MAC PROTOCOLS FOR WIRELESS NETWORKS : SPATIAL-REUSE AND ENERGY-EFFICIENCY

TAN HOCK LAI PAUL

NATIONAL UNIVERSITY OF SINGAPORE

2009

MAC PROTOCOLS FOR WIRELESS NETWORKS : SPATIAL-REUSE AND ENERGY-EFFICIENCY

BY

TAN HOCK LAI PAUL

(*B.Eng. (Hons), UNSW*)

A THESIS SUBMITTED
FOR THE DEGREE OF MASTER OF SCIENCE
DEPARTMENT OF COMPUTER SCIENCE
SCHOOL OF COMPUTING
NATIONAL UNIVERSITY OF SINGAPORE

2009

To my family

Acknowledgements

Foremost, I would like to express my sincere gratitude to my supervisor Dr. Chan Mun Choon for the continuous support of my part-time postgraduate studies, for his patience. Without his patient guidance, this work would not even have been possible.

My sincere thanks also goes to my employer, Thales Technology Centre Singapore, for their moral and financial support in my upgrading of myself.

Last but not least, I would like to thank my wife, Teresa, for her understanding and support throughout this entire process and for giving me three lovely children - Phoebe, Priscilla and Theodore. They have certainly provided me with the loving inspiration when I really needed it most to complete my postgraduate studies.

Table of Contents

List of Figures

List of Tables

Abstract

This thesis addresses the problem of developing MAC protocols for wireless networks, in particularly, wireless sensor networks and wireless ad-hoc network. Firstly, to provide energy-efficient and low-latency medium access in diverse traffic conditions and second, by exploiting multi-channel radio capability to provide concurrent transmissions in areas where traffic is dense or exhibits traffic funneling effect. The contributions of the thesis are as follows:

- This thesis presents *AMCM*, a traffic-adaptive multi-channel MAC protocol that increases the capacity of wireless network by enabling multiple concurrent transmissions on orthogonal channels using a single half-duplex transceiver. *AMCM* is based on the IEEE 802.11 MAC but provides fine-grain, asynchronous coordination among locally interfering nodes for channel negotiation. The protocol has several key features. Firstly, the protocol does not requires network-wide synchronization nor does it requires any dedicated control channel for channel negotiation purposes. Next, by dynamically adapting the size of the control window to varying traffic load, our protocol mimics single-channel IEEE 802.11 MAC during low load, while enabling

multiple concurrent transmissions during high load.

- This thesis presents GMAC, an energy-efficient and low-latency convergecast MAC protocol for data gathering system. GMAC adopts a synchronized low duty cycling approach to minimize the cost of idle listening by allowing network nodes to sleep most of the time. GMAC adopts a simple, low-overhead reservation-based route-aware TDMA approach to facilitate low-latency packet forwarding along a route towards the sink, thus it also minimizes both packet collisions and overhearing.

Chapter 1

Introduction

The advance in micro-technology has revolutionized the way in which information is being sensed and processed. Micro-sensors coupled with data-processing and wireless communication capabilities have made it possible for a large-scale of low-powered, low-cost, small but smart devices to collaborate among themselves to achieve larger sensing task such as an environmental monitoring application [1]. Unlike traditional networks, WSN relies on the distributive & collective effort of all sensor nodes to provide greater accuracy of the information through collaboration and online information processing [2]. Depending on the type of application, a large number (in the order of thousands) of sensor nodes can be *randomly* deployed *densely* near the region of interest. In some cases, battery-operated sensor nodes may not be convenient or possible to replenish. Such characteristics have therefore made the design of any protocols even more challenging. Earlier WSN deployments such as environmental monitoring applications collect data at a low rate, and place greater emphasis on network lifetime instead of performance. However, there is growing trend of WSN applications to support more complex operations such as target tracking and area surveillance, particularly for the military environment. Such complex operations introduce new and tough challenges that are not faced in low-rate monitoring applications. Thus, this thesis aims to identify the requirements and challenges, particularly on the data-link layer (MAC protocol), to realize such complex applications with stringent requirements.

The outline of this chapter is as follows. First, we first present an overview of WSN highlighting its characteristics, challenges, briefly the sensor platform together with its functional building blocks and some applications for WSNs to highlight the importance of adopting an application-driven approach in any protocol designs. Next, we introduce the challenges and requirements of the MAC protocol for WSN to achieve good energy-efficiency and also the opportunity to exploit multi-channel communication capability to solve several issues in WSN.

1.1 Wireless Sensor Network

A WSN is a multi-hop ad-hoc wireless network where several hundreds or even thousands of low-cost battery-powered sensor nodes with relatively high node density in the order of 20 nodes/m^3 [3] self-organize and collaborate to accomplish a common sensing task such as environment monitoring, target tracking, intrusion detection, wildlife habitat monitoring, climate control, and disaster management.

Unlike traditional ad-hoc networks, WSNs are usually battery-powered, and it is often very difficult to change batteries for all the nodes. *In such a resource-constrained communication system, it is important that all of the layers in the protocol stack are optimized to support the specific needs of the application running on top of it, rather than providing* *flexibility.* Also, the ad-hoc deployment of nodes, network scale and possible application traffic patterns pose numerous challenges which are not typically encountered in traditional ad-hoc networks.

In WSNs, the most common form of communication pattern in WSN is called convergecast, where, every sensor node reports the collected data to a sink (a distant base station) node over several multi-hop transmissions. In some deployment scenarios, energy replenishment or maintenance is impossible. Furthermore, harsh and dynamic operating environment further complicates the operation of the WSN. Therefore, the network must self-organize and also be robust and resilient to moderate node failure (e.g. energy depleted, hardware failure or external factors) and also in the presence of time-vary channel dynamics. As such, large scale of sensor (redundant) nodes are deployed in a dense and ad-hoc manner. This redundancy also means more co-related data among a group of neighbors and suggests; a need for *data-fusion* or *data-aggregation*. In general, it is assumed that the computation involved in WSNs is relatively cheap compared to the communication cost. Typically, the packet size is small (e.g. tens of bytes) and only simple computations such as aggregation are required. Therefore, the challenge here is to minimize as much communications. For example, some level of in-network processing (such data aggregation) can be perform to avoid unnecessary transmissions across the network. It is also possible for a node to turn off its radio when it does not have packets to send. More importantly, WSN differs from traditional networks in that it follows a *data-centric* communication paradigm. In WSNs, applications are not interested to know the identities of every sensor nodes, but rather the content/data. For example, monitoring application is interested if any sensor nodes have temperature above certain threshold. For now, WSNs operate under a set of constrained resources. Without a good understanding of these constraints, it is hard to design any systems that can to meet the requirements, and yet prolonging the lifetime of the network by utilizing these resources in an efficient manner.

1.1.1 Hardware Motes

Even when higher computational powers are being made available in smaller and cheaper processors, the capacity of processing and memory are still scarce resources in sensor networks. More recently, there are several (μAMPS [9], WINS [6], PicoRadio [7], SmartDust [8]) projects have attempted to integrate sensing, signal processing, and radio elements onto a single integrated circuit with the aim to enable wide-area distributed sensing. For instance, the μAMPS (micro-Adaptive Multi-domain Power-aware Sensors) [9] node is a wireless sensor node that exposes the underlying parameters of the physical hardware to the system designer. This enable a node to scale the energy consumption of the entire system in response to changes in the environment, the state of the network, and the protocol and application parameters in order to maximize system lifetime and reduce global energy consumption. Thus, all layers of the system, including the algorithms, operating system, and network protocols, can adapt to minimize energy usage.

The primary component of the data and control processing subsystem is the StrongARM SA-1110 microprocessor. Selected for its low power consumption, performance, and static CMOS design, the SA-1110 runs at a clock speed of 59 MHz to 206 MHz. The processing subsystem also includes RAM and flash ROM for data and program storage. In our experiments, we used UC Berkeley motes (Mica [10]) as the sensor nodes. Mica mote uses a single channel, 916MHz radio from RF Monolithics to provide bidirectional

Mote	WeC	rene	dot	mica	mica2	mica2 dot
Released	1999	2000	2001	2002	2003	2003
Processor	4 MHz				7 MHz	4 MHz
Flash (code, kB)	8	8	16	128	128	128
RAM (kB)	0.5	0.5	1	4	4	4
Radio (kBaud)	10	10	10	40	40	40
Radio Type	RFM				ChipCon	ChipCon
μcontroller	Atmel					
Expandable	no	yes	no	yes	yes	yes

Figure 1.1: Hardware Platform Evolution [16]

communication at 40kbps. an Atmel Atmega 103 micro-controller running at 4MHz, and considerable amount of nonvolatile storage (512 KB). A pair of conventional AA batteries and a DC boost converter provide a stable voltage source, though other renewable energy sources can be easily used. The RF transmit power of the Mica radio can be tuned to operate at different levels. The second generation of Mica platform called Mica2 uses an Atmega 128L microprocessor, with a faster processor clock running at 7.38Mhz, but the amount of programmable and data memory remains the same. The radio is based on a Chipcon [14] CC1000 FSK based tunable-RF transceiver capable of delivering 38.4kbps of raw data.

1.1.2 Operating System

TinyOS is an operating system for WSNs. It is an event-driven operating system that allows for high concurrency to be handled in a very small amount of space (kilobytes of memory). A complete system configuration consists of a tiny scheduler and a graph of components. A component has four interrelated parts: a set of command handlers, a set of event handlers, an encapsulated fixed-size frame, and a bundle of simple tasks each of which operate on its task. Each component has its tasks clearly declared to facilitate modularity. The high-level

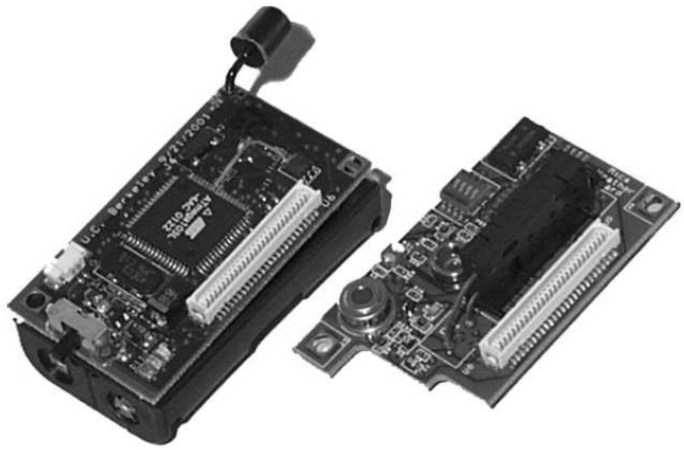


Figure 1.2: Mica Hardware Platform: The Mica sensor node (left) with the Mica Weather Board developed for environmental monitoring applications. [4]

components issue commands to lower level components and lower level components signal events to the higher level components.

1.1.3 Energy

Figure 1.3 shows a high-resolution data capture of the current consumption for transmitting a radio message. In this example the mote starts in a low power state (consuming less than 100 A), wakes up, and transmits the message. The TinyOS radio stack uses the Carrier Sense Multiple Access (CSMA) collision avoidance protocol. When using CSMA, sending a message requires the mote to listen to the radio channel to detect potential collisions before beginning transmission. The figure clearly shows the discrete power levels for each of these operations.

Most of the platforms described above are powered by batteries. In μAMPS [9], node is powered by the battery subsystem via a single 3.6V DC source with an energy capacity

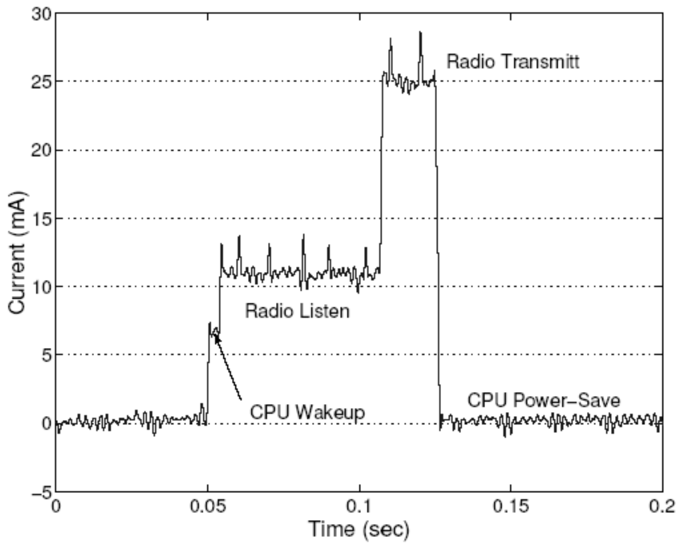


Figure 1.3: Measured current consumption for transmitting a single radio message at maximum transmit power on the Mica2 node. [16]

of approximately 1500 mAH. If the energy consumptions for various activities are known before deployment, application designer can tune (e.g. sleep duty-cycle) their application accordingly so as to operate within the requirements (e.g. operational for 1 year). For example, the habitat monitoring application in [4] needs to run for nine months. Each Mica mote runs on a pair of AA batteries supplying 2200 mAh at 3 volts. Assuming the system will operate uniformly over the deployment period, each node has 8.148 mAh per day available for use. With this, the network designer can now choose how to allocate this energy budget between sleep modes, sensing, local calculations and communications. However, the power requirement for each node is location-dependent. For example, nodes near the sink node may need to forward all (route-thru traffic) messages from downstream nodes. In this case, these forwarding nodes will consume more energy than those source

Mode	Current	Mode	Current
CPU		**Radio**	
Active	8.0 mA	Rx	7.0 mA
Idle	3.2 mA	Tx (-20 dBm)	3.7 mA
ADC Noise Reduce	1.0 mA	Tx (-19 dBm)	5.2 mA
Power-down	103 μA	Tx (-15 dBm)	5.4 mA
Power-save	110 μA	Tx (-8 dBm)	6.5 mA
Standby	216 μA	Tx (-5 dBm)	7.1 mA
Extended Standby	223 μA	Tx (0 dBm)	8.5 mA
Internal Oscillator	0.93 mA	Tx (+4 dBm)	11.6 mA
LEDs	2.2 mA	Tx (+6 dBm)	13.8 mA
Sensor board	0.7 mA	Tx (+8 dBm)	17.4 mA
EEPROM access		Tx (+10 dBm)	21.5 mA
Read	6.2 mA		
Read Time	565 μs		
Write	18.4 mA		
Write Time	12.9 ms		

Figure 1.4: Power model for the Mica2. The mote was measured with the micasb sensor board and a 3V power supply. [16]

nodes which detected the event. Therefore, we need to budget our power with respect to the energy bottleneck of the network; since the network is disconnected once these forwarding nodes completely drain all their energy.

From the above observations, the system is constrained by 3 dimensions: the computation power, data storage, communication bandwidth and energy. With the limited amount of computational and storage capacity, there is a need for a *simple* and *stateless* protocol design. Since communications occur over the shared wireless medium, communication overheads (e.g. control overheads) must also be reduced to avoid unnecessary energy dissipation.

On the other hand, Moore's Law suggests that memory density and processor speed will continue to grow at an exponential rate: in ten years, devices as large as a mote will have the processing power and storage of today's server-class machines. In contrast, neither the energy density nor energy costs of communication are expected to scale in this fashion. Similarly, the radio bandwidth is not expected to scale as dramatically as processor speed or RAM capacity. Thus, future sensor networks will be computationally-rich, but still continue to be bandwidth and energy limited. In this case, it appears more energy-efficient to perform in-network (local computation to exploit the high computational power) processing in an attempt to reduce the number of transmissions.

Sources of Energy Wastage

It is important to identify possible sources of energy wastage [21], and therefore seek ways to alleviate such waste in the MAC protocol.

- Collision

  Collision occurs when two nodes transmit at the same time and therefore causes interference at the receiver. Not only is energy wasted during the transmission and reception, additional energy is required for subsequent re-transmissions. Even though the exchange of RTS/CTS messages can help alleviate the collision problem, the control overheads required to overcome this problem can be inefficient in terms of energy and utilization since application data size is usually small in such network. For time-sensitive sensing applications, repeated collisions can increase latency too.

- Overhearing

  Overhearing is a result of a node receiving packets that are not destined for it. Since energy is required to receive and decode the packets, therefore one way to conserve energy is to switch off the radio totally if a node knows that it will not be involve in any communication (reception) for some period.

- Idle listening

  In cases where a node is not aware of possible reception from one of its neighboring node, it must turn on its radio and continuously monitor or listen to the medium for any possible receptions. In WSNs where traffic is extremely low, nodes can spend most of their lifetime listening to receive possible traffic that is not sent. According to [21], idle listening consumes 50-100% of the energy required for receiving.

- Overheads

  There are several forms of overheads. Firstly, control or signaling packets consume resources too. Therefore, it is wise to measure the impact of using such overheads in overcoming its original intention. For example, in wireless sensor networks, application data is usually small (e.g. tens of bytes), therefore the use of the RTS/CTS/ACK messages can be significant. Secondly, most of the MAC protocols require some form of carrier-sensing in order to infer a *free* channel. When channel is physically sensed as busy, a backoff procedure is performed. In the presence of a large, sudden and co-related events detected at some sensor nodes, not only will the collisions increases, but also poor packet delivering factor and also unnecessary energy wastage during the channel sensing process. Ideally during this scenario, the energy consumption should be kept constant even when packet delivery ratio is low. Thirdly, switching between various radio's states requires time. Therefore, MAC protocols which leverages on periodic state transition (e.g. sleep-awake schedule in [21]) must take this into consideration.

1.1.4 Applications Requirements & Characteristics

Sensor networks may consist of many different types of micro-sensors capable of monitoring a wide range of ambient conditions such as temperature, humidity, pressure. The concept of micro-sensing and wireless connection of these nodes promise many new application areas. In general, these applications can be categorize into military, environment, health, space exploration, chemical processing, disaster relief, home and other commercial areas [5]. One example is habitat monitoring on Great Duck Island (GDI). In [4], a system architecture is proposed to address a set of system requirements for habitat monitoring. These requirements cover the hardware design of the nodes, the design of the sensor network, and the capabilities for remote data access and management. Collaborating closely with biologists from the College of the Atlantic, a network consisting of 32 nodes was deployed on a small island off the coast of Maine for monitoring seabird nesting environment and behavior.

Since WSNs are application-specific, there is a need to adopt an application-driven approach for protocol design. By taking into consideration the underlying application's requirements or specifications, unnecessary levels of abstraction can be avoided [22]. The traffic pattern also differs from traditional networks, and also varies for each application. Most applications tend to use *many-to-one* communication paradigm, whereby many sensor nodes communicate with their distant sink node in either a single or multi-hop manner. In general, some applications require either *periodic data-gathering* from the sensor nodes (source nodes), or *on-demand data-reporting*. In the former case, sensor nodes are configured to report their data periodically to the sink node. However, in the latter case, sensor nodes only report their data when specific events of interest are detected. Therefore, depending on the type of traffic types, the design of various protocols and also the coordination among sensor nodes can vary drastically.

A classification of data delivery models in WSNs and the corresponding requirements is presented in [15]. Depending on the application requirements, there are three basic data delivery models: continuous model, query-driven model, and event-driven model. In the following, we explain the characteristics of these models:

- Continuous Data Delivery: In this model, sensor nodes transmit the collected data at periodic intervals. It is the basic model for traditional monitoring applications based on data collection. The data rates are usually low and to save energy the radios can be turned on only during data transmissions.

- Query-Driven: In this model, sensors only report data in response to an explicit request from the sink. The response to the query provides the user with a snapshot of the monitored conditions or a stream of data for a short interval. The sink may also initiate a query to reconfigure/reorganize the sensor nodes such as upgrading the system software running on the nodes.

- Event-Driven: In this model, sensor nodes report data only if an event of interest occurs. Usually, the events are rare. Yet, when an event occurs, a burst of packets is often generated that needs to be transported reliably, and usually in real-time, to a base station. The success of the network depends on the efficient detection and notification of the event that is of interest to the user.

Traditionally, WSNs tend to exhibit specific data delivery model in the network. For example, in environmental monitoring, it exhibits the continuous data delivery model, which is the typical data-collection applications where delay and loss of data may be tolerated. For such continuous data-collection applications, prolonging the network lifetime is more critical than performance such as throughput or bandwidth utilization. Therefore, sensors are usually configured to report their data in larger (depending on application requirements) time intervals, so as to conserve energy by turning off their radio/transceiver - These networks are idle most of the time. In the query-driven model, tolerance of delay depends on the query characteristics. If the query requests streams of data to be collected quickly, large amounts of data may need to be delivered in a short period. Throughput, timely delivery of data and bandwidth may become important concerns. In the event-driven model, bursty-traffic generated in case of an event needs to be delivered to the sink node as quickly and as reliably as possible. In this model, the network should be able to provide high throughput and timely delivery of the data.

1.2 Challenges in Energy-Efficiency

In most application scenarios where energy replenishment is impossible, sensor nodes must operate in an energy-efficient manner to perform their sensing task for as long as possible and at the same time, satisfy their application requirements or performance metrics such as throughput, latency and information fidelity. Energy-efficiency is thus the critical performance metric and usually, the primary objective of maximizing the network lifetime. In fact, the design goal of most sensor MAC protocols is energy conservation and is achieved at the expense of other performance metrics. The challenge is therefore to achieve an optimal tradeoff between energy and performance. This task becomes more challenging with diverse set of applications' requirements.

From section 1.1.3, it is clear that the communication activity of sensor nodes is more energy-consuming than other activities such as sensing and computation. With this knowledge, most protocol designers attempt to minimize energy consumed during all radio activities such as idle listening, overhearing and retransmissions as identified in Section 1.1.3.

1.2.1 Synchronized low duty cycling

To prolong the operational network lifetime, MAC protocol designers have adopted duty-cycle approach whereby radios are turned on and off to reduce energy wasted in idle listening. The cost of idle listening is high especially in many sensor network application where there is no data to send during the period when nothing is sensed. Traditional MAC (e.g. IEEE 802.11) protocols were designed to listen actively to the channel, therefore consuming unnecessary energy. Since most sensor networks are required to operate over long period of time, and nodes will operate in idle state for a long time, therefore, idle listening is a dominant factor of radio energy consumption and thus must be minimized.

Radio duty cycling is one of the techniques to reduce the power consumption due to idle listening when there is no traffic. It is quite effective and is adopted in many sensor MAC protocols (see Section 2.2). Unfortunately, existing sensor MAC protocols achieved good result in energy conservation, but at the expense of degraded performance such as throughput and latency which are critical performance metrics for complex applications such as track tracking and area surveillance. For example, introducing low duty cycle can incur additional latency if the intended receiver follows the duty cycle period strictly. This problem is severe with increasing hop count (larger networks), even in a low-load network.

1.2.2 Scheduled-based transmission

In WSNs, most MAC protocols are contention-based (CSMA) schemes such as S-MAC [21] and B-MAC [49]. CSMA-based approach is commonly used due to its simplicity, adaptivity and robustness. More importantly, it does not require clock synchronization or information about the global network topology. CSMA-based protocols have a lower delay and promising throughput potential at lower traffic loads, which generally happens to be the case in WSNs. However, additional collision avoidance or collision detection methods should be employed to handle the collision possibilities, especially in dense network or synchronized transmissions resulting from similar event detection. Unfortunately, the cost of collisions under high contention and also required protocol overheads (e.g. RTS-CTS handshake) to avoid collisions due to hidden-terminals, make CSMA-based approach not efficient. On the other hand, scheduled-based approach such as TDMA has a natural advantage of collision-free medium access. However, the overhead incurred in ensuring clock synchronization and efficiency in slot utilization still remain a challenge in a dynamic WSNs where topology can change. For the latter, transmissions over scheduled/dedicated time slots result in higher delays and decreased throughput as compared to CSMA-based approach.

1.2.3 Parallel Communications

Existing sensor devices provide very limited single-channel bandwidth, 19.2Kbps in MICA2 [10] and 250Kbps in MICAz [11] and Telos [12], it is imperative to design multi-channel MACs that can achieve a higher throughput through parallel communications. While existing hardware such as CC2420 radio [13] (found in MICAz and Telos motes) already provides multiple physical channels, most sensor MAC protocols currently are designed to achieve better energy-efficiency and throughput.

While there are several multi-channel MAC protocols designed for ad-hoc networks, these designs are not applicable directly on WSNs due to the several challenges. Firstly, sensor devices must be simple (in terms of computation and hardware configuration) and energy-efficient. Therefore, only a single radio transceiver can be used. Second, since WSN exhibits very limited communication bandwidth, therefore, any control messages to facilitate multi-channel communications must be smaller than typical the length of WSN data packets (20-50 bytes).

1.3 Contributions & Report Organization

This thesis makes several contributions, each addressing the challenges described in Section 1.2.

1.3.1 Adaptive Multi-Channel MAC Protocol (AMCM)

The availability of multi-channel hardware capability in existing sensor devices paved the way for a multi-channel sensor MAC protocol to exploit parallel communications in WSNs.

This is the motivation of our design for a high-throughput traffic-adaptive CSMA/CA-based MAC protocol called Adaptive Multi-Channel MAC Protocol (AMCM). One key feature of AMCM is that nodes dynamically negotiate and switch channel in a *distributed and asynchronous* manner. There is no static negotiation period or pre-assigned dedicated channel for negotiation. Instead, the protocol let nodes dynamically synchronize/align themselves locally to a common notification window for secondary channel acquisition. In addition, the duration of NW and reservation duration per channel are adapted according to the traffic load and topology. We also performed extensive simulations to study the performance under both infrastructure WLAN (single-hop) and multi-hop wireless networks and concluded that AMCM adapts well to varying traffic load and that, given a *N-channel* wireless networks, our single transceiver solution achieved nearly $N\times$ performance gain over single-channel network. The key contributions of the AMCM design are as follows.

- We proposed AMCM, a novel multi-channel MAC protocol, which improves spatial reuse through parallel communications over orthogonal channels.
- We compared the performance of AMCM against existing single- and multi-channel protocols through ns-2 simulation.

1.3.2 Energy-efficient Low-Latency MAC Protocol (GMAC)

Existing sensor MAC protocols are designed with a key focus primarily on energy-efficiency, but at the expense of performance such as latency, throughput and reliability. Motivated by these observations, this thesis describe the design an energy-efficient, low-latency duty-cycle MAC protocol for data gathering system. The key contributions of the GMAC design are as follows.

- We propose GMAC to achieve energy-efficiency, performance and adaptivity. GMAC adopts a TDMA-like approach to provide collision-free transmissions with good channel utilization at low load. GMAC achieves lower packet forwarding latency through route-aware scheduling.
- We compared the performance of GMAC against RMAC (published in INFOCOM 2007 paper) using ns-2 simulation.

1.3.3 Report Organization

The rest of the report is organized as follows. Chapter 2 surveys multi-channel MAC protocols for wireless ad-hoc networks and also energy-efficient MAC protocols for WSNs. Chapter 3 presents the design and evaluation of our traffic-adaptive multi-channel MAC protocol for wireless ad-hoc network. In Chapter 4, the design and evaluation of GMAC protocol is presented. Finally, 5 concludes the dissertation.

Chapter 2

Literature Review

A Medium Access Control (MAC) protocol decides when competing nodes may access the transmission channel, and tries to ensure that no two nodes are interfering with each others transmissions. This channel allocation or multiple access problem is challenging since collisions resulting from two nodes sending data at the same time can increase energy cost due to both corrupted transmission and follow-on retransmissions. Existing sensor MAC protocols focus on a single most important goal - energy efficiency. Unfortunately, there is a need for new sensor MAC protocols to also meet traditional goals such as delay, throughput, channel utilization and fairness.

In this Chapter, we first present the IEEE 802.11 MAC protocol to better understand and appreciate its basic design when used in wireless networks. We then survey existing multi-channel MAC protocols for wireless ad-hoc network for increased throughput through spatial reuse. Next, we survey several energy-efficient sensor MAC protocols in order to understand both their strengths and weaknesses.

2.1 Multi-Channel MAC Protocol for Wireless Ad-hoc Networks

IEEE 802.11 [26] is the *de-facto* wireless networking standard for wireless local area network (WLAN). Currently, the standard has four specifications which includes IEEE 802.11, IEEE 802.11a, IEEE 802.11b and IEEE 802.11g. Each of these specifications differs in their operating frequency range, modulation scheme and transmission speed. The standard also supports the use of multiple channels. This enables multiple transmissions to take place simultaneously without causing interference to each other. Clearly, by exploiting multiple channels, the capacity of the wireless network can be increased. Unfortunately, the original MAC protocol is designed for single-channel wireless network and thus cannot capitalized on this multi-channel capability.

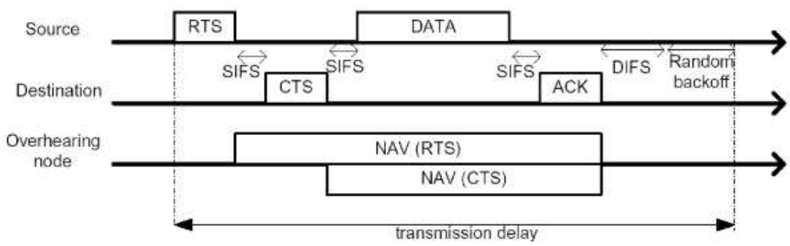


Figure 2.1: Distributed Coordination function.

Specifically, IEEE 802.11 standard defines both the Physical layer (PHY) and the Medium Access Control (MAC) layers. The PHY layer specifies the physical modulation scheme used and signaling characteristics for the transmission through radio frequencies whereas the latter specifies rules to access the shared medium. The MAC layer supports two modes of operations. The first mode is the point coordination function (PCF) and the second mode is the distributed coordination function (DCF) as shown in Figure 2.1. The DCF mode uses the carrier sense multiple access with collision avoidance (CSMA/CA) protocol. In this mode, when a station wants to transmit data packets over the shared medium, it must first sense (physically) the wireless medium. If the medium is sensed busy, it randomly chooses a *backoff counter* to wait and then re-attempt to contend for the medium. To reduce the effect of *hidden-terminals*, the standard also specifies the use of short control messages prior to the exchange of the large data packets. Specifically, after sensing an idle channel, a station transmits *Request-to-Send* (RTS) to the receiving station, which then responds with *Clear-to-Send* (CTS). Neighboring stations will then be aware of the upcoming transmission; therefore defer their access until the end of the transmission indicated in the *Network Allocation Vector (NAV)* field in both control frames. Unfortunately, the effectiveness of RTS/CTS frames reduces when the network traffic increases since these control frames are broadcast messages, and are therefore also prone to collision.

2.1.1 Challenges

Common Problems in Wireless Ad-hoc Problems

A hidden terminal is one that is unaware of a transmission in its vicinity and its attempt to transmit will eventually cause collision at the receiving node. In our example shown in Figure 2.2, host C is the hidden node since host B lies in between the transmission range of both host A and C; both host A and C are mutually hidden since they cannot sense each other's transmissions. Fortunately, this hidden-terminal problem can be alleviated by extending the DCF basic mechanism through a *virtual carrier sensing* mechanism that is based on the exchange of RTS and CTS) control frames.

Exposed nodes are complementary to hidden nodes. An exposed node is one that is

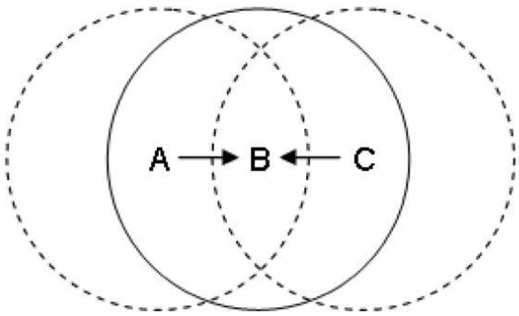


Figure 2.2: Hidden-terminal Problem: Host C cannot sense the transmission from host A, thus causing collision at host B when it attempts to transmit to host B.

within the sensing range of the sender but outside the interfering range of the destination. In Figure 2.3, node C must defer its transmission with node D due to the ongoing transmission between node A and B. The IEEE 802.11 MAC uses carrier sense with sender-initiated RTS/CTS handshake to alleviate hidden node problem. Traditionally, IEEE 802.11 DCF is designed for wireless LAN (infrastructure networks), and therefore performs badly in multi-hop wireless networks due to an increase in both hidden/exposed terminals.

Impact of Location-dependent Interference

There are several works [19, 20] on studying the performance of IEEE 802.11 MAC protocol in multi-hop wireless networks. The RTS/CTS exchange is proposed to counter the problem of hidden-terminal problems. However, this solution is based on an basic assumption that all nodes are within the transmission range of receivers. This can be understood since IEEE 802.11 MAC protocol was originally designed for single-hop wireless LAN environments. In this environment, all nodes are within transmission range of either transmitters or receivers. However, in multi-hop networks such MANETs, some nodes which

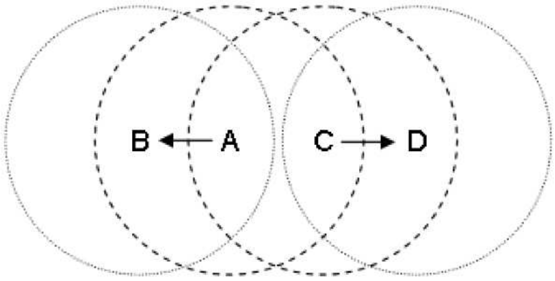


Figure 2.3: Exposed-terminal Problem: Host C cannot transmit to host D since it has earlier detected that the channel has been reserved by host A. Therefore, host C must wait until host A completes its current transmission.

are not within the transmission range of the receiver, but still within the interference range will cause serious problems at the receiver. Before proceeding, it is essential to understand the radio ranges related to a wireless radio.

- Transmission Range (R_{tx})

  The range within which a packet is successfully received assuming no interference (at the receiver) from other transmitters.

- Carrier Sensing Range (R_{cs})

  The range within which a transmitter can detect carrier signal. Once detected, the channel is considered busy and therefore performed the backoff procedure.

- Interference Range (R_i)

  The range within which receiver will not be able to receive (decode) any packets since packets are corrupted due to interfering transmissions. According to [19], the interference range is not a fixed range. Rather it is essentially related to the distance between the transmitter and receiver. In some situations, the interference range can goes far beyond the transmission range, resulting in various problems.

When a signal is propagated from a transmitter to a receiver, whether the signal is valid at the receiver largely depends on the receiving power at the receiver. Given transmission power (P_t), the receiving power (P_r) is mostly decided by pathloss over the transmitter-receiver distance, which models the signal attenuation over the distance. Other factors include multi-path fading, shadowing, environment noise etc. Here we ignore these factors since they are minor factors in the open space environment. According to [18], in the open space environment, the receiving power (P_r) of a signal from a sender d meters away can be modeled as equation 2.1.

$$P_r = P_t G_t G_r \frac{h_t^2 h_r^2}{d^k} \tag{2.1}$$

where G_t and G_r are antenna gains of transmitter and receiver respectively, h_t and h_r are the height of both antennas.

According to [18], k reflects the rate in which signal decays. The larger it is, the faster the signal attenuates. In the open space environment, the two-ray ground pathloss model is generally adopted. Within this model, when the transmitter is close to the receiver, receiving signal power is inverse proportional to d^2. When their distance is larger (e.g. outside of Freznel zone), the receiving signal power is then inverse proportional to d^4. Another common pathloss model used in wireless networks is the free-space pathloss model, which has k as 2. A signal arriving at a receiver is assumed to be valid if the Signal-to-Noise

Ratio (SNR) is above a certain threshold (T_{SNR}). Now, we assume a transmission is going from a transmitter to a receiver with transmitter-receiver distance as *d* meters and at the same time, an interfering node *r* meters away from the receiver starts another transmission. Let P_r denote the receiving power of signal from transmitter and P_i denote the power of interference signal at the receiver. Then, SNR is given as SNR = P_r/P_i. Therefore,

$$SNR = \frac{Pr}{Pi} = (\frac{r}{d})^k \geq T_{SNR} \tag{2.2}$$

$$r \geq \sqrt[k]{T_{SNR}} * d \tag{2.3}$$

Therefore, in order for the receiver to correctly decode packets (SNR $\geq T_{SNR}$), the interfering nodes must be at least $\sqrt[k]{T_{SNR}}*d$ meters away from the receiver. For example, T_{SNR} is usually set to 10. For a two-ray ground pathloss model with *k* set to 4, we have interference range as $R_i = \sqrt[4]{10}*d = 1.78*d$. When *d* is larger than $0.56*R_{tx}$ ($d \geq R_{tx}*T_{SNR}^{k-1}$). Therefore, with higher interference range relative to the transmission range, it only takes a small transmission power to interfere with the packet reception.

From Figure 2.4, when the transmitter-receiver distance *d* exceeds $0.56*R_{tx}$, the effectiveness of RTS/CTS handshake drops rapidly. This reduction is due to collisions as a result of large interference range and also hidden-terminal problem.

Impact of Interference Range on Data Forwarding

As we see later in the section, the effect of overhearing range which is limited by the radio sensitivity can affect the continuous flow of data towards the sink node. Since nodes which are more than two hops away from the receiver are not aware of the ongoing data reporting,

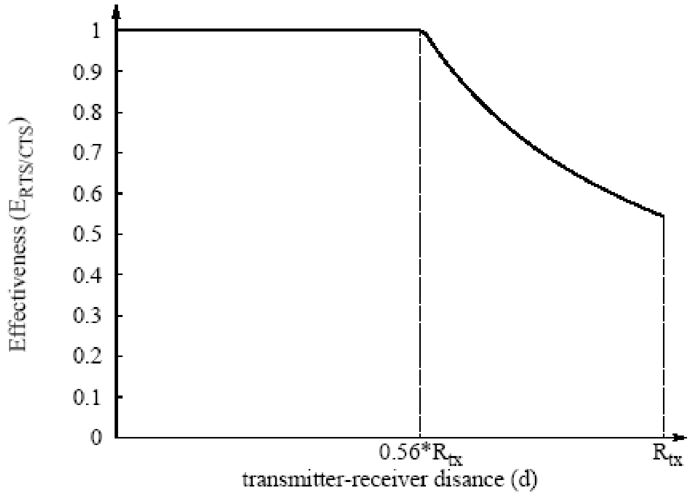


Figure 2.4: Effectiveness of RTS/CTS handshake for two-ray ground model with SNR threshold as 10 [19]

therefore they return to their basic sleep schedule which results in an increase in the sleep latency.

2.1.2 Multi-Channel MAC Protocols

Several methods have been proposed to increase the capacity of wireless networks such as IEEE 802.11 DCF enhancements [27, 28, 29] , the use of directional antennas [30, 31, 32] and multi-channel MAC [33, 34, 35, 36, 37] protocols.

So and Vaidya proposed Multi-channel MAC (MMAC) [33], a single-transceiver solution which uses the *Ad Hoc Traffic Indication Messages (ATIM)* to perform channel reservation. MMAC requires nodes to be synchronized such that every node can start the beacon interval at about the same time. Unfortunately, this tight synchronization requirement can be a problem in multi-hop networks. Even though MMAC uses all available channels for data exchange, the overheads incurred by the periodic beacon transmissions and ATIM packets can result in lower performance gain over IEEE 802.11 MAC.

Another single-transceiver solution is SSCH [34]. It differs from ”rendezvous” channel coordination mechanism (such as [33]) whereby nodes periodically meet on the primary channel to perform channel negotiations. In contrast, SSCH adopts a pseudo-random sequence to allow nodes to decide which channel to switch for the next 10ms. This duration is chosen as a tradeoff between the channel switching overheads and forwarding delay in multi-hop wireless networks.

Nasipuri et al. [35] propose a *soft channel reservation*-based multi-channel CSMA protocol. It assumes that each node can listen to all *N* channels simultaneously. To transmit, the sender must first search for an idle channel. When more than one idle channel exists, the channel that was used during the last transmission is always preferred; thus *soft-reservation*. This protocol has low control overheads, but unfortunately increases the hardware cost and complexity of the node since *N* transceivers are required.

Wu et al. [36] propose an on-demand dynamic channel assignment protocol (DCA) which assigns a dedicated channel for control purposes, and other channels for data. As such, DCA requires each node to be equipped with *two* transceivers. The idea is to listen to both control and data channels at the same time. The channel assignment/negotiation is done during the RTS/CTS exchange. One of the advantages of DCA is the non-existence of *multi-channel hidden-node problem* since nodes always listen to the control channel. Apart from increased in per-node hardware cost, DCA requires the use of a dedicated control channel in IEEE 802.11b (3 channels) results in 33% of the total bandwidth as the control overhead and possible poor channel utilization. With higher number of channels available (e.g. IEEE 802.11a), *control channel saturation* [33] problem can arise since all data channel assignments/negotiations are performed over a single control channel.

Another multi-transceiver solution is the Multi-radio Unification Protocol (MUP)[37]. The authors proposed a new link layer protocol that coordinates multiple IEEE 802.11 radios operating over multiple channels.

2.2 Energy-Efficient MAC Protocols for WSNs

This section surveys several MAC protocols for WSNs and highlights their design and limitations. We choose to focus on synchronized duty-cycle MAC protocol for its simplicity, energy-efficiency and adaptivity.

WSNs are expected to operate for months if not years on small inexpensive batteries with limited lifetimes. Therefore energy efficiency is typically the primary goal in these networks.

Previous works [21] have identified that the sources for major causes of energy waste are (i) collision, (ii) idle listening (iii) overhearing and (iv) control overheads. Among all, idle listening of the radio is a major source of energy wastage. Measurements on existing sensor device radios show that idle listening consumes nearly the same power as receiving. Specifically, idle listening is the time that the node is awake listening to the medium even though no packets are being transmitted to that node. In sensor network applications where the traffic load is very light most of the time, it is therefore desirable to turn off the radio when a node does not participate in any data delivery.

One of the primary approach for achieving low energy operation in energy-constrained

WSNs is duty cycling. In this approach, each sensor node periodically cycles between an awake state and a sleep state. Since the period of a duty cycle is equivalent to its sleep time plus awake time, it is obvious that to conserve more energy, ones have sleep most of the time, thus leading to a lower duty cycle. For example, nodes can wake up only 1ms every 100ms. This results in 1% duty cycle. Unfortunately, introducing such low duty cycle can incur additional latency if the intended receiver follows the duty cycle period strictly. This problem is severe with increasing hop count (larger networks), even in a low-load network.

In traditional ad-hoc networks, classical MAC protocols (e.g. IEEE 802.11) consumes too much energy due to idle listening - listening to receive messages that are never sent. Moreover, the nature of the sensor network applications (e.g. low data rate, small message size, event-based) means that sensor nodes are doing nothing for 99% of the time. It is thus not hard to understand why most MAC protocols for sensor networks adopt the duty-cycle approach to reduce the cost of idle listening.

Standard MAC protocols developed for duty-cycled WSNs can be roughly categorized into synchronized and asynchronous approaches, along with hybrid combinations. These approaches share a common primary goal - reduce idle listening. Hybrid protocols ([52] [53]) combine a synchronized protocol with asynchronous Low-Power Listening (LPL) approach.

2.2.1 Synchronized Approach

Synchronized protocols negotiate a common schedule that specifies when nodes are awake and asleep within a frame. Since all nodes know when is the time to be awake for communication, idle listening is therefore reduced significantly.

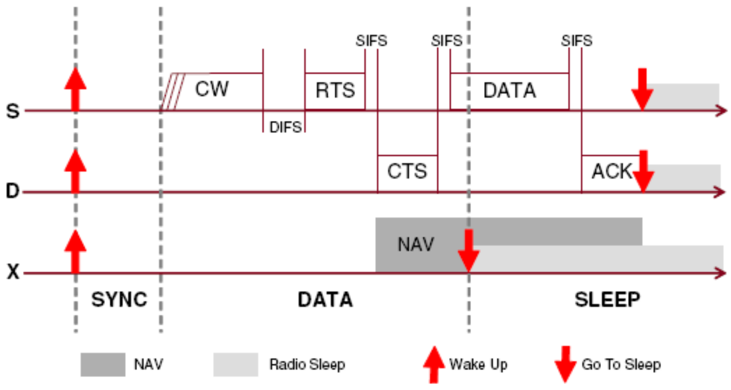


Figure 2.5: S-MAC: A typical duty-cycle MAC protocol for sensor networks

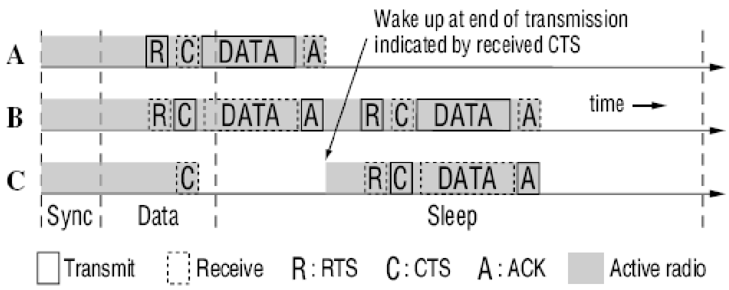


Figure 2.6: SMAC with adaptive listening: Node A sending packet to destination node C

S-MAC (2002, IEEE Infocomm)

S-MAC [21] is one of the first synchronized periodic duty-cycle based MAC protocol for sensor networks. S-MAC divides time into operational cycles (or frames) with an active period and a sleeping period. During sleep periods, the radios are completely turned off, and during active periods, they are turned back on to transmit and receive messages. Each operational cycle is divided into three periods: Sync, Data and Sleep. During the Sync period, nodes wake up to synchronize their clocks with their neighbors. During the Data period, all nodes remain active. If a node has a packet to send to a neighbor node, they exchange Request-to-Send (RTS) and Clear-to-Send (CTS) frames during the Data period, followed by the transmission of the data packet and the return of an Acknowledgement (ACK) frame. Note that this data transmission can extend into the Sleep period. Nodes not involved in communication initiated during the Data period return to the sleep state at the start of the Sleep period; other nodes return to the sleep state only after completion of the ACK frame.

While S-MAC reduces idle listening, it incurs significant latency in multi-hop packet forwarding since a packet can be delivered over only a single hop in an operational cycle (single active/sleep period). This sleep latency increases proportionally with hop length. Clearly, this deficiency is unacceptable for time-critical applications and large networks.

In a later paper [47][48], when a node overhears an RTS or CTS, the node wakes up for a short period of time after the transmission of the packet (for which the CTS was intended). If the node is the next-hop node, then it can immediately receive the packet from its neighbor. Therefore, adaptive listening can deliver a packet up to 2 hops per cycle.

However, adaptive listening also consumes more energy, since many neighboring nodes receive an RTS or CTS and stay awake, but only one of them is the next hop. T-MAC [47] improves on the design of S-MAC by shortening the awake period if the channel is idle. In S-MAC, the nodes will remain awake through the entire awake period even if they are neither sending nor receiving data. T-MAC improves S-MAC by listening to the channel for only a short time after the synchronization phase, and if no data is received during this window, the node returns to sleep mode. If data is received, the node remains awake until no further data is received or the awake period ends.

While adaptive listening adapts the length of the active period to the varying traffic load;

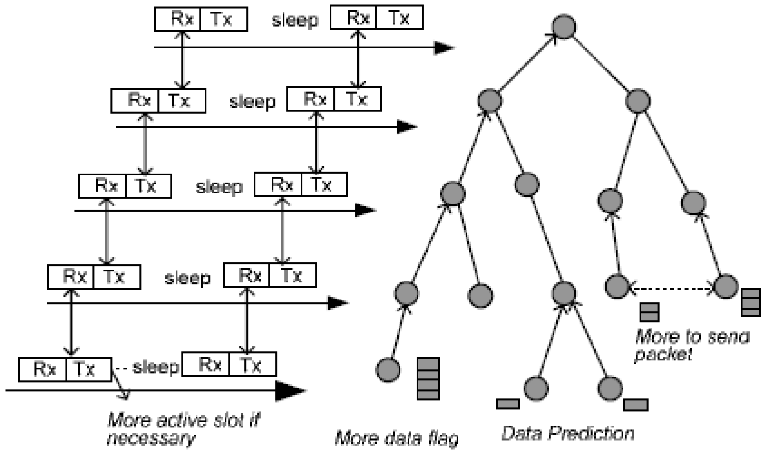


Figure 2.7: DMAC: Overview & Covergecast Tree

thus reducing sleep latency, these gains usually come at the cost of reduced throughput and increased latency. Finally, in large (longer path) and dense networks, these adaptive duty-cycle protocols can suffer from overhearing and idle listening.

DMAC (2004, IPDPS)

DMAC [51] overcame the sleep latency (data forwarding interruption problem) problem for convergecast communication pattern by employing a staggering activity schedule such that nodes along the multi-hop forwarding path are wake up sequentially like a chain reaction. DMAC also supports adaptive duty-cycle to handle varying traffic load. DMAC uses the *more-data* flag in the MAC header (DATA & ACK frames) to inform forwarding nodes along the multi-hop path to increase their duty cycle accordingly. Using a slot-by-slot renewal mechanism, DMAC can react quickly to traffic rate variations to be both energy efficient and to maintain low data delivery latency.

While staggering activity scheduling is a promising approach for future research, there are issues with DMAC. In DMAC, there is a substantial overhead in case of network topology instabilities or fluctuations since all nodes need to re-construct their schedule again. Also, local synchronization (nodes need to be aware of neighbors' schedule) at the nodes is an overhead. In dense network where a parent node can have many children, DMAC employs CSMA mechanism. Sending nodes along the same level must contend for the channel during the send interval. In order to accommodate or receive all packets from their children, parent node must adapt their duty cycle accordingly to its children's demands. Since brothers (siblings at the same level of the tree) must buffer their transmission until the next send schedule, therefore unnecessarily increase the the sleep delay. The interference between nodes with different parents could cause a traffic flow be interrupted because the nodes on the multi-hop path may not be aware of the interference. Also, collisions can be significant as children detecting a common event attempt to transmit to the parent. Lastly, not all nodes on a multi-hop path are aware of the data delivery, thus leading to interruption in forwarding.

SCP-MAC (2006, ACM Sensys)

SCP-MAC (Scheduled Channel Polling MAC, [56]) combines the advantages of LPL and scheduled protocols. In LPL, nodes poll channel asynchronously to test for possible traffic. To send a packet, the sender adds a preamble before the packet. The preamble must be at least as long as the channel polling period to ensure all receivers will detect it. The performance of LPL is sensitive to the channel polling period, since longer periods reduce receiver costs but increase sender costs. Selecting an optimal value requires knowledge of network size and completely periodic traffic [56].

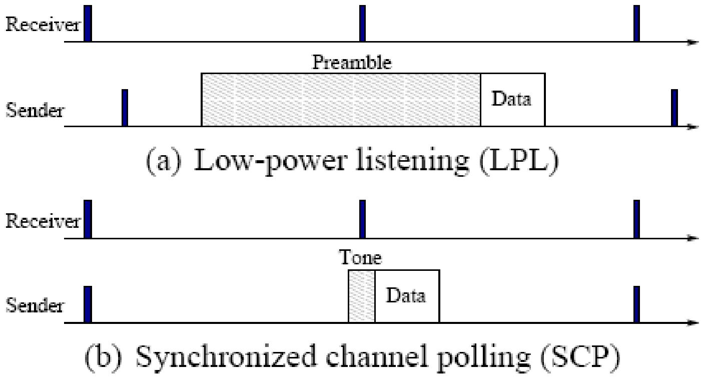


Figure 2.8: SCP-MAC

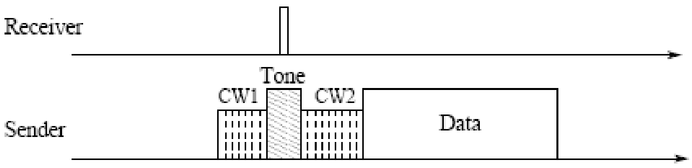


Figure 2.9: SCP-MAC: Two-phase contention in SCP-MAC - First, the sender transmits a short wakeup tone timed to intersect with the receivers channel polling. After waking up the receiver, the sender transmits the actual data packet (RTS-CTS-DATA-ACK).

The key insight of SCP-MAC is to first have nodes that have data to send contend (preferably using the Sift distribution, but SCP-MAC operates with a uniform distribution) in a first contention window for tone (shorter length compared to LPL) transmission. Possible collisions in tone transmission are allowed, because what is important is the presence of the tone. The potential receiving nodes poll the media for short time (around 2-3 ms), just enough to detect the tone. If there is no tone, the receiving nodes return to the sleep state. If there is a tone, they remain woken up for a further data transmission.

In SCP-MAC, actual data transmission is performed during a second contention window with reduced contenders from the first window. The novel introduction of small windows minimize the collisions from multiple senders during high traffic load. However, there is still non-negligible amount of collisions in the second window due to the grouping of all communication in the slotted active period. Unlike LPL protocols, the length of the busy tone together with the scheduled polling is more energy-efficient. There is a significant dependency between duty-cycle and transmission delay. In SCP-MAC, operating in ultra-low duty-cycle requires longer polling periods which then imply that the hop-by-hop delay will increase. This dependency can pose a problem in large sensor network with longer path. In this case, the dual requirement of operating in ultra-low duty-cycle mode and low-latency packet delivery can be challenging or problematic.

RMAC (2007, IEEE Infocom)

RMAC (Routing enhanced MAC) [54] exploits cross-layer routing information to reduce latency in multi-hop forwarding without sacrificing energy efficiency. In RMAC, a control frame, called a Pioneer frame (PION), is forwarded over multiple hops (e.g. A → B → C)

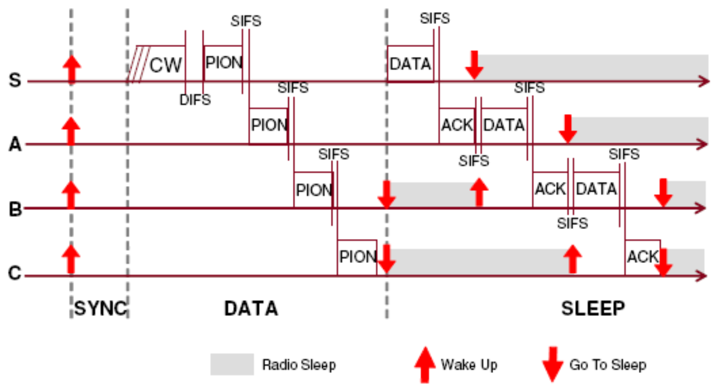


Figure 2.10: RMAC: Overview

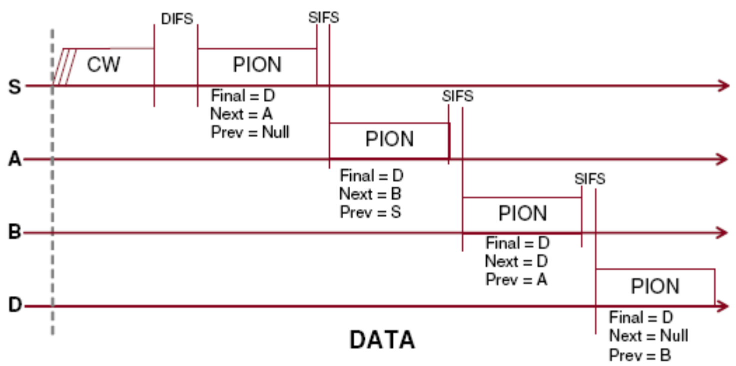


Figure 2.11: RMAC: PION transmission example - A node sends a PION to allocate the transmission time along the routing path.

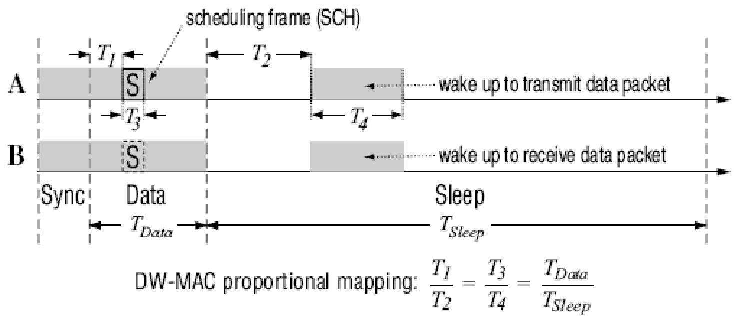


Figure 2.12: DW-MAC: Overview of scheduling in DW-MAC

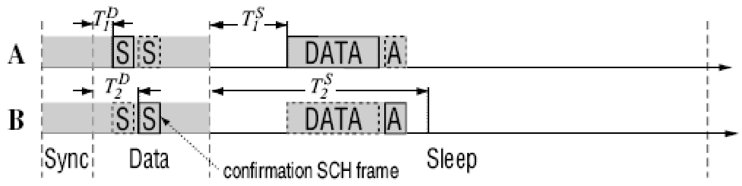


Figure 2.13: DW-MAC: Unicast in DW-MAC

during a Data period in order to inform nodes B and C when to wake up during the Sleep period to receive or transmit the corresponding data packet. The number of hops over which RMAC can forward a data packet during an operational cycle is limited by the duration of the Data period but may be set to any value depending on the parameters used.

As noted in [55], RMAC does not mitigate hidden-terminal problem. This occurs when a source node always starts transmitting a data packet at the beginning of a Sleep period when two hidden sources that successfully performed PION scheduling during the Data period. This eventually cause collisions at the beginning of the next Sleep period.

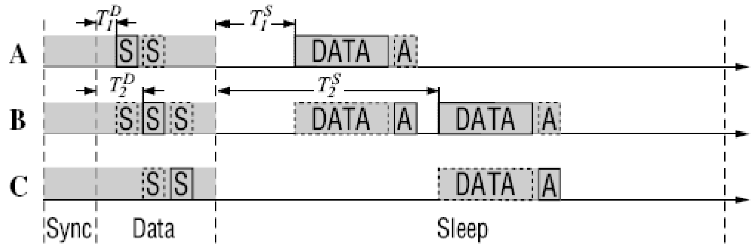


Figure 2.14: DW-MAC: Optimized multihop forwarding of a unicast packet. Node B sends an SCH to wake up node C at the time indicated by T_2^s and confirms the SCH received from node A

DW-MAC (2008, ACM MobiHoc)

DW-MAC (Demand Wakeup MAC, [55]) is a synchronized duty cycle MAC protocol, where each cycle is divided into three periods: Sync, Data, and Sleep (Figure 2.12). We denote the duration of each period by T_{Sync}, T_{Data}, and T_{Sleep}, respectively. The basic concept of DW-MAC is to wake up nodes on demand during the Sleep period of a cycle in order to transmit or receive a packet. This demand wakeup adaptively increases effective channel capacity during a cycle as traffic load increases, allowing DW-MAC to achieve low delivery latency under a wide range of traffic loads including both unicast and broadcast traffic. DW-MAC is unique in the way it schedules nodes to wake up during the Sleep period of a cycle. In DW-MAC, medium access control and scheduling are fully integrated. In a Data period, a node with pending data contends for channel access using a CSMA/CA protocol as in IEEE 802.11. DW-MAC, however, replaces RTS/CTS with a special frame called a scheduling frame (SCH). The interval of time during which the transmission of a SCH occupies the wireless medium automatically and uniquely reserves the proportional interval of time in the following Sleep period for transmitting and receiving the pending data packet. Essentially, DW-MAC sets up a one-to-one mapping between a Data period and the following Sleep period. An SCH carries no timing information, and the transmission of an SCH simply replaces that of RTS/CTS for medium access control. In this way, DWMAC minimizes scheduling overhead. As in an RTS, an SCH contains the destination address, so this SCH wakes up only the intended receiver, minimizing energy consumption due to unnecessary wake-ups. Furthermore, this integration ensures that data transmissions do not collide at their intended receivers.

While the idea of one-to-one mapping between a Data period and the following Sleep period is interesting. It introduces some implementation issues when operating under extremely low duty cycles. This is due to the proportional time for a data packet based on the size of an SCH frame, as indicated by our mapping function, can become unnecessarily long due to the very large ratio of T_{Data} to T_{Sleep}. Therefore, it results to poor channel utilization and also significant amount of energy is consumed inside the remaining Sleep period. DW-MAC does not support well multi-hop forwarding in a single operational cycle due also to the mapping function. Lastly, it is not sure if DW-MAC can deliver good throughput, latency, delivery and energy performance for moderate-to-high traffic load as the simulation were performed with rather low event rate (1 event per 200 seconds) - with 500 meter sensing range, there are only 15 events.

2.2.2 LPL-based Protocols

Low-power listening (LPL) is an approach where the channel is sampled very briefly - in an uncoordinated fashion - for presence of activity rather than for specific data. This approach is effective in reducing the energy consumed in idle listening since LPL is about 10 times less expensive than listening for full contention period in existing scheduled protocols [56]. Therefore, LPL protocols are quite attractive as compared to existing scheduled protocols especially in lightly loaded networks. In addition, lacking the need to synchronize simplifies the LPL implementation and reduces code and memory size.

Ironically, idle listening is reduced in asynchronous protocols by shifting the burden of synchronization to the sender. When a sender has data, the sender transmits a preamble that is at least as long as the sleep period of the receiver. The receiver will wake up, detect the preamble, and stay awake to receive the data. This allows low power communication without the need for explicit synchronization between the nodes. The receiver only wakes for a short time to sample the medium, thereby limiting idle listening.

While LPL approach is simple, asynchronous, and energy-efficient, it is not. Unfortunately, existing LPL-based protocols [49][50][57][58] have three major problems. First, receiver and polling efficiency is gained at the much greater cost of senders. In fact, the duty cycle is limited to 1 - 2% because the polling frequency needs to balance the cost on sending preambles and polling the channel. Second, it is sub-optimal in terms of energy consumption at both the sender and receiver. First, it does not efficiently reduce idle listening since receiver typically has to wait the full period until the preamble is finished before the data/ack exchange can begin, even if the receiver has woken up at the start of the preamble. It is also not efficient for the transmitter to send unnecessary long preamble. Second, it does not efficiently reduce overhearing cost in particularly in dense network. In this case, receivers who are not the target of the sender also wake up during the long preamble and have to stay awake until the end of the preamble to find out if the packet is destined for them. Recently, there are approaches (X-MAC [59]) to tackle this long preamble problem by replacing the original preamble with slowly increasing preamble length. The idea here is to use initially short preamble and increasing it slowly to catch the neighbor. Third, while LPL protocols can be optimized for known, periodic traffic, expected neighborhood size and traffic rate, its performance may significantly degrade at bursty and varying traffic loads. Fourth, it does not solve the latency issue in multi-hop network since the target receiver has to wait for the full preamble before receiving the data packet. Finally, recent hardware shift towards packet-based radios (e.g. IEEE 802.15.4) deprives new LPL protocols from low-level techniques like varying preamble size; thus making LPL approach less attractive.

2.3 Opportunity of Multi-channel Communications in WSNs

WSNs can be considered as a sub-class of wireless ad hoc networks. Unlike wireless ad-hoc networks which are usually resource-rich, there are several differences between them and WSNs. In WSNs, sensor nodes are simple devices equipped with simple radios. These radios are bandwidth-limited (e.g. $<$ 250kbps) as compared to much higher data rates available in wireless ad-hoc networks. As we see in Section 2, most multi-channel MAC protocols extends the basic IEEE 802.11 protocol to provide channel coordination. Such complex coordination algorithm may not be feasible on sensor devices. In this thesis, we focus on the use of multi-channel communication to increase the throughput through better spatial reuse. However, there are also exist studies that utilize multi-channel communication for other objectives. In [39], channel surfing mechanisms have been introduced such that the jammed nodes dynamically change their operating frequency. The idea of applying multi-channel clustering [40] where the nodes that hold correlated data are clustered together and communicate on the same frequency, which is different from the communication frequency of the other clusters. The Typhoon protocol [41] adopts multi-channel communication to provide reliable data dissemination from the sink to the sensor nodes. Specifically, it uses channel switching to reduce contention in the broadcast medium which, in turn, reduces the completion time of data dissemination. In alleviating congestion due to contention and interference, [42] proposes the use of multi-channel communication for this purpose.

Recently, [43] proposed the MMSN multi-frequency MAC protocol especially designed for WSNs. It is a slotted CSMA protocol and at the beginning of each time slot nodes need to contend for the medium before they can transmit. MMSN assigns channels to the receivers. When a node intends to transmit a packet it has to listen for the incoming packets both on its own frequency and the destinations frequency. A snooping mechanism is used to detect the packets on different frequencies which causes the nodes to switch between channels frequently. MMSN uses a special broadcast channel for broadcast traffic and the beginning of each time slot is reserved for broadcasts. TMCP [44] is a tree-based multi-channel protocol for data collection applications. The goal is to partition the network into multiple subtrees while minimizing the intra-tree interference. The protocol partitions the network into subtrees and assigns different channels to the nodes residing on different trees. TMCP is designed to support convergecast traffic and it is difficult to have successful broadcasts due to the partitions. Contention inside the branches is not resolved since the nodes communicate on the same channel. Lastly, Y-MAC [45] is another recent multi-channel MAC protocol designed for WSNs. Similarly, it is based on scheduled access. However, time slots are not assigned to the senders but to the receivers. At the beginning of each time slot, potential senders for the same receiver contend for the medium. Each time slot is long enough to transmit one data message. If multiple packets need to be transmitted, then the sender and the receiver hop to a new channel according to a predetermined sequence. Other potential senders also follow the hopping sequence of the receiver. As we mentioned, increased contention especially around the sink node with high data rate scenarios is hard to solve with contention-based protocols.

Chapter 3

Adaptive Multi-Channel MAC Protocol

This chapter presents AMCM, a traffic-adaptive multi-channel MAC protocol that increases the capacity of wireless network by enabling multiple concurrent transmissions on orthogonal frequency channels using a single half-duplex transceiver. AMCM is based on the IEEE 802.11 MAC but provides fine-grain, asynchronous coordination among locally interfering nodes for channel negotiation. By incorporating load-awareness, channel availability awareness and batch transmissions, our window-based approach achieves high channel utilization under varying load, while avoiding the control-window saturation problem as the number of channels increases. For single-hop scenarios, we show that, at low load, AMCM is comparable to IEEE 802.11 MAC, while under high load, AMCM delivers almost $N\times$ improvement gain over IEEE 802.11 MAC protocol, where N is the number of channels. AMCM also outperforms existing multi-channel MAC protocols [24, 23] by 100% and

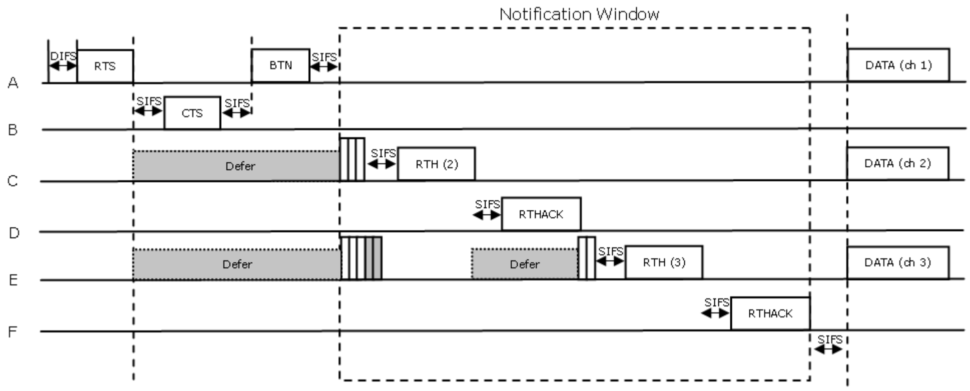


Figure 3.1: Operations of AMCM with 3 competing traffic flows (A→B, C→D, E→F)

150% respectively under high load at a lower hardware cost and complexity. In multi-hop scenarios, AMCM achieves performance improvement of 190% and 90% for both dense and sparse network over IEEE 802.11 MAC respectively. In both scenarios, AMCM achieves close to full utilization of all channels with good protocol efficiency.

3.1 Design

In this section, we first give an overview of the design of AMCM follow by detail descriptions.

First, given a set of N (> 1) orthogonal channels, a *primary or default channel* is chosen and is assumed to be known to all nodes in the network. The rest of the channels (N - 1) are *secondary channels*. Figure 3.1 shows an illustration of a typical operation cycle of the AMCM protocol. At the beginning of the cycle, node A (the sender) acquires the default channel through the use of RTS/CTS exchanges with node B. As a result, knowing that the default channel is not available, nodes C to F will attempt to communicate by acquiring one of the secondary channels. This acquisition is performed in a duration called *N*otification

Window (*NW*) that begins after the reception of the *B*TN message sent by node A (sender of the winner of the primary channel). The duration of *NW* is announced in the RTS, CTS and BTN messages. Notification frames exchange during *NW* include information on the *s*econdary channel to be acquired and the *reservation duration* in which both nodes will spend on the new channel. Nodes that successfully acquired the secondary channels switch to their respective secondary channels only at the end of *NW*. Upon completion, these nodes switch back to the default channel. The cycle then repeats.

One key feature of AMCM is that nodes dynamically negotiate and switch channel in a *distributed and asynchronous* manner. There is no static negotiation period or pre-assigned dedicated channel for negotiation. Instead, the protocol let nodes dynamically synchronize/align themselves locally to a common notification window for secondary channel acquisition. In addition, the duration of *NW* and reservation duration per channel are adapted according to the traffic load and topology.

The three components of AMCM are *acquisition of secondary channel* (Section 3.1.1), *operating in secondary channel* (Section 3.1.2) and *return to primary channel* (Section 3.1.3). The details are presented in the following sections.

3.1.1 Acquisition of Secondary Channels

Behavior on Primary Channel

As shown in Figure 3.1, when a node has packets to transmit, it transmits RTS to its intended receiver, which then responds with CTS. In addition, upon receiving the CTS, the sender transmits a new frame called *Begin-To-Notify* (BTN).

This three-way handshake achieves the follow purposes: (1) reserves the primary channel for data transmission; (2) alleviates potential hidden-terminal problem, which is similar to IEEE 802.11 DCF and (3) announces to the neighboring nodes the upcoming *NW* and its duration.

Item (3) is key to the asynchronous operations of AMCM. Note that nodes overhearing RTS frame should not take it as a confirmation that the primary channel has been acquired by the RTS-sender since it is possible that the receiver might not respond with a CTS for some reasons. Instead, overhearing nodes should treat it as a *tentative confirmation* and only update their NAV accordingly. For this reason, only the CTS and BTN frames are taken as a confirmation (for both the neighbors of the CTS-sender and RTS/BTN-sender). Once a confirmation is overheard, nodes must now align themselves to the upcoming *NW* and also the duration advertised.

Nodes overhearing CTS frame will start *NW* after τ_{cts} seconds, while nodes which overhear BTN frame will start *NW* after τ_{btn} seconds, where

$$\tau_{cts} = 2 \times t_{SIFS} + t_{BTN} \quad (3.1)$$

$$\tau_{btn} = t_{SIFS} \quad (3.2)$$

where t_{BTN} is the time taken to transmit a BTN frame using base rate. These values are shown in Table 4.1.

Duration of *NW* is carried in the RTS, CTS and BTN messages to ensure that it is heard by all nodes in the RTS/CTS range. For convenient, we will call both the BTN-sender and CTS-sender *NW-Initiators*. Determining the duration of *NW* is discussed in more detail in Section 3.1.1.

Once neighboring nodes detect a busy primary channel during an attempt to transmit,

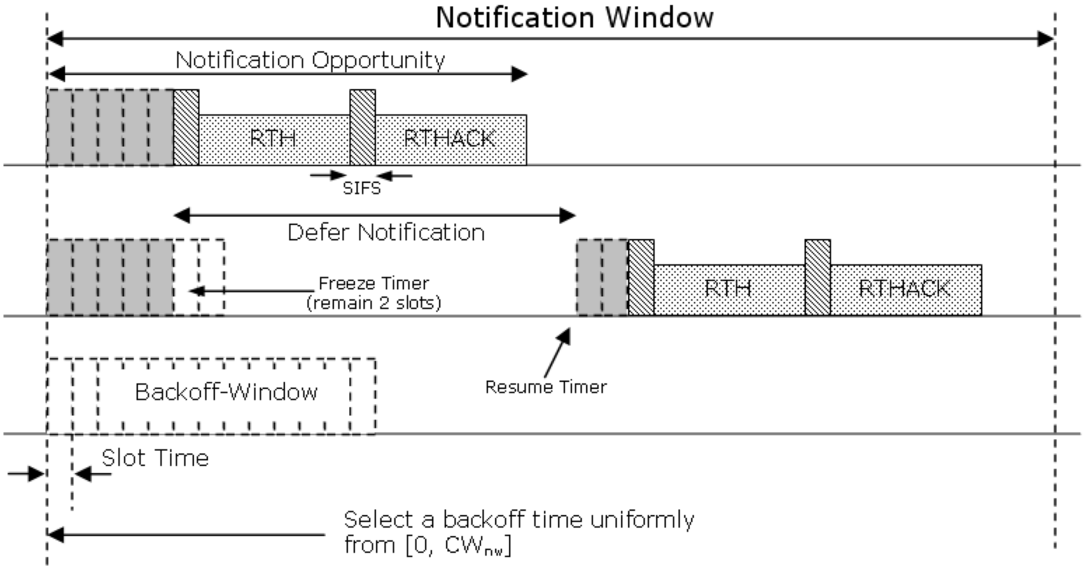


Figure 3.2: Contention-Window inside NW

it freezes its backoff timer and update the NAV accordingly. It also aligns itself to the upcoming *NW*. Cases whereby a CTS response frame is not received after sending RTS frame should not be perceived as a collision or exposed-terminal problem. In the latter case, it could be that the RTS-recipient might be inside *NW* and therefore is unable to respond. In this case, the RTS-recipient (inside *NW*) can inform the RTS-sender about the current *NW*, and also to notify it to switch channel. We termed this as receiver-initiated notification.

In a multi-hop environment, it is also possible (though infrequent) that a node hears multiple NWs. In such cases, a node will accept only the first BTN/CTS message it receives and discard the rest.

Behavior Inside NW

Within a NW, all nodes competing for a secondary channel will now randomly select a backoff time in the interval [0,CW_{NW}]. The backoff time is decremented as long as the channel is sensed idle, stops when the channel is busy and resumes again when the channel is idle again. When the timer reaches zero, the node transmits RTH (*Request-to-Hop*) frame to its receiver, which then responds with RTHACK (*Request-to-Hop-Acknowledge*) frame. Note that each node maintains a separate backoff timer for this purpose. If nodes are unsuccessful in reserving a secondary channel during the current NW, they do not reset their timer, but instead resume it in the next NW. Nodes with no packet to transmit must update their channel usage list accordingly and remain idle throughout NW. Figure 3.2 shows an illustration of the contention and backoff inside a NW. Since all nodes are aware of the duration of NW, nodes will make sure that they do not transmit a RTH frame if the notification exchange (RTH/RTHACK) cannot be completed within the NW.

Both the RTH/RTHACK frames contain the information on the selected channel to switch to and the reservation duration on that new channel. Overhearing nodes will update their free-channel list accordingly to avoid selecting the same channel. These overhearing nodes also include those (new nodes) that are not aware of the current NW. Lastly, all nodes must stay on the primary channel throughout NW. This is to minimize any inconsistency in the channel usage list. The requirement for nodes to stay until the end of NW can pose a problem when the number of channels is large (such as IEEE 802.11a). Fortunately, this problem can be alleviated by adapting NW, which we will verify later through simulation in Section 3.8.

Again, in a multi-hop environment, it is possible (though unlikely) that a node receives RTH/RTHACK messages from different NWs. In such cases, nodes should still update their channel list accordingly. In addition, such conflict can also lead to inconsistency in the status of free channel list among different nodes. However, since such collision are rare and collision on the secondary channel can be detected and resolved easily (see Section 3.1.2), the protocol is sufficiently robust and efficient.

One of the key parameters of AMCM is the value of CW_{NW}. CW_{NW} should be chosen such that it is large enough for a secondary channel to be acquired without collision in a single notification exchange and at the same time without incurring too much overhead. The collision probability depends on the number of flows contending for a secondary channel in a single collision domain. Through simulation and assuming a uniform random distribution, Figure 3.3 shows the probability that a secondary channel is successfully acquired when the number of nodes increases for CW_{NW} values of 15 and 31. While a value of 15 is sufficient for moderate number of nodes, a value of 31 ensures that even with more than 100 nodes contenting, the probability of success is still greater than 99%. Such flow density is sufficiently high for most network density settings and is used as the default value.

Adaptive Notification Window

The size of NW (γ) is decided by the RTS-sender and is broadcasted again by the CTS/BTN-sender. γ determines the number of RTH/RTHACK (notification) exchanges possible within one NW and should be a function of the number of free channels and traffic load.

At low load scenario whereby secondary channels are not required, γ should be small enough to reduce overhead on the default channel. During high load where multiple con-

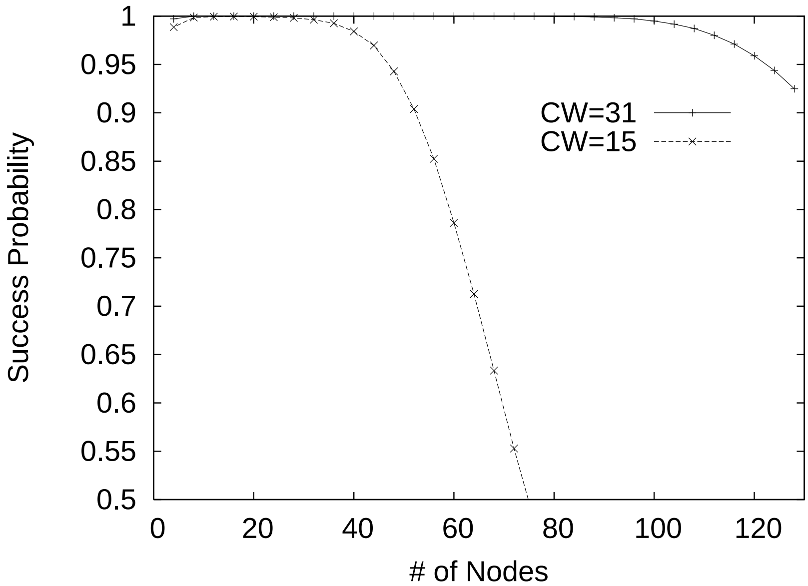


Figure 3.3: Probability of Acquiring Channel

current transmissions are desired, γ should be adjusted accordingly to utilize all available secondary channels. Ideally, it is desired that all available secondary channels be reserved during a single NW.

γ can be expressed in terms of number of *Notification Opportunity* (NOP). Ideally, each opportunity allows a secondary channel to be reserved/utilized. γ is calculated as follow:

$$\gamma = NOP \times t_{NOP} \tag{3.3}$$

$$t_{NOP} = CW_{NW} \times t_{slot} + t_{RTH} + t_{SIFS} + t_{RTHACK} \tag{3.4}$$

where t_{NOP} is the (maximum) size of each opportunity, t_{RTH} and t_{RTHACK} is the transmission time of RTH and RTHACK frame respectively. Assuming that one secondary channel is always acquired in each opportunity, then $NOP \leq N - 1$.

Here, we adopt a simple and effective algorithm to adapt NOP to both the traffic load and channel availability. NOP_{min} is 0 by default. The values of NOP are updated at the end of each NW and before the start of a NW. *NW-initiator* advertises the value of NOP using

RTS, CTS and BTN frames on the primary channel and it is assumed that nodes have equal opportunity to acquire the primary channel.

The idea is to decrement NOP if there are any free secondary channels based on the current channel list information. Availability of free channel likely indicates that traffic load is low or no demand for secondary channels, thus γ or NOP can be reduced. In addition, nodes contending for a secondary channel will also check if they were successful in reserving a secondary channel during NW. Unsuccessful nodes will set their *unsuccessful_notify* flags to *true*.

On the other hand, NOP is incremented if the *NW-Initiator* was unsuccessful during the previous NW. However, in cases whereby there is no free channel available, the value of NOP should remain the same. As the channel reservation duration is included in the RTH/RTHACK messages, nodes can estimate the number of free channels (channel availability) during each NW, and can better determine the proper value of NOP.

To summarize, for a *NW-Initiator*, the value of NOP: (1) is decrement if there is free channel at the end of the previous NW and the *unsuccessful_notify* flag is not set. (Lower bound is NOP_{min}); (2) is increment if the *unsuccessful_notify* flag is set and there are free channels available. (Upper bound is $N-1$); (3) otherwise, it remains the same.

Channel Switching Threshold (CST)

Within a single notification opportunity, nodes indicate the reservation duration, $T_{Reserve}(i)$, they will stay on the new channel i after NW elapses. Clearly, a longer duration on the new channel increases the total throughput by allowing more DATA packets to be transmitted. On the other hand, a shorter duration allows other nodes to contend for the secondary channel sooner and also to communicate with them. However, the overhead increases with smaller threshold, resulting in lower channel utilization.

CST prevents nodes with high sending rate (source-traffic) from occupying he new channel for too long, avoid unreachability* problem and also to minimize any unfairness problem.

Specifically, RTH-sender inspects its packet queue for the first few packets designated for the same destination node (next hop) and sets the duration ($T_{Reserve}(i)$) as the time it takes to transmit these packets. CST specifies the maximum number of packets in which a node is allowed to transmit on the negotiated channel. Assuming packets are of the same size,

$$T_{Reserve}(i) = t_{data/ack} \times min\{N_{queue}, CST\} + t_{rts/cts} \quad (3.5)$$

where N_{queue} is the number of first few packets in the queue designated for the destination node and $t_{data/ack}$ is the duration for a single DATA/ACK transmission. $t_{rts/cts}$ is the time taken for a single RTS/CTS exchange on the secondary channel.

When a node is transmitting to many destinations (such as relays in wireless ad-hoc networks), N_{queue} may be small. One possible approach to increase N_{queue} is to employ per-destination queuing such that even packets that are queued behind other packets can be transmitted. In general, a small CST ensures node reachability and is more fair, whereas a larger CST allows more packets to be transmitted per channel switch and also reduces the switching frequency. Unfortunately, if the threshold is too small, nodes can spend most of their time switching across channels, incurring more overheads and thus not utilizing the channels efficiently. The impact of CST settings will be investigated using simulation in

*nodes operating on the secondary channels will still be reachable within a short time frame

Section 3.2.2.

3.1.2 Operating in Secondary Channel

Once switching over to the secondary channel, nodes perform an initial channel sensing for activity and then a single RTS/CTS exchange to avoid potential hidden-terminal problem. Once the exchange is completed, nodes initiate their data transmission with a series of DATA-ACK exchanges. We term this batch transmission. NAV information is also included in the RTS/CTS/DATA frame indicating the remaining channel reservation duration.

By default, nodes on the secondary channel return back to the default (primary) channel after the reservation duration which they advertised during NW. However, nodes can still return prematurely when a collision is detected. In cases where nodes detect busy channel during the initial channel sensing, both nodes must return. However, since this is probably due to incorrect channel usage list, therefore both nodes can defer their return until they overhear the NAV information contains in subsequent DATA transmission. This information can then be used to update the nodes' current channel list.

In AMCM, each node maintains one data structure, *Neighbor Channel List* (NCL) to update the status of all channels. For each channel i, NCL[i] contains address of the sender, address of the receiver and channel reservation duration.

Every node in the network maintain their own *NCL*. Upon overhearing control frames, nodes refresh their list accordingly to obtain an up-to-date information of all channels. In fact, this list is equivalent to the *NAV* in IEEE 802.11 MAC but is meant for multiple channels. Nodes also use this list to avoid transmission/notification to any node specified in the list.

We adopt a simple channel selection based on the NCL. In AMCM, RTH-senders choose the next available channel based on NCL. When a receiver receives a RTH frame inside NW, it consults the NCL and checks if the proposed channel is available. If the proposed channel has been reserved, the receiver responds with a *negative* RTHACK (channel ID is 0).

3.1.3 Return to Primary Channel

Since returning nodes (and in fact new nodes) may not be aware of the state of the primary channel and secondary channels, they must remain silence (to avoid collision) until they either overhear an upcoming NW initiation (i.e. RTS/CTS/BTN), current NW (RTH/RTHACK) or at least the duration of a maximum transfer unit (MTU).

When nodes return prior to an upcoming NW, nodes can overhear and decode the NAV information inside RTS/CTS/BTN frames and therefore defer themselves and align to the upcoming NW. Otherwise, if they return in the midst of NW, nodes remain silence until they can correctly deduce the state of all channels. When nodes overhear either RTH/RTHACK frames, they should align to the current NW. Node can then contend inside NW to reserve any free secondary channel.

Note that it is possible that returning nodes miss an earlier reservation and therefore try to reserve the same channel due to inconsistent NCL. However, when these nodes switch over to the new channel, they will eventually sense a busy channel due to ongoing data transmissions. In this case, nodes will update their channel list accordingly before switching back to the default channel.

3.2 Simulation Evaluation

In this section, we evaluate AMCM through simulation against both single-channel (original IEEE 802.11 MAC) and multi-channel (MMAC[33], DCA[36]) MAC protocols.

For evaluation purpose, we use aggregate throughput, average packet delay, control overhead and fairness index as our performance metrics. For ease of comparison, we defined throughput ratio as the ratio of the throughput achieved by AMCM over the throughput achieved by IEEE 802.11MAC. Similarly, delay ratio is defined as the ratio of the average packet delay achieved by IEEE 802.11 MAC over AMCM. Control overhead is defined as the total number of control packets per DATA packet delivered. This metric measures the protocol efficiency of AMCM (under varying load) since it relies on additional messages exchange for both channel negotiation and collision avoidance.

Ideally, given N channels, the optimum aggregate throughput of a multi-channel MAC protocol should be $N \times \alpha$, where α is the per-channel saturated throughput. Unfortunately, this idealistic improvement cannot be easily realized due to the control overheads incurred in some multi-channel MAC protocol ([33, 36, 34]) and the control-channel saturation problem [36]. As we will demonstrate later, besides achieving significant performance improvement in high load, AMCM achieves almost ideal linear performance scale up with the number of channels and works well in low load by reducing the control overhead adaptively.

3.2.1 Simulation Model

We simulate AMCM using Glomosim in both single-hop (Section 3.2.2) and multi-hop (Section 3.2.3) wireless networks. In each simulation, all nodes are configured with the

Transmission rate	2 Mbps
Transmission range	250m
Slot time	20μsec
SIFS	10μsec
Synchronization time	192μsec
RTS/CTS/BTN/ACK	20 bytes (272μsec)
RTH/RTHACK	32 bytes (320μsec)
Size of one NOP	960μsec

Table 3.1: Simulation Parameters

same MAC protocol, operating at a raw data rate of 2 Mbps and a transmission range of 250m. For multi-channel MAC protocols, we assumed 3 orthogonal channels. The channel switching overhead is negligible and is ignored in this study. All nodes running AMCM are equipped with a single half-duplex transceiver. For traffic type, each source node generates and transmits constant-bit rate (CBR) traffic sending 1000 packets per second (pps). By default, γ, CW_{NW} and CST are configured to be 5 notification opportunities, 31 time-slots and 100 packets respectively. Each simulation run lasts for a duration of 300 seconds, and all results are averaged over 10 independent runs. The parameters used in this section are summarized in Table 3.1.

3.2.2 Single-Hop

For single hop scenario, nodes are randomly placed within a square area and are within wireless coverage of each other. Therefore, every source node can reach its destination in a

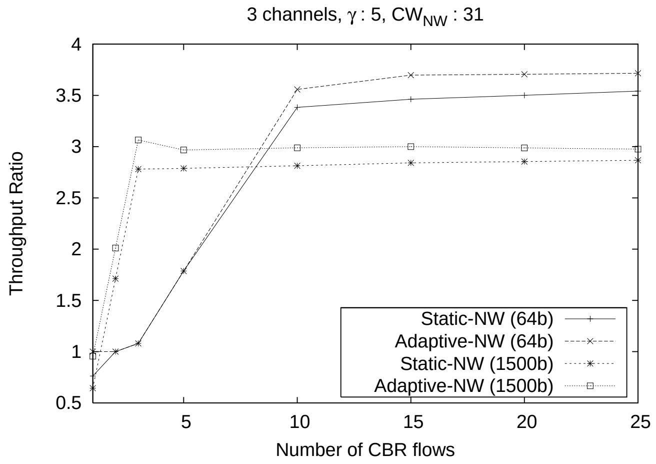


Figure 3.4: WLAN: Impact of number of traffic flows

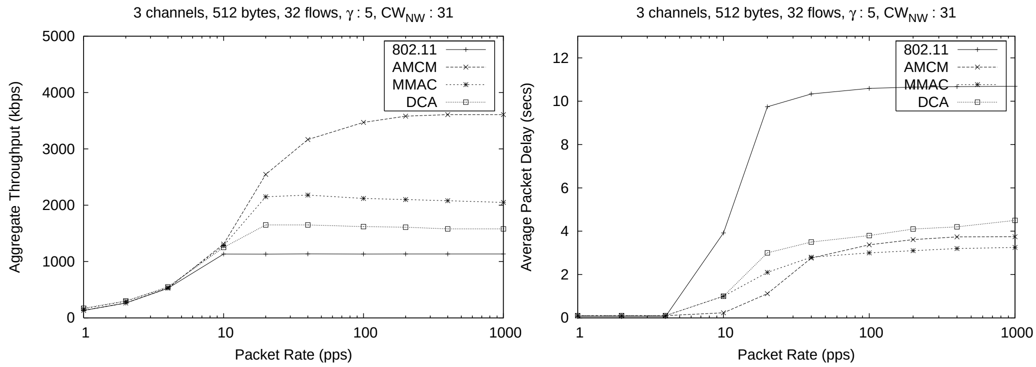
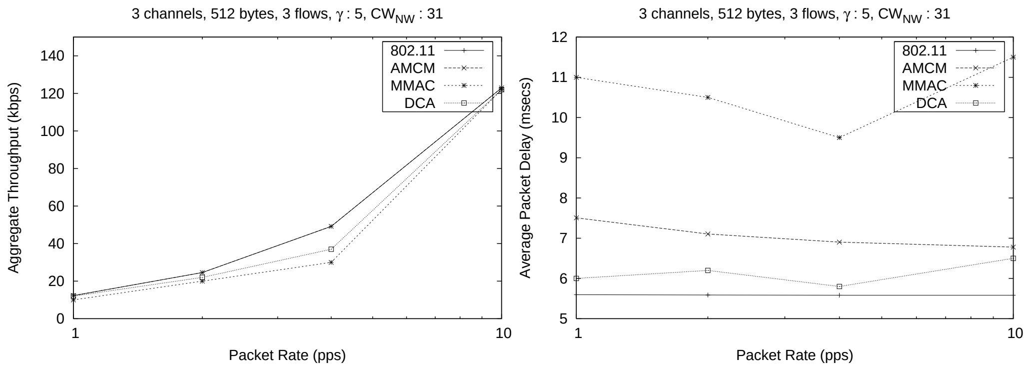


Figure 3.5: WLAN: Impact of traffic load on aggregate throughput and delay

single-hop. We randomly select half of the nodes to be sources, while the other half to be destinations. We do not consider cases where a node sends to multiple destinations. The impact of the following parameters are studied: (i) traffic load, (ii) number of channels and (iii) CST.

Impact of Number of flows

In this section, we keep the traffic rate per flow constant (at 1000pps) and study the capacity of the wireless network as the number of flows increases. Since nodes are within each



(a) Aggregate throughput (b) Average packet delay

Figure 3.6: WLAN: Performance impact under low load

other's transmission and interference range, therefore only one communication flow can exist at any given time for the single-channel (IEEE 802.11) case.

From Figure 3.4, we observed that with only a single flow and packet size of 64 bytes, static AMCM (without NW adaptation) performs much worse (throughput ratio of 0.75) compared to IEEE 802.11 MAC due to the overheads of the BTN frames and large (static) NW. The throughput obtained in this case were 120kbps, 85kbps and 110kbps for IEEE 802.11, static AMCM and AMCM[†] respectively. Hence, the use of adaptive NW mechanism is crucial in reducing the performance gap at low load as the overhead is reduced to only BTN frames exchange. The throughput ratios of AMCM (with NW adaptation) over IEEE 802.11 are 1.0 for both 64 and 1500 byte packets.

As the number of traffic flows increases, the performance gain achieved by AMCM becomes more significant. For IEEE 802.11 MAC, the saturated throughput (one-hop capacity) of the network is only 320kbps (1.6Mbps) for 64 (1500) bytes packet, while AMCM delivers aggregate throughput of up to 1.1Mbps (4.8Mbps) for packet size of 64 (1500) bytes. Compared to IEEE 802.11 MAC, AMCM achieved significant throughput gain of

[†]AMCM always means with NW adaptation

3.5 and 3 for 64 and 1500 bytes packet respectively.

Interestingly, we observed that the achievable throughput gain with smaller packet is higher than larger packet and *the improvement factor can exceed the number of channels available*. This is possible since with smaller packet size, the occurrence of NW on the primary (control) channel increases, thus allowing more batch transmissions (of several small packets) on several secondary channels.

The result shows that for a single flow, AMCM with the use of NW adaptation can be almost as efficient as IEEE 802.11 MAC. In addition, as the number of flows increases, the network operates in the saturated region and the aggregate throughput approaches $N \times \alpha$, where N and α are the number of orthogonal channels and per-channel saturated throughput respectively.

Impact of Traffic Load

In this section, we vary the network load with different packet sending rate while keeping the number of flows constant (32) and compare AMCM with single- and multi-channel (DCA [36] and MMAC [33]) MAC protocols.

From Figure 3.5, AMCM achieved similar performance as the rest when the network load is low. As the packet rate increases, AMCM outperforms both DCA and MMAC, which saturates at around 1.5Mbps and 2Mbps respectively using packet size of 512 bytes. Saturation throughput for IEEE 802.11 occurs at 1.2Mbps. As reported in [33], DCA begins to suffer from control-channel saturation problem with high traffic load. In contrast, AMCM does not rely on a dedicated control channel, but instead adapts NW to increase the channel utilization. At 1000pps, AMCM achieves an aggregate throughput of 3.8Mbps. Unlike

DCA and MMAC, AMCM achieves better utilization of all the three channels, and also achieved almost 3× throughput improvement over its single-channel counterpart.

Figure 3.5 also illustrates the effectiveness of AMCM in terms of packet delay. AMCM incurs much a smaller end-to-end packet delay than IEEE 802.11 MAC, and is also lower than both DCA and MMAC protocols below 40pps where it is already supporting a much higher throughput.

Figure 3.6 illustrates the performance under low load using only 3 CBR traffic flows. We observed that at very low load, AMCM delivers identical throughput performance as IEEE 802.11 MAC but slightly higher average packet delay due to BTN message overhead ($\gamma = 0$). Nevertheless, the average delay is much lower than MMAC and slightly larger than DCA. Note that DCA requires two transceivers.

Figure 3.7 shows the control overhead ratio for varying traffic load. AMCM on average requires 4 control packets for a DATA packet delivery during low load. At higher load, AMCM incurs only 2 control packets for each DATA packet delivered. This is due to lower collisions rate and more useful DATA packets delivered while on the secondary channel (i.e. batch transmission). For single-channel IEEE 802.11 MAC, 3 control packets are required and it increases slightly with more flows due to collisions. Interestingly, AMCM incur slightly lower overheads with increasing flows. This is because nodes are aware of their neighboring channel availability information and therefore will not compete/negotiate for channel during NW.

The results demonstrate the ability of AMCM to mimic single-channel IEEE 802.11 at low load, while able to almost fully utilize all available channels during high load with good protocol efficiency.

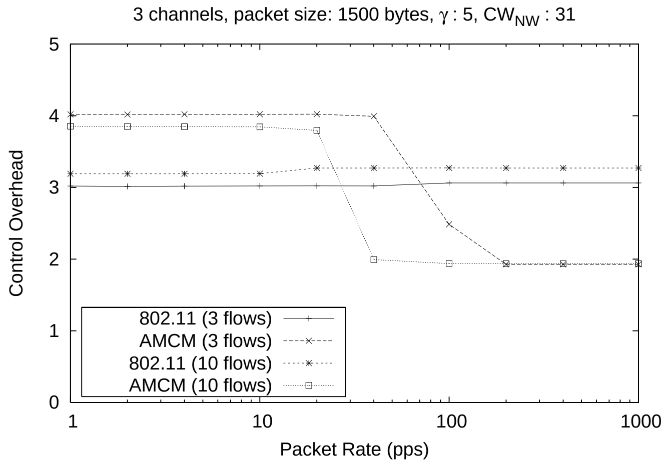
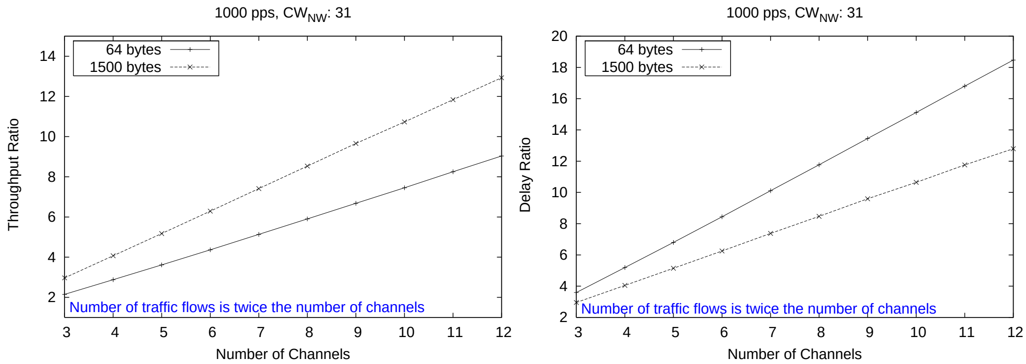


Figure 3.7: WLAN: Comparsion of control overhead against IEEE802.11



(a) Throughput (b) Delay

Figure 3.8: WLAN: Impact of number of channels

Impact of number of channels

In this experiment, we study the performance impact with different number of channels (N). γ is initialized to N notification opportunities (NOP). In order to utilize all channels, *the number of traffic flows is set to twice the number of channels simulated.*

Our simulation result from Figure 3.8a shows that as the number of channels increases, AMCM is able to adapt γ dynamically to both the traffic load and the availability of sec-

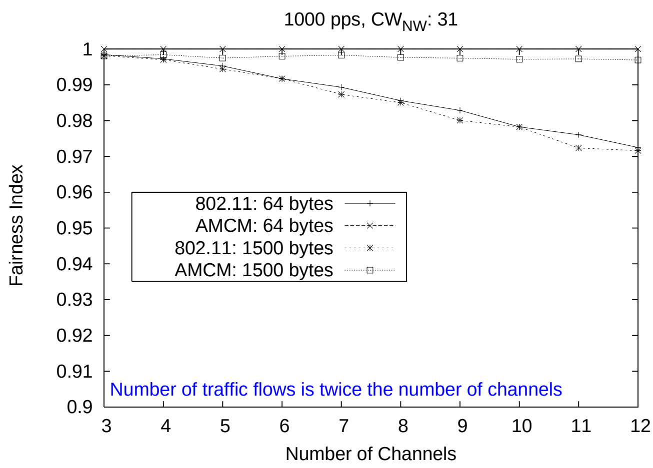


Figure 3.9: WLAN: Impact of number of channels on fairness

ondary channels. The throughput gain over IEEE 802.11 MAC increases with the number of channels (and flows) for both small and large packet size. With 12 channels (24 flows), AMCM achieves almost a throughput ratio of 13 and 9 for small and large packet size respectively. More importantly, this observation also demonstrates that AMCM does not suffer from the control-window saturation problem as the number of channels increased. For up to 12 channels, AMCM delivers a performance gain that scales linearly with the number of channels.

In terms of packet delay, Figure 3.8b shows that AMCM also achieves significant reduction when compared to IEEE 802.11 MAC. With 3 channels, the reduction is 3.8 and 3 for 64 bytes and 1500 byte packets respectively. This reduction also increases with the number of channels. With 12 channels, the reduction is 18 and 13 for 1500 byte and 64 byte packets respectively.

In order to explore how well the channel capacity is shared among all the flows, the

Jain's Fairness Index [38] is used. This fairness index is defined as:

$$Fairness\ Index, f(x) = \frac{(\sum_i^m x_i)^2}{m \sum_{i=1}^m x_i^2} \tag{3.6}$$

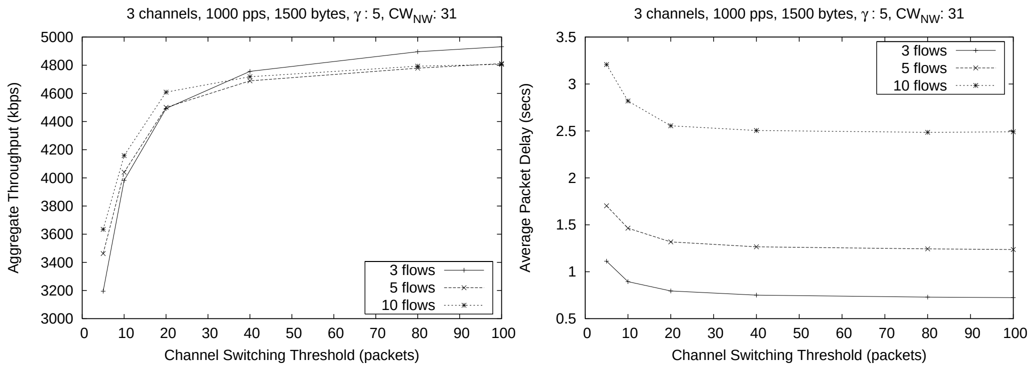
where m is the number of flows, and x_i is the throughput for flow i. Ideally, perfect fairness of $f(x) = 1$ is desired. In the worst case, $f(x) = \frac{1}{M}$, where M is the number of contending nodes.

Figure 3.9 shows the fairness index of AMCM and IEEE 802.11 single-channel MAC. The results show that as the number of channels increase, AMCM (being multi-channel) is able to distribute the capacity evenly among all traffic flows. For large packet size, the fairness index is lower compared to small packet size. This is due to the increase in reservation duration on the secondary channel. As the duration increases, nodes on the primary channel must wait longer in order to acquire the channel once it is free. Overall, AMCM still outperforms IEEE 802.11 MAC as the number of traffic flows increases for the two packet sizes shown.

From these observations, we conclude that AMCM can adapt very well to different number of channels and provide high spatial reuse factor with an increase of number of channels. We also noted that the packet size determines the degree of tradeoff between high throughput, lower delay and fairness.

Impact of Channel Switching Threshold

Recall that switching nodes calculate and advertise their reservation duration on the selected secondary channel during NW. This duration is calculated as described in Section 3.1.1.



(a) Aggregate throughput (b) Delay

Figure 3.10: WLAN: Impact of CST

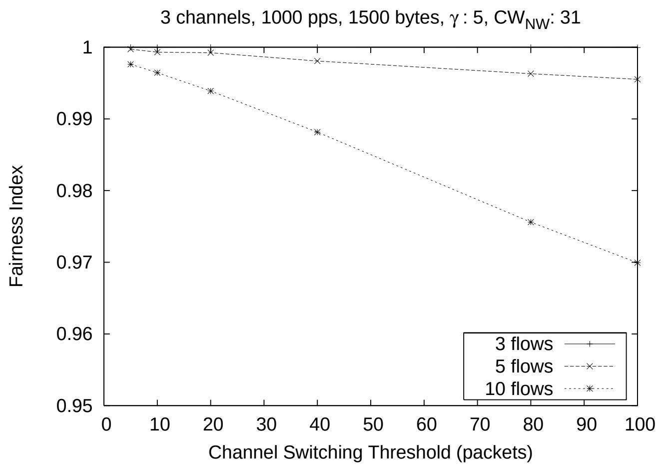
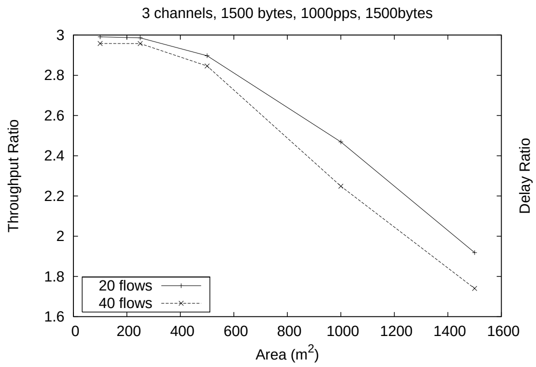
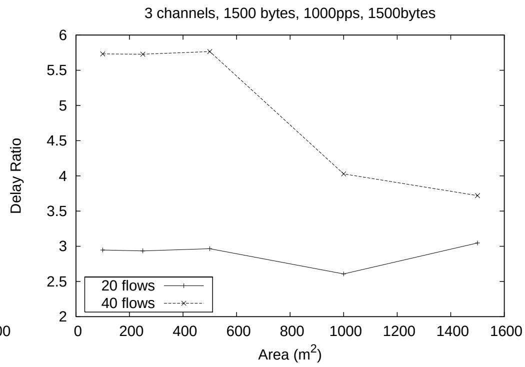


Figure 3.11: WLAN: Impact of CST on fairness

All simulations described above consists of traffic patterns which are disjoint (i.e unique source/destination pairs). Ideally, once a sender succeeds in capturing a secondary channel during NW, it would preferably want to take advantage of this opportunity to *flush* its packet queue for the intended receiver so as to increase the overall throughput and channel utilization. The value of CST determines the maximum period two nodes can reserve on the secondary channel. In this experiment, we study the effect of CST on both the throughput and delay for disjoint traffic pattern.



(a) Throughput



(b) Delay

Figure 3.12: Multi-hop: Effects of Network Density

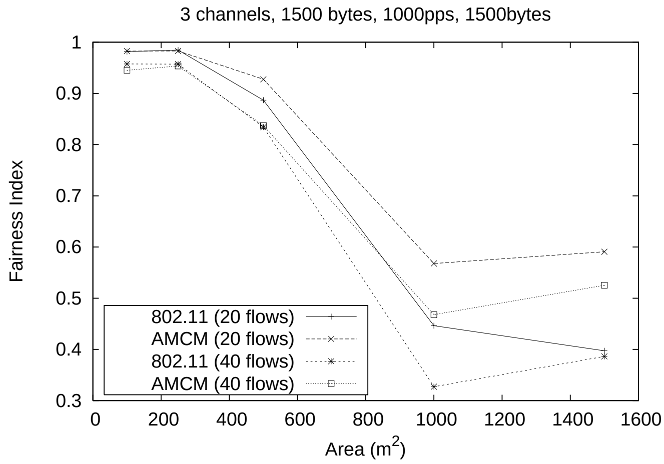


Figure 3.13: Multi-hop: Effects of Network Density

From Figure 3.10a, as expected, the aggregate throughput increases with the threshold value since more DATA packets can be transmitted in a single switch to the secondary channel. We observed that with smaller threshold, the aggregate throughput is higher with more flows since the overhead of *NW* is amortized by having more transmissions on the secondary channels. In general, the throughput performance increases rapidly with smaller threshold (≤ 20). As the threshold increases beyond 40 packets, the aggregate throughput for both 5 & 10 flows decreases since 3 channels are used for this simulation. For packet delay, as the threshold increases, average packet delay also decreases since the aggregate throughput has increased by a sufficiently large amount. An increase of CST beyond certain threshold does not increase the performance gain further and in fact can increase the average delay slightly if the load is increased further.

Figure 3.10c shows the fairness property of AMCM. As the threshold increases, the fairness index decreases. In addition, the degree of unfairness becomes more severe as the network load increases.

The results in this section shows that there is a minimum threshold in which significant performance is achieved. Operating below this threshold results in reduced channel utilization due to increased NW-related overheads. Once CST increases beyond 40-50 packets, the rate of improvement decreases and is bounded by the number of channels (3 in this case). However, larger CST can also result in more unfairness.

3.2.3 Single-hop Communications in Multi-hop Network

In this evaluation, nodes are randomly placed in an area. The source-destination pairs are unique and are within communication range of each other. In addition, a node can only be part of one flow. This scenario is unlike the previous single-hop scenario since the level of interference and traffic within a local interference region varies significantly throughout the wireless network.

First, we study the performance of AMCM under different network densities. In this experiment, we randomly placed 100 nodes in a square area. We simulated using 20/40 CBR traffic flows generating 1000 packets (1500 bytes) per second. Throughout the simulation (300 secs), all nodes are configured with three channels.

Figure 3.12 shows the performance of AMCM and IEEE 802.11 MAC. Figure 3.12a shows that AMCM achieved higher spatial reuse in dense network. In fact, scenarios with areas smaller than 200m by 200m are simply the single-hop scenarios. We achieved almost 3× throughput gain over single-channel IEEE 802.11 MAC in such cases. However, as the network becomes more sparse, the gain decreases to ≈1.8 for an area of $1500m^2$. For such sparse networks, the number of contending flows in some locations may be insufficient to exploit all three channels and hence cannot achieve sufficient improvement compared to single-channel.

Figure 3.12b shows the delay performance against IEEE 802.11 MAC. As the network becomes more sparse, the delay ratio decreases. In a $100m^2$ network, AMCM achieves almost a reduction of 3 and 5.5 for 20 and 40 flows respectively. In general, AMCM achieved lower packet delay in both network configurations compared to IEEE 802.11 MAC. This is true since the use of multi-channel MAC also helps to alleviate both contentions and collisions, which can significantly reduce the time spent in performing backoff and retransmissions.

Figure 3.12c shows the Jain's fairness index of AMCM and IEEE 802.11 MAC protocol. In dense network whereby hidden terminal problem is not dominant, AMCM achieves good fairness index similarly to IEEE 802.11 MAC. As the network becomes sparse, AMCM suffers similarly to IEEE 802.11 since we use similar RTS/CTS/BTN mechanism to alleviate hidden-terminal problem. In addition, for sparse network whereby both nodes' location and traffic might not be uniformly distributed, unfairness can occurred in certain locations (*hotspots*) where channel access is higher, as compared to other less-contention locations.

Finally, as an indication of how well AMCM utilizes all available channel resources,

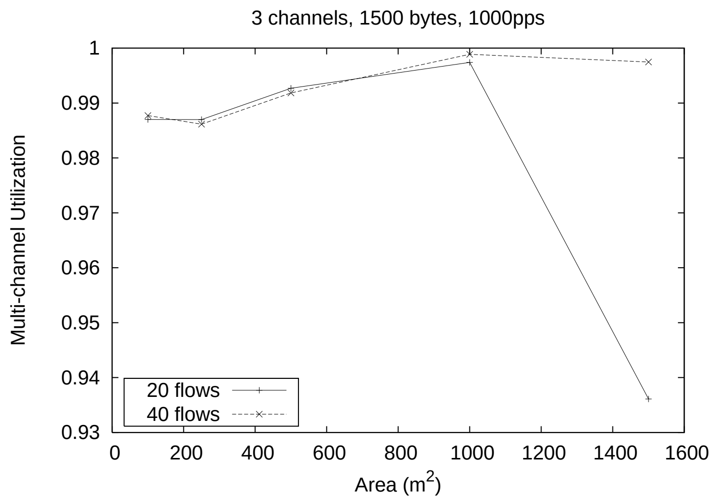


Figure 3.14: Multi-hop: Multi-Channel Utilization

Figure 3.14 shows the channels utilization. When the demands for multiple channel is higher especially in dense network, we observe that AMCM utilizes all available channels efficiently. Load is distributed evenly over all channels as indicated by a fairness index close to 1. Even as the network becomes sparse and there are smaller number of flows, AMCM can still achieve high channel utilizations and spreads the load evenly among the channels.

.

The results in this section demonstrate that AMCM is able to perform well in a multi-hop network scenario and delivers close to the maximum throughput possible with the utilizations for all channels are close to 1 in most cases. However, the amount of improvement over IEEE 802.11 MAC depends on the amount of spatial reuse possible. The main factor affecting the number of channels that can be utilized is the density of active flows. Such density depends on the network density and the number of active flows in the network.

3.3 Summary

This Chapter describes a new multi-channel CSMA/CA-based MAC protocol (AMCM) to enable multiple concurrent transmissions. We performed extensive simulations to study the performance under both infrastructure WLAN (single-hop) and multi-hop wireless networks. We have several observations. Firstly, our traffic-adaptive window-based scheme adapts well to varying traffic load, and thus achieved high channel utilization. Secondly, given a *N-channel* wireless networks, we showed that our single transceiver solution achieved nearly $N\times$ performance gain over single-channel network. Lastly, AMCM has the ability to mimic single-channel IEEE 802.11 at low load, while utilizing all available channels during high load with good protocol efficiency.

Chapter 4

Energy-Efficient Low-Latency Convergecast MAC Protocol

This chapter presents the design of GMAC focusing on three important goals: Energy-efficiency, performance and adaptivity. GMAC adopt several approaches to meet these challenging goals. First, to prolong the operational lifetime of the network, sensor nodes operate in a synchronized duty-cycle sleep/awake schedule. Second, to meet applications' requirements such as low latency, GMAC provides energy-efficient rapid path establishment mechanism to increase the number of packets forward in a single operational cycle. To reduce convergecast latency further, GMAC adopt route-aware TDMA-like approach to provide collision-free transmission. Third, to adapt to variable or asymmetric traffic loads in the network, GMAC adopts a combination of timeout-based reservation and on-demand opportunistic stage approach in an attempt also to increase channel utilization. Initial ns-2 simulation also demonstrated significant performance improvement over RMAC.

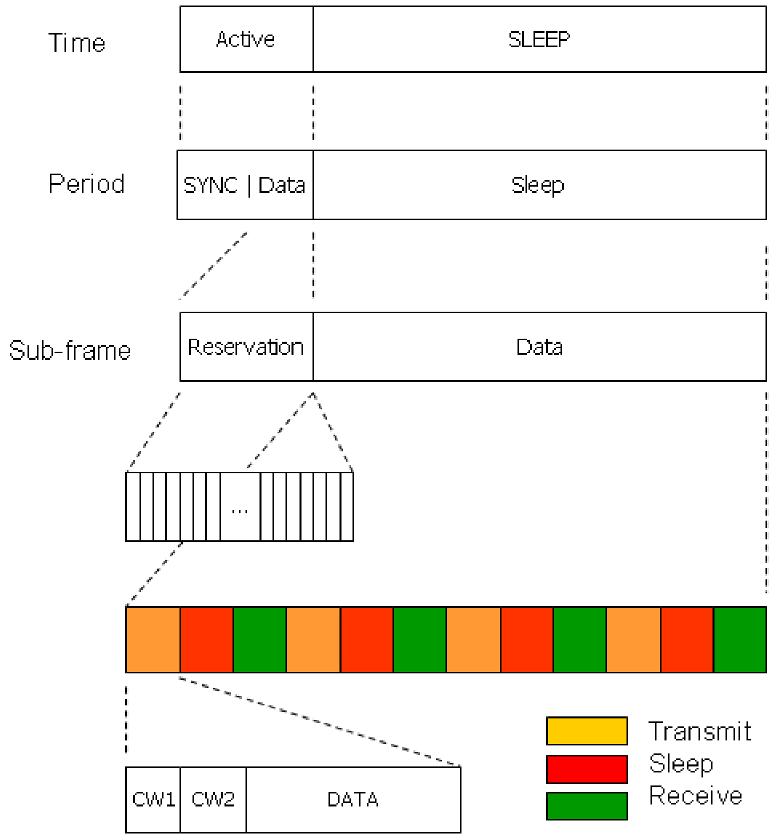


Figure 4.1: GMAC: Frame Structure

4.1 Design

GMAC is a synchronized duty cycle MAC protocol, where each operational cycle is divided into three periods: Sync, Data, and Sleep. We denote the duration of each period by T_{Sync}, T_{Data}, and T_{Sleep}, respectively. Similar to prior work, GMAC assumes that a separate protocol (e.g. [60][61]) is used to synchronize the clocks in sensor nodes during the Sync period with required precision.

Figure 4.1 shows the frame structure of GMAC. In GMAC, time is divided into frames, and each frame is divided into sub-frames. There are two types of sub-frames in GMAC:

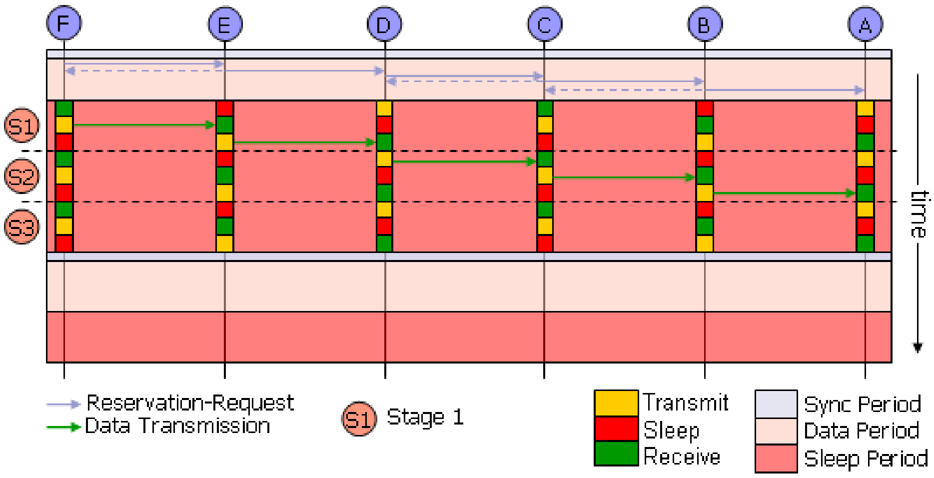


Figure 4.2: GMAC: Overview

reservation sub-frame and *data sub-frame*. The reservation sub-frame operates in the Data period while the data sub-frame operates in the Sleep period in every operational cycle (Figure 4.2). During the reservation sub-frame, all nodes wake up to listen for an incoming reservation request as well as to transmit a reservation request if there are packets to transmit or forward. The goal here is to provide an energy-efficient mechanism for nodes to rapidly establish a pipeline along the multi-hop path towards the sink. The idea is to quickly establish this pipeline and continuously grow/shrink the pipe according to the traffic load along the path. During the data sub-frame, time is divided into stages; each consists of 3 slots - TX, RX, SLEEP. The number of stages is configurable parameters which provides a ”control knob” for designer to achieve tradeoff between performance and cost. To conserve energy, nodes which are not involved in any communication activity sleep throughout the Sleep period (or Data sub-frame). For nodes which have successfully performed the reservation procedure, these nodes only wake up during their respective reserved stage for packet transmission and reception.

4.1.1 Multi-hop Pipeline Establishment

GMAC is designed to meet the low-latency requirement for moderate-to-large sensor network without sacrificing the energy efficiency achieved by the synchronized duty-cycle mechanism. As these moderate-to-large network exhibits longer paths (increased hop-count), it is important to establish a steady-state pipeline with optimal [62] scheduling for packet forwarding towards the sink.

In order to rapidly establish the pipeline (i.e wake up all nodes along the data forwarding path), during the reservation sub-frame, reservation request messages are transmitted along the path towards the sink. Specifically, nodes with packet to transmit, contend during the reservation sub-frame to transmit a reservation request message to its downstream parent. On receiving it, parent node transmits a reservation request message to its parent node. This forwarding of reservation request messages continue until the reservation sub-frame ends.

In GMAC, the reservation request message serves as both request to its parent and acknowledgement from the parent. This is similar to RTS/CTS frames used in IEEE 802.11 MAC protocol. At the start of the reservation sub-frame, a node checks if it has data to send. If it has data to send, it initiates its reservation request at the start of a Data period. It includes address fields such as source and destination (next-hop) address. More importantly, the request message also includes the sender's request stage to reserve.

For example, in Figure 4.3, source node F has data to send to the final destination A; the next-hop node is E. Node F first picks a random period from the contention window and wait for the medium to be quiet for that period and an additional DIFS period before

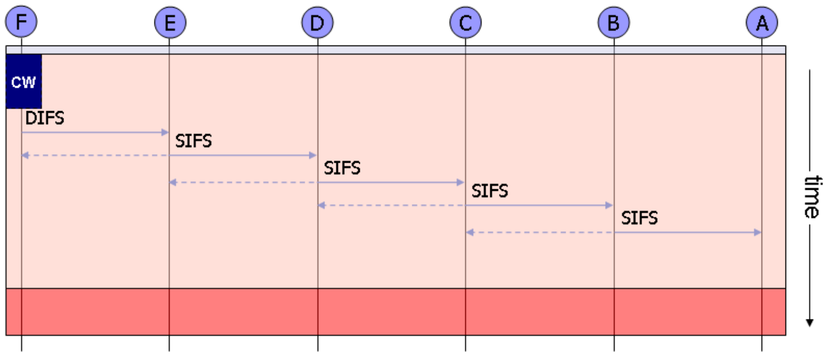


Figure 4.3: GMAC: Multi-hop Pipeline Establishment

sending a reservation request to node E. When node E receives F's request, unless node E is the final destination of this flow, E gets the next-hop address for this destination from its own network layer. Node E then waits a SIFS period before transmitting its own request to its downstream parent. As in IEEE 802.11, SIFS is long enough for a node to switch its transmitting/receiving mode and to do the necessary data processing. When node F overhears the request transmitted by node E, this request message also serves to acknowledge from node E to F. To avoid conflicting stage reservation, all nodes are aware of the status of the stages from its parent. If the parent node cannot reserved the requested stage, it must indicate this failure to the requester. Once node F receives the acknowledgement from node E, it does not transmit the data frame immediately but instead waits for the start of the upcoming data sub-frame. At the start of the data sub-frame, node F only wakes up during the reserved stage for packet transmission and sleeps for the rest of the stages. This approach is unlike most synchronized duty-cycle protocols where by the data transmission proceeds immediately after the channel access. Subsequently, when node D receives the request from node E, node D performs the reservation procedures similar to node E. This process of receiving a request and transmitting request continues until (i) the final destination (sink) has received the request, (ii) the end of the current Data period (reservation sub-frame) is reached or (iii) the parent node cannot reserved the requested stage indicated by its child. In GMAC, contention resolution is improved with Sift distribution [63] for choosing the moment to start sending. Sift distribution is essentially a truncated geometric distribution, which results in fewer collisions than using a uniform distribution. The choice of using Sift also reduces the collisions from hidden-terminals as well as to rapidly wake up downstream nodes along the path to establish the pipeline.

4.1.2 Low-latency & Collision-free Convergecast Scheduling

GMAC is designed to support time- or mission-critical applications which requires good packet reliability/delivery and a bound on covergecast latency. It is well-known that contention-based MAC protocols like CSMA suffers in increased collisions under moderate-high traffic load. These requirements are even challenging in the presence of both asymmetric traffic load and long paths. To tackle this challenge, GMAC adopts TDMA to provide collision-free transmission as well as to reduce the time required to complete convergecast.

In GMAC, only nodes which are involved in data transmission or reception remains awake in their reserved stage during the Sleep period (data sub-frame). Others sleep throughout the data sub-frame to conserve energy. The data sub-frame consists of several stages (S_1,S_2...S_n) as shown in Figure 4.2. Each stage consists of three time slots operating in different states (TX, RX, SLEEP). Both TX and RX time slots contain two contention windows; CW1 & CW2. The purpose of CW1 is to handle possible transmissions from neighboring slidings (with different parent nodes) while CW2 provides opportunity for non-

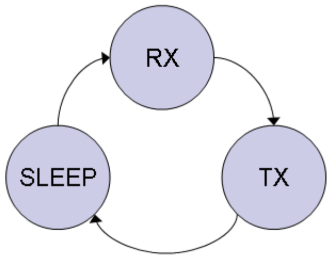


Figure 4.4: GMAC: State Transition

owners to utilize idle reserved stages.

GMAC exploits cross-layer information from the routing layer. Specifically, routing layer provides information regarding the hop-count to reach the destination node (sink). In the data sub-frame, the state of the first time slot of every stage is based on this hop-count information. Simply, if the hop-count is h, the slot is assigned a (i) state TX if h mod 3 is 1, (ii) state SLEEP if h mod 3 is 2 and (iii) state RX if h mod 3 is 0. Once the initial state is determined, node transits into another state following a state transition diagram shown in Figure 4.4.

From Figure 4.2, assuming node F has successfully reserved stage 1 with its parent, node E. In stage 1, node F wakes up to transmit the data frame to node F inside the TX time slot whereby node E wakes up during stage 1 for packet reception in the RX slot. Since node E has also reserved stage 1 with node D, it will forward the data frame in the next slot (which is a TX for node E) to node D. At this time, stage 1 has expired. Therefore, node F and E go to sleep in the remaining Sleep period. Likewise for the rest of the nodes, each node will wake up in their reserved stage, and perform the respective actions depending on the state of the time slot. In this example, node F which is 5 hops away to node A, took 6 time slots to complete the data forwarding.

4.1.3 Adaptivity

The presence of asymmetric traffic load means that nodes will experience the traffic intensity differently. Therefore, an efficient use of the channel is important here. GMAC increased channel utilization by allowing nodes to share idle reserved stages (i.e. owner not using). GMAC adopts two approaches namely *stage stealing*, *timeout-based reservation termination* and *on-demand opportunistic stage*.

Stage Stealing

In GMAC, the node which successfully reserved a stage with its parent during the reservation sub-frame is called the *owner*. Note that owner has the highest priority to transmit in the reserved stage, follow by the non-owners. To allow non-owners to use an idle reserved stage, these non-owners must check for any activity from the owner at the start of the stage (in CW1). If there is no activity detected on the media, non-owners will enter a second contention window (CW2) whereby they contend to transmit to the parent node.

Timeout-based Reservation Termination

Nodes compete during the reservation sub-frame to transmit a reservation request to the parent. Once the sender receives an acknowledgement from the parent, it starts transmitting data packet to the parent in the reserved stage during the Data sub-frame.

In GMAC, nodes do not need to explicitly terminate the reservation or specified the duration of the reservation in the request. For the sender (owner), it maintains a counter to keep track of the number of idle transmission stage. If this counter exceeds a threshold (*idleTxSlotThreshold*), it locally expired and remove the reservation. The counter is reset (to zero) when a transmission occurred in the TX slot. The threshold has no impact on the energy consumption since sender sleeps immediately in its reserved stage (TX slot) if it has no data to transmit. Similarly, on the receiver, it tracks the number of idle RX slots during its reserved stage with the child. If this counter exceeds a threshold (*idleRxSlotThreshold*), it locally expired and remove the reservation. Similarly, the counter is reset when a packet reception occurred in the RX slot. Since the receiver has to be awake during the RX slot of the reserved stage, it consumes a small amount of energy waiting for data packet from the child. Therefore, the threshold value can help to minimize this amount of idle listening on the receiver.

On-demand Opportunistic Stage

In GMAC, sender can only be owner of a single stage. In cases where the sender requires more stages in the current data sub-frame, it can indicate this demand using the *more* flag during the transmission of the data frame. In Figure 4.5, node F requires more stages. After transmitting the data frame in its reserved TX slot (i.e. stage1 - S1), it stays awake during the SLEEP slot (instead of sleeping) in S1 to overhear from the parent on the outcome of the demand. When receiver (node E) receives the data frame, it will be aware of the demand for more stages from the sender. In order to inform the sender of the availability of opportunistic stage in the current data sub-frame, the receiver when transmitting the data frame downstream (to node D), can indicate the next available opportunistic stage (those not reserved) for which it is willing to be awake in the current data sub-frame. In this example,

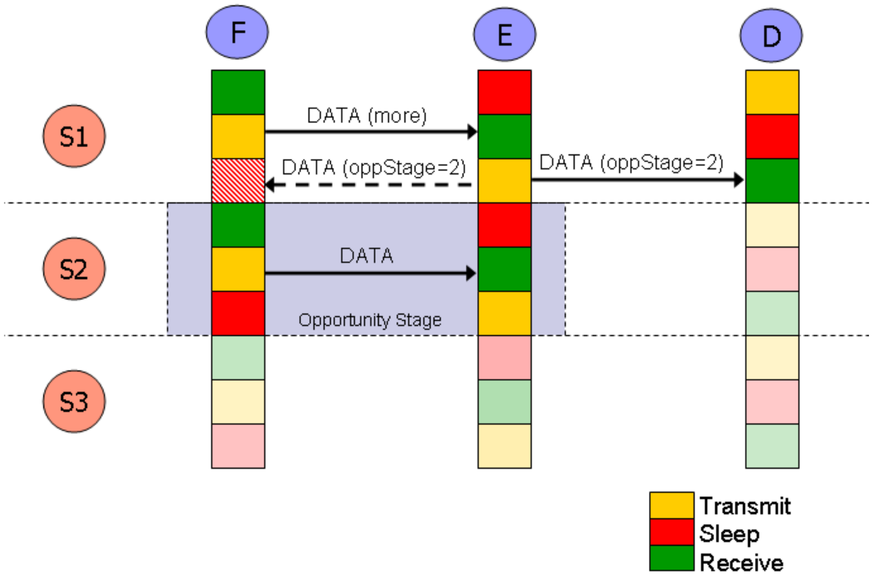


Figure 4.5: GMAC: Piggybacking Opportunistic Stage

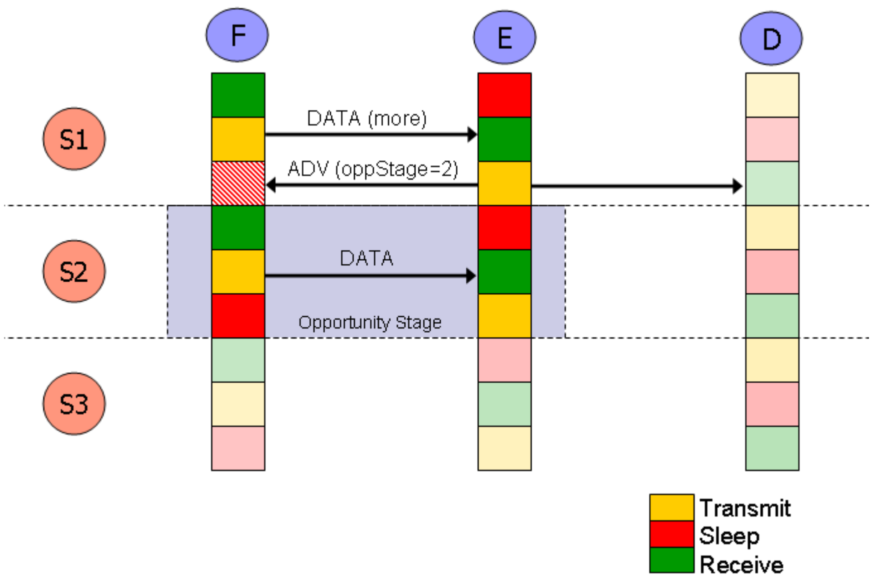


Figure 4.6: GMAC: Broadcast Opportunistic Stage in ADV control message

node E advertises stage 2. To support this, the data frame consists of two additional fields: *more* flag and *OppStage*. In Figure 4.5, node E has a reserved stage with node D, therefore, it can piggyback the *oppStage* information to node F. However, in cases (see Figure 4.6) where node E does not have a reservation with its downstream parent, it broadcasts the *ADV* message which contains the *oppStage* field. The size of *ADV* message is identical to *RREQ* message.

Once the sender overhears its parent's data transmission, it checks the *OppStage* field and learns which stage it needs to be awake to opportunistically in S2 to transmit data frame to the parent. Apart from the reserved stage (S1), the sender will wake up during this opportunistic stage (S2) for data transmission. During the opportunistic stage, the sender wakes up in the TX slot and transmits immediately. This procedure of requesting an opportunistic stage repeats as long as there is a demand. Since opportunistic stages advertised by the receiver are only valid for each data sub-frame, the sender must repeat the procedure of requesting more stages for subsequent frames.

This approach is simple, effective and consumes little control overhead. Only a small amount of overhearing energy is required on the sender, but this energy cost is offset by the ability to improve channel utilization and also achieving low latency performance.

4.2 Simulation Evaluation

To evaluate our GMAC design, we evaluated it using version 2.29 of the ns-2 simulator. We simulate the Two Ray Ground radio propagation model and a single omni-directional antenna at each sensor node and a Two ray Ground radio propagation model in air. For

Bandwidth	20 kbps	Channel Encoding Ratio	2
Sleep Power	0.05 W	RX Power	0.5 W
TX Range	250 m	TX Power	0.5 W
Carrier-Sense Range	550 m	Contention Window (CW)	64 ms
TX Power	0.5 W	DIFS	10 ms
Idle Power	0.45 W	SIFS	5 ms
CW1	8	CW2	16
idleRxSlotThreshold	10	*idleTxSlotThreshold*	5
Reservation slot time (R_{slot})	16.60ms	Reservation sub-frame	$10 \times R_{slot}$

Table 4.1: Networking Parameters

	Frame Size (bytes)	Tx latency (msecs)
Reservation Request (RREQ)	10	11.0
Opportunity Advertisement (ADV)	10	11.0
DATA	50	43.0

Table 4.2: Transmission Duration Parameters

most of our experiments, we compare GMAC against RMAC. For all simulation results, each average value is calculated from the results of 10 random runs.

Table 4.1 shows the key parameters we used in our simulations; these are the default settings in the standard R-MAC simulation module distributed with the ns-2.29 package. According to the ns2 documentation, the default 250m transmission range and the 550m carrier sensing range are modeled after the 914MHz Lucent WaveLAN DSSS radio interface, which is not typical for a sensor node. However, measurements have shown that similar proportions of the carrier sensing range to the transmission range are also observed in

Stages	Cycle time (ms)	Sleep Time (ms)	Listen Time (ms)	Duty Cycle (%)
10	2190.00	1968.00	222.00	10.13
20	4158.00	3936.00	222.00	5.33
30	6126.00	5904.00	222.00	3.62
40	8094.00	7872.00	222.00	2.74

Table 4.3: GMAC Operation Parameters

Figure 4.7: Chain Topology

some state-of-art sensor nodes [5]. In the future, we will investigate the impact of smaller carrier sensing range. In our simulations, traffic loads are generated by constant bit rate (CBR) flows, and all data packets are 50 bytes in size. Intermediate relaying nodes do not aggregate or compress data. We also assume that the application data processing at any node can be finished within a SIFS period; thus, data processing introduces no extra delivery delays. The transmission latencies for different types of packets are shown in Table 4.2.

In our simulations, we assume all the nodes in the network have already been synchronized to use a single wake-up and sleep schedule. There is no synchronization traffic during our simulations, but nodes still wake up at the beginning of the SYNC period and listen to the medium. Also, we do not include any routing traffic in the simulations, as we assume the existence of a routing protocol deployed to provide the shortest path between any two nodes.

4.2.1 Chain Scenario

All nodes are equally spaced in a straight line, and neighboring nodes are placed 200 m apart. One single CBR (constant bit rate) flow sends packets from the node 0 to the node n. The length of the chains varies from 1 hop to 24 hops. The chain scenario helps us to study the protocols for basic multi-hop delivery.

In this subsection, we evaluate the performance of end-to-end delivery latency, energy consumption, throughput and active duty cycle. For GMAC, we simulate for 10, 20, 30 and 40 stages to study the impact of varying stages on the performance. The operation parameters for these simulated stages are shown in Table 4.3.

For traffic load, we initiated a CBR flow generating a traffic load of 100 packets at different packet arrival interval (1,5, 10, 15 and 50 seconds). Unlike RMAC [54] which only evaluated its performance using low traffic load (rate of 1 packet every 50 seconds), we compared GMAC with RMAC in both high and low traffic load conditions. We choose to compare against RMAC due to its simplicity and performance improvement over S-MAC [48] as described in their recent IEEE Infocom 2007 paper [54].

Latency Evaluation

Figure 4.8 shows the average packet delivery latency for chain scenario. The results presented in [54] was obtained for low traffic load (1 packet every 50 seconds). Even at low load, the latency of RMAC is higher than GMAC as the hop count increases. RMAC performs very badly at higher load when the source node is 4 hops away from the sink.

With 10 stages, GMAC delivers good latency performance under both low and high traffic load. At 24 hops, the average latency is $\approx$ 5 seconds. With increased hop count,

	Latency (secs)	T_{cycle} (secs)	$\frac{Latency}{T_{cycle}}$ (cycles)	$\frac{Pathlength * T_{cycle}}{Latency}$ (hops/cycle)
RMAC	17.4	4.465	3.90	6.16
GMAC (10 stages)	5.82	2.19	2.66	9.03
GMAC (20 stages)	10.49	4.16	2.56	9.52

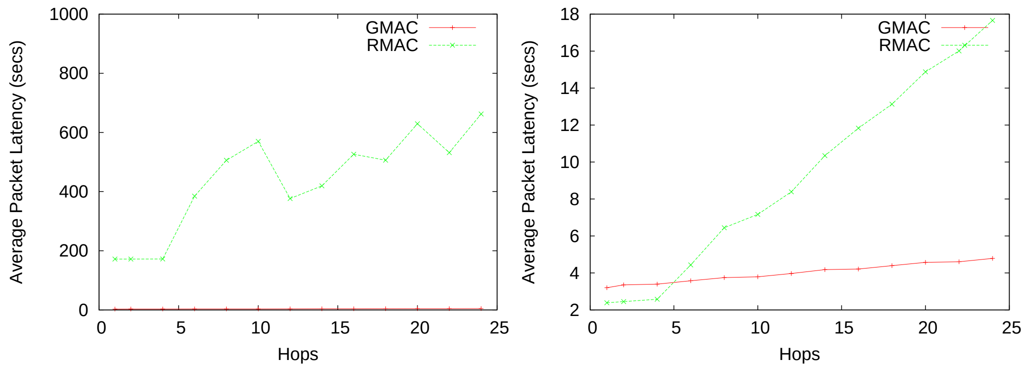
Table 4.4: Number of forwarding per cycle (24 hops, 1 packet every 50 seconds

the rate of increased for the latency is slow. In GMAC, once a pipeline towards the sink is established, nodes can forward packets along pipeline more quickly; thus reducing the forwarding latency. The design of stages (with TX/RX/SLEEP slots) in the data sub-frame which is dependent on the hop count information provides possibility for slot-reuse; thus increased spatial reuse. Not shown in here, we also observed that with 20 stages, the end-to-end packet delivery latency is slightly higher compared to 10 stages. We suspect this increase is due to the possibility of the small reservation sub-frame which was fixed for all simulations. Clearly, for more stages, we need larger reservation sub-frame to increase the number of packet forwardings in an operational cycle.

Table 4.4 shows the number of hops per cycle for a 24-hops network. GMAC can deliver packets on average over 9.03 and 9.52 hops for 10 stages and 20 stages respectively. This demonstrated the capability of GMAC to deliver packets over more hops in an operational cycle. The Table also shows GMAC outperforms RMAC for both 10 and 20 stages.

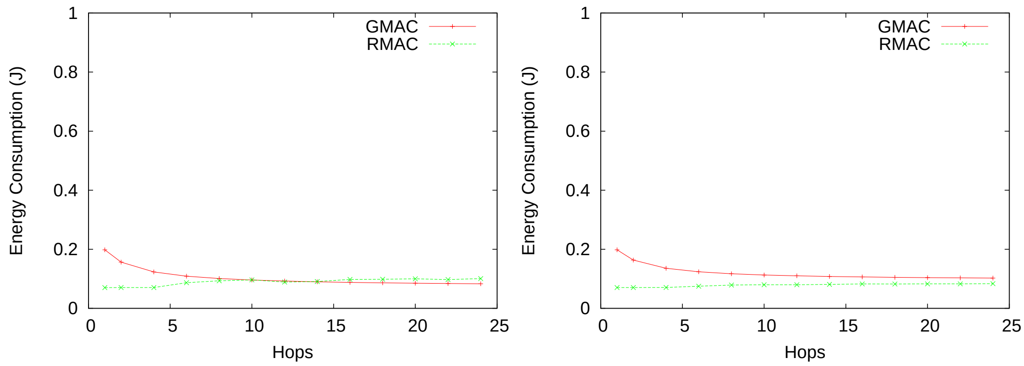
Energy Consumption Evaluation

In Figure 4.9, we observed that for GMAC, the average per-node energy consumption decreases with increased hop count. In GMAC, nodes are only active during their reserved stage; and sleep otherwise. The performance difference (not shown here) between 10 and



(a) 1 packet/sec (b) 1 packet/50 secs

Figure 4.8: Chain Scenario: Multi-hop Forwarding Latency



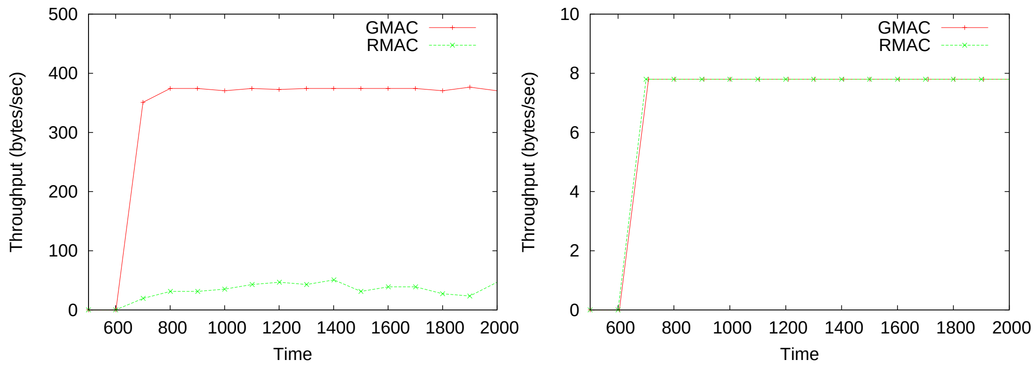
(a) 1 packet/sec (b) 1 packet/50 secs

Figure 4.9: Chain Scenario: Average Per-node Energy Consumption

20 stages however is not significant. From earlier results on latency performance, this suggests that operating with 10 stages is sufficient enough to achieve good performance for both latency and energy consumption. While the difference between GMAC and RMAC is less significant, RMAC tends to consume more energy with longer paths.

Throughput Evaluation

Figure 4.10 shows the throughput of GMAC under varying traffic load over time. Note that data transmission starts after 600 seconds. For packet arrival interval of 1 second, GMAC delivers almost ≈380 bytes/second, while RMAC delivers ≈50 bytes/second. This signif-



(a) 1 packet/sec (b) 1 packet/50 secs

Figure 4.10: Chain Scenario: Throughput

icant improvement is attributed to GMAC's ability to provide collision-free transmission with spatial reuse (slot reuse) as well as its pipeline transmission schedule. At low load, GMAC performs similarly as RMAC, which demonstrates its ability to adapt to varying traffic load, yet with meeting energy-efficiency.

Duty Cycle

Figure 4.11 shows the average per-node duty-cycle throughout the simulation. Not surprisingly, GMAC with different number of stages pushes the duty-cycle from 10% to 3%. In GMAC, we adopted a simple, low-overhead and timeout-based mechanism to expire active reservation. On the receiver, when it detects that there were no packet reception activity in the reserved stage (RX slot) for some threshold, it expires the reservation. Similarly, the sender expires its reservation when it has no transmission activity in the reserved stage (TX slot). In GMAC, the sender's threshold (*idleTxSlotThreshold*) is smaller than the receiver's threshold (*idleRxSlotThreshold*). For all simulations, *idleRxSlotThreshold* is fixed at 10 while *idleTxSlotThreshold* is halved of *idleRxSlotThreshold*. The idea is let the sender expires its reservation first to avoid transmitting later to the receiver if the receiver expires

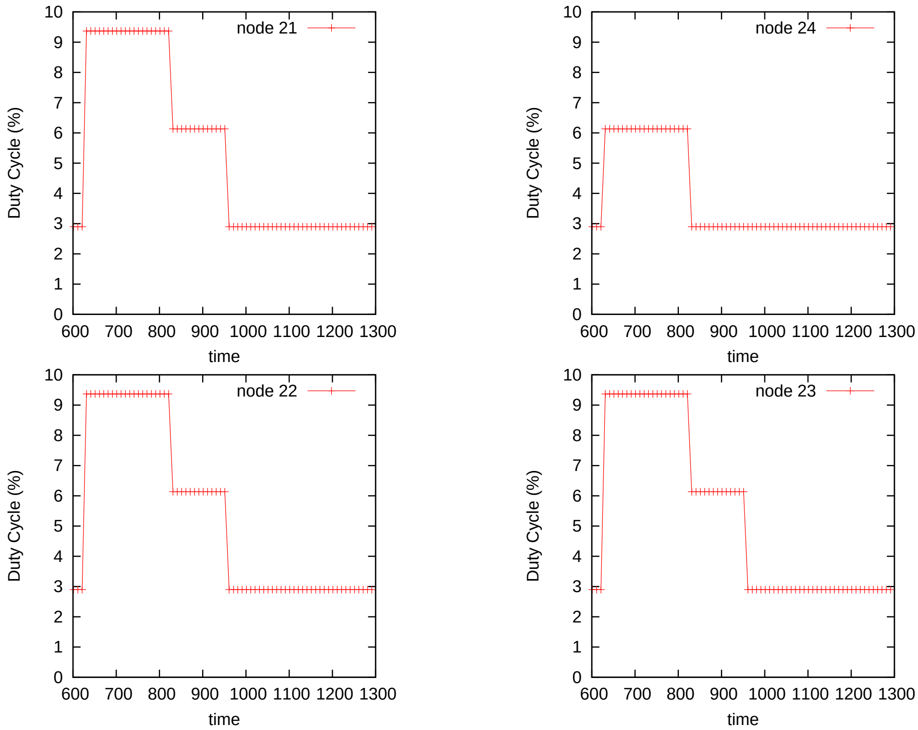


Figure 4.11: Chain Scenario: Traffic-adaptive duty-cycling

first. While the idea is simple and effective (from the Figure), the goal is achieved at the expense on the receiver (incurred idle listening cost in unused reserved stage). However, this idle listening cost is offset by potential non-owners attempting to exploit idle reserved stages.

Effects of Group/Stage Size

Figure 4.12 and 4.13 show the effects of varying number of groups (or numCycles) and stages on the end-to-end packet latency and average energy consumption. At low-load (Figure 4.12), there is little impact on both the packet latency and energy consumption. On the other hand, at high-load (Figure 4.13), packet latency decreases with increase of number of groups in a single operational cycle. In GMAC, nodes with more packets to forward can opportunistically notify that parent node to wake up in subsequent group for more reception. However, packet latency increases with fixed number of groups (10 groups in this simulation) with varying number of stages per group. This is because even nodes can opportunistically request for more subsequent groups from parent, it has to wait for the current group to expire first. This delay is dependent on the number of stages per group, thus as the number of stages per group increases, nodes eventually need to wait longer.

In the current design, each node can only reserve a single stage with its parent node. In order to avoid such long waiting time, one quick solution is to allow nodes to contend for multiple stages in a group. However, one problem will arise when the network is dense and the number of competing nodes increases. Thus, by allowing nodes to own multiple stages, it can potentially block other nodes from transmitting. Therefore, the choice of the number of stages per group needs to take into account the density of the network.

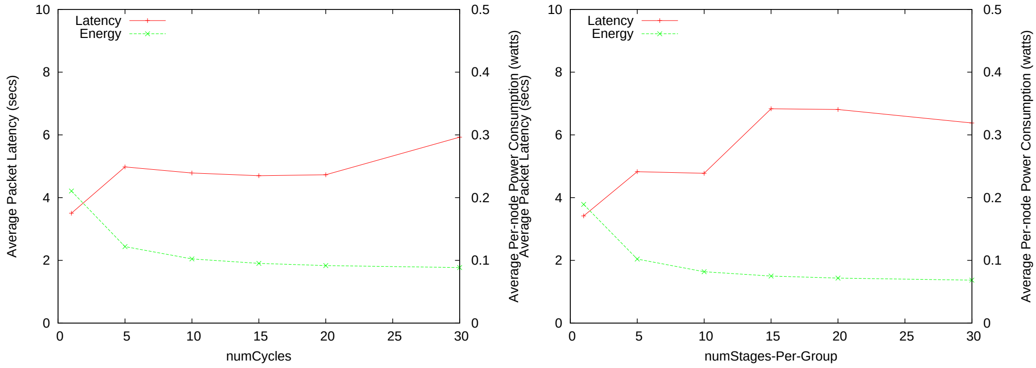


Figure 4.12: Chain Scenario: Effects of varying group/stage size under low-load

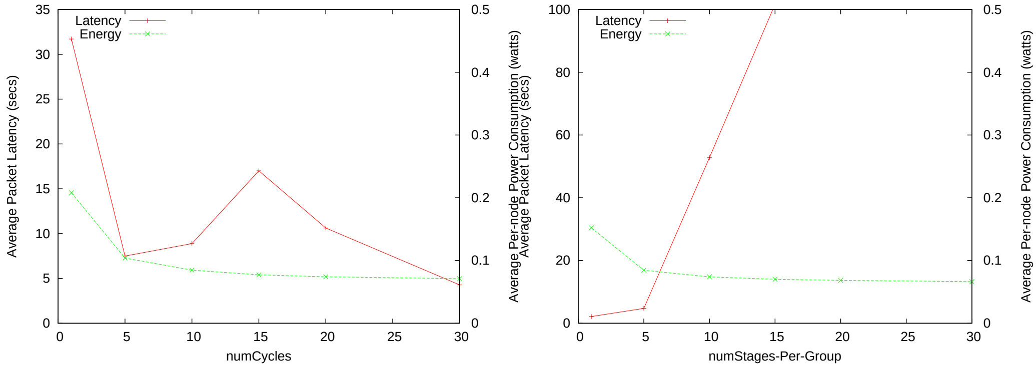


Figure 4.13: Chain Scenario: Effects of varying group/stage size under high-load

4.2.2 Realistic Scenario

Figure 4.14 shows an example of a realistic scenario. The sensor network is composed of 200 sensor nodes and a sink node. The 200 sensor nodes are uniform randomly distributed in a 2000 m by 2000 m square area, and the sink node is located at the top right corner of the square. The maximum path length from a sensor to the sink is 15 hops, and most of the sensor are about 7 to 13 hops away from the sink. All the traffic in the network is from a sensor node to the sink. The traffic load in this scenario is generated as follows: at a periodic interval, a random sensor node is selected to send 100 data packet (to the sink) with packet interval of 10 seconds.

In Figure 4.15a, the average time for a packet to reach the sink node for RMAC and GMAC is $\approx$ 550 seconds and 13.96 seconds respectively. Figure 4.15b shows the packet

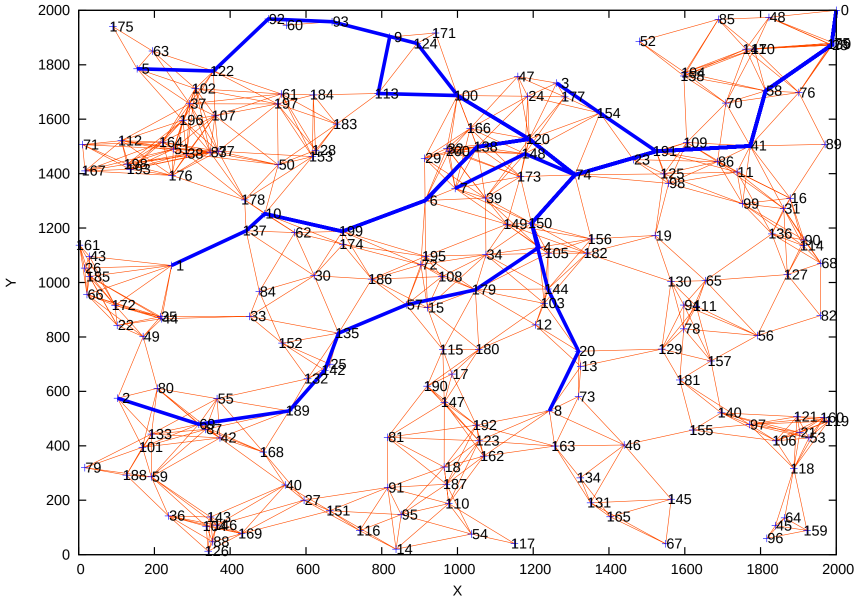


Figure 4.14: GMAC: Realistic 200 node topology

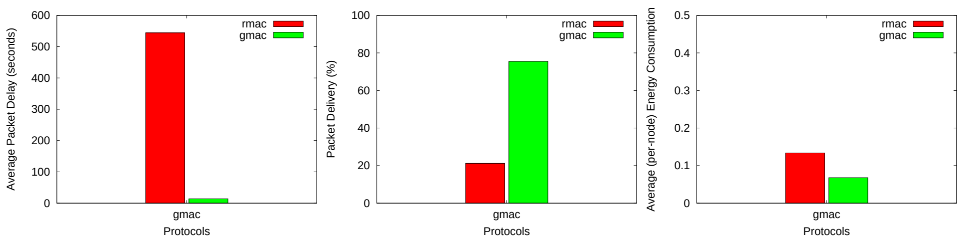


Figure 4.15: Realistic Scenario

delivery performance. GMAC outperforms RMAC in packet delivery. However, GMAC delivered only $\approx$ 80% of the total 1000 packets transmitted. On closer inspection, we observed data packets collision in GMAC due to schedule conflict, especially in *junction* nodes (e.g. node 74). We are currently investigating this problem. Figure 4.15 shows the average per-node energy consumption. Again, GMAC is energy-efficient while delivering good performance in terms of latency and packet delivery as compared to RMAC.

4.3 Conclusion

This Chapter presents the design and simulation evaluation of a synchronized duty-cycle MAC protocol called GMAC. GMAC has three important goals: Energy-efficiency, performance and adaptivity. While operating at low duty cycle rate to conserve energy, GMAC provides rapid path establishment mechanism to wake up multiple nodes along the path towards the sink to establish a steady-state low-latency pipeline. The result is an increased in the number of packet forwardings during an operational cycle; high throughput and low-latency. To reduce energy cost resulting from transmission collisions and also to reduce convergecast latency, GMAC adopted on-demand reservation-based TDMA approach to provide collision-free and fast packet forwarding. Lastly, GMAC adapts to the presence of asymmetric traffic load in the network by allowing non-owners to contend for an idle stage, low-overhead timeout-based termination of reservation and on-demand opportunistic stage during an operational cycle.

Chapter 5

Conclusion & Future Work

The focus of this thesis is on the design of MAC protocols for wireless networks with specific design goals - spatial reuse and energy-efficiency. First, to increase the capacity of the wireless network, Chapter 3 discusses AMCM, a traffic-adaptive multi-channel MAC protocol for wireless networks. AMCM attempts to increase the capacity by enabling multiple concurrent transmissions on orthogonal frequency channels using a single half-duplex transceiver. AMCM provides fine-grain, asynchronous coordination among locally interfering nodes for channel negotiation. Extensive simulation results have shown that AMCM outperforms several multi-channel MAC protocol for both single- and multi-hop scenarios. Second, this thesis proposed GMAC (Chapter 4), to achieve low-latency convergecast and energy-efficiency. To reduce the idle listening and overhearing, GMAC adopts the synchronized duty-cycled MAC protocol. GMAC adopts a TDMA-based approach with route-aware scheduling where all transmissions from the child nodes towards their parent nodes are scheduled based on their hop count towards the sink node. Our simulation further demonstrate that it results in collision-free transmissions with this simple route-aware time-slot scheduling. Initial simulations results have shown that GMAC achieve the design goal and outperform an existing sensor MAC protocol (RMAC [54]) significantly.

While the preliminary simulation results from the above protocols looked promising, there are still work to be done. First, we need evaluate GMAC in a realistic large-scale scenario to study the effect of varying path length, route changes and parameters for the contention resolution on the performance of GMAC. Finally, implementing and evaluating GMAC in a small testbed network of MICA2/MICAz motes running TinyOS will certainly validate its design. Finally, even though, AMCM is designed for ad-hoc networks for high-throughput, the fundamentals of the proposed channel negotiation mechanism can guide the design of future energy-efficient multi-channel protocol for bandwidth-limited WSNs.

Bibliography

[1] D. Estrin, et al., Next century challenges: Scalable Coordination in Sensor Networks, in the Proceedings of the 5th annual ACM/IEEE international conference on Mobile Computing and Networking (MobiCom99), Seattle, WA, August 1999.

[2] I. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, A survey on Sensor Networks, IEEE Communications Magazine, Aug 2002.

[3] S.Cho E. Shih and A. Chandrakasan, "Physical Layer Driven Protocol and Algorithm Design for Energy-Efficient Wireless Sensor Networks", in Proceedings of ACM SIGMOBILE, July 2001

[4] A. Mainwaring, J. Polastre, R. Szewczyk, and D. Culler.Wireless sensor networks for habitat monitoring. In ACMWorkshop on Sensor Networks and Applications, 2002

[5] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. Wireless Sensor Networks: A Survey. Computer Networks, 38(4):393- 422, March 2002.

[6] G. Asada et al., Wireless Integrated Network Sensors: Low Power Systems on a Chip, in Proc. ESSCIRC 98, 1998.

[7] J. Rabaey et al., PicoRadio Supports Ad Hoc-Ultra-Low Power Wireless Networking, in Computer, pp. 4248, July 2000

[8] J. Kahn, R. Katz, and K. Pister, Next Century Challenges: Mobile Networking for Smart Dust, in Proc. ACM MobiCom 99, pp. 271278, August 1999.

[9] E. Shih, S. Cho, N. Ickes, R. Min, A. Sinha, A. Wang, A. Chandrakasan, Physical layer driven protocol and algorithm design for energy-efficient wireless sensor networks, Proceedings of ACM MobiCom01, Rome, Italy, July 2001, pp. 272286.

[10] J. Hill and D. Culler. A wireless embedded sensor architecture for system-level optimization. In UC Berkeley Technical Report, 2002.

[11] XBOW MICA2 Mote Specifications, http://www.xbow.com.

[12] J. Polastre, R. Szewczyk, and D. Culler, Telos: Enabling Ultra-Low Power Wireless Research, in ACM/IEEE IPSN/SPOTS, 2005.

[13] CC2420 2.4 GHz IEEE 802.15.4 / ZigBee-ready RF Transceiver,http://www.chipcon.com.

[14] Chipcon. http://www.chipcon.com

[15] S. Tilak, N. B. Abu-Ghazaleh, and W. Heinzelman. A taxonomy of wireless microsensor network models. SIGMOBILE Mobile Computing and Communications Review, 6(2):2836, 2002. ISSN 1559-1662.

[16] Simulating the Power Consumption of Large-Scale Sensor Network Applications, Victor Shnayder, Mark Hempstead, Bor-rong Chen, Geoff Werner-Allen, and Matt

Welsh. To appear in Proceedings of the Second ACM Conference on Embedded Networked Sensor Systems (SenSys'04), Baltimore, MD, November 2004

[17] J. M. Hellerstein, S. Madden, and W. Hong. The sensor spectrum: technology, trends and requirements. SIGMOD Record, December, 2003.

[18] T. Rappaport, Wireless Communications: Principles and Practice, Prentice Hall, New Jersey, 1996.

[19] Xu, K., Gerla, M., and Bae, S.: Effectiveness of RTS=CTS handshake in IEEE 802.11 based adhoc networks, Ad Hoc Networks J., 2003, 1, (1), pp. 107123

[20] Jinyang Li, Charles Blake, Douglas S. J. De Couto, Hu Imm Lee, and Robert Morris. Capacity of ad hoc wireless networks. In Proceedings of the 7th ACM International Conference on Mobile Computing and Networking, pages 6169, Rome, Italy, July 2001.

[21] W. Ye, J. Heidemann and D. Estrin. Sensor-MAC (S-MAC): Medium Access Control for Wireless Sensor Networks. In Proceedings of the 21st International Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM 2002), New York, NY, USA, June, 2002.

[22] J. Heidemann, F. Silva and D. Estrin, Matching Data Dis-semination Algorithms to Application Requirements, SenSys 2003

[23] S.-L. Wu, C.-Y. Lin, Y.-C. Tseng and J.-P. Sheu, "A New Multi-Channel MAC Protocol with On-Demand Channel Assignment for Multi-Hop Mobile Ad Hoc Networks"

in Int'l Symposium on Parallel Architectures, Algorithms and Networks (I-SPAN), 2000.

[24] J. So and N. H. Vaidya, A Multi-Channel MAC Protocol for Ad Hoc Wireless Networks Proc. ACM Mobihoc04, May, 2004.

[25] A. Adya, P. Bahl, J. Padhye, A. Wolman, and L. Zhou. A multi-radio unification protocol for IEEE 802.11 wireless networks. In BroadNets.

[26] ANSI/IEEE. 802.11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications, 2003.

[27] F. Cal, M. Conti, and E. Gregori, Dynamic Tuning of the IEEE 802.11 Protocol. ACM/IEEE Transactions on Networking, 2000.

[28] Y. Kwon,Y . Fang,an d H. Latchman, A Novel MAC Protocol with Fast Collision Resolution for Wireless LANs. In Proc. of INFOCOM, March-April 2003.

[29] M Heusse, F Rousseau, R Guillier, A Duda, Idle Sense: An Optimal Access Method for High Throughput and Fairness in Rate Diverse Wireless LANs. In Proc. of the ACM SIGCOMM Computer Communication Review, October 2004.

[30] R.R. Choudhury, X. Yang, R. Ramanathan, N.H. Vaidya, Using directional antennas for medium access control in ad hoc networks. In Proc. of the ACM MOBICOM, September 2002.

[31] Y.B. Ko, V. Shankarkumar, N.H. Vaidya, Medium Access Protocols using directional antennas in ad hoc networks. In Proc. of the IEEE INFOCOM, March 2000.

[32] A. Nasipuri, S. Ye, J. You, R. Hiromoto, A MAC protocol for mobile ad hoc networks using directional antennas. In Proc. of the IEEE WCNC, Chicago, IL, September 2000.

[33] J. So and N. H. Vaidya, A Multi-Channel MAC Protocol for Ad Hoc Wireless Networks. In Proc. of ACM Mobihoc, May 2004.

[34] Paramvir Bahl, Ranveer Chandra, and John Dunagan, SSCH: Slotted Seeded Channel Hopping for Capacity Improvement in IEEE 802.11 Ad-Hoc Wireless Networks. In Proc. of ACM Mobicom, 2004

[35] A. Nasipuri, J. Zhuang and S. R. Das, Multichannel CSMA MAC Protocol for Multihop Wireless Networks. In Proc. of IEEE Wireless Communications and Networking Conference (WCNC), September 1999.

[36] S.-L. Wu, C.-Y. Lin, Y.-C. Tseng and J.-P. Sheu, A New Multi-Channel MAC Protocol with On-Demand Channel Assignment for Multi-Hop Mobile Ad Hoc Networks. In Proc. of Int'l Symposium on Parallel Architectures, Algorithms and Networks (I-SPAN), 2000.

[37] A. Adya, P. Bahl, J. Padhye, A. Wolman, and L. Zhou, A multi-radio unification protocol for IEEE 802.11 wireless networks. In Proc. of First BroadNets 2004.

[38] R. Jain, D.M. Chiu, W.R. Hawe, A Quantitative Measure of Fairness and Discrimination for Resource Allocation Shared Computer Systems, Digital Equipment Corporation technical report TR-301, 1984.

[39] G. Alnifie and R. Simon. A multi-channel defense against jamming attacks in wireless sensor networks. In Q2SWinet 07: Proceedings of the 3rd ACM workshop on QoS and security for wireless and mobile networks, pages 95104, New York, NY, USA, 2007. ACM. ISBN 978-1-59593-806-0.

[40] A. Gupta, C. Gui, and P. Mohapatra. Exploiting multi-channel clustering for power efficiency in sensor networks. In Comsware 2006: First International Conference on Communication System Software and Middleware, pages 110, Jan. 2006

[41] C.M. Liang, R. Musaloiu, and A. Terzis. Typhoon: A reliable data dissemination protocol for wireless sensor networks. In EWSN 2008: Proceedings of the 5th European Conference on Wireless Sensor Networks, pages 268285, 2008.

[42] A. Giannoulis, T. Salonidis, and E. Knightly. Congestion control and channel assignment in multi-radio wireless mesh networks. SECON 08: Proceedings of the 5th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks, pages 350358, June 2008.

[43] G. Zhou, C. Huang, T. Yan, T. He, J. Stankovic, and T. Abdelzaher. Mmsn: Multi-frequency media access control for wireless sensor networks. In Infocom 06: Proceedings of the 25th IEEE International Conference on Computer Communications, pages 113, 2006.

[44] Y. Wu, J.A. Stankovic, T. He, and S. Lin. Realistic and efficient multi-channel communications in wireless sensor networks. In Infocom 08: Proceedings of the 27th IEEE International Conference on Computer Communications, pages 11931201, 2008.

[45] Y. Kim, H. Shin, and H. Cha. Y-mac: An energy-efficient multi-channel mac protocol for dense wireless sensor networks. In IPSN 08: Proceedings of the 7th international conference on Information processing in sensor networks, pages 5363, April 2008.

[46] Wei Ye, John Heidemann, and Deborah Estrin, An Energy-Efficient MAC protocol for Wireless Sensor Networks, in IEEE Infocom, 2002

[47] Tijs van Dam, Koen Langendoen, An Adaptive Energy-Efficient MAC Protocol for Wireless Sensor Networks, in ACM Sensys Nov. 2003

[48] W. Ye, J. Heidemann, and D. Estrin, Medium Access Control with Coordinated, Adaptive Sleeping for Wireless Sensor Networks, Technical Report USC ISI-TR-567, January, 2003. (Accepted to appear in ACM/IEEE Transactions on Networking.)

[49] J. Polastre, J. Hill, and D. Culler. Versatile low power media access for wireless sensor networks. In The Second ACM Conference on Embedded Networked Sensor Systems (SenSys), pages 95107, November 2004.

[50] A. El-Hoiydi and J. Decotignie. Low power downlink mac protocols for infrastructure wireless sensor networks. ACM Mobile Networks and Applications, 2005.

[51] G. Lu, B. Krishnamachari, and C. S. Raghavendra. An adaptive energy-efficient and low-latency mac for data gathering in wireless sensor networks. In International Parallel and Distributed Processing Symposium (IPDPS), 2004.

[52] Injong Rhee, AjitWarrier, Mahesh Aia, and Jeongki Min. Z-mac: a hybrid MAC for wireless sensor networks. In Proceedings of the 3rd ACM SenSys Conference, pages 90101, San Diego, CA, USA, November 2005. ACM

[53] G. S. Ahn, E. Miluzzo, A. T. Campbell, S. G. Hong, and F. Cuomo, Funneling-MAC: A Localized Sink-Oriented MAC for Boosting Fidelity in Sensor Networks, in the Proceedings of the 4th ACM Conference on Embedded Networked Sensor Systems

[54] S. Du, A. K. Saha, and D. B. Johnson, RMAC: a Routing-Enhanced Duty-Cycle MAC Protocol for Wireless Sensor Networks, in Proceedings of the 26th IEEE International Conference on Computer Communications (Infocom 07), pp. 1478-1486, May 2007.

[55] Yanjun Sun, Shu Du, Omer Gurewitz, and David B. Johnson. ”DW-MAC: A Low Latency, Energy Efficient Demand-Wakeup MAC Protocol for Wireless Sensor Networks”. In Proceedings of the Ninth ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc 2008), pages 5362, May 2008.

[56] Ye, W., Silva, F., Heidemann, J. ”Ultra-Low Duty Cycle MAC with Scheduled Channel Polling”. In: Proceedings of the 4th ACM Conference on Embedded Networked Sensor Systems (SenSys 2006). Boulder, Colorado, USA (2006)

[57] Curt Schurgers, Vlasios Tsiatsis, Saurabh Ganeriwal, andMani Srivastava. ”Optimizing sensor networks in the energy-latency-density design space”. IEEE Transactions on Mobile Computing, 1(1):7080, January 2002

[58] A. El-Hoiydi. ”Spatial TDMA and CSMA with preamble sampling for low power ad hoc wireless sensor networks”. In Proceedings of the SeventhInternational Symposium on Computers and Communications (ISCC), pages 685692, July 2002.

[59] Michael Buettner, Gary V. Yee, Eric Anderson, and Richard Han. ”X-MAC: A Short Preamble MAC Protocol for Duty-Cycled Wireless Sensor Networks”. In Proceedings

of the 4th International Conference on Embedded Networked Sensor Systems, pages 307320, 2006.

[60] J. Elson, L. Girod, and D. Estrin, ”Fine-grained network time synchronization using reference broadcasts,” in Proceedings of the Fifth Symposium on Operating Systems Design and Implementation (OSDI 2002), Dec. 2002.

[61] S. Ganeriwal, R. Kumar, and M. B. Srivastava, ”Timing-sync protocol for sensor networks,” in Proceedings of the First International Conference on Embedded Networked Sensor Systems (SenSys 2003), Nov. 2003, pp. 138.149.

[62] S. Gandham, Y. Zhang, and Q. Huang, Distributed time-optimal scheduling for convergecast in wireless sensor networks, Comput. Netw., vol. 52, no. 3, pp. 610629, 2008

[63] Jamieson, K., Balakrishnan, H., Tay, Y.C.: Sift: a MAC protocol for event-driven wireless sensor networks. In: Proc. of the 3rd European Workshop on Wireless Sensor Networks (EWSN). (2006) 260275