Novel Medium Access Control (MAC) Protocols for Wireless Sensor and Ad hoc Networks (WSANs) and Vehicular Ad hoc Networks (VANETs)

Md. Kowsar Hossain

A thesis submitted to the Faculty of Graduate Studies
in partial fulfilment of the requirements
for the degree of

Master of Science

Graduate Programme in Computer Science and Engineering
York University
Toronto, Ontario
September, 2014

Abstract

Efficient medium access control (MAC) is a key part of any wireless network communication architecture. MAC protocols are needed for nodes to access the shared wireless medium efficiently. Providing high throughput is one of the primary goals of the MAC protocols designed for wireless networks. MAC protocols for Wireless Sensor and Ad hoc networks (WSANs) must also conserve energy as sensor nodes have limited battery power. On the other hand, MAC protocols for Vehicular Ad hoc networks (VANETs) must also adapt to the highly dynamic nature of the network. As communication link failure is very common in VANETs because of the fast movement of vehicles so quick reservation of packet transmission slots by vehicles is important.

In this thesis we propose two new distributed MAC algorithms. One is for WSANs and the other one is for VANETs. We demonstrate using simulations that our algorithms outperform the state-of-the-art algorithms.

Acknowledgements

I would like to express my heartfelt gratitude to my supervisor Prof. Jeff Edmonds and co-supervisor Prof. Suprakash Datta for their continuous support in the preparation and completion of this thesis. This thesis would not have been possible without their effort. They guided me with tireless patience throughout the thesis. Whenever I had any questions they were always there to answer.

I would like to thank the rest of my thesis committee members: Prof. Mokhtar Aboelaze and Prof. Yongsheng Chen for their time and valuable comments. My sincere thanks also go to my Professors of my courses. I really appreciate their assistance and inspiration. Also, I am grateful to the staff of the department of Computer Science and Engineering for their help.

Last but not the least, I give my wholehearted thanks to every single member of my family for their endless support and encouragement. They have always been a constant source of inspiration in my life.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

A communication network is a collection of nodes which communicate with each other through different types of communication links such as cables, satellites, radio frequency waves, and infrared waves. Based on communication link types, a communication network can be categorized as wired networks and wireless networks. Wireless Networks can be further divided into *infrastructure based* and *ad hoc based* networks. In *infrastructure based* networks, there are no direct communications between wireless nodes. Instead, nodes communicate with each other through access points. These access points control medium access making the topology of the network very simple. Moreover, these access points act as gateways if there is a necessity for nodes in one network to communicate with nodes in other networks. On the other hand, *ad hoc* wireless networks do not need access points. Nodes in this network category communicate with each other directly and maintain connectivity in a decentralized manner. As a result, each node has to implement a medium access control algorithm. Two very important *ad hoc* wireless networks that are studied in this thesis are *Wireless Sensor and Ad hoc Networks (WSANs)*, and *Mobile Ad hoc Networks (MANETs)*.

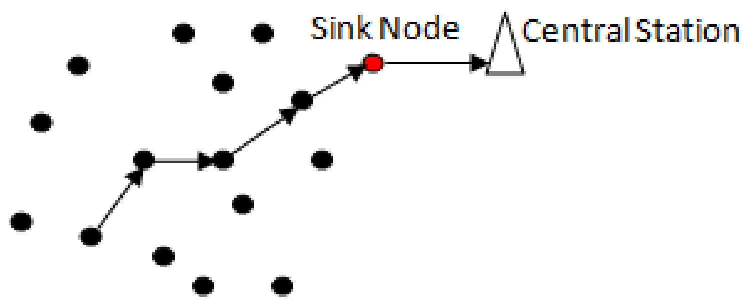


Figure 1.1: Communication of sensor nodes with central station

Wireless Sensor and Ad hoc Networks (WSANs): WSANs are ad hoc networks with a large number of small, inexpensive nodes deployed over geographical areas to monitor physical or environmental conditions. Each of these nodes has a microcontroller, wireless transceiver, an antenna, a power source (normally a battery), memory, and one or more sensors [3]. As shown in figure 1.1 these sensors collect relevant data from the environment and then send or relay that data via the ad hoc network to a central station. The central station can be queried to further process the data. As the nodes are inexpensive and small in size, they have very limited resources (battery power, memory space, and processing speed). The nodes are often placed in environments where maintenance (e.g., replacement of battery or damaged nodes) is difficult. The network lifetime thus depends on the energy efficiency of the sensors. So it is necessary to design algorithms that minimize energy usage.

Mobile Ad hoc Networks (MANETs): MANETs are ad hoc networks comprising mobile nodes. Nodes in MANETs can move in any direction without restrictions, leading to changes in neighbourhood structure, which, in turn, alters the network topology in a frequent manner. This makes the design of network protocols harder. A class of MANETs called *Vehicular Ad hoc Networks (VANETs)* has seen a lot of research activity in the last decade. Nodes in VANETs are vehicles that comply with street traffic regulations while moving. VANETs support both vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I) communications which is shown in figure 1.2. In V2V communications, vehicles exchange information with each other. V2I communications involve message exchanges between vehicles and traffic lights or between vehicles and roadside monitors known as road side units (RSUs). The vehicles can access the internet through RSUs. Each vehicle is equipped with a controller called on-board unit (OBU) that supports the V2V and V2I communications.

The WHO *Global status report on road safety 2013* [6] states that road accidents

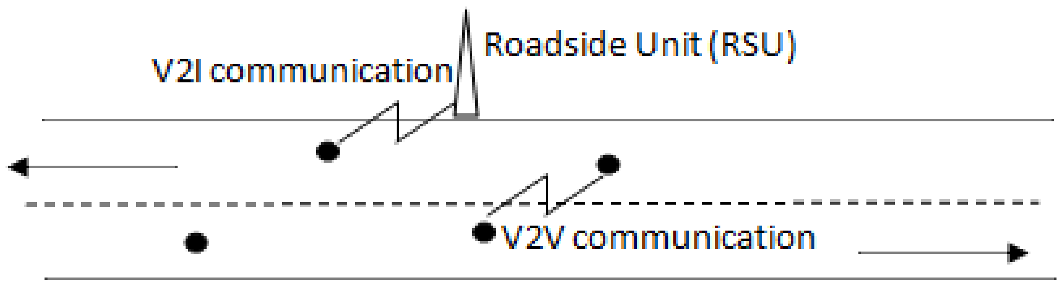


Figure 1.2: V2V and V2I communications

cause around 1.24 million deaths and 20 – 50 million non-fatal injuries each year. Moreover, according to the Texas Transportation Institute [5] the traffic congestion cost in the US was approximately $115 billion based on wasted time and fuel. So it is necessary to have some methods of communication between vehicles so that drivers and passengers can be warned of accidents and traffic congestions that may exist ahead. Also, different types of applications are required to improve driver and passenger comforts. *Intelligent Transportation Systems (ITS)* [5] support V2V and V2I communications by applying information technologies in vehicles and RSUs in order to improve road safety and provide passenger and driver comforts. Therefore, VANETs form an important part of *ITS*. In addition, *ITS* are used to increase the efficiency of transportation and reduce air pollution.

1.1 Motivation for this thesis

In wireless networks, collisions happen when wireless nodes contend for the shared medium. More specifically, when two nodes within each other's radio range transmit packets at the same time, or two packets are sent by nodes to the same destination node at the same time, a collision is said to take place. Collisions waste energy, increase packet delay, and decrease throughput. As pointed out earlier, nodes in WSANs have limited energy, making it necessary to design energy efficient MAC protocols in order for nodes to preserve battery power. This, in turn, prolongs the network lifetime. In MAC protocol design for VANETs however, energy conservation is not as important. One of the primary issues in VANETs is that the network topology changes very rapidly, causing more collisions. So the quick access of the shared medium with less collisions is important in designing MAC protocols for VANETs.

1.2 Objectives

The main objectives of this thesis are

- To design and implement a novel MAC protocol for WSANs that can reduce energy consumption, decrease delay, and increase throughput.

- To design and implement a novel MAC protocol for VANETs that can adapt to fast network topology changes.

1.3 Thesis Organization

The thesis is organized as follows–

- Chapter 2 presents the basic TCP/IP architecture for communication networks; then, the characteristics and applications of WSANs and VANETs are listed, and some common challenges in designing wireless MAC protocols are presented.
- In chapter 3, the major sources of energy waste are described first. Next, some existing wireless MAC protocols and some WSAN MAC protocols related to our work are reviewed. Finally, our proposed MAC protocol Ad-ATMA for WSANs is described and its performance is evaluated.
- Chapter 4 provides an overview of some TDMA-based VANET MAC protocols related to our work. This is followed by a detailed description of our proposed protocol ResVMAC. Finally, the performance of ResVMAC is evaluated and compared with two related protocols.
- Some concluding remarks and future work are presented in Chapter 5.

Chapter 2

Characteristics, Applications, and Challenges in WSANs and VANETs

This chapter starts with the simplified TCP/IP model which is a standard network architecture model. The characteristics and applications of WSANs and VANETs are explained next, followed by a discussion of some challenges in designing MAC protocols for wireless networks. Finally, some metrics used for evaluating MAC protocols are introduced.

2.1 TCP/IP: A Standard Network Architecture

The sender and receiver must complete some complex tasks in order to facilitate communication in a communication network. To complete these tasks smoothly, the total tasks are divided into sub-tasks and depicted as layers in hierarchical architectures, each of which defines a different set of sub-tasks/layers. These include the seven layered Open System Interconnection (OSI) architecture and the widely used five layered TCP/IP architecture which is shown in figure 2.1. As the TCP/IP architecture is the de facto standard today, we describe the TCP/IP model below.

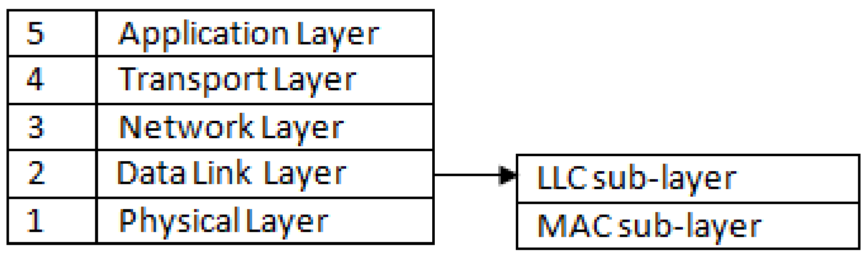


Figure 2.1: Five layered TCP/IP Model

Physical Layer: The physical (bottom) layer implements modulation, transmission, and receiving techniques [10]. This layer has the following responsibilities:

- Converting bit streams into analogue signals.
- Sending signals over the shared medium or receiving analogue signals from the shared medium.
- Converting the received signals to bit streams and sending them to the layer above it.

- Employing the Clear Channel Assessment (CCA) scheme to measure the energy level of the medium and uses that to detect whether the medium is free or not.

Data Link Layer: This is the second lowest layer of TCP/IP model, residing above the physical layer and below the network layer. This layer is logically divided into two sub-layers namely, the Logical Link Control (LLC) and MAC sub-layers. The LLC sub-layer connects the MAC sub-layer and the network layer. The MAC sub-layer defines how the shared medium is accessed among devices. It also uses the error control techniques for reliable communication between network devices. In general, MAC protocols are designed to optimize performance metrics like throughput and delay. However, MAC protocols for WSANs should be energy-efficient and MAC protocols for VANETs should perform in networks of fast moving vehicles.

Network Layer: The functions of this layer are efficient route discovery, addressing, and routing. The primary concern in the design of routing protocols for WSANs is the energy consumption. The sensors have limited battery power and they need to deliver data in such a way so that little energy is consumed. Frequent network topology changes and fragmentation are the main concerns for the routing protocols in VANETs. In general two types of routing are used: proactive routing (LSR, FSR [41]) and reactive routing (AODV [43], DSR [31]). In proactive routing each node maintains a routing table. Frequent control packets are sent in order to keep the table up-to-date that consumes more energy. Furthermore, it may not be possible to discover a path due to the network partition that occurs when the nodes move with high velocities. On the other hand, the reactive routing discovers a path on demand which takes less control packets. This way, however, more time is needed to establish a path. Also, this algorithm may not adapt to the frequently changing network topology caused by mobile nodes.

Transport Layer: The transport layer is responsible for reliable or unreliable end-to-end delivery of data between the sender and the receiver. It also implements flow and congestion control. The Transmission Control Protocol (TCP) is used for reliable data delivery and the User Datagram Protocol (UDP) is used for unreliable delivery. The TCP congestion control algorithm was designed for wired networks, and assumes that packets are lost only due to network congestion resulting in buffer overflow. However, in wireless networks packet loss happens not only due to congestion but also due to attenuation, reflection, refraction, diffraction, and scattering. So the traditional TCP is not suitable and, therefore, many variants or extensions of TCP have been proposed for wireless networks.

Application Layer: The top layer of TCP/IP model is the application layer. The host programs access the network using this layer.

2.2 Characteristics and Applications of WSNs

WSNs have the following characteristics and applications:

2.2.1 Characteristics of WSANs

The main characteristics of WSANs are as follows [9, 55]:

- Limited Resources: Sensor nodes have limited computational power, bandwidth, and memory capacity. They use a short-range transceiver to communicate with other nodes.

- Network lifetime: The battery capacity of the sensor nodes are limited, and nodes fail when their batteries are depleted. So energy efficient protocols are necessary to increase the longevity of the networks.

- Dense, ad hoc deployment: A large number of nodes are deployed densely in the area of interest. The deployment is often ad hoc rather than planned.

- Ability to cope with node failures: As nodes may die because of depleted batteries, WSANs must have the ability to adapt to node failures.

- Ability to tolerate hostile environmental conditions: Nodes may be deployed in harsh environments such as forests, war zones, or harmful industrial environments. WSANs must, therefore, be able to function in hostile environments.

2.2.2 Applications of WSANs

In WSANs, wireless sensors are designed to monitor temperature, humidity, lighting conditions, pressure, and noise level [25, 8]. As a result, WSANs can be used in a wide variety of applications. These applications can be grouped into the following areas.

- Military Applications: WSANs can be an essential part of military command, control, communications, computing, intelligence, surveillance, reconnaissance, and targeting (C4ISRT) systems [10]. The wireless sensor nodes are cheap so the use of WSANs is a cost effective way of target tracking, battle field surveillance, and battle damage assessment [34].

- Environmental Applications: WSANs can be used in continuous data collection for an extended period of time to detect forest fires, floods, and pollution. Sensor nodes can be deployed in a forest and whenever a fire is detected the system could notify authorities before the fire gets out of control. Floods can be detected early using WSANs [14]. The ALERT flood detection system [17, 55] consists of different types of sensor nodes deployed over a large area where the nodes are used to measure rainfall, water levels, and other weather information. These sensors collect data and send them to a central station for further processing.

- Health Applications: In hospitals, WSANs track and monitor the positions and conditions of patients, and if a patient is in critical condition, the hospital staff is automatically notified [26, 37].

- Habitat Monitoring: WSANs are also used to monitor habitats. For example, they detect the environmental changes that occur in the burrows and their surroundings during breeding seasons of small seabirds, the access pattern of the nesting burrows by the parents between incubation and feeding [36].

- Home Applications: Sensor nodes can be attached to home appliances such as vacuum cleaners, refrigerators, DVD players [44], and water monitoring systems [33] to manage the appliances locally or remotely [10].

- Industrial Applications: In industrial automation such as process control, building automation, and access control, wired sensor networks have been used. But the cost of wiring and maintaining the sensor networks, the safety concerns of using cables in dangerous areas, and different protocols for different sensors make WSANs suitable [53].

2.3 Characteristics and Applications of VANETs

The characteristics and applications of VANETs are given below:

2.3.1 Characteristics of VANETs

Although VANETs and MANETs have some characteristics in common, VANETs have some unique features that are described below [11]:

- Highly Dynamic Topology: Due to the fast movement of vehicles, the network topology of VANETs can change very quickly. This makes conventional MAC protocols unsuitable for VANETs.

- Variable Node Density: In VANETs, the number of vehicles in a region varies over time and is dependant on the situation. Node density is typically low in rural areas but high in urban areas.

- Fast Node Movement: Vehicles can move very fast in VANETs. When two vehicles move in opposite directions at very high speeds (e.g., 100 km/h), they remain in each other's radio range for a very short period. MAC protocols for VANETs should consider the frequent link failure due to the high mobility.

- Predictable Network Topology: The movements of nodes can be predicted better than in general MANETs because they move on roads, follow traffic signals, and road signs [29, 16, 56].

- Available Battery Power: There is no limitation of battery power. Thus VANET protocols do not have a great need to be energy-efficient.

- Enough computational resources: Each vehicle is equipped with Global Positioning System (GPS), a high speed CPU, and many sensors. These resources help the vehicles to get exact information about their current position, speed, and direction [39] and run computationally complex algorithms.

2.3.2 Applications of VANETs

Each vehicle gets location information and data from the on-board GPS and sensors respectively. It communicates and shares information with other vehicles that are in close proximity to it. The uses of resources and the communication with others provide road safety and passenger comfort. The applications that are available in VANETs can be categorized into two types [11]:

- Non-Safety applications: These applications are used to provide comfort to the drivers and passengers by providing different types of information such as weather and traffic information. Users can also get information about nearby restaurants, hotels, and gas stations. In addition to that, they can receive or send text messages, access the internet, and play on-line games if the vehicle is connected to infrastructure [60, 29, 64].

- Safety applications: The safety applications help to improve road safety and avoid accidents. These applications provide warning about accidents, violation of traffic signals and stop signs [39], intersection collisions, approaching emergency vehicles, lane changes, wrong way driving, dangerous road conditions, post crash, pedestrian crossings [27], and much more.

2.4 Challenges in designing Wireless MAC protocols

Efficient medium access is one of the key concerns during communication in both wired and wireless networks. MAC protocols designed for wired networks are not suitable for wireless networks because of some inherent challenges in wireless medium that are described next.

Signal Fading: Vehicles, buildings, and trees serve as obstacles to radio signal propagation over the wireless medium. Radio signal propagates to the receiver on multiple paths due to reflection, refraction, scattering, and diffraction. This causes signal fading; signals may vary in length, may reach to the receiver at overlapping times because of the multipath propagation effect, and data may not be decoded by the receiver due to the phase distortion and inter-symbol interference.

Bandwidth: In WSANs and VANETs, nodes may not get the chance to access the channel for long period of time due to the limited bandwidth. So, effective bandwidth allocation method is required for both WSANs and VANETs.

Half-duplex radio: In wired networks, the Carrier Sense Multiple Access with Collision Detection (CSMA/CD) technique is used to access the shared medium. In CSMA/CD, a node does not start sending data if it senses any ongoing transmission and if it detects any collision during its data transmission then it terminates the transmission immediately. On the other hand, nodes use half-duplex radios in WSANs and VANETs thus they cannot detect collisions while transmitting.

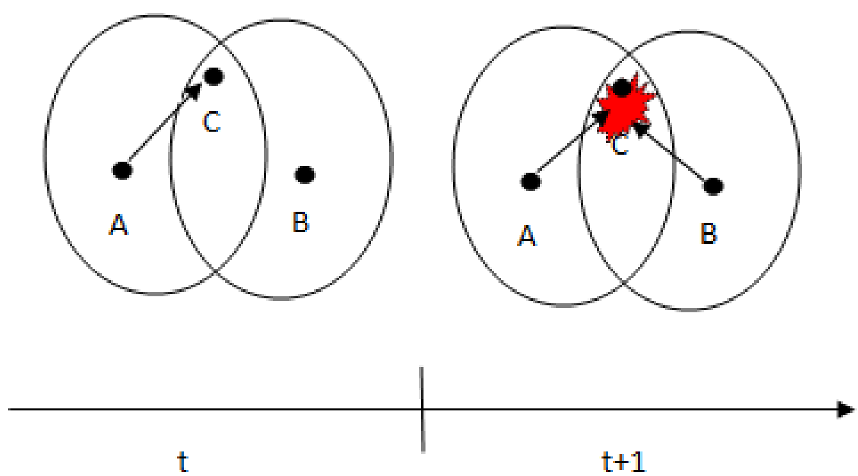


Figure 2.2: The Hidden Terminal Problem

Hidden Terminal Problem: The hidden terminal problem is one of the primary problems in multi-hop wireless networks that cause collision. Even when nodes use the carrier sense multiple access (CSMA) technique, this problem occurs when more than one sender that are not within each other's radio range transmit data at the same time and a collision happens at the receiver. Figure 2.2 illustrates this problem. The three nodes are labelled A, B, and C. Circles centered at node positions represent the radio ranges of each node. Note that A and B are not within each other's radio range. A transmits data to C at time *t* and B does not know about the transmission as B is out of the radio range of A. B transmits at time *t+1*. As a result, packets from A and B collide at C.

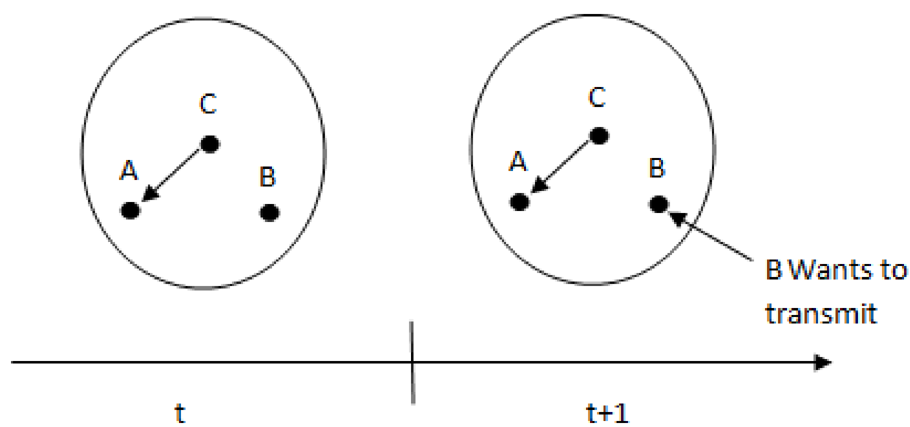


Figure 2.3: The Exposed Terminal Problem

Exposed Terminal problem: The exposed terminal problem causes unnecessary delay. In figure 2.3 A and B are within the radio range of C but A and B are not within each other's radio range. C is transmitting data to A at time *t*. B wants to transmit data to another node at time *t+1*. But it detects the transmission of C and waits until the transmission is over. However, since A is not within the radio range of B, it would not experience a collision if B transmitted its packet. Unless B detects this scenario it would wait, thus adding packet delay.

Security [13]: This is a major issue in wireless networks. The transmitted messages can be eavesdropped on and false messages can be inserted by the outsider. If the nodes are placed in hostile environments then attackers can take control over the nodes, may access sensitive information, and possibly change their behaviour. Also, if the topology changes dynamically then compromised nodes may broadcast false route or location information to others.

2.5 Metrics for evaluating MAC protocols

As mentioned earlier, WSAN nodes are typically placed in unattended areas where it is difficult to change batteries. Therefore, efficient use of battery power is required to prolong the network lifetime. So, energy consumption should be considered as the main metric in designing MAC protocol for WSANs. The other metrics are scalability, adaptability to network changes, latency, throughput, and fairness [23].

On the contrary, VANETs should take care of adaptability to rapid network changes, time responsiveness, average number of packet delivered, average number of packet collisions, throughput, reliability, fairness, and Quality of Service (QoS)[13].

Chapter 3

Ad-ATMA : A Novel Medium Access Control Protocol for Wireless Sensor and Ad hoc Networks

Our proposed MAC protocol for WSANs is presented in this chapter. Section 3.1 reviews the major sources of energy waste in WSANs. Existing wireless MAC protocols are surveyed in section 3.2. Some MAC protocols related to our work are described in detail in section 3.3. In section 3.4 our proposed MAC protocol is specified. Finally, Section 3.5 outlines our metrics and provides the performance evaluation of Ad-ATMA through simulation experiments.

3.1 Major sources of energy waste in WSAN MAC protocols

Energy is typically wasted from idle listening, overhearing, collision, and control packet overhead [19, 7].

Idle Listening: When the radio of a sensor node is on but the medium is free then the sensor node listens the medium idly. As a result, energy is wasted. More specifically, when data is generated in bursts, sensor nodes send data for small amounts of time but nodes listen to the medium idly the rest of the time they are awake. We note that MicaZ motes and Tmote Sky motes exhibit power consumption ratios of 1.13(receive):1(send):1.13(idle) and 1.11(receive):1(send):1.11(idle) [4, 48] which implies that nodes waste significant amount of energy due to idle listening.

Overhearing: Overhearing occurs when a node receives a packet which is not destined for it. Nodes use early rejection and message passing techniques to avoid message overhearing. In the early rejection technique, if nodes find that the message is not destined for it after decoding the header then it discards the remaining message. On the other hand, using message passing techniques, nodes can go to sleep if the message is not destined for itself. The nodes can get the duration of the message transmission from the RTS/CTS control packet and thus turn off their radios for the duration of the transmission.

Collision: Collisions happen when more than one transmitted packets are received by a receiver at overlapping times. As a result, the receiver cannot decode the packets. The main causes of collision occurring is the hidden terminal problem and the propagation delay (time required for the first bit of a packet to reach from the sender to the receiver) of packets. The collision forces the senders to send the packet again thus increases the delay.

Control Packet Overhead: Exchanging control packets among nodes is another source of energy wastage. MAC protocols often require the exchange of control packets for synchronization, neighbour and route discovery, and coordinating communication.

3.2 Existing Wireless MAC protocols

There are many ways to classify existing wireless MAC protocols. One way is to divide them into contention-based, contention-free, and hybrid protocols.

Contention-based protocols allow nodes to access the medium with very few restrictions. In contention-based protocols, nodes contend for the shared medium and this causes collisions. These protocols often incorporate strategies to reduce the number of collisions, like the Distributed Coordination Function (DCF) in the IEEE802.11 family. In DCF, a node senses the channel first for DIFS (Distributed Inter Frame Space) amount of time before transmitting and if the channel is free then it sends data immediately. Otherwise it waits for the transmission to over, waits DIFS amount of time and uses a random backoff to avoid the collision which is shown in figure 3.1. It chooses a uniform random value between 1 and the Contention Window (*CW*) and sets a countdown timer to that value. Initially, the *CW* is set to CW_{min}. During this counting if the channel becomes busy again then it freezes its counter and when the channel becomes idle, it waits DIFS amount of time again and then resumes the counter. It starts transmitting when the timer value reaches to zero. After receiving packet the receiver waits SIFS (Short Inter Frame Space) amount of time and then sends an ACK. If collision happens during the transmission the sender doubles the *CW* and it keeps doubling the *CW* for each transmission collision until it reaches to CW_{max}.

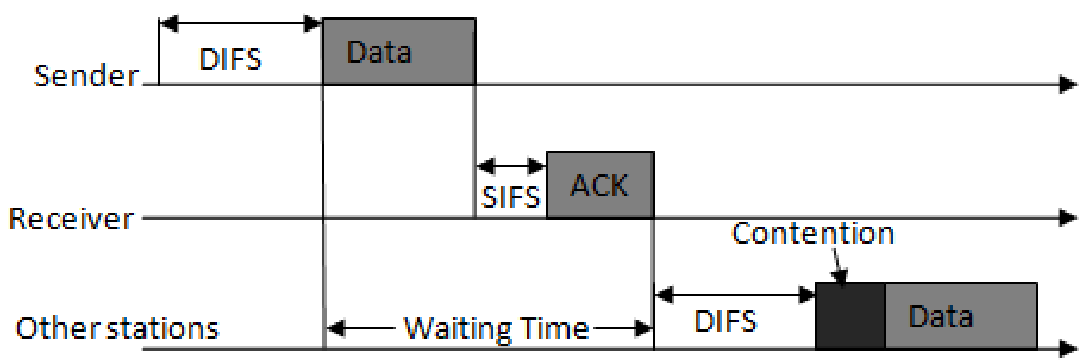


Figure 3.1: Unicast data transmission using IEEE 802.11 (adapted from [51])

The RTS/CTS mechanism shown in figure 3.2 is used to avoid the hidden terminal problem and the exposed terminal problem. After waiting DIFS amount of time a sender sends a RTS (Request to send) control packet which includes the receiver ID and the amount of time it wants to reserve the channel for transmitting data. After receiving the RTS, every neighbour sets its network allocation vector (NAV) for the whole data transmission period. If the receiver can accept this packet, it waits for SIFS amount of time and then replies with a CTS (clear to send) control packet that contains the duration for transmitting the data. So all the neighbours of the receiver can update their timetable or NAVs after detecting the CTS.

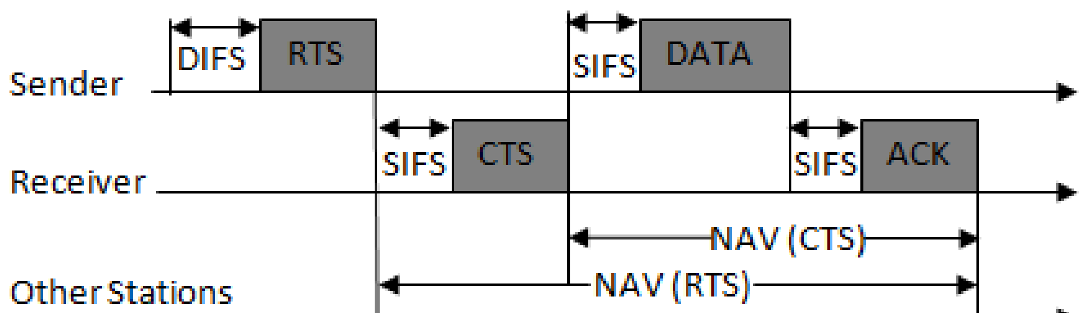


Figure 3.2: RTS/CTS mechanism for the hidden terminal problem

(adapted from [51])

Contention-free (Reservation-based) protocols (attempt to) prevent contention during packet transmission by explicitly scheduling packets. Frequency division multiple access (FDMA), code division multiple access (CDMA), and time division multiple access (TDMA) are examples of contention-free MAC protocols. In FDMA, the available frequency band is partitioned into number of channels and each channel is assigned to at most one node at any time. Thus the nodes can access their channels without any collisions. On the other hand, the ability of the receiver to tune to the channel that is used by the transmitter during communication, the narrowband nature of the communication and frequency synchronization make the implementation of FDMA complex. In CDMA, each node is assigned with a unique code that helps each node to access the shared medium without any contention but managing the codes is not simple. TDMA is considered the most suitable for WSAN and VANET nodes. In TDMA, each frequency channel is split into number of time slots and these slots are allocated to nodes. Nodes can access the medium in their own time slot without any collisions. To avoid overspreading of the channel in adjoining time slots, tight time synchronization is required and this can be achieved using one of the many good time synchronization algorithms that have been proposed in the literature [54].

Hybrid protocols attempt to combine the advantages of contention-free and contention-based protocols by allowing an initial contention period which is used by nodes to reserve time slots and then a contention-free period during which nodes that with reserved slots transmit their data without collisions.

3.3 Existing WSAN MAC protocols

Some WSAN MAC protocols are TDMA based, e.g., [28, 38], while others are contention-based protocols [61, 58]. TDMA-based protocols are intrinsically more energy efficient due to the absence of collisions. However, it is hard to design fully distributed TDMA protocols. Contention-based MAC protocols for WSANs can be further classified as synchronous and asynchronous.

3.3.1 Asynchronous Protocols

Asynchronous protocols (e.g., BMAC [45], WiseMAC [24], and XMAC [20]) allow nodes to have independent sleep-listen schedules, but with fixed-length sleeping periods. A sender having data to send must precede the data packet with an extended preamble (at least as long as the sleep period of the receiver). Typically, asynchronous protocols perform worse in heavy loads. This is due to lack of clock synchrony and also due to the higher latency and lower throughput, caused by the long preambles preceding data packets.

Berkley-MAC (B-MAC) [45] is an asynchronous protocol that is decentralized in nature. In B-MAC, nodes do not need any explicit synchronization during communication. Nodes use a low power listening (LPL) mode to sample the channel each time after periodic wake up to detect transmitted packets. As there is no explicit synchronization so sender must send a preamble that is longer than each node's sampling period. So after wake up, receiver senses the preamble and receives the packet. Receivers using WiseMAC [24] send their wake up schedules in the acknowledgement packets. Therefore, sender knows the wake up time of the receiver thus shortens the preamble.

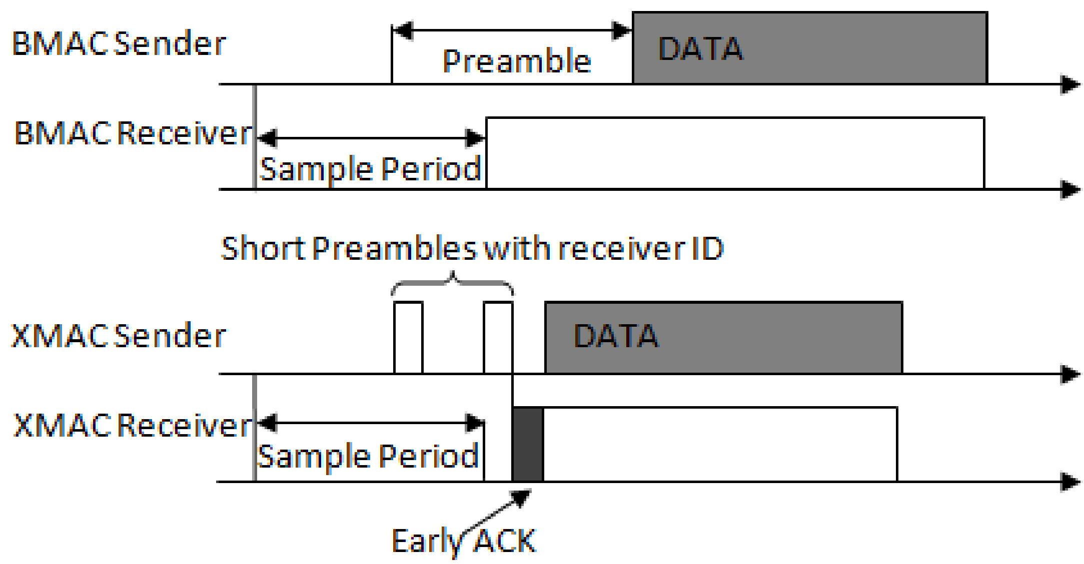


Figure 3.3: Illustration of B-MAC and X-MAC (adapted from [7])

X-MAC [20] improves on B-MAC by shortening the extended preamble as presented in figure 3.3. Long preambles result in wasted energy due to the overhearing problems at the non-receivers to check if this is destined for them. Also, the receiver has to wait the whole preamble period even if it wakes up at the beginning or in the middle of a preamble and that causes energy wastage both at sender and receiver. Moreover, the long preamble increases the latency at each hop and thus decreases throughput. X-MAC uses a set of short preambles rather than a long preamble that helps to avoid overhearing problem by including the receiver ID in the preambles so that nodes other than the receiver can go back to sleep quickly. If the intended receiver wakes up early then after getting the ID in the preamble it sends acknowledgement to the sender and that helps the sender to stop sending the remaining preambles. Thus X-MAC reduces energy consumption and decreases latency.

3.3.2 Synchronous Protocols

In synchronous approaches like SMAC [61], TRAMA [46], and ADV-MAC [49] nodes synchronize their sleep-listen schedule with their neighbours.

3.3.2.1 Traffic-Adaptive MAC (TRAMA)

TRAMA [46] is an energy-aware, collision free MAC protocol based on TDMA. TRAMA does not assign any time slot to nodes that have no packets to send. It uses a distributed election scheme to select a node that can use a particular slot. The traffic information at each node is used as selection criteria. The TRAMA frame has two parts. The first part is known as reservation period used to exchange two-hop neighbours information and their schedules using neighbour protocol (NP) and schedule exchange protocol (SEP). After that Adaptive Election Algorithm (AEA) selects the transmitter and receiver in order to send data without any collisions in the second part which is schedule-access period. However, it uses many control packets and a lot of computation is required to schedule the nodes for transmitting the packets.

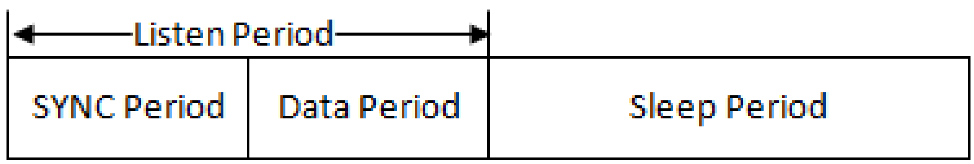


Figure 3.4: Listen period and Sleep period of SMAC (adapted from [19])

3.3.2.2 Sensor MAC (SMAC)

SMAC [61] is an energy efficient contention-based protocol. In order to conserve energy, SMAC introduces a sleep period just after the listen period and follows a periodic listen-sleep cycle. As shown in figure 3.4 the listen period is further divided into a SYNC period and a data period. During the SYNC period a node exchanges its schedule information with its neighbours. Thus a virtual cluster of nodes that follow the same schedule is formed. In the data period, each node uses the IEEE 802.11 protocol with the RTS/CTS mechanism to send a packet. The node that sends the RTS first accesses the medium and other neighbouring nodes go to sleep. The sleep period of SMAC is set at the beginning of node deployment. Note that a longer sleep period allows nodes to conserve more energy but increases latency. On the other hand, latency can be reduced by making the sleep period smaller that increases the energy consumption. The static sleep period makes SMAC inefficient in variable traffic load.

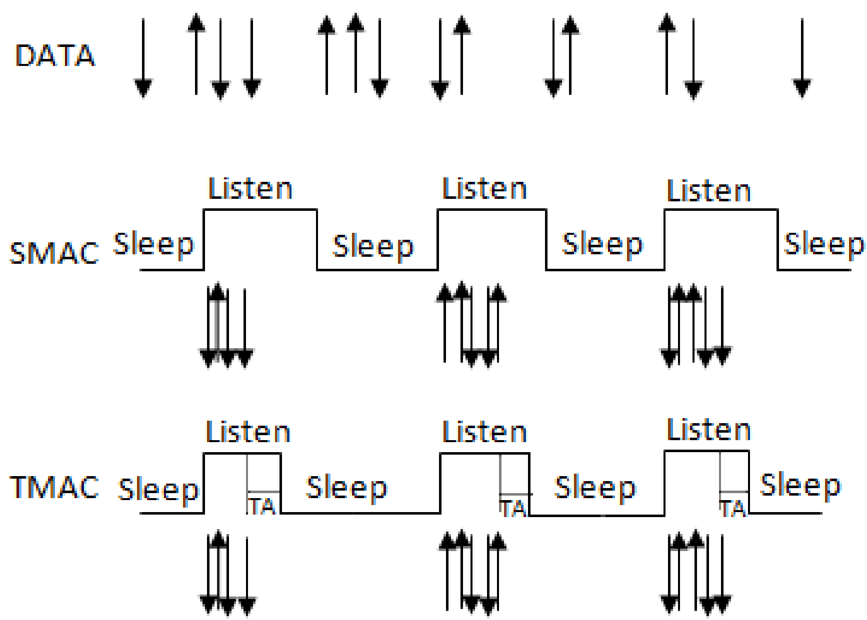


Figure 3.5: Difference between TMAC and SMAC (adapted from [19])

3.3.2.3 Timeout MAC (TMAC)

In SMAC, nodes sense the medium in the listen period even if the medium is free that causes energy waste due to idle listening. TMAC [58] reduces this energy consumption by adapting the listen period dynamically. A listening timeout mechanism is used when the node does not receive any packet for a certain period of time known as *TA* period in the listen period and goes to sleep after the *TA* period. Thus TMAC increases the sleep period and conserves energy. The *TA* period is greater than the time requires a hidden node to receive the RTS, can be expressed as $TA > (T + R + C)$ where T is the duration of the maximum contention window $(CWmax)$, R is the length of the RTS, and C is the time interval between the RTS packet and CTS packet. Figure 3.5 shows that TMAC conserves more energy than SMAC by dynamically adjusting the listen period.

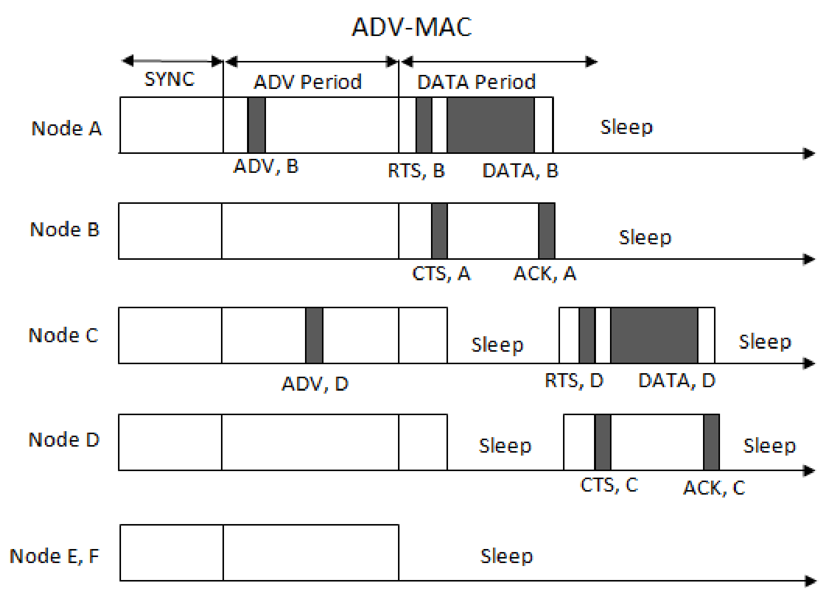


Figure 3.6: Example of ADV-MAC (adapted from [7])

3.3.2.4 Advertisement-Based MAC (ADV-MAC)

Although TMAC reduces energy waste due to idle listening, nodes still listen the medium idly if they hear a CTS or RTS even if they are not the part of the transmissions. A better way of energy consumption is proposed in ADV-MAC [49] which is shown in figure 3.6. ADV-MAC also works well under different traffic conditions like TMAC. In ADV-MAC, the listen period is now divided into SYNC period, ADV period, and data period. The ADV period is split into several slots. A node wishing to transmit data randomly selects a slot and waits for its slot time. If the channel is free at the beginning of the chosen slot, it transmits a small ADV packet that contains the receiver ID. If the channel is busy then it randomly picks another slot from the remaining slots and waits again for its slot time. All nodes except the senders and the receivers go to sleep in the data period. In the data period, senders wait random amounts of time before sending RTS and the sender that sends the RTS first continues its transmission. A sender uses a single RTS for all its data packets. The neighbours of the sender other than the receiver switch their radios off for the duration of the transmission indicated in the RTS packet. Note that receivers do not send ACK upon receiving ADV packets. So if an ADV packet collides, the intended receivers are asleep in data period. If this happens, a sender times out as no CTS is received and also goes to sleep.

3.3.2.5 Advertisement-Based TDMA (ATMA)

ATMA [50] is a distributed algorithm running at all nodes (that are assumed to be closely synchronized), and divides time into frames. Figure 3.7 illustrates a typical trace of ATMA. Each frame has two parts - a contention window and a data window. The contention window is used to schedule data transmissions in the data window without any contention or collisions in the ideal case. ATMA calls the contention window the ADV (Advertisement) period, and packets transmitted in this window are called ADV packets. Nodes use ADV packets to both inform receivers as well as reserve data slots. Each ADV packet has the sender and receiver IDs and the data slot (in the data window) that the sender would like to reserve. Nodes receiving ADV packets send an acknowledgement called A-ACK to inform all nodes in the senders two-hop neighbourhood about the upcoming data transmission.

The contention window is divided into microslots which are typically smaller than ADV packet durations. This is a key feature of ATMA. Each node that has packets to send chooses a microslot number, and starts a countdown timer at the beginning of the contention window with this number. This timer is paused whenever the node senses the medium to be busy. When this timer eventually expires, the node transmits its ADV packet. If the receiver receives this packet, an ACK packet is sent immediately. If there is a collision, or the packet is corrupted in the network, no ACK is sent. Note that this freezing of the timer implies that some nodes may not get a chance to send an ADV packet in a frame even though it chose a valid microslot number. We say a node is frozen out if this happens. Nodes that experience collisions or being frozen out attempt to transmit the ADV packet again in the next frame. There is a tradeoff involved in choosing the number of microslots: the finer the division of time the lower the collision probability but the tighter the time synchronization required. Of course, slow hardware and non-real-time operating systems on most available sensor nodes limit the time synchronization accuracy achievable in practice. Our proposed algorithm Ad-ATMA outperforms ATMA without using more energy, by dealing carefully with frozen out nodes. Note that unlike ATMA, Ad-ATMA reserves slots in only one frame at a time. However, Ad-ATMA can be easily modified to handle multi-frame reservations.

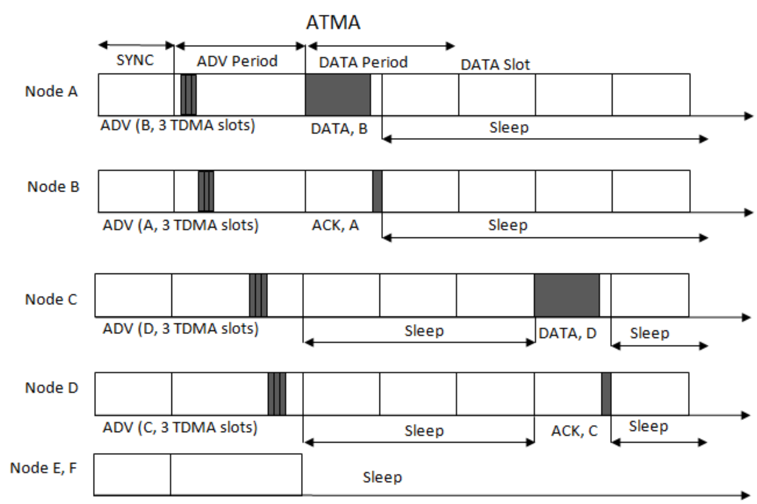


Figure 3.7: Example of ATMA (adapted from [7])

3.3.2.6 Adaptive MAC (AdAMAC)

AdAMAC [7] is another recent protocol that removes some limitations of ATMA and provides better performance. In AdAMAC, the contention window is known as reservation period and the data period represents the data window. Like ATMA, it follows the same contention procedure for reserving a data slot. Moreover, it prioritizes the nodes that have contended for a data slot in the contention window but due to collisions or run out of time unable to reserve data slots in the data window. To prioritize these unsuccessful nodes AdAMAC makes the reservation period (*RSVmax*) half for that unsuccessful nodes and continues reducing by half until it reaches to *RSVmin*. If any unsuccessful node reserves a data slot successfully then from next attempt it uses the *RSVmax* again. Reducing the reservation period increases the probability of collisions so the unsuccessful nodes contend in one of the frames from the upcoming two successive frames only if the number of senders greater than a certain threshold.

It also uses RSV/ACK1/ACK2 mechanism instead of ADV/A-ACK mechanism of ATMA so that more sender can contend in the contention window. In ATMA, the neighbours of the sender but not the receiver assume that the sender has reserved the intended data slot even if the sender does not get any ACK1 (similar to A-ACK) from the receiver. Thus those neighbours of that sender do not contend for that data slot. To solve this problem, AdAMAC introduces a small ACK2 packet which is sent by the sender to the receiver immediately after getting the ACK1 packet from the receiver so that all the neighbours of the sender are aware of the transmission within that reserved data slot. If the neighbours do not hear the ACK2 from that sender then they know that the data slot is not reserved by the sender and they can contend for that data slot.

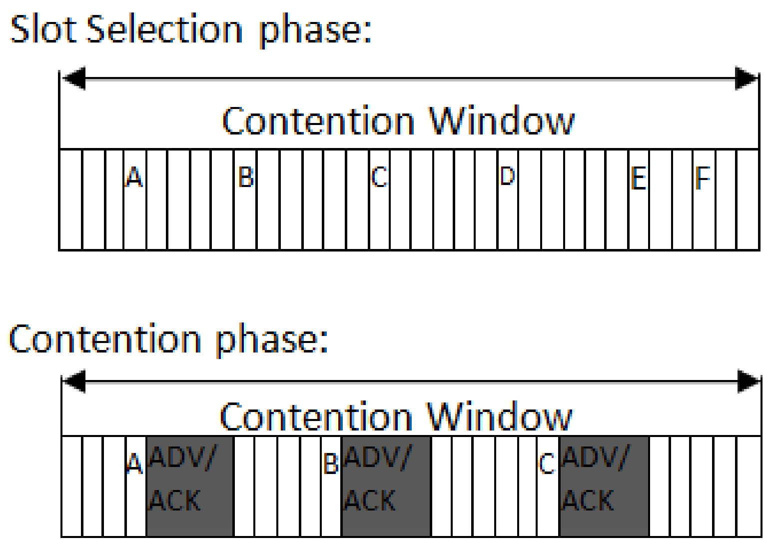


Figure 3.8: Frozen out nodes are unable to send their ADV packets in ATMA

3.4 Algorithm Ad-ATMA : Our Proposed MAC Protocol

In ATMA, some nodes that pick random microslots in the Advertisement period do not get the chance to access the medium because they run out of time. So they contend again in the next frame. Thus energy consumption and latency are increased and throughput is decreased. As presented in figure 3.8, although node D, E,and F have picked random microslots, the Contention window is over before they get the chance to access the medium.

Ad-ATMA improves on the ATMA algorithm by splitting the contention window into two parts which is shown in figure3.9. The first part is called the Selection Window and has the same function as the contention window of ATMA. The second part is known as the Surplus Window. This is used to allow nodes frozen out in the Selection Window to transmit their ADV packets. No other packets are sent in the Surplus Window.

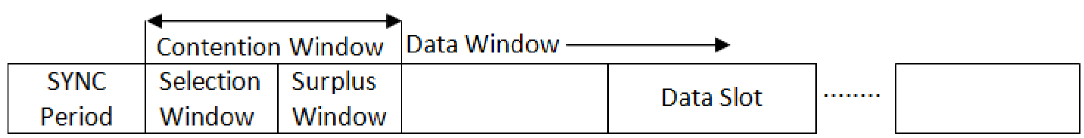


Figure 3.9: Frame Structure of Ad-ATMA

For fair comparison, the Selection and Surplus Windows should together equal the contention window in ATMA and AdAMAC. As shown in figure 3.10 now the senders A, B, C, D, E, and F choose random micro slots within the Selection Window rather choosing within the whole contention window like ATMA (figure3.8). Thus D and E also get the chance of sending data in this frame that decreases delay and increases throughput.

Setting the value of the Selection Window is not straightforward. Intuitively, re-

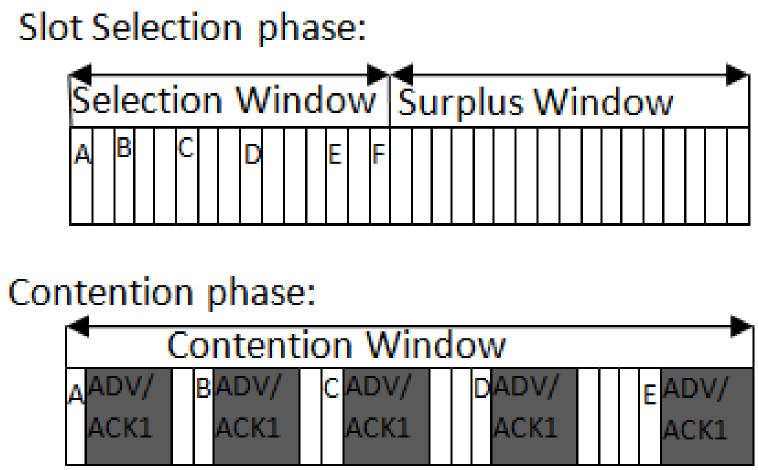


Figure 3.10: Frozen out nodes are also sending their ADV packets in Ad-ATMA

ducing the Selection Window increases the probability of collisions and increasing the Surplus Window allows frozen out nodes to successfully send ADV packets. Having a Surplus Window is beneficial when the benefit of having frozen out nodes sending their ADV packets outweighs the extra delay caused by collisions. Intuitively for very small number of senders, there are very few frozen out nodes and having small Surplus Windows suffices. When there is a very large number of nodes, we expect that any reduction in the Selection Window will increase in collisions and will probably outweigh the gains from having fewer frozen out nodes. However at moderate values of senders there is a significant performance gain to be had from the splitting of the contention window. This intuition is borne out in our simulations.

All of this intuition still does not explain how to set the value of the Selection Window. In Ad-ATMA we adaptively set the Selection Window by estimating the number of neighbouring senders (Section 3.4.1) and choosing the best Selection Window for that number of senders (Section 3.4.2).

3.4.1 Estimating the number of neighbouring senders

A node running Ad-ATMA cannot directly measure the number of senders in its neighbourhood. So it indirectly estimates this number by observing the contention window (Selection and Surplus Windows) and counting the number of microslots with collisions and those in which successful ADV packets are transmitted. Then it uses a formula similar to that in RMAC [22], and sets the estimated number of senders as #senders = # successful packets +2 #collisions.

3.4.2 Choice of the best Selection Window

We choose the best size for the Selection Window given the number of senders using simulation experiments. We assume every node has packets to send at every frame for a fixed number of frames. We generate random deployments of nodes fixing the number of neighbours a node has and fix the size of the Selection Window. All nodes run Ad-ATMA to send packets for a fixed number of frames. We measure the number of microslots in which a single ADV packet was transmitted and the number of microslots with collisions. We repeat the experiment for different numbers of senders and different sizes of the Selection Window. We select the best value of the Selection Window from the measurements using a weighted sum of the number of collisions produced and the number of successful packets transmitted as the objective function. The weights used were the relative energy consumption of transmitting and receiving for real sensors [1, 4].

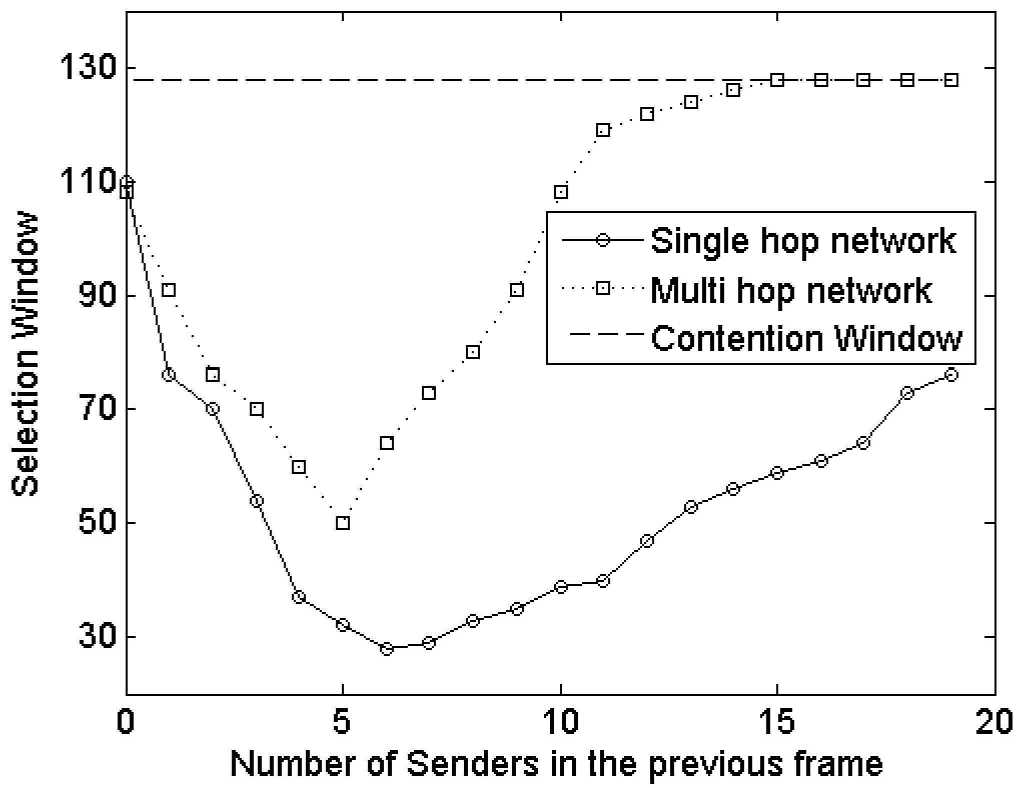


Figure 3.11: Best Selection Window with fixed contetion window

In figure 3.11 we plot the best Selection Window values in single-hop (all nodes are within each others radio range) and multi-hop networks that we have found from our experiments. The best Selection Window decreases with the number of senders initially because bigger Surplus Windows help reduce the number of frozen out nodes. However, when the number of senders are high, increasing the Selection Window reduces collisions and this outweighs the benefit from fewer frozen out nodes.

3.4.3 Effects of hidden terminal problem on the Selection Window

In single-hop networks, if a node starts transmitting all nodes can hear the transmission as they are within each other's radio range so there is no hidden terminal problem. Collisions only happen when more than one nodes pick the same microslot value. On the other hand, the hidden terminal problem is obvious in multi-hop networks. If more than one packets reach to the receiver at overlapping times then receiver is unable to extract the information of those packets. So for the same number of senders more collisions happen in multi-hop networks than in single-hop networks. As a result, The Selection Window values in multi-hop networks are larger than in single-hop networks. Since collisions due to the hidden terminal problem increases with the number of senders so the best Selection Window in multi-hop networks increases more quickly than in single-hop networks. Figure 3.11 shows that for the number of sender more than 14 the size of the Selection Window in multi-hop networks is equal to the size of contention window because the loss due to the collisions outweigh the benefit of having smaller Selection Window (larger Surplus Window).

3.4.4 Other heuristics

We add one heuristic to the steps to improve performance when the number of senders is very low. Intuitively when there are very few nodes, it makes sense to allow nodes to send more than one packet in a frame to reduce latency. Also, when there are very few nodes sending packets there could be collisions due to random choices (and not congestion) and rather than waiting a frame to try to retransmit it makes sense to attempt the retransmission in the current frame.

Putting these intuitions together, Ad-ATMA allows nodes to send a second ADV packet in a frame provided the number of senders is low. Ad-ATMA assumes that the number of senders is low if at least 70% of the microslots in the previous frame are empty. If the number of senders is low a node chooses a microslot randomly from the microslots remaining in the entire contention window (not just the Selection Window) and attempts to transmit a second ADV packet in that microslot.

3.4.5 Ad-ATMA description

Ad-ATMA runs in a distributed manner at each node. In each frame a node listens to the entire contention window.

1. At the start of each frame, each sender computes the estimated number of neighbouring senders, chooses a random microslot n in the Selection Window, starts a countdown timer at n and listens to the channel.

2. As soon as a node detects a microslot in use it freezes its timer until the channel is free again.

3. When the timer expires it sends the ADV packet. If the receiver gets the ADV pkt, it sends an ACK packet immediately. The sender upon receiving an ACK sends another ACK so that all nodes within the one-hop neighbourhood of the sender and receiver learn about the upcoming data transmission.

4. If the timer does not expire but the frame ends or too few microslots are left when the timer expires, the node tries to send the ADV packet in the next frame.

5. Those nodes that successfully reserved data slots transmit packets in the data slot. The nodes that do not send or receive packets in this frame are free to sleep through the entire data window.

3.5 Performance Evaluation

We implemented Ad-ATMA, ATMA, and AdAMAC in Matlab R2013a. We did not simulate SMAC, TMAC, ADVMAC as [50] showed that ATMA outperforms all of them in terms of energy consumption, latency, and throughput.

3.5.1 Our Model

We assume that our WSAN consists of nodes placed randomly in a rectangular two-dimensional region free of obstacles. Nodes are assumed to be identical and static, i.e., they do not change positions after deployment. We assume that the nodes are capable of sensing the channel and distinguishing between an idle channel, a single packet transmission in progress, and collisions (two or more packets being transmitted simultaneously). We assume that time is discretized and that all nodes operate in synchrony. Thus we assume implicitly that there is reasonable clock synchrony among nodes. We do not assume the presence of a routing infrastructure for our algorithm, since this is typically built using the MAC protocol.

We use two very simple models of traffic. The random traffic model assumes that each node generates a packet with probability p at each time step. The bursty traffic model assumes that a burst of data packets is generated at each sender periodically. Packet destinations are chosen uniformly at random from neighbours of the sender.

3.5.2 Our Metrics

We evaluate the performance of our algorithm using latency, packet delivery ratio (PDR), and energy consumption as our metrics. Latency is the time taken by data packets in travelling from the senders to the receivers. We will use both the latency distribution as well as the average latency to compare algorithms. PDR is the fraction of data packets successfully delivered to the intended receivers. We measure the fractions of time a node is asleep, idle listening and transmitting or receiving. We approximate energy consumed by nodes from these times using energy consumption figures obtained from real sensor hardware.

3.5.3 Simulation details

We simulate the algorithms in both single-hop and multi-hop networks. Most parameters are similar to [50]. We set the transmission rate to 250 kbps, the simulation time to 400 sec, and averaged measurements over 50 runs. The duration of frame is 236.4 ms and the duration of Contention Window is 12.8 ms. ADV slots are 0.1 ms long. The data slots are 12ms each. An ADV packet and the two ACKs are together 2 ms long. The radio range of each node is set to 100 m. Each packet is sent to a node chosen randomly from the senders one hop neighbours. We have also plotted the errorbars for latency plots that are very close to the mean values and almost impossible to notice.

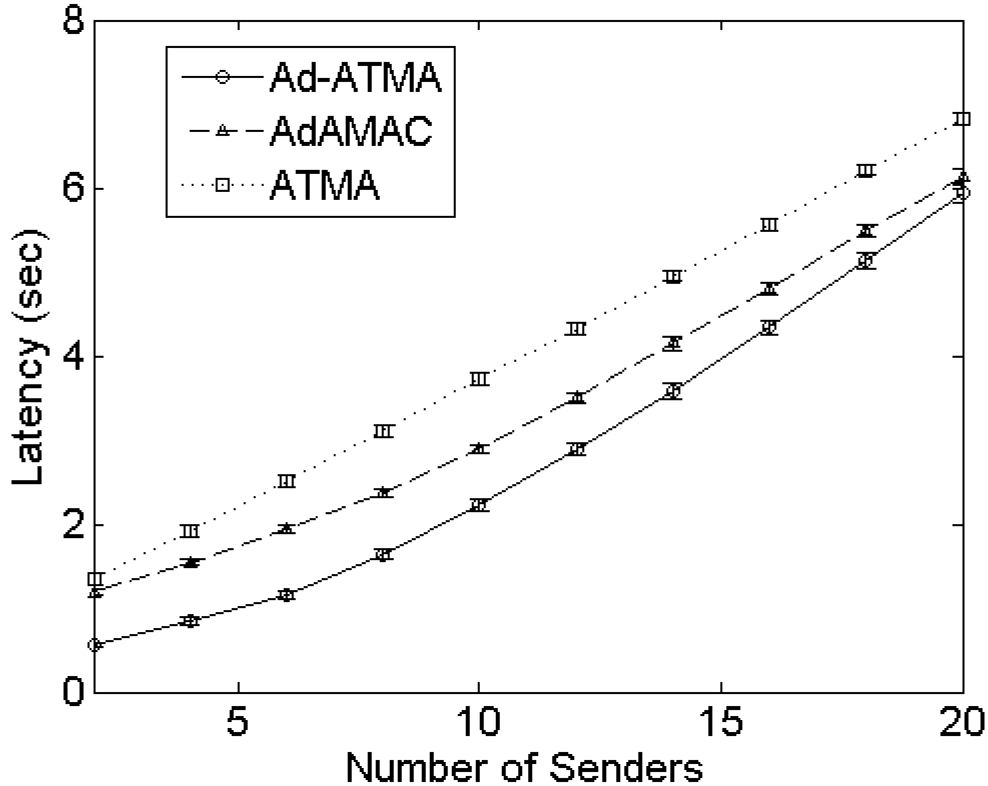


Figure 3.12: Latency vs Number of senders

3.5.4 Results for Single-hop networks using bursty traffic model

2-20 nodes are deployed randomly in an area of 50 m x 50 m. Nodes generate data packets in bursts of 3.5 sec at intervals of 20 sec. One data packet is generated per node per frame in a burst. Figure 3.12 shows that Ad-ATMA has 10-53% lower latency than AdAMAC and 22-58% lower latency than ATMA for upto 16 senders. For higher number of senders the Surplus Window is small and so Ad-ATMA shows less improvement.

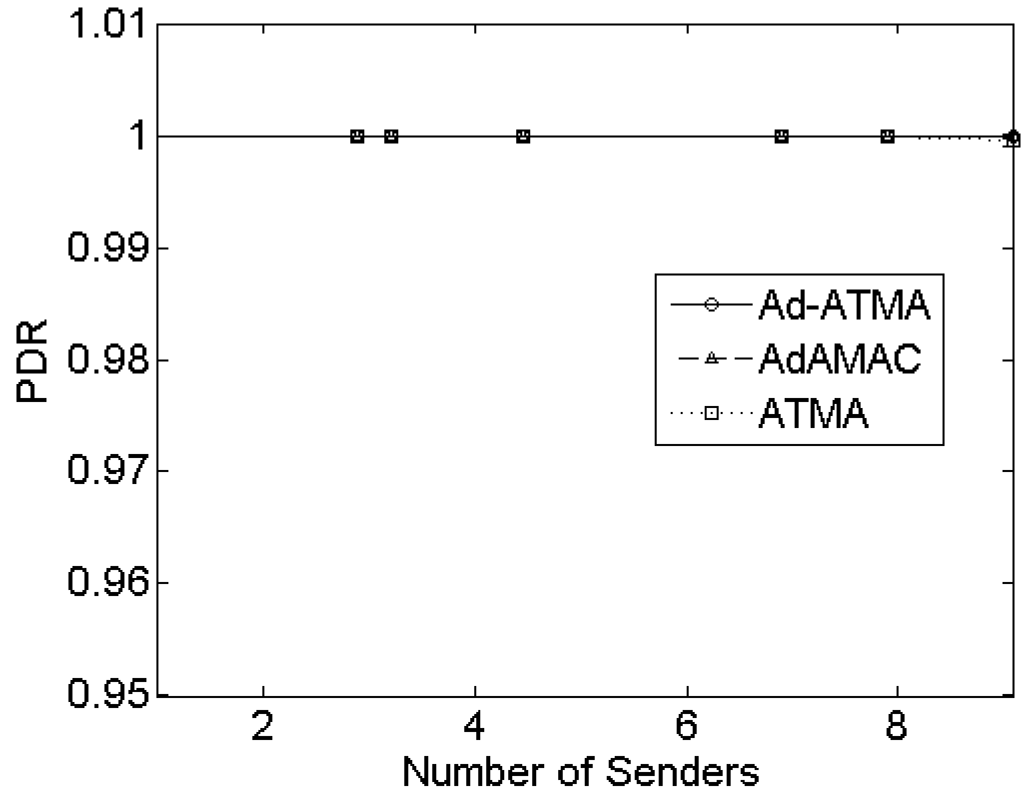


Figure 3.13: PDR vs Number of senders

All three algorithms deliver almost 100% packets which is shown in figure 3.13.

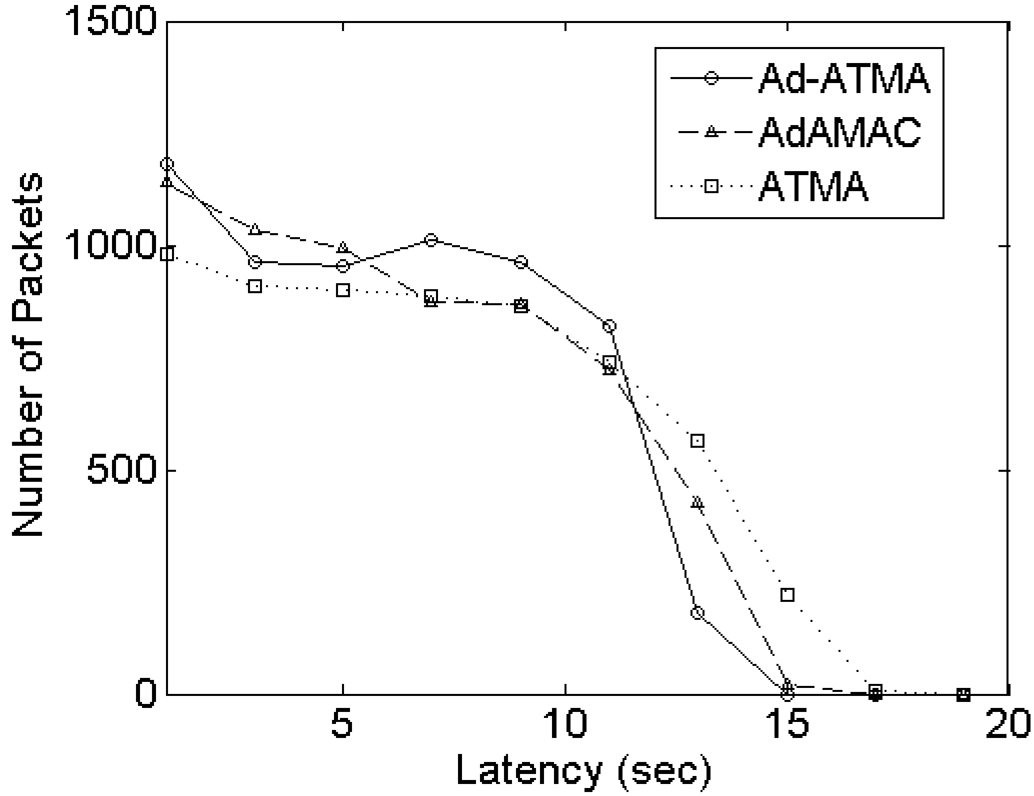


Figure 3.14: Latency distributions of delivered packets

We plotted the latency distribution for 20 senders in figure 3.14 which shows that Ad-ATMA has lower latency variation than both AdAMAC and ATMA.

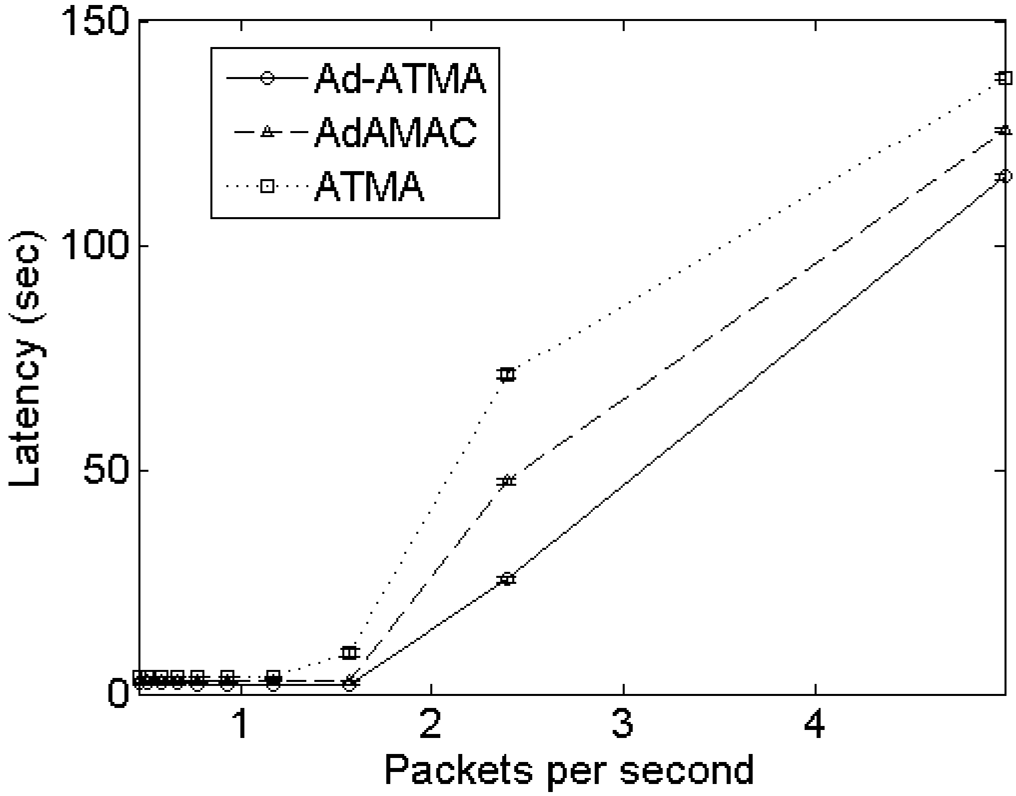


Figure 3.15: Latency vs data rate

Next, we varied the traffic load from 0.4 pkts/sec to 5 pkts/sec by keeping the number of nodes fixed at 10. Figure 3.15 shows that Ad-ATMA can reduces latency up to almost 20% less than AdAMAC and upto 40% less than ATMA.

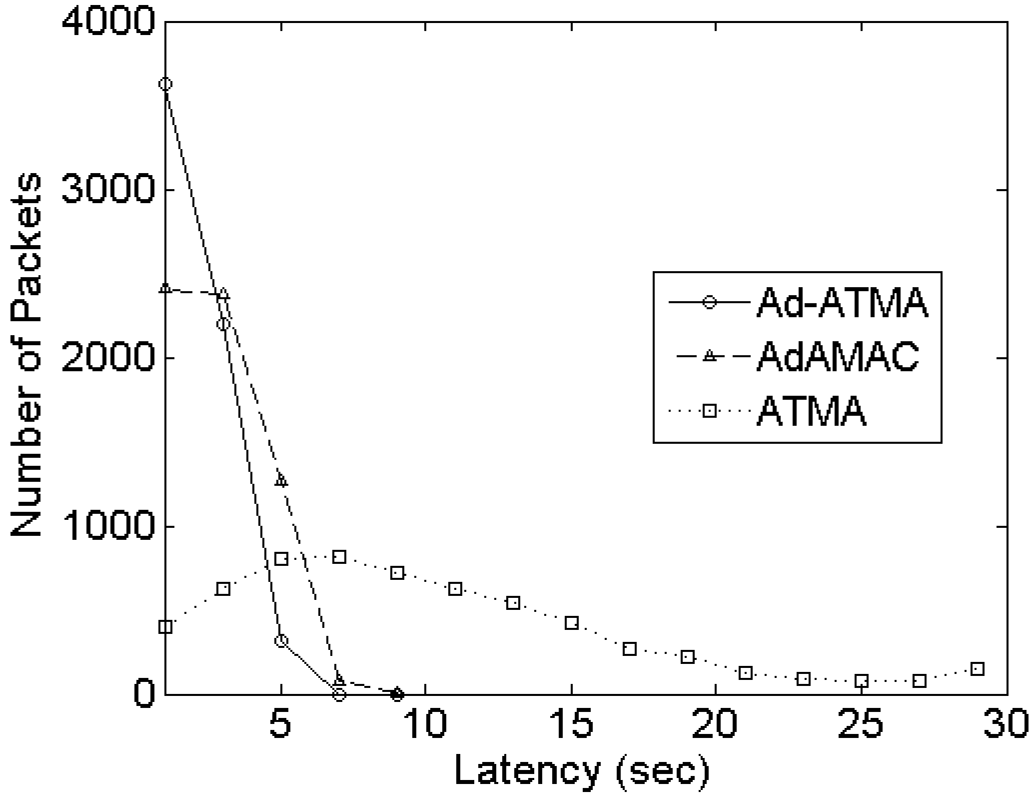


Figure 3.16: Latency distributions of delivered packets (1.56 pkts/sec)

Figure 3.16 shows the latency distribution of delivered packets, and Ad-ATMA is seen to have a much lower delay variation than the other two algorithms.

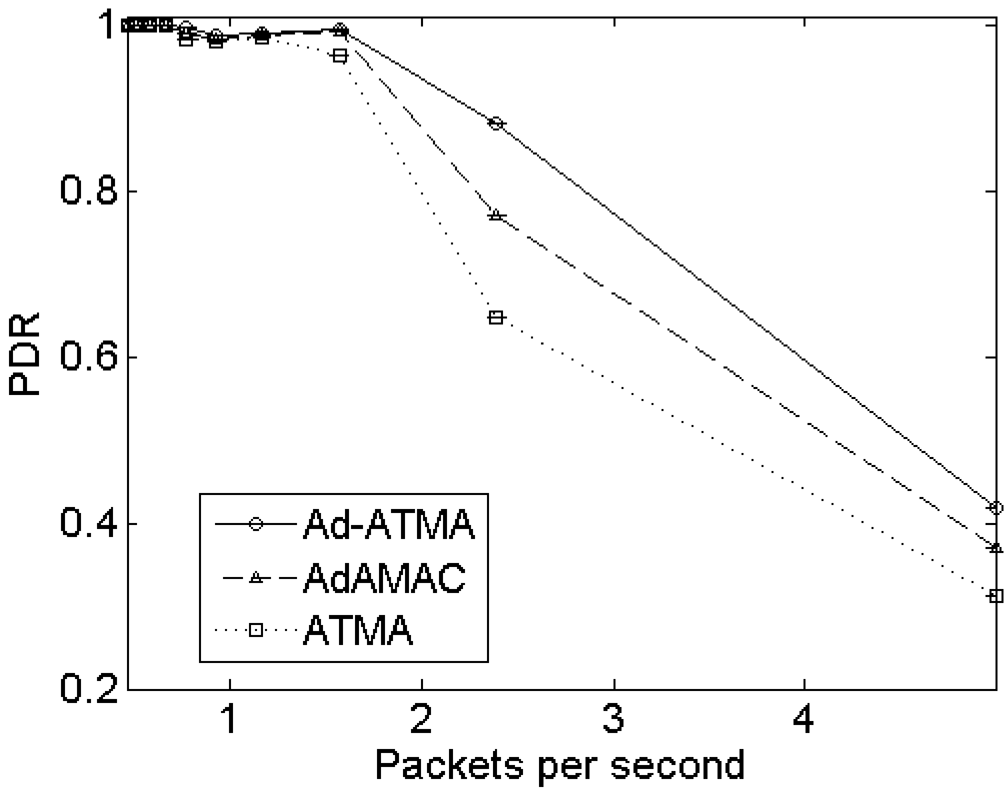


Figure 3.17: PDR vs Data Rate

In figure 3.17, we see that for higher data rates, Ad-ATMA has 13% higher PDR than AdAMAC and 26% higher than ATMA.

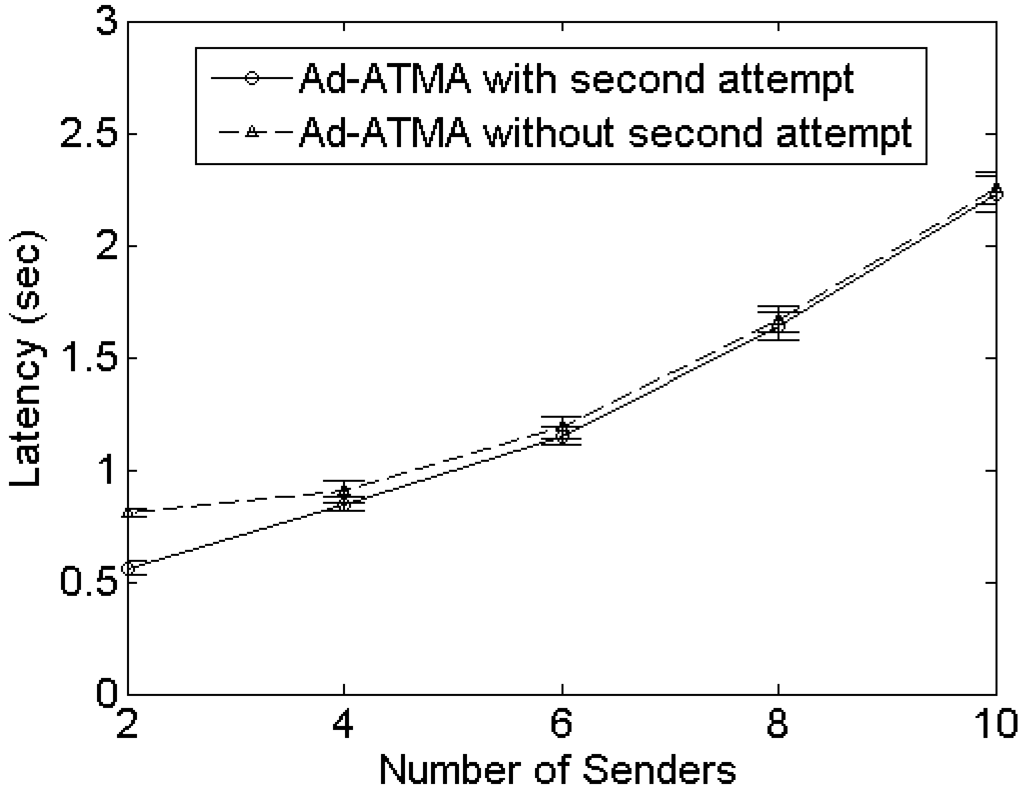


Figure 3.18: Effect of second attempt

The last figure (figure 3.18) of this section shows the effect of second attempt heuristic. Ad-ATMA allows senders to attempt second time if the number of sender is low. As a result, Ad-ATMA provides significantly better performance for senders upto 4 and after that Ad-ATMA shows slight improvement with the second attempt heuristic.

3.5.5 Results for Multi-hop networks using bursty traffic model

In this set of simulations, nodes were deployed randomly over an area of 300m x 300m to create multi-hop networks. Nodes use the same bursty traffic model to generate packets in burst. Node degree is varied from 1 to 10.

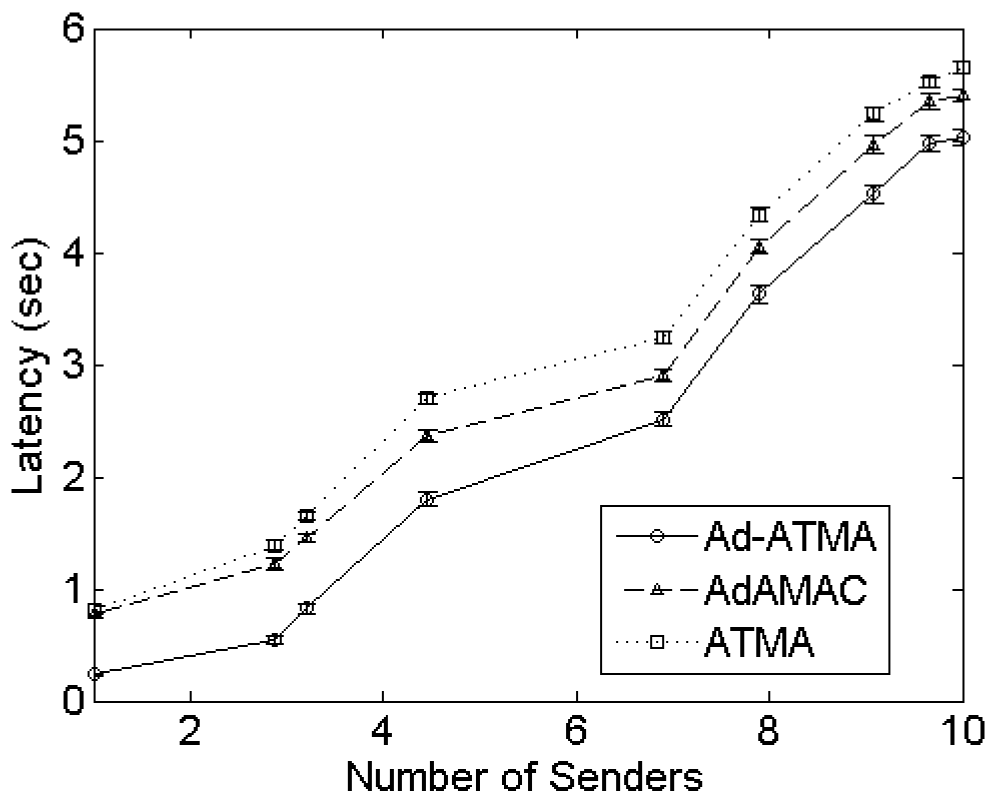


Figure 3.19: Multi-hop: Latency vs number of senders

Figure 3.19 shows that Ad-ATMA provides 10-69% latency reduction over AdAMAC and 16-71% over ATMA for 1-8 senders. The improvements decrease for higher numbers of senders due to collisions from hidden nodes.

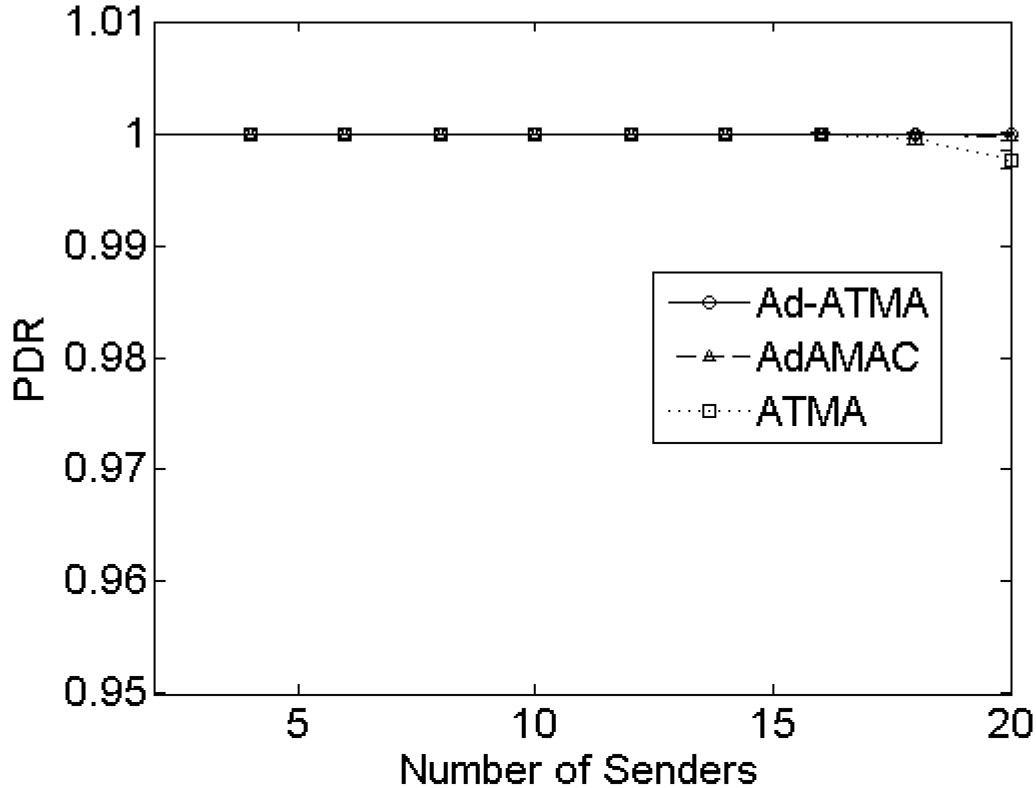


Figure 3.20: Multi-hop: Latency vs number of senders

Figure 3.20 shows that All three algorithms attained PDR close to 100% but Ad-ATMA has slightly higher PDR for large number of senders. It is because of the Surplus Window of Ad-ATMA which helps to clear more data packets.

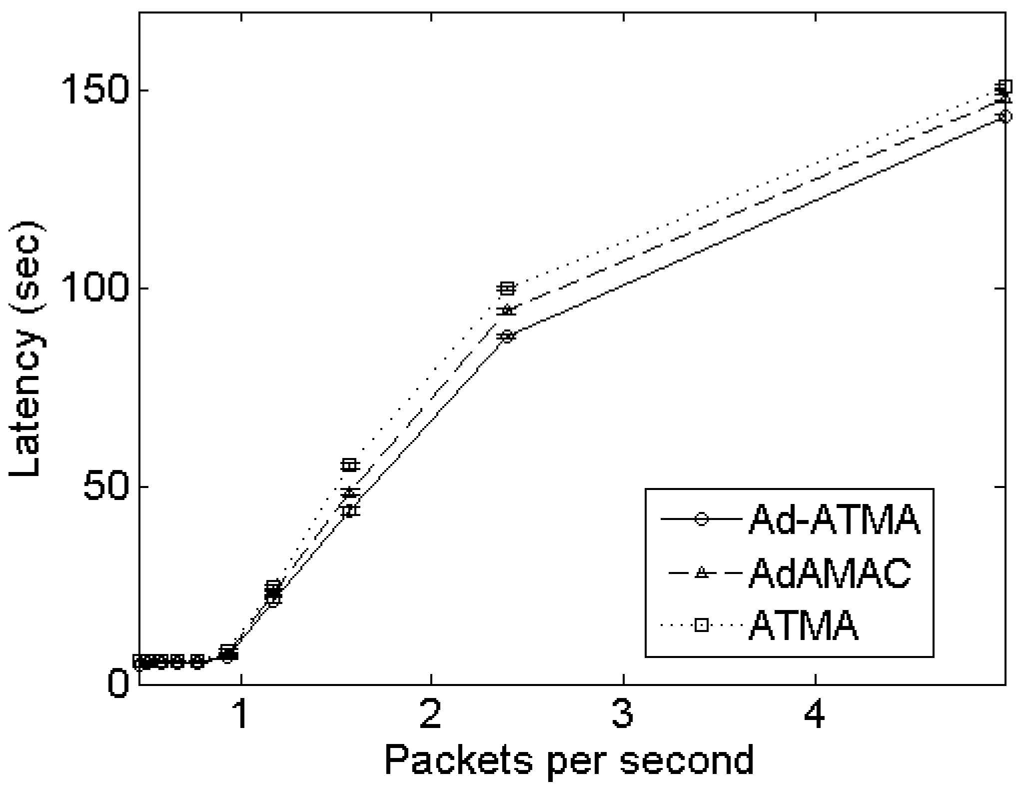


Figure 3.21: Multi-hop: Latency vs data rate

Finally, the node degree is set to 10 and the data rate is varied from 0.4 pkts/sec to 5 pkts/sec. We observe in figure 3.21 that Ad-ATMA produces up to 9% and 23% less latency than those of AdAMAC and ATMA respectively.

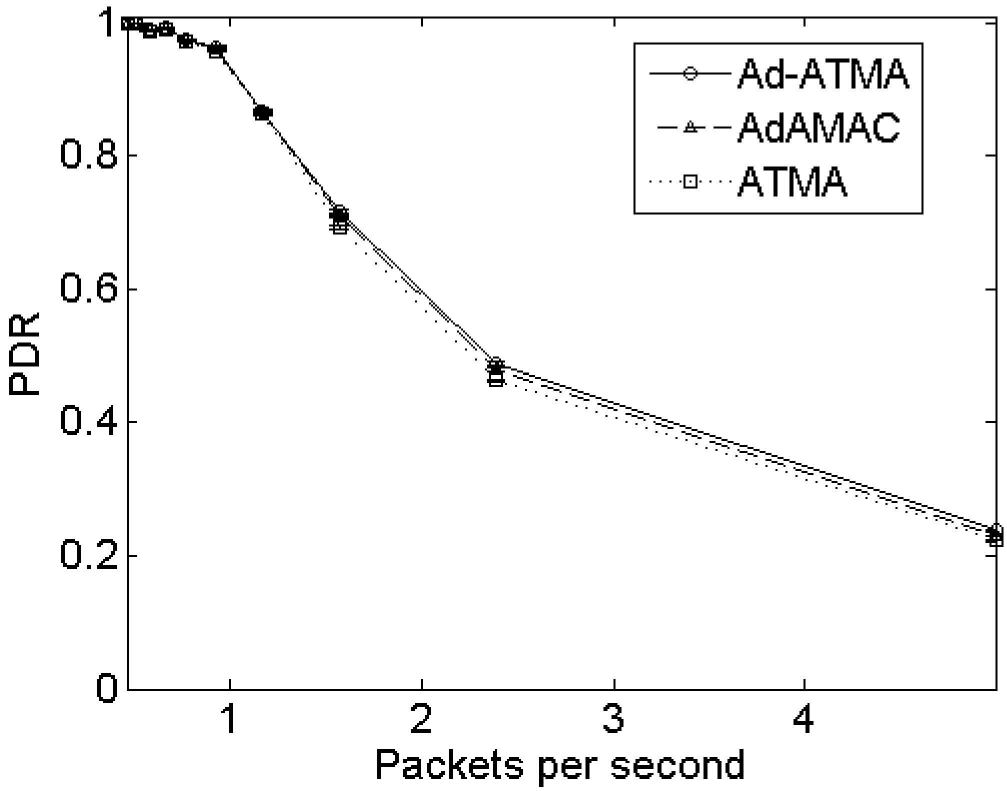


Figure 3.22: Multi-hop: PDR vs number of senders

In figure 3.22 Ad-ATMA has slightly higher PDR than ATMA and AdAMAC.

3.5.6 Energy Consumption for single-hop networks using bursty traffic model

We vary the number of nodes from 2 to 20 to analyze and compare the energy consumption of Ad-ATMA , AdAMAC and ATMA. We simulate the three algorithms until all the senders deliver their all generated packets to the intended receivers.

Table 3.1: Energy Consumption Comparison in Joules

Nodes	Ad-ATMA	AdAMAC	ATMA
2	2.60	2.61	2.61
4	5.22	5.24	5.25
6	7.85	7.88	7.91
8	10.49	10.53	10.58
10	13.15	13.21	13.27
12	15.83	15.90	15.98
14	18.54	18.62	18.70
16	21.25	21.35	21.44
18	24.00	24.09	24.20
20	26.76	26.87	26.96

In each algorithm nodes consume the same amount of energy during data periods as they send same number of packets. During ADV period of ATMA, RSV period of AdAMAC, and Contention period of Ad-ATMA all nodes are awake. Nodes consume energy by sending, receiving or idle listening during this time. We approximate the amount of time used for sending, receiving and idle listening by the nodes and calculate the total energy consumed during the simulation time. We use energy parameters of MicaZ motes [1, 4] to approximate energy consumption. MicaZ motes consume 17.4 mA current while transmitting and 19.1 mA current while receiving and idle listening. Assuming 3V batteries, transmission takes 52.2 mW and reception and idle listening takes 59.1 mW. We simulated the three algorithms for a fixed number of packets and measured total energy consumption.

As presented in table 3.1, we found that Ad-ATMA consumes slightly less energy than those of AdAMAC and ATMA because it delivers packets faster on the whole.

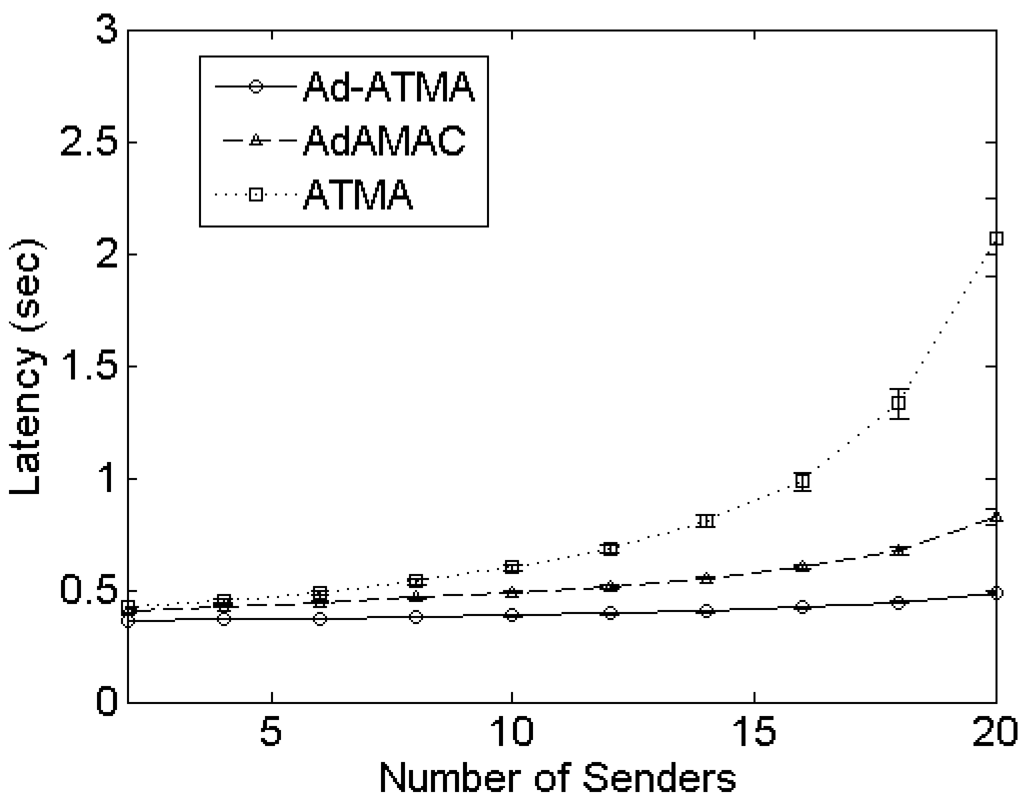


Figure 3.23: Latency vs Number of senders

3.5.7 Results for Single-hop networks using random traffic model

Nodes deployment and packet generation rate in random traffic model are similar with the bursty traffic model. In this model the performance of the algorithms are evaluated and compared based on random traffic. Figure 3.23 shows that Ad-ATMA has 11–40% and 14–76% less latency than AdAMAC and ATMA respectively. The Packet delivery rate is plotted in figure 3.24 and all the algorithm deliver almost all the packets.

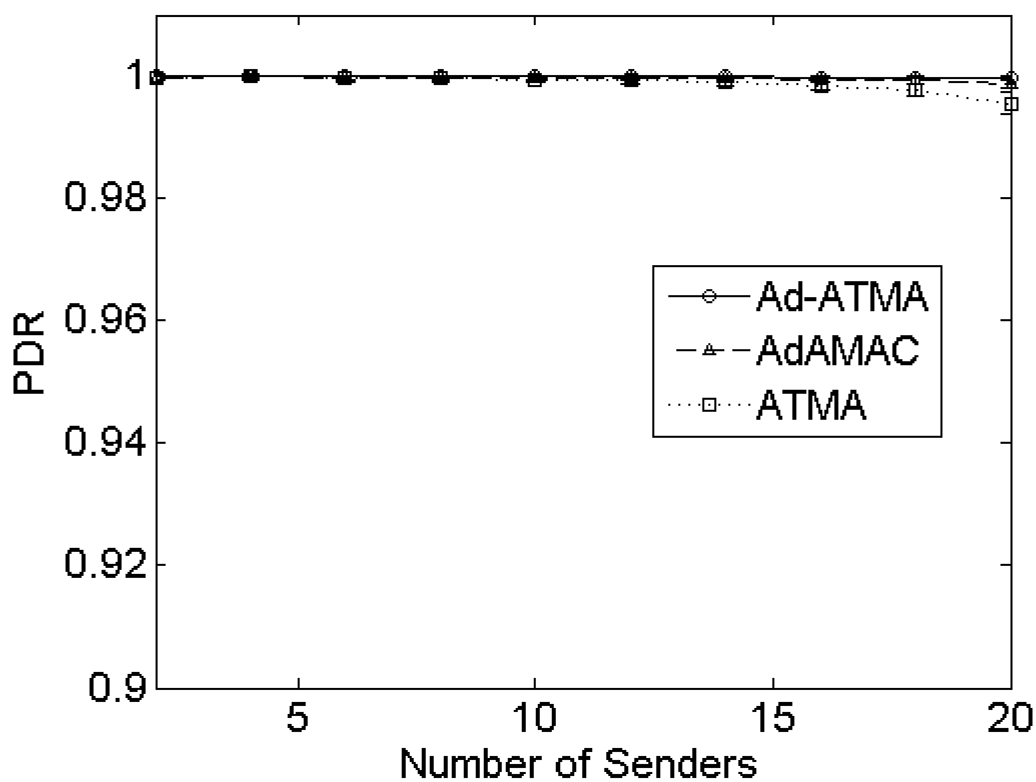


Figure 3.24: PDR vs Number of senders

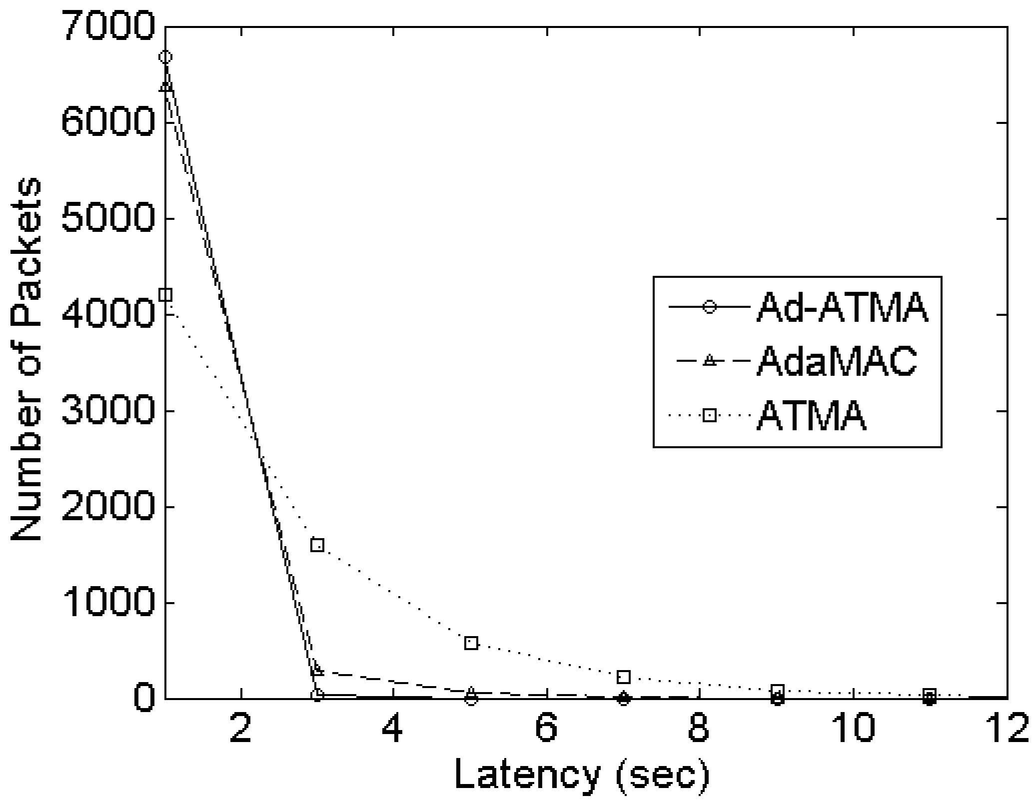


Figure 3.25: Latency distributions of delivered packets

Figure 3.25 shows the latency distribution when the number of senders 20. It is seen from the figure that the variation of latency of Ad-ATMA is less than AdAMAC and ATMA.

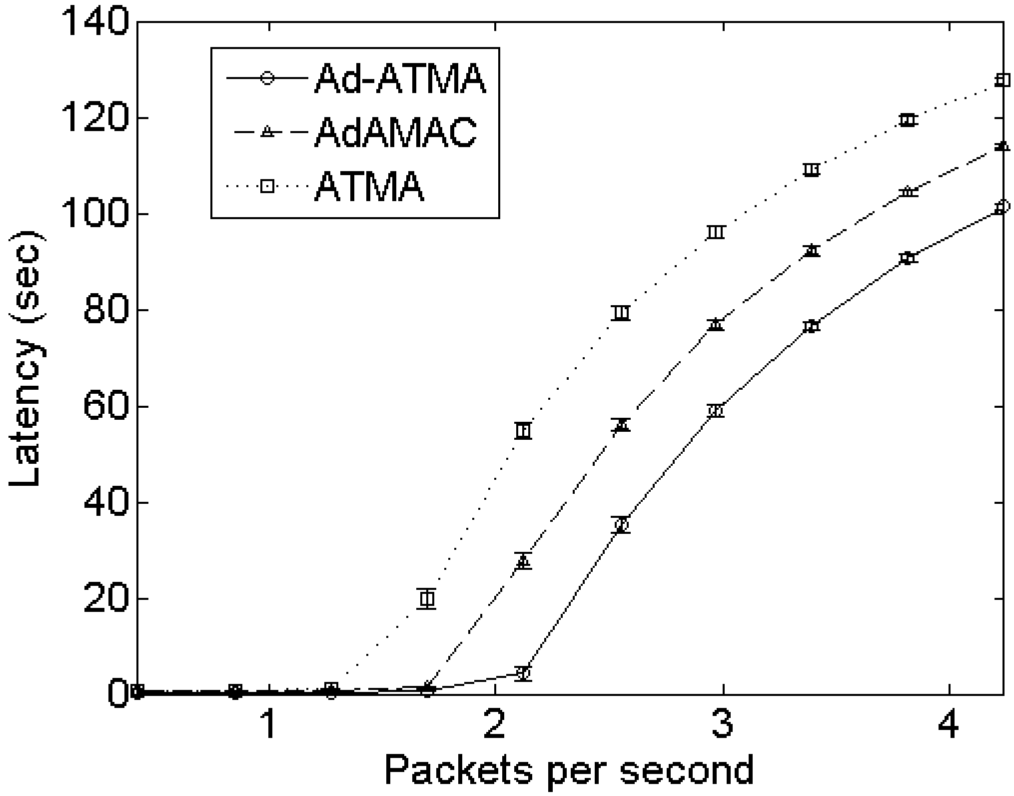


Figure 3.26: Latency vs Data rate

Similar to bursty traffic model we vary the data rate from 0.4 pkts/sec to 4.3 pkts/sec by fixing the node at 10 and observe the effect on latency and PDR in figure 3.26 and 3.27. Ad-ATMA reduces the latency upto 40% than AdAMAC and 50% than ATMA. Also Ad-ATMA delivers 13% and 27% more packets than other two algorithms respectively. The improvement in PDR is more pronounced from 2.12 pkts/sec to 4.3 pkts/sec.

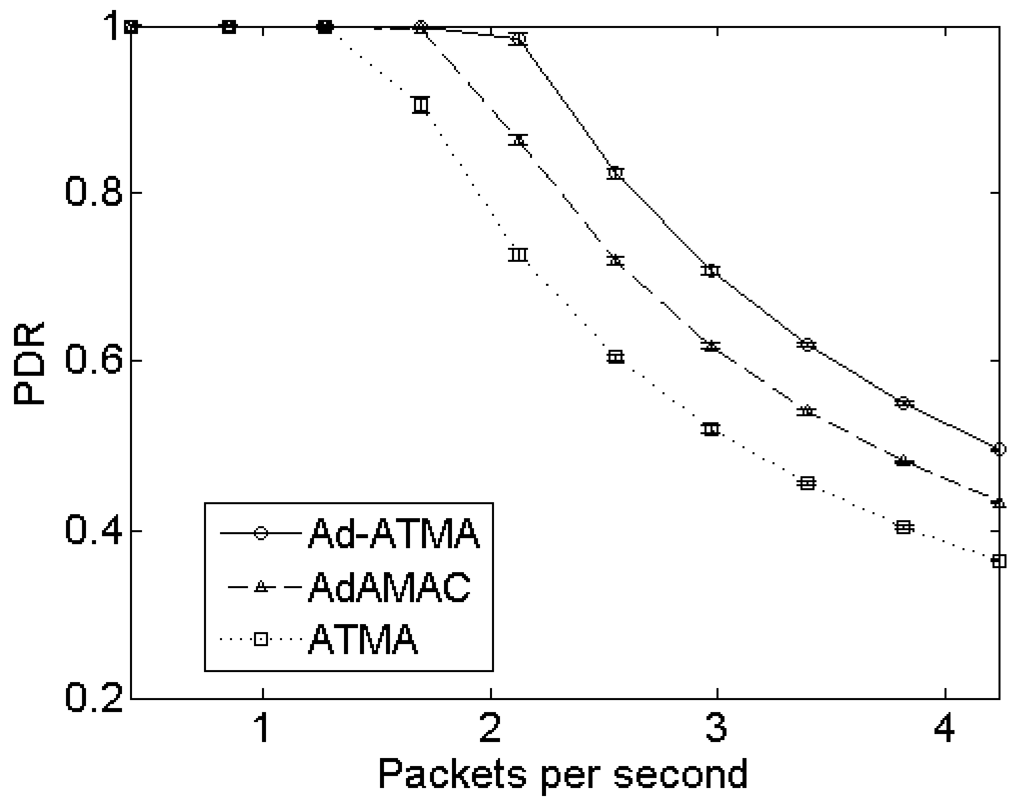


Figure 3.27: PDR vs Data rate

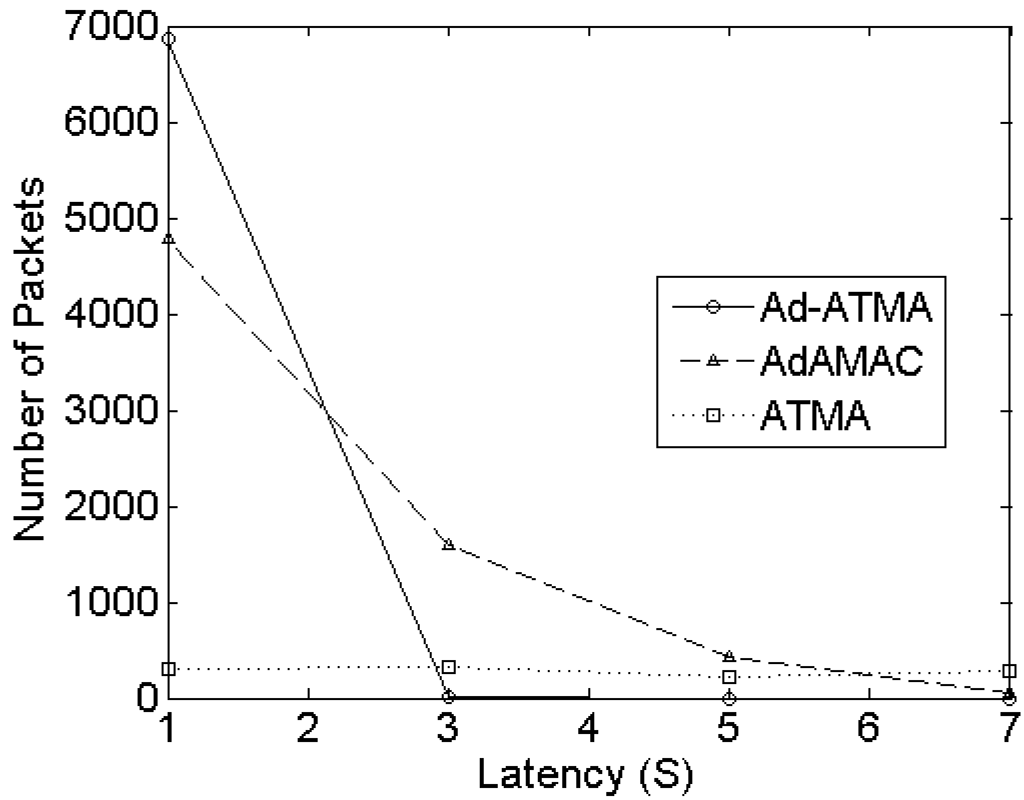


Figure 3.28: Latency distributions of delivered packets (1.69 pkts/sec)

The latency variation of delivered packets is shown in figure 3.28.

3.5.8 Results for Multi-hop networks using random traffic model

Multi-hop network is created by deploying nodes in a 300m x 300m area. the node degree is varied between 2 to 13. Random traffic model is used to generate packets.

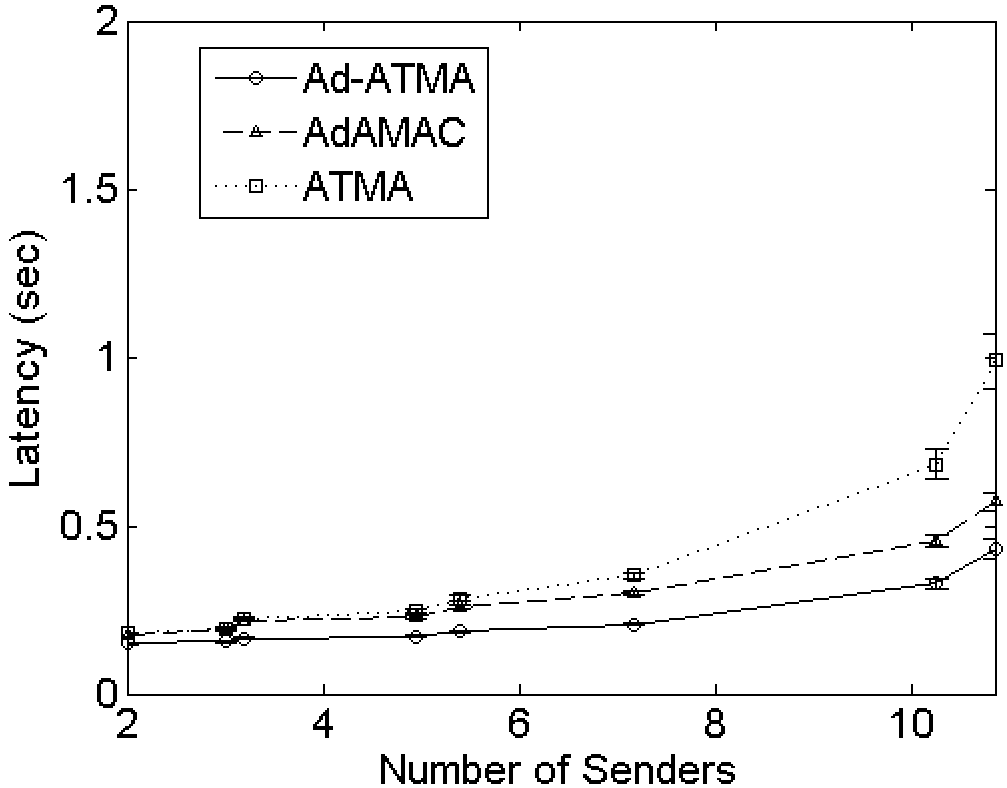


Figure 3.29: Multi-hop: Latency vs Number of senders

In figure 3.29, it is shown that Ad-ATMA has almost 15-30% less latency than AdAMAC and 17-60% less than ATMA. The improvement is less than Single-hop networks due to the collisions of hidden terminal problem.

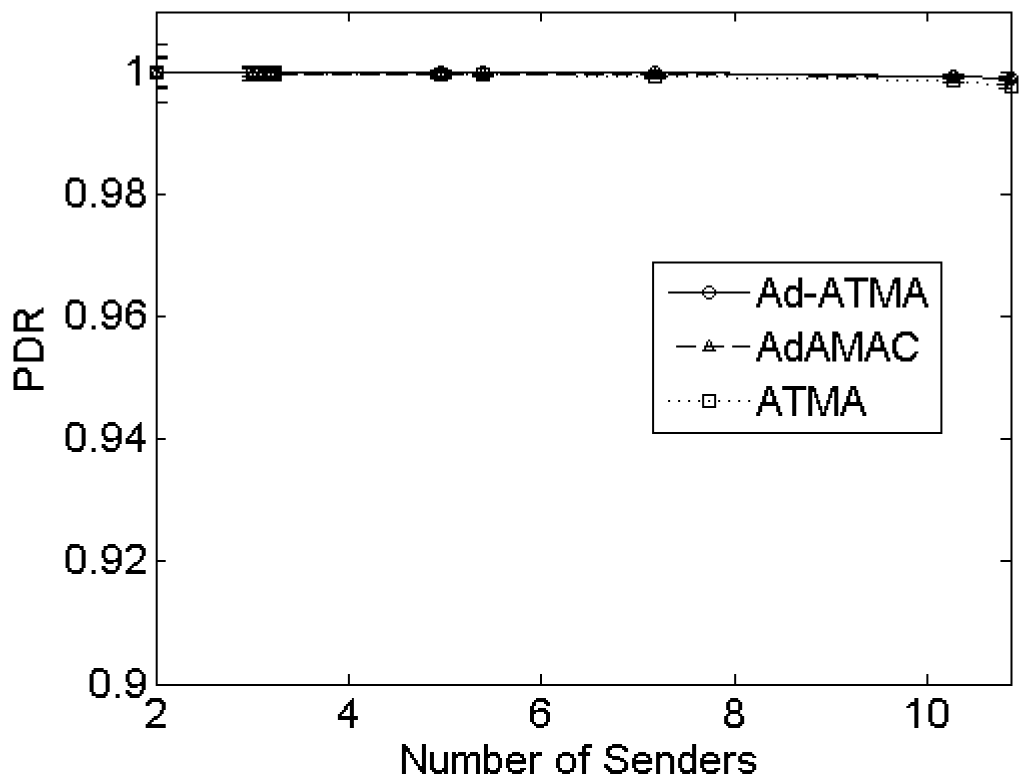


Figure 3.30: Multi-hop: PDR vs Number of senders

The PDR of Ad-ATMA , AdAMAC and Ad-ATMA are shown in figure 3.30.

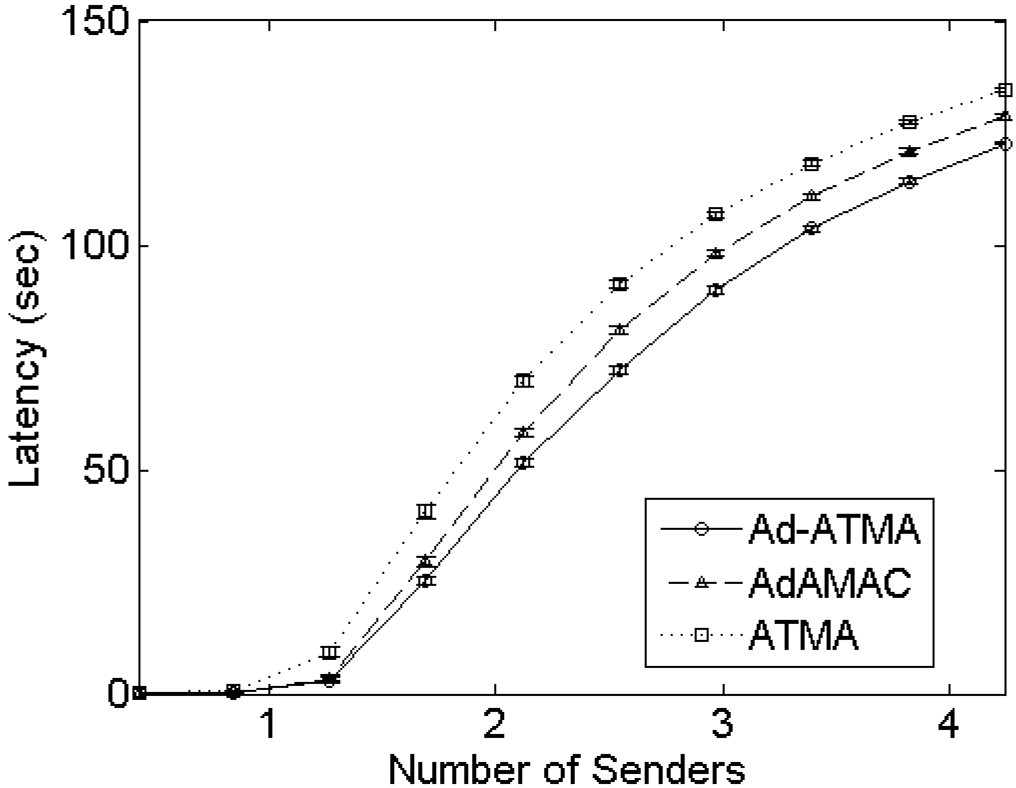


Figure 3.31: Multi-hop: Latency vs Data rate

Now we fix the number of nodes to 10 and vary the data rate from 0.4 pkts/sec to 4.3 pkts/sec. As shown in figure3.31 Ad-ATMA provides upto 32% latency reduction over AdAMAC and 64% latency reduction over ATMA.

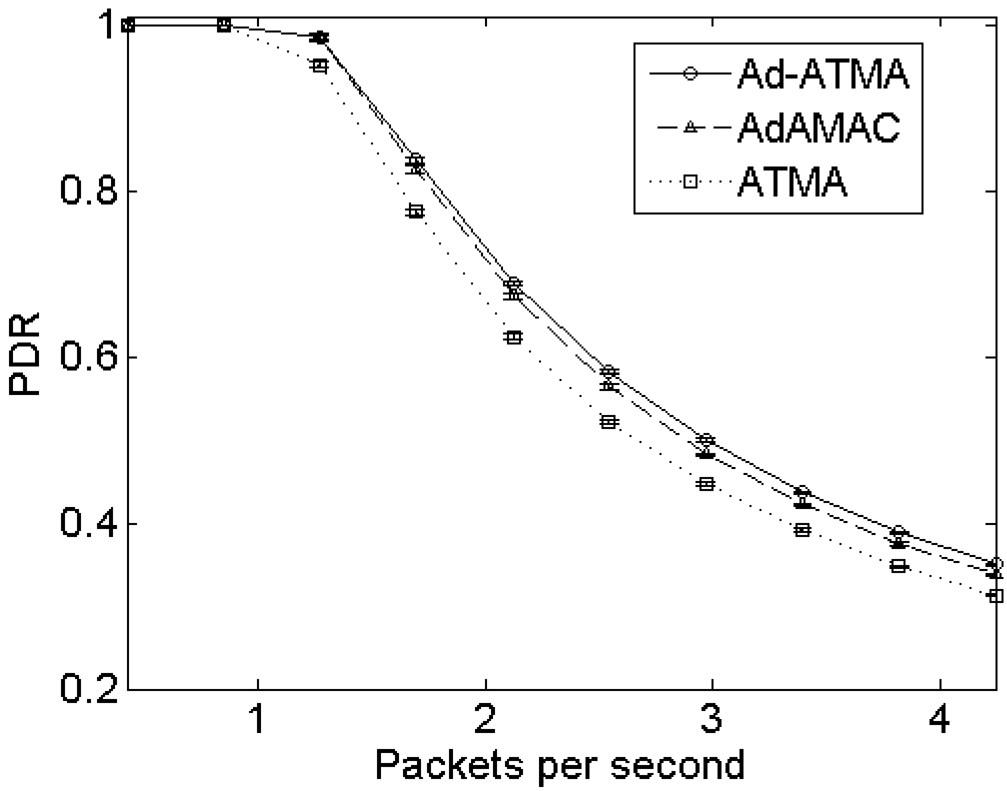


Figure 3.32: Multi-hop: PDR vs Data rate

We also plot the PDR of the three algorithms in figure 3.32. The PDR of Ad-ATMA is slightly higher than AdAMAC and ATMA.

Chapter 4

ResVMAC: A Novel Medium Access Control protocol for Vehicular Ad hoc Networks

In this chapter, we present our novel MAC protocol ResVMAC for VANETs. Before that, we review some existing TDMA-based MAC protocols for VANETs that are related to our work in section 4.1. We describe ResVMAC in section 4.2. In section 4.3 our network model and metrics are described. Finally, performance evaluation results of ResVMAC are presented in section 4.4.

4.1 Related Work

Like WSANs, various MAC protocols have been proposed and implemented for VANETs. We begin by explaining the IEEE 802.11p [57] protocol because this is the common standard for vehicular communication protocols. Later, we focus our attention on some existing TDMA-based protocols.

Frequency Band(GHz)	Channel Name
5.855 – 5.864	Service Channel (Ch 172)
5.865 – 5.874	Service Channel (Ch 174)
5.875 – 5.884	Service Channel (Ch 176)
5.885 – 5.894	Control Channel (Ch 178)
5.895 – 5.904	Service Channel (Ch 180)
5.905 – 5.914	Service Channel (Ch 182)
5.915 – 5.924	Service Channel (Ch 184)

Figure 4.1: US DSRC spectrum assignment

4.1.1 The IEEE 802.11p standard for VANETs

The Federal Communication Commission (FCC) of USA provides 75 MHz of frequency spectrum from 5.850 GHz to 5.925 GHz for Dedicated Short-Range Communications (DSRC) technology in order to facilitate vehicular communications [32]. DSRC is a technology for short to medium range communication that is operable in the 5.9 GHz frequency band in order to provide public safety and also used in private applications [32]. The 75 MHz frequency spectrum is divided into seven channels and the bandwidth of each channel is 10 MHz. One of the seven channels (channel 178) is designated as the control channel (CCH) and is reserved for control information and safety applications. The other six channels are called service channels (SCHs) and support non-safety applications. Figure 4.1 shows the frequency spectrum allocation for CCH and SCHs.

<table>
<tr><td rowspan="2">Application</td><td>Resource Manager</td><td>IEEE 1609.1</td></tr>
<tr><td>Security Services</td><td>IEEE 1609.2</td></tr>
<tr><td colspan="2">Transport (UDP/TCP)</td><td rowspan="2">IEEE 1609.3</td></tr>
<tr><td colspan="2">Networking (IPv6)</td></tr>
<tr><td colspan="2">LLC</td><td>IEEE 802.2</td></tr>
<tr><td colspan="2" rowspan="2">MAC</td><td>IEEE 1609.4 (Upper MAC)</td></tr>
<tr><td>IEEE 802.11P (Lower MAC)</td></tr>
<tr><td colspan="2">PHY</td><td>IEEE 802.11P</td></tr>
</table>

Figure 4.2: Protocol stack of WAVE(adapted from [13])

The IEEE 1609 family of standards, shown in figure 4.2, is an architecture for Wireless Access Vehicular Environment (WAVE) that specifies the protocols for the communication between vehicles and between vehicles and Roadside Units (RSUs) [2]. The IEEE 1609.4 standard of this family mainly focuses on the development of the MAC layer and the Physical layer. According to the specification of the IEEE 1609.4 standard, time is divided into sync intervals. Each sync interval consists of a 50ms CCH interval (CCHI) and a 50 ms SCH interval (SCHI) as shown in figure 4.3. During the CCHI all nodes listen to the CCH for emergency messages and if a node wants a specific service then it tunes to the SCH in which that service is provided during SCHI.

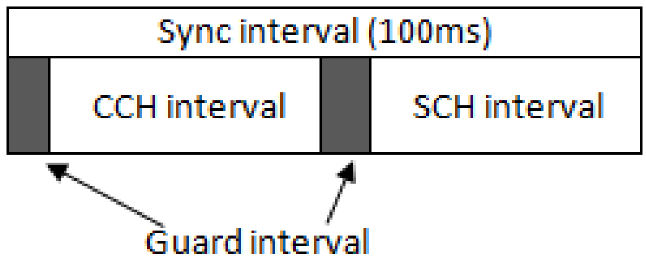


Figure 4.3: CCH interval and SCH interval

The WAVE architecture uses the IEEE 802.11p protocol in the physical and link layers. The IEEE 802.11p protocol is a part of the IEEE 802.11 [30] standard. It relies on the DSRC frequency spectrum (figure 4.1) for communications. In the early versions of IEEE 802.11, all classes of traffic (such as voice, video, and data) are treated with the same priority. The IEEE 802.11e protocol provides different priorities for different traffic. It uses AIFS (Arbitration Inter Frame Space) instead of DIFS. To support Quality of Service (QoS) it uses different AIFSs and *CWs* for different types of traffic. IEEE 802.11p extends IEEE 802.11e in order to support QoS for various classes of traffic that are available in VANETs. Our objective is to develop a MAC protocol for broadcast communications in VANETs without providing QoS guarantees. QoS guarantees are more relevant to unicast communication, so we will not discuss further about how to provide QoS guarantees in VANETs.

One of the limitations of the IEEE 802.11p is that it is not suitable for broadcast communications. The RTS/CTS mechanism is not applicable when vehicles broadcast packets because if a vehicle broadcasts a RTS packet then all the active one-hop neighbours will broadcast their CTS packets. As a result the channel will be accessed by multiple nodes. Moreover, the hidden terminal problem cannot be alleviated without the RTS/CTS mechanism in the IEEE 802.11p. Another problem is that broadcast communications do not use ACKs. In unicast communications, the receiver replies with an ACK if it successfully receives a packet from the sender. If the sender does not receive any ACK in a certain amount of time it doubles the *CW* and retransmits the packet again. Since packets are not retransmitted in broadcast communications, the *CW* remains fixed all the time. Having a fixed CW increases the probability that two nodes pick the same random value and therefore collide when there is more traffic in the network.

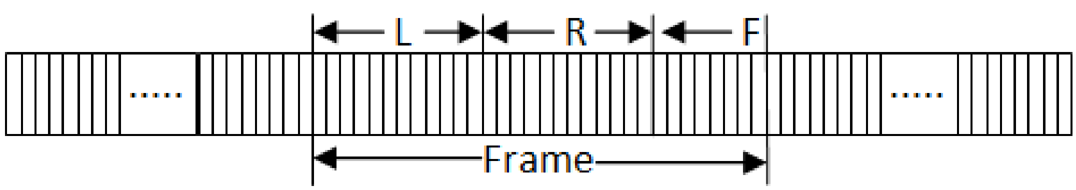


Figure 4.4: L, R, and F sets in a VeMAC Frame (adapted from [40])

4.1.2 VeMAC

VeMAC [40] is a multichannel TDMA based MAC protocol which reduces the probability of collisions by using different sets of time slots for the vehicles moving in opposite directions and RSUs. Similar to the WAVE, it has one control channel (CCH) and multiple service channels (SCHs). Each vehicle uses two transceivers. Transceiver 1 is for CCH and transceiver 2 is tuned to one of the SCHs. Each vehicle is considered to be equipped with a GPS to find its current position and direction. Time is partitioned into frames, and frames are divided into many slots. Each frame has three sets of slots L,R, and F (figure 4.4). The sets L and R are assigned to the vehicles of opposite directions and the set F is used for RSUs. In this protocol, each vehicle must access a CCH slot in each frame. The CCH slot is used to send

- its SCH slot number in which it provides the non-safety services,
- its direction and position, and
- the CCH time slots of all one-hop neighbours.

Thus each vehicle gets an equal chance of accessing the SCHs. Each vehicle also knows about the slots used by its one-hop neighbours and two-hop neighbours by analysing the packets received from its one-hop neighbours and thus avoids collisions due to the hidden terminal problem.

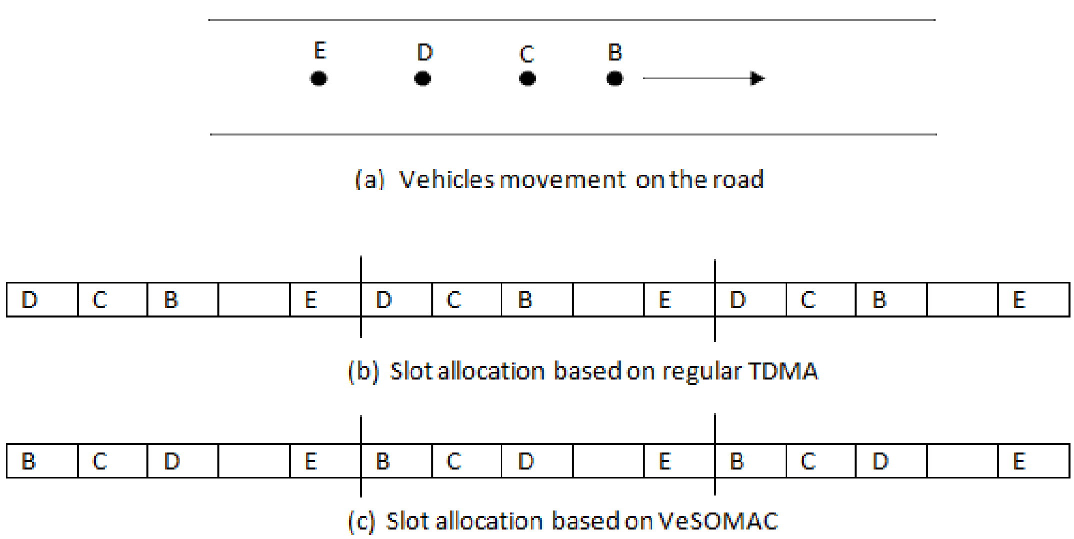


Figure 4.5: Slot allocation using VeSOMAC (adapted from [62])

4.1.3 VeSOMAC

Vehicular Self-Organizing MAC (VeSOMAC) [62] is a novel MAC protocol for vehicle-to-vehicle communications that uses the DSRC standard. It provides quick message delivery with deterministic delay bounds. Each vehicle gets its slot based on its location. Figure 4.5 shows the slot assignments of 4 vehicles labelled as B, C, D, and E. As vehicle B is ahead of all other vehicles, B accesses the channel first. After that C gets access to the channel, followed by D and E, in that order. If an accident occurs in front of B then this information can go to E within one frame while a regular TDMA based technique would take three frames. Each packet has two parts: a header followed by a data part. Each vehicle puts a bitmap vector in the header which is used to exchange the relative transmission times of all one-hop neighbours. In order to send up-to-date information each vehicle sends a packet in each frame. So a vehicle learns about its one-hop and two-hop neighbours' transmission times by receiving bitmaps from its one-hop neighbours. This helps to avoid collisions and also alleviates the hidden terminal problem.

4.1.4 TC-MAC

The IEEE 1609 architecture uses a single DSRC radio in order to support both safety and non-safety applications. The safety application packets must be broadcast by each vehicle at least once in every 100 msec. So, each vehicle contends in the CCHI in order to broadcast safety application packet. If the number of vehicles increases then some vehicles cannot get enough time to broadcast the safety application packets in every 100 msec. So, the duration of CCHI must be increased and the duration of SCHI must be reduced in order to retain the 100 msec requirement of safety application packet broadcast [59]. A Cluster-based TDMA scheduling MAC (TC-MAC) [12] is proposed to overcome this problem. It uses a method that allows vehicles to broadcast both safety and non-safety application packets without compromising one of them. In TC-MAC, a centralized approach is used to manage the cluster, and a lightweight

TDMA-based slot reservation technique is used to reserve slots.

4.1.5 RR-ALOHA

RR-ALOHA [18] is a distributed protocol that divides time into fixed length virtual frames. Each virtual frame has a fixed number of slots $[1 \ldots N]$ known as Basic Channels (BCHs). A virtual frame consists of previous N observed BCHs. A vehicle must reserve a BCH in order to access the wireless channel. At first, a vehicle monitors the channel for one virtual frame if it wants to send data. Then it contends for an unassigned BCH by broadcasting a FI (Frame Information) packet in that BCH. The FI packet contains the request for the reservation and the status of the perceived BCHs of the previous frame. Thus a node would know which BCH is reserved by which neighbour and which BCH is free. Each FI packet also includes some other information which will be described later. The FI packet that is used for reservation is known as a REQ packet.

Now let us consider that a vehicle V wants to reserve a free BCH j. So it broadcasts its REQ packet in j after observing the channel for one virtual frame. Then it waits for the FI packets from its active one-hop neighbours. The active one-hop neighbours broadcast their views of the previous virtual frame through their FI packets in their reserved BCHs. The one-hop neighbours that hear the REQ packet of V properly, assigns j to V in their FI packets. But they make j free in their FI packets if they hear collisions in j. If j is assigned to V by all the active one-hop neighbours in their FI packets then V starts to access the channel in j from the upcoming next frame and continues to broadcast its FI packets in j until a collision occurs. So Each vehicle knows about its two-hop neighbours by receiving the FI packets from its active one-hop neighbours and this helps to avoid the hidden terminal problem.

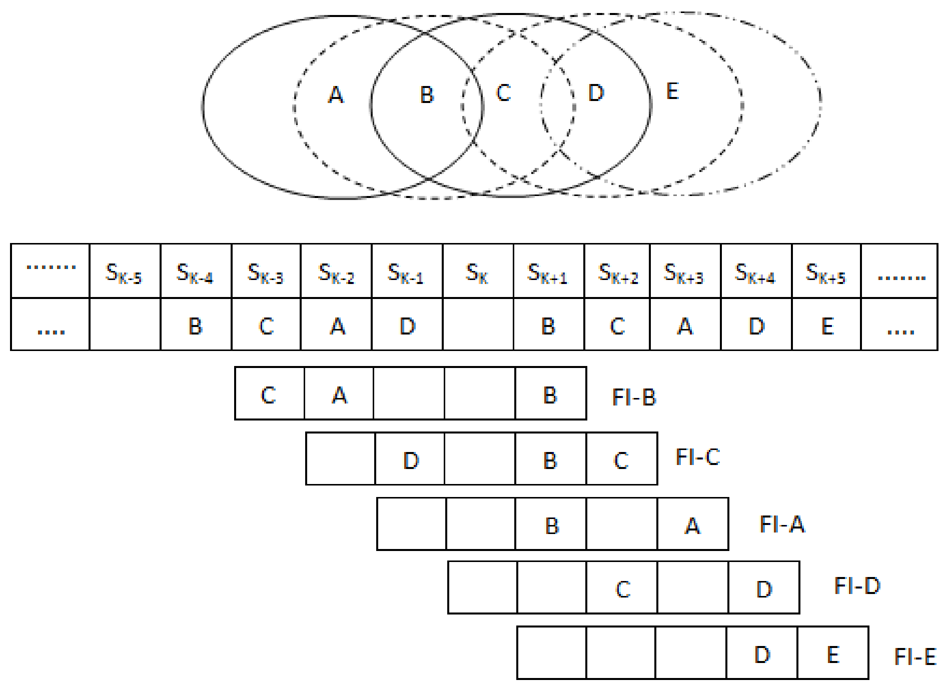


Figure 4.6: Propagation of FI packets

Figure 4.6 shows the exchange of FI packets among vehicles. In this figure, five vehicles are used and they are labelled as A, B, C, D, and E. The circles represent the radio range of each vehicle. $[S_{k-5} \ldots S_{k+5}]$ represent the BCHs from time $k-5$ to $k+5$ respectively. Each virtual frame consists of five BCHs. For the sake of simplicity let us assume that all vehicles other than E have already reserved their BCHs. E has started monitoring the channel from slot S_k. B has two one-hop neighbours A and C so B includes A and C in its FI packet (FI-B) and broadcasts the FI-B in its own BCH S_{k+1}. C has B and D as one-hop neighbours. So C includes B and D in FI-C and broadcasts it in S_{k+2}. A has one one-hop neighbour B and B is included in FI-A that is broadcast in BCH S_{k+3}. D only knows about C because E has not sent any REQ packet yet. As E sensed the channel for one virtual frame so it received the FI-D at S_{k+4} and knows that C is using BCH S_{k+2}. Thus E does not reserve S_{k+2} to avoid the collision with C. It also knows that other three BCHs (S_k, S_{k+1}, S_{k+3}) of the previous frame were free and it can choose any one of the three free BCHs. So it broadcasts FI-E in S_{k+5}. D receives FI-E properly and includes E in its FI packet. So C also knows about E after receiving the FI-D.

4.1.5.1 Detailed Structure of the FI packets

Each vehicle that has reserved a BCH sends its FI packet along with a payload. The FI packet contains as many fields as the number of BCHs in a virtual frame. Each field includes the following information which is shown in figure 4.7

- STI (Source Temporary Identifier) : The STI is used to uniquely identify the vehicle that has reserved this BCH. It is 8 bit long.
- PSF (Priority Status Field): The 2 bit long PSF sets the priority of the transmitted data.
- BUSY : If this BCH is free then BUSY bit is 0 otherwise 1.
- PTP : This bit is used to set up point-to-point communication.

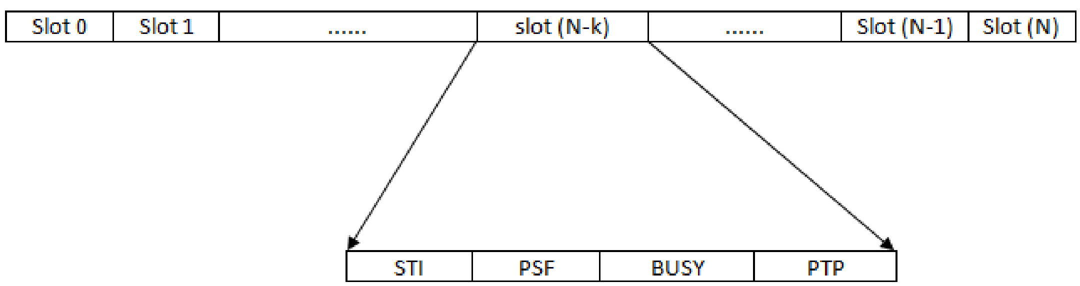


Figure 4.7: FI packet structure (RR-ALOHA)

4.1.6 Cooperative ADHOC MAC (CAH-MAC)

CAH-MAC [15] improves the reliability of RR-ALOHA by introducing cooperative communication. A packet may fail to reach its destination due to poor channel conditions. In CAH-MAC, one-hop neighbours of a sender cooperate with the sender by retransmitting the packet to the receiver. As shown in figure 4.8, CAH-MAC uses the same packet fields like RR-ALOHA and in addition, a cooperation header field is introduced. A node C can cooperate only if the direct transmission from node A to node B fails, C receives the packet from A successfully, B is reachable from C and there is an available time slot for B to retransmit the packet to C. If the node C decides to cooperate then it sends the following information to A in its reserved time slot.

- Its intention to cooperate.
- The time when transmission failure has occurred.
- The available time slot when it wants to retransmit the packet to the receiver.

The one-hop neighbours of C that also want to cooperate do not do so after receiving the packet from C.

If more than one node that are not in each other's radio range want to retransmit a packet in the same time slot, then the destination node D selects the node E which sent its intention to cooperate first. D sends an A-ACK at the beginning of the time slot selected by the nodes containing E's ID. So nodes other than E do not transmit. Thus collision due to hidden nodes is avoided.

Frame Information (FI)	**Cooperation Header (COH)**	Payload Data	CRC

Figure 4.8: CAHMAC packet structure (adapted from [15])

4.1.7 RR-ALOHA+

RR-ALOHA+ [21] improves the performance of RR-ALOHA by overcoming some of its limitations. In RR-ALOHA, the BCH reservation of a sender becomes invalid if at least one active one-hop neighbour of the sender says that the BCH is free. This can happen due to a collision or because the one-hop neighbour has just entered into the sender's radio range and has that BCH unassigned. Intuitively, the reservation should not be cancelled for a newly entered vehicle, and RR-ALOHA+ prevents this from happening. Moreover, RR-ALOHA+ proposes that the status of each BCH should be refreshed after a certain amount of time to avoid the propagation of outdated information. In RR-ALOHA, the BUSY bit is used to identify whether a BCH is busy or free. A BCH may be free due to a collision or because no vehicle has reserved it. Using 1 bit it is not possible to identify all the three states. So, RR-ALOHA+ uses an extra bit known as COL bit. Now all the states of a BCH can be represented correctly using BUSY bit and COL bit which is shown in table 4.1.

Table 4.1: Status of BCH in RR-ALOHA+

BUSY	COL	STATUS
0	0	free
1	0	busy
0	1	collision
1	1	unused

4.1.8 MS-ALOHA

As mentioned, RR-ALOHA+ uses the BUSY and COL bits to identify the states of a BCH. From the table 4.1, it can be seen that the last combination is unused. In MS-ALOHA [52], a vehicle sets both the BUSY bit and COL bit of a BCH to 1 if this BCH is used by one of its two-hop neighbours. So after receiving a FI packet, if a vehicle A that wants to reserve a free BCH, finds that the BUSY and COL bits are set to 1 then it knows that this BCH is being used by a vehicle B that is three hops away. A does not reserve this BCH to avoid collisions that may occur if A and B move towards each other. If a vehicle receives $[1, 0]$ and $[1, 1]$ for the same BCH, it means that two vehicles are using the same BCH of which one vehicle is two hops away and another one is three hops away. So it considers $[1, 0]$ to be the status of that BCH.

4.1.9 MARR-ALOHA

In RR-ALOHA, a vehicle broadcasts a packet (FI or REQ) at the beginning of its reserved BCH. So a collision occurs if more than one vehicle broadcast their packets in the same BCH. MARR-ALOHA [35] uses CSMA and a backoff algorithm within a BCH to reduce the possibility of collision of REQ packets and FI packets. The FI packet is not transmitted at the beginning of the BCH; rather the transmission starts after FI Backoff Timer (FIBT) is over. The FIBT is a backoff timer whose value depends on the following two parameters:

- Backoff_Unit: The time unit of the FIBT.
- Max_Backoff: The maximum number of Backoff_ Units.

So the value of the FIBT can be expressed as *Backoff_Unit x K* where *K* is a random positive number less than the Max_Backoff.

If a vehicle senses the transmission of a packet before the expiration of its FIBT then it gives up its attempt and tries to reserve another free BCH. The backoff timer value for the REQ packet is larger than the FIBT and a REQ packet does not interfere with the transmission of the FI packet. To avoid the collision between REQ packets, a shorter REQ packet is used. A REQ packet includes the following:

- the STI (Source Temporary Identifier) of the vehicle.
- the priority of the data.
- BUSY bits for all BCHs of the previous virtual frame.

If two REQ packets collide at a common neighbour, then the common neighbour indicates this in its next FI packet. If a common neighbour gets more than one REQ packets for a specific BCH, it selects one vehicle for that BCH using an arbitration mechanism and indicates this choice in its FI packet. So after receiving the FI packet from the common neighbour other candidate vehicles quit their intention to send their FI packets in that BCH.

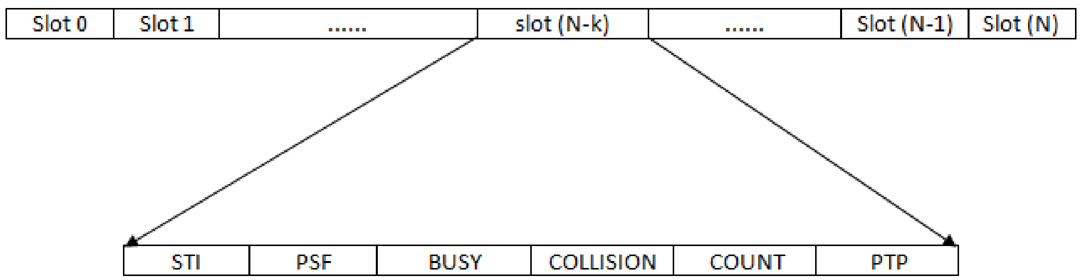


Figure 4.9: FI packet structure (MARR-ALOHA)

A vehicle that is three hops away can move into the one hop neighbourhood very quickly if it moves with high speed. So it is important to get an idea about the status of each BCH in the three hop neighbourhood. So, an extra field COUNT (8 bits) is added to each BCH in the frame structure of MARR-ALOHA, as shown in figure 4.9. The COUNT field of a BCH indicates the number of two-hop neighbours of a vehicle that are using this BCH. The number of three-hop neighbours of a vehicle that are using this BCH is the summation of COUNTs received from all one-hop neighbours of that vehicle. Ideally, this value should be small to reduce the possibility of collisions. Like MS-ALOHA, it also uses COL and BUSY bits in order to describe the condition of each BCH.

The above mentioned distributed TDMA-based protocols attempt to reserve packet transmission slots quickly. Notable among these are MARR-ALOHA and RR-ALOHA. In MARR-ALOHA and RR-ALOHA, a vehicle cannot reserve an available BCH immediately after sending its REQ packet. In each protocol, the vehicle waits one frame to get BCHs from all the active one-hop neighbours, and checks that the neighbours acknowledge its BCH reservation attempt. It starts sending data from the next frame. Our proposed ResVMAC algorithm outperforms MARR-ALOHA and RR-ALOHA by reserving BCHs more quickly which is described in the following section.

4.2 Algorithm ResVMAC: Our Proposed MAC protocol

In VANETs, network topology can change very quickly due to the frequent and fast movement of vehicles. So it is better to reserve BCHs as quickly as possible. In our proposed distributed TDMA-based MAC protocol, ResVMAC, we use a faster reservation scheme to reserve the BCH more quickly than MARR-ALOHA and RR-ALOHA. ResVMAC is described next.

4.2.1 Frame Structure

ResVMAC uses the same frame structure like RR-ALOHA and MARR-ALOHA as shown in figure 4.10. It divides time into fixed length virtual frames. Each virtual

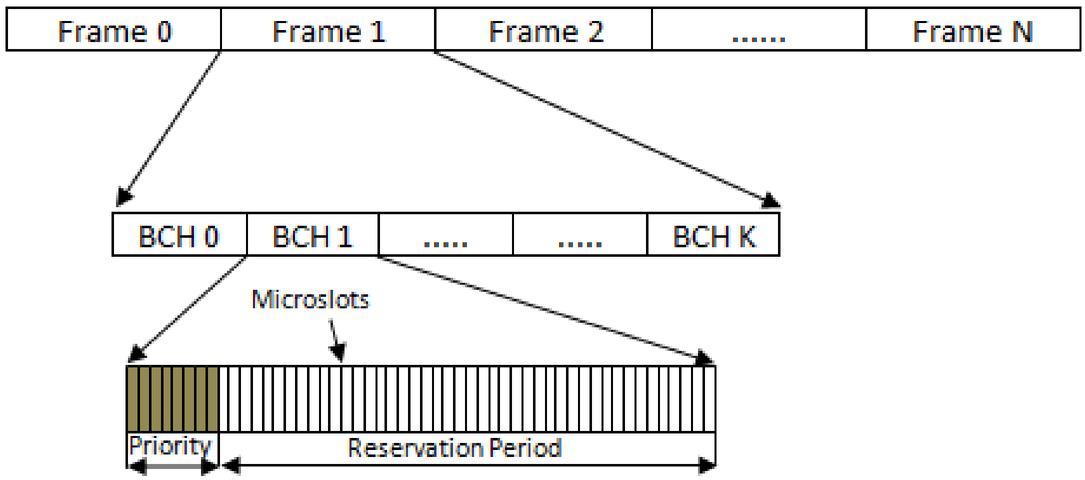


Figure 4.10: Frame and Basic Channel structure of ResVMAC

frame consists of a fixed number of BCHs. A vehicle that wants to send data must reserve a free BCH first.

4.2.2 REQ packet and FI packet

Similar to RR-ALOHA and MARR-ALOHA, ResVMAC uses a REQ packet to reserve an available BCH. The REQ packet of ResVMAC is shorter than that of RR-ALOHA and MARR-ALOHA, because it contains only the sender address and the free BCH number that the sender wants to reserve. After reserving a BCH, a sender periodically sends FI packets in its reserved BCH until a collision occurs or it releases the BCH voluntarily. The FI packet contains the status of the slots of the previous frame. Similar to MS-ALOHA and RR-ALOHA, ResVMAC also uses COL and BUSY bits in each

FI packet to represent the status of each BCH correctly. Moreover, the COUNT field of MARR-ALOHA is inherited to estimate the number of vehicles using each BCH in three hop neighbourhood.

4.2.3 BCH Structure

If a vehicle wants to reserve an available BCH then it estimates the total number of available BCHs in the upcoming frame by monitoring the previous frame. The first free upcoming BCH is known as the contention BCH for this vehicle. The contention BCH has two parts– a Priority period followed by a Reservation period.

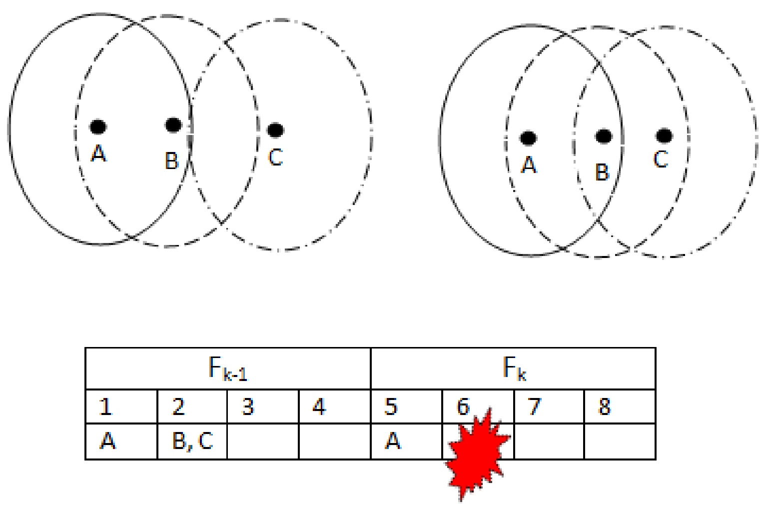
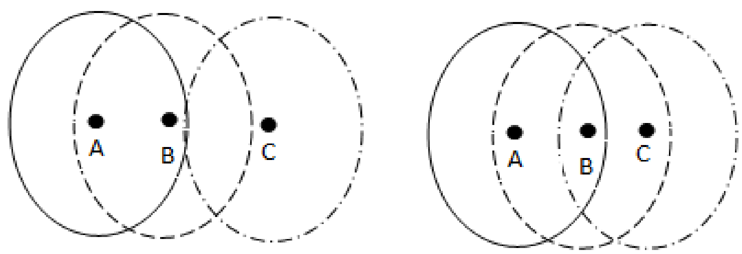


Figure 4.11: Collision between the BCH packets (B and C have reserved the same BCH and they enter each other's radio range at the beginning of F_{k})

The Priority period has the same function as the Max_Backoff of MARR-ALOHA. It is used to prioritize the vehicles that already have BCHs over the vehicles that want to reserve one. Moreover, it reduces the possibility of collision between vehicles that have reserved the same BCH. Vehicles that wish to send data contend in the Reservation period to reserve an available BCH. Like the Contention BCH, the reserved BCH also starts with the Priority period.



F_{k-1}				F_k			
1	2	3	4	5	6	7	8
A	C			A			

Figure 4.12: Collision between the BCH packet and the REQ packet (B and C enter each other's radio range at the beginning of F_k and B wants to reserve the BCH which is occupied by C)

4.2.4 Priority period

If a vehicle B sends its FI packet in its reserved BCH or REQ packet in an available BCH immediately after entering into the radio range of another vehicle C that has already reserved that BCH then the FI packets of B and C or the FI packet of C and the REQ packet of B collide with each other as shown in figure 4.11 and figure 4.12.

To avoid the collision between the FI packet and the REQ packet, vehicles that have reserved BCHs start transmitting their FI packets in the Priority period and the vehicles that want to reserve available BCHs send their REQ packets in the Reservation period. So, if a vehicle B wants to reserve a BCH that is already occupied by another vehicle C then B postpones its attempt after listening the FI packet of C. In addition, to reduce the possibility of collision between the FI packets the Priority period is split into m microslots. Each vehicle that wants to send its FI packet randomly selects a microslot number from $[1 \ldots m]$ in the Priority period and initializes a countdown timer with that number. The vehicle whose timer reaches to zero first start transmitting its FI packet and all other vehicles that want to send their FI packets in the same BCH cancel their reservation and attempt to reserve available BCHs in the upcoming Contention BCH.

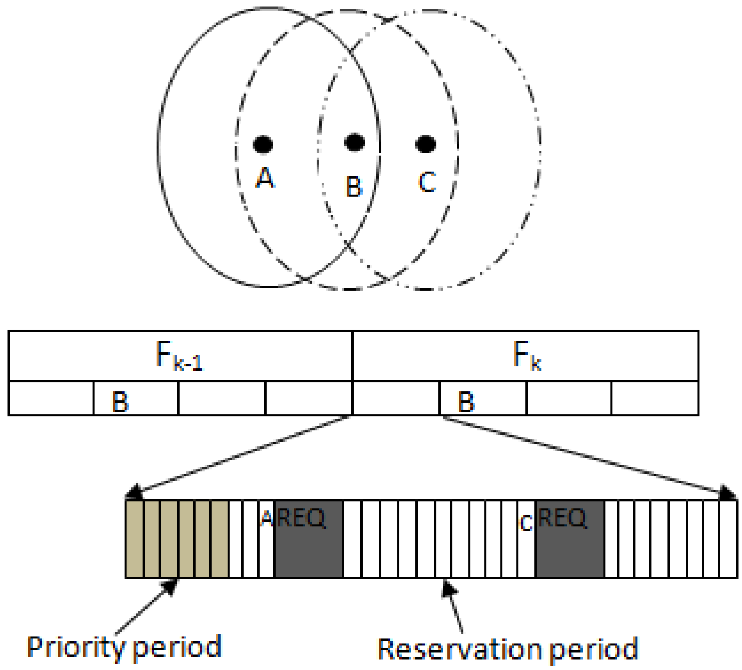


Figure 4.13: Reservation of available BCH

4.2.5 Reservation period

The Reservation period is divided into n microslots. If a vehicle wants to reserve an available BCH, it randomly selects a microslot number in $[1 \ldots n]$ in the Reservation period and uses it as the initial value of a countdown timer. When the value of the timer becomes zero the vehicle broadcasts its REQ packet. The free BCH number indicated in the REQ packet is reserved by the sender if all the active one-hop neighbours receive the REQ packet as shown in figure 4.13. If any collision occurs at any active one-hop neighbour during this transmission, the neighbour broadcasts a NACK which is shown in figure 4.14. Moreover, if the sender wants to reserve a BCH that is already reserved by one of its two-hop neighbours then the common neighbour broadcasts a NACK after receiving the REQ packet. If the sender receives NACK from at least one of its active one-hop neighbours then it contends again in the upcoming contention BCH for an available BCH. Other one-hop neighbours that want to reserve an available BCH freeze their timers during this whole transmission. They resume their timers when the transmission is over and try to reserve one of the remaining available BCHs if there is enough time left.

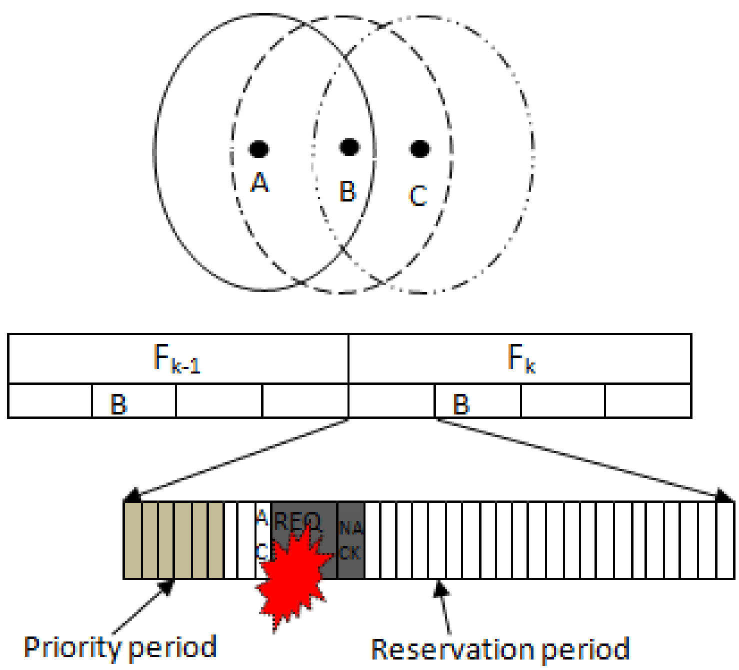


Figure 4.14: Negative Acknowledgement from node B

4.2.6 Broadcasting data

The sender starts broadcasting data in its reserved BCH if it does not receive any NACK from its active one-hop neighbours. If more than one receivers reply with NACKs then a collision occurs and the channel remains busy. The sender interprets this as a NACK. It is worth pointing out that sensing a busy channel in wireless networks has been shown to be feasible in many papers, e.g. [47, 42, 63].

4.2.7 Alleviating the Hidden Terminal Problem

In ResVMAC, collisions may occur during the transmission of REQ packets due to the hidden terminal problem. However, since the REQ packet is very small in length, the probability of such collisions happening is low.

4.2.8 Significance of Negative Acknowledgement

In VANETs, vehicles can move very fast so it is necessary to establish communication link with neighbouring vehicles as quick as possible. The NACK helps the colliding senders to get the collision information immediately after the collision. So the colliding senders can contend for the remaining available BCHs very quickly. Moreover, if a sender sends a REQ packet for a BCH that is already reserved by another vehicle then the common neighbour sends a NACK immediately after getting the REQ packet. Thus the sender knows that the BCH is already reserved by one of its two-hop neighbours and postpones its attempt. As a result the collision is prevented.

4.3 Experimental setup for comparing TDMA-based protocols

In this section, we describe our model and metrics that we have used to evaluate the performance of ResVMAC, MARR-ALOHA, and RR-ALOHA.

4.3.1 Model and metrics

In our model, we assume each vehicle can move with a fixed velocity along a circular road. The circle has a radius of 1 km. The road has two lanes with 5 m width as shown in figure 4.15. All the vehicles in a lane move in the same direction and vehicles in the other lane move in the opposite direction. The duration of each BCH is very small, so our simulations assume that the vehicles change their positions after each frame rather than after each BCH. This road model is used so that the performance of our proposed protocol can be analysed when vehicles of one lane speed past vehicles in the other lane. The vehicles are placed on the road using a Poisson distribution. For simplicity, we do not consider Minimum Safety Distance (MSD) in our model. Thus we ignore physical collisions among vehicles as they move. We use a simple packet generation model that assumes that each vehicle generates one packet in each frame.

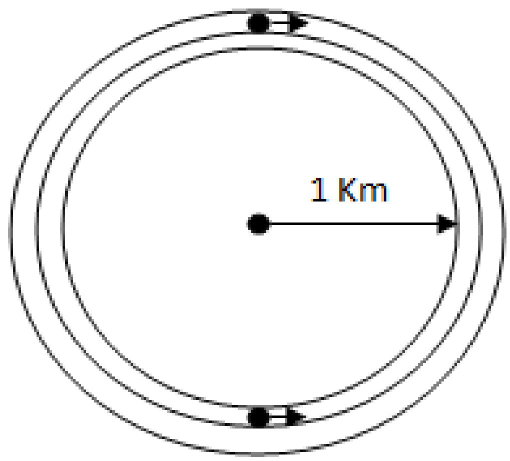


Figure 4.15: A circular road with two lanes

We use average time responsiveness, average number of packets delivered and the average number of collisions to evaluate the performance of our algorithm. The average time responsiveness is defined as the average amount of time each vehicle takes to reserve a BCH. To evaluate the time responsiveness we consider the vehicles motionless and examine how quickly the vehicles reserve their BCHs. We calculate the average number of packets delivered by considering that each vehicle moves with a fixed velocity on the road for a certain amount of time. In both cases, we calculate the average number of collisions.

4.4 Results and Comparison

We implemented RR-ALOHA, MARR-ALOHA, and ResVMAC in Matlab R2013a. The transmission rate is assumed to be 10 Mbps. This is consistent with the hardware available currently. We averaged the measurements over 50 runs. The duration of a BCH is set to 0.5 ms. For ResVMAC, the duration of one microslot is 0.01 ms, and the duration of a REQ packet and a NACK packet are 0.03 ms and 0.01 ms respectively. The duration of a REQ packet for MARR-ALOHA is set at 0.06 ms and for RR-ALOHA it is set at 0.5 ms. The Priority period and the Reservation period of ResVMAC are set to 0.05 ms and 0.45 ms respectively. The Max_Backoff and the Backoff_Unit of MARR-ALOHA are considered 5 and 0.01 ms respectively.

4.4.1 Average Time Responsiveness and Average Number of Collisions

We simulated 200 vehicles on the circular roadway. We evaluated the average time responsiveness of RR-ALOHA, MARR-ALOHA, and ResVMAC for 35, 45, and 55 BCHs per frame when the radio range is 200 m. In a second set of simulations, we used 55, 65, and 75 BCHs for a radio range of 300 m.

Figure 4.16 to figure 4.18 show the number of frames needed to reserve a BCH by each vehicle when the radio range is 200 m.

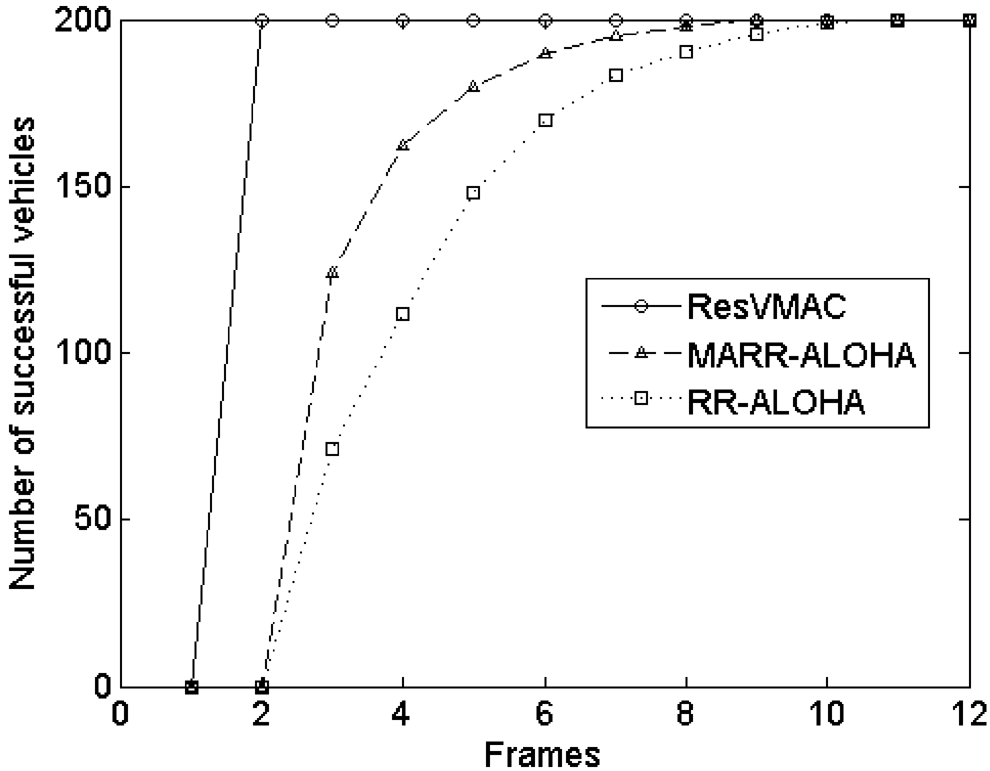


Figure 4.16: Average Time Responsiveness for 35 BCHs/ frame (radio range= 200m)

Figure 4.16 shows the time responsiveness when the BCH per frame is 35. ResVMAC takes only one frame when MARR-ALOHA and RR-ALOHA takes 8 frames and 11 frames respectively. Note however, that most of the vehicles reserve their BCHs within 6 frames in MARR-ALOHA and 8 frames in RR-ALOHA.

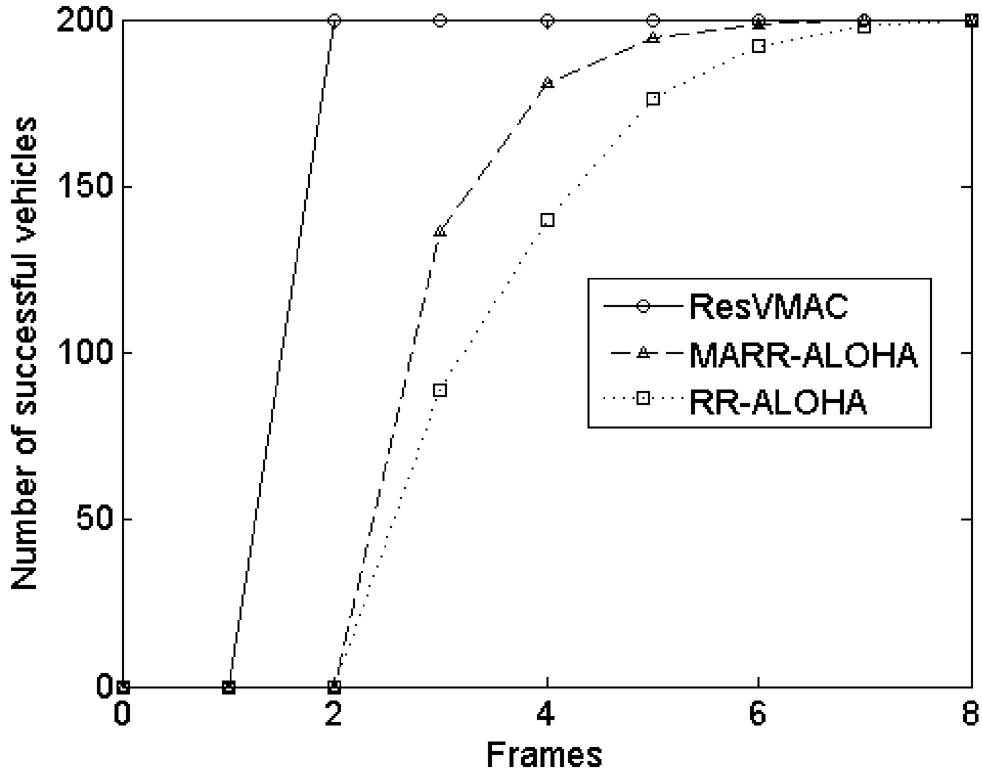


Figure 4.17: Average Time Responsiveness for 45 BCHs/frame (radio range= 200m)

The time responsiveness for the 45 BCH per frame is shown in figure 4.17. All the vehicles reserve their BCHs in the first frame in ResVMAC. Seven frames is needed for all nodes to reserve the BCHs in MARR-ALOHA, and eight frames are needed for the same in RR-ALOHA.

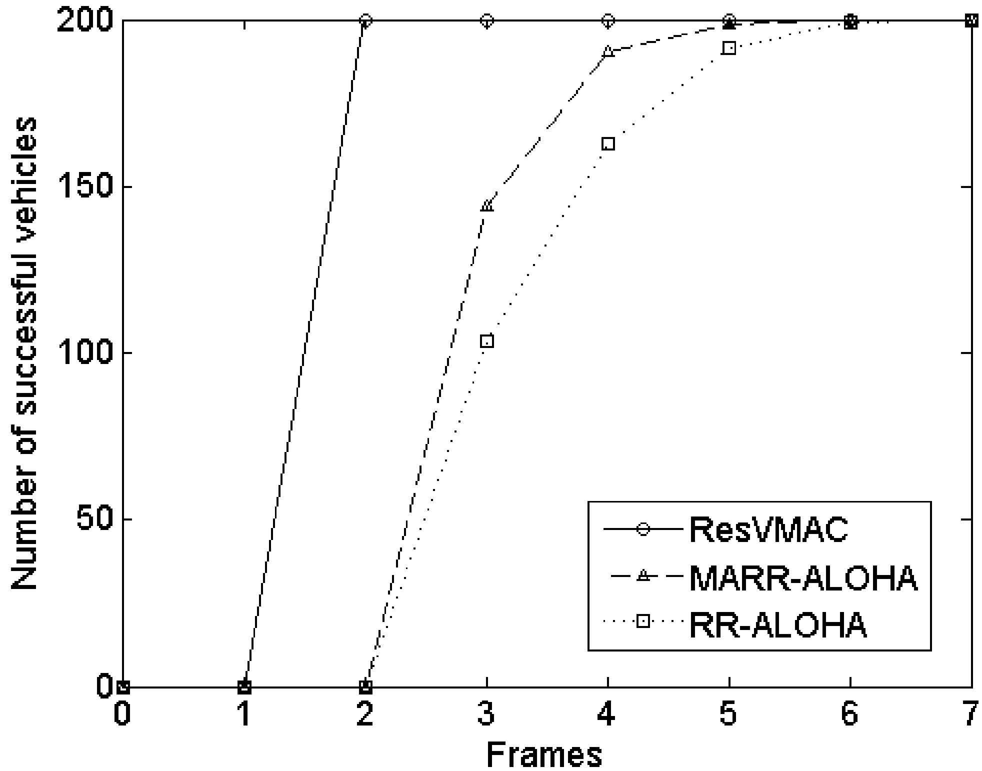


Figure 4.18: Average Time Responsiveness for 55 BCHs/frame (radio range= 200m)

In figure 4.18, we see that for 55 BCHs per frame, all vehicles get their BCHs in one frame using ResVMAC but need six frames in MARR-ALOHA and seven frames for RR-ALOHA.

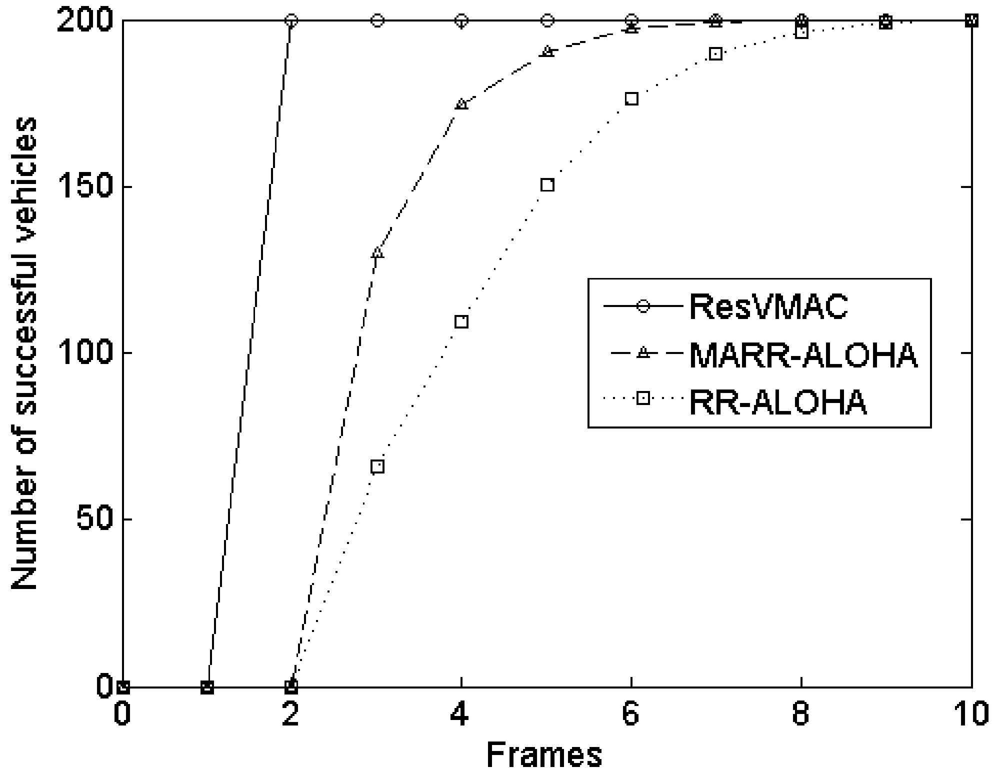


Figure 4.19: Average Time Responsiveness for 55 BCHs/ frame (radio range= 300m)

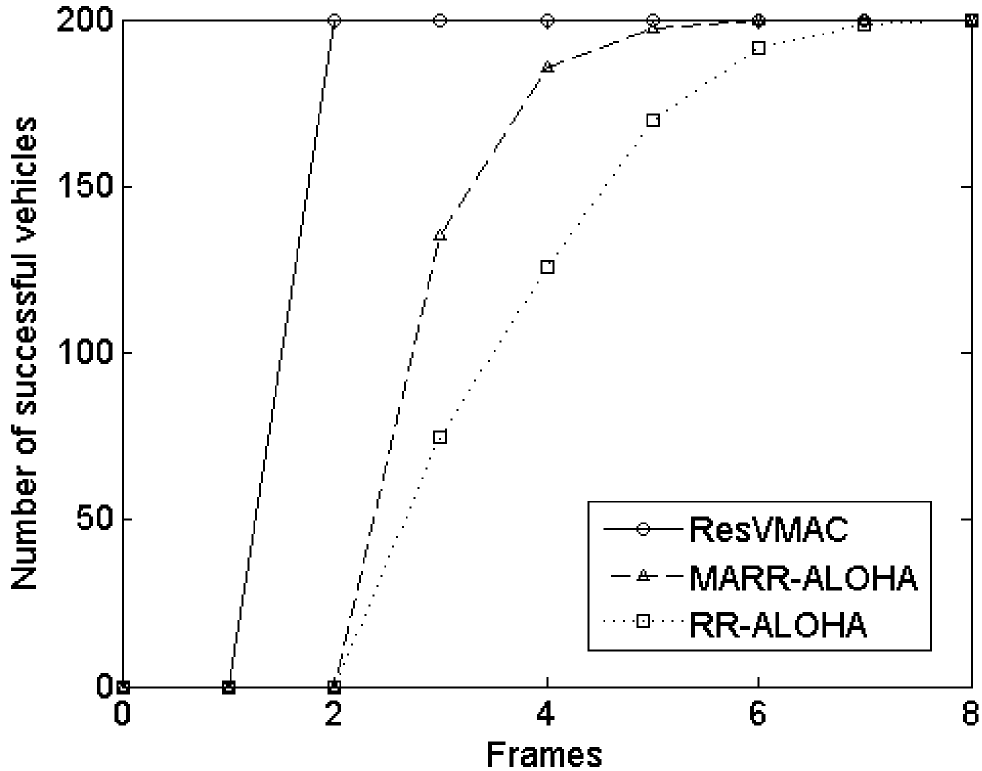


Figure 4.20: Average Time Responsiveness for 65 BCHs/frame (radio range= 300m)

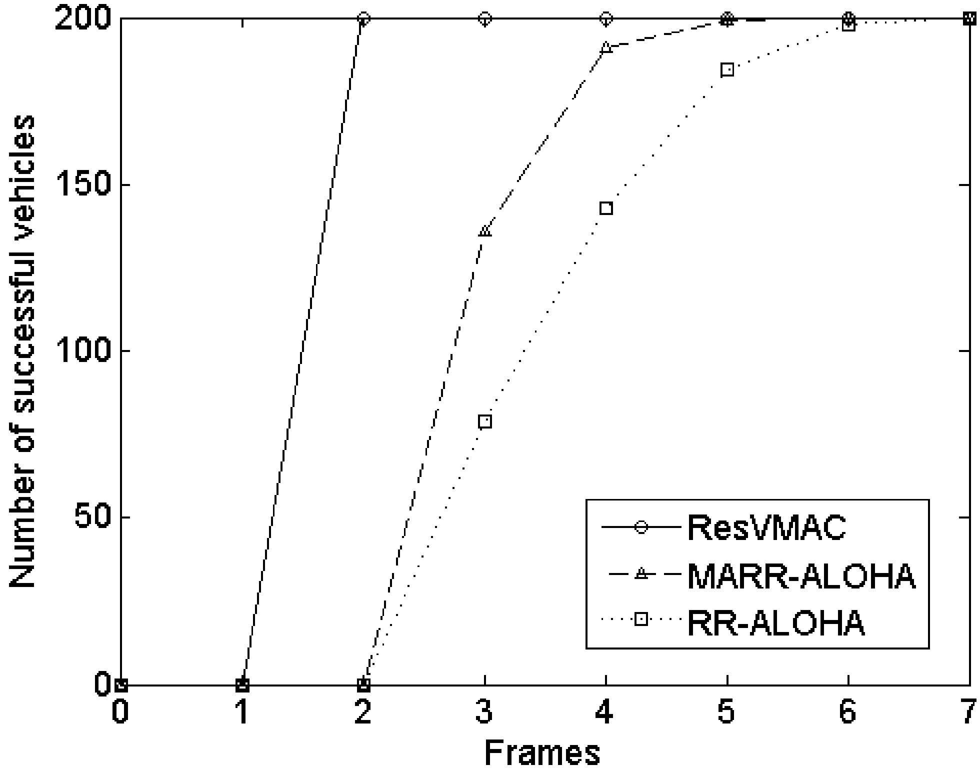


Figure 4.21: Average Time Responsiveness for 75 BCHs/frame (radio range= 300m)

A similar set of results are obtained when the radio range of each vehicle is set to 300 m. Figure 4.19 through figure 4.21 compares the average time responsiveness of ResVMAC to RR-ALOHA and MARR-ALOHA for 55, 65, and 75 BCHs per frame. We observe again that ResVMAC has lower Average Time Responsiveness than RR-ALOHA and MARR-ALOHA.

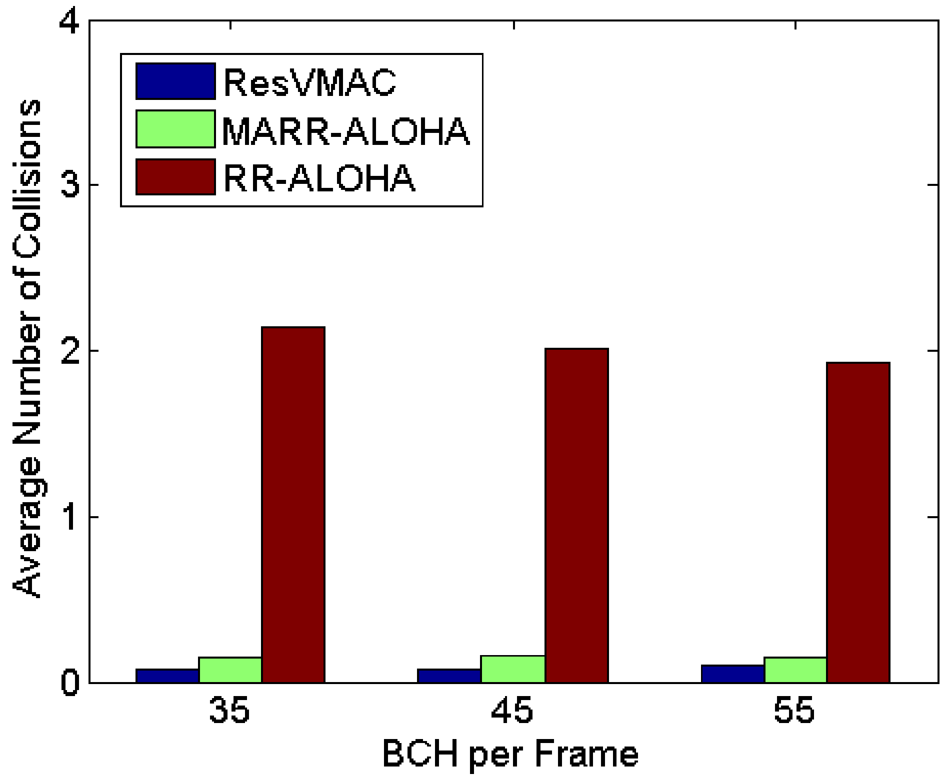


Figure 4.22: Average Number of Collisions while reserving BCHs (radio range= 200m)

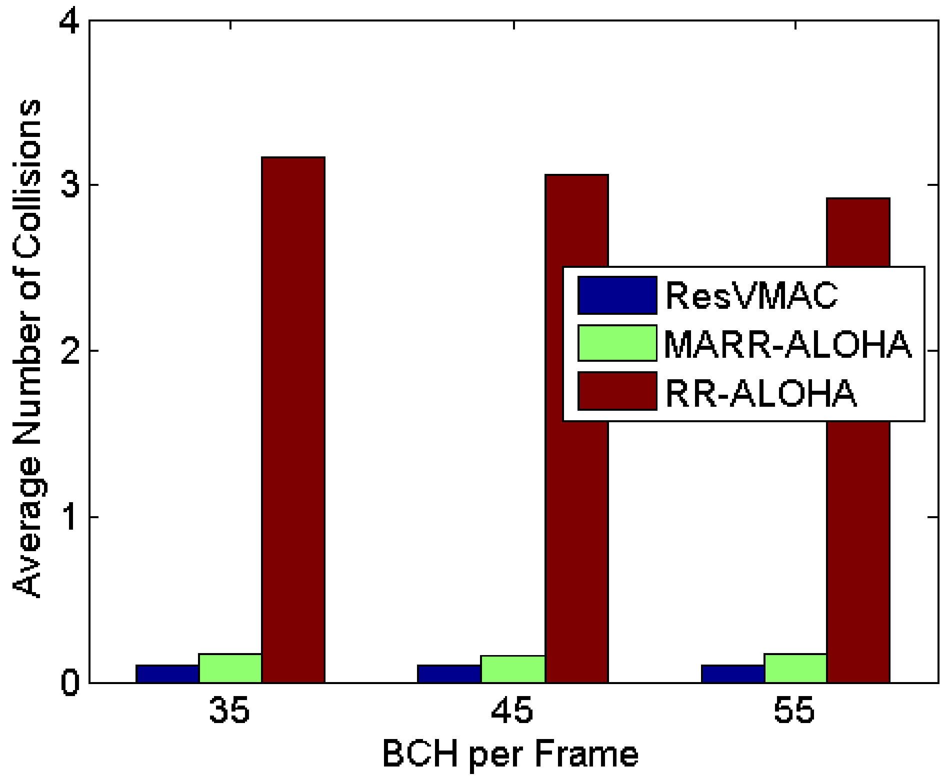


Figure 4.23: Average Number of Collisions while reserving BCHs (radio range= 300m)

Figure 4.22 and figure 4.23 show the number of collisions that occur during REQ and FI packet transmission for different number of BCHs per frame and different radio ranges. Since the REQ packet of ResVMAC is shorter, the possibility of collisions occurring is low, and this is confirmed in both figures. Almost all the vehicles have reserved their BCHs with very few collisions. On the other hand, the REQ packet of MARR-ALOHA is larger than ResVMAC so more collisions occur in MARR-ALOHA. We also observe that the number of collisions in RR-ALOHA is more than ResVMAC and MARR-ALOHA because of its larger REQ packets.

4.4.2 Average Number of Packets Delivered and Average Number of Collisions

In this set of simulations, we placed 100 vehicles in one lane and 100 vehicles in the other lane. Each vehicle moves on the road with a randomly chosen velocity between 10 km/h and 60 km/h and the velocity remains fixed throughout the simulation time. The radio range is set to 200m.

Table 4.2: Average Number of successful broadcasts

BCH/frame	ResVMAC	MARR-ALOHA	RR-ALOHA
40	4960.6	4918.1	4950.5
45	4963.0	4940.3	4955.0
50	4990.1	4951.8	4969.0

The simulation time is set 100, 112.5, and 125 sec for 40, 45, and 50 BCHs per frame respectively so that each vehicle has the chance to broadcast upto 5000 packets. Table 4.2 shows the average number of packets delivered for different numbers of BCHs per frame. We observe that the average packet delivery in ResVMAC is better than MARR-ALOHA and RR-ALOHA, implying that ResVMAC can reserve the channel quickly, and delivers more packets than MARR-ALOHA and RR-ALOHA.

The mean collisions per vehicle is plotted in figure 4.24 which shows that ResV-MAC has less number of mean collisions because it has smaller REQ packet than MARR-ALOHA and RR-ALOHA.

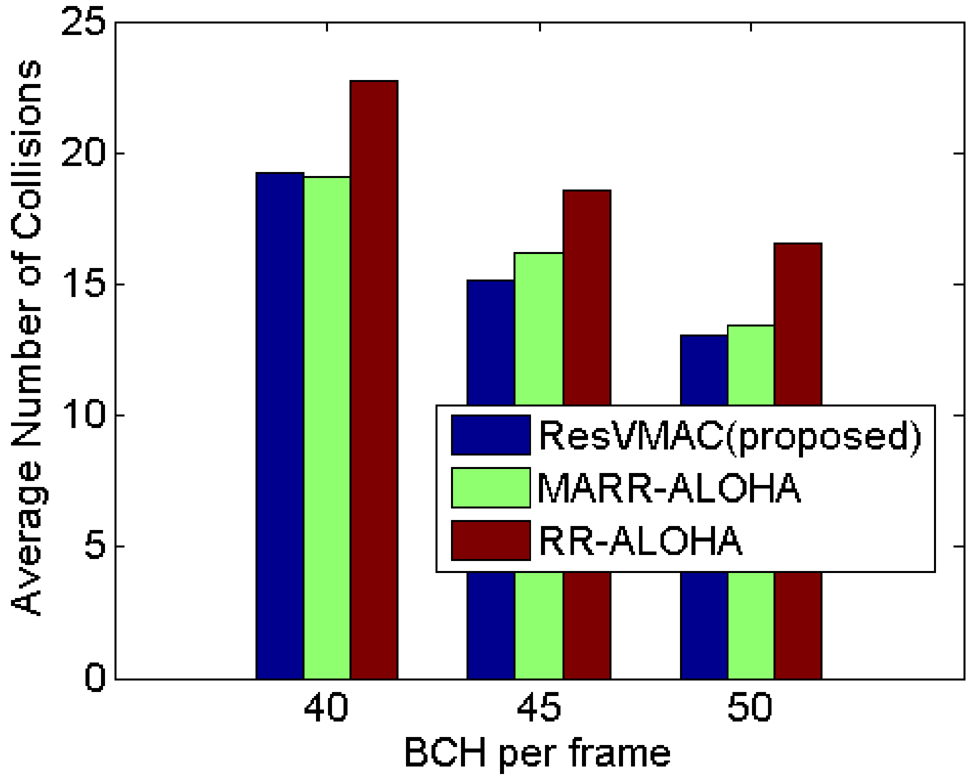


Figure 4.24: Average Number of Collisions per Vehicle (radio range= 200m)

Chapter 5

Conclusions and Future Work

In this chapter we summarize our contributions and provide some future directions to improve Ad-ATMA and ResVMAC.

5.1 Our contributions

WSAN MAC algorithm: We proposed algorithm Ad-ATMA that improves on the performance of ATMA algorithm. Recall that the ATMA algorithm was shown to outperform other existing algorithms for networks with medium to heavy loads. Our improvement was proposed after studying ATMA and seeing that nodes were sometimes unable to send packets because they were waiting behind other nodes. We alleviate the problem by dividing the contention window in ATMA into two parts, called the Selection Window and Surplus Window. If a node wants to reserve a data slot, it contends in the Selection Window. Nodes that are still waiting to send at the end of the Selection Window get a chance to transmit their ADV packets in the Surplus Window. Of course the sizes of these windows are crucial since decreasing the Selection Window increases the probability of collisions. So, we choose the best Selection Windows for different number of senders through experiments so that the benefit of splitting the contention window outweigh the loss due to the collisions. Our simulations show that Ad-ATMA improves on a state of the art algorithms ATMA and AdAMAC in terms of latency and PDR while using slightly less energy than them.

VANET MAC algorithm: As mentioned before, VANET topologies can change very quickly due to the high mobility of vehicles. The distributed TDMA-based protocols all attempt to achieve quick reservation of packet transmission slots (BCH) by vehicles that wish to send packets. Notable among these are MARR-ALOHA and RR-ALOHA. The vehicles using these protocols monitor the status of the BCHs at least one frame before they attempt to reserve a free BCH. They send REQ packets in the free BCHs that they want to reserve and wait for positive acknowledgements from their one-hop neighbours. After getting positive acknowledgements, they start sending data from the next frame. So, in these protocols vehicles waste two frames before sending data. For a highly dynamic technology like VANETs, quick reservation is very important. We designed ResVMAC to use a faster reservation scheme. In ResVMAC, if a node wants to send data it contends for a free BCH in the contention BCH by selecting a random number within the Reservation period. If it does not get any NACK from the one-hop neighbours immediately after its REQ packet transmission, it sends data in its reserved BCH. On the other hand, if it gets at least one NACK then it contends again in the next contention BCH. Thus ResVMAC allows vehicles to reserve BCHs quicker than MARR-ALOHA and RR-ALOHA which is showed in our simulations. As ResVMAC uses smaller REQ packet than MARR-ALOHA and RR-ALOHA, we also observe in our simulations that ResVMAC produces fewer collisions than MARR-ALOHA and RR-ALOHA.

5.2 Future Directions

- WSANs are increasingly designed with mobile nodes. Therefore, the most important extension of Ad-ATMA would be to mobile networks.
- We have simulated Ad-ATMA using Matlab R2013a. We have plan to implement it on sensor hardware.
- Mobility patterns have a big impact on the performance of VANET MAC algorithms. We have used a simple mobility model (a circular road with two lanes) to

evaluate the performance of our proposed VANET MAC algorithm ResVMAC. We plan to simulate more complex mobility models and see how ResVMAC performs.

Bibliography

[1] Crossbow Technology Inc. http://www.xbow.com.

[2] IEEE Trial-Use Standard for Wireless Access in Vehicular Environments (WAVE) - Networking Services,. *IEEE Std 1609.3-2007*.

[3] Sensor Nodes. http://en.wikipedia.org/wiki/Sensor_node.

[4] Sentilla Corporation. http://www.sentilla.com.

[5] Texas Transportation Institute. http://www.its.dot.gov/index.htm.

[6] World Health Organization. http://www.who.int/features/factfiles/roadsafety/en/.

[7] Md. Tareq Adnan. Practical Protocols for Wireless Sensor Networks. Master's thesis, York University, Canada, 2012.

[8] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. Wireless Sensor Networks: A Survey. *Comput. Netw.*, 38(4):393–422, March 2002.

[9] I.F. Akyildiz, Weilian Su, Y. Sankarasubramaniam, and E. Cayirci. A survey on sensor networks. *Communications Magazine, IEEE*, 40(8):102–114, Aug 2002.

[10] I.F. Akyildiz and M.C. Vuran. *Wireless Sensor Networks*. Advanced Texts in Communications and Networking. Wiley, 2010.

[11] Saif Al-Sultan, Moath M. Al-Doori, Ali H. Al-Bayatti, and Hussien Zedan. A comprehensive survey on vehicular Ad Hoc network. *Journal of Network and Computer Applications*, 37(0):380 – 392, 2014.

[12] M.S. Almalag, S. Olariu, and M.C. Weigle. TDMA cluster-based MAC for VANETs (TC-MAC). In *World of Wireless, Mobile and Multimedia Networks (WoWMoM), 2012 IEEE International Symposium on a*, pages 1–6, June 2012.

[13] Stefano Basagni, Marco Conti, Silvia Giordano, and Ivan Stojmenovic. *Mobile Ad Hoc Networking*. Wiley, 2004.

[14] Elizabeth A. Basha, Sai Ravela, and Daniela Rus. Model-based Monitoring for Early Warning Flood Detection. In *Proceedings of the 6th ACM Conference on*

Embedded Network Sensor Systems, SenSys '08, pages 295–308, New York, NY, USA, 2008. ACM.

[15] Sailesh Bharati and Weihua Zhuang. Performance analysis of cooperative AD-HOC MAC for vehicular networks. In *GLOBECOM*, pages 5482–5487, 2012.

[16] J.J. Blum, A. Eskandarian, and L. Hoffman. Challenges of intervehicle Ad Hoc networks. *Intelligent Transportation Systems, IEEE Transactions on*, 5(4):347–351, Dec 2004.

[17] P. Bonnet, J. Gehrke, and P. Seshadri. Querying the physical world. *Personal Communications, IEEE*, 7(5):10–15, Oct 2000.

[18] Flaminio Borgonovo, Antonio Capone, Matteo Cesana, and Luigi Fratta. RR-ALOHA, a Reliable R-ALOHA broadcast channel for ad-hoc inter-vehicle communication networks. In *in: Proceedings of Med-Hoc-Net 2002*, 2002.

[19] Michael Ignatius Brownfield. *Energy-efficient Wireless Sensor Network MAC protocol*. PhD thesis, Virginia Polytechnic Institute and State University, 2006.

[20] Michael Buettner, Gary V. Yee, Eric Anderson, and Richard Han. X-MAC: a short preamble MAC protocol for duty-cycled wireless sensor networks. In *Pro-*

ceedings of the 4th international conference on Embedded networked sensor systems, SenSys '06, pages 307–320, New York, NY, USA, 2006. ACM.

[21] H.A. Cozzetti and R. Scopigno. RR-ALOHA+: A slotted and distributed MAC protocol for vehicular communications. In *Vehicular Networking Conference (VNC), 2009 IEEE*, pages 1–8, Oct 2009.

[22] S. Datta. A randomized adaptive access control algorithm for sensor networks. In *Proceedings of the Second Workshop on Sensor and Actor Networks and Protocols (SANPA), Boston*, 2004.

[23] I Demirkol, C. Ersoy, and F. Alagoz. MAC protocols for wireless sensor networks: a survey. *Communications Magazine, IEEE*, 44(4):115–121, April 2006.

[24] A. El-Hoiydi and J.-D. Decotignie. WiseMAC: an ultra low power MAC protocol for the downlink of infrastructure wireless sensor networks. In *Proceedings of the Ninth International Symposium on Computers and Communications (ISCC)*, volume 1, pages 244 – 251 Vol.1, june-1 july 2004.

[25] Deborah Estrin, Ramesh Govindan, John Heidemann, and Satish Kumar. Next Century Challenges: Scalable Coordination in Sensor Networks. In *Proceedings of the 5th Annual ACM/IEEE International Conference on Mobile Computing*

and Networking, MobiCom '99, pages 263–270, New York, NY, USA, 1999. ACM.

[26] Tia Gao, C. Pesto, L. Selavo, Yin Chen, JeongGil Ko, Jong Hyun Lim, A. Terzis, A. Watt, J. Jeng, Bor rong Chen, K. Lorincz, and M. Welsh. Wireless Medical Sensor Networks in Emergency Response: Implementation and Pilot Results. In *Technologies for Homeland Security, 2008 IEEE Conference on*, pages 187–192, May 2008.

[27] H. Hartenstein and K.P. Laberteaux. A tutorial survey on vehicular Ad Hoc networks. *Communications Magazine, IEEE*, 46(6):164–171, June 2008.

[28] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan. Energy-efficient Communication Protocols for Wireless Microsensor Networks. In *Proceedings of the Hawaii International Conference on Systems Sciences*, 2000.

[29] J. Jakubiak and Y. Koucheryavy. State of the Art and Research Challenges for VANETs. In *Consumer Communications and Networking Conference, 2008. CCNC 2008. 5th IEEE*, pages 912–916, Jan 2008.

[30] D. Jiang and L. Delgrossi. IEEE 802.11p: Towards an International Standard for Wireless Access in Vehicular Environments. In *Vehicular Technology Conference, 2008. VTC Spring 2008. IEEE*, pages 2036–2040, May 2008.

[31] David B. Johnson, David A. Maltz, and Josh Broch. Ad hoc networking. chapter DSR: The Dynamic Source Routing Protocol for Multihop Wireless Ad Hoc Networks, pages 139–172. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2001.

[32] J.B. Kenney. Dedicated Short-Range Communications (DSRC) Standards in the United States. *Proceedings of the IEEE*, 99(7):1162–1182, July 2011.

[33] Younghun Kim, Thomas Schmid, Zainul M. Charbiwala, Jonathan Friedman, and Mani B. Srivastava. NAWMS: Nonintrusive Autonomous Water Monitoring System. In *Proceedings of the 6th ACM Conference on Embedded Network Sensor Systems*, SenSys '08, pages 309–322, New York, NY, USA, 2008. ACM.

[34] Dan Li, K.D. Wong, Yu Hen Hu, and A.M. Sayeed. Detection, classification, and tracking of targets. *Signal Processing Magazine, IEEE*, 19(2):17–29, Mar 2002.

[35] Ruilin Liu, Yong Xiang, and Yanqing Yang. MARR-ALOHA: A mobility adaptive variety of RR-ALOHA for vehicular Ad-hoc networks. In *Consumer Electronics, Communications and Networks (CECNet), 2012 2nd International Conference on*, pages 3304–3309, April 2012.

[36] Alan Mainwaring, David Culler, Joseph Polastre, Robert Szewczyk, and John Anderson. Wireless Sensor Networks for Habitat Monitoring. In *Proceedings of*

the 1st ACM International Workshop on Wireless Sensor Networks and Applications, WSNA '02, pages 88–97, New York, NY, USA, 2002. ACM.

[37] David Malan, Thaddeus Fulford-jones, Matt Welsh, and Steve Moulton. CodeBlue: An ad hoc sensor network infrastructure for emergency medical care. In *In International Workshop on Wearable and Implantable Body Sensor Networks*, 2004.

[38] J. Mao, Z. Wu, and X. Wu. A TDMA scheduling scheme for many-to-one communications in wireless sensor networks. *Computer Communications*, pages 863–872, 2007.

[39] Stephan Olariu and Michele C Weigle. *Vehicular networks: from theory to practice*. CRC Press, 2010.

[40] H.A. Omar, Weihua Zhuang, and Li Li. VeMAC: A TDMA-Based MAC Protocol for Reliable Broadcast in VANETs. *Mobile Computing, IEEE Transactions on*, 12(9):1724–1736, Sept 2013.

[41] Guangyu Pei, Mario Gerla, and Tsu-Wei Chen. Fisheye State Routing in Mobile Ad Hoc Networks. In *In ICDCS Workshop on Wireless Networks and Mobile Computing*, pages 71–78, 2000.

[42] Jun Peng, Liang Cheng, and B. Sikdar. A Wireless MAC Protocol with Collision Detection. *Mobile Computing, IEEE Transactions on*, 6(12):1357–1369, Dec 2007.

[43] C. Perkins, E. Belding-Royer, and S. Das. Ad Hoc On-Demand Distance Vector (AODV) Routing, 2003.

[44] E.M. Petriu, Nicolas D. Georganas, D.C. Petriu, D. Makrakis, and V.Z. Groza. Sensor-based information appliances. *Instrumentation Measurement Magazine, IEEE*, 3(4):31–35, Dec 2000.

[45] J. Polastre, J. Hill, and D. Culler. Versatile low power media access for wireless sensor networks. In *Proceedings of IEEE International Conference on Embedded Networked Sensor Systems (Sensys)*, pages 95–107, 2004.

[46] Venkatesh Rajendran, Katia Obraczka, and J.J. Garcia-Luna-Aceves. Energy-Efficient, Collision-Free Medium Access Control for Wireless Sensor Networks. *Wireless Networks*, 12(1):63–78, 2006.

[47] Iyappan Ramachandran and Sumit Roy. On the Impact of Clear Channel Assessment on MAC Performance. In *GLOBECOM*. IEEE, 2006.

[48] S. Ray, I. Demirkol, and W. Heinzelman. ATMA: Advertisement-Based TDMA Protocol for Bursty Traffic in Wireless Sensor Networks. In *Global Telecommunications Conference (GLOBECOM 2010), 2010 IEEE*, pages 1–5, Dec 2010.

[49] S. S. Ray, I. Demirkol, and W. R. Heinzelman. Advertisement-Based MAC Protocol for Wireless Sensor Networks. In *Proceedings of 5th International Conference on Mobile Ad-hoc and Sensor Networks (MSN)*, pages 265–272, 2009.

[50] S. S. Ray, I. Demirkol, and W. R. Heinzelman. ATMA: Advertisement-Based TDMA Protocol for Bursty Traffic in Wireless Sensor Networks. In *Proceedings of IEEE GLOBECOM*, pages 1–5, 2010.

[51] John Schiller. *Mobile Communications*. Pearson Education Limited, UK, 2003.

[52] R. Scopigno and H.A. Cozzetti. Mobile Slotted ALOHA for VANETs. In *Vehicular Technology Conference Fall (VTC 2009-Fall), 2009 IEEE 70th*, pages 1–5, Sept 2009.

[53] Xingfa Shen, Zhi Wang, and Youxian Sun. Wireless sensor networks for industrial applications. In *Intelligent Control and Automation, 2004, WCICA 2004, Fifth World Congress on*, volume 4, pages 3636–3640 Vol.4, June 2004.

[54] F. Sivrikaya and B. Yener. Time synchronization in sensor networks: a survey. *Network, IEEE*, 18(4):45–50, July 2004.

[55] I. Stojmenovic. *Handbook of Sensor Networks: Algorithms and Architectures*. Wiley Series on Parallel and Distributed Computing. Wiley, 2005.

[56] Y. Toor, P. Muhlethaler, and A. Laouiti. Vehicle Ad Hoc networks: applications and related technical issues. *Communications Surveys Tutorials, IEEE*, 10(3):74–88, Third 2008.

[57] R. Uzcategui and G. Acosta-Marum. WAVE: A tutorial. *Communications Magazine, IEEE*, 47(5):126–133, May 2009.

[58] T. van Dam and K. Langendoen. An adaptive energy-efficient MAC protocol for wireless sensor networks. In *Proceedings of the 1st international conference on Embedded networked sensor systems*, SenSys '03, pages 171–180, New York, NY, USA, 2003. ACM.

[59] Zhe Wang and Mahbub Hassan. How Much of DSRC is Available for Non-safety Use? In *Proceedings of the Fifth ACM International Workshop on VehiculAr Inter-NETworking*, VANET '08, pages 23–29, New York, NY, USA, 2008. ACM.

[60] L. Wischhof, A. Ebner, and H. Rohling. Information Dissemination in Self-organizing Intervehicle Networks. *Trans. Intell. Transport. Sys.*, 6(1):90–101, March 2005.

[61] W. Ye, J. Heidemann, and D. Estrin. An energy-efficient MAC protocol for wireless sensor networks. In *Proceedings of the IEEE Infocom. New York, NY, USA: IEEE*, pages 1567–1576, 2002.

[62] Fan Yu and S. Biswas. Self-Configuring TDMA Protocols for Enhancing Vehicle Safety With DSRC Based Vehicle-to-Vehicle Communications. *Selected Areas in Communications, IEEE Journal on*, 25(8):1526–1537, Oct 2007.

[63] Hongqiang Zhai, Yuguang Fang, and Multihop Wireless. Physical Carrier Sensing and Spatial Reuse in Multirate and Multihop Wireless Ad Hoc Networks, 2006.

[64] M. Zhang and R.S. Wolff. Routing protocols for vehicular Ad Hoc networks in rural areas. *Communications Magazine, IEEE*, 46(11):126–131, November 2008.