).

This protocol can work in either synchronized or non-synchronized modes of which we consider the synchronized mode as it provides a better performance in

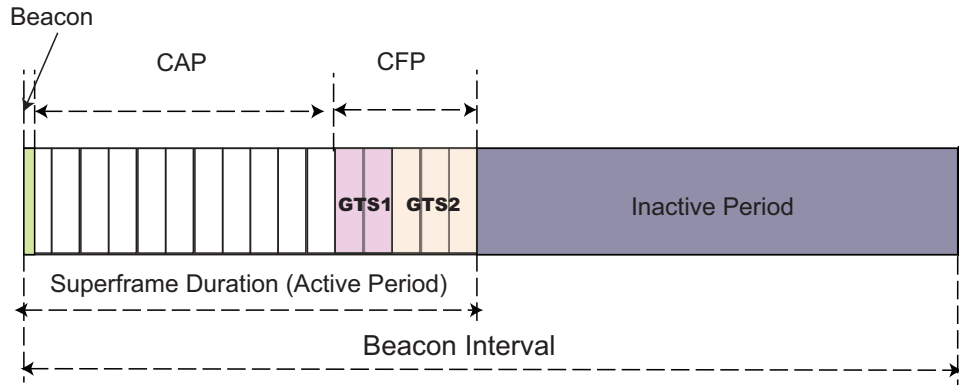


Figure 2.6: Super-frame structure in IEEE 802.15.4

terms of PDR and throughput. The nodes are synchronized by a beacon packet that contains the timing information for nodes in the cluster. A beacon is sent by the coordinators at the beginning of the super-frames, and nodes of this coordinator update their timers each time a beacon is received.

The super-frame is divided into 16 slots of equal duration. Within the super-frame, nodes may access the medium via a Contention Access Period (CAP) or through a Contention Free Period (CFP). The CAP starts immediately after the beacon transmission is over. Within the CAP, medium access is acquired via standard contention access methods. Nodes that must transmit regularly can also reserve slots in the super-frame by sending requests to the coordinator during the CAP. If such requests are granted, the coordinator includes the relevant information, such as the number of slots allocated and the beginning slot, in the next beacon. These slots are termed as Guaranteed Time Slots or GTS. The GTS are allocated from the end of the super-frame and cannot be more than 7 slots in total.

One of the strong points of 802.15.4 is that it is very flexible, allowing for both contention based and TDMA channel access, thus enabling nodes with different requirements to access the channel according to their specific needs. Also, IEEE 802.15.4 is standardized by the IEEE, unlike most other protocols, and hence it

is more likely to be adopted for industrial and commercial purposes.

However, the protocol is not without its own drawbacks. The very flexible nature of 802.15.4, which is an advantage of the protocol, adds several other disadvantages. The nodes belonging to a cluster in 802.15.4 cannot communicate directly to other nodes in the network. This communication must proceed through the cluster-head. Depending on whether the destination node is in the same cluster or not, the cluster-head will forward the message to the node or to the cluster-head of the node. This adds to the delay compared to distributed protocols where nodes communicate directly if they are in communication range.

Another disadvantage of 802.15.4 is that it can only allocate the last 7 slots for TDMA access because it needs to provide for contention based channel access in all super-frames. Thus even if nodes in a cluster produce data at high rates, which is best addressed by allocating all of the 16 slots to TDMA access, 802.15.4 can only allocate a maximum of 7 slots. This in turn reduces throughput and also increases latency.

Since all communication must be done through the cluster-heads, they also end up depleting their energy faster than the other nodes in the cluster. This problem of non-uniform energy consumption may lead to a disconnected network.

2.4.2 Z-MAC

Z-MAC [49] is a hybrid protocol in which nodes access the medium with a combination of contention based and contention-free access. Time in Z-MAC is divided into slots. Each of these slots are assigned to nodes in such a way that no two nodes within a two-hop communication neighborhood are assigned the same slot. This assignment is done using DRAND [40], a distributed TDMA slot assignment algorithm. The number of nodes in the two hop neighborhood of a node determines the period or the interval (in number of slots) at which the node owns

a slot. In Z-MAC, a node can operate in either of the modes: low contention level (LCL) or high contention level (HCL). A node enters HCL mode only when it receives one or more explicit contention notifications (ECN) from a two-hop neighbor within the last period. Otherwise, by default, the node is in LCL mode. When a node experiences high contention from one of its neighbors, it sends an ECN message to that node.

Although a slot may belong to a particular node, channel access within a slot begins with a contention access mechanism. In LCL mode, any node can compete to transmit in a particular slot even if it does not own that slot. In HCL mode, only the owners and their one-hop neighbors are allowed to compete for the channel access of the particular slot. Owners have higher priority over non-owners in both modes of operation. If a slot does not have an owner or its owner has no data to send, non-owners can steal the slot in that case. Z-MAC uses the same methods of backoff, Clear Channel Assessment (CCA) and Low Power Listening (LPL) as that of B-MAC [37].

Since Z-MAC uses the LPL interface of B-MAC, it uses preamble sampling, i.e., a node sends a preamble before sending the data. Receivers must wake up at intervals smaller than the preamble to be able to receive packets. Thus, all nodes in a two hop neighborhood must wake up at intervals smaller than the preamble in all slots so that they can receive data that may be destined for them. In most distributed TDMA protocols, senders and receivers know the slots in which they are supposed to exchange data and hence need not wake up in all slots. Also there is no contention to send data in a slot in such protocols. The combination of these two factors should lead to higher power consumption for Z-MAC compared to distributed TDMA protocols.

When a new node joins the network in Z-MAC, the DRAND algorithm is run locally to compute the new period for the nodes in that neighborhood, and this information is then propagated throughout the network. This process requires

significant energy expenditure.

2.5 Summary

As seen from this discussion, different MAC protocols aim to improve one performance parameter at the cost of another. Many of these protocols aim to conserve energy, but the trade-off is a lower packet delivery ratio, higher delay or restriction to a specific topology. In my dissertation, I try to improve the energy efficiency of MAC protocols as well as their overall performance by applying advertisement techniques and by investigating the improvements and limitations under different network scenarios.

3 ADV-MAC: Advertisement-based MAC Protocol for Wireless Sensor Networks

Several Medium Access Control (MAC) protocols have been proposed for wireless sensor networks with the objective of minimizing energy consumption. For example, Sensor-MAC (S-MAC) was proposed to reduce energy consumption by introducing a duty cycle. The fixed duty cycle of S-MAC results in energy loss under variable traffic load due to idle listening, along with higher latency and lower throughput compared to an adaptive duty cycle. Timeout-MAC (T-MAC) introduced such an adaptive duty cycle to handle variable traffic loads. However, nodes in T-MAC that do not take part in data exchange waste energy because of continuous renewal of their timeout values.

In this chapter, we propose Advertisement MAC (ADV-MAC), a MAC protocol for wireless sensor networks that eliminates this energy wasted in idle listening by introducing the concept of advertising for data contention. ADV-MAC minimizes the energy lost in idle listening while maintaining an adaptive duty cycle to handle variable loads. Additionally, ADV-MAC introduces an energy efficient

data-centric MAC-level multicasting scheme where a node can send data to a subset of its neighbors. Such MAC-level multicasting is not possible in S-MAC and T-MAC, which must resort to broadcasting. We provide detailed comparisons of the ADV-MAC protocol with S-MAC and T-MAC through extensive simulations. The simulation results show that ADV-MAC efficiently handles variable load situations and provides substantial gains over S-MAC and T-MAC in terms of energy (reduction of up to 45%) while faring as well as T-MAC in terms of throughput and latency.

The rest of this chapter is organized as follows. In Section 3.1, we describe the design of the ADV-MAC protocol and compare ADV-MAC to both S-MAC and T-MAC qualitatively. In Section 3.2, we describe our simulation setup, followed by a detailed discussion of the simulation results. In Section 3.3, we summarize the chapter.

3.1 ADV-MAC Design Overview

Reducing energy consumption is the main objective of ADV-MAC. The protocol aims to reduce energy wasted in idle listening as much as possible. In ADV-MAC, all nodes that are not part of any current data exchanges are put to sleep, saving valuable energy. Only nodes that are part of a data transmission stay awake for the contention.

3.1.1 Basic Operation of ADV-MAC

Fig. 3.1 shows the basic principle of ADV-MAC along with a comparison to S-MAC and T-MAC. The figure shows nodes A through G, all of which are in transmission range of each other. Node A and node C have data for node B and node D, respectively. Only the active times of the nodes are shown. All

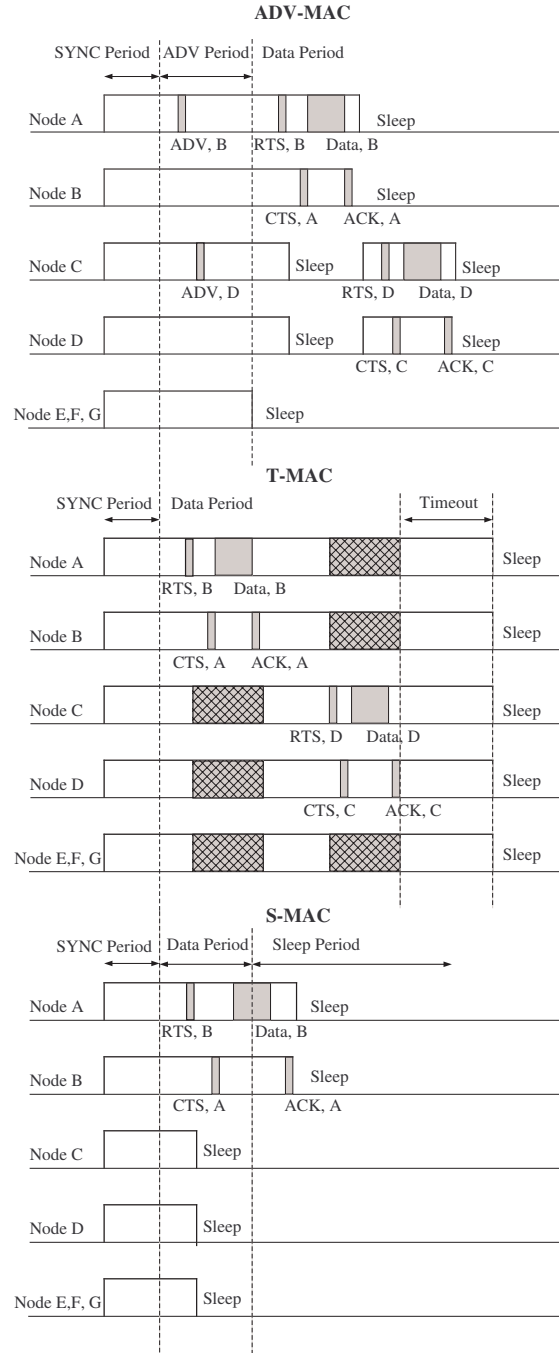


Figure 3.1: Examples of ADV-MAC, T-MAC and S-MAC communication. The letters after the packets indicate the destination nodes. If the overhearing avoidance is used by T-MAC, the nodes will be in sleep mode for the hatched areas.

three protocols start their active times with a SYNC period, which is used for synchronization and virtual clustering of the nodes as in [50]. Each frame in S-MAC consists of a fixed-length SYNC period, a fixed-length data period and a sleep period that depends on the duty cycle. T-MAC also has a fixed-length SYNC period, but the length of the data period and the length of the sleep period both depend on the local traffic conditions. While S-MAC and T-MAC begin their data period after the SYNC period, ADV-MAC defines another short period called Advertisement period (ADV period) before the data period. The advertisement period is used to transmit Advertisement packets (ADV packets), which contain the ID of the intended receivers. ADV-MAC thus has a fixed-length SYNC period and a fixed-length ADV period, followed by a variable-length data period and a variable-length sleep period. It should be noted that while the data and sleep periods are variable in both ADV-MAC and T-MAC, the total frame time is fixed. Also, unlike S-MAC, ADV-MAC does not have a fixed duty cycle. Depending on the expected traffic load, we can fix the total frame length as well as the length of the ADV period before the deployment of the network.

If a node has any data to send, it will contend in the ADV period to send its ADV packet. More than one nodes can send ADV packets in the ADV period. If the ADV packet is received by its intended receiver, that node will be aware that there is data pending for it. Thus, after the end of the ADV period, only the nodes that sent ADV packets and the intended receivers who successfully received the ADV packets will be awake for the data time. Note that no acknowledgments are sent for ADV packets. For this reason, in case of an ADV collision, the nodes whose packets collided will not know of their collision and will be awake while their intended receivers will be asleep.

After the ADV period, nodes that sent ADV packets will contend for the medium by listening to the medium for a random amount of time from the beginning of the data period and then sending an RTS packet. The node that wins the

medium completes its data exchange. Nodes can send multiple packets. Once a node has won the medium, it need not send RTS packets for all the data packets, it just sends the data packets and the receiver replies back with an ACK. Since RTS and CTS packets contain the duration of the entire exchange time, the other remaining nodes having data to send will defer until the end of the data exchange as in IEEE 802.11 [19] and go to sleep for that duration. These nodes then wake up after the data exchange is over and begin contending for the medium. The nodes whose ADV packets collided will also try to send RTS packets. However, their intended receivers will be asleep, and these nodes will eventually go to sleep after their CTS timeout. These nodes will try again in the next frame. In multi-hop networks, there may be hidden terminals. In such cases, a sender may not get back a reply from its intended receiver in one CTS timeout period and will go to sleep, even if it transmitted the ADV packet successfully. If a node fails to transmit its data packet in a frame after transmitting the RTS packet, it will not try to retransmit in the same frame but it will retry in the next frame.

As Fig. 3.1 suggests, S-MAC should have the minimum total energy consumption assuming the same frame sizes for all three protocols. However, this comes at the price of low throughput and high latency, as nodes in S-MAC can only transmit one packet in each frame. In order to improve the latency and throughput, the duty cycle must be increased for S-MAC. According to the S-MAC protocol, the active time is fixed. Thus, increasing the duty cycle means that the sleep period and hence the total frame time will be shorter. Nodes in S-MAC will wake up more frequently, leading to more frames in the same time duration. Hence, nodes end up using more energy to get better throughput and latency. For T-MAC, nodes that are not required for the data exchange stay awake and waste energy. As shown in the simulation results, ADV-MAC provides the lowest energy consumption while achieving high throughput and low latency.

Although ADV-MAC adds a new time period after the SYNC period, the

energy consumption of ADV-MAC is not greater than that of S-MAC and T-MAC even in low traffic loads. The reason is as follows. Let us consider the case of no traffic with all three protocols having the same frame length. If the data period of S-MAC, the timeout period of T-MAC and the ADV period of ADV-MAC have the same duration, the energy consumption would be the same in all cases. This is because after the SYNC period, all nodes in S-MAC will be awake for the data period, all nodes will remain awake in T-MAC until they time out and all nodes in ADV-MAC will be awake for the ADV period. In our experiments, we set the ADV period to be equal to the Timeout period given in [46] and the experimental results show that ADV-MAC gives the lowest energy consumption for that throughput and latency as compared to S-MAC and T-MAC.

ADV-MAC uses the same method for virtual clustering and loose synchronization as in both S-MAC [50] and T-MAC [46]. ADV-MAC also uses both virtual and physical carrier sense as employed by S-MAC [50] for collision avoidance.

3.1.2 Contention Resolution in ADV-MAC

A two-level contention mechanism is defined in ADV-MAC. Nodes that have data packets to send first contend to announce their receivers in the ADV period, and then in the data period, nodes contend to send their data packets. The contention mechanisms of both the ADV period as well as the data period are described in this section.

3.1.2.1 ADV period contention

The advertisement time is divided into several slots. At the beginning of the advertisement time, if a node has any data to send, it randomly picks a slot and starts to listen to the channel until its slot time arrives. If there is no ADV transmission going on when its slot time arrives, it transmits its ADV packet.

Note that other nodes may have completed their ADV transmission before this slot which enables multiple ADV transmissions in an ADV period. If the node senses a busy channel when its slot time arrives, it waits until the transmission is over, and then chooses a new random slot from the remaining slots and starts to listen to the channel again. The node will continue to do this until it is successful in transmitting the ADV packet or until the advertisement time has ended.

3.1.2.2 Data period contention

The main idea of the ADV contention method is also used in the data period contention. Let Γ be the duration of the data period and S_{data} be the duration of the contention window used before each data transmission, both in unit *slots*. Nodes that transmit in the ADV period, select a slot out of S_{data} and set their timer to the duration until their selected slot. When the timer of a node reaches zero, the node begins the data exchange by sending an RTS packet. All nodes hearing a transmission cancel their timers and choose a new slot out of S_{data} once the ongoing transmission ends. This process is repeated until all nodes finish contending or the end of the data period is reached.

Nodes whose ADV packets collided in the ADV period also contend in the data period. However, they cannot receive any CTS as their corresponding receivers are asleep, which results in the nodes timing out and to going to sleep. If RTS packets collide, the corresponding receivers will wait for the entire duration of S_{data} and not go to sleep. The senders in this case will again go to sleep after the CTS timeout.

3.1.3 Early Sleeping Problem of T-MAC

The basic T-MAC protocol suffers from the so called *early sleeping problem* [46]. Suppose node A has data for node B and node A loses contention because it hears

an RTS or CTS from another data exchange. If node B is out of the range of this transmission, it will eventually time out and go to sleep before node A can send its data. This will result in an increase in latency and a decrease in throughput values. Early sleeping can also happen if a receiver cannot reply back with a CTS because it hears an RTS/CTS exchange from another data exchange.

The ADV-MAC protocol, however, is inherently immune to the early sleeping problem. In ADV-MAC, only the nodes that are indicated as intended receivers in ADV packets remain awake in the data part of the active time. If they overhear the data exchange between other nodes (via RTS or CTS), they just go to sleep for the duration of the data exchange and wake up again to listen to the medium. If they do not hear anything, they will still stay awake for the RTS because they have prior knowledge of data waiting for them.

3.1.4 MAC Multicasting

In S-MAC and T-MAC, broadcasting takes place without any RTS/CTS mechanism, and data packets are sent directly. There may be situations where the sources broadcast different types of data and each receiving node is interested in a particular data type. For instance, there may be nodes equipped with different sensors broadcasting individual sensor measurements as separate packets, with nodes interested in only certain types of sensor data. In this type of application, a MAC level multicasting scheme can enable significant energy savings. Since nodes in S-MAC and T-MAC have no prior knowledge of which type of data is being broadcast, all nodes receive the data being broadcast even if they are not interested in that type of data, hence losing valuable energy. In ADV-MAC, ADV packets may have a field that contains the type of the data being sent. Only nodes that are interested in those types of data will stay awake in the data period. This enables efficient single-hop multicasting at the MAC level and saves a great deal of energy.

3.1.5 Energy Consumption

The energy consumed by the three protocols can be calculated approximately for simple cases. We assume that transmission, reception and idle energy consumption values are all approximately the same, as per the MicaZ and Tmote Sky energy dissipations [1][3]. Let us consider the case of N nodes in a virtual cluster, all of which are within transmission range of each other.

3.1.5.1 S-MAC

Let p be the duty cycle and t_{sim} be the simulation time. If w is the transmission, reception or idle listening power, then the total energy consumed per node in t_{sim} seconds is calculated as

$$E_{smac} = wpt_{sim}. \quad (3.1)$$

This equation does not consider any collisions or any data transmission continuing into the sleep part. In the original S-MAC protocol, nodes exchange data during the sleep time. This data exchange during the sleep time results in additional energy consumption which is not captured by (3.1). Also there are quite a few collisions in the SYNC period, which make the nodes go to sleep, hence saving energy. This is also not considered by (3.1). However, these two effects basically cancel each other, and (3.1) provides a reasonable approximation of the energy consumption. The equation remains the same for unicast and broadcast transmissions.

3.1.5.2 T-MAC

To calculate the total energy consumption in T-MAC, first let us calculate the total time spent awake by all nodes in the virtual cluster. We consider T-MAC with overhearing avoidance. Let N_s be the number of sources in the network each transmitting a packet every t_r seconds. The total time spent awake in the SYNC

Table 3.1: ADV-MAC: Notations Used

Quantity	Notation
Total Simulation Time	t_{sim}
Tx/Rx/Idle Listening Power	w
No. of Nodes	N
No. of Source Nodes	N_s
No. of Frames/Cycles in t_{sim}	N_c
No. of Packets Exchanged in t_{sim}	N_p
Duration of Sync Period	t_{sync}
Duration of Time-out Period	t_{TA}
Duration of ADV Period	t_{ADV}
Duration of Average Contention Period	\bar{t}_{cw}
Duration of Control Packet (RTS, CTS, ACK, ADV)	$t_{control}$
Duration of Data Packet	t_{data}
Duty Cycle (%)	p

period and the final timeout period by all the N nodes during the simulation period is

$$NN_c(t_{sync} + t_{TA}), \quad (3.2)$$

where N_c is the total number of cycles in the simulation time t_{sim} . The duration of SYNC and time-out periods are denoted by t_{sync} and t_{TA} . The time spent in sending and receiving the CTS and data packets is

$$2N_p(t_{control} + t_{data}), \quad (3.3)$$

where $t_{control}$ is the duration of control packets (RTS, CTS or ACK), N_p is the total number of packets exchanged within the simulation time t_{sim} and calculated as $N_s t_{sim}/t_r$, and t_{data} is the duration of a data packet. Since we consider overhearing avoidance, the factor 2 is used in (3.3) to indicate that only two nodes are awake. Without overhearing avoidance, all nodes hearing the data would be awake, and the factor would be N instead of 2. The total time spent during the course of N_c cycles in transmitting and receiving the control packets and waiting in contention period is

$$NN_p(2t_{control} + \bar{t}_{cw}), \quad (3.4)$$

where \bar{t}_{cw} is the average time spent in contention by a node.

Combining (3.2), (3.3) and (3.4), the total time spent awake by all N nodes in a unicast scenario is found to be

$$\begin{aligned} T_{tmac} = & NN_c(t_{sync} + t_{TA}) + 2N_p(t_{control} + t_{data}) \\ & + NN_p(2t_{control} + \bar{t}_{cw}). \end{aligned} \quad (3.5)$$

For a broadcast scenario, this equation becomes

$$T_{tmac} = NN_c(t_{sync} + t_{TA}) + NN_p(t_{data} + \bar{t}_{cw}). \quad (3.6)$$

If w is the transmission, reception or idle listening power, then the total energy spent per node is

$$E_{tmac} = \frac{T_{tmac}w}{N}. \quad (3.7)$$

3.1.5.3 ADV-MAC

As in T-MAC, let us calculate the total time spent awake by all nodes in the virtual cluster. Let n_a be the average number of packets transmitted in each cycle, given by N_p/N_c . The total time spent awake in the SYNC period and the advertisement period by all the N nodes during the duration of the simulation is

$$NN_c(t_{sync} + t_{ADV}). \quad (3.8)$$

where t_{ADV} is the duration of the advertisement period. The time spent in sending and receiving the data, CTS and ACK packet is given by

$$2N_p(t_{data} + 2t_{control}). \quad (3.9)$$

Again, the factor 2 is used since only two nodes are awake. The time spent during the course of N_c cycles in transmitting and receiving the control packets (RTS) and waiting in the contention period is given by

$$2N_c \sum_{i=0}^{[n_a]} (n_a - i)(t_{control} + \bar{t}_{cw}). \quad (3.10)$$

The summation term reflects the number of nodes that are waiting in contention and then receiving or transmitting RTS and CTS packets. For example, suppose there are 3 nodes waiting to transmit within a given cycle. Then, 6 nodes will wait for $(2t_{control} + \bar{t}_{cw})$ amount of time. The node that wins the contention will stay awake with its destination node, while the other 4 nodes will go to sleep. When this data transfer is complete, these 4 nodes will wake up and again spend $(2t_{control} + \bar{t}_{cw})$ amount of time. Then two nodes will stay awake and the remaining two will go to sleep and wake up when the data exchange is over. The summation is to add up all these $(2t_{control} + \bar{t}_{cw})$ periods spent awake by the nodes. Combining (3.8), (3.9) and (3.10), the total time spent awake by all N nodes together for a unicast scenario is

$$\begin{aligned} T_{advmac} = & NN_c(t_{sync} + t_{ADV}) + 2N_p(t_{data} + 2t_{control}) \\ & + 2N_c \sum_{i=0}^{[n_a]} (n_a - i)(t_{control} + \bar{t}_{cw}). \end{aligned} \quad (3.11)$$

For a broadcast scenario, the above equation becomes

$$\begin{aligned} T_{advmac} = & NN_c(t_{sync} + t_{ADV}) + \\ & (1 + \frac{N - N_s}{n_t})N_c \sum_{i=0}^{[n_a]} (n_a - i)(t_{data} + \bar{t}_{cw}). \end{aligned} \quad (3.12)$$

The summation term has a coefficient of $(1 + \frac{N-N_s}{n_t})$ instead of 2, where n_t is the number of different data types. This is because for each sender there is more than one receiver. Since each receiver selectively receives the broadcast packets, for each type of data packet, there will be $\frac{N-N_s}{n_t}$ receivers. As before, the total energy spent per node is

$$E_{advmac} = \frac{T_{advmac}w}{N}. \quad (3.13)$$

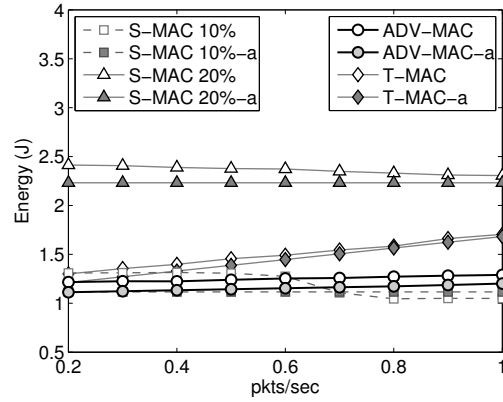
The energy consumption equations of S-MAC, T-MAC and ADV-MAC do not consider any collisions. However, if the network is not highly loaded, they provide a reasonable approximation.

3.2 ADV-MAC Performance Evaluation

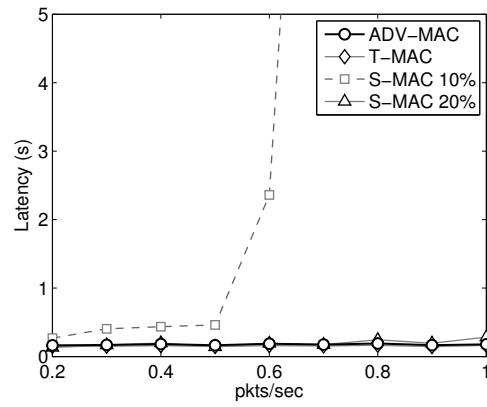
In our simulations, we compared the performance of the three protocols: S-MAC, T-MAC and ADV-MAC. We used energy consumption, throughput and latency as the three performance metrics for comparison.

3.2.1 Simulation Setup

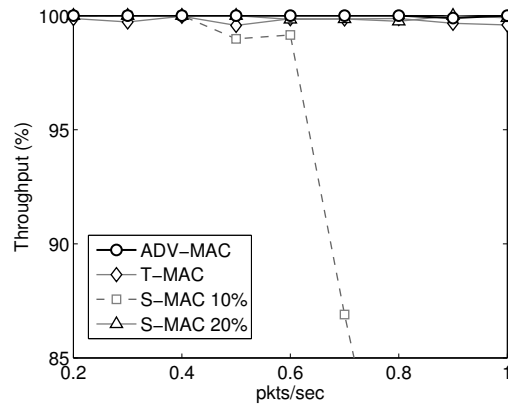
We performed all simulations in ns 2.29 [34]. The S-MAC code is included in this version of ns. We coded T-MAC and ADV-MAC in ns-2 as well. We use three different duty cycle settings (10%, 20% and 30%) for S-MAC because one fixed duty cycle is not suitable for all traffic loads investigated. Also, we limit the data rate to 1 *pkt/sec*. Although ADV-MAC and T-MAC both can handle higher traffic loads, S-MAC cannot handle such high loads. The frame time for 10% duty cycle is 238.4 *ms*, and we set this frame time for T-MAC and ADV-MAC as well. We used a duration of 15 *ms* for the time-out periods of T-MAC as in [46]. Since transmit, receive and idle listening have close energy consumption values [1][3], we set a common value of 55.8 *mW* for all three operations in accordance with



(a) Energy Consumption vs. Data Rate.



(b) Latency vs. Data Rate.



(c) Throughput vs. Data Rate.

Figure 3.2: Single hop, unicast vs. data rate: Performance comparison of ADV-MAC, T-MAC and S-MAC. The extension ‘-a’ corresponds to analytical results.

Table 3.2: ADV-MAC: Parameter Values

Parameters	Values
Total Simulation Time (t_{sim})	200 <i>s</i>
Tx / Rx / Idle Listening Power	55.8 <i>mW</i>
Transmission Rate	250 <i>Kbps</i>
Transmission, Carrier sense Range	100, 200 <i>m</i>
Duration of frames (t_{frame})	238.4 <i>ms</i>
Dur. of sync (t_{sync}), contention period (t_{cw})	8.4, 13 <i>ms</i>
Dur. of time-out period (t_{TA}), ADV period (t_{ADV})	15 <i>ms</i> , 15 <i>ms</i>
Dur. of control ($t_{control}$), data packet (t_{data})	0.9, 9.5 <i>ms</i>

[1] where the average current consumption value is 18.6 *mA* and a battery of 3 V is used. The transmission rate is 250 *kbps*, and the transmission range is 100 *m*, while the interference or carrier sense range is 200 *m*. The Advertisement period and the contention period are divided into slots of 0.1 *ms* each. All nodes in the simulations are placed randomly. We use T-MAC with over-hearing avoidance, as it is used in the simulations in [46]. The values used for the simulation parameters are summarized in Table 3.2.

We used a duration of 15 *ms* for the Advertisement period, which is the same as the TA period TA period of T-MAC. However, since the Advertisement period is an important parameter for determining the efficiency of the ADV-MAC protocol, we performed an experiment to determine whether or not 15 *ms* would be acceptable. It was found that for the highest traffic load used in the simulations, the probability of ADV packet collision is 2% which is an acceptable value. Within a frame, we consider no retransmissions. If a node transmits its RTS but fails to

transmit data packet successfully, it retries in the next frame.

3.2.2 Results

3.2.2.1 Effect of data rate on single hop, unicast scenario

In the first set of simulations, we investigate the effects of traffic load on energy consumption, latency and throughput. We consider an area of $50\text{ m} \times 50\text{ m}$ with all nodes in transmission range of each other. There are 20 nodes in the area including 5 sources. The traffic load is varied by increasing the data rate from 0.2 pkt/sec to 1 pkt/sec.

Fig. 3.2(a) shows the energy consumptions obtained from the simulations as well as the results obtained from the energy equations (3.1), (3.7) and (3.13). Figs. 3.2(b) and 3.2(c) show the corresponding latency and throughput values, respectively. As seen from the figure, ADV-MAC and S-MAC with 10% duty cycle give the best energy consumption results, with ADV-MAC having slightly lower energy consumption for lower traffic loads and 10% S-MAC having slightly lower energy consumption for higher traffic loads. However, as seen from the corresponding latency and throughput values, which are shown in Figs. 4 and 5, respectively, S-MAC with such a low duty cycle actually cannot handle the high traffic loads and gives very poor throughput and very high latency results. However, ADV-MAC presents stable latency and throughput results for all traffic loads, showing its resiliency to variable data traffic loads and high traffic loads. As data rate increases beyond 0.5 pkt/sec, 10% S-MAC is no longer sufficient because of high latency and low throughput. At high data rates, 20% duty cycle gives acceptable values of latency and throughput for S-MAC. However, the energy consumption of ADV-MAC is 44% less than the energy consumption of S-MAC with 20% duty cycle at the highest data rate.

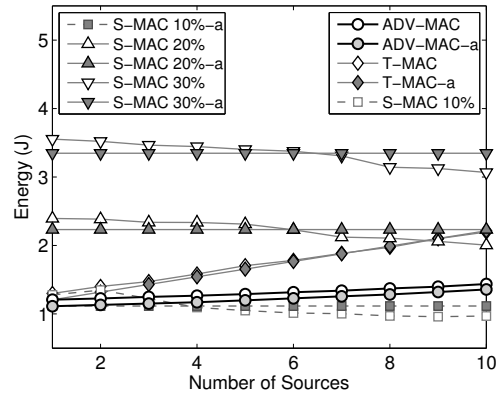
It is seen that as data rate increases, the energy consumption of ADV-MAC

increases very little, but that of T-MAC increases much faster. This happens because nodes that are not a part of a data exchange can selectively go to sleep in ADV-MAC, but all nodes in the carrier sense range must be awake in T-MAC. From Fig. 3.2(a) we can see that the energy consumption of ADV-MAC at higher data rates is as much as 24% lower than that of T-MAC. Also, ADV-MAC has the least latency and the maximum throughput at all data rates. 20% S-MAC and T-MAC also have the least latency and maximum throughput, but their energy consumptions are much higher than ADV-MAC, as pointed out before. Thus ADV-MAC successfully adapts to traffic load, providing low energy consumption with high throughput and low latency over all traffic conditions.

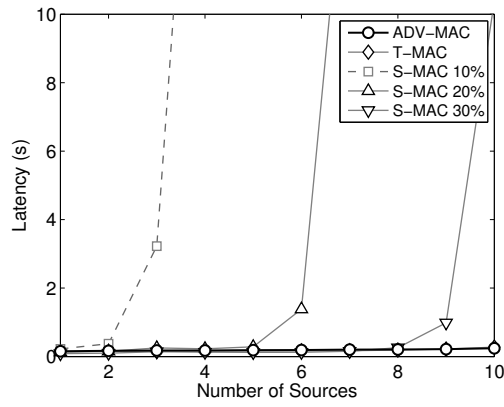
3.2.2.2 Effect of number of sources on single hop, unicast scenario

In the second set of simulations, we investigate the effect of different numbers of sources on the performance of the three MAC protocols. The simulation setup is similar to the previous case, but we vary the number of sources from 1 to 10 and keep the data rate fixed at 1 pkt/sec for all sources.

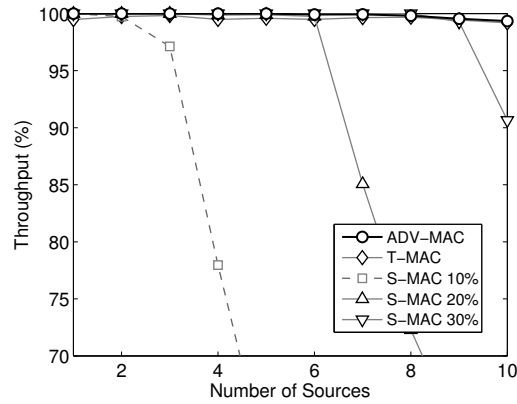
Fig. 3.3(a) shows the energy consumptions for the three protocols obtained from simulations and analysis. Fig. 3.3(b) and Fig. 3.3(c) show the latency and throughput comparisons, respectively. 10% S-MAC is suitable only for 1-2 sources. 20% S-MAC gives acceptable values of latency and throughput up to 6 sources. Beyond that, 30% S-MAC is needed. T-MAC adapts successfully to increasing number of sources (i.e., increasing load) with the least latency and the maximum throughput. However, the energy consumption of T-MAC increases with the number of sources. ADV-MAC, on the other hand, shows very little increase in energy consumption while maintaining similar latency and throughput as T-MAC. ADV-MAC provides up to 35% reduction in energy compared to T-MAC. It is to be noted that at 10 sources, both T-MAC and ADV-MAC initially have all the nodes awake for data transfer. However, as each pair of node complete their



(a) Energy Consumption vs. Number of Sources.



(b) Latency vs. Number of Sources.



(c) Throughput vs. Number of Sources.

Figure 3.3: Single hop, unicast vs. number of sources: Performance comparison of ADV-MAC, T-MAC and S-MAC. Extension '-a' represents analytical results.

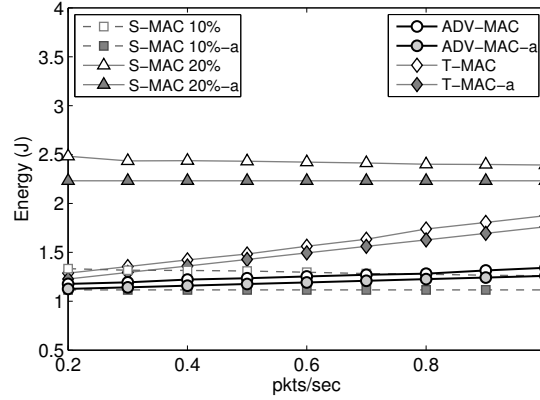


Figure 3.4: Single hop, multicast: Energy consumption vs. data rate. The extension ‘-a’ corresponds to analytical results.

data exchange, they go to sleep in ADV-MAC. However, all nodes in T-MAC keep renewing their timers and stay awake until the last pair of nodes has completed their data exchange. Thus the energy consumption of T-MAC is considerably higher compared to ADV-MAC. Thus, ADV-MAC also adapts successfully to different numbers of sources with the least energy consumption and while still maintaining the minimum latency and the maximum throughput.

3.2.2.3 Effect of data rate on single hop, multicast scenario

In the third simulation setup, we consider the performance of the three protocols under multicasting as we vary the data rate. We consider an area of $50\text{ m} \times 50\text{ m}$ with 20 nodes including 4 sources. All nodes are in transmission range of each other. There are 4 types of data, and each source broadcasts a specific type of data. Each receiving node is interested in only one of these four data types. The receiving node types are uniformly distributed. As in the first set of simulations, we vary the traffic load by increasing the data rate from 0.2 pkt/sec to 1 pkt/sec. Fig. 3.4 shows the energy consumption of the three protocols obtained from simulations and analysis under multicasting. As seen from the figure, ADV-

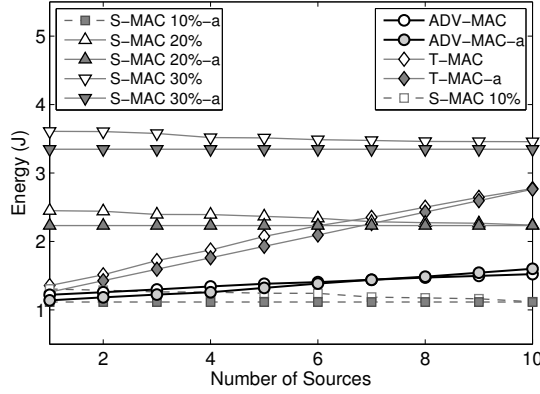
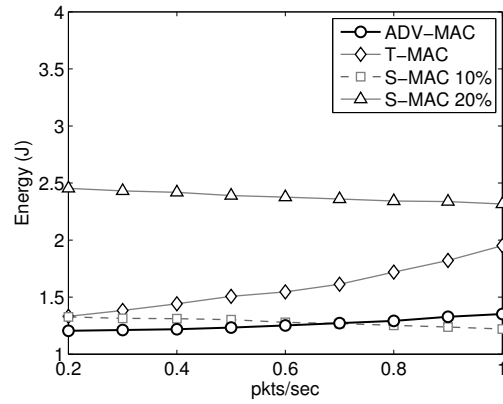


Figure 3.5: Single hop, multicast: Energy consumption vs. number of sources. The extension ‘-a’ corresponds to analytical results.

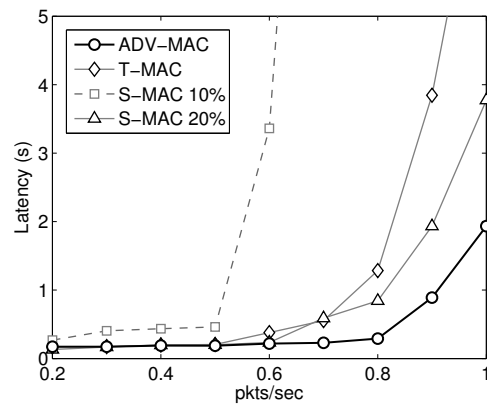
MAC results in the least energy consumption values for all data rates. The energy savings of ADV-MAC is because ADV packets contain the type of the data to be broadcast. Hence, nodes that are not interested in that type of data go to sleep, saving energy. As a result, ADV-MAC provides up to 28% reduction in energy compared to T-MAC. The latency and throughput trends are similar to the first set of simulations and are not shown. Thus, ADV-MAC has the least energy consumption in a multicasting scenario with high throughput and low latency.

3.2.2.4 Effect of number of sources on single hop, multicast scenario

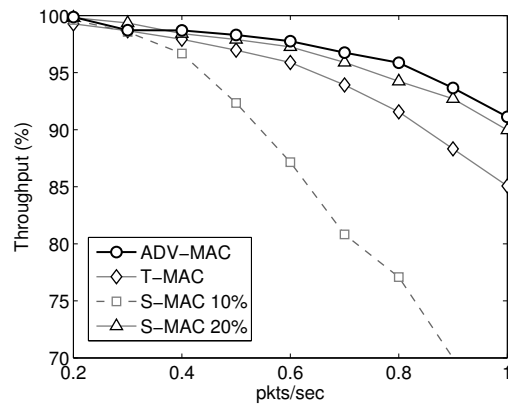
In this set of simulations, we consider the performance of the three protocols under multicasting as we increase the number of sources from 1 to 10. As in the previous simulations, there are 20 nodes including the sources in an area of 50 m x 50 m . The sources are transmitting at the rate of 1 pkt/sec. There are 4 data types, and the data types are uniformly distributed among the sources as well as the receivers. Fig. 3.5 shows the energy consumption results for the three protocols. The trend is similar to the unicast case in the second set of simulations. The energy consumption of ADV-MAC is 45% less compared to T-MAC for 10



(a) Energy Consumption vs. Data Rate.



(b) Latency vs. Data Rate.



(c) Throughput vs. Data Rate.

Figure 3.6: Multi-hop, unicast vs. data rate: Performance comparison of ADV-MAC, T-MAC and S-MAC.

sources. The latency and throughput (not shown) are similar to the unicast case in the second set of simulations where ADV-MAC achieves the highest throughput and the least latency.

3.2.2.5 Effect of data rate on multi-hop, unicast scenario

To investigate the performance of the three MAC protocols in a multi-hop communication environment, we define a new simulation set. We consider an area of $700\text{ m} \times 700\text{ m}$. There are 312 nodes with each node having an average of 20 neighbors in its transmission range. There are 20 source nodes, and we increase the data rate from 0.2 pkt/sec to 1 pkt/sec. On the average, each receiver will have 5 transmitting nodes in its carrier sense range. Fig. 3.6(a) shows the energy consumptions obtained from the simulations. Since the energy equations are valid only for the single hop case, we do not show analytical results. In the previous simulations, all nodes could hear each other. Therefore, T-MAC did not suffer from the early sleeping problem. However, since this is a multi-hop case, the early sleeping problem is present in T-MAC, and the effects are visible in the simulation results. It is seen that the energy consumption of ADV-MAC is as much as 30% less compared to T-MAC and as much as 41% less compared to S-MAC with 20% duty cycle.

Fig. 3.6(b) shows the latency comparison. The latency of T-MAC is more than that of ADV-MAC at higher data rates. This is because of the early sleeping problem. However ADV-MAC is immune to the problem, and its latency does not increase as much. The effect of the early sleeping problem is also visible in the throughput comparison as seen in Fig. 3.6(c). It is seen that the throughput of T-MAC drops faster than ADV-MAC as the data rate increases.

3.3 Summary

This chapter presents ADV-MAC, a new MAC protocol for wireless sensor networks. ADV-MAC minimizes the energy lost due to idle listening by introducing the concept of advertising for contention. Simulations show that the protocol adapts nicely to low and high traffic loads as well as to variable loads. ADV-MAC provides further reductions in energy compared to S-MAC and T-MAC while not sacrificing throughput or latency. In fact, in multi-hop variable load cases, ADV-MAC not only has the least energy consumption, but also has better latency and throughput compared to T-MAC. Also, ADV-MAC introduces an energy efficient multicasting mechanism at the MAC level that is absent in S-MAC and T-MAC.

In the next chapter we derive an analytical model for the packet delivery ratio and the energy consumption of the ADV-MAC protocol. We verify the analytical model with simulations and use the model to choose an optimal value of the advertisement period. The optimized ADV-MAC provides substantial energy gains over the non-optimized version while faring as well as T-MAC in terms of packet delivery ratio and latency.

4 ADV-MAC: Analysis and Optimization for Energy Efficiency

In Chapter 3, we presented Advertisement-MAC (ADV-MAC), a MAC protocol designed to minimize the energy wasted in idle listening. The energy consumption of ADV-MAC is up to 45% less than the two well known MAC protocols, Sensor-MAC (S-MAC) [50] and Timeout-MAC (T-MAC) [46] for the scenarios investigated.

Although ADV-MAC as proposed in Chapter 3 provides substantial energy savings as well as good throughput and latency performances compared to S-MAC and T-MAC, it is not optimal. We had chosen an arbitrary value of the advertisement (ADV) period duration as well as the duration of the data contention period in Chapter 3. Depending on the network load, an arbitrary choice of these durations may lead to energy waste due to idle listening, or it may lead to degradation in throughput and latency as well as energy loss due to excessive collisions. To prevent this, we propose an analytical model for ADV-MAC in this chapter. We perform simulations to validate this analytical model, and then we use the model to determine the optimal value for the ADV period duration as well as the duration of the data contention period. Also, we propose a more

efficient contention method for data contention, which provides further energy savings. Simulation results show that the optimized ADV-MAC provides a reduction in energy dissipation ranging from 30% to 70% or more while maintaining the packet delivery ratio and latency of the original ADV-MAC for the scenarios investigated.

The rest of this chapter is organized as follows. In Section 4.1, we describe the updated design of the ADV-MAC protocol. In Section 4.2, we present the analytical model of ADV-MAC. In Section 4.3, we present the performance results. In this section, we first verify the analytical model with simulations and choose the optimal duration of the ADV and data contention periods. Then, we present a detailed performance comparison of the optimized ADV-MAC protocol with the previous ADV-MAC protocol, as well as S-MAC and T-MAC. Finally in Section 4.4, we conclude the chapter.

4.1 ADV-MAC Design Updates

The basic design of the ADV-MAC protocol remains the same as described in Chapter 3. However we change the contention method of both the ADV period and the Data Period.

4.1.1 ADV Period Contention

We change the contention method of the ADV period slightly, as it simplifies the analytical model without decreasing the packet delivery ration in the ADV period. Let S_{adv} be the duration of the ADV period and t_{adv} be the duration of an ADV packet, both in unit *slots*. Each node contending for the medium to send an ADV packet selects a slot randomly from the first $S_{adv} - t_{adv}$ slots. The last t_{adv} slots are not selected, since an ADV packet transmission cannot finish within

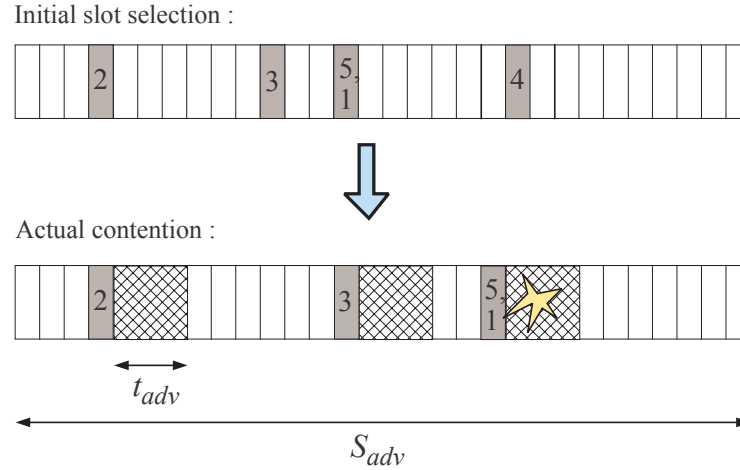


Figure 4.1: Contention method in the ADV period. In this example, 5 nodes contend to send, two successfully transmit, two collide and one defers transmission.

the ADV period in that case. Each node starts a timer for the duration until its selected slot. If a node hears any transmission before its timer reaches zero, it will freeze the timer and resume it when that transmission ends (previously, a node hearing a transmission would cancel its timer and start afresh after the end of that transmission). When the timer reaches zero, the node will transmit if enough time is left in the ADV period to transmit. Otherwise, it will defer its transmission and try again in the next frame. Fig. 4.1 illustrates the contention of 5 nodes for $S_{adv} = 30$ and $t_{adv} = 3$. Node 2 selects the 4th slot and starts transmitting from the 5th slot, i.e., when its timer reaches zero. Once all other nodes hear the transmission of node 2, they freeze their timers for t_{adv} slots. However, since node 4 freezes its timer 3 times, the ADV period finishes before its timer expires, which makes node 4 defer its transmission and contend again in the next frame. Note that no acknowledgments are sent for ADV packets. For this reason, nodes 1 and 5, whose ADV packets were sent but collided, will not know of the collision and will try to contend for the medium during the data period.

4.1.2 Data Period Contention

The main idea of the ADV contention method is also used in the data period contention. Let Γ be the duration of the data period and S_{data} be the duration of the contention window used before each data transmission, both in unit *slots*. Let us consider no retransmissions first. Nodes that transmit in the ADV period, select a slot out of S_{data} and set their timer to the duration until their selected slot. When the timer of a node reaches zero, the node begins the data exchange by sending an RTS packet. RTS and CTS packets contains the duration of the data exchange, so that nodes that are not part of the ongoing data exchange can freeze their timers for the duration of this data exchange.

Nodes whose ADV packets collided in the ADV period also contend in the data period. However, they cannot receive any CTS as their corresponding receivers are asleep, which results in the nodes to timing out and to going to sleep. If RTS packets collide, the corresponding receivers will wait for the entire duration of S_{data} and then go to sleep. The senders in this case will again go to sleep after the CTS timeout. If we consider one retransmission, senders will randomly select a new slot out of S_{data} and repeat the process if they do not receive CTS.

This contention method defined in this chapter is different than the one defined in Chapter 3. In the method proposed in Chapter 3, all nodes hearing a transmission cancel their timers and choose a new slot out of S_{data} once the ongoing transmission ends. This process is repeated until all nodes finish contending or the end of the data period is reached. We evaluated both methods with simulations for various numbers of contending nodes to analyze their individual advantages. We consider no retransmissions. The number of receiver nodes are assigned to be equal to the number of contending nodes in the simulations. The total time spent (in units of slots) by all nodes (senders and receivers) in contentions are shown in Fig. 4.2 along with the collision probabilities observed. Using the contention

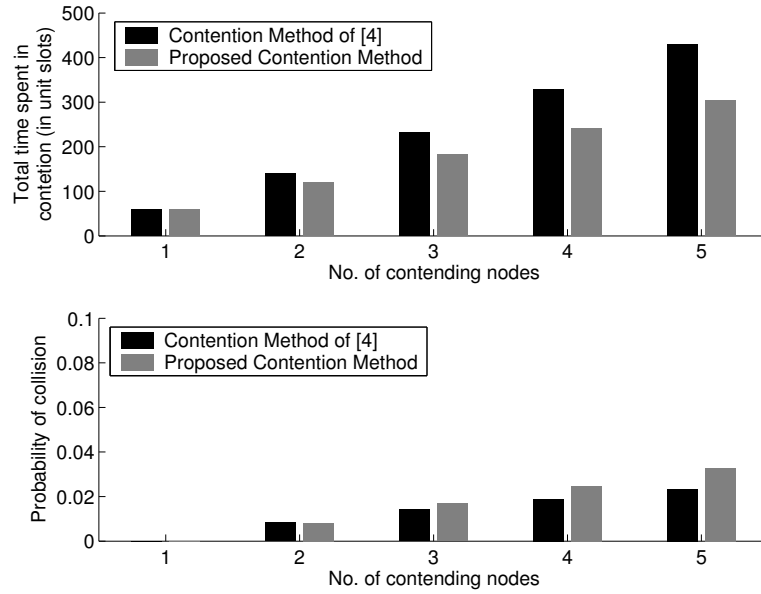


Figure 4.2: Total average time spent by all contending nodes and the probability of collisions in both contention methods.

method proposed in this chapter decreases the total time spent by nodes in contention. When there are 5 contending nodes, the total time spent in contention is reduced by as much as 25% compared to the contention method of [38]. Since the time spent awake is directly proportional to energy consumption, there is also an energy savings of 25%. This energy gain is obtained with almost no change in the probability of collision, which increases only 1% for 5 nodes.

4.2 Analytical Model of ADV-MAC

In this section, we model the packet delivery ratio and the energy consumption of ADV-MAC. We assume a perfect channel and no retransmissions within a frame.

4.2.1 Packet Delivery Ratio

To determine the packet delivery ratio, we first calculate the average throughput for a single frame given that there are N nodes contending. We analyze the two levels of contention sequentially: the contention in the ADV period and the contention in the data period. For convenience, important notations used in the analysis are presented in Tables 4.1.

4.2.1.1 Contention in the ADV period

Let us consider N nodes competing to transmit in the ADV period. To be able to calculate the number of nodes that will contend in the data period, we need to find the number of nodes that transmit ADV packets in the ADV period, either successfully (with no collision) or unsuccessfully (with collision). This is a random variable, which we denote by X and its corresponding value by x . Hence, $x \leq N$. Assume that x nodes choose g of S_{adv} slots, where $g \leq x$. If $g = x$, there will be no collisions. Let $\Theta(x, g)$ denote the number of possible ways to assign x nodes to g slots. Mathematically, this problem is the same as the number of different distributions when placing x distinguishable balls into g indistinguishable bins such that each bin has at least one ball. Hence, the expression of $\Theta(x, g)$ is found to be

$$\Theta(x, g) = \frac{1}{g!} \sum_{s=0}^{g-1} (-1)^s \binom{g}{g-s} (g-s)^x. \quad (4.1)$$

Proof of (4.1) is given in A.1.

Let $\xi(x, g)$ denote the number of slot selections by N nodes that result in g distinct slots to be selected, where x nodes can transmit, and $N - x$ nodes defer their transmission due to insufficient time in the ADV period. The formula and proof of $\xi(x, g)$ is given in A.2. Let $g_{max}(x)$ denote the maximum g value, i.e., the maximum number of distinct slots used by x transmitting nodes. If t_{adv} denotes

Table 4.1: ADV-MAC Analytical Model: Notations Used I

Symbol	Description
c_i	Slot no. of i^{th} selected slot from the beginning of the data contention window
$g_{max}(x)$	Maximum number of distinct slots for x transmitting nodes
\bar{i}	Expected no. of nodes that get to transmit out of the \bar{X} nodes contending in the data period
M	Total no. of nodes in one hop neighborhood
N	No. of nodes contending at the beginning of a frame
N_r	No. of nodes that are to make their r^{th} retrials contending at the beginning of a frame at steady state
$P_{adv-nc},$ $P_{data-nc}$	Probability of no collision for any node in ADV and data period, respectively
PDR	Packet delivery ratio
S	Index number of the slot of ADV period up to which nodes can choose slots
S_{adv}	Duration of ADV period
S_{data}	Duration of contention window in Γ
S_{sync}	Duration of SYNC period
t_{adv}	Duration of ADV packet
t_{ctrl}	Duration of a control packet (RTS, CTS, ACK)
t_{data}	Duration of data packet
t_{eifs}	Duration of extended inter-frame space
$t_{timeout}$	Duration of CTS timeout

Table 4.2: ADV-MAC Analytical Model: Notations Used II

Symbol	Description
X	Random variable denoting no. of nodes that transmit in ADV period, i.e., no. of nodes contending in the data period
x	value assumed by random variable X
\bar{X}	Expected value of X
X_c	No. of nodes that transmit with collision in ADV period
X_s	No. of nodes that transmit with success in ADV period
\bar{Y}_s	Expected no. of nodes that transmit with success in data period
Γ	Duration of data period
ν_{rx}, ν_{tx}	Power for reception and transmission, respectively
$\Theta(x, g)$	No. of ways to assign x nodes to g slots
ρ	Throughput of a single frame in steady state
τ_k	Duration of the time for which nodes freeze their timer for the k^{th} transmission
ε_{adv}	Total energy consumption in the ADV period
ε_{data}	Total energy consumption in the data period
ε_{sync}	Total energy consumption in the SYNC period
ε_{total}	Energy consumption per node per packet
$\omega(x, N)$	No. of ways of slot selection that give x transmitting nodes and $N - x$ deferring nodes
$\xi(x, g)$	No. of ways in which we can choose g distinct slots which will result in x transmitting nodes and $N - x$ deferring nodes in an ADV period

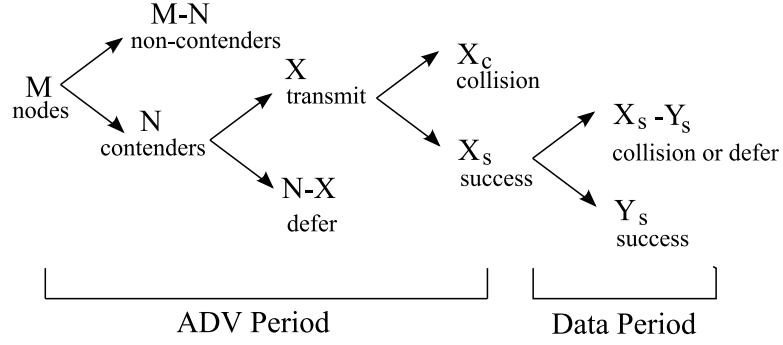


Figure 4.3: Symbols used in the analysis.

the duration of an ADV packets, and S_{adv} denotes the duration of the ADV period (both in units of slots), then

$$g_{max}(x) = \min \left(x, \left\lfloor \frac{S_{adv}}{t_{adv} + 1} \right\rfloor \right). \quad (4.2)$$

Note that we divide S_{adv} with $(t_{adv} + 1)$, since the transmission of an ADV packet requires one preceding empty slot. The number of slot selections that result in x transmitting nodes and $N - x$ deferring nodes, $\omega(x, N)$, is given by

$$\omega(x, N) = \binom{N}{x} \sum_{g=1}^{g_{max}(x)} \Theta(x, g) \xi(x, g). \quad (4.3)$$

All N nodes select a slot out of the first $S = S_{adv} - t_{adv}$ slots of the ADV period to be able to finish their transmission before the end of the the ADV period. Hence, the number of all possible slot selections by N nodes is S^N . Consequently, the probability of having x transmitting nodes out of N nodes, $P(X = x | N)$, can be written as

$$\begin{aligned} P(X = x | N) &= \frac{\omega(x, N)}{S^N} \\ &= \frac{1}{S^N} \binom{N}{x} \sum_{g=1}^{g_{max}(x)} \Theta(x, g) \xi(x, g). \end{aligned} \quad (4.4)$$

For contention analysis of the data period, the number of ADV transmissions that result in collisions and success should be derived. The probability that a slot

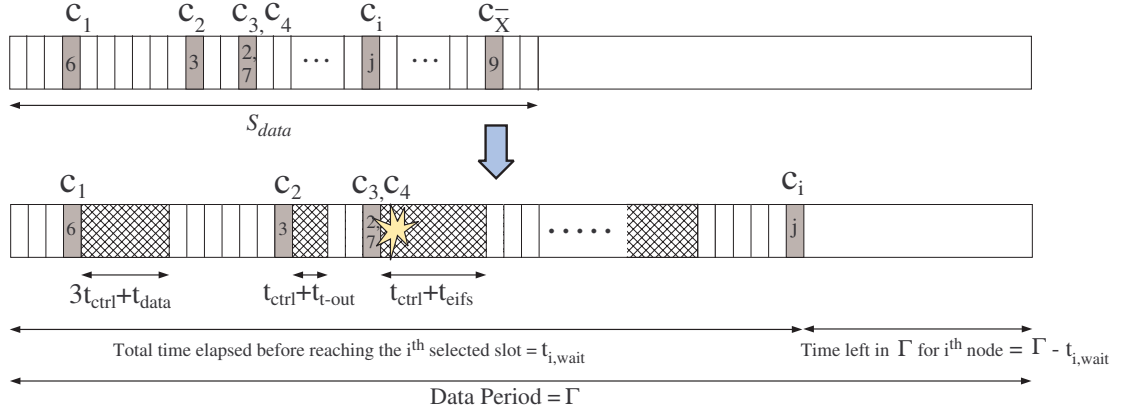


Figure 4.4: Example of contention in data period: the node with index number 6 was successful in both the ADV as well as the data periods. The node with index number 3 had a collision in the ADV period and hence times out. The nodes with indices 2 and 7 collide in the data period.

selected by a node is collision-free is

$$P_{adv-nc} = \frac{S(S-1)^{N-1}}{S^N}. \quad (4.5)$$

Thus, X_s , the expected number of nodes that transmit successfully out of X transmitting nodes, and X_c , the expected number of nodes whose ADV packets collide are given by

$$X_s = \bar{X} P_{adv-nc}, \quad (4.6)$$

$$X_c = \bar{X} (1 - P_{adv-nc}). \quad (4.7)$$

where \bar{X} represents the expected value of X and can be calculated as

$$\bar{X} = E[X|N] = \sum_{x=1}^N x P(X = x|N). \quad (4.8)$$

Note that the value of \bar{X} that is used to calculate X_s and X_c may be fractional. In the following analysis, we first consider \bar{X} to be an integer variable. Then, we present a method to apply fractional values of \bar{X} to the derived analysis.

4.2.1.2 Contention in the Data period

The expected number of nodes that transmit DATA packets successfully in the data period, \overline{Y}_s , depends on the ADV packet transmission success and failure of the nodes, i.e., X_s and X_c . We assume that there are no retransmissions, i.e., if a node does not receive a CTS packet reply for its RTS packet, it will not attempt again in that frame.

Let the duration of the data contention period be S_{data} in unit *slots*. Therefore, $S_{data} \leq \Gamma$, where Γ is the data period length in unit *slots*. Each of the \overline{X} nodes will randomly select a slot out of S_{data} and set its contention timer to the value of the selected slot.

Consider the slot selections from the contention window as illustrated in Fig. 4.4. Let c_i represent the i^{th} selected slot from the beginning of the contention window and n_i represent the node that selected c_i . If multiple nodes select the same slot, that slot is given different indices as shown in Fig. 4.4. We want to calculate the expected number of nodes that can transmit within one frame. Let us consider the n_i^{th} node and find the probability that it can send within the current frame. For that, we need to find the total time elapsed until the n_i^{th} node's contention timer reaches zero, $t_{i,wait}$, as illustrated in Fig. 4.4. There are two components of $t_{i,wait}$. The first component is the total time spent by the n_i^{th} node for carrier sensing the medium. This is given by c_i , i.e., the initial value of its contention timer. The other component is the time elapsed during the transmissions of the $i - 1$ preceding nodes. Whether there are one or more transmitting nodes that starts transmission in the same preceding slot, it is considered as one transmission for the n_i^{th} node. Hence, the number of preceding transmissions is $i - 1$ for the n_i^{th} node in case of no collisions, whereas in the case of collisions, there will be less than $i - 1$ preceding transmissions.

When a transmission begins in the channel, the nodes that are not part of this

transmission freeze their timers for a specific duration. Assume that τ_k is the duration of the time the n_i^{th} node freezes its timer for the k^{th} preceding transmission. Depending on whether the k^{th} transmission is a collision or whether the receiving nodes successfully received the ADV packet in the ADV period, there are four different values of τ_k corresponding to one of the following four possible cases:

1. Success in the ADV and success in the data period:

$$3t_{ctrl} + t_{data} \text{ (RTS+CTS+DATA+ACK duration)}$$

2. Success in the ADV and collision in the data period:

$$t_{ctrl} + t_{eifs} \text{ (RTS+Extended Interframe Space duration)}$$

3. Collision in the ADV and success in the data period:

$$t_{ctrl} + t_{timeout} \text{ (RTS+Timeout duration)}$$

4. Collision in the ADV and collision in the data period:

$$t_{ctrl} + t_{eifs} \text{ (RTS+Extended Interframe Space duration),}$$

where, the duration of control packets such as RTS, CTS or ACK is denoted by t_{ctrl} and the duration of a data packet is denoted by t_{data} . The duration of an extended inter-frame space is denoted by t_{eifs} . The extended inter-frame space is used to defer transmission if a previously received packet contains an error. The duration of an CTS timeout is denoted by $t_{timeout}$.

Then,

$$t_{i,wait} = c_i + \sum_k \tau_k, \tag{4.9}$$

where \sum_k adds the durations of the timer suspensions of node n_i due to the preceding transmissions.

The n_i^{th} node will transmit, if

$$\begin{aligned}
\Gamma - t_{i,wait} &\geq 3t_{ctrl} + t_{data} \\
\Rightarrow \Gamma - (c_i + \sum_k \tau_k) &\geq 3t_{ctrl} + t_{data}, \\
\Rightarrow \Gamma - (c_i + \sum_k \tau_k + 3t_{ctrl} + t_{data}) &\geq 0.
\end{aligned} \tag{4.10}$$

To calculate the expected number of successful data transmissions, \bar{Y}_s , we need to find the expected number of transmitting nodes, \bar{i} and the probability of *Case 1*. However, \bar{i} is determined by the expected value of τ_k and c_i , i.e., $\bar{\tau}_k$, and \bar{c}_i , respectively. The expression of \bar{c}_i is given in A.3. Note that since we consider the expected value of c_i , no cases of two or more nodes selecting the same slot, i.e., no cases of collisions, exist. Hence, there will be exactly $i - 1$ transmissions before the n_i^{th} node. Consequently, we can rewrite (4.10) as

$$\Gamma - (\bar{c}_i + (i - 1)\bar{\tau} + 3t_{ctrl} + t_{data}) \geq 0. \tag{4.11}$$

The expected value of τ , $\bar{\tau}$, incorporates the effects of collisions in (4.11) and is given by

$$\begin{aligned}
\bar{\tau} &= P_{adv-nc}P_{data-nc}(3t_{ctrl} + t_{data}) \\
&+ P_{adv-nc}(1 - P_{data-nc})(t_{ctrl} + t_{eifs}) \\
&+ (1 - P_{adv-nc})P_{data-nc}(t_{ctrl} + t_{timeout}) \\
&+ (1 - P_{adv-nc})(1 - P_{data-nc})(t_{ctrl} + t_{eifs}),
\end{aligned} \tag{4.12}$$

where $P_{data-nc}$ is the probability that a transmission results in no collision. As for the probability of no collision in the ADV period, i.e., as in (4.5), $P_{data-nc}$ is given by

$$P_{data-nc} = \frac{S_{data}(S_{data} - 1)^{\bar{X}-1}}{S_{data}^{\bar{X}}}. \tag{4.13}$$

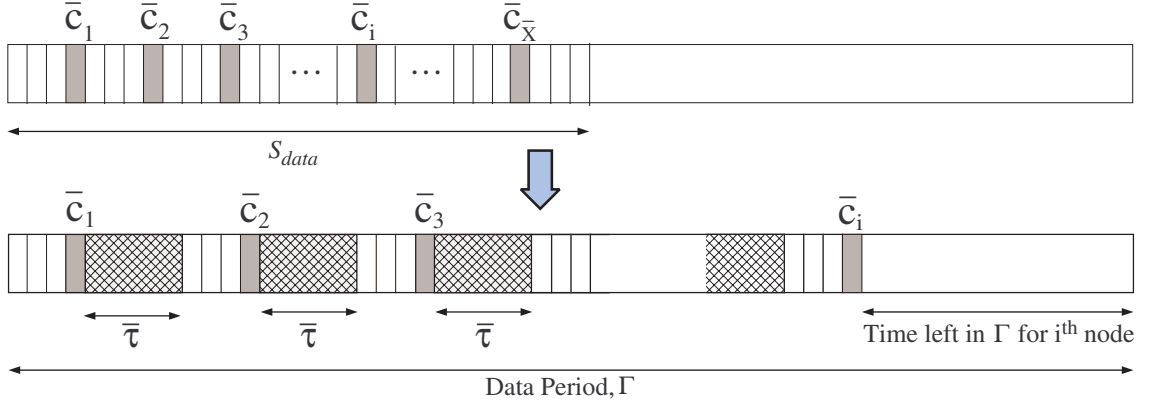


Figure 4.5: Contention in the data period with expected values of the chosen slots

The number of nodes that get to transmit out of \bar{X} contending nodes is defined and calculated as

$$\bar{i} = \max_{\substack{\Psi_i \geq 0 \\ i \in (0, \bar{X})}} i, \quad (4.14)$$

where Ψ_i represents the LHS of the inequality given in (4.11). The average number of nodes that can transmit successfully in the data period is, then,

$$\bar{Y}_s = \bar{i} P_{data-nc} P_{adv-nc}. \quad (4.15)$$

Note that \bar{Y}_s is determined by \bar{X} through (4.13)-(4.15), however, \bar{X} is assumed to be an integer. In fact, \bar{X} has fractional values more often than not. For these cases, we present a linear interpolation method using the two closest integer values, $\lfloor \bar{X} \rfloor$ and $\lceil \bar{X} \rceil$, as follows. Let the number of nodes successfully transmitting in the data period from $\lfloor \bar{X} \rfloor$ and $\lceil \bar{X} \rceil$ contending nodes be \bar{Y}_{s1} and \bar{Y}_{s2} , respectively. Given that

$$\bar{X} = q \lfloor \bar{X} \rfloor + (1 - q) \lceil \bar{X} \rceil, \quad (4.16)$$

the linear interpolation of \overline{Y}_s is calculated as

$$\overline{Y}_s' = q\overline{Y}_{s1} + (1 - q)\overline{Y}_{s2}. \quad (4.17)$$

For a given number of contending nodes, the expected number of successful nodes in the data period can be found using (4.17). However, to find the *successful packet delivery ratio*, retries must be incorporated to the successful contention calculations. For that, we consider the steady state case, where there are N_r number of nodes attempting their r^{th} retrials. The variable N_0 corresponds to the new contending nodes at the beginning of the frame, which is a system parameter to be used to evaluate the MAC protocol performance. Considering that the maximum number of retries is 2, the total number of nodes contending at the beginning of a frame in the steady state is $N = N_0 + N_1 + N_2$. Therefore, we need to derive the formula for N_1 and N_2 given N_0 , to incorporate the retries in the successful contention calculations.

Let the steady state throughput for each frame, i.e., the ratio of successful nodes, for N contending nodes, be ρ . Then,

$$N_1 = N_0(1 - \rho). \quad (4.18)$$

This is because, out of N_0 fresh nodes, $N_0(1 - \rho)$ will fail to transmit and try in the next frame. Similarly,

$$\begin{aligned} N_2 &= N_1(1 - \rho), \\ &= N_0(1 - \rho)^2. \end{aligned} \quad (4.19)$$

Then, the total number of nodes contending at the beginning of a frame in the steady state is given by

$$\begin{aligned} N &= N_0 + N_1 + N_2, \\ &= N_0(1 + (1 - \rho) + (1 - \rho)^2), \\ &= N_0(\rho^3 - 3\rho + 3) = \eta(\rho), \end{aligned} \quad (4.20)$$

where $\eta(\rho)$ is a function of ρ . Note that the steady state value of N can be fractional. However, an integer value of N is needed for the equations in the ADV period calculations given in Section 4.2.1.1. In that case, we apply the linear interpolation method used for the calculation of \overline{Y}_s' with fractional \overline{X} values, i.e., Equations (4.16)-(4.17).

The throughput of the steady state contention is

$$\begin{aligned}\rho &= \frac{\overline{Y}_s'}{N} = \frac{\overline{Y}_s'}{\eta(\rho)}, \\ \Rightarrow \rho - \frac{\overline{Y}_s'}{\eta(\rho)} &= 0.\end{aligned}\tag{4.21}$$

We can solve (4.21) numerically in the feasible interval $[0, 1]$ and find the steady state value of the throughput ρ . Since there are N_0 fresh nodes at a frame, then the total number of packets transmitted successfully is $N\rho = N_0(\rho^3 - 3\rho + 3)\rho$ as we have N contending nodes for the frame. Then, the successful packet delivery ratio is found to be

$$\begin{aligned}PDR &= \frac{N_0(\rho^3 - 3\rho + 3)\rho}{N_0} \\ &= (\rho^3 - 3\rho + 3)\rho.\end{aligned}\tag{4.22}$$

4.2.2 Energy Consumption

In this section, we present the mathematical analysis for average energy consumption of the nodes that employ ADV-MAC. We consider a single hop neighborhood where all nodes are in transmission range of one another, i.e., there are no hidden terminals. Let the total number of nodes in this neighborhood be M . Although all of the M nodes are awake in the SYNC and ADV period, only those nodes that transmitted ADVs or successfully received ADVs will be awake in the data period. Let the transmission and reception power be denoted by ν_{tx} and ν_{rx} , respectively.

Then, the expected total energy consumption in one ADV period is

$$\varepsilon_{adv} = \overline{X}t_{adv}\nu_{tx} + (S_{adv}M - \overline{X}t_{adv})\nu_{rx}, \quad (4.23)$$

since only \overline{X} of the M nodes transmit ADV packets with duration t_{adv} and all of the M nodes listen to the medium except the time they transmit an ADV packet.

The energy consumption of the SYNC period is

$$\varepsilon_{sync} = S_{sync}M\nu_{rx}. \quad (4.24)$$

For the sake of the brevity, we ignore the infrequent SYNC packet transmissions such as one transmission in 50 frames.

For the data period, we consider N_0 fresh nodes for a frame as in Section 4.2.1.2. Then, the total number of contending nodes at the beginning of a frame in steady state, N , can be calculated using (4.20). Of these N nodes, \overline{X} nodes transmit in the ADV period on average. Out of these \overline{X} nodes, X_s will be successful and X_c will result in collision, on average. Since only X_s nodes transmit successfully in the ADV period, there will be X_s corresponding receivers awake in the data period. However, X_c nodes will have no receivers waiting for them.

Expected chosen slots are shown to be equally spaced in Section 4.2.1.2 and are illustrated in Fig. 4.5. We will evaluate the energy consumption in this expected slot assignment case. Let us consider a transmission that starts at \overline{c}_i and the transmitter and the destined receiver node of this transmission. Depending on the success or the collision of the ADV packet transmission and the data packet transmission, there are four possible cases that the transmission starting at \overline{c}_i can have. These cases are:

- *Success in both ADV and data packet transmissions:* All nodes go to sleep after receiving the RTS packets intended for other nodes. Moreover, once node n_i finishes its transmission in the data period, it and the corresponding receiver go to sleep until the end of the frame. For the sake of brevity, we

ignore the case of a transmitter node being a receiver of another transmission. The energy consumption of the transmitting node in the data period is, then,

$$\begin{aligned}\varepsilon_{s_1,i} &= \nu_{rx}\overline{c_i} + \nu_{rx}(i-1)t_{ctrl} \\ &+ \nu_{tx}(t_{ctrl} + t_{data}) + \nu_{rx}2t_{ctrl},\end{aligned}\tag{4.25}$$

where the first term in (4.25) corresponds to the energy consumed due to carrier sensing and the second term corresponds to the energy consumed during reception of the RTS packets of the preceding $i-1$ transmissions. The last two terms are the energy consumptions due to transmitting an RTS and a data packet and receiving the corresponding CTS and ACK packets. Similarly, the energy consumption of the corresponding receiver is

$$\begin{aligned}\varepsilon_{r_1,i} &= \nu_{rx}\overline{c_i} + \nu_{rx}(i-1)t_{ctrl} \\ &+ \nu_{rx}(t_{ctrl} + t_{data}) + \nu_{tx}2t_{ctrl}.\end{aligned}\tag{4.26}$$

The first two terms of (4.26) are the same as that of equation (4.25). The last two terms are because the receiver receives the RTS and DATA packets and transmits the CTS and ACK packets. Thus, the total energy consumption of the sender and the corresponding receiver of the i^{th} transmission is

$$\varepsilon_{i_1} = \varepsilon_{s_1,i} + \varepsilon_{r_1,i}.\tag{4.27}$$

- *Success in the ADV packet transmission and collision in the RTS packet transmission:* Since the ADV packet is successfully transmitted, there will be a corresponding receiver awake in the data period. The energy consumption of the transmitting node is

$$\begin{aligned}\varepsilon_{s_2,i} &= \nu_{rx}\overline{c_i} + \nu_{rx}(i-1)t_{ctrl} \\ &+ \nu_{tx}t_{ctrl} + \nu_{rx}t_{timeout},\end{aligned}\tag{4.28}$$

where the last two terms correspond to the transmission of an RTS packet and to waiting for a reply for a duration of $t_{timeout}$. Due to the collision, the corresponding receiver will wait until the end of the last transmission. However, the energy consumption of the corresponding receiver is

$$\varepsilon_{r2,i} = \nu_{rx}\overline{c_i} + \nu_{rx}\bar{i}t_{ctrl} + \kappa t_{timeout}, \quad (4.29)$$

where

$$\kappa = \begin{cases} 0 & \text{if } \bar{i} < \overline{X} \\ 1 & \text{if } \bar{i} = \overline{X} \end{cases} \quad (4.30)$$

The term κ is derived as follows. After each transmission, the receiving node calculates the remaining time in the data period. Since the corresponding transmitting node's RTS collided, the receiving node will have no knowledge that its transmitter has already tried to transmit. So the receiving node will keep on listening to the medium until the data period ends or until it timeouts. When the data period is not big enough to accommodate all of the \overline{X} nodes to transmit, i.e., $\bar{i} < \overline{X}$, the receiving node will go to sleep directly after the last RTS transmission (collision or success), as it will know that there is not enough space in the data period for another transmission. When the data period is big enough to accommodate all of the \overline{X} nodes, i.e., $\bar{i} = \overline{X}$, the receiving node will timeout and go to sleep. Thus, during the data period, the total energy consumption of the sender and the receiver of the i^{th} transmission is

$$\varepsilon_{2,i} = \varepsilon_{s2,i} + \varepsilon_{r2,i}. \quad (4.31)$$

- *Collision in the ADV packet transmission and success in the RTS packet transmission:* In this case, the receiving node will not be awake in the data

period. Thus, the total energy consumption for the transmitter node is

$$\begin{aligned}\varepsilon_{3,i} &= \nu_{rx}\overline{c_i} + \nu_{rx}(i-1)t_{ctrl} \\ &+ \nu_{tx}t_{ctrl} + \nu_{rx}t_{timeout}.\end{aligned}\tag{4.32}$$

The first term of (4.32) is the energy consumption due to carrier sensing during contention and the second term is due to the reception of the $i-1$ previous RTS packets. The last two terms are due to the RTS packet transmission of the node and its channel listening during the following timeout.

- *Collision in the ADV packet transmission and in the RTS packet transmission:* The transmitter node will have no receiver awake for it in the data period and the energy consumption is exactly same as the previous case, which is

$$\begin{aligned}\varepsilon_{4,i} &= \nu_{rx}\overline{c_i} + \nu_{rx}(i-1)t_{ctrl} \\ &+ \nu_{tx}t_{ctrl} + \nu_{rx}t_{timeout}.\end{aligned}\tag{4.33}$$

On average, there will be \bar{i} nodes out of \overline{X} nodes to transmit, where $\bar{i} \leq \overline{X}$. Combining (4.27), (4.31), (4.32) and (4.33), the total average energy spent in the data period is given by

$$\begin{aligned}\varepsilon_{data} &= \sum_{i=1}^{\bar{i}} [P_{adv-nc}P_{data-nc}\varepsilon_{1,i} + P_{adv-nc}(1 - P_{data-nc})\varepsilon_{2,i} \\ &+ (1 - P_{adv-nc})P_{data-nc}\varepsilon_{3,i} + (1 - P_{adv-nc})(1 - P_{data-nc})\varepsilon_{4,i}] \\ &+ (1 - \kappa) \sum_{i=\bar{i}+1}^{\overline{X}} [P_{adv-nc}2(\nu_{rx}\overline{c_i} + \nu_{rx}\bar{i}t_{ctrl}) \\ &+ (1 - P_{adv-nc})(\nu_{rx}\overline{c_i} + \nu_{rx}\bar{i}t_{ctrl})].\end{aligned}\tag{4.34}$$

The first four terms within the first summation of (4.34) are due to the four possible cases of the i^{th} transmission. The second summation is for the transmissions that are deferred due to lack of time in the data period. The first term of this

summation is for the transmitting nodes whose ADV packet transmissions are successful. In that case the transmitting node will have a corresponding receiving node awake in the data period. Both these nodes will listen to the channel for a period of $\overline{c_i}$ and receive \overline{i} RTS packets. Similarly, the second term of the second summation is for the transmitting nodes whose ADV packets are collided. Again, the linear interpolation defined in Section 4.2.1.2 is used for the second summation since \overline{X} can be fractional.

Finally, using (4.23), (4.24) and (4.34), the average energy consumption per node per packet can be found as

$$\varepsilon_{total} = \frac{\varepsilon_{data} + \varepsilon_{adv} + \varepsilon_{sync}}{MN_0PDR}. \quad (4.35)$$

4.3 Optimized ADV-MAC Performance Evaluation

In the original ADV-MAC protocol proposed in Chapter 3, the duration of the ADV period is set to $15ms$ which is equal to the timeout period of T-MAC. Also, the data contention period is set to $12.7ms$. These parameters are not optimal for the traffic load and caused energy waste. In this section, we first verify the analytical models presented in Section 4.2 with simulations. Then, we use the analytical models to find the optimal values of the ADV period and the data contention window size and use the optimal values to evaluate the performance of the ADV-MAC protocol. In our simulations, we compared the performance of the optimized ADV-MAC with three protocols: the original ADV-MAC in Chapter 3, T-MAC and S-MAC. We use energy consumption per successfully transmitted packet, packet delivery ratio and latency as the metrics for comparison as we vary the traffic rate or the number of sources.

We performed all simulations in ns 2.29 [34]. The S-MAC code is included

in this version of ns. We coded T-MAC and ADV-MAC in ns-2 as well. We use different duty cycle settings, 10% and 20%, for S-MAC because one fixed duty cycle is not suitable for all traffic loads investigated. The frame time for 10% duty cycle of S-MAC is $236.4ms$, and we set this frame time for T-MAC and ADV-MAC as well. The transmission rate is $250kbps$, and the transmission range is $100m$, while the interference and carrier sense range is $200m$. Constant bit rate traffic generators with random noise in the scheduled departure times are used in the simulation. We use the power consumption values of MicaZ motes [1] to derive realistic energy consumption values. The current consumption values of MicaZ motes are $17.4mA$ and $19.1mA$ for transmission and reception, respectively. Considering a battery of 3 V, the transmission and reception powers are $52.2mW$ and $59.1mW$, respectively. The idle power consumption is the same as that for reception. The ADV period and the data contention period are divided into slots of $0.1ms$ each. For the multi-hop simulations, the nodes are deployed in the target area with uniformly random distribution. We use T-MAC with overhearing avoidance, since it is used in the simulations in [46]. The values used for the simulation parameters are summarized in Table 4.3.

4.3.1 Verification of the Analytical Models

We consider 20 nodes in an area of $50 \times 50m$. There are 10 sources, each transmitting $1pkt/sec$. This means that the average number of fresh nodes contending for the medium in each frame is $N_0 = 10 * frame\ duration = 10 * 0.236 = 2.36$. To verify the mathematical models, we calculated the packet delivery ratio (PDR) and energy consumption per node per packet for this scenario for different values of the ADV period using the proposed models. Since the frame size is fixed, varying the ADV period duration results in varying the data period duration. We used ns-2.29 to simulate the same scenario and the simulation results are compared to the analytical results. The PDR results are shown in Fig. 4.6, and the energy

Table 4.3: ADV-MAC Optimized Parameter Values

Parameters	Values
Carrier sense Range	200m
Duration of ADV period (Original ADV-MAC) (S_{adv})	15ms
Duration of frames	236.4ms
Duration of one slot	0.1ms
Duration of sync (S_{sync})	8.4ms
Duration of time-out period ($t_{timeout}$)	15ms
Duration of control packet (t_{ctrl})	0.9ms
Duration of data packet (t_{data})	8.5ms
Rx Idle Listening Power (ν_{rx})	59.1mW
Total Simulation Time	200s
Tx Listening Power (ν_{tx})	52.2mW
Transmission Rate	250Kbps
Transmission Range	100m

consumption values are shown in Fig. 4.7. The 95% confidence intervals are also shown in the figures. Each point in Figs. 4.6 and 4.7 are the average of 100 runs. In all subsequent simulations, each point is the average of 50 runs. The confidence intervals for the simulation results are observed to be very small, and hence are not shown in the figures for the sake of brevity.

In Fig. 4.6, we vary the ADV period duration from 1ms, i.e., from 10 slots

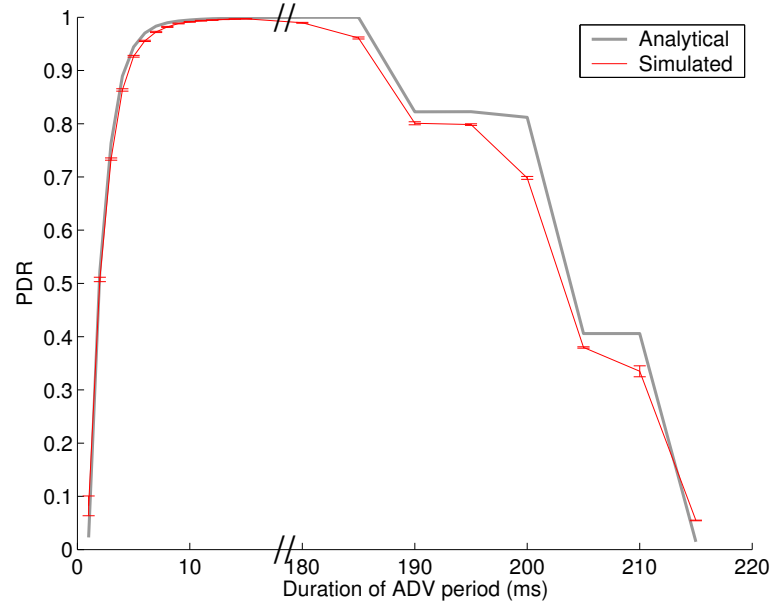


Figure 4.6: PDR values obtained from analysis and simulation. This graph is for $N_0 = 2.36$. Confidence intervals of 95% are shown for the simulation results.

to $215ms$, i.e., to 2150 slots. When the ADV period duration is very small, few nodes get to transmit ADV packets successfully. This results in very low PDR values. With a sufficient duration of the ADV period, the PDR rises to almost 100%. For very high values of the ADV period duration, most nodes successfully transmit their ADV packets. However, the time remaining for the data period in the frame becomes not enough to accommodate all the nodes that successfully transmitted ADV packets. Therefore, the PDR falls for very high ADV period duration values. With ADV period durations between $10ms$ and $185ms$, the PDR remains at around 100%. We use a break in the graph to hide that part from the graph in Fig. 4.6.

The average energy consumption per successfully transmitted packet initially decreases with increasing ADV period duration as shown in Fig. 4.7. The minimum is reached when the ADV period duration is around $4ms$. After this point, the energy consumption increases with increasing ADV period duration. This

happens because with a very small duration of the ADV period, the packet delivery ratio is very small, and hence, the average energy consumption per successfully transmitted packet is very high. As we increase the ADV period duration, the PDR increases rapidly, and the total energy consumption also increases due to increase in the duration of the ADV period, although the average energy consumption per successful packet decreases. With a sufficiently large value of the ADV period, the value of PDR remains fixed around 1. Increasing the ADV period duration beyond that value just increases the total energy consumption. Thus, the average energy consumption per successful packet increases after reaching a minimum.

From Figs. 4.6 and 4.7, it is seen that that curves from the analytical model follow the simulated curves very closely. We have done several other simulations with different values of N_0 , and observed that the analytical results closely follow the simulation results for different N_0 values, also. Hence, we conclude that our analytical models predict the average PDR and the average energy consumption of ADV-MAC accurately.

4.3.2 Selection of ADV Period Duration and Data Contention Window Duration

The PDR and energy consumption values of ADV-MAC depend on two parameters: S_{adv} or the ADV period duration and S_{data} or the duration of the data contention period. If S_{adv} is smaller than optimal, the PDR will be poor, and if S_{adv} is greater than optimal, it will cause energy waste. If S_{adv} is too large, it will lead to poor PDR as well. Similarly, smaller than optimal S_{data} values provide poor PDR, and greater than optimal S_{data} values cause energy waste. We use the analytical model to select the optimal values of both S_{adv} and S_{data} to meet a given PDR requirement at minimum energy expenditure.

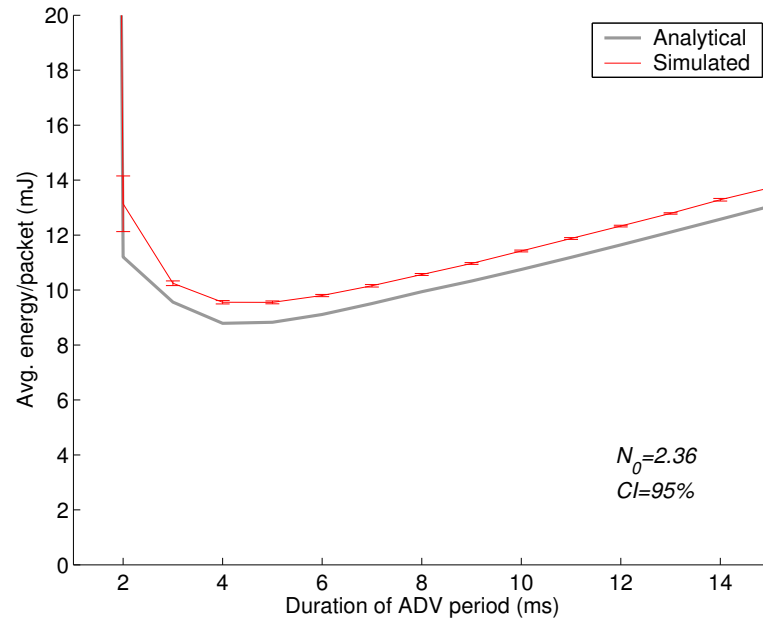


Figure 4.7: Average energy per node per packet obtained from analysis and simulation. This graph is for $N_0 = 2.36$. Confidence intervals of 95% are shown for the simulation results.

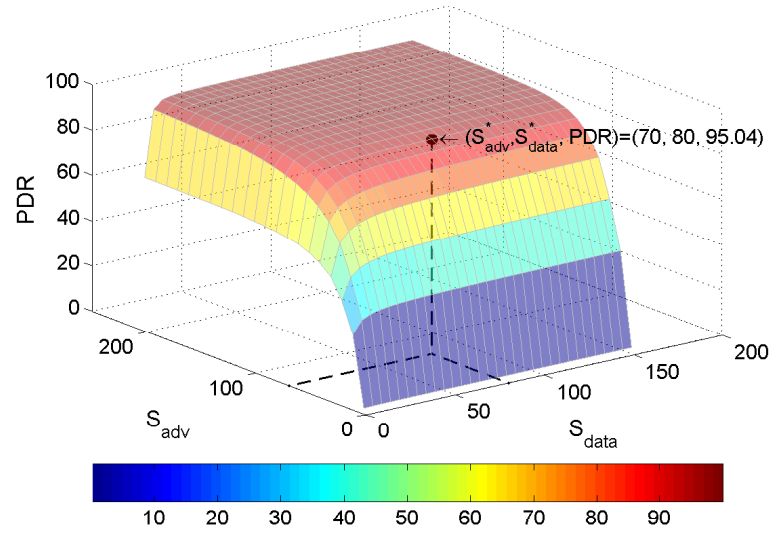
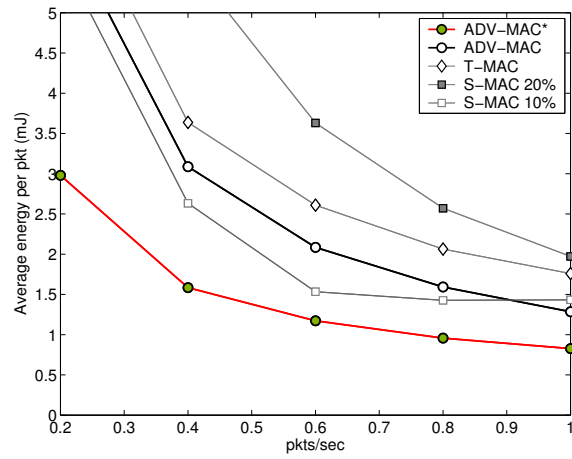
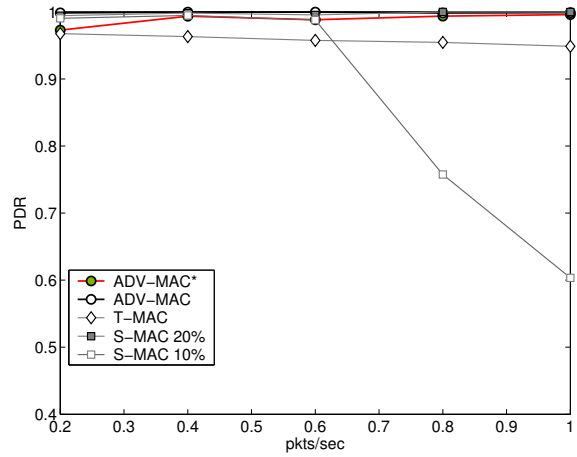


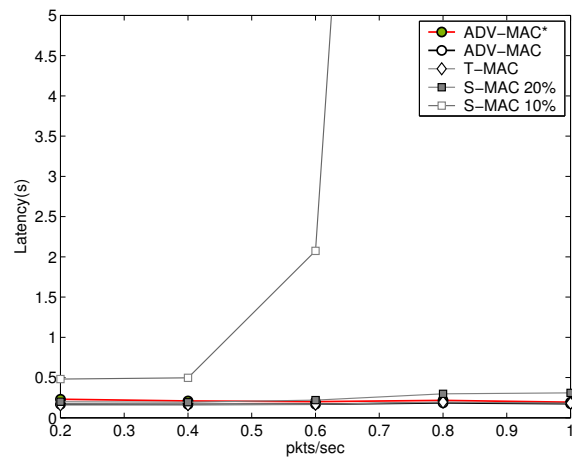
Figure 4.8: Optimal value of S_{adv} and S_{data} for a required PDR of 95% or more for $N_0 = 5$.



(a) Average energy consumption per node per packet

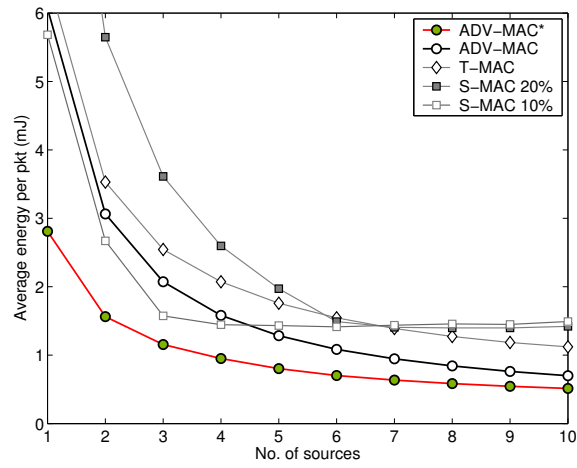


(b) Packet delivery ratio

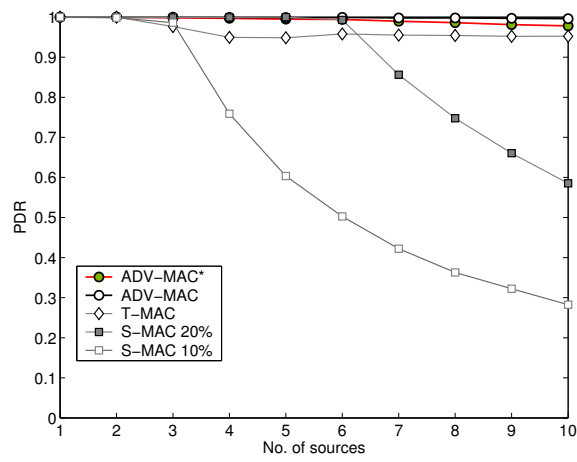


(c) Average latency

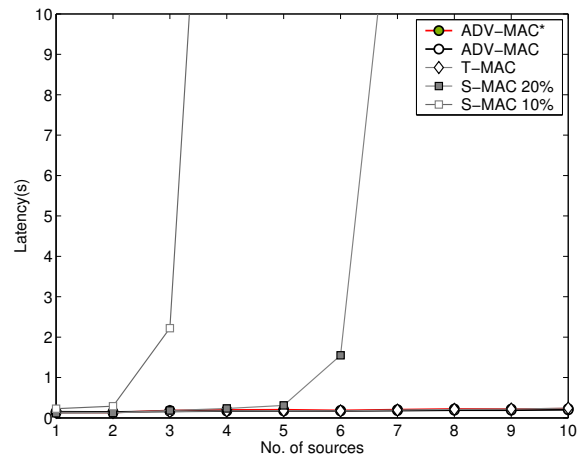
Figure 4.9: Single hop: Effect of data rate on different metrics.



(a) Average energy consumption per node per packet



(b) Packet delivery ratio



(c) Average latency

Figure 4.10: Single hop: Effect of number of sources on different metrics.

Let us consider an example where the average number of fresh nodes contending for the medium at the beginning of each frame is 5, i.e. $N_0 = 5$. Suppose, the PDR requirement is to be at least 95%. From Fig. 4.8, it is seen that the required PDR requirement is satisfied at $(S_{adv}^*, S_{data}^*) = (70 \text{ slots}, 80 \text{ slots})$. At higher values of S_{adv} and S_{data} , the PDR values are equal to or higher than 95%, however the energy consumption at each of these points are higher than that for (S_{adv}^*, S_{data}^*) .

4.3.3 Optimal ADV-MAC Performance Analysis

4.3.3.1 Effect of data rate in single hop scenario

In the first set of simulations, we investigate the effects of traffic load on energy consumption, latency and throughput. We consider an area of $50m \times 50m$ with all nodes being in transmission range of each other. There are 20 nodes in the area including 5 sources. The traffic load is varied by increasing the data rate from 0.2 pkt/sec to 1 pkt/sec . Fig. 4.9(a) shows the energy consumption per node per packet as the data rate varies. Figs. 4.9(b) and 4.9(c) show the corresponding PDR and latency results. We denote the optimized ADV-MAC protocol with a ‘*’ in the figures. S-MAC 10% denotes S-MAC with a duty cycle of 10%, and S-MAC 20% denotes S-MAC with 20% duty cycle.

From Fig. 4.9(a), it is seen that the optimized ADV-MAC protocol achieves 35 – 51% less energy consumption compared to the original ADV-MAC proposed in [38] with empirical parameter settings and a less efficient back-off mechanism. Compared to T-MAC, the energy consumption is reduced by 53 – 56%, and compared to S-MAC 10% and S-MAC 20%, the energy consumptions are reduced by 42 – 47% and 58 – 74%, respectively. As seen from Fig. 4.9(b) and Fig. 4.9(c), these energy savings are obtained without compromising the PDR or latency. The PDR is close to 1 while the latency is the least among all the protocols.

The original ADV-MAC also provides similar PDR and latency values as the optimized ADV-MAC, but the original ADV-MAC has much higher energy consumption. However, it can still achieve better results compared to S-MAC 20% and T-MAC. Its energy consumption is more than S-MAC 10%, however S-MAC 10% gives poor latency and PDR values and hence is infeasible to use for high data rates.

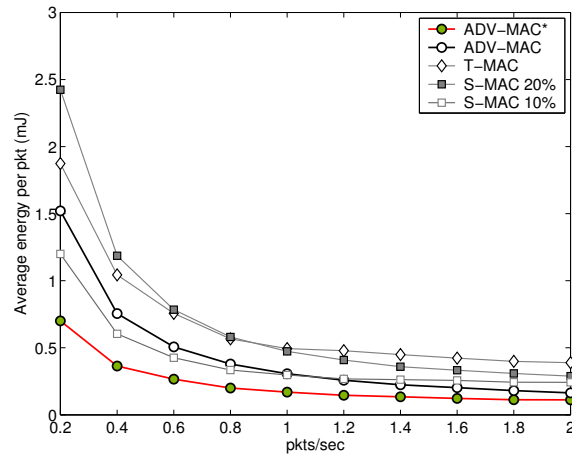
4.3.3.2 Effect of number of sources in single hop scenario

In the second set of simulations, we investigate the effect of the number of sources on the performance of the MAC protocols. The simulation setup is similar to the previous simulation set, however we vary the number of sources from 1 to 10 and keep the data rate fixed at 1 pkt/sec for all sources.

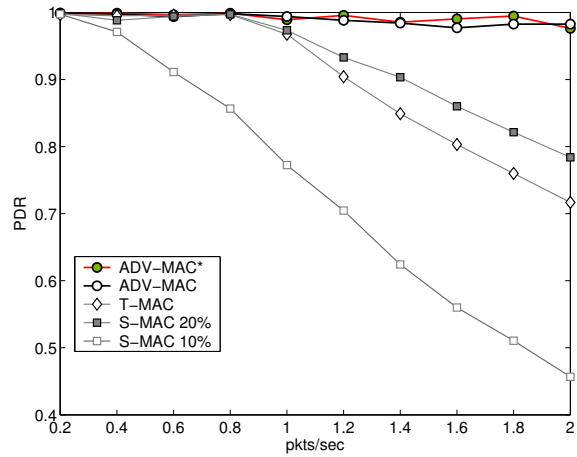
From Fig. 4.10(a), it is seen that the optimized ADV-MAC protocol has 27 – 53% less energy consumption compared to the original one. Compared to T-MAC, the energy consumption is reduced by 54 – 56%, and compared to S-MAC 10% and S-MAC 20%, the energy consumptions are reduced by 50 – 65% and 63 – 75%, respectively. As before, these energy savings are obtained without compromising the PDR or latency, as is evident from Figs. 4.10(b) and 4.10(c). Thus, the optimal ADV-MAC provides substantial energy savings while providing good PDR and latency. S-MAC 10% and S-MAC 20% have very poor PDR and latency as the number of sources increase. This happens because at most one packet can be sent in a frame in S-MAC.

4.3.3.3 Effect of data rate in multi-hop scenario

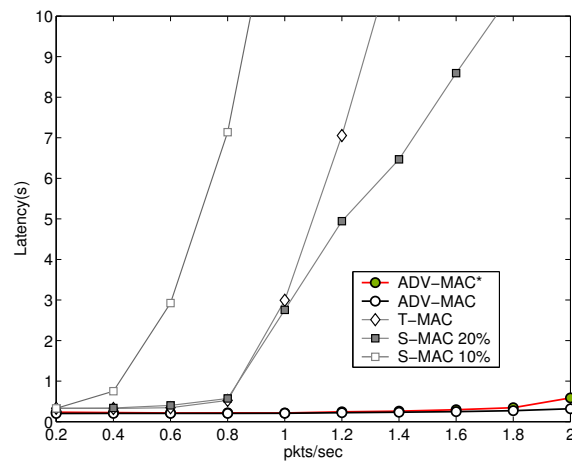
To investigate the performance of the MAC protocols in a multi-hop communication environment, we define the third simulation set where we consider an area of $700\text{m} \times 700\text{m}$ with 312 nodes, with each node having an average of 20 neigh-



(a) Average energy consumption per node per packet



(b) Packet delivery ratio



(c) Average latency

Figure 4.11: Multi-hop: Effect of data rate on different metrics.

bors in its transmission range. There are 20 source nodes, and we increase the data rate from 0.2 pkt/sec to 2 pkt/sec . On the average, each receiver will have 5 transmitting nodes in its carrier sense range. Since all nodes are not in transmission range of one another, there are hidden terminals. The analytical model is derived for single hop scenarios. However, considering the average number of transmitting nodes in the carrier sense range, we can approximate the optimal values of the ADV period and the data contention period. We use one retransmission within each frame in both ADV-MAC protocol versions. T-MAC and S-MAC also have one retransmission. Fig. 4.11(a) shows the energy consumption per node per packet as we increase the data rate. Figs. 4.11(b) and 4.11(c) show the corresponding PDR and latency.

From Fig. 4.11(a), it is seen that the optimized ADV-MAC protocol has 31–53% less energy consumption compared to the original ADV-MAC. Compared to T-MAC, the energy consumption is reduced by 62–71%, and compared to S-MAC 10% and S-MAC 20%, the energy consumptions are reduced by 41–53% and 61–71%, respectively. As seen from Figs. 4.11(b) and 4.11(c), these energy savings are obtained without compromising the PDR or latency. The PDR is close to 1 while the latency is the least among all the protocols for most data rates.

The original ADV-MAC protocol has similar PDR and latency performance as the optimized one, but its energy consumption is much higher. However, this protocol is much better than the other three protocols. S-MAC 10% is almost unusable other than at low data rates. This is because S-MAC can only accommodate one node to access the medium in one frame. S-MAC 20% performs better, and its PDR and latency do not degrade until the data rate increases beyond 1 pkt/sec . After this, its PDR and throughput degrades as it can no longer sustain the higher traffic loads. At such high loads, a higher duty cycle is needed for S-MAC. T-MAC actually performs worse than S-MAC 20% at higher traffic loads. This is primarily because of the early sleep problem described in [46]. This

degrades the PDR and latency at higher traffic loads. ADV-MAC is more immune to this problem as receiving nodes have prior knowledge of data waiting for them. Thus, they do not go to sleep, keeping the PDR and latency constant as traffic increases.

4.4 Summary

In Chapter 3, we presented ADV-MAC, a new MAC protocol for wireless sensor networks, that minimizes the energy lost due to idle listening by introducing the concept of advertising for contention. ADV-MAC provides substantial reductions in energy consumption compared to S-MAC and T-MAC while not sacrificing throughput or latency. However, the ADV period duration and the duration of the data contention period of ADV-MAC were not optimal. In this chapter, we presented an analytical model for ADV-MAC. We used simulations to verify the model and used the model to select optimal values of the ADV period and the data contention period. Simulation results show that the optimized ADV-MAC provides substantial energy savings compared to the original protocol in Chapter 3 as well as S-MAC and T-MAC while not sacrificing throughput or latency. The optimized ADV-MAC protocol adapts nicely to low and high traffic loads as well as to variable loads, in both single hop and multi-hop scenarios.

In the ADV-MAC protocol, the contention is done in two phases: first in the ADV period and then in the data period. In the next chapter, we develop a distributed TDMA based version of this protocol that replaces the second contention phase with assigned times slots, resulting in further energy savings.

5 ATMA: Advertisement-Based TDMA Protocol for Bursty Traffic in Wireless Sensor Networks

In the ADV-MAC protocol presented in Chapter 4, the contention is done in two phases: first in the ADV period and then in the data period. In this chapter, we develop Advertisement-based Time-division Multiple Access (ATMA), a distributed TDMA based version of this protocol that replaces the data contention phase with assigned times slots and utilizes the bursty nature of the traffic, resulting in further energy savings.

ATMA defines an Advertisement (ADV) period, where data slot reservations are made through ADV packets, which include the intended receiver information, and the corresponding ACK packets. This assures successful reservations and reduces the hidden terminal problem. With this approach, the most crucial sources of energy waste in medium access, namely idle listening and overhearing, are minimized, since the nodes that are not part of any transmission will go to sleep after the Advertisement period. ATMA reserves data slots for a specific time, which enables adaptation to varying traffic.

We compare the performance of ATMA to that of another TDMA-based protocol, TRAMA [31], to three contention-based protocols, Sensor-MAC (S-MAC) [50], Timeout-MAC (T-MAC) [46] and ADV-MAC as well as to a hybrid protocol Z-MAC [49]. Simulation results show that ATMA performs very well in bursty traffic, with substantial energy savings as high as 80%, while achieving the best packet delivery ratio and the best latency performances among these protocols.

The rest of this chapter is organized as follows. In Section 5.1, we discuss the design of the ATMA protocol. In Section 5.2, we present an analytical model to determine the optimal value of the advertisement period of ATMA for continuous traffic. In Section 5.3, we describe our simulation setup, followed by a detailed comparison of ATMA with S-MAC, T-MAC, ADV-MAC, TRAMA and Z-MAC using extensive simulations. Finally, in Section 5.4 we conclude the chapter.

5.1 ATMA Design Overview

In ATMA, time is divided into frames. Each frame begins with a SYNC period, followed by an advertisement period and ending with a data period. The SYNC period is used for loose synchronization between nodes. The advertisement period is contention-based, and used for advertising for data and for reserving data slots. The data period is divided into slots for data exchange and is accessed in a contention free manner.

The method of synchronization is the same as in S-MAC [50]. A frame is on the order of a second, which is 10^4 times normal clock drifts. The SYNC and Advertisement periods are contention-based, and, as such, these small drifts do not prevent the exchange of packets. To minimize the effect of these clock drifts in the contention-free data period, we set the data slots to be slightly larger than the duration to transmit a data and an ACK packet. Also, the sending nodes begin transmitting after a small offset. Thus, small clock drifts do not effect the

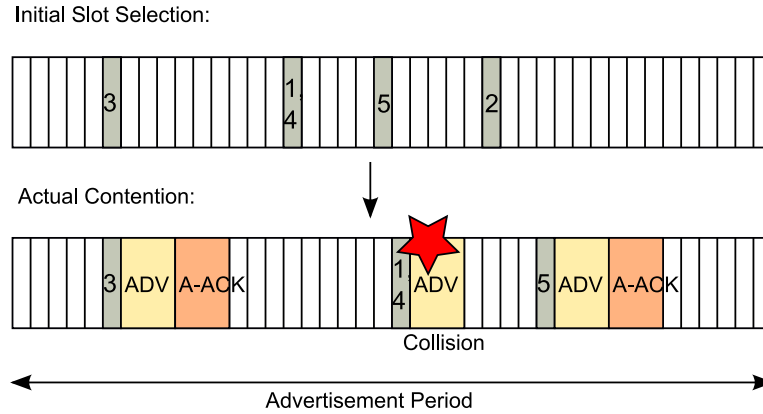


Figure 5.1: Examples of the operation of the advertisement period. The numbers in the slots indicate the IDs of nodes choosing those slots.

synchronization.

5.1.1 Advertisement Period

The advertisement (ADV) period is divided into many small slots, the size of which depends on the clock resolution. The size of each ADV packet is a multiple of the slot duration. Each ADV packet contains the ID of the receiver and the chosen data slot number of the data period. All nodes are awake during this period. Each node having data to send randomly selects a slot at the beginning of the ADV period and initializes a timer to a value equal to $slot\ number \times slot\ duration$. When the timer reaches zero, the node transmits an ADV packet and waits for an acknowledgement from the intended receiver.

If the intended receiver receives the ADV packet successfully, it replies with an advertisement-acknowledgement packet, or A-ACK, that contains its own ID and the data slot number. Successful transmission of an ADV and its A-ACK will ensure that all nodes in the two-hop neighborhood are aware of which data slots are being used. Hence, different senders will choose unique slots, preventing collisions in the data period. If an intended receiver node receives an ADV packet

and knows that the selected slot is already occupied by another node, it does not send an A-ACK.

If a node that is waiting to transmit an ADV hears another ADV transmission, it will freeze its timer and wait for the entire transmission (ADV and A-ACK) to be over. Then, it will resume its timer again and transmit its ADV packet when the timer expires if there is time left in the ADV period for its ADV packet transmission and the corresponding A-ACK packet.

Fig. 5.1 shows an example of the operation of the ADV period. In the example, nodes with index numbers from 1 to 5 select random slots and initialize their timers. Node 3 selects slot 6, nodes 1 and 4 both select slot 16, and so on. When the timer of node 3 reaches zero, it transmits its ADV packet with the ID of the intended receiver and the chosen data slot number. Other nodes will hear this transmission and will freeze their timers. Since no node chooses slot 6 other than node 3, its transmission will be a success, and it will receive back an A-ACK from its intended receiver. All nodes will resume their timers after the end of this A-ACK transmission. Nodes 1 and 4 chose the same slot and hence their ADV packets possibly collide. If a collision happens at an intended receiver, no A-ACK will be received by the corresponding transmitter. After these transmissions, node 5 will transmit successfully, but node 2 will have no time left in the ADV period to complete its transmission, and thus it will postpone its transmission to the next frame.

We can take advantage of bursty traffic to reduce the overhead in the ADV period and hence reduce the size of the ADV period to save energy. When a node needs to transmit packets in a burst, one single ADV/A-ACK exchange may reserve the same data slot for L_R consecutive frames, and, hence, the transmitting node need not send an ADV for each data packet. This can greatly reduce the traffic in the ADV period, saving energy. The number of consecutive frames for which a data slot is reserved can be fixed or can be announced in the ADV and

A-ACK packets. In this work, we consider a fixed value of L_R . Nodes may have fewer than L_R packets. In that case, the receiver will timeout and go to sleep in the selected data slot. This technique also works with periodic traffic, where a single ADV/A-ACK exchange can reserve a particular data slot, and the transmitter node can renew its channel access periodically.

5.1.2 Data Period

The data period is divided into longer slots compared to the ADV period. Each slot is large enough to transmit a data packet and an ACK. If a transmitting and a receiving node exchange ADV and A-ACK packets successfully, they will wake up at the beginning of their chosen slot to exchange data. In all other slots, they will be asleep. Nodes that do not have any data to send or receive will be asleep during the entire data period. Thus, the energy waste reduced is significant, since overhearing is completely avoided and idle listening is reduced considerably.

New nodes joining the medium will initially have no information about the available data slots. If such a node has data to send, it can stay awake in the data period to find an un-occupied slot. It may attempt to reserve that slot in the next frame. If, however, the intended receiver knows that this slot is already reserved by another node (not in the transmitter's range), the intended receiver will simply not reply to the ADV packet.

5.1.3 Hidden Terminal Interference Problem

Successful exchange of ADV/A-ACK packets guarantees that no node in the transmission range of either the sender or the receiver is using that slot. This, however, does not guarantee that a node in the carrier sense range of either the sender or the receiver or both, is not using that data slot. This situation is shown in Fig. 5.2. Node A and node B are in transmission range of one another, and node A is

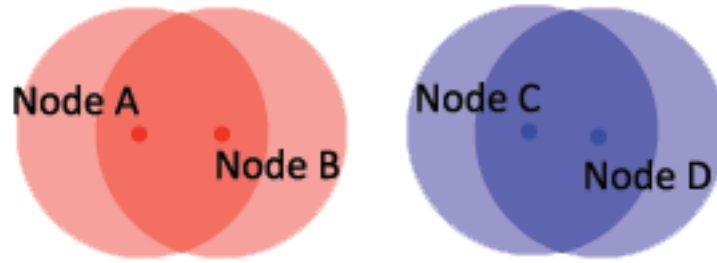


Figure 5.2: Example of packet corruption by simultaneous transmissions in the carrier sense range. Circles around each node denote their transmission ranges. All nodes are in the carrier sense range of each other.

transmitting a packet to node B. Simultaneously, node C is transmitting a packet to node D. Node C and node D are not in the transmission range of node A or node B, but they are in each other's carrier sense range. As a result, node A's data packet might get corrupted and node B will not send an ACK. Even if the data packet is received correctly, node B's ACK packet may get corrupted. In either case, node A will not receive an ACK successfully and will regard this as an unsuccessful transmission. The same thing may occur in the case of node C and node D. Also, since each pair of nodes has reserved the same data slot for L_R consecutive frames, all L_R transmissions will be unsuccessful for each pair of nodes. This will reduce PDR in multi-hop cases.

To prevent this, as soon as a node does not successfully get an ACK, it will retain that data packet and resend its ADV in the next frame with a request to reserve a new data slot. This substantially improves PDR in multi-hop networks.

5.2 Optimization of ADV Period

In this section, we model the packet delivery ratio of ATMA which can then be used to find the optimal ADV period.

5.2.1 Packet Delivery Ratio Analysis

To determine the packet delivery ratio, we first calculate the average throughput for a single frame given that there are N nodes contending. For convenience, important notations used in the analysis are presented in Table 5.1. We consider a perfect channel. We assume that all nodes have data to send continuously. If a node has data to send with no prior reservation, and then fails to exchange ADV/A-ACK packets successfully, it will drop its current packet. The node will contend again in the next frame, as it will get a fresh packet in the next frame.

Let us consider N nodes competing to transmit in the ADV period. To calculate PDR, we need to find the number of successful ADV/A-ACK exchanges in the ADV period. This derivation is similar to the one in our previous work for ADV-MAC in Chapter 4. However, in this chapter, t_{adv} is used to denote one whole ADV/A-ACK exchange instead of one ADV packet.

Let S_{adv} denote the duration of the ADV period (in unit *slots*). Initially, all N nodes select a slot out of the first $S = S_{adv} - t_{adv}$ slots of the ADV period to be able to finish their transmission before the end of the ADV period. To be able to calculate the number of nodes that successfully exchange ADV/A-ACK packets, we need to find the number of nodes that transmit ADV packets in the ADV period, either successfully (with no collision) or unsuccessfully (with collision). This is a random variable, which we denote by X , and its corresponding value by x . Hence, $x \leq N$. Assume that x nodes choose g distinct slots out of the S_{adv} slots, where $g \leq x$. If $g = x$, there will be no collisions.

Let $\Theta(x, g)$ denote the number of possible ways to assign x nodes to g slots. Mathematically, this problem is the same as the number of different distributions when placing x distinguishable balls into g indistinguishable bins such that each bin has at least one ball. Hence, the expression of $\Theta(x, g)$ is found to be

$$\Theta(x, g) = \frac{1}{g!} \sum_{s=0}^{g-1} (-1)^s \binom{g}{g-s} (g-s)^x. \quad (5.1)$$

Proof of (5.1) are given in Chapter 4.

Also, let $\xi(x, g)$ denote the number of slot selections by N nodes that result in g distinct slots to be selected, where x nodes can transmit, and $N - x$ nodes defer their transmission due to insufficient time in the ADV period. Expression and proof of $\xi(x, g)$ is given in Chapter 4. Since x nodes may be chosen out of N nodes in $\binom{N}{x}$ ways, the number of slot selections that result in x transmitting nodes and $N - x$ deferring nodes, $\omega(x, N)$, is given by

$$\omega(x, N) = \binom{N}{x} \sum_{g=1}^{g_{max}(x)} \Theta(x, g) \xi(x, g), \quad (5.2)$$

where $g_{max}(x)$ denotes the maximum g value, i.e., the maximum number of distinct slots used by x transmitting nodes. If t_{adv} denotes the duration of an ADV/A-ACK exchange, and S_{adv} denotes the duration of the ADV period (both in units of *slots*), then

$$g_{max}(x) = \min \left(x, \left\lfloor \frac{S_{adv}}{t_{adv} + 1} \right\rfloor \right). \quad (5.3)$$

Note that we divide S_{adv} with $(t_{adv} + 1)$, since an ADV/A-ACK exchange requires one preceding empty slot. Since all N nodes select a slot out of the first S slots of the ADV period, the number of all possible slot selections by N nodes is S^N . Consequently, the probability of having x transmitting nodes out of N nodes, $P(X = x | N)$, can be written as

$$\begin{aligned} P(X = x | N) &= \frac{\omega(x, N)}{S^N} \\ &= \frac{1}{S^N} \binom{N}{x} \sum_{g=1}^{g_{max}(x)} \Theta(x, g) \xi(x, g). \end{aligned} \quad (5.4)$$

The proof of (5.4) is given in Chapter 4. The probability for a slot selected by a node to be collision-free is

$$P_{adv-nc} = \frac{S(S-1)^{N-1}}{S^N}. \quad (5.5)$$

Table 5.1: ATMA: Notations Used

Symbol	Description
$g_{max}(x)$	Maximum number of distinct slots for x transmitting nodes
L_R	Reservation length (in number of frames)
N_{TX}	Total no. of nodes having data to transmit in one hop neighborhood
N	No. of nodes contending at the beginning of a frame
PDR	Packet delivery ratio
S_{adv}	Duration of ADV period (in slots)
t_{adv}	Duration of one ADV/A-ACK exchange (in slots)
X	Random variable denoting no. of nodes that transmit in ADV period, i.e., no. of nodes contending in the data period
x	Value assumed by random variable X
\bar{X}	Expected value of X
X_s	No. of nodes that transmit with success in ADV period
$\Theta(x, g)$	No. of ways to assign x nodes to g slots
ρ	Throughput of a single frame in steady state
$\omega(x, N)$	No. of ways of slot selection that give x transmitting nodes and $N - x$ deferring nodes

Thus, X_s , the expected number of nodes that transmit successfully out of X transmitting nodes is given by

$$X_s = \overline{X}P_{adv-nc}, \quad (5.6)$$

where \overline{X} represents the expected value of X and can be calculated as

$$\overline{X} = E[X|N] = \sum_{x=1}^N xP(X = x|N). \quad (5.7)$$

Now, let us assume that N_{TX} nodes have data every frame. Of these N_{TX} nodes, only N nodes contend, and the remaining $N_{TX} - N$ nodes have reserved slots in the data period. X_s of the N nodes successfully complete ADV/A-ACK exchanges. We assume that nodes that get slots in a data period always transmit successfully. Also, we assume that nodes drop packets if they get no access to the medium. Thus, the throughput is given by

$$\rho = \frac{X_s + (N_{TX} - N)}{N_{TX}}. \quad (5.8)$$

The number of contending nodes, N , is actually a function of N and is given by

$$N = N(\rho) = (1 - \rho)N_{TX} + \rho \frac{N_{TX}}{L_R}, \quad (5.9)$$

where L_R is the number of consecutive frames for which a slot is reserved, which is referred as *reservation length* in the rest of the chapter. The first term in (5.9) denotes the failed contenders in the previous frame, and the second term is the expired reservations. Thus, in steady state, we can write

$$\begin{aligned} \rho &= \frac{X_s + (N_{TX} - N(\rho))}{N_{TX}} \\ \Rightarrow \rho - \frac{X_s + (N_{TX} - N(\rho))}{N_{TX}} &= 0. \end{aligned} \quad (5.10)$$

Since nodes drop packets if they do not get access to the medium, and every frame has N_{TX} nodes with data, the solution of (5.10) gives the steady state PDR.

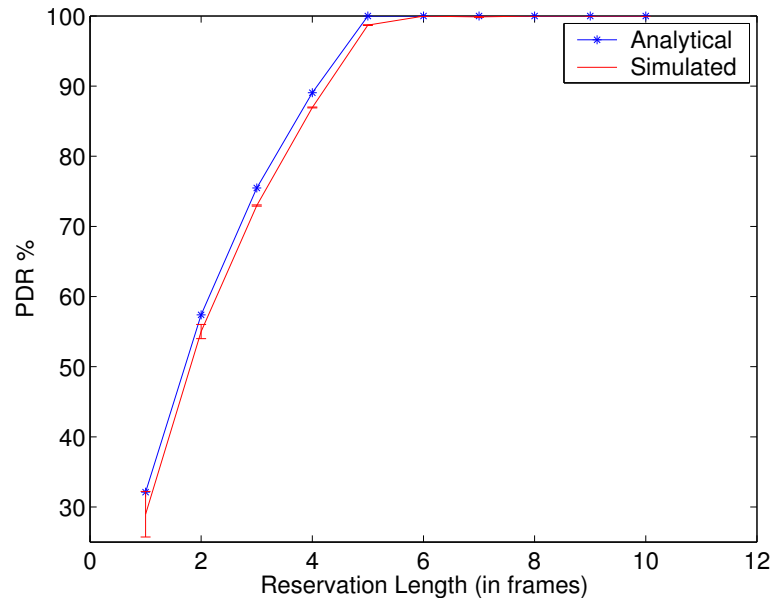


Figure 5.3: PDR values obtained using the derived analysis and ns-2 simulations for $N_{TX} = 5$. Confidence intervals of 95% are shown for the simulation results.

5.2.2 Verification of Analysis

To verify the mathematical model, we calculated the packet delivery ratio (PDR) for different values of the reservation length using the proposed model. We also performed ns-2.29 simulations with the same scenarios, and the simulation results are compared to the analytical results. We consider 20 nodes in an area of $50m \times 50m$. There are 5 sources, each transmitting 4.24 pkt/s (one packet every frame). The ADV period is fixed at $5ms$. The PDR results are shown in Fig. 5.3. Each simulation point in Fig. 5.3 is the average of 100 runs. The 95% confidence intervals of the simulations are also shown in the figures.

From Fig. 5.3, we can see that the curves from the analytical model follow the simulated curves very closely. We can also see that the PDR jumps to 100% when the reservation length is greater than or equal to 5. At lower reservation length values, nodes fail to transmit their ADV packets due to inadequate remaining time in the ADV period, and they drop their current packet, which results in the

corresponding PDR drop.

5.2.3 Optimal Duration of ADV Period and Reservation Length

The PDR and energy consumption values of ATMA depend on two parameters: S_{adv} , or the ADV period duration, and L_R , or the reservation length. If S_{adv} is smaller than optimal, the PDR will be poor, and if S_{adv} is greater than optimal, it will cause energy waste. Similarly, smaller than optimal L_R values provide poor PDR, and greater than optimal L_R values will leave less un-reserved slots. This prevents new nodes from accessing the medium and will reduce PDR. We use the analytical model to calculate the PDR for different values of both S_{adv} and L_R . The lowest value of S_{adv} for which a given PDR requirement is met is considered to be the optimum S_{adv} value. The corresponding L_R is the optimal L_R value. This set of optimal values will give the lowest energy consumption, as the duration of the ADV period is the least, and will also meet the given PDR requirement. From Fig. 5.4, we see that for a given N_{TX} of 5, the minimum value of S_{adv} for which we get above 98% PDR is 50 slots. The corresponding L_R is 5. Thus, using the analytical model derived for PDR, we can find the energy optimal duration of the ADV period for a given N_{TX} and a PDR constraint.

5.3 ATMA Performance Evaluation

In our experiments, we compared the performance of ATMA with three contention-based protocols: S-MAC, T-MAC and ADV-MAC, and one TDMA-based protocol: TRAMA. We used energy consumption, PDR and latency as the three performance metrics for comparison. For the static multi-hop cases we also compare

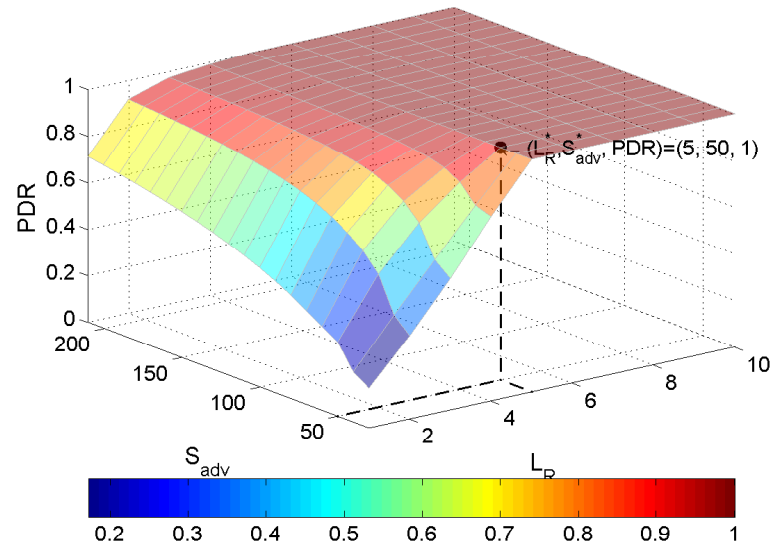


Figure 5.4: PDR values obtained using the derived analysis for different S_{adv} and L_R values for $N_{TX} = 5$. The optimal point for energy minimization is shown for a given PDR constraint of at least 98%.

ATMA with Z-MAC. However, we only show the PDR and latency values as the Z-MAC ns-2 code does not support energy tracing.

IEEE 802.15.4 is another well known hybrid protocol. However there are no known ns-2 code for the protocol. But from Section 2.4.1 we can say qualitatively that the protocol is not without its own drawbacks. The very flexible nature of 802.15.4, which is an advantage of the protocol, adds several other disadvantages. The nodes belonging to a cluster in 802.15.4 cannot communicate directly to other nodes in the network. This communication must proceed through the cluster-head. Depending on whether the destination node is in the same cluster or not, the cluster-head will forward the message to the node or to the cluster-head of the node. This adds to the delay compared to ATMA, where nodes communicate directly if they are in communication range.

Another disadvantage of 802.15.4 is its lower throughput compared to ATMA. 802.15.4 cannot utilize the entire active period of 16 slots for TDMA access. It can only allocate the last 7 slots for TDMA access because it needs to provide for contention based channel access in all super-frames. Thus even if nodes in a cluster produce data at high rates, which is best addressed by allocating all of the 16 slots to TDMA access, 802.15.4 can only allocate a maximum of 7 slots. This in turn reduces throughput and also increases latency. This problem does not arise in ATMA, as it uses its entire data period for TDMA access.

Since all communication must be done through the cluster-heads, they also end up depleting their energy faster than the other nodes in the cluster. This problem of non-uniform energy consumption may lead to a disconnected network, which is not a problem in ATMA.

Table 5.2: ATMA: Simulation Parameter Values

Parameter	Value
Duration of ADV period of ADV-MAC	15ms
Duration of ADV period of ATMA	5ms
Duration of frames	236.4ms
Duration of one ADV slot	0.1ms
Duration of data slots of ATMA	12ms
Duration of data slots of TRAMA	14ms
Duration of random access period of TRAMA	72 data slots
Duration of SYNC period	8.4ms
Duration of control packet	0.9ms
Duration of data packet	8.5ms
Rx/Idle listening power	59.1mW
Tx power	52.2mW
Total simulation time	200s
Transmission rate	250Kbps
Transmission range	100m
Carrier sense range	200m

5.3.1 Simulation Setup

We performed all simulations in ns 2.29 [34]. The S-MAC code is included in this version of ns. We coded T-MAC, ADV-MAC, TRAMA and ATMA in ns-2 as well¹. We used the Z-MAC code made available by the authors². In our TRAMA implementation, we assumed that each node has a 100% correct one-hop neighbor

¹Ns code for ATMA, T-MAC, ADV-MAC and TRAMA are available at <http://www.ece.rochester.edu/projects/wcng/>.

²Z-MAC code available at <http://www4.ncsu.edu/rhee/export/zmac/software/zmac/zmac.htm>

information, instead of implementing the random access period. We used different duty cycle settings for S-MAC, specifically duty cycles of 10% and 20%, because one fixed duty cycle is not suitable for all traffic loads investigated. The frame time for 10% duty cycle of S-MAC is $236.4ms$, and we set this as the frame time for ATMA, T-MAC and ADV-MAC as well.

The random access period of TRAMA is fixed to 72 transmission slots and is repeated once every 10,000 transmission slots as in [31]. The duration of one transmission or data slot of TRAMA is set to $14ms$. This is higher than ATMA as the data packet header is much larger for TRAMA due to the winning slot bitmaps and fields such as number of winning slots and timeout. The *schedule interval* of TRAMA is set to 200 transmission slots for single hop and 400 transmission slots for multi-hop scenarios. In multi-hop scenarios, a longer duration of *schedule interval* is used, because each node has more nodes in its two-hop neighborhood. Hence, a smaller duration may not result in that every node owns at least one slot within a schedule interval.

The owner contention window size (T_o) and non-owner contention window size (T_{no}) of Z-MAC are set to 8 and 32 respectively as in [49]. Since Z-MAC uses the LPL interface of B-MAC, it uses preamble sampling, i.e., a node sends a preamble before sending the data. Receivers must wake up at intervals smaller than the preamble to be able to receive packets. Thus, all nodes in a two hop neighborhood must wake up at intervals smaller than the preamble in all slots so that they can receive data that may be destined for them. In ATMA, senders and receivers know the slots in which they are supposed to exchange data and hence need not wake up in all slots. Also there is no contention to send data in a slot in ATMA. The combination of these two factors should lead to lower power consumption for ATMA than Z-MAC in multi-hop cases.

When a new node joins the network in Z-MAC, the DRAND algorithm is run locally to compute the new period for the nodes in that neighborhood, and this

information is then propagated throughout the network. This process requires significant energy expenditure. No such procedure is required for ATMA.

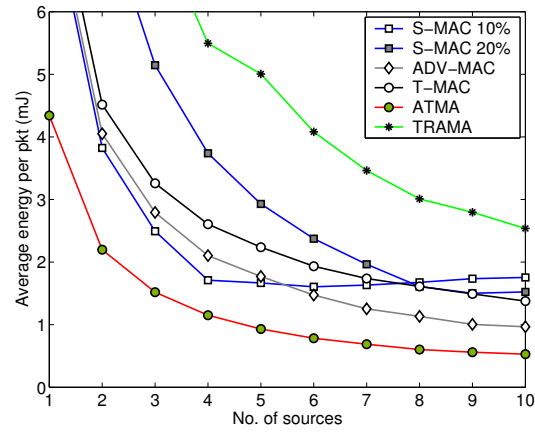
In a single hop scenario, Z-MAC may provide faster access to the medium, as nodes can use any slot, which is not possible in ATMA. This will result in smaller delays in Z-MAC with similar PDR values. Hence we only compare ATMA with Z-MAC in multi-hop cases as Z-MAC provides similar results in single hop cases.

Based on MicaZ specifications [1], the transmission rate is set to $250Kbps$, and the transmission range is set to $100m$, while the interference and carrier sense range is set to be two times the transmission range, which is $200m$. We use the power consumption values of MicaZ nodes. The current consumption values of MicaZ nodes are $17.4mA$ and $19.1mA$ for transmission and reception, respectively. Considering a battery of 3 V, the transmission and reception powers are $52.2mW$ and $59.1mW$, respectively. The idle power consumption is the same as that for reception.

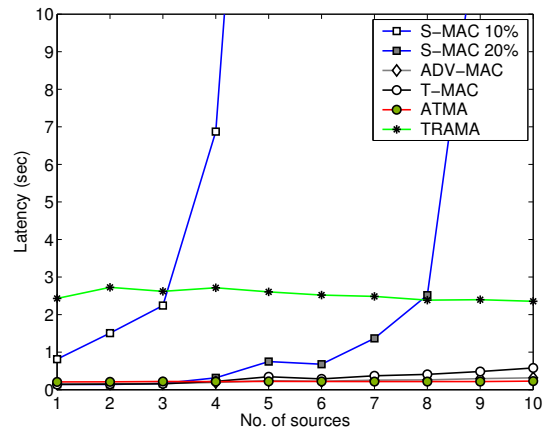
The ADV period and the data contention period are divided into slots of $0.1ms$ each. For the multi-hop simulations, the nodes are deployed in the target area with a uniformly random distribution. We use T-MAC with over-hearing avoidance, since it is used in the simulations in [46], which achieves more energy savings. The simulation time is 200s, and each point in the figures is the average of 50 runs. The confidence intervals for the simulation results are observed to be very small, and hence are not shown in the figures for the sake of brevity. The values used for the simulation parameters are summarized in Table 5.2. In all experiments, a source node generates one packet in each frame within a burst.

5.3.2 Effect of Number of Sources, Single Hop Scenario

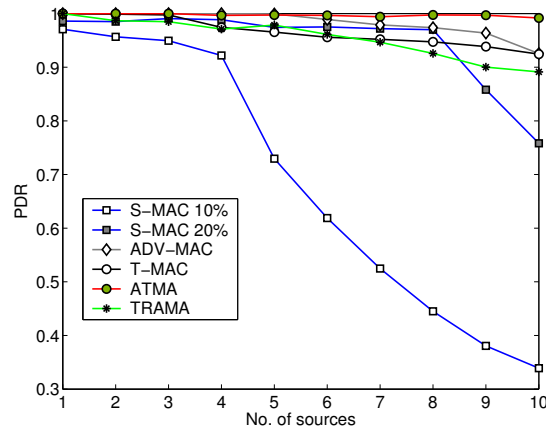
In the first set of simulations, we investigate the effect of the number of sources on the performance of the MAC protocols. We consider 20 nodes in an area of



(a) Energy Consumption vs. No. of Sources.



(b) Latency vs. No. of Sources.



(c) PDR vs. No. of Sources.

Figure 5.5: Single hop scenario. Performance as a function of the number of sources for ATMA, ADV-MAC, T-MAC, S-MAC and TRAMA.

$50m \times 50m$ with all nodes being in the transmission range of each other, and we vary the number of sources from 1 to 10. Each source randomly transmits data in bursts of $3.5s$ at intervals of $20s$. Within a burst, each node generates one packet per frame, leading to a packet arrival rate of 4.24 pkt/s . In [38], the duration of the advertisement period of ADV-MAC is fixed at $15ms$, which is the optimal value for 5 fresh packets every frame, according to the analytical model derived. This value provides an acceptable performance for different numbers of sources. The duration of the advertisement period of ATMA is fixed at $5ms$, which was found to be an adequate value for different numbers of sources. The length of the frame reservation, L_R , is set to 5 frames.

Figs. 5.5(a), 5.5(b) and 5.5(c) show the energy consumption, latency and PDR results of the investigated protocols, respectively. As seen in the figures, ATMA requires the minimum energy consumption per packet received successfully among all of the protocols, with the lowest latency and the highest PDR. The PDR of ATMA is almost 100%, with latency around $200ms$. From Fig. 5.5(a), we can see that ATMA consumes 43 – 45% less energy than ADV-MAC, 45 – 61% less energy than T-MAC, 44 – 75% less energy than S-MAC 10% and 64 – 70% less energy than S-MAC 20%. The reduced energy consumption of ATMA is due to a smaller duration of the advertisement period, which is possible because one ADV/A-ACK exchange can reserve a slot for $L_R = 5$ consecutive frames. This greatly reduces energy consumption. Compared to TRAMA, the energy consumption of ATMA is 78 – 80% less. These high energy savings occur, since TRAMA defines each node to send its schedule once in every *schedule interval*. Also, since a schedule interval duration is $200 \times 14ms = 2.8s$, the latency of TRAMA is around $2s$. Thus, it is seen that ATMA can adapt well to varying aggregate loads, providing the minimum energy consumption with high PDR and low latency values for bursty traffic scenarios.

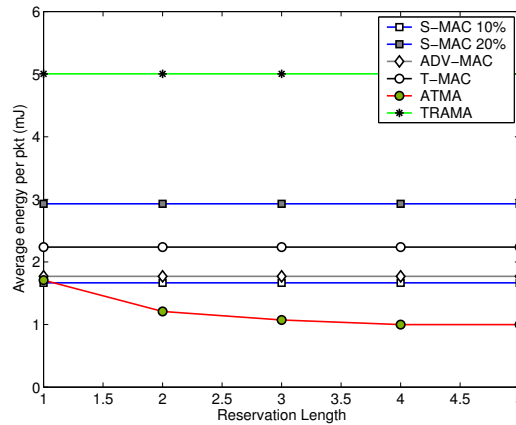


Figure 5.6: Single hop scenario. Average energy per packet as a function of data reservation length.

5.3.3 Effect of Reservation Length, Single Hop Scenario

In the second set of simulations, we investigate the effect of reservation length on the performance of ATMA. Only ATMA depends on the reservation length. We consider the same area of $50m \times 50m$ with all nodes being in the transmission range of each other, and we set the number of sources to 5. We vary the reservation length from 1 to 5 frames. As before, each source will randomly transmit data in bursts of $3.5s$ at intervals of $20s$. Within a burst, each node generates one packet per frame. The duration of the advertisement period of ADV-MAC is fixed at $15ms$ as before. The duration of the ADV period of ATMA is varied from $15ms$ to $5ms$ for acceptable PDR values. We need longer ADV periods for smaller reservation lengths due to increased contention.

From Fig. 5.6, we can see that initially, ATMA has the same energy consumption as ADV-MAC. This is because the ADV period duration of both protocols are $15ms$ initially, and a successful ADV/A-ACK exchange only reserves a data slot in that frame. ATMA requires a larger ADV period for reservation length 1, as it needs to reserve a data slot in each frame for each packet in the burst. This creates high contention overhead. As the reservation length increases, the duration

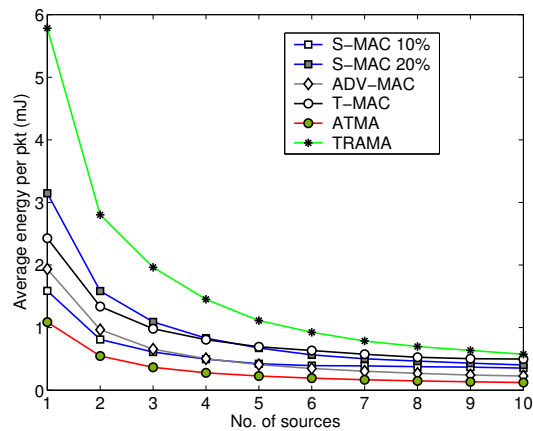
of the ADV period decreases, and the energy consumption decreases. Starting from reservation length 4, the energy consumption per packet becomes constant for ATMA, as a duration of $5ms$ is enough for the ADV period. This constant behavior is observed to continue for higher reservation lengths than 5 frames, which we omit in the figures. The latency and PDR of ATMA stays around $200ms$ and 100%, respectively, for all reservation length values investigated and are not shown.

5.3.4 Effect of Number of Sources, Multi-hop Scenario

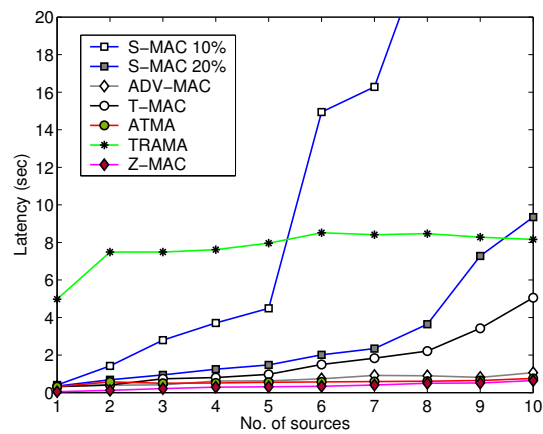
In the third set of simulations, we investigate the effect of the number of sources on the performance of the MAC protocols in a multi-hop scenario. We consider 150 nodes in $700m \times 700m$. The sources are deployed uniformly randomly in the area. We vary the average number of sources in the carrier sense range of any node from 1 to 10. For example, to get an average of 5 sources in the carrier sense range of any node, we set the total number of sources to 20, which are randomly placed in the area of $700m \times 700m$. Thus, for any arbitrary node, the average number of sources in its carrier sense range of $200m$ is:

$$\begin{aligned} \text{Total no. of sources} \times \frac{\text{Area of CS range}}{\text{Total simulation area}} \\ = 20 \frac{\pi(200)^2}{(700)^2} = 5.12 \approx 5. \end{aligned}$$

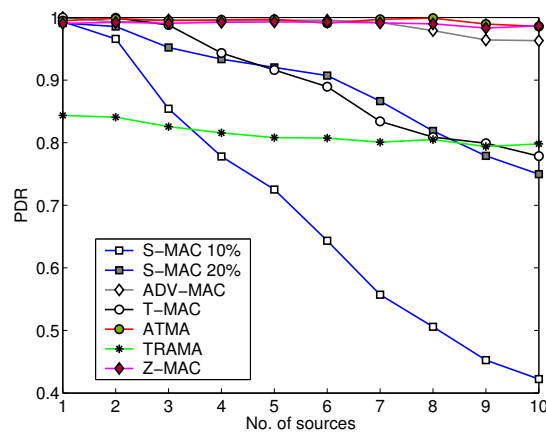
In the multi-hop scenarios, each source chooses a random destination in its one hop neighborhood. Thus, a routing protocols is not needed for the transmission of packets. It is important not to use a routing protocol while testing a MAC protocol because the performance of the MAC protocol will be affected by the type of routing protocol used. As before, each source randomly transmits data in bursts of $3.5s$ at intervals of $20s$. The duration of the advertisement period of ADV-MAC is fixed at $15ms$, whereas the duration of the same for ATMA is fixed at $5ms$. The length of the frame reservation is set to 5 frames.



(a) Energy vs. No. of Sources.



(b) Latency vs. No. of Sources.



(c) PDR vs. No. of Sources.

Figure 5.7: Multi-hop scenario. Performance as a function of the number of sources for ATMA, ADV-MAC, T-MAC, S-MAC, TRAMA and Z-MAC (energy not included for Z-MAC).

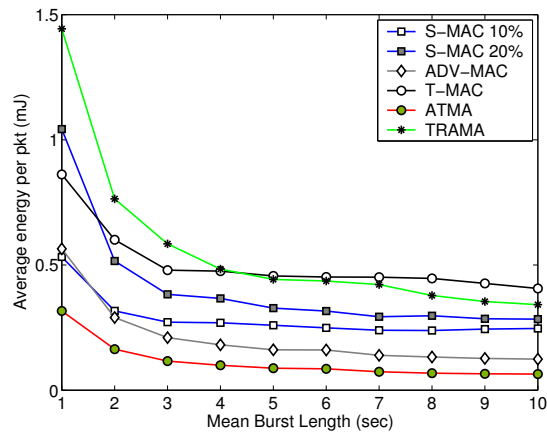
The energy consumption, latency and PDR results of the protocols are shown in Figs. 5.7(a), 5.7(b) and 5.7(c), respectively. As seen in the figures, ATMA requires the minimum energy consumption per packet among all of the protocols, while achieving the lowest latency and the highest PDR at the same time. The PDR remains above 98% with latency below 800ms. From Fig. 5.7(a), we can see that ATMA consumes 43–45% less energy than ADV-MAC, 55–75% less than T-MAC, 31–65% less than S-MAC 10%, 65–69% less than S-MAC 20% and 78–81% less than TRAMA. Also, since a schedule interval duration is $400 \times 14 \text{ ms} = 5.6 \text{ s}$, the latency of TRAMA is around of 7 s. From Figs. 5.7(a), 5.7(b) and 5.7(c), we can also see that T-MAC suffers from the *early sleeping problem* in the multi-hop case. As a result, the PDR decreases and the latency increases due to hidden terminals. Overall, it is seen that ATMA can adapt to varying loads quite well in multi-hop scenarios, providing the minimum energy consumption with a high PDR of almost 100% and low latency.

From Figs. 5.7(b) and 5.7(c), it is seen that Z-MAC has performance similar to ATMA with respect to latency and PDR. However, in the ns-2 code for Z-MAC, as provided by the authors of the protocol, the two-hop slot assignments are done offline and fed into the simulation, providing all nodes the optimal slot allocation. In a real implementation of Z-MAC, this would not be possible. A non-optimal slot allocation will result in collisions and would in turn increase the delay and decrease the PDR. The ATMA code computes the slot ownership on the fly and requires no such assumption. We do not show the energy consumption of Z-MAC as the ns-2 code for Z-MAC does not provide energy tracing. However, as explained previously, the energy consumption of ATMA will be less than Z-MAC, as nodes in Z-MAC wake up for all slots to do preamble sampling.

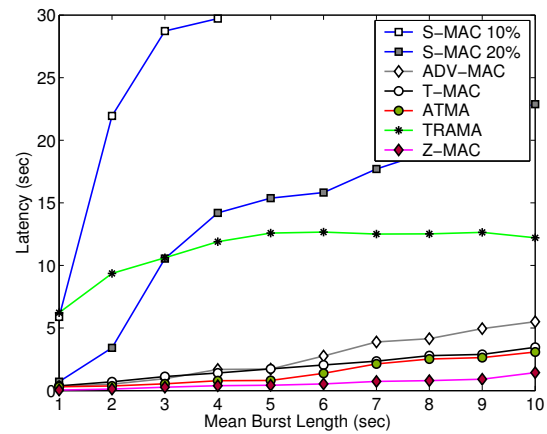
5.3.5 Effect of Random Burst Length on Multi-hop Scenario

In the fourth set of simulations, we explore a completely random traffic pattern with random burst lengths and inter-burst durations in a multi-hop scenario. We consider 150 nodes in an area of $700m \times 700m$. As before, the nodes are deployed uniformly randomly in the area. We fix the average number of sources in the carrier sense range of any node to 10, i.e., a total of 39 sources selected randomly from the nodes in the simulation area. Each source randomly transmits data in bursts. The durations of these bursts are set to be exponentially distributed. We vary the mean length of the burst from 1s to 10s. The interval between bursts is also exponentially distributed with a mean of 15s. Because the burst lengths as well as the inter-burst durations are exponentially distributed, this simulates an event-driven traffic pattern. As before, the duration of the advertisement period of ADV-MAC is fixed at 15ms, the duration of the same for ATMA is fixed at 5ms, and the length of the frame reservation is set to 5 frames.

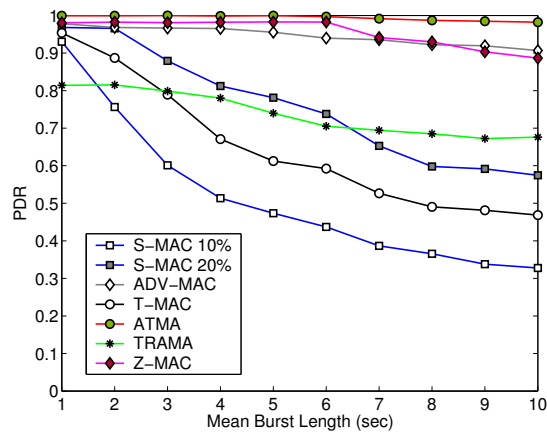
The energy consumption, latency and PDR results of the protocols are shown in Figs. 5.8(a), 5.8(b) and 5.8(c), respectively. As seen in the figures, ATMA requires the minimum energy consumption per packet among all of the protocols, while achieving the lowest latency and the highest PDR at the same time. The PDR is almost 100% with latency below 800ms. From Fig. 5.7(a), it is seen that ATMA consumes 44 – 47% less energy than ADV-MAC, 63 – 84% less than T-MAC, 40 – 73% less than S-MAC 10%, 69 – 77% less than S-MAC 20% and 78 – 81% less than TRAMA. As before, the latency of TRAMA is as high as 12s due to its schedule interval duration and high traffic load. T-MAC suffers heavily from the early sleeping problem, with poor PDR and energy consumption values. ADV-MAC gives good PDR, but has higher energy consumption and latency values than ATMA. ATMA maintains a near 100% PDR with the lowest



(a) Energy vs. Mean Burst Length.



(b) Latency vs. Mean Burst Length.



(c) PDR vs. Mean Burst Length.

Figure 5.8: Multi-hop scenario. Performance as a function of mean burst length for ATMA, ADV-MAC, T-MAC, S-MAC, TRAMA and Z-MAC (energy not shown for Z-MAC).

latency and energy consumption values. Thus among all the compared protocols, ATMA adapts the best also to dynamic and bursty traffic with a near perfect packet delivery ratio and the lowest energy consumption and latency.

From Figs. 5.8(c) and 5.8(b), it is seen that ATMA gives better PDR performance than Z-MAC and similar latency performance as Z-MAC in this realistic multi-hop scenario, even with predetermined optimal slot allocations for Z-MAC. The PDR of ATMA is as much as 10% better than Z-MAC even with Z-MAC's optimal slot allocation. The reason behind this PDR drop of Z-MAC under predetermined slot allocation may be due to increased contention due to the ECN messages as well as collisions in the data slots because of contention. Again, for the energy consumption, we can conclude that the energy consumption of ATMA would be less than Z-MAC, as nodes in Z-MAC wake up for all slots to do preamble sampling.

5.4 Summary

This chapter presents ATMA, a new TDMA-based MAC protocol for wireless sensor networks. ATMA uses a contention-based advertisement period and a TDMA-based slotted data period. Nodes exchange ADV/A-ACK packets in the ADV period to reserve slots for data exchange in the data period. Because of the ADV/A-ACK, the data slot is allotted exclusively to one pair of nodes in a two-hop neighborhood, which provides collision-free exchange of data. Nodes that are not a part of the data exchange sleep in the data period, saving energy. To take advantage of bursty traffic, each ADV/ADV-ACK exchange will reserve the same slot in L_R consecutive frames. The value of L_R is a system parameter that can be determined analytically. Because a data slot is reserved for L_R consecutive frames, overhead in the ADV period is reduced, leading to a smaller ADV period and higher energy savings.

Detailed simulations show that the proposed ATMA protocol adapts nicely to dynamic bursty traffic, providing reductions in energy consumption as high as 80% compared to other WSN MAC protocols such as Sensor-MAC (S-MAC), Timeout-MAC (T-MAC), Advertisement MAC (ADV-MAC), TRAMA and Z-MAC. An important achievement of ATMA is that it provides these significant energy savings without sacrificing the latency performance or the PDR, which is maintained at almost 100% for all the simulated scenarios. We also present an analytical model to calculate PDR and derive the optimal duration of the ADV period under a constant traffic load. Finally, we evaluate the performance of ATMA in mobile scenarios. Simulations show that ATMA performs well in mobile scenarios with very little performance degradation.

6 Hardware Implementation of ATMA on the SORA Platform

In wireless communications, we often use software simulators to test new protocols for the Medium Access Control (MAC), Routing or Network layers. Such simulations alone cannot reflect many of the challenges faced by real implementations of these protocols, such as clock-drift, synchronization, imperfect physical layers, and interference from other transmissions. Such issues may cripple a protocol that otherwise performs very well in software simulations. Thus, hardware implementation is essential for testing a protocol before any practical deployment.

As seen in Chapter 5, the ATMA protocol performs very well in simulations. However to fully explore its potential as well as its limitations, we need to test it with a real hardware implementation. In this chapter, we implement the ATMA protocol on the hardware platform developed by Microsoft Asia called SORA [44] or Software Radio. SORA is a fully programmable software radio platform that can be used with commodity PC architectures. The goal of this project is to set up a reusable hardware framework to evaluate the real-life performance of wireless protocols for real-time communication in mobile ad hoc and sensor networks.

We compare the results obtained from the hardware experiments to that ob-

tained from the software simulations. The comparisons show that the protocol does perform well under real-life scenarios, with the hardware experimental results closely mirroring the software simulation results. However, we do see the effects of interference, clock-drift and other factors in the form of some packet losses.

The rest of this chapter is organized as follows. In Section 6.1, we discuss the choice of SORA as our hardware platform. In Section 6.2, we describe the general SORA architecture, and in Section 6.3, we describe the implementation of the ATMA protocol on the SORA platform along with the challenges and difficulties faced during the process. In Section 6.4, we describe our experimental and software simulation setup, followed by a detailed comparison of the results obtained from both processes. Finally, in Section 6.5, we conclude the chapter.

6.1 Choice of Hardware Platform

In real-life wireless communication systems, lower layer processing such as those in the MAC layer require high-computational and real-time requirements. As such, they are often fabricated in Application Specific Integrated Circuits or ASIC chips [43]. A separate ASIC chip must be designed for each protocol to accommodate the different functionalities of that protocol. As a result, the process is costly, both economically and time-wise. Moreover, once the chip has been fabricated, no further changes or upgrades can be provided. Hence, using ASIC chips for research in wireless communication is a major challenge due to the lack of flexibility or re-programmability and the costs involved.

An alternative for hardware implementation of these protocols is to use Software Radios or Software Defined Radios (SDRs) [28]. SDRs use a software re-programmable wireless communication hardware system as a substitute for the lowest communication layers, which are usually fabricated in custom ASIC chips. To meet the processing and timing requirements of the lower layers, many of these

SDR platforms are based on programmable hardware such as field programmable gate arrays (FPGAs) [39], or embedded digital signal processors (DSPs). This poses a problem for developers, as learning to program each particular embedded architecture is difficult and oftentimes lacks rich development environments or tools for debugging.

There are SDR platforms that are based on general purpose processor architectures (GPP) such as commodity PCs. Here, developers have a rich front-end for programming and debugging. For example they may use familiar languages such as C/C++ for programming the hardware. However, the drawback is that these platforms oftentimes cannot meet the computing or timing requirements of the lower layers since PC hardware and software have not been designed for wireless signal processing. As a result, we only get limited performance and it is difficult to implement a state-of-the-art protocol on such a platform.

The Wireless and Networking Group in Microsoft Research Asia has resolved the SDR dilemma by combining the advantages of both SDR platforms. The result is Software Radio or SORA. SORA is a fully programmable SDR platform that meets the high computational and timing requirements of high-speed wireless protocols such as IEEE 802.11 by using commodity PCs only. Developers need not learn a specific embedded architecture but deal with a familiar high level programming environment using C/C++. Once a protocol is programmed into the SORA hardware, it appears as any other network device in the PC and gives the same performance as specially fabricated ASIC chips. The drawback in the case of SORA is the size of the radio control board, which is much bigger than a dedicated ASIC chip, as shown in Fig. 6.1.

SORA uses both hardware and software techniques for achieving a high-speed SDR in a commodity PC environment. The SORA radio control board (RCB) is equipped with a radio front-end for transmission and reception. A PCIe bus is used by the RCB to bridge the RF front-end to the PC memory. With the help

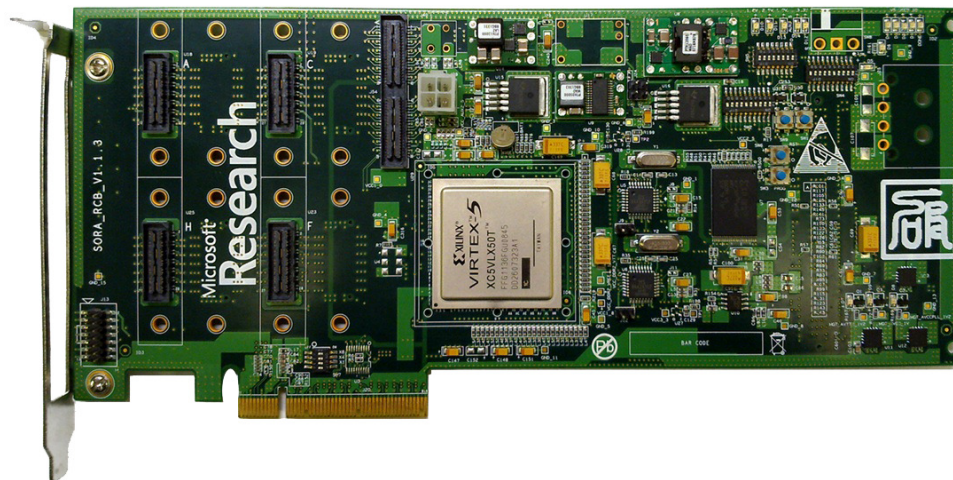


Figure 6.1: SORA programmable radio control board.

of the PCIe bus, the RCB can support up to 16.7 *Gbps* throughput with sub-microsecond latency. This is sufficient to satisfy the computational and timing requirements of modern wireless communication protocols.

There is another similar SDR platform called the Universal Software Radio Peripheral (USRP). However, we decided to go with SORA because of its rich developer resources and support from Microsoft.

6.2 The SORA Platform Architecture

In this section, we describe the general architecture of SORA and its Application Protocol Interfaces (APIs).

6.2.1 SORA Architecture

The general architecture of SORA is presented in Fig. 6.2. The RCB is connected to the commodity PC through a PCIe bus, which meets the throughput and latency requirements for digital signal processing. The RF front-end is connected to the RCB. The RCB communicates with the PC using the PCIe interface,

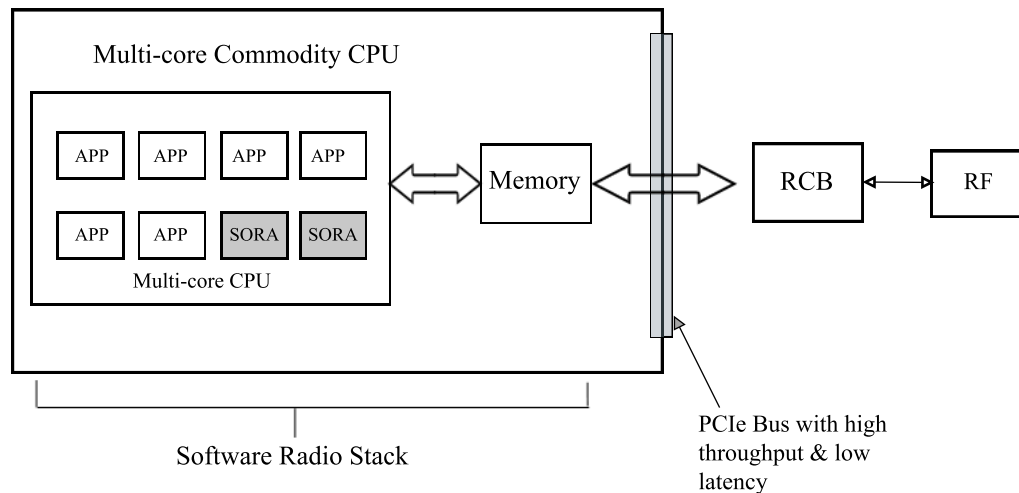


Figure 6.2: General architecture of SORA.

and reads and writes digital signal samples from/to the PC memory using direct memory access (DMA). The commodity PC must have multiple CPU cores as the SORA platform needs them to allocate resources required by the protocol running on the RCB. In Fig. 6.2, two of the eight cores of the PC are being used by the SORA RCB.

6.2.2 User-Mode eXtension

The SORA SDK provides a programming model called User-Mode eXtension or UMX. UMX allows developers to access the radio resources with built-in APIs such as those for Carrier Sensing, Transmission or Reception. As a result, the baseband-processing, which is usually done in the lower layers such as the MAC and PHY, can be handled through C/C++ functions and subroutines. This greatly reduces both programming and debugging efforts. UMX APIs facilitate high performance and low latency DSP implementation in user-mode, such as parallel threads (known as ethreads) and integration with the network stack. We implement the ATMA protocol using the APIs provided by UMX.

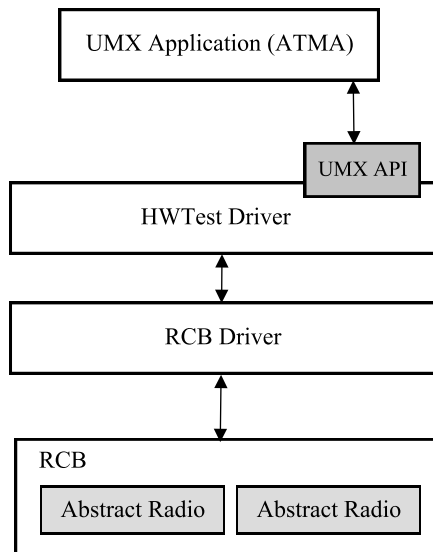


Figure 6.3: General architecture of SORA UMX.

SORA UMX functions mainly work with the help of Abstract Radio (AR) objects. An AR is a software abstract of a hardware radio and contains a transmission channel, a reception channel, and a set of control registers. An UMX SDR application uses abstract radio objects (ARO) to implement its functionalities. Every AR object is mapped to a real RF front-end by the RCB driver. If an UMX application such as the ATMA protocol sets a Control Register of an AR object, the abstract command is translated into a real operation sequence to the RF front-end chip-set by the RCB driver. Currently, a SORA RCB supports up to eight ARs that can map to different RF front-ends or can be grouped to form a MIMO system.

The general UMX architecture is shown in Fig. 6.3. All operations of the UMX Application (the ATMA protocol) proceed through the UMX APIs. The APIs, in turn, instruct the HWTest driver as well as the RCB driver to allocate AR objects for the protocol, which are turned into real operation sequences by

the RCB driver. Some of the important APIs used by any UMX application for processes such as initialization, transmission and reception are listed in Table 6.1.

6.2.3 Exclusive Threads

SORA provides real-time support with the help of exclusive threading. An exclusive thread or ethread is a non-interruptible thread that runs on one of the cores of a multi-core CPU. The core assignment and allocation is done dynamically by the RCB driver. An ethread is started simply by calling the function *SoraThreadAlloc*. Terminating an ethread is done by calling *SoraThreadStop*. As described in the next section, we use this feature of the UMX API for our ATMA implementation to reduce unpredictable delays such as transferring a signal to an RCB.

6.3 Implementation of ATMA on the SORA Platform

In this section, we provide an overview of the implementation of the ATMA protocol on the SORA architecture. We also discuss real-life issues that we faced including synchronization issues, unexpected delays in the transfer of signals into the RCB and the partial mitigation of this problem through the use of the Exclusive Thread (ethread) API.

The basic structure of the ATMA UMX Application is shown in Fig. 6.4. The figure assumes that all nodes are synchronized. We will discuss the synchronization procedure in the next section. The application has three parallel ethreads – the transmission thread, the reception thread and the ATMA State Machine thread. All these three threads run concurrently and are allocated resources dynamically by the RCB driver. Each node runs these three threads concurrently,

Table 6.1: Important SORA UMX APIs

Function Name	Description
<i>SoraUInitUserExtension</i>	Initializes UMX library
<i>SoraURadioStart</i>	Starts the radio
<i>SoraUCleanUserExtension</i>	Cleans up allocated resources
<i>SoraURadioMapRxSampleBuf</i>	Returns pointer to the receiving buffer and the buffer size
<i>RX_STREAM</i>	Provides a stream of I/Q samples received from the RF front-end
<i>SoraURadioAllocRxStream</i>	Gives the UMX application an <i>RX_STREAM</i> from the receiving buffer
<i>SoraRadioReadRxStream</i>	Reads a signal block
<i>SoraURadioReleaseRxStream</i>	Releases <i>RX_STREAM</i> before exiting
<i>SoraURadioUnmapRxSampleBuf</i>	Releases the memory mapped to the RX buffer of a radio
<i>SoraURadioMapTxSampleBuf</i>	Provides access to sample buffer to store signals
<i>SoraURadioTransfer</i>	Transfers modulated signal to RCB
<i>SoraURadioTx</i>	Transmits the stored signal from RCB
<i>SoraURadioTxFree</i>	Unbinds all TX resources

whether they are a transmitter or a receiver node. There is a global *period_type* flag that is used to identify the period type in each of these threads. For example when the period type is SYNC, a node that has a SYNC packet to transmit will call the appropriate function to create a SYNC packet, modulate and transfer the signal to the RCB, wait for the time chosen by the node to transmit the packet, perform carrier sense and then transmit the SYNC packet if the channel is found to be clear.

In a software simulator such as ns-2, the process of creating a packet, carrier sensing and transmitting are instantaneous. The only time required is the time duration required to transmit the packet itself. There is no separate stage for transferring the modulated signal to the RCB. In practice, all of these three processes take time and often these times are unpredictable as the CPU core to which this thread is allocated may be performing other functions (multi-threading). In our experiments, sometimes transferring a modulated signal to the RCB may take time in the order of a millisecond, which is significant as theoretically the packet transmission time is around 0.048 *ms*. This is where parallel ethreads are extremely useful. If a protocol requires a SYNC packet, or any packet as a matter of fact, to be sent in the next frame, the transmission thread can prepare the packet as well as transfer it to the RCB beforehand to prevent wasting any extra time. When the time comes to transmit the packet, the node can just carrier-sense, which is always being done by the reception thread, and transmit the packet once the channel is free. Thus, efficient use of ethreads will optimize any UMX application and provide results that are closer to the simulation results.

A similar procedure is followed for transmitting an ADV packet. A node will know at the beginning of the frame if it needs to transmit an ADV packet. Hence the transmission thread will prepare and transfer the modulated signal into the RCB beforehand and transmit when the time comes. The same is not true for the ADV-ACK. An ADV-ACK is prepared only when an ADV packet is received. As

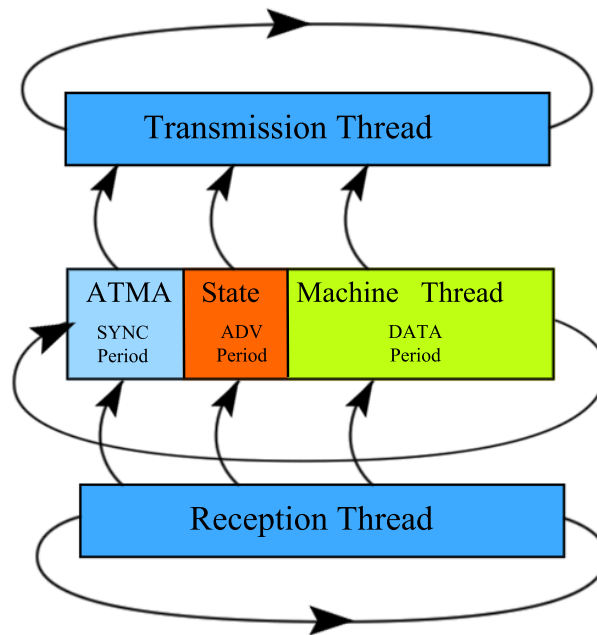


Figure 6.4: General structure of the ATMA UMX Application. The figure assumes that all nodes are synchronized. Also, the figure is not drawn to scale.

a result, the transfer process of the modulated ADV-ACK packet into the RCB takes a random amount of time that could be on the order of a *ms*. Similarly, the DATA packets are prepared ahead of time but its corresponding ACK packet is prepared only when the DATA packet is successfully received.

6.3.1 Synchronization of Nodes

In our software simulations we either assume that the nodes are synchronized or that the synchronization is done automatically by the code itself. In real-life implementations, synchronization is a non-trivial problem. Clock drifts, interference from other transmissions and many other factors effect synchronization. In order to implement the ATMA protocol on UMX in SORA, we had to synchronize all the nodes, which we did not have to do in the software simulations. In

fact, synchronization techniques are themselves a separate research topic. For our implementation, we follow a relatively simple procedure.

We assume a single hop network where all nodes are in the transmission range of one another. The node with index number 0 is set as the synchronizing node. All other nodes follow this node. When the ATMA UMX application is initialized, the 0^{th} node starts its frame at an arbitrary time. All other nodes keep listening to the medium until they receive a SYNC packet from node 0. They do not initialize their own frames. The duration of our SYNC period is 10 *ms*. We divide the SYNC period into 10 slots of 1 *ms* each. When the 0^{th} node transmits a SYNC packet, it randomly chooses one of these 10 slots to transmit and includes the slot number in the packet itself. When a non-synchronized node receives this packet, it can calculate the time remaining in the frame of the 0^{th} node as

$$T_{left} = t_{FRAME} - (slot_no * slot_duration + t_{SYNC}) \quad (6.1)$$

where t_{FRAME} is the fixed duration of each frame, and $slot_no$ is the slot chosen by the SYNC node. $slot_duration$ is the duration of each SYNC slot, and t_{SYNC} is the duration of a SYNC packet. Oftentimes, a receiving node takes some time to decode the packet and do the necessary calculation. Hence, to make T_{LEFT} more accurate, we note the time t_0 when the node starts receiving the SYNC packet and also the time t_1 when it has demodulated, decoded and extracted the slot number from the received packet. Then, we modify equation 6.1 as

$$T_{left} = t_{FRAME} - (slot_no * slot_duration + t_{SYNC}) - (t_1 - t_0) \quad (6.2)$$

Equation 6.2 is more accurate in keeping the nodes synced for a longer duration of time. However, we repeat the sync process every 8 frames to eliminate clock drifts. We use *QueryPerformanceCounter()*, which is a high-resolution performance counter function along with *QueryPerformanceFrequency()*, which retrieves its frequency, to calculate the time stamps.



Figure 6.5: Setup of the SORA experiments.

6.4 Experimental Setup and Results

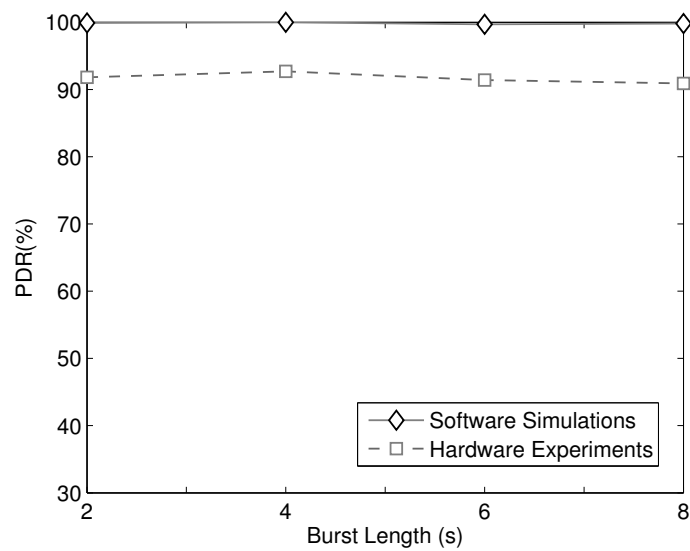
In this section, we describe the setup for our hardware experiments as well as the software simulations and present a comparison of the results obtained. We construct a single hop network with 4 SORA nodes, two transmitters and two receivers. The setup is shown in Fig. 6.5. We drop any ADV or DATA packet if the channel is busy. There is no retransmission mechanism. We construct a similar scenario of 4 nodes in a single hop network in ns-2.29 [34]. We use the 802.11b PHY layer provided by the UMX application and only alter the medium access protocol to ATMA. The frame duration is kept to 250 ms . Both SYNC and ADV periods are 10 ms each, while the Data period is 230 ms . Each data slot is 10 ms . Hence, there are 23 data slots. The simulation time is 20 s , and each point in the figures is the average of 50 runs. The important parameters are presented in

Table 6.2. ¹ For the experiments, we fix the inter-burst length to 2 *s* and vary the burst length from 2 *s* to 8 *s* in steps of two. A source node generates one packet in each frame within a burst. It is seen from Figs. 6.6(a) and 6.6(b) that the packet delivery ratio (PDR) for the software simulations remains almost constant at 100% while the latency remains constant around 120 *ms*. As for the hardware experiments, the values also follow a similar trend, but the PDR values are slightly lower, around 91%, while the latency values are slightly higher, at around 130 *ms*. This drop in performance is expected for the hardware experiments because the hardware experiments were conducted using 802.11b baseband physical layer and there are many other transmissions going on in the same spectrum from WiFi transmissions. Since there are no retransmissions, once an ADV or a DATA packet is not transmitted because the medium is busy, it is dropped. Sometimes, even if packets are transmitted, they are corrupted due to interference from other transmissions. This is the cause for the decrease in PDR. The slight increase in average latency is due to the time taken for decoding, demodulating and other hardware factors. Thus, the hardware experiments reflect real life conditions that the software simulations cannot. However, the overall trend is similar to those obtained from simulations, which in turn validates the working principle of the ATMA protocol.

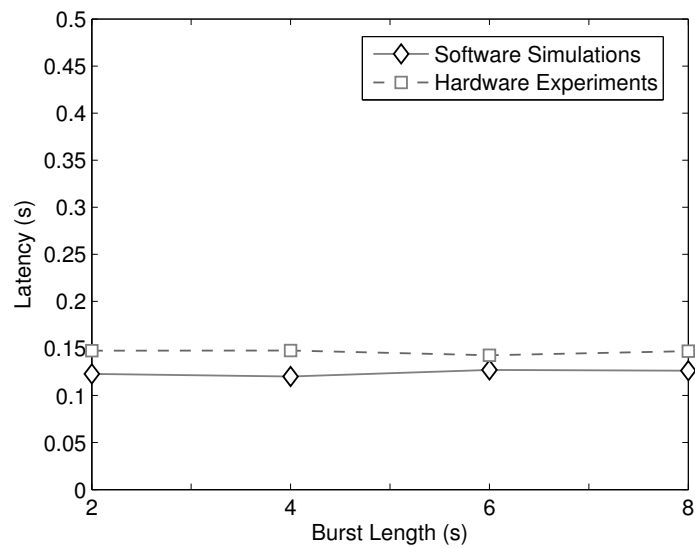
6.5 Summary

This chapter presents a hardware implementation of the ATMA protocol on the SORA platform developed by Microsoft Asia. Although software simulations show that the ATMA protocol performs very well under different scenarios, they do not reflect real-life conditions such as interference, clock drift and many other real-

¹The SORA code of the hardware implementation of ATMA is available at <http://www.ece.rochester.edu/projects/wcng/>.



(a) Packet delivery ratio



(b) Latency

Figure 6.6: Comparison of PDR and latency values obtained from hardware experiments and software simulations.

Table 6.2: Important Parameters for the SORA ATMA Experiments

Parameter	Values
Duration of Frame	250 <i>ms</i>
Duration of SYNC Period	10 <i>ms</i>
Duration of ADV Period	10 <i>ms</i>
Duration of DATA Period	230 <i>ms</i>
Duration of a SYNC Slot	1 <i>ms</i>
Duration of a ADV Slot	1 <i>ms</i>
Duration of DATA Slot	10 <i>ms</i>
Duration of SYNC Packet	0.048 <i>ms</i> (2 bytes + 4 bytes of CRC)
Duration of ADV/ADV-ACK/ACK Packets	0.056 <i>ms</i> (3 bytes + 4 bytes of CRC)
Duration of DATA Packet	0.856 <i>ms</i> (103 bytes + 4 bytes of CRC)
Transmission Rate	1000 <i>kbps</i>
Simulation Time	20 <i>s</i>
ATMA Reservation Length	6 frames

life factors. Comparing the results obtained from the hardware experiments and the software simulations, we see that both present a very similar trend and the PDR and latency values are similar in both cases. This validates the protocol and serves as a proof of concept before fabricating a dedicated ASIC chip to implement ATMA in a real sensor network.

7 Conclusions and Future Directions

7.1 Conclusions

In this dissertation, I have explored the fundamental problem of limited energy supply for wireless sensor networks through the use of Advertisement techniques to reduce energy waste and optimize performance. The results described in this dissertation indicate that advertisement techniques can provide substantial energy savings for wireless sensor networks while improving performance metrics such as packet delivery ratio and latency. The contributions of the research are summarized below:

1. I proposed ADV-MAC, a new MAC protocol that uses Advertisement for Data packets to save energy that is otherwise lost in idle listening or overhearing. ADV-MAC incorporates a multicasting technique that is absent in S-MAC and T-MAC. A detailed performance comparison of ADV-MAC with S-MAC and T-MAC is presented using the metrics of energy consumption, throughput and latency under various traffic conditions in single hop and multi-hop scenarios.
2. I proposed an improved method of contention in the Advertisement period and an analytical model for computing the energy and packet delivery ratio

of the ADV-MAC protocol. Using the analytical model, the optimal duration of the Advertisement period for optimal packet delivery ratio and the lowest energy consumption is determined. Then, a detailed performance comparison of the improved and optimized ADV-MAC protocol with the original ADV-MAC as well as S-MAC and T-MAC is presented under different traffic loads in single hop and multi-hop scenarios.

3. I developed an approach to remove the second contention of the data period in the ADV-MAC protocol and convert it into a TDMA based MAC protocol called Advertisement-based Time-division Multiple Access (ATMA), saving further energy under bursty traffic conditions.
4. I proposed an analytical model to determine the optimal Advertisement period of ATMA. Then, I presented a detailed performance comparison of ATMA with a similar TDMA based MAC protocol TRAMA [31], a hybrid protocol Z-MAC [49] along with the contention based protocols ADV-MAC, S-MAC and T-MAC with energy consumption, packet delivery ratio and latency as the performance metrics.
5. Software simulations alone cannot reflect many of the challenges faced by real implementations of MAC protocols, such as clock-drift, synchronization, imperfect physical layers, and irregular interference from other transmissions. Hence, I conclude my research with a hardware implementation of the ATMA protocol on the SORA platform and compare the results obtained from the hardware experiments to that from software simulations.

7.2 Future Directions

While this dissertation has provided new advertisement techniques to improve the energy efficiency at the MAC layer, along with qualitative and quantitative anal-

ysis of the benefits in terms of extending the lifetime of wireless sensor networks by using advertisement techniques, some open questions still remain.

7.2.1 Comparison with Additional Protocols

In my research, I have so far compared ATMA with contention based protocols such as S-MAC, T-MAC and ADV-MAC, a TDMA based protocol TRAMA and a hybrid protocol Z-MAC. I have provided only qualitative analysis of the hybrid MAC protocol IEEE 802.15.4 [2], which is a MAC protocol for low-rate wireless personal area networks (LR-WPANs). It will be interesting to see the advantages and limitations of the advertisement based approach of ATMA compared with the hybrid approach of IEEE 802.15.4 for energy saving with software simulations rather than simple qualitative analysis. IEEE 802.15.4 has a beacon mode that provides a Guaranteed Time Slot (GTS) allocation. In the contention period, slots are reserved and then contention free data transmission is provided. The similarity of this protocol with ATMA presents an interesting direction for future research.

Another interesting direction would be to compare ATMA with the TRACE family of protocols [45]. As discussed in Chapter 2, MH-TRACE has a lot of similarities with ATMA. In essence, ATMA is a completely distributed approach while MH-TRACE is a cluster-based, somewhat centralized approach for dealing with the same problem of optimal bandwidth allocation and extending network lifetime.

7.2.2 Extension of Current Hardware Implementation

To clearly demonstrate the functionality of ATMA and advertisement based approaches, it is important to implement the protocols on a real hardware testbed. The current hardware setup of ATMA is only valid for single hop networks, and

does not consider the hidden terminal problem. A full implementation of the ATMA protocol would be beneficial before fabricating a dedicated ASIC chip to implement ATMA in a real sensor network.

Bibliography

- [1] Crossbow Technology Inc. <http://www.xbow.com/>, 2010.
- [2] IEEE 802.15 WPAN Task Group. <http://www.ieee802.org/15/pub/TG4.html>, 2010.
- [3] Sentilla Corporation. <http://www.sentilla.com/>, 2010.
- [4] T. Ahonen, R. Virrankoski, and M. Elmusrati. Greenhouse Monitoring with Wireless Sensor Network. In *Mechtronic and Embedded Systems and Applications, 2008. MESA 2008. IEEE/ASME International Conference on*, pages 403–408, Oct. 2008.
- [5] J. G. T. Anderson. Pilot Survey of Mid-coast Maine Seabird Colonies: An Evaluation of Techniques. *Report to the State of Maine Dept. of Inland Fisheries and Wildlife*, 1995.
- [6] George E. Andrews. *Number Theory*. Saunders, Philadelphia, USA, 1971.
- [7] M. Antoniou, M.C. Boon, P.N. Green, P.R. Green, and T.A. York. Wireless Sensor Networks for Industrial Processes. In *Sensors Applications Symposium, 2009. SAS 2009. IEEE*, pages 13–18, Feb. 2009.
- [8] Manuel Asn, Alicia; Calahorra. Sensor Networks to Monitor Air Pollution in Cities. <http://www.libelium.com/libeliumworld/articles/102732734500>, 2010.

- [9] Lichun Bao and J.J. Garcia-Luna-Aceves. A New Approach to Channel Access Scheduling for Ad Hoc Networks. In *Proc. ACM Seventh Annual International Conference on Mobile Computing and Networking*, pages 210–221, 2001.
- [10] Michael Buettner, Gary V. Yee, Eric Anderson, and Richard Han. X-MAC: A Short Preamble MAC Protocol for Duty-Cycled Wireless Sensor Networks. In *ACM SenSys '06*, pages 307–320, New York, NY, USA, 2006. ACM.
- [11] George Casella and Roger L. Berger. *Statistical Inference*, chapter Order Statistics, pages 226–228. Duxbury Press, second edition, 2001.
- [12] E. Egea-Lopez, J. Vales-Alonso, A. Martinez-Sala, P. Pavon-Mario, and J. Garcia-Haro. Simulation Scalability Issues in Wireless Sensor Networks. *Communications Magazine, IEEE*, 44(7):64–73, 2006.
- [13] Haigang Gong, Jiannong Cao, Ming Liu, Lijun Chen, and Li Xie. A Traffic Aware, Energy-Efficient MAC Protocol for Wireless Sensor Networks. *Int. J. Ad Hoc Ubiquitous Comput.*, 4(3/4):148–156, 2009.
- [14] S. Gupta, S. Varma, G.S. Tomar, and R.K. Abrol. Wireless Sensor Network Based Industrial Monitoring and Diagnostic System. In *Computational Intelligence, Communication Systems and Networks, 2009. CICSYN '09. First International Conference on*, 2009.
- [15] Tian He, Liqian Luo, Ting Yan, Lin Gu, Qing Cao, Gang Zhou, R. Stoleru, P. Vicaire, Qiuhua Cao, J.A. Stankovic, S.H. Son, and T.F. Abdelzaher. An Overview of the VigilNet Architecture. In *Embedded and Real-Time Computing Systems and Applications, 2005. Proceedings. 11th IEEE International Conference on*, pages 109 – 114, Aug. 2005.
- [16] Wendi Rabiner Heinzelman, Anantha Chandrakasan, and Hari Balakrishnan. Energy-Efficient Communication Protocol for Wireless Microsensor Net-

- works. In *Proceedings of the HICSS '00*, page 8020, Washington, DC, USA, 2000. IEEE Computer Society.
- [17] L. Van Hoesel and P. Havinga. A Lightweight Medium Access Protocol (LMAC) for Wireless Sensor Networks. Tokyo, Japan, Jun 2004.
- [18] Hongwei Huo, Youzhi Xu, Hairong Yan, S. Mubeen, and Hongke Zhang. An Elderly Health Care System Using Wireless Sensor Networks at Home. In *Sensor Technologies and Applications, 2009. SENSORCOMM '09. Third International Conference on*, pages 158–163, June 2009.
- [19] IEEE Standard 802.11, Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications. LAN MAN Standards Committee of the IEEE Computer Society, June 1999.
- [20] Philo Juang, Hidekazu Oki, Yong Wang, Margaret Martonosi, Li Shiuan Peh, and Daniel Rubenstein. Energy-Efficient Computing for Wildlife Tracking: Design Trade-offs and Early Experiences with ZebraNet. *SIGPLAN Not.*, 37:96–107, October 2002.
- [21] Farinaz Koushanfar, Miodrag Potkonjak, and Alberto Sangiovanni-vincentelli. Fault Tolerance in Wireless Sensor Networks, book chapter. In *Handbook of Sensor Networks, I. Mahgoub and M. Ilyas*, page 2004.
- [22] S. Kumar and D. Shepherd. SensIT: Sensor Information Technology for the Warfighter. In *Proceedings of 4th International Conference on Information Fusion*, 2001.
- [23] Younggoo Kwon, Yuguang Fang, and Haniph Latchman. A Novel MAC Protocol with Fast Collision Resolution for Wireless LANs. In *IEEE Infocom*, pages 793–807, 2003.

- [24] Na Li, Nan Zhang, Sajal K. Das, and Bhavani Thuraisingham. Privacy Preservation in Wireless Sensor Networks: A State-of-the-Art Survey. *Ad Hoc Netw.*, 7:1501–1514, November 2009.
- [25] P. Lin, C. Qiao, and X. Wang. Medium Access Control with a Dynamic Duty Cycle for Sensor Networks. In *Wireless Communications and Networking Conference, 2004. WCNC. 2004 IEEE*, volume 3, pages 1534 – 1539 Vol.3, 21-25 2004.
- [26] Jingyu Liu and Yanjun Fang. Urban Traffic Control System Based on Wireless Sensor Networks. In *Information Acquisition, 2006 IEEE International Conference on*, pages 295 –300, Aug. 2006.
- [27] Errol L. Lloyd. Broadcast Scheduling for TDMA in Wireless Multihop Networks. *Handbook of Wireless Networks and Mobile Computing*, pages 347–370, 2002.
- [28] A. Luiz Garcia Reis, A.F. Barros, K. Gusso Lenzi, L.G. Pedroso Meloni, and S.E. Barbin. Introduction to the software-defined radio approach. *Latin America Transactions, IEEE (Revista IEEE America Latina)*, 10(1):1156–1161, 2012.
- [29] F. O'Reilly. M. Connolly. Sensor Networks and the Food Industry. In *REALWSN '05 Workshop on Real-World Wireless Sensor Network*, 2005.
- [30] Alan Mainwaring, David Culler, Joseph Polastre, Robert Szewczyk, and John Anderson. Wireless Sensor Networks for Habitat Monitoring. In *Proceedings of the 1st ACM International Workshop on Wireless Sensor Networks and Applications*, WSNA '02, pages 88–97, New York, NY, USA, 2002. ACM.
- [31] Jianlin Mao, Zhiming Wu, and Xing Wu. A TDMA Scheduling Scheme for Many-to-One Communications in Wireless Sensor Networks. *Comput. Commun.*, 30(4):863–872, 2007.

- [32] John McCulloch, Paul McCarthy, Siddeswara Mayura Guru, Wei Peng, Daniel Hugo, and Andrew Terhorst. Wireless Sensor Network Deployment for Water Use Efficiency in Irrigation. In *Proceedings of the Workshop on Real-world Wireless Sensor Networks*, REALWSN '08, pages 46–50, New York, NY, USA, 2008. ACM.
- [33] Ar Milenkovi, Chris Otto, and Emil Jovanov. Wireless Sensor Networks for Personal Health Monitoring: Issues and an Implementation. *Computer Communications (Special issue: Wireless Sensor Networks: Performance, Reliability, Security, and Beyond)*, 29:2521–2533, 2006.
- [34] The Network Simulator NS-2. <http://www.isi.edu/nsnam/ns/>, 2010.
- [35] Guangyu Pei and C. Chien. Low Power TDMA in Large Wireless Sensor Networks. In *Military Communications Conference, 2001. MILCOM 2001. Communications for Network-Centric Operations: Creating the Information Force. IEEE*, volume 1, pages 347 – 351 vol.1, 2001.
- [36] Adrian Perrig, John Stankovic, and David Wagner. Security in Wireless Sensor Networks. *Commun. ACM*, 47:53–57, June 2004.
- [37] Joseph Polastre, Jason Hill, and David Culler. Versatile Low Power Media Access for Wireless Sensor Networks. In *ACM SenSys '04*, pages 95–107, New York, NY, USA, 2004. ACM.
- [38] Surjya Ray, Ilker Demirkol, and Wendi Heinzelman. ADV-MAC: Advertisement-based MAC Protocol for Wireless Sensor Networks. In *The Fifth International Conference on Mobile Ad-hoc and Sensor Networks*, Wu Yi Mountain, China, December 2009. IEEE.
- [39] B.C. Readler. *Verilog by Example: A Concise Introduction for FPGA Design*. Full Arc Press, 2011.

- [40] I. Rhee, A. Warrier, Jeongki Min, and Lisong Xu. DRAND: Distributed Randomized TDMA Scheduling for Wireless Ad Hoc Networks. *Mobile Computing, IEEE Transactions on*, 8(10):1384–1396, oct. 2009.
- [41] F. Salvadori, M. de Campos, P.S. Sausen, R.F. de Camargo, C. Gehrke, C. Rech, M.A. Spohn, and A.C. Oliveira. Monitoring in Industrial Systems Using Wireless Sensor Network With Dynamic Power Management. *Instrumentation and Measurement, IEEE Transactions on*, 58(9):3104–3111, sept. 2009.
- [42] Suresh Singh and C. S. Raghavendra. PAMAS—Power Aware Multi-Access Protocol with Signalling for Ad Hoc Networks. *SIGCOMM Comput. Commun. Rev.*, 28(3):5–26, 1998.
- [43] Michael John Sebastian Smith. *Application-Specific Integrated Circuits*. Addison-Wesley Professional, 1st edition, 2008.
- [44] SORA (Software RADIOS. <http://research.microsoft.com/en-us/projects/sora/>, 2012.
- [45] B. Tavli and W.B. Heinzelman. MH-TRACE: Multihop Time Reservation Using Adaptive Control for Energy Efficiency. In *Military Communications Conference, 2003. MILCOM 2003. IEEE*, volume 2, pages 1292–1297 Vol.2, oct. 2003.
- [46] Tijs van Dam and Koen Langendoen. An Adaptive Energy-Efficient MAC Protocol for Wireless Sensor Networks. In *ACM SenSys '03*, pages 171–180, New York, NY, USA, 2003. ACM.
- [47] Yu Wang and I. Henning. A Deterministic Distributed TDMA Scheduling Algorithm for Wireless Sensor Networks. *Proceedings of WiCom Conference '07*, pages 2759–2762, Sept. 2007.

- [48] Tim Wark, Peter Corke, Pavan Sikka, Lasse Klingbeil, Ying Guo, Chris Crossman, Phil Valencia, Dave Swain, and Greg Bishop-Hurley. Transforming Agriculture through Pervasive Wireless Sensor Networks. *IEEE Pervasive Computing*, 6:50–57, April 2007.
- [49] Ajit Warrier, Jeongki Min, and Injong Rhee. Z-MAC: A Hybrid MAC for Wireless Sensor Networks. pages 90–101. ACM Press, 2005.
- [50] Wei Ye, John Heidemann, and Deborah Estrin. An Energy-Efficient MAC Protocol for Wireless Sensor Networks. In *Proceedings of the IEEE Infocom*, pages 1567–1576, New York, NY, USA, June 2002. IEEE.
- [51] Wei Ye, Fabio Silva, and John Heidemann. Ultra-Low Duty Cycle MAC with Scheduled Channel Polling. In *ACM SenSys '06*, pages 321–334, New York, NY, USA, 2006. ACM.
- [52] Liyang Yu, Neng Wang, and Xiaoqiao Meng. Real-Time Forest Fire Detection with Wireless Sensor Networks. In *Wireless Communications, Networking and Mobile Computing, 2005. Proceedings. 2005 International Conference on*, volume 2, pages 1214 – 1217, sept. 2005.
- [53] Hesheng Zhang, Cheng Pan, Jun Yang, Honghui Dong, Yong Qin, and Limin Jia. SN-UTIA: A Sensor Network for Urban Traffic Information Acquisition. In *Intelligent Vehicles Symposium (IV), 2010 IEEE*, pages 566 –571, June 2010.

Appendices

A Proofs for ADV-MAC Theorems

A.1 Proof of Equation (4.1)

To prove (4.1), we need to use the principle of inclusion-exclusion [6] and certain results that follow from it. The inclusion-exclusion principle is stated in Lemma 1 and the related results are stated in Lemmas 2 and 3 along with their proofs.

Lemma 1 (The principle of inclusion-exclusion [6]). *Suppose, we have t finite sets A_1, A_2, \dots, A_t , then*

$$\begin{aligned} \left| \bigcup_{i=1}^t A_i \right| &= \sum_{1 \leq i \leq t} |A_i| - \sum_{1 \leq i < j \leq t} |A_i \cap A_j| + \dots \\ &\dots + (-1)^{(s+1)} \sum_{1 \leq i_1 < \dots < i_s \leq t} |A_{i_1} \cap \dots \cap A_{i_s}| + \dots \\ &\dots + (-1)^{t+1} |A_1 \cap \dots \cap A_t|. \end{aligned}$$

where $|A|$ is the cardinality of set A .

Lemma 2 (Corollary of inclusion-exclusion principle). *If we have some universal set U , and a collection of finite sets A_1, A_2, \dots, A_t contained in U , then*

$$\left| \bigcap_{i=1}^t A_i^c \right| = |U| - \sum_{1 \leq i \leq t} |A_i| + \sum_{1 \leq i < j \leq t} |A_i \cap A_j| + \dots$$

$$\begin{aligned}
& \dots + (-1)^{(s)} \sum_{1 \leq i_1 < \dots < i_s \leq t} |A_{i_1} \cap \dots \cap A_{i_s}| + \dots \\
& \dots + (-1)^t |A_1 \cap \dots \cap A_t|.
\end{aligned}$$

Proof. The cardinality of $A_1^c \cap A_2^c \cap \dots \cap A_t^c$ is same as the cardinality of the set $U - A_1 \cup A_2 \cup \dots \cup A_t$. From the rule of sum, the cardinality of this set is $|U| - |A_1 \cup A_2 \cup \dots \cup A_t|$. Hence, the result follows from Lemma 1. \square

Lemma 3. *Suppose there are n_t distinguishable balls, and g distinguishable bins. Then, the number of different ways of distributing the balls into the bins so that each bin contains at least one ball is*

$$\begin{aligned}
& g^{n_t} - g(g-1)^{n_t} + \binom{g}{g-2}(g-2)^{n_t} + \dots \\
& \dots + (-1)^s \binom{g}{g-s}(g-s)^{n_t} + \dots + (-1)^{g-1}g \\
& = \sum_{s=0}^{g-1} (-1)^s \binom{g}{g-s}(g-s)^{n_t}.
\end{aligned}$$

Proof. For $1 \leq i \leq g$, let A_i be the set of the distributions that has no balls in bin number i . This means, we want those distributions which does not lie in any of the sets A_1, A_2, \dots, A_g . Thus, we want to find $|A_1^c \cap \dots \cap A_g^c|$. By lemma 2, this is equal to

$$\begin{aligned}
& |U| - \sum_{1 \leq i \leq g} |A_i| + \sum_{1 \leq i < j \leq g} |A_i \cap A_j| + \dots + \\
& (-1)^{(s)} \sum_{1 \leq i_1 < \dots < i_s \leq g} |A_{i_1} \cap \dots \cap A_{i_s}| + \dots \\
& \dots + (-1)^g |A_1 \cap \dots \cap A_g|
\end{aligned} \tag{A.1}$$

where U is the set of all possible distributions.

Without any restrictions, the number of ways of distributing n_t balls into g bins is g^{n_t} . Thus, $|U|$ is g^{n_t} . Now, let us consider the cardinality of $A_{i_1} \cap \dots \cap A_{i_s}$. This set comprises those distributions that leave bins i_1, i_2, \dots, i_s empty. This is same as distributing the n_t balls in the remaining $g - s$ bins, and this may be done in $(g - s)^{n_t}$ ways. Now, there are $\binom{g}{s} = \binom{g}{g-s}$ ways of choosing the $g - s$ bins from the g bins. Combing these facts we get,

$$\sum_{1 \leq i_1 < \dots < i_s \leq g} |A_{i_1} \cap \dots \cap A_{i_s}| = \binom{g}{g-s} (g-s)^{n_t}.$$

Substituting all these in (A.1), we get the result.

□

Equation (4.1) states that the number of ways of distributing n_t distinguishable balls into g indistinguishable bins such that each bin has at least one ball is given by

$$\Theta(n_t, g) = \frac{1}{g!} \sum_{s=0}^{g-1} (-1)^s \binom{g}{g-s} (g-s)^{n_t}$$

Proof. Suppose the total number of possible distributions of n_t distinguishable balls in g indistinguishable bins be $\Theta(n_t, g)$. Each of these distributions naturally corresponds to a distribution with distinguishable bins. For each of the $\Theta(n_t, g)$ distributions with the indistinguishable bins, we can label each bin from 1 to g . Since no bin is empty, each distribution with indistinguishable bins will give $g!$ distributions with distinguishable bins. Thus, the number of distributions with distinguishable bins is exactly $g!$ times the number of distributions considering indistinguishable bins. This means

$$g! \Theta(n_t, g) = \sum_{s=0}^{g-1} (-1)^s \binom{g}{g-s} (g-s)^{n_t}$$

This proves (4.1).

□

A.2 Expression of $\xi(x, g)$ and Its Proof

We need to find $\xi(x, g)$, the number of ways that g distinct slots may be chosen that will result in x transmitting nodes and $N - x$ deferring nodes. We assume that there is enough space for x nodes to choose g distinct slots, which will result in g distinct transmissions, i.e., $S_{adv} \geq g(t_{adv} + 1)$. This was taken into account in Section 4.2.1.1 when we computed $g_{max}(x)$, which is the maximum g value, i.e., the maximum number of distinct slots for x transmitting nodes. Initially all of the N nodes will select a slot out of the first $S = S_{adv} - t_{adv}$ slots. They will not select any slot after S , as there will be not enough time left in the ADV period to complete the transmission. In order for x of the N nodes to be able to transmit, the selected slots have the following properties:

1. The slots chosen by the x transmitting nodes must be before those selected by the $N - x$ deferring nodes.
2. The x transmitting nodes choose slots within the first S_U slots which is given by

$$S_U = S_{adv} - gt_{adv}. \quad (\text{A.2})$$

If any of the slot numbers selected by x transmitting nodes is higher than S_U , there will not be enough time left for g distinct transmissions.

3. For the remaining $N - x$ nodes to defer, they must not choose any slots from the first S_L slots, however they choose from the slots between $(S_L + 1)^{th}$ and S^{th} . When the ADV period is big enough for more than g distinct transmissions, i.e., $S_{adv} \geq (g + 1)(t_{adv} + 1)$, the value of S_L is given by

$$S_L = S_{adv} - (g + 1)t_{adv}. \quad (\text{A.3})$$

If the size of the ADV period is sufficient only for g distinct transmissions, then $S_L = g$.

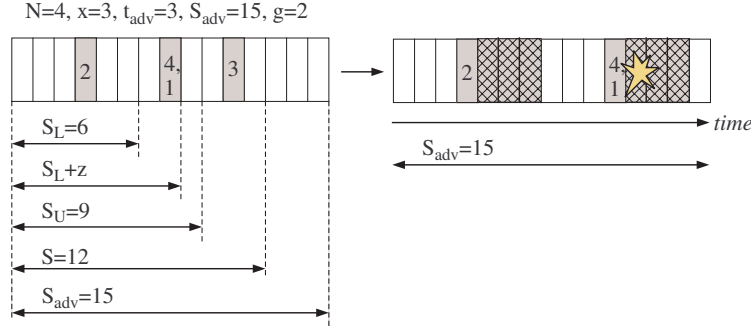


Figure A.1: Example of slot selection in ADV period: Four nodes contend of which three get to transmit as they select slots within $S_U = 9$. The three nodes select two distinct slots. Nodes 1 and 4 select the same slot and result in a collision. Node 3 selects a slot after S_U and hence has to defer.

Fig. A.1 shows the above restrictions for $N = 4$, $x = 3$, $t_{adv} = 3$ and $S_{adv} = 15$. In this case, g may take any value in $\{1, 2, 3\}$, but the figure is for the case when $g = 2$.

Let us try to find the number of possible cases of slot selection that gives exactly x transmitting nodes with g distinct slots where all the g slots lie within S_L . The x transmitting nodes may select g distinct slots from the first S_L slots in $S_L(S_L - 1) \dots (S_L - g + 1) = \prod_{k=0}^{g-1} (S_L - k)$ ways. For each of these ways of slot selection by the first x nodes, the remaining $N - x$ nodes may select their slots from the range $[S_L + 1, S]$ in $(S - S_L)^{N-x}$ ways. Thus all possible cases of slot selections where the g distinct selected slots lie within S_L and give exactly x transmitting nodes is given by

$$\delta_1 = \left(\prod_{k=0}^{g-1} (S_L - k) \right) (S - S_L)^{N-x}. \quad (\text{A.4})$$

When $S_L < S_U$, the x nodes may also select any slot from the range $[S_L + 1, S_U]$. Let us try to find all possible cases of slot selection that gives exactly x transmitting nodes with g distinct slots such that at least one of the g slots lie in

the range $[S_L + 1, S_U]$. Let the g^{th} distinct slot be the $(S_L + z)$. The g distinct slots may be chosen from the first $S_L + z$ slots in $(S_L + z)(S_L + z - 1) \dots (S_L + z - g + 1) = \prod_{k=0}^{g-1} (S_L + z - k)$ ways. However, this includes the cases where the $(S_L + z)^{th}$ slot is not chosen. We need to subtract those cases. Then, the number of possible ways in which x node can select g distinct slots such that the last selected slot is the $(S_L + z)^{th}$ slot is given by

$$\psi = \prod_{k=0}^{g-1} (S_L + z - k) - \prod_{k=0}^{g-1} (S_L + z - 1 - k), \quad (\text{A.5})$$

where the last product term subtracts all possible cases where the $(S_L + z)^{th}$ slot is unoccupied. For each of these ψ ways by which the first x nodes may choose their slots, the remaining $N - x$ nodes may choose their slots in the range $[S_L + z + 1, S]$ in $(S - (S_L + z))^{N-x}$ ways. Thus, the total number of possible cases of slot selection that gives exactly x transmitting nodes with g distinct slots such that the g^{th} slot is the $(S_L + z)$ is $\psi(S - (S_L + z))^{N-x}$. Taking all possible values of z into account, the total number of possible cases of slot selection that gives exactly x transmitting nodes with g distinct slots such that at least one of the g slots lie in the range $[S_L + 1, S_U]$ is given by

$$\begin{aligned} \delta_2 &= \sum_{\substack{z=1 \\ S_U > S_L}}^{S_U - S_L} \psi(S - (S_L + z))^{N-x} \\ &= \sum_{\substack{z=1 \\ S_U > S_L}}^{S_U - S_L} \left(\prod_{k=0}^{g-1} (S_L + z - k) - \prod_{k=0}^{g-1} (S_L + z - 1 - k) \right) (S - (S_L + z))^{N-x}. \end{aligned} \quad (\text{A.6})$$

Using (A.4) and (A.6), the number of ways that g distinct slots may be chosen which will result in x transmitting nodes and $N - x$ deferring nodes is given by

$$\xi(x, g) = \delta_1 + \delta_2. \quad (\text{A.7})$$

A.3 Expected Value of c_i

The \bar{X} nodes contending to transmit in the data period choose one slot each out of the contention window S_{data} . Let c_i represent the i^{th} selected slot from the beginning of the contention window. If multiple node select the same slot, that slot is given different indices as in Fig. 4.4. The definition of c_i implies that $c_1 \leq c_2 \leq c_3 \leq \dots \leq c_{\bar{X}}$. Thus, to find the expected value of c_i , we can use order statistics [11]. Let p_j be the probability of choosing the j^{th} slot. Since each node choose a slot uniformly randomly,

$$p_j = \frac{1}{S_{data}}. \quad (\text{A.8})$$

Let us define P_j as

$$\begin{aligned} P_j &= p_1 + p_2 + \dots + p_j, \\ &= j \frac{1}{S_{data}}. \end{aligned} \quad (\text{A.9})$$

The probability of c_i being the j^{th} slot of S_{data} is given by

$$\begin{aligned} P(c_i = j) &= \sum_{k=i}^{\bar{X}} \binom{\bar{X}}{k} \left[P_j^k (1 - P_j)^{\bar{X}-k} \right. \\ &\quad \left. - P_{j-1}^k (1 - P_{j-1})^{\bar{X}-k} \right] \end{aligned} \quad (\text{A.10})$$

Then expected value of c_i is given by

$$\begin{aligned} \bar{c}_i &= E(c_i) \\ &= \sum_{j=1}^{S_{data}} j P(c_i = j) \\ &= \sum_{j=1}^{S_{data}} j \sum_{k=i}^{\bar{X}} \binom{\bar{X}}{k} \left[P_j^k (1 - P_j)^{\bar{X}-k} \right. \\ &\quad \left. - P_{j-1}^k (1 - P_{j-1})^{\bar{X}-k} \right] \end{aligned} \quad (\text{A.11})$$

Substituting expression of P_j in (A.11), we get

$$\begin{aligned}
\bar{c}_i &= E(c_i) \\
&= \sum_{j=1}^{S_{data}} j \sum_{k=i}^{\bar{X}} \binom{\bar{X}}{k} \left[\left(\frac{j}{S_{data}} \right)^k \left(1 - \left(\frac{j}{S_{data}} \right) \right)^{\bar{X}-k} \right. \\
&\quad \left. - \left(\frac{j-1}{S_{data}} \right)^k \left(1 - \left(\frac{j-1}{S_{data}} \right) \right)^{\bar{X}-k} \right] \\
&= \frac{1}{S_{data}^{\bar{X}}} \sum_{j=1}^{S_{data}} \sum_{k=i}^{\bar{X}} \binom{\bar{X}}{k} \left[j^{k+1} (S_{data} - j)^{\bar{X}-k} \right. \\
&\quad \left. - j(j-1)^k (S_{data} - (j-1))^{\bar{X}-k} \right] \tag{A.12}
\end{aligned}$$