Application of Reinforcement Learning on Medium Access Control for Wireless Sensor Networks

Yi Chu

Ph.D. Thesis

University of York

Electronics

September 2013

Abstract

This thesis investigates the application of Reinforcement Learning (RL) on Medium Access Control (MAC) for Wireless Sensor Networks (WSNs). RL is applied as an intelligent slot selection strategy to Framed ALOHA, along with analytical and experimental performance evaluation. Informed Receiving (IR) and ping packets are applied to multi-hop WSNs to avoid idle listening and overhearing, thereby further improving the energy efficiency.

The low computational complexity and signalling overheads of the ALOHA schemes meet the design requirement of energy constraint WSNs, but suffer collisions from the random access strategy. RL is applied to solve this problem and to achieve perfect scheduling. Results show that the RL scheme achieves over 0.9 Erlangs maximum throughput in single-hop networks. For multi-hop WSNs, IR and ping packets are applied to appropriately switch the relay nodes between active and sleep state, to reserve as much energy as possible while ensuring no information loss.

The RL algorithms require certain time to converge to steady state to achieve the optimum performance. The convergence behaviour is investigated in this thesis. A Markov model is proposed to describe a learning process, and the model produces the proof of the convergence of the learning process and the estimated convergence time. The channel performance before convergence is also evaluated.

Contents

List of Figures

List of Tables

Acknowledgements

I would like to dedicate this thesis to my parents in recognition of their continued support, encouragement and endless love.

I would like to thank my first supervisor Dr. Paul Mitchell for his invaluable guidance, support and advice. I also would like to acknowledge the suggestions and comments provided by my second supervisor Dr. David Grace.

It has been a pleasure to work with my colleagues in Communications Group who have provided a comfortable working environment. Finally, I have enjoyed my life in the University of York with students from China and all other countries over the world.

Declaration

Some of the research presented in this thesis results in a number of publications. All contributions specified in this thesis are as original to the best knowledge of the author. Appropriate references and acknowledgements to other researchers in the field have been given.

1. Introduction

Contents

1.1. Motivation

Medium Access Control (MAC) has a significant impact on the energy efficiency and channel performance of Wireless Sensor Networks (WSNs). As a result of the constrained energy supply of WSNs, specific MAC protocols are required to ensure the network meets its intended application goals. Based on the principle of periodically switching nodes between sleep and active states, MAC protocols can significantly improve the lifetime of WSNs.

Researchers have proposed many MAC protocols for WSNs, which have achieved considerable improvements in lifetime, throughput and delay performance. However, tradeoffs exist between these energy efficiency benefits and the expenses of higher overheads and ever increasing complexity (e.g. some protocols require periodic update of the scheduling information of one-hop or even two-hop neighbours). A much simpler MAC protocol with energy efficiency, appropriate delay and throughput is potentially promising, and is well suited to extremely small and basic sensor nodes for future applications.

ALOHA based schemes have the advantages of simplicity and low overheads, but poor throughput performance resulting from their blind transmission strategy. This thesis proposes Reinforcement Learning (RL) as an intelligent slot selection strategy to avoid collisions and retransmissions, thereby improving throughput and energy efficiency. For single-hop networks, nearly perfect slot scheduling can be achieved by RL, with the only overheads required being acknowledgement (ACK) packets. For multi-hop networks, considering the node density and network size, further changes need to be made to adjust to different topologies and sleeping schedules. Results show that RL algorithms increase the maximum throughput to almost three times the throughput of the schemes without intelligence, with much lower delay and reduced energy cost. A detailed investigation and evaluation of novel Reinforcement Learning based MAC protocols are presented in Chapter 4 and Chapter 6.

1.2. Thesis Structure

This thesis comprises 8 chapters and the remainder of the thesis is organised as follows.

Chapter 2 provides an overview of WSNs, including their development history, purpose and requirements. This chapter briefly introduces the applications and devices, and states the constraints to which they are subject. As the primary research topic in this thesis, the basic principles of MAC protocols are presented in this chapter, as well as the RL techniques used in subsequent protocol design.

Chapter 3 provides background information and a survey of work related to the research in this thesis. The principles of medium access control and multiple access are discussed, as well as the design requirements for MAC protocols for WSNs. An extensive literature review is provided in this chapter and categorises the MAC protocols into contention-based and schedule-based, or a combination of both.

Chapter 4 provides a detailed description of applying Reinforcement Learning to ALOHA by considering single-hop network topologies. Weights are applied to the slots of frame-based ALOHA, with successful transmissions and collisions all contributing to the weight update. Results show that a steady state of nearly perfect scheduling is achieved by nodes learning from their transmission history, given appropriate frame sizes. The achievable throughput is close to the theoretical maximum without introducing additional overheads.

Chapter 5 focuses on the behaviour of RL schemes during the learning period before reaching steady state. According to the special characteristic of learning-based protocols, a certain time is required for the network to converge to an optimal steady state. The convergence time can be considered as the initialisation time of the network, and the network does not perform optimally during the learning period. The convergence time can be affected by many factors such as the node density, communication radius, network topology routing and learning strategy. This chapter proposes a Markov model to estimate the convergence time for single-hop networks and validate it by simulations. A discussion about channel performance during the learning process is also provided in this chapter.

Chapter 6 extends the work in Chapter 4 to multi-hop networks. The constraints of multi-hop networks are discussed and the work in this chapter mainly focuses on linear networks and random networks. Informed Receiving (IR) and ping packets are applied to appropriately switch the nodes between active and sleep modes, thereby improving energy efficiency and channel performance. Results show that the throughput is significantly improved by applying learning, and the unnecessary energy costs and overheads are kept at low levels.

Chapter 7 describes potential further research to extend this thesis. The convergence time and steady state performance can be improved by applying different RL algorithms. In WSNs it is difficult to implement global synchronisation, so it is important to consider the adaptability of WSN MAC protocols to unsynchronised conditions. Introducing RL to frame size selection makes the schemes applicable to networks with different topologies and adaptive to changing scenarios (e.g. new nodes introduced, old nodes dying, changing propagation conditions, variable traffic requirements), and eliminates the requirements for prior frame size estimation.

Chapter 8 summarises the work in this thesis. The novelty and original contributions are highlighted along with publications developed as a result of the work undertaken.

2. Background

Contents

Wireless Sensor Networks (WSNs) are designed for the purpose of completing different monitoring tasks under various environmental conditions at low cost [1]. With an increasing demand for precise observation of physical phenomena, conventional detection and monitoring techniques become cost-prohibitive especially for long-term tasks. WSNs with cost effective nodes are a viable solution for these tasks. Cost effective nodes can be deployed in different sensing areas and complete the tasks by collaboration. The development of micro-electro-mechanical systems (MEMS) brings various capabilities to sensor nodes. Therefore, WSNs have a wide range of applications in military environments, for medical treatments, industry, security and environmental monitoring [1].

A WSN is usually a self-organising network consisting of a large number of sensor nodes which are able to sense and process environmental data through multi-hop communication and coordination. Sensor nodes are usually supplied by limited power sources (such as AA batteries) but are designed so the failure of single nodes will not affect the task of the network. However according to the application goals, sensor nodes may be deployed in unreachable areas (such as animal bodies, contaminated regions or battle fields [1]) which makes charging or replacing batteries difficult. Therefore the lifetime of the network will mainly rely on energy efficiency. Given the limited channel resources in WSNs, nodes within a certain range have to share the same medium, which makes Medium Access Control (MAC) very important. The energy consumed by radio transceivers is mostly from transmitting, receiving, idle listening and over hearing. In addition to the useful throughput, energy is consumed by packet collisions, retransmissions, control packet overheads, idle listening and over hearing [2]. A well designed MAC protocol should keep these energy waste mechanisms to a low level, while achieving good throughput and delay performance.

Due to the special characteristics of WSNS, conventional MAC protocols may not be directly applicable to them. The communication range of a node is usually much smaller than the range of the network, which makes global control very difficult. Conventional MAC protocols usually focus on how to provide better quality of service (QoS), achieve higher bandwidth efficiency and deal with user mobility. However in WSNs, energy efficiency is considered to be the priority rather than QoS. MAC protocols for WSNs also need to be adaptable to topology changes [2].

Researchers have proposed many MAC protocols for WSNs and achieved significant improvements on energy efficiency and channel performance. However, these improvements have been introduced at the expense of higher overheads and increasing complexity. Given the long-term vision of huge numbers of devices embedded into machines everywhere, there is a need to develop much simpler protocols which can nonetheless provide energy-efficient communication, good throughput and adequate delay. ALOHA based schemes have the benefits of simplicity, low computation and overheads, but suffer collisions as a result of the blind transmission strategy. Intelligent slot selection strategies can significantly improve the channel performance and energy efficiency. In this thesis, Reinforcement Learning (RL) is applied to ALOHA for this purpose, to enable nodes to develop an effective transmission strategy based on their prior experience on the channel.

Reinforcement Learning (RL) is a technique where users learn effective strategies through trial-and-error interactions in a dynamic environment, with future actions determined by prior experience [3]. It has been widely used in research on Artificial Intelligence (AI), and has recently been applied to communication problems and MAC layer protocols. In [4] and [5] several Reinforcement Learning based MAC protocols are proposed. Although they are not specifically designed for WSNs, a similar strategy can be brought to the design of MAC protocols for WSNs.

2.1. Overview of Wireless Sensor Networks

With their characteristics of low cost, self-organisation and multi-hop communication, WSNs have unique capabilities and design requirements. This section provides a brief introduction to WSNs including their applications, existing devices, system architecture and design constraints.

2.1.1. Applications of WSNs

During the 1980s, research on sensor networks was started by the Distributed Sensor Network (DSN) program at the Defence Advanced Research Projects Agency (DARPA), and WSNs were initially produced to detect enemy submarines and aircraft in the Cold War [5]. Compared to the traditional detection and tracking methods such as radar, sonar and satellite, a sensor network has a lot of advantages. Easy deployment improves tracking precision and detection range, with destruction of a proportion of sensor nodes not particularly significant because of the low device costs, the distributed processing and redundancy among sensor nodes. WSNs provide many possibilities for the military and home land security. By carrying sensors, units can report their location and situation anywhere, any time. Sensors can be deployed in critical terrains for reconnaissance, or in friendly regions to detect and warn people of nuclear, biological and chemical attack [6].

WSNs can be deployed and complete long-term tasks in areas which are difficult to reach for manual monitoring, which makes WSNs suitable for environmental monitoring tasks. In 2004 and 2005, a team of scientists from Harvard University, the University of North Carolina and the University of New Hampshire built a WSN for monitoring an active volcano [7]. They deployed a WSN with 16 nodes, which collected data for three days and recorded 230 volcanic events. In 2002 a group of scientists from The College of the Atlantic deployed a WSN on Great Duck Island to observe the breeding and living of the Leach’s Storm Petrel [8]. The sensor nodes were deployed near the nesting burrows of the Petrels and they obtained the daily patterns of presence and absence of the adult birds by sensing temperature differences. Using WSNs only requires at most two visits: deployment and recycling, which keeps the disturbance to the individual ecosystem very low. WSNs can also be deployed in bridges for structural health monitoring to detect and locate damage and estimate their lifetime [9]. Compared to conventional wired sensing systems, WSNs have lower cost, greater density and coverage,

With their remote monitoring and wireless features, WSNs can be applied to the oil and gas industry [10]. Report [11] shows the capability of WSNs for these industries. With secure and reliable communications, WSNs have advantages that wired systems cannot achieve. WSNs can provide real-time data for monitoring pipeline integrity, tank levels, equipment condition, pipeline pressure and refinery pressure. By removing wires, WSNs also significantly cut down the cost of monitoring huge pipeline networks. The work in [12] presents the application of WSNs to machinery fault diagnosis. Sensors cause minimum disturbance to the normal production of enterprises by removing the wire. WSNs monitor the condition of the machines by collecting vibration, noise and temperature data.

2.1.2. Devices

Each sensor node includes four basic subsystems: the sensing subsystem, the processing subsystem, the communication subsystem and the power supply subsystem [13]. The sensing subsystem contains one or more sensors with analogue-to-digital convertors which can collect various types of environmental data and quantise them into digital signals. The processing subsystem consists of a microcontroller, which is the core component in charge of data and protocol processing. The communication subsystem includes a transceiver, which could be radio frequency, optical or infrared. This subsystem is the most power consuming component. The power supply subsystem usually has a battery as the power source, and its capacity determines the lifetime of the sensor node.

There are two popular options for the communication technique: radio frequency (RF) and optical transmissions. With respect to RF, the size of the node only offers limited space for antennas and requires high frequency (short wave length) transmissions. RF also needs modulation, demodulation and band pass filtering which require complex circuitry and lead to difficulties in reducing power consumption [14]. However, since a clear line-of-sight cannot be guaranteed most of the time in typical WSN environments, RF are preferred for the nodes considered in this thesis.

Most of the RF transceivers are designed for the ISM bands because these frequencies do not require license and offer more possibilities for power efficient designs [15]. Considering the energy consumption, size and cost of a sensor node, the processor, storage and data rate are all limited. For example, the CITRIS node [16] has only a 4 MHz CPU and 8-bit processor. The storage is only 8 kB instruction flash and 512 B RAM, and the RF transceiver operates on 916 MHz ISM band with a bandwidth of only 10 kbps. Mica motes [17] have only 128 kB flash, 4 kB data storage and 20 kbps data rate. Table 2.1 gives a list of the characteristics of some existing commercial sensor nodes [18]. Some simulation parameters in other sections are taken from the nodes in this table.

Node name	Controller	Transceiver	Program and Data memory	External memory	Programming	Platform
BTnode [19]	Atmel ATmega 128L	Chipcon CC1000 and Bluetooth	64+180K RAM	128K FLASH ROM, 4K EEPROM	C and nes C	BTnut and TinyOS
EPIC mote [20]	Texas Instruments MSP430	250 kbit/s 2.4 GHz IEEE 802.15.4 Chipcon	10 K RAM	48 K flash		TinyOS
FlatMesh [21]	16 MHz	802.15.4-compliant		660 sensor readings	Over-air control	Commercial system
GWnode [22]	PIC18LF8722	BiM (173 MHz) FSK	64K RAM	128K flash	C	Custom OS
IMote [23]	ARM core 12 MHz	Bluetooth	64K SRAM	512K flash		TinyOS
Iris Mote [24]	ATmega 128	Atmel AT86RF230 802.15.4/ZigBee	8K RAM	128K flash	nesC	TinyOS, MoteWorks
Mica [25]	ATmega 103	RFM TR1000	128+4K RAM	512K flash	nesC	TinyOS
MicaZ[26]	ATmega 128	TI CC2420 802.15.4/ZigBe	4K RAM	128K flash	nesC	TinyOS, SOS, MantisOS and Nano-RK
Mulle [27]	Renesas M16C	Atmel AT86RF230 802.15.4 / Bluetooth 2.0	31K RAM	384K+4K Flash, 2 MB EEPROM	nesC, C	Contiki, TinyOS, lwIP
NeoMote [28]	ATmega 128L	TI CC2420 802.15.4/ZigBe	4K RAM	128K flash	nesC	TinyOS, SOS, MantisOS, Nano-RK and Xmesh
Firefly [29]	ATmega 1281	Chipcon CC2420	8K RAM	128K FLASH ROM, 4K EEPROM	C	Nano-RK RTOS

Table 2.1 Some existing commercial sensor nodes

2.1.3. System Architecture

A WSN consists of five basic architectural layers: application layer, transport layer, network layer, data link layer and physical layer [1]. The application layer determines the relevant data according to the application tasks and informs the lower layers of these interests. In data-centric sensor networks, the application layer can also identify the nodes by the data they sense. The transport layer is in charge of connecting WSNs to other networks when required for specific applications. The network layer forwards data through chosen directions (known as routes) to the sink nodes. Medium Access Control (MAC) protocols in the data link layer manage channel sharing between neighbouring nodes. The physical layer provides signal transmission and reception, modulation and coding.

The research work in this thesis focuses on MAC protocols in the data link layer, but routing protocols in the network layer is a highly related research area and cross-layer protocols are also developed. The routing protocol is in charge of achieving energy efficiency on setting up the network topology and routing the data from the source to the sink nodes or the base station.

Routing protocols categorise the network topology into two basic types, flat and hierarchical. In flat routing, all nodes in the network have the same status and they can communicate with each other without crossing a level boundary in a network hierarchy. The routing protocol routes data to the sink nodes through one-hop or multi-hop paths, choosing the optimal path by analysing the characteristics of the nodes on different routes. In hierarchical routing the network is usually cluster based, and a cluster usually consists of several nodes and a cluster head. Nodes in one cluster only talk to the cluster head, and the cluster head does not need to take on sensing tasks. However, it will collect data from other nodes in the cluster, aggregate the data and send it back to the base station. Low Energy Adaptive Clustering Hierarchy (LEACH) is a classic hierarchical routing strategy [30] for WSNs, which reduces and evenly distributes the energy consumption among the nodes in the network. LEACH is a cluster based routing strategy, with the cluster head in charge of collecting and aggregating data and sending it back to the base station. It can be considered as a cross-layer protocol because it also schedules transmissions in each individual clusters.

2.1.4. Constraints of WSNs

Given the various application goals of WSNs, sensors can be deployed in unreachable areas such as contaminated areas, animal bodies, battle fields and under water, which causes difficulties in replacing or recharging their batteries. With limited energy sources, WSN lifetimes can become a problem especially for long-term tasks. To improve the lifetime, nodes either carry larger batteries or external energy sources which sacrifice size and weight advantages, or apply specific protocols to keep the energy waste to a minimum. As a result, for most applications energy efficiency will always be the priority unless much better energy sources are found.

The energy constraints of WSNs bring special requirements for the design of routing and MAC protocols. Conventional routing protocols usually select routes by considering QoS (available bandwidth, data rate and delay), however for WSNs such principles will drain the energy of nodes on a certain route much faster than others and cause a gap in the network. With limited communication range and no global control, nodes can hardly forward information to other nodes more than one-hop away. This causes difficulties for MAC protocols in scheduling transmissions and avoiding collisions.

2.2. Overview of MAC protocols for WSNs

According to the energy constraints of WSNs, sensor nodes usually have limited hardware and communication radius, which makes Medium Access Control (MAC) important in ensuring that the network remains operational. The special characteristics of WSNs cause many difficulties to the design of MAC protocols. The multi-hop topology brings hidden terminal problems [2] which lead to collisions and retransmissions. Sensor nodes usually switch between active (transmitting, receiving or listening) and sleep modes (turning off the radio) to save energy, which requires the MAC scheduling to be precise and effective. However, forwarding scheduling information to nodes more than one-hop away is energy consuming for WSNs supporting a large coverage area because of the limited communication radius. A well designed MAC protocol needs find solutions to these problems and keep the unnecessary energy consumption to a minimum.

MAC protocols are in charge of setting up the network infrastructure and assigning channel resources. MAC protocols for WSNs usually belong to one of two categorises: contention based and schedule based [2]. For contention based MAC protocols, nodes do not have pre-scheduled transmissions and they need to contend for the channel when they have data to transmit. Schedule based MAC protocols usually assign channel resources through reservations, with transmission slots reserved for each node according to the node’s traffic flow. In most monitoring applications, nodes need to continue collecting data for weeks or even months, so MAC protocols also need to improve the lifetime of the network. Many MAC protocols introduce a duty cycle to achieve a longer lifetime [2]: a node has a wake and a sleep mode, and during the wake periods the nodes exchange data with the others and during the sleep periods the nodes turn off their radio to reduce the power consumption. With different sleeping schedules, there will always be a certain amount of active nodes in the network. Long network lifetime can ensure the completion of the sensing task by only one deployment (replacing or recharging of the battery is not required). A more detailed literature review of MAC protocols for WSNs is provided in Chapter 3.

2.3. Background of Reinforcement Learning

The idea of Reinforcement Learning (RL) first appeared in 1979 when machine learning and neuron-like elements for artificial adaptive intelligence were developed. The potential of RL was rapidly explored in psychology, control theory, neuroscience, optimal control and dynamic programming [29]. RL was developed from an understanding from human interactions with the environment, and how we make decisions in response to the environment. RL records historical experience built up by rewards and punishments, learning behaviour and determines future actions through trial-and-error interactions with a dynamic environment. Many mathematical approaches (such as dynamic programming, Monte Carlo methods and temporal difference learning [29]) have been developed to solve different problems.

Based on the trial-and-error interaction with the environment, RL usually quantises the environment to different states, and agents learn their behaviours through actions and rewards. Each state has a subset of actions, and the actions are independent for different states. Agents receive rewards after each action, and the rewards only affect the current action and state pair. The received rewards usually depend on the goal of the RL process. Actions that push the process to the goal can have positive rewards and vice versa. Each action and state pair has an associated weight, which is updated by received rewards and indicates the preference with which they are selected.

Figure 2.1 shows how RL can be used in a simple square maze. A person stands at point A in a square maze, and the destination is B. The position of this person is considered as the state of the RL algorithm, and there are 36 squares in this maze so the associated RL process has 36 states. In each state, the person can take four possible actions: move to the four neighbour squares (up, down, left and right). Each action receives a reward of -1, unless the person falls off the edge (eg. moves down from point A) when the reward becomes -100. Reaching the destination B returns a reward of +100. In each state the agent selects the action with the highest weight, or if several actions have the same highest weight, it selects a random one. After many trials, the RL process produces an optimum route, the two solid routes in Figure 2.1 can possibly be obtained. The optimum routes must have 7 steps (which is minimum), but the dashed route will not be selected by the RL process (although it is an optimum route), because it is close to the edge and the actions have a higher probability of receiving negative rewards.

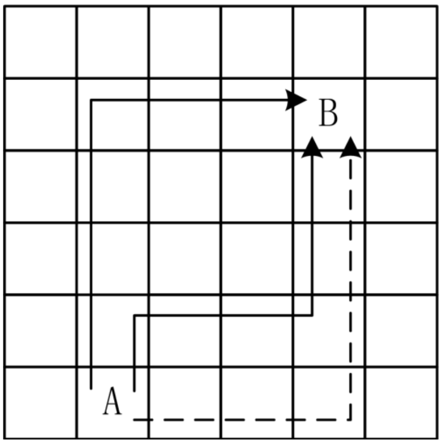


Fig. 2.1 An example of Reinforcement Learning

The trade off between exploration and exploitation is an important challenge in RL. To achieve higher weights and rewards, agents usually select actions based on their current experience and history, so the actions which receive good rewards will always be preferred. However, it is possible that the preferred action is a sub-optimal solution. The agent needs to explore different actions to find an optimum solution in the future, but the current experience cannot tell the existence of a better solution and there is a risk that the cost of exploration produces nothing. Determining the balance between exploration and exploitation has a significant impact on RL.

Figure 2.2 shows an example of exploration and exploitation. A person stands at point A in a similar square maze as in Figure 2.1, and the destination is point B. This maze has the same actions, states and rewards as the previous one except for the grey area. Once the person moves into the grey area, a -100 reward will be returned, same as falling off the edge of the maze. For the RL algorithm without exploration, the solid route will be selected, but it is a sub-optimal solution with 9 steps to the destination. The dashed route will not be selected because the actions in these states have a large probability of receiving a -100 reward. Through exploration (eg. try a sub-optimal action with a low probability in each state), the optimum dashed route can be found, which is 7 steps to the destination.

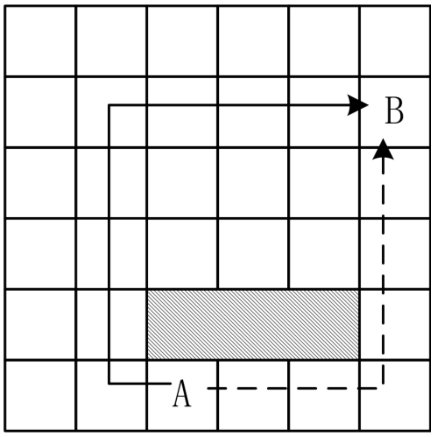


Fig. 2.2 Exploration and Exploitation

From the communication aspect, RL has recently been used in cognitive radio research. In [4] and [5] RL brings intelligence to the channel selection strategy and adaptability to environmental changes. In this thesis, we primarily consider channel sharing under single channel conditions. We apply RL as an intelligent slot selection strategy which requires minimal overheads and computation, thereby improving the energy efficiency and lifetime of WSNs. The detailed RL strategies are introduced in Chapters 4 to 6.

2.4. Summary

This chapter provides background information relevant to the research reported in this thesis. WSNs are briefly introduced, including applications, devices, system structure and design requirements. MAC protocols for WSNs are the main subject investigated in this thesis, and a detailed literature review about MAC protocols for WSNs are provided in Chapter 3. This chapter also introduces Reinforcement Learning, which is an important technique used in the implementation of the intelligent slot selection strategies in later chapters.

3. MAC Protocols for WSNs – Literature Review

Contents

This chapter provides background information on Medium Access Control (MAC) protocols for WSNs and an extensive literature review. Firstly, basic multiple access techniques are introduced, which demonstrate important principles of medium access control. Then, existing MAC protocols are categorised as contention-based, schedule-based, and a combination of both. A brief introduction to each protocol is provided.

3.1. Introduction

MAC protocols control the way that users access a shared channel resource. MAC protocols have a variety of design purposes depending on their intended goals. For conventional communication systems such as cellular mobile phone networks, ad hoc networks and Bluetooth, MAC protocols usually focus on how to provide better Quality of Service (QoS), achieve higher bandwidth efficiency, and deal with user mobility. For WSNs, energy consumption is almost always the priority. WSNs MAC protocols therefore need to achieve better energy efficiency and adaptability to frequent topology changes, with other issues secondary to these priorities [2]. Due to these differences in design purposes between WSNs-specific MAC protocols and general MAC protocols, some of the existing MAC protocols are inappropriate to directly apply to WSNs. Therefore, specific MAC protocols must be developed to achieve the best performance in WSNs.

MAC protocols have a significant impact on the energy consumed by the radio transceiver. Typically energy consuming behaviours include transmission, reception and idle listening. The energy consumption of transceivers during their sleep state can be neglected comparing to other states [31]. Necessary energy consumption refers to energy consumed by useful throughput; unnecessary energy consumption mainly comes from retransmissions, idle listening, overhearing and control packet overheads [2]. To improve the energy efficiency and lifetime of the network, unnecessary energy consumption must be minimised.

The specific features of WSNs cause many difficulties in the design of MAC protocols. As a result of the limited radio hardware of WSNs, MAC protocols usually consider time-sharing under single channel conditions, which makes collision avoidance important to achieve energy efficiency. Reservations can be used to avoid most collisions, however this approach increases the overheads of the protocol. MAC protocols for WSNs have to switch nodes between sleep and active modes to save energy. However the intended receiver sleeping will block the incoming transmissions and lose information. On the other hand, idle listening and channel sensing can ensure reception and collision avoidance but lead to more energy consumption. Generally, the principle for MAC protocols is simple: switch nodes to active mode when they need to transmit or receive data and switch them to sleep mode between transmissions and receptions, but this is a complicated goal to achieve. A well designed MAC protocol needs to find solutions for the issues above to achieve appropriate channel performance.

3.2. Multiple Access Techniques

Multiple access techniques control how users share the same medium. Several basic multiple access techniques are introduced in this section. Considering the design characteristics described for WSNs, the focus of this thesis is upon Time Division Multiple Access (TDMA).

3.2.1. Frequency Division Multiple Access (FDMA)

In an FDMA based network, different frequency bands are assigned to users to continuously transmit data. Users require a precise local oscillator and frequency synchronisation. Otherwise, adjacent-channel interference due to carrier frequency shift may cause the failure of transmissions as shown in Figure 3.1. Many nodes in Table 2.1 use FDMA as their multiple access techniques. Orthogonal Frequency Division Multiple Access (OFDMA) is an advanced form of FDMA, which assigns subcarriers to different users to achieve simultaneous low data rate transmissions, but OFDMA has high peak to average power ratio (PAPR) which is energy consuming, resulting in its limited applicability in WSNs.

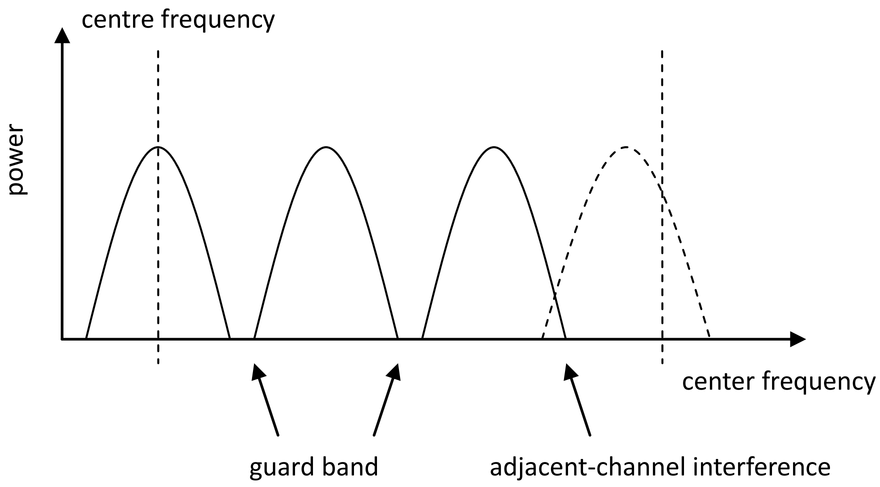


Fig. 3.1 An example of FDMA

3.2.2. Time Division Multiple Access (TDMA)

In a TDMA based network, unique time periods are assigned to users to avoid collisions. TDMA is commonly used in MAC protocols for WSNs. TDMA can assign time periods of different lengths to users to satisfy their various capacity requirements. The potential for asymmetric channel assignment is the major advantage of TDMA. TDMA requires a precise local timer for synchronisation, with large time shifts causing collisions as shown in Figure 3.2. Low Energy Adaptive Clustering Hierarchy (LEACH) [30] uses TDMA for in-cluster communication: nodes in the same cluster (except the cluster head) send data to the cluster head by TDMA. Further discussions about TDMA schemes are provided in later chapters.

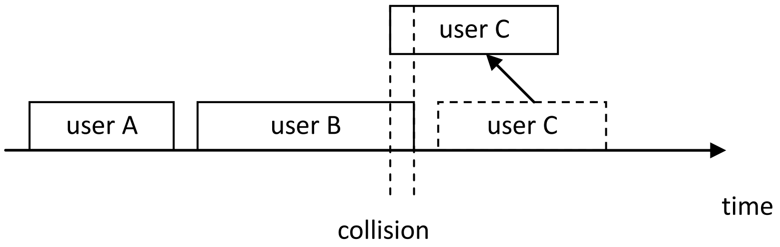


Fig. 3.2 An example of TDMA and collision

3.2.3. Code Division Multiple Access (CDMA)

CDMA assigns unique spreading codes to the users in the network, known also by the receiver. Users transmit data as spread spectrum signals and the receiver can demodulate particular signals by using the associated spreading code. The orthogonality among different spreading codes implies that other spreading codes are considered as random noise and filtered out at demodulation. CDMA allows a large number of simultaneous transmissions in the network, but it performs the best when the number of users is low. With an increasing number of users, the noise level at the receiver also increases, thereby reducing communication range within the channel, which is called the breathing effect of CDMA. IRIS [31] and Micaz [32] nodes use direct sequence spread spectrum to reduce RF (radio frequency) interference and provide data security.

3.2.4. Space Division Multiple Access (SDMA)

In traditional communication systems, base stations usually have no location information for the connected users. All downlink transmissions are radiated broadly in order to provide necessary coverage and ensure reception, which causes energy waste and interference. By applying smart antenna and beamforming techniques, base stations can focus the power of their signals in the directions of associated users, thereby improving energy efficiency and reducing interference. Considering the limited communication ranges and unknown physical positions of nodes in WSNs, SDMA is usually not applied to the design of WSNs.

3.2.5. Random Multiple Access

In distributed WSNs, users usually transmit at a low data rate with unpredictable packet arrival times. Random multiple access schemes support the variable arrival patterns of packets by contending at transmission time for access to the channel, thereby improving the channel performance when packet arrival times are unpredictable. Carrier Sense Multiple Access (CSMA) [33] and ALOHA [34] are representative random multiple access techniques. In CSMA based schemes, a user first listens to the carrier before transmitting any packets. If the carrier is not occupied, the user can send the data, otherwise it has to wait to avoid collision. There are two basic CSMA schemes: persistent and non-persistent CSMA. In persistent CSMA, if a user senses the channel idle, it sends data. If the channel is occupied, it waits until the channel is clear and sends data with probability p. In non-persistent CSMA, if the user senses the channel idle, it transmits the data. If the channel is busy it follows the retransmission delay distribution to schedule a delayed transmission (using carrier sense as well) [33]. ALOHA is another random access scheme, in which users access the channel immediately after the packets are generated [34].

3.3. MAC Protocols for WSNs

This section categorises MAC protocols for WSNs as contention-based, schedule-based, or the combination of both. In order to efficiently access the channel resources, certain types of MAC protocols can be matched to the WSNs with different topologies and applications. Considering flat routing (without control nodes such as a cluster head) protocols, centralised control is mostly not applicable. Therefore contention-based MAC protocols are a natural distributed solution with throughput and delay performance benefits through their relatively independent transmission scheduling and collision avoidance mechanisms. For hierarchical routing protocols, schedule-based MAC protocols can take advantage of the topology to use cluster heads for scheduling, thereby avoiding in-cluster collisions and achieving greater channel efficiency [35]. In long-term applications (such as environmental and agricultural monitoring tasks) with low data rates, contention-based MAC protocols achieve energy efficiency through their low overheads and avoidance of idle listening. For applications with heavy traffic load and real-time requirement (such as WSNs for fire rescue and battle field surveillance), schedule-based MAC protocols benefit delay and throughput performance. There are a huge number of MAC schemes developed for WSNs. This thesis selected the well established and represented MAC protocols to introduce in this literature review. Figure 3.3 categories some existing MAC protocols into different types. These protocols are introduced in the subsequent sections.

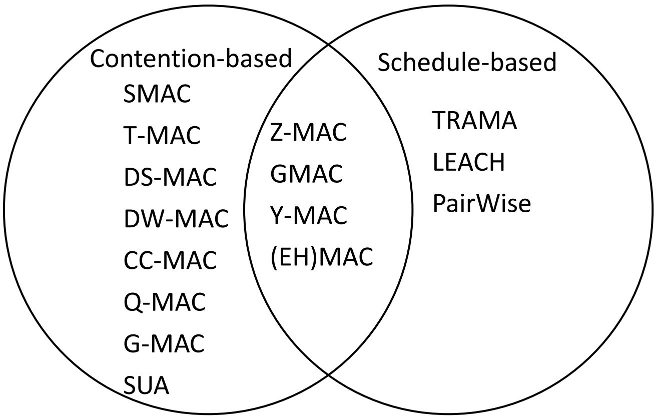


Fig. 3.3 Categories of some existing MAC protocols

3.3.1. Contention-based MAC Protocols

Contention-based scheme are a natural approach for distributed multi-hop networks and most of the contention-based MAC protocols are based upon CSMA. Sensor – MAC (SMAC) [36] has explored the classic periodic listen/sleep duty cycle which becomes the basis of a series of protocols. An overview of several representative contention-based MAC protocols is provided in this section.

A. SMAC

To improve energy efficiency and the lifetime of a sensor network, SMAC [36] is a well known contention-based scheme which periodically switches nodes between sleep and listening modes. It is designed for long-term non-real-time applications which prioritise network lifetime rather than throughput and delay performance. To avoid unnecessary energy consumption, SMAC applies three operations: period listening, collision avoidance and overhearing avoidance, and message passing.

The duty cycle of SMAC has listen and sleep periods. During the listen period, nodes exchange data and control information. During the sleep period, nodes initiate their scheduled transmissions and receptions or turn off their radio in order to save energy. Nodes are willing to use the same sleeping timetable as their neighbours to reduce control overheads [36]. However neighbouring nodes may have different sleep schedules because they have different nodes in their neighbourhood. Nodes receiving different sleep schedules will ensure they only sleep when permitted by all schedules. This ensures receivers do not miss any information, but leads to additional energy consumption.

In the listening period, nodes exchange schedule information and control packets. The data transmission is the same as IEEE 802.11 [36], which is contention-based and uses Request to Send (RTS) packets and Clear to Send (CTS) packets. When the first RTS packet arrives at an idle receiver, that node wins access to the medium and a CTS packet is returned to the winning node to indicate that it may now send. The synchronisation is done by exchanging SYNC packets which are quite short and include the sender’s address and its next sleep schedule.

Collision avoidance in SMAC is done by physical and virtual carrier sensing with RTS/CTS packets. A duration field is included in each transmitted packet which records the time the transmission will last. Once a node overhears a packet destined for another node, it knows how long the channel will be captured for and records this time in its network allocation vector (NAV) [37]. The NAV is a timer that is decremented by one every slot until it reaches zero. Every time a node wants to send data, it first checks the NAV, and the channel is considered busy if NAV is not zero. This is called virtual carrier sensing. The channel will not be considered clear unless both physical and virtual carrier sensing determine it is free.

In summary, SMAC improves energy efficiency by reducing idle listening, collision and overhearing. Synchronisation is performed by broadcasting SYNC packets. The collision of SYNC packets may cause synchronisation errors but since the listen and sleep periods are quite long, SMAC is not very sensitive to synchronisation problems. The predefined duty cycle introduces some delay and inefficiency to the network, however this can be improved as described in further research [38].

B. Timeout – MAC (T - MAC)

T-MAC is a contention-based MAC protocol based on SMAC which introduces adaptive duty cycling to reduce the energy consumption caused by idle listening [35]. Given that SMAC has a fixed duty cycle, the listen period must be long enough to handle the largest traffic level. If the output traffic level fluctuates significantly, the listening period will be too long for lower traffic loads and energy will be wasted by idle listening. The main idea of T-MAC is to reduce idle listening by transmitting all messages in bursts of variable length, and by sleeping between bursts.

In T-MAC, a node will end the listen period if it hears nothing and has no data to send after a certain time TA [39]. The threshold TA determines the minimum amount of idle listening in each duty cycle. The TA is longer than the sum of the lengths of the contention interval, the RTS packet and the propagation time to avoid missing any transmissions. T-MAC introduces Future Request to Send (FRTS) packets to solve the early sleeping problem. Early sleeping means that when a node fails to contend with another node, its destination node will switch to sleep mode for the rest of the current duty cycle. In T-MAC, a node sends a short FRTS to its destination node after it fails contending a slot. FRTS includes the time the source node will wake up, which is also the time the current transmission will last. The FRTS increases throughput but also energy consumption, therefore T-MAC has better energy efficiency under high traffic levels [39]. Compared with SMAC, T-MAC improves energy efficiency by reducing idle listening in the listen period, but has similar delay issues to SMAC.

C. Dynamic Sensor MAC (DSMAC)

The fixed duty cycle is always a problem in SMAC which affects delay and throughput performance and adaptability to different levels of traffic load. DSMAC [40] extends SMAC, introducing a dynamic duty cycle to achieve better balance between the latency and the energy consumption with different traffic levels.

Packets in DSMAC have a field which records the one-hop latency [40]. The one-hop latency is defined as the delay between the time a packet is generated and the time it is successfully transmitted. This value can be retrieved at the receiver node. The average latency is calculated to estimate the current traffic level. Nodes also keep track of their energy consumption per packet and which is delivered as an indicator of energy efficiency. If the latency of a node exceeds a certain level, it doubles its duty cycle by retaining the same listen period but reducing its sleep period. Similar to T-MAC, DSMAC has better performance than S-MAC under high traffic levels.

D. Demand Wakeup MAC (DW-MAC)

Demand Wakeup MAC (DW-MAC) [41] is a contention-based MAC protocol which applies a low-overhead scheduling algorithm to wake up the nodes when they need to transmit or receive in order to increase the effective channel capacity and adapt the network to a variety of traffic loads.

Similar to SMAC and T-MAC, DW-MAC has a duty cycle and neighbour nodes use the same sleeping schedule. One duty cycle includes three periods: Sync, Data and Sleep. Nodes exchange synchronisation information in the Sync period. The Data period is reserved for scheduling frames (SCH) and scheduled data is transmitted during the Sleep period. Nodes which do not need to transmit or receive switch to low power mode during the Sleep period [41].

In the Data period, nodes contend for channel access using CSMA/CA but instead of RTS and CTS, DW-MAC uses SCH for scheduling. Similar to RTS, SCH contains the destination address but not any timing information. The time that the node starts to transmit/receive SCH provides the scheduling information. The ratio of the length of SCH to the length of the data packet indicates the time that nodes start to transmit/receive packet, because it is the same as the ratio of the time interval between the start of the Data period and the start of the transmission/reception of SCH to the time interval between the start of Sleep period and the start of transmission/reception of data. So after a node successfully transmits/receives a SCH, a period of time in the Sleep period will be reserved for data transmission/reception [41].

DW-MAC introduces multi-hop forwarding. A node routes the SCH to the next hop after it receives the SCH from the previous hop. A data reception and data transmission period in its Sleep period are automatically reserved. Generally, DW-MAC integrates scheduling and access control to achieve low latency and high power efficiency, delivering performance benefits under high traffic levels. However, this feature also makes DW-MAC very sensitive to synchronisation errors [41].

E. Correlation-Based Collaborative MAC (CC-MAC)

In order to reduce redundant transmissions from nodes which detect the same events, CC-MAC [48] proposes improving energy efficiency by exploiting spatial correlation. CC-MAC is a contention-based cross-layer protocol which selects representative nodes within a certain radius to transmit data, and routes the representative data to the sink with a higher priority. CC-MAC has two sub-protocols, Event-MAC (E-MAC) and Network-MAC (N-MAC).

CC-MAC requires global information and network statistics for network initialisation. A spatial correlation model is proposed to determine the minimum number of representative nodes that achieve the distortion constraint given by the application layer [48]. CC-MAC introduces Vector Quantisation (VQ) and Iterative Node Selection (INS). These distributed processes determine the number and location of representative nodes using topology information, the network density and type of distribution. Only the representative nodes are allowed to transmit sensing data in the associated correlation region.

Packet transmissions in CC-MAC are based upon CSMA/CA, using RTS/CTS packets with embedded correlation information [48]. E-MAC filters out the correlated event information of the nodes in the same correlation region. When nodes detect an event, they contend for the channel using RTS/CTS packets. The winner becomes the representative node and continues transmitting data. Upon overhearing packets from representative nodes, other nodes calculate their distance to the representative nodes and determine if they are in the correlation region. Nodes in the correlation region switch to the sleep state for a certain time and nodes outside the correlation region keep contending for the channel if they have packets to transmit. Since the correlation is filtered out by E-MAC, packets from the representative nodes have priority over other packets in the network. N-MAC switches nodes in the random listening period in sleep mode into active mode when they hear RTS packets, to receive data packets from representative nodes.

CC-MAC therefore achieves energy efficiency by considering the spatial structure of the topology, but this also makes CC-MAC sensitive to topology changes. The correlation radius and representative nodes need to be re-determined once the network topology changes. The performance of E-MAC may also be affected by network dynamics. E-MAC performs different sleep schedules depending upon whether the correlation radius is larger than the reception range, or vice versa. Applying incorrect sleep schedules due to change in this parameter may block transmissions and cause additional energy consumption.

F. Quorum-Based MAC (Q-MAC)

The fixed duty cycle and frame sizes in SMAC cause inefficiency when handling different levels of traffic load. Q-MAC [50] introduces the concept of a quorum to adapt the sleep schedules to improve the energy efficiency and delay performance. The quorum-based wake up scheme determines the wake-up frequency by considering the current traffic load upon an individual node. Nodes with a light load switch into sleep mode more frequently. To reduce the delay caused by a long sleep period, Q-MAC considers the next-hop topology structure to increase the transmission opportunities when relaying packets

Q-MAC divides the network into adjacent coronas (concentric circles) centred at the sink by broadcasting packets with hop count during initialisation [50]. Frames are quantised to an $n \times n$ grid and $2n - 1$ frames (one row and one column) are randomly selected as the quorum frames in which the nodes switch to the active mode. Nodes with the same grid size definitely intersect so that they can maintain communication. To solve the problem of asymmetric traffic caused by the increasing relay burden from more central nodes, grid sizes are assigned to the nodes in different coronas based on the ratio between the areas of the circles. Therefore, nodes in the inner coronas have smaller grids and more frequent access to the channel.

To reduce the latency caused by sleeping nodes, Q-MAC allows nodes to obtain a list of nodes which are potentially their next hop (nodes of the inner-adjacent coronas). When a node has data to transmit, it first multicasts RTS packets to nodes in the list, and nodes receiving RTS packets send back CTS packets with a certain backoff determined by the remaining energy. Nodes overhearing data transmissions switch to sleep mode to avoid collisions and save energy. Q-MAC alleviates the asymmetric traffic conditions caused by packet relaying and improves the energy efficiency of both the MAC and routing layers. However it presents limits for the traffic model and applications. If the sensing field is located at the far side of the network to the sink and most of nodes in the network are relay nodes, different grid sizes may cause additional delays in the packet transmissions.

G. Game-Theoretic MAC (G-MAC)

Based on SMAC, [50] proposes an incomplete cooperative game-theoretic heuristic-based constraint optimisation algorithm to adaptively adjust the duty cycle to the current traffic in the network, thereby achieving better energy efficiency, throughput and delay performance.

In [50] a game-based optimisation algorithm is proposed to estimate the simultaneous competing nodes in the network. The algorithm requires complete schedule information and explicit cooperation of nodes. It is then simplified to adjust to WSNs by using virtual saturated traffic conditions (assuming packet collision/success based on carrier sensing rather than sending real packets) to satisfy the requirements of the algorithm without additional control packet overheads.

G-MAC achieves better energy efficiency and channel performance than SMAC. However the hidden terminal problem [36] in multi-hop networks may affect the performance of G-MAC. The virtual collision/success situation in virtually saturated traffic conditions is determined by local carrier sensing, which can be inaccurate when hidden terminals exist.

H. Schedule Unifying Algorithm (SUA) for SMAC

To solve the problem of high energy consumption of border nodes in SMAC, [51] proposes a schedule unifying algorithm for network-wide synchronisation of the sleep schedules. In SMAC, a node either follows a sleep schedule received from a neighbour or broadcasts its own sleep schedule. But if it receives multiple schedules it synchronises to all of them, which increases the length of the listening period and causes more energy consumption. These nodes are called border nodes. SUA introduces synchroniser ID in each SYNC packets and nodes follow the sleep schedule of the highest synchroniser ID. Nodes also record their neighbours' sleep schedules and unicast their own sleep schedule (after following the highest synchroniser ID) to the nodes with different schedules, therefore the network has the same sleep schedule after a certain time.

SUA enhances the sleep schedule of SMAC and alleviates the additional energy consumption of border nodes. However the convergence of the algorithm requires more time in larger networks, and SUA is incompatible with networks with multiple sink nodes.

3.3.2. Schedule-based MAC Protocols

Schedule-based schemes represent an alternative approach to the medium access control problem, although this is difficult to achieve in a distributed fashion. The improved throughput and delay performance are mostly offset by the increased complexity and overheads. Compared with contention-based MAC protocols, schedule-based protocols have better general performance at high traffic levels.

A. Traffic-Adaptive medium access control (TRAMA)

Traffic-Adaptive Medium Access Control (TRAMA) protocol is a TDMA-based energy-efficient schedule-based MAC protocol [37]. TRAMA is similar to Node Activation Multiple Access (NAMA) [42] which uses a distributed election algorithm to make sure only one node can transmit in two hops to avoid the hidden terminal problem. TRAMA is better than NAMA because it introduces a low-power mode to reduce energy consumption when the node is not transmitting or receiving. A difference with SMAC is that the listen/sleep schedule is not predefined but dynamically calculated from the current traffic level, thereby achieving better energy efficiency. It provides transmissions without collisions, idle listening and idle senders [37].

In TRAMA, nodes exchange their two-hop information and transmission schedules periodically, with transmitter and receiver nodes of each time slot determined by this information. The neighbour and transmission schedule information exchange is done by the Neighbour protocol (NP) and the Schedule Exchange Protocol (SEP). The selection of transmitters and receivers is done by Adaptive Election Algorithm (AEA), which can also switch other nodes into low-power mode [37].

The duty cycle of TRAMA consists of the random access period and the scheduled access period. The scheduled access period is seven times longer than the random access period [37]. The random access period includes signalling slots for the NP to exchange one-hop information. In the random access period, all transmissions are contention-based, with ACKs and retransmissions introduced to ensure the reception of information. Node additions, deletion and synchronisation are also performed in this period. In the scheduled access period, nodes use the information obtained in random access period and transmit packets without collisions.

The control packet overheads are a problem in TRAMA. At least 12.5% of the channel capacity is consumed exchanging control information. A certain type of hidden terminal problem also exits in TRAMA. Suppose a node has data to send and it has the highest priority in the current slot. The other nodes in its two-hop neighbourhood all switch to low power mode. At this moment, if a node three hops away has data to send to a two-hop neighbour, its transmission will certainly fail to be received because the destination node has turned off its radio [37].

B. Low Energy Adaptive Clustering Hierarchy (LEACH)

Low Energy Adaptive Clustering Hierarchy (LEACH) [30] is a schedule-based cross-layer protocol which reduces and evenly distributes the energy consumption of the nodes in the network. LEACH is a cluster-based protocol. Periodically, nodes in the network are randomly elected to be cluster heads. Each cluster head gathers and aggregates the data in its own cluster and sends the data back to the base station. After a certain period of time, nodes will elect new cluster heads. Periodically switching cluster heads reduces the consumption of single nodes and prolongs the lifetime of the entire network because a cluster head consumes more energy than the ordinary nodes.

The process of LEACH has two phases: the set up phase and the steady phase [43]. In the set up phase, nodes randomly select a number between 0 and 1. If the number is less than the threshold *T(n)*, the node becomes the cluster head in the current round. After the election, the selected cluster heads broadcast advertisements to all the nodes in the network to notify that they became the new cluster heads. Nodes receiving the advertisement will send back messages to confirm that they are members of the certain clusters. After receiving all the replies from the nodes belonging to the cluster, the cluster head will assign transmission times to the nodes based on the number of nodes in the cluster to ensure TDMA-based data transmission in the steady phase. The transmissions within each individual cluster use different CDMA codes to avoid interference between neighbouring clusters [43].

LEACH is difficult to apply to large-area applications, because every node has the possibility of being a cluster head and the cluster heads need to communicate with the base station, which requires that all the nodes are able to contact the base station. Nodes also should require the capabilities (and sufficient resources) to complete the potentially complicated task of a cluster head. The probability of being a cluster head is uniformly distributed. Therefore, in some rounds, cluster heads can appear on one side of the network and nodes one the other side of the network have no cluster heads in their communication range.

C. PairWise

PairWise [55] is a TDMA schedule-based MAC protocol which establishes transmissions between pairs of nodes and does not require global synchronisation. The channels (not frequency but time) of node pairs hop pseudo randomly in time according to a seed and maximum rendezvous period (MRP) to avoid collisions and adapt different traffic loads. Before transmitting data packets, nodes need to perform neighbour discovery, channel establishment and synchronisation.

Neighbour discovery is implemented by exchanging invite messages, channel request messages (CRM) and channel acknowledgment messages (CAM) [55]. After a node switches on, it first waits a certain time for an invite message. If no invite message arrives, the node broadcasts its own invite message which includes the MRP, available seeds and constants for calculating rendezvous periods (RPs) for uplink and downlink channels. Nodes receiving an invite message return a CRM which includes the selected seeds and constants for both nodes to calculate the RP (in which nodes wake up to exchange data packets) according to the channel establishment algorithm. Nodes receiving multiple invite messages do not return the CRM but broadcast their own invite message after certain time.

In the case that the same seeds result in the same RPs, to avoid collisions, PairWise provides a large number of available seeds in the invite message to reduce the probability of overlapping. Nodes also record the seeds used by their neighbours and broadcast them in their invite messages for nodes which overhear the messages to calculate the reserved RPs and select their own RPs with the lowest overlap. PairWise achieves good throughput, delay performance and very few collisions. However it requires constant exchange of two-hop information which increases the control packet overheads.

3.3.3. Other MAC Protocols

By combining the features of both contention-based and schedule-based protocols, MAC protocols can achieve better throughput, delay performance and improved energy efficiency. But these protocols usually have higher complexity and require more calculation, which also leads to additional energy consumption.

A. Z-MAC

Z-MAC [44] combines the advantages of TDMA and CSMA to achieve better channel utilisation and lower latency under different traffic levels. It has a setup phase of neighbour discovery, slot assignment, local frame exchange and global time synchronisation. Neighbour discovery is applied by broadcasting ping messages to the neighbours periodically [44]. After obtaining one-hop neighbour information, it is used as the input parameters to a time slot assignment algorithm DRAND [45]. DRAND can assign unique slots to the neighbour nodes and avoid the hidden terminal problem.

Nodes in Z-MAC can be in one of two modes: low contention level (LCL) or high contention level (HCL) [44]. A node is in HCL only when it receives an explicit contention notification (ECN) from a two-hop neighbour. In LCL, any node can contend to transmit in any slot. However, in HCL, only the owners of the current slot and their one-hop neighbours are allowed to contend for the channel. The owner of the slot always has the highest priority but if it does not have any data to send, other nodes can steal the slot [44].

Z-MAC is a complicated frame-based and collision-free MAC protocol which differs from other protocols. It has the advantages of TDMA under high traffic levels and the benefits of CSMA under low traffic levels. Compared to CSMA, it has similar performance under low contention levels and better performance when the contention level is medium or high [44].

B. Global Time Synchronised MAC (GMAC)

Global Time Synchronised MAC (GMAC) [46] is a contention-based and schedule-based MAC protocol using TDMA and FDMA. The network topology of GMAC is tree-based, with each node assigned to a parent closer to the sink node with the smallest number of children. Nodes only send data to their parents and the parents then relay data to the sink nodes.

GMAC divides the time into super-frames. Each super-frame includes a control frame and several data frames. In control frames, nodes broadcast their synchronisation information, selected frequencies, time slots, neighbour, children and parent lists to their neighbours using CSMA. In data frames, the child nodes wake up on their selected slots and send data on certain frequencies. The parent nodes wake up during their slots assigned to their children, and tune to the right frequencies to receive data [46].

GMAC uses a resource allocation algorithm to avoid collisions with neighbouring nodes. Generally, each node has a “blacklist” of resource units (frequencies and time slots) that it cannot use. When a node joins the network, it first listens and collects information to obtain its "blacklist". The resource units of its parent's neighbour nodes, its neighbours' child nodes and the time slots of its parents and brothers are recorded in the "blacklist" [46].

GMAC performs well when the traffic level is high, but its energy efficiency declines when the traffic level decreases. Latency is the biggest problem for GMAC especially when the sources are densely distributed. The "blacklist" is obtained at initialisation which can be out of date considering the dynamics of typical network, thereby limiting the throughput and delay performance.

C. Y-MAC

Y-MAC [47] is similar to GMAC, it divides time into frames which have a broadcast period and a unicast period, both of which contain a number of time slots. The number of slots in the unicast period is configurable and depends on the QoS requirements.

During the broadcast period, neighbour nodes contend to broadcast their control information. The occupied slots of the nodes' one-hop neighbours are included in the control information, so all nodes can obtain a slot allocation vector of their two-hop neighbours to avoid collisions and the hidden terminal problem. If a node does not receive the control information from a neighbour after certain time, this neighbour will be deleted to release the time slots [47].

Y-MAC uses frequency hopping to adapt to high traffic load situations. Each node has several available frequencies, of which one is considered to be the basic frequency. The exchange of control information only happens on the basic frequency. Data transmitted during the unicast period is usually on the basic frequency but can also hop to other frequencies. If too many packets are queued to the same destination, the nodes can hop to another available frequency and then contend to send the data [47]. The use of frequency hopping reduces the delay experienced by the packets.

By applying frequency hopping, Y-MAC maintains low delivery latency and high data reception rate when the traffic load is high, but under low traffic conditions, frequency hopping is usually unnecessary. Moreover, when sensor nodes are densely deployed, the control packet overhead becomes a problem for the energy efficiency.

D. MAC for WSNs with Energy Harvesting (EH)

Energy harvesting (EH) devices alleviate certain fixed energy constraints typically in conventional WSNs. The sustainable energy supply and temporary energy shortages bring new concerns for MAC protocol research. The work in [52] proposes dynamic-framed-ALOHA (DFA) for WSNs with EH devices which aim to improve the packet delivery probability.

Instead of using repeated frames with fixed length, DFA categorises nodes with new packets and certain remaining energy which suffer continuous collisions into a backlog set. The sink schedules extended frames for nodes in the backlog set to transmit their data. The size of the backlog set is estimated at the sink by using the distribution of the energy storage and prior channel outcomes without obtaining any information from nodes in the network [52]. DFA has low overheads and achieves good delivery probability. However the protocol is implemented based on single-hop networks which have no problems with network-wide scheduling. The protocol needs to be extended and applied to multi-hop networks where centralised control is not available.

3.4. Discussions and Conclusions

The MAC protocol is an important component in a WSN and has a significant impact on energy efficiency, network lifetime and channel performance. This chapter presents a brief overview of the existing MAC protocols and categorises them into contention-based, schedule-based and the combination of both. The general principle used in the MAC schemes explored is switching nodes between active and sleep modes appropriately to save energy, and ensuring transmissions and avoiding collisions by channel contending or scheduling.

SMAC explores the concept of a duty cycle, switching nodes between active and sleep modes periodically. A series of extended protocols are developed based on SMAC and CSMA to further improve the energy efficiency and adaptability. Contention-based MAC protocols usually contend for the channel by transmitting short RTS packets instead of long data packets and avoid collisions by channel sensing and overhearing. Contention-based protocols are natural approaches for low-traffic distributed networks and have the advantages of low overheads and delay, and the performance has certain robustness to topological changes. However, the performance of contention-based protocols degrades under high traffic levels, and collision avoidance through channel sensing is sensitive to hidden terminal problems. In this thesis we intend to develop a collision avoidance mechanism (without using CSMA) robust to hidden terminal problems to improve the performance of the MAC protocol for multi-hop networks.

Schedule-based protocols are alternative approaches for WSNs MAC protocols, although scheduling is difficult to achieve in a distributed fashion. TRAMA and LEACH are representative schedule-based protocols which ensure no collisions by exchanging two-hop scheduling information or in-cluster scheduling.

Schedule-based protocols alleviate the impact of collisions and achieve high throughput performance, but the overheads become an issue for energy efficiency especially under low traffic conditions. Some protocols combine the advantages of both contention and schedule-based schemes to improve energy efficiency and achieve good performance under different traffic conditions. However the complexity and overheads are still problems for overall performance. Simpler schemes with less overhead but similar performance are better alternatives for MAC protocols for WSNs. Instead of exchanging scheduling information within two-hop, later chapters of this thesis focus on developing schemes which only require communication between the associated transmitter node and receiver node pair, thereby alleviating redundant traffic in the network and achieving energy efficiency.

Instead of using the same protocols, MAC protocols for WSNs are usually application-specific because the network topology and application requirements have significant impact on the design of MAC protocols. Contention-based protocols are typically favoured for long term monitoring tasks because these applications usually have low traffic loads and prioritise network lifetime rather than throughput and delay performance. For applications requiring real-time data flows (such as fire rescue and battlefield surveillance), schedule-based protocols are better options because the good throughput and delay performance under high traffic load conditions.

4. Reinforcement Learning and ALOHA for Single-hop WSNs

Contents

This chapter presents the basic principles of applying Reinforcement Learning (RL) and ALOHA to implement MAC protocols for WSNs. Framed ALOHA is the basic protocol to which RL is applied. In this chapter we consider the performance of framed ALOHA with random access and compare it to the scheme with RL slot selection strategy. Results show that RL significantly increases the throughput and delay performance of ALOHA and achieves perfect scheduling with appropriate system configuration.

4.1. Introduction

Given the energy limitations of WSNs, MAC protocols for most of their applications usually focus on improving energy efficiency (unless the applications do not concern network lifetime). However, it is not necessary to seek better energy efficiency by sacrificing too much Quality of Service (QoS), or vice versa. Maintaining the balance between energy efficiency and QoS is crucial in developing MAC protocols for WSNs. In current MAC protocol research, there is a trend towards raising complexity to achieve better channel performance, at the cost of adding overheads. Improving performance is the right principle but the applicability of complex protocols to simple WSNs devices is questionable. The ALOHA schemes have low complexity and overheads, but suffer poor throughput performance as a result of the blind transmission strategy (only 37% of channel can be used for useful throughput when the offered traffic is equal to the channel capacity). This chapter shows that by applying an intelligent slot selection strategy, ALOHA can become a promising MAC protocol which has low complexity and overheads while maintaining good channel performance.

This chapter introduces the basic principles of a novel approach which applies RL to ALOHA schemes. In framed ALOHA, weights are given to each slot in the repeating frame and updated by successful transmissions or failures. Nodes determine their preferred slots using the weights and optimal slots can be found when RL converges. When the frame size is appropriately configured, perfect scheduling can be achieved at convergence.

4.2. Reinforcement Learning on Framed ALOHA

RL is the process through which a node determines its optimal transmission schedule by learning from its transmission history. We first implement RL based on Framed ALOHA with exponential backoff and maximum retry limit (EB-F-M) described in the appendices. In EB-F-M, the transmission slot is selected in two steps: first randomly select a frame in the current contention window then schedule a packet transmission in a random slot in that frame. Exponential backoff with Reinforcement Learning (EB-RL) has a similar two-step structure but a different second step. The in-frame slot selection is performed based upon the results of the RL process. It has a similar backoff scheme to the original EB-M when a packet collides, EB-RL exponentially backoff the retransmission and it also has a maximum retransmission limit (EB-RL-M) of 6 according to the IEEE standard [74].

Specifically, each node in EB-RL records an array of weights, one per slot, based on the number of slots in the repeating frame. The weights represent the preference of transmitting in each slot in a frame. If a collision occurs in a certain slot, the associated weight is decreased by one, and if a packet is successfully transmitted in a slot, then that weight is increased by one. At the beginning of each contention window, the slot with the largest weight is selected as the transmission slot. If multiple slots have the largest weight, the node randomly selects one from them. The pseudo code below describes this learning algorithm. A scheduled slot denotes the preferred slot selected to send a data packet, and a scheduled frame indicates the frame when the node has the contention window size of one frame, or the frame selected to send a data packet when the node is in retransmission mode (contention window size larger than one frame).

```
loop sleep until the next scheduled frame
        if multiple slots have the same greatest weight
           randomly select one of these slots as scheduled slot
        else
           scheduled slot = the slot with the greatest weight
```

```
    sleep until the next scheduled slot
    if one or more packets in the queue
        send the packet at the queue head
    if ACK successfully arrives
        return +1 reward and update the associated weight
        contention window = 1 frame
        retry number = 0
    else
        return -1 reward and update the associated weight
        contention window = contention window * 2
        retry number = retry number + 1
        if retry number > retry limit
            contention window = 1 frame
            discard the packet
        else
            scheduled frame = a random frame in the current contention window
end loop
```

Figure 4.1 shows an example of EB-RL. In this example one frame contains two slots and the initial contention window is one frame. In frame 1, the node randomly selects slot 1 to send a packet and the transmission succeeds. In frame 2 and 3, the node sends a packet in slot 1 because it has the largest weight. The transmission in frame 3 fails, the node enters retransmission mode and doubles the contention window to two frames which are frame 4 and 5. Frame 5 is randomly selected and the node sends a packet in slot 1 because it still has the largest weight. The transmission fails again and the contention window becomes four frames (frame 6 to 9). Frame 8 and slot 2 are randomly selected and the transmission succeeds. The node starts a new contention window with the initial length at frame 9 and then sends a packet in slot 2.

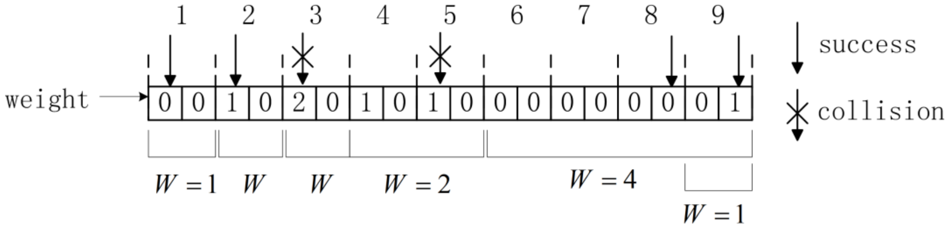


Fig. 4.1 An example of EB-RL

The principle of RL described above is simple and novel, and it requires no additional packet exchange with other nodes compared with the original framed ALOHA. Based only on whether the ACK (acknowledgement packet) is received or not, RL updates the weights and determines future slot selections. With appropriate frame length, RL can achieve perfect scheduling and significantly increase the throughput performance. The simplicity and low computation of the RL scheme make it ideally suited to WSNs (especially to long-term monitoring applications), and this RL scheme can also be an alternative to other distributed networks with similar requirements.

4.3. Modelling and Performance Results

In this section, the performance of EB-RL-M is evaluated by starting with a simple but commonly used single-hop network topology. This topology can also represent a one-hop sub-network or in-cluster communication in hierarchical WSNs (eg. WSNs using LEACH). As the initial evaluations, this section fully explores the potential of EB-RL-M under various environmental conditions (different propagation delays and synchronisation shifts) and considers its optimal configuration.

4.3.1. Topology and Assumptions

We consider a single-hop network in which nodes only communicate with the sink. Consider the worst case of packet reception that any partial or entire overlap of multiple packets causes the failure of these packets. All nodes maintain synchronisation. It is not necessary to synchronise the start of each frame among nodes because RL can perform properly as long as the start of each slot is synchronised. A full set of simulation parameters can be found in Table 4.1. Each data packet consists of 1024 bits for data, 8 bits each for destination/source address and 4 bits for packet type. Each ACK packet consists of 8 bits each for destination/source address and 4 bits for packet type.

The results in this chapter are collected through the whole simulation including the learning process. To show the best performance of EB-RL requires consideration of the performance after the system converges. Different frame sizes are considered in the simulations, and simulations with small frame sizes may not have optimal convergence because there are insufficient slots for all nodes to have their unique slots (the network can easily be overloaded). So to be fair to all schemes, results of all simulation time (from the start of each simulation to the end) is considered. In Chapter 5 and Chapter 6, the optimal frame size is considered and therefore results are collected after the network converges. Later chapters provide more study on convergence behaviour and this section mainly focuses on channel performance evaluation.

Parameter	Value
Transmitter Data Rate	250 kbits/sec
Receiver Data Rate	250 kbits/sec
Data Packet Size	1044 bits
ACK Packet Size	20 bits
Slot Length	0.0044 sec
Backoff Factor	2
Maximum Retry Limit	6
Simulation Period	500,000 slots

Table 4.1 Simulation Parameters

4.3.2. Traffic Models

Two traffic models are used to evaluate the performance of EB-RL-M. A saturated traffic model (nodes always have packets to send) is used to determine the performance of EB-RL-M under full contention. The other traffic model is more practical, in which the packet inter-arrival time is exponentially distributed. All nodes in the network have the same average packet inter-arrival time (same traffic level). The overall generated traffic and offered traffic of the network can be controlled by setting different packet inter-arrival times, thereby evaluating the performance of EB-RL-M with different traffic levels.

4.3.3. Performance with the Saturated Traffic Model

Figure 4.2 compares the normalised throughput between EB-RL-M and EB-F-M with the saturated traffic model. The results show that EB-RL-M has a much higher maximum throughput than EB-F-M and the peak point usually exists when the initial contention window W is close to the number of nodes. When the number of nodes is much larger than the frame size, EB-RL-M has similar performance to EB-F because there are insufficient slots for all the nodes, therefore transmissions keep failing and the network acts like random access. When the number of nodes is smaller than the frame size, EB-RL-M has lower throughput (but still better than EB-F-M) because a node can send one packet each frame and there are always unused slots in each frame. The improvement in maximum throughput with EB-RL-M is significant, considering that it is over 0.9 Erlangs which is over twice the maximum throughput of EB-F-M (0.37 Erlangs) and close to the theoretical maximum (approximately 0.95 Erlangs considering 5% overheads).

To validate the connection between frame size and number of nodes, Figure 4.3 compares the throughput and offered traffic when W is set the same as the number of nodes. It shows that EB-RL-M has throughput over 0.9 Erlangs for most traffic levels. The throughput of EB-RL-M is very similar to the offered traffic. This means that most packets are received without collisions, while about 40% of the packets are lost in EB-F-M. The lower throughput with higher numbers of nodes is caused by the longer convergence time, because in this chapter the results are collected from the whole simulations considering the fairness comparison (schemes with frame sizes lower than number of nodes may not converge to a steady state). In later chapters we will focus on the results of two different states (learning/convergence states). The throughput does not reach 1 Erlang because a certain proportion of each slot is reserved for ACK packets and other overheads. In the simulations, the slot length is set to the time needed to transmit/receive 1100 bits of data. The size of a data packet is 1044 bits, therefore the maximum possible throughput is approximately 0.95 Erlang. In Figure 4.3, EB-RL-M approaches the theoretical maximum when there are few nodes in the network. EB-RL-M also has a much lower average delay than EB-F-M under this situation as Figure 4.4 shows. For all different network sizes, the delay is kept under 1 second. Both schemes have the trend of increasing delay with the rising number of nodes, which is caused by larger frame sizes.

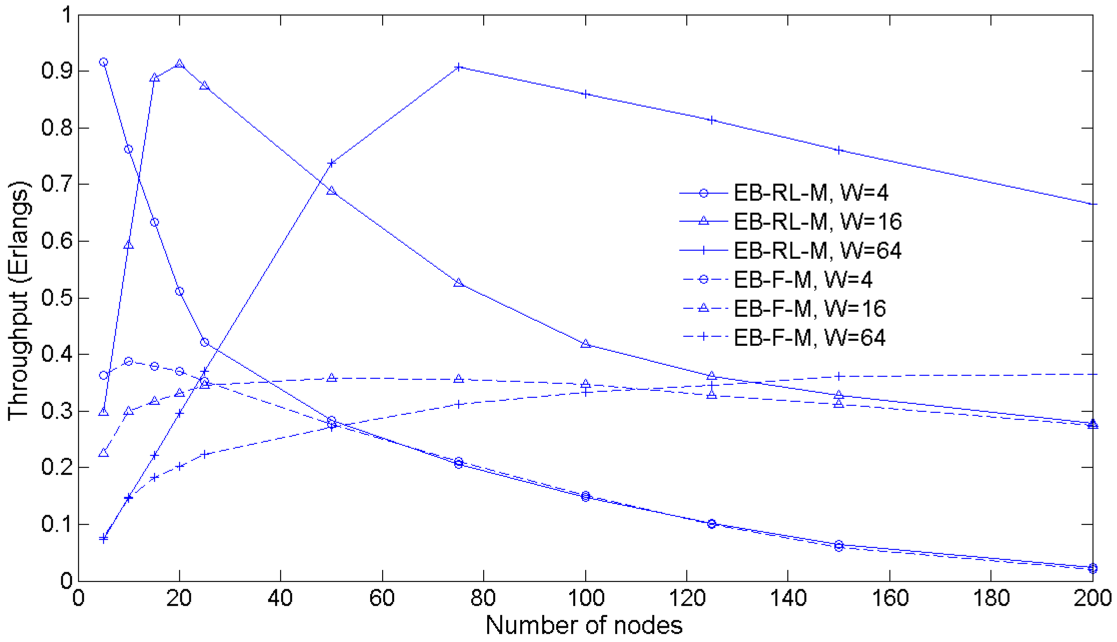


Fig. 4.2 Throughput of EB-RL-M with saturated traffic model

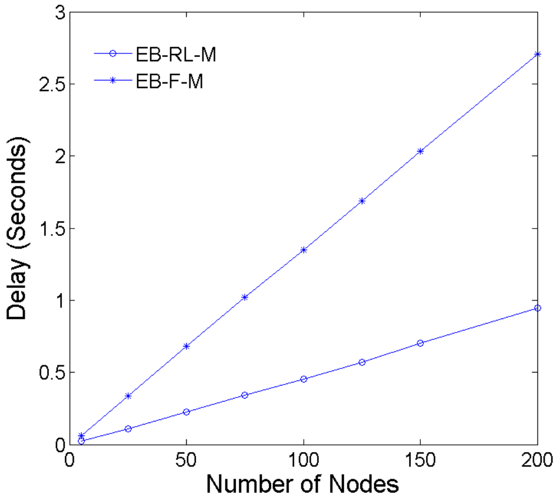


Fig. 4.3 Delay of EB-RL-M

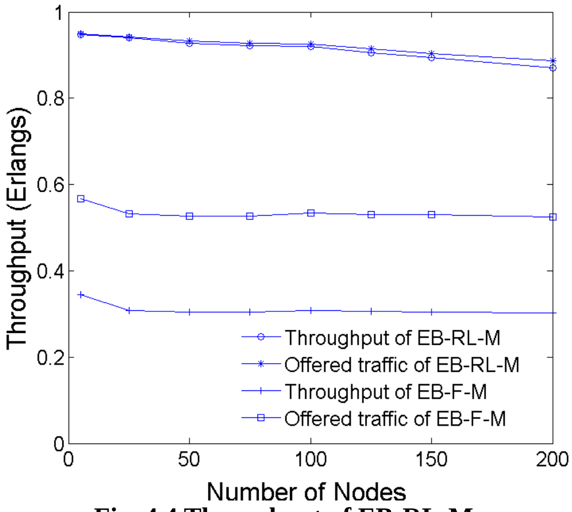


Fig. 4.4 Throughput of EB-RL-M

4.3.4. Performance with unsynchronised Conditions

In this section the performance of EB-RL-M with synchronisation shifts is evaluated to testify its robustness. After initialisation, nodes start the first slot at a uniform randomly distributed time, within the range of zero to the length of one frame. Under unsynchronised conditions, the theoretical probability of collision for EB-F-M is double the collision probability of synchronised conditions. Therefore, the expected throughput can be reduced by half. However, according to the collision avoidance performed by RL, a less than half throughput loss for EB-RL-M can be expected compared with EB-F-M. Figure 4.5 shows the throughput with synchronisation shift under the saturated traffic model. In accordance with the theory, the throughput of EB-F-M decreases to about half of the throughput in Figure 4.2 with the maximum at about 0.19 Erlangs. The throughput of EB-RL-M drops but is still more than half of the throughput shown in Figure 4.2, as expected. EB-RL-M has a maximum throughput of about 0.64 Erlangs when the frame size is set appropriately. Considering the existence of backoff, nodes enter backoff mode more frequently under unsynchronised conditions, which declines the offered traffic of each node and allows RL to select slots with relatively lower probabilities of collision across frames. Therefore the throughput loss is less than half compared with the EB-F-M.

Figure 4.6 compares the delay of EB-RL-M and EB-F-M under unsynchronised conditions. Similar to the throughput results, EB-RL-M has a better delay performance than EB-F-M when the number of nodes is not much larger than the frame size. Under the conditions of heavy contention and small frame size, EB-RL-M has similar performance as EB-F-M, because RL acts like random access and is difficult to benefit throughput with large probability of collisions. The results in Figures 4.5 and 4.6 demonstrate that EB-RL-M has certain robustness to synchronisation errors and still achieves necessary performance without synchronisation.

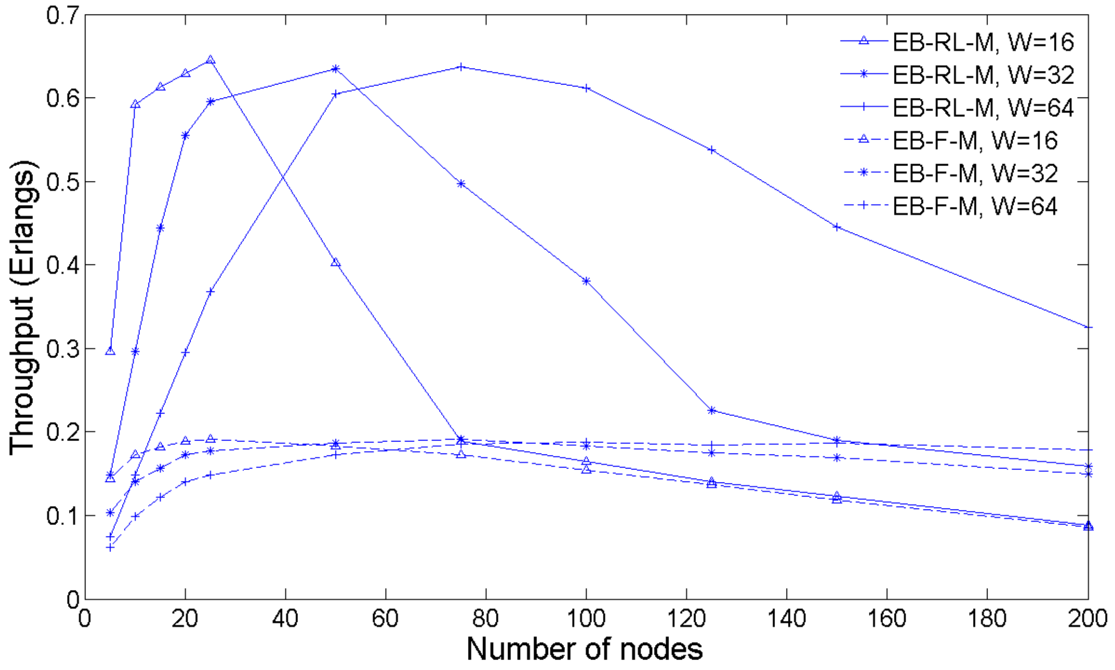


Fig. 4.5 Throughput of EB-RL-M under unsynchronised condition

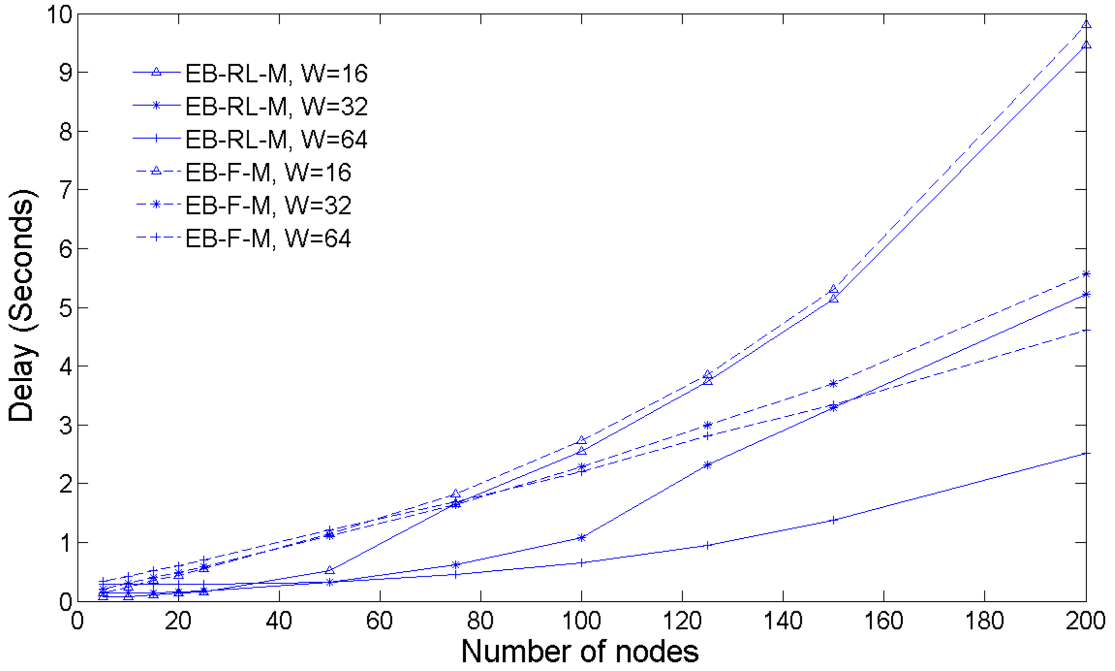


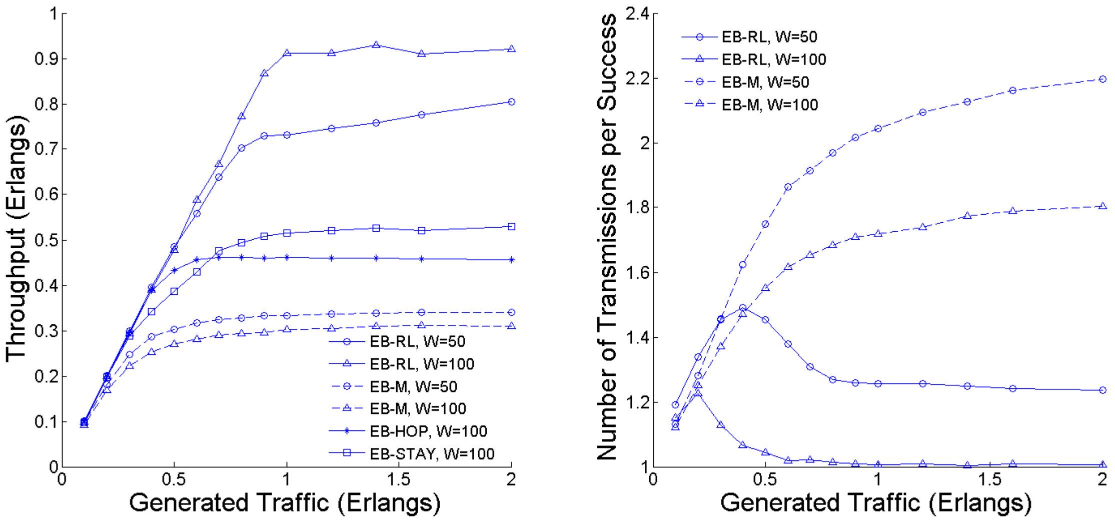
Fig. 4.6 ETE delay of EB-RL-M under unsynchronised condition

4.3.5. Performance with Exponentially Distributed Traffic Model

In this section we assume the network is synchronised, consists of 100 nodes and that the generated traffic is evenly allocated across all the nodes in the network (as a common WSNs environmental monitoring application). In addition to the original EB-M, two more frame-based EB schemes (EB-HOP and EB-STAY) with certain intelligence are also simulated for comparison. Nodes in EB-HOP starts with random access, keep using the slot with the first success until a packet transmission fails, and return to random access. Nodes in EB-STAY keep using the slot (forever) in which the first successful transmission occurs. Through comparison with these schemes, it shows the benefits of RL and the performance that RL can achieve.

Figure 4.7(a) compares the normalized throughput of four EB schemes with different initial contention window sizes. The results show that EB-RL has much higher throughput than the other schemes and the best performance exists when the initial contention window size is the same as the number of nodes. The throughput increases linearly with the generated traffic. In Figure 4.7(a), when the initial contention window size is set to 100, it achieves throughput of 0.92 Erlangs. When the generated traffic is equal to or larger than 1 Erlangs, this performance is quite close to the theoretical maximum of 0.95 Erlangs. It also shows that the configuration of the initial contention window size is very important for the EB-RL scheme. The original EB-M achieves maximum throughput of 0.32 Erlangs, the two less intelligent schemes (EB-HOP and EB-STAY) achieve maximum throughputs of 0.45 Erlangs and 0.51 Erlangs respectively. Compared with these schemes, EB-RL has much higher throughput performance but no additional overheads.

Figure 4.7(b) compares the energy efficiency between EB-RL and EB-M by calculating the average number of transmissions per success. When the contention level is relatively low, EB-RL has similar energy efficiency as EB-M. However with increasing generated traffic, the energy efficiency benefits from EB-RL become more and more significant. The figures show that with the initial contention window size of 100, the average number of transmissions per success is very close to 1 when the generated traffic is more than 0.5 Erlang. These results show that EB-RL generally obtains much better energy efficiency at medium and high contention levels. The larger energy cost at low traffic levels is caused by the larger convergence time (because of fewer learning opportunities). It is worthy to be noted that there is a peak of the number of transmissions per success for EB-RL with 100 frame size, and this effect is amplified when the frame size is 50. This effect is caused by the balance between the lower probability of collision under low traffic levels and more learning opportunities provided by high traffic levels. Both extreme conditions (low and high traffic levels) offer the RL faster convergence and benefit the energy performance, which cause the effect in Figure 4.7(b). Chapters 5 and 6 will provide a comprehensive study of the convergence and present a solution for different convergence times under various traffic levels.



(a) (b)

Fig. 4.7 Throughput and efficiency of EB-RL

4.3.6. Performance with Various Propagation Delays

In previous simulations, the propagation delay is assumed to be zero (in a typical WSN the propagation delay is less than 10^{-6}s considering neighbour nodes are 100m apart). In this section we evaluate the performance of EB-RL-M with different propagation delay to testify its robustness. The network consists of 50 nodes and the frame size is set to optimum (50 slots/frame). The maximum propagation delay is calculated from the ratio of the time needed to transmit or receive a data packet. For example, propagation delay of 0.1 represents 0.1 times the length of a data packet. The propagation delay of each packet is uniformly distributed between 0 and the maximum propagation delay. For different maximum propagation delay levels, the slot length is enlarged by twice the maximum propagation delay to reserve enough time for the transmission and reception of the data and ACK packets. According to the simulation parameters, the theoretical maximum throughput can be calculated as:

$$throughput = \frac{1044}{1100 + 2 \times 1044 \times maxpropdel}$$

where 1044 is the data packet length, 1100 is the original slot length and $maxpropdel$ is the maximum propagation delay ratio. By using the equation, the maximum throughput of five different maximum propagation delay levels can be obtained as 0.9314, 0.8668, 0.6879, 0.4437 and 0.3275 Erlang, which are shown as the dashed lines in Figure 4.8. The results clarify the robustness of EB-RL to propagation delay and that its performance is still close to the theoretical maximum.

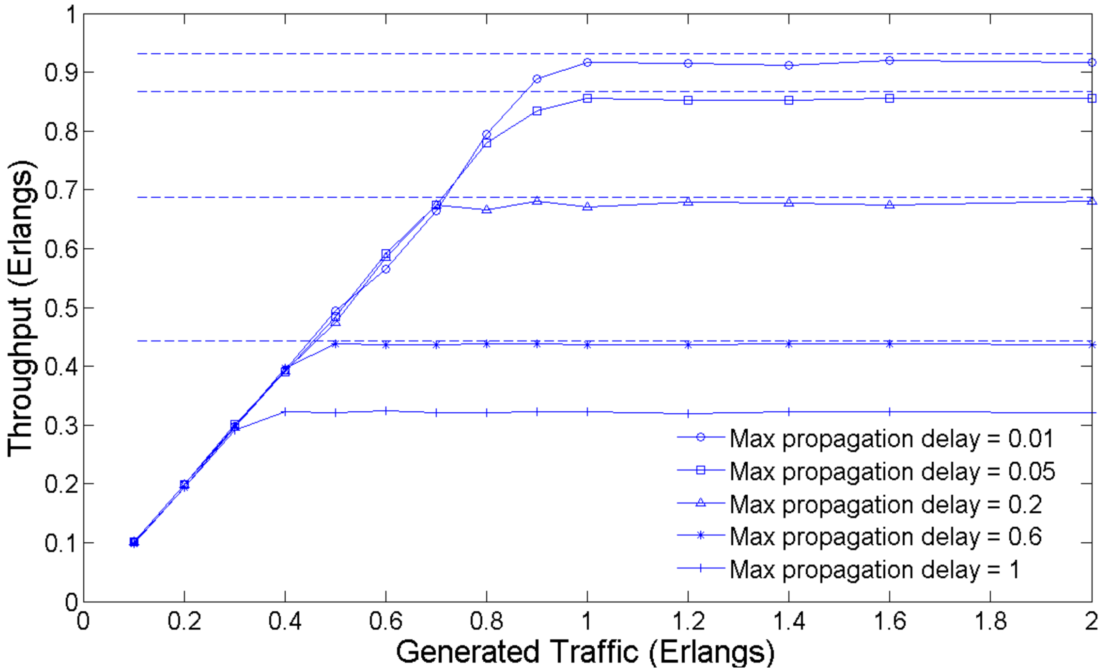


Fig. 4.8 EB-RL-M with different propagation delay

4.4. Conclusions

Due to the issues arising from increasing overheads and complexity, ALOHA becomes suitable alternative MAC protocol for WSNs, motivated by its advantages of simplicity and low overheads. However the ALOHA schemes suffer low throughput problems as a results of their blind transmission strategy. This chapter introduced the principles of implementing RL on framed ALOHA as an intelligent slot selection strategy, thereby improving the throughput and delay performance. Results show that EB-RL-M significantly improves the throughput and delay performance without additional overheads, and that the performance is robust under various conditions (with different traffic models, synchronisation shifts and propagation delay).

The basic idea of EB-RL-M is to assign weights to each slot of the repeating frame in framed ALOHA. Weights are updated by success and failures, and the slots with higher weights are selected. However, the weight can become a large value after many successful transmission, which may affect the performance after the environment changes (the preferred slot becomes unavailable). In later chapters, Q-Learning is applied as the slot selection strategy. The weights (Q values) of Q-Learning are bounded by the rewards to prevent the nodes from using sub-optimal slot selections.

The RL slot selection strategy only requires ACK packets without the overhead of any additional transmissions or fields in any packets (ACK packets are also required in any other MAC protocols including retransmission). Results show that the maximum throughput reaches 0.92 Erlangs (the theoretical maximum is 0.95 Erlangs) which is over twice compared with the scheme without intelligence (EB-F-M). It is notable that, for fairness, the data is collected from the whole simulation including the learning period (during which the performance is not optimum). A comprehensive study is provided on these separated network states (learning/convergence) in later chapters.

The frame size (initial contention window size) is an important parameter to EB-RL-M, which has a significant impact on the performance. Results indicate that the optimal frame size should be equal to the number of nodes in the network, so each node can own a unique slot after the RL algorithm converges. Setting the frame size too small will cause the network to never reach an optimum convergence. However, if frame size is larger than the number of nodes, RL can still converge but the maximum throughput decreases. In a practical WSNs task, the network size is usually known to the users. However, if the size of a network is unknown, it is better to set the frame size large enough to suit the largest expected network size.

Simulations in this chapter evaluated the performance of EB-RL-M with saturated traffic conditions, exponentially distributed traffic conditions, unsynchronised conditions and various propagation delay conditions. EB-RL-M significantly improves the performance under different condition compared with EB-F-M. Results bring confidence that it is a promising MAC technique for single-hop networks. In later chapters, similar protocols are applied to multi-hop networks, and more study is conducted upon its convergence behaviour.

5. Convergence Behaviour of RL Algorithms

Contents

In previous chapters, greater emphasis was placed on steady state performance because it shows the fundamental capability of our protocols. On the other hand, the convergence time is also important, since it indicates the initialisation time of the network and the network performs sub-optimally before convergence. The convergence time is related to many factors such as the learning algorithm, learning rate, returning reward and environment. To a WSN, the network topology, communication range of the nodes and traffic level all have an impact on the convergence time.

5.1. Introduction

Based on the experience of prior protocol implementation, ALOHA-Q is proposed in this chapter as a simple MAC protocol for single-hop WSNs which consists of the basic Q-Learning and slot selection algorithm described in Chapter 4. ALOHA-Q is applied to study the convergence behaviour of such schemes. A Markov model is implemented for the scheme based on the assumptions of a certain learning rate and starting Q values (learning rate is 1 and the Q values start at -1). The Markov states are built based on the number of nodes which have found their preferred slots, and state transitions are considered every slot. The convergence to the optimal steady state (all nodes find their preferred slots) is proved by calculating the state transition probabilities over infinite time. The expected convergence time is obtained from the Marko model and simulations are implemented to match the results. Results on real-time throughput are also provided during the RL process to show the performance of the scheme while learning.

5.1.1. Introduction to Markov Chains

Markov Chains and Markov processes were first implemented by Russian mathematician Andrey Markov in 1906 [63]. Different forms of Markov Chains are widely used for analysis in physics, chemistry, information science and queuing theory [64]. Discrete-time discrete-state Markov chains are the type mainly considered in this thesis, and will be used in this chapter to describe a learning process and obtain the expected convergence time.

A Markov Chain is a mathematical system which usually consists of a finite number of states with transitions between them [65]. These states are usually represented by numbers for the convenient expression of state transition probabilities. For example, $p_{i,j}$ represents the one-step transition probability from state i to state j, and ${p_{i,j}}^{(2)}$ represents the two-step transition probability (via two transitions) from state i to state j. The transition probabilities are numbers between 0 and 1, and the transition probabilities out from a state (including the probabilities of returning to the same state and transferring to other states) must sum to 1 [66]. In this thesis we consider a Markov Chain as a memoryless random process in which the next state only depends on the current state and the associated state transmission probabilities. A Markov chain in which the transition probability from one state to itself equals to unity is defined as an absorbing Markov chain and the associated state an absorbing state [67]. This definition is used later in this chapter.

As an example, a Markov chain can be used to describe the customer checkout lanes in a supermarket store. Assuming that the average customer arrival rate is λ (number of customers per second) and the average service rate is τ (customers per second). Suppose the customers do not select the checkout lanes, with 5 other customers queuing. By using the number of queued customers as the states and considering the state transitions in every second (assuming only one event can occur in one second), a Markov Chain can be formulated to describe these events. Figure 5.1 shows the Markov chain with associated state transitions and transition probabilities.

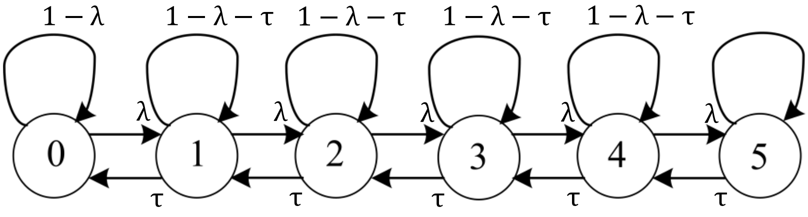


Fig. 5.1 An example of Markov chain

5.2. ALOHA-Q Protocol Design

5.2.1. Basic Principles

A similar scheme to those described in Chapters 4 is used. Nodes in the network all start with random access (all Q values are 0), learn through transmission, and finally reach their optimal transmission strategy in which the majority of nodes have found unique slots and subsequently experience few collisions. Q values are denoted $Q_t(i,k)$, indicating the current preference of node *i* to transmit a packet in slot k. The previous Q values and current reward contribute to the future Q value ($Q_{t+1}(i,k)$) update according to equation (5.1) after every data packet transmission:

$$Q_{t+1}(i,k) = Q_t(i,k) + \alpha\big(r - Q_t(i,k)\big) \tag{5.1}$$

Where $\alpha \in (0,1]$ is the learning rate and$r \in \{\text{-}1,1\}$ is the current reward.

If succeeds data packet and the associated ACK are both correctly received, a reward of +1 is returned otherwise the reward is -1. Slots with higher Q values are preferred but if multiple slots have the same higher Q value, one (or more) will be randomly selected from the set. The pseudo code below demonstrates this learning algorithm.

```
loop sleep until the next frame
        if multiple slots have the same greatest weight
           randomly select one of these slots as scheduled slot
        else
           scheduled slot = the slot with the greatest weight
        sleep until the next scheduled slot
        if one or more packets in the queue
           send the packet at the queue head
        if ACK successfully arrives
```

```
            return +1 reward and update the associated Q value
            retry number = 0
        else
            return -1 reward and update the associated Q value
            retry number = retry number + 1
            if retry number > retry limit
                discard the packet
end loop
```

Figure 5.2 shows an example of the frame structure and Q-Learning algorithm. One of the Q values of a node is updated after each data packet transmission. Idle listening is not used in ALOHA-Q (except the sink node). Time references for synchronisation are embedded in the ACK packets sent from the sink node, so that the transmitting nodes are able to maintain synchronisation with the sink node as long as they transmit data packets to the sink and receive ACK packets.

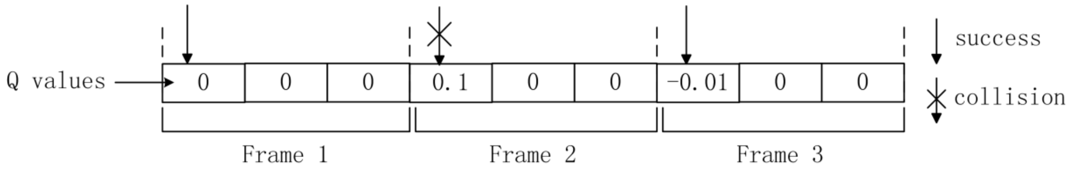


Fig. 5.2 Example of Q values and the repeating frame

5.2.2. Learning Process and Steady State

The learning process results in a node having different Q values for each time slot in the repeating frames. According to (5.1), a negative reward will have a greater impact on the current Q value, when the Q value is positive and vice versa. A slot which regularly receives a negative reward is therefore unlikely to be the preferred slot. This results in the node seeking a slot which will continually return a positive reward. Through this learning process the network tends towards to the optimal steady state condition where all nodes have unique slots. It behaves like a schedule-based network but without the need for scheduling information exchange or determination of node priorities in each slot, which is critical in WSNs without centralised control.

To reach the optimal steady state, the parameters need to be appropriately set. The learning rate $\alpha \in (0,1]$ controls the speed at which a Q value converges to the current reward. The higher α, the faster the Q value converges to reward r. α is usually set to a small value so that the steady state will exhibit some robustness to small changes in channel conditions (e.g. infrequent collisions). Another important parameter is the frame size N. Nodes will be unable to find unique slots if there are insufficient slots in a frame. N should be just large enough to ensure this. Overestimating N will introduce additional latency and reduce the maximum achievable throughput. In a single hop network, the number of nodes deployed is often known. In a multi-hop network, an appropriate value for N can be determined by estimating the number of nodes within communication range, based on the deployment density.

The learning algorithm is able to adapt to changes in the network topology should nodes die or additional nodes be deployed. When a node dies, its preferred slots automatically become available for others to use. New nodes will learn from scratch, but reach the optimal steady state much more quickly than if the whole network is initialised, because they are learning from a steady environment and can more easily find unique slots. The learning schemes can achieve perfect scheduling in the steady state following a period of convergence. The time taken to converge is important to the network and it varies due to the random slot selection process when multiple slots have the same maximum Q value. In the next section an analytical model is developed to estimate the convergence time of a simple learning process.

5.2.3. Fairness on Quality of Service

When the network converges to the steady state, nodes achieve perfect scheduling and experience an equal Quality of Service (QoS). However, during the learning process, nodes usually find their preferred slots sequentially, which means that some nodes obtain better QoS earlier than others, resulting in an imbalance in fairness between the nodes. Such fairness issues exist in the majority of the learning based MAC schemes, for example the two intelligent CSMA schemes proposed in [81] and [82]. By applying slots and repeating frames to CSMA, the scheme described in [81] starts with random access, and users continually select slots with successful transmissions until two consecutive collisions. Similarly, the scheme presented in [82] is initialised to random access, and the user keeps using the slot with correct packet reception until a collision, then this slots have a decreasing probability of reselection (other slots have equal probability of selection). In both schemes, nodes obtain their slots after different periods of time, and nodes have different probabilities of success before the network converges, which makes the network convergence time an important measurement.

Compared with the learning schemes proposed in [81] and [82] which have an equal probability of selecting other slots when a collision occurs in one slot, Q-Learning has certain advantages and benefits from recording the transmission history. When only a minority of nodes has not converged, they have experience with regard to slot choice based on their transmission history, and the probability of them finding their preferred slots increases as time passes. Consider, for example, a network with 100 nodes where 99 nodes have found their preferred slots and one node is searching for its own unique slot. Assuming the worst case that it has no prior transmission experience (all Q values are zero), it needs a maximum of 100 transmissions (99 collisions and one success) to find its preferred slot and the probabilities of locating its preferred slot during each transmission are 1/100, 1/99, 1/98…etc until the 100th transmission is reached and the probability becomes 1. In later sections a Markov model is presented to estimate the convergence time and present simulations results of convergence time with different learning rates. Note that the convergence time is that measured when all nodes in the network find their preferred slots (not the average convergence across all nodes).

5.3. Convergence Time of ALOHA-Q

5.3.1. Markov Model

For the theoretical analysis, we consider a single-hop network with *N* nodes and saturated traffic conditions (nodes always have packets to transmit). The learning rate (α) is set to 1 and the Q values are all initialized to -1, so that they can either be +1 or -1. The frame size is set equal to the number of nodes *N* and each node is allowed to transmit one packet per frame. Figure 5.3 depicts the Markov process which describes this scheme. State *k* represents the current number of steady-state nodes (nodes which have a Q value of +1 for a particular slot) in the network. State transitions take place in every slot, and the process can only move forwards or backwards or stay in the same state after each slot. A node which has reached steady-state is referred to as a *steady* node whereas a node still learning is referred to as a *hopping* node. When the process absorbs at state *N* (the absorbing state), all nodes have found their unique slots and the system is deemed to have converged.

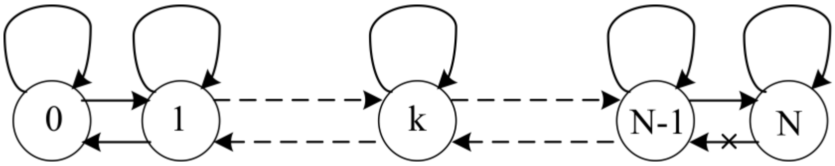


Fig. 5.3 Markov model of ALOHA-Q

Let $p_{i,j}$denote the state transition probability from state $i, i = 0, 1, 2 \dots N$ to state $j, j = 0, 1, 2 \dots N$. For the model in Figure 5.3, the relevant transition probabilities are: $p_{k,k}$, $p_{k,k-1}$ and $p_{k,k+1}, k = 0, 1, 2 \dots N$, which arise from the following situations:

- $p_{k,k}$: success of a *steady* node in an *occupied* slot; collision of two or more *hopping* nodes in an *unoccupied* slot; this slot is empty.
- $p_{k,k-1}$: collision of a *steady* node in an *occupied* slot.
- $p_{k,k+1}$: success of a *hopping* node in an *unoccupied* slot.

where an *occupied* slot represents a slot for which only one node has a +1 Q value. An *unoccupied* slot is a slot for which all nodes have a -1 Q value. Based on the previous definition, a *hopping* node is a node which has a -1 Q value for all slots.

More specifically, to stay in the same state ($p_{k,k}$) either:

- The current slot is *occupied* and no *hopping* nodes select the current slot.
- The current slot is *unoccupied* and two or more *hopping* nodes transmit packets in it.
- The current slot is *unoccupied* and there are no transmissions in it.

To move down one state ($p_{k,k-1}$):

- The current slot is *occupied* and one or more *hopping* nodes transmit packets in it.

To move up one state ($p_{k,k+1}$) :

- The current slot is unoccupied and only one hopping node transmits a packet in it.

For a given value of k and knowing N, the state transition probabilities can be obtained as:

$$p_{k,k} = \frac{k}{N}\left(\frac{N-1}{N}\right)^{N-k} + \frac{N-k}{N}\left(1 - \frac{N-k}{N}\left(\frac{N-1}{N}\right)^{N-K-1}\right) \tag{5.2}$$

$$p_{k,k-1} = \frac{k}{N}\left(1 - \left(\frac{N-1}{N}\right)^{N-k}\right) \tag{5.3}$$

$$p_{k,k+1} = \left(\frac{N-k}{N}\right)^{2}\left(\frac{N-1}{N}\right)^{N-k-1} \tag{5.4}$$

5.3.2. Average Convergence Time

The convergence time for this scheme is the accumulated time the system spends in all states before reaching state N, because once the process reaches state N it will never move back to the previous states. Consider $\boldsymbol{P}$ as the state transition probability matrix which has the elements:$p_{i,j},\ i,j = 0, 1, 2 \ldots N$. In this model, $\boldsymbol{P}$ is a sparse matrix. Defining $\boldsymbol{P}^2 = \boldsymbol{PP}$, the matrix element:

$$p_{i,j}{}^{(2)} = \sum_{m=0}^{N}(p_{i,m}\, p_{m,j}) \tag{5.5}$$

represents the probability that the process visits state j via state m (where m represents any state) starting in state i.

The elements in $\boldsymbol{P}^3$**:**

$$p_{i,j}{}^{(3)} = \sum_{m=0}^{N}(p_{i,m}{}^{(2)}\, p_{m,j}) \tag{5.6}$$

denote the probabilities that the process visits state j via all possible states after two transitions from state $i \; \forall \; i, j$, via any two transition states.

So, from (5.5) and (5.6), $\boldsymbol{P}^n$ is the matrix of visit probabilities between any state and all others via n arbitrary transitions. The element ${p_{i,j}}^{(n)}$ is the probability that the process visits state j after n transitions, starting in state i. Alternatively, ${p_{i,j}}^{(n)}$ can be described as the expected number of visits to state j at the nth state transition starting from state i, or the expected number of slots the process stays in state j at the nth state transition starting at state i. As with $\boldsymbol{P}$, each row sum of $\boldsymbol{P}^n$ will always equal unity.

Convergence of the scheme (unity probability of reaching state N) can be proved if:

$$\lim_{n\to\infty} {p_{i,N}}^{(n)} = 1, i = 0, 1, 2 \dots, N \tag{5.7}$$

Proof:

We base our proof on the relationship between $\boldsymbol{P}$ and its Jordan normal form [77] $\boldsymbol{J}$, and invoking the associated similarity transform property:

$$\boldsymbol{B}^{-1}\boldsymbol{P}\boldsymbol{B} = \boldsymbol{J} \Leftrightarrow \boldsymbol{P}^n = \boldsymbol{B}\boldsymbol{J}^n\boldsymbol{B}^{-1} \tag{5.8}$$

To obtain the Jordan normal form $\boldsymbol{J}$, we need to calculate the eigenvalues of $\boldsymbol{P}$ by solving its characteristic equation:

$$det\,(\lambda\boldsymbol{I} - \boldsymbol{P}) = 0 \tag{5.9}$$

Where $\boldsymbol{I}$ is the identity matrix and $\boldsymbol{P}$ is an $(N + 1)$ square matrix which has only one non-zero element in the last row.

Using conventional matrix indexing:

$$\boldsymbol{P}(N+1, N+1) = p_{N,N} = 1 \tag{5.10}$$

We expand (5.9) by using the last row according to the Laplace Expansion [78]:

$$det\ (\lambda\boldsymbol{I} - \boldsymbol{P}) = (-1)^{N+1+N+1}(\lambda - 1)M_{N+1,N+1} \tag{5.11}$$

where $M_{N+1,N+1}$ is the determinant of the $N+1, N+1$ minor matrix of $\boldsymbol{P}$, which is the $N \times N$ matrix resulting from removing the last column and row of $\boldsymbol{P}$.

Clearly $\lambda = 1$ is one eigenvalue of $\boldsymbol{P}$. To determine the range of the remaining eigenvalues, we employ the Gershgorin Circle Theorem [79]. Begin by calculating absolute row sums excluding non-diagonal elements over $\boldsymbol{P}$:

$$R_i = \sum_{j \neq i} \left| p_{i,j} \right|, i = 0 \ldots N \tag{5.12}$$

Then a set of Gershgorin discs can be drawn, centered at p_{ii} with radius R_i in the complex domain as $D\left(p_{i,i}, R_i\right)$. The Gershgorin Circle Theorem states that the eigenvalues of $\boldsymbol{P}$ lie within the relevant Gershgorin disc. Except for $p_{N,N}$, the eigenvalues associated with all other diagonal elements have an absolute value less than unity, because each row sum of $\boldsymbol{P}$ equals unity.

Suppose $\boldsymbol{J}$ has m Jordan blocks and $\boldsymbol{J}$ can be represented by $J_i, i = 1,2 \ldots m$ where:

$$J_i = \begin{bmatrix} \lambda_i & 1 & 0 & \cdots & & 0 \\ & \lambda_i & 1 & 0 & \cdots & 0 \\ & & \lambda_i & \ddots & \ddots & \vdots \\ & & & \ddots & & 0 \\ & \boldsymbol{0} & & & \lambda_i & 1 \\ & & & & & \lambda_i \end{bmatrix}_{r_i \times r_i} \tag{5.13}$$

where r_i is the multiplicity of eigenvalue λ_i.

We know that $J_m = 1$. The matrix $\boldsymbol{J}^n$ can be calculated by diagonally aggregating the individual Jordan Blocks $J_i{}^n, i = 1,2 \dots m$, each of which can be obtained from:

$$J_i{}^n = \begin{bmatrix} \lambda_i{}^n & \binom{n}{1}\lambda_i{}^{n-1} & \cdots & \binom{n}{r_i-1}\lambda_i{}^{n-r_i+1} \\ & \lambda_i{}^n & \ddots & \vdots \\ & \mathbf{0} & & \binom{n}{1}\lambda_i{}^{n-1} \\ & & & \lambda_i{}^n \end{bmatrix}_{r_i \times r_i} \tag{5.14}$$

From the calculated bound on the eigenvalues above, with $n \to \infty$, we can see that $\boldsymbol{J}^n$ is an $(N+1)$ square matrix with the $(N+1, N+1)$ indexed element equal to unity and all others equal to zero.

We calculate the matrix $\boldsymbol{B}$ from $\boldsymbol{B}^{-1}\boldsymbol{P}\boldsymbol{B} = \boldsymbol{J} \to \boldsymbol{P}\boldsymbol{B} = \boldsymbol{Q} = \boldsymbol{B}\boldsymbol{J}$**.** Suppose the last column of $\boldsymbol{B}$ is $b_{i,N+1}, i = 1 \dots N+1$, then calculate the last column of $\boldsymbol{Q}$ as $q_{j,N+1}$, $j = 1 \dots N+1$. From $\boldsymbol{Q} = \boldsymbol{P}\boldsymbol{B}$ we get:

$$q_{j,N+1} = \sum_{i=1}^{N+1} p_{j,i-1} b_{i,N+1} \tag{5.15}$$

From $\boldsymbol{Q} = \boldsymbol{B}\boldsymbol{J}$ we can get:

$$q_{j,N+1} = b_{j,N+1} \tag{5.16}$$

Moreover, we have $\sum_{i=1}^{N+1} p_{j,i-1} = 1, j = 1 \dots N+1$. By substituting (5.16) into (5.15) the last column of **B** can be obtained as all 1s:

$$b_{i,N+1} = 1, i = 1 \dots N+1 \tag{5.17}$$

The last row of $\boldsymbol{B}$ is $b_{N+1,i}, i = 1 \dots N+1$. Calculating the last row of $\boldsymbol{Q}$ as $q_{N+1,j}$, $j = 1 \dots N+1$. From $\boldsymbol{Q} = \boldsymbol{P}\boldsymbol{B}$ we can get:

$$q_{N+1,j} = b_{N+1,j}, j = 1 \dots N+1 \tag{5.18}$$

From $\boldsymbol{Q} = \boldsymbol{B}\boldsymbol{J}$ we can get:

$$q_{N+1,j} = \sum_{i=1}^{N+1} b_{N+1,i} J_{i,j}\ , j = 1 \ldots N+1 \tag{5.19}$$

Substituting (5.18) into (5.19) we have $b_{N+1,1} = b_{N+1,1}\ J_{1,1}$, where $J_{1,1}$ is non-zero (see the proof in the appendix), so $b_{N+1,1} = 0$. Then we have $b_{N+1,2} = b_{N+1,2}\ J_{2,2}$, so $b_{n+1,2} = 0$… and we can calculate the rest: $b_{N+1,i} = 0, i = 1 \ldots N$. The matrix $\boldsymbol{B}$ has its last column all 1 and last row all 0 except for $b_{N+1,N+1} = 1$:

$$\boldsymbol{B} = \begin{bmatrix} \boldsymbol{B}_{N\times N} & & 1 \\ & & \vdots \\ 0 & \ldots & 1 \end{bmatrix}_{(N+1)\times(N+1)} \tag{5.20}$$

Then we calculate $\mathbf{B}^{-1}$ by using:

$$\boldsymbol{B}^{-1} = \frac{\boldsymbol{B}^*}{det\ (\boldsymbol{B})} \tag{5.21}$$

Where $\boldsymbol{B}^*$ is the adjoint matrix of $\boldsymbol{B}$. By expanding the last row of $\boldsymbol{B}$, we can get $det(\boldsymbol{B}) = det\ (\boldsymbol{B}_{N\times N})$. $\boldsymbol{B}^*$can be represented by the transpose matrix of a matrix which has its element equal to $(-1)^{i+j} M_{i,j}, i,j = 1,2, \ldots N+1$, the cofactors of $\boldsymbol{B}$. $M_{i,j}$ is the determinant of the i,j minor matrix of $\boldsymbol{B}$**.** The last row of $\boldsymbol{B}^*$ are: $(-1)^{i+N+1} M_{i,N+1}, i = 1 \ldots N+1$, and we also know that $M_{i,N+1} = 0, i = 1 \ldots N$, $M_{N+1,N+1} = det(\boldsymbol{B}_{N\times N}) = det(\boldsymbol{B})$. So the last row of $\boldsymbol{B}^{-1}$ has zero entries except for the final one which is unity. Finally we calculate $lim_{n\to\infty} \boldsymbol{P^n} = \boldsymbol{BJ^nB^{-1}}$ using the previous results as:

$$lim_{n\to\infty} \boldsymbol{P^n} = \begin{bmatrix} & 1 \\ \boldsymbol{0} & \vdots \\ & 1 \end{bmatrix}_{(N+1)\times(N+1)} \tag{5.22}$$

This proves the convergence of the learning scheme.

To calculate the time before convergence we need the expected time that this process stays in all states except state *N*, which is equal to the expected number of visits to these states across all n^{th} transitions where $n = 0, 1, 2 \ldots, \infty$:

$$E\{convergence\ time\} = \sum_{n=1}^{\infty} \sum_{j=0}^{N-1} p_{i,j}{}^{(n)} \tag{5.23}$$

Which is the expected convergence time starting with state i.

The expected convergence time from the initialisation of the process can be obtained by calculating.

$$E\{convergence\ time\} = \sum_{n=1}^{\infty} \sum_{j=0}^{N-1} p_{0,j}{}^{n} \tag{5.24}$$

5.4. Performance Evaluation

The convergence time obtained from the Markov model is compared to simulations from OPNET for the purpose of validation. The steady-state performance characteristics of ALOHA-Q are then evaluated in OPNET and compared with slotted ALOHA, S-MAC and Z-MAC to show the improvements achieved by applying learning and the overall capability of the scheme compared to a well established scheme for WSNs. For the performance evaluation, a single-hop network is considered with 200 nodes that all generate data packets and send them directly to the sink. All nodes have the same mean packet inter-arrival time and the inter-arrival time is exponentially distributed. A list of simulation parameters can be found in Table 5.1. All schemes are assumed to be synchronised.

Parameter	Value
Channel Data Rate	250 kbits/sec
Data Packet Size	1044 bits
ACK Packet Size	20 bits
Slot Length	0.0044 sec

Table 5.1 Simulation Parameters

5.4.1. Convergence Results

Experiments have been undertaken which match all the assumptions made in the analysis in the previous section. Figure 5.4 shows the average convergence time with different numbers of nodes in the network and each marker represents the average of 200 simulations. The convergence time has been determined for a maximum of 15 nodes in Figure 5.4 since this demonstrates a close match between the analytical model and simulation, and the theoretical computation becomes prohibitive for larger numbers of nodes. Theoretically, the expected convergence time requires a large number of computations, as (5.24) shows with $n \rightarrow \infty$. When the convergence time is calculated for 15 nodes, $n = 10^9$ so that $\boldsymbol{P}^n$ converges. This requires multiplication of two 16×16 matrices 10^9 times and it takes about 6 hours to obtain the results. Moreover, the time required to calculate the convergence time increases exponentially with each additional node, so results cannot be easily provided for larger number of nodes. The 95% confidence interval for the simulation results associated with 3 nodes is about $\pm$ 3%, and the confidence interval increases with the number of nodes. The confidence interval for 15 nodes is approximately $\pm$12%. The confidence intervals are not shown on the figure as they are almost invisible given the logarithmic scale of convergence time on the y axis.

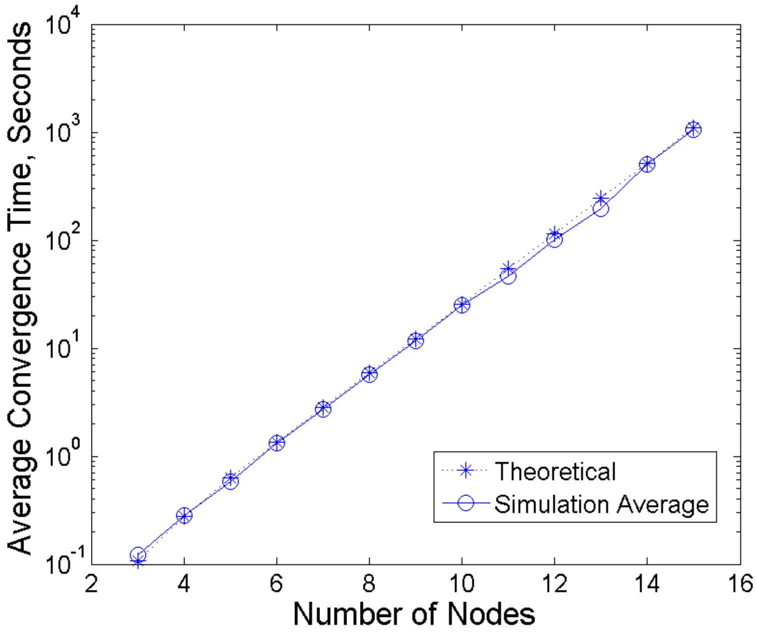


Fig. 5.4 Average convergence time

Figure 5.5 shows the Cumulative Distribution Function (CDF) [80] of the convergence time of the networks with different number of nodes in the network. Each curve shows the results of 1000 simulations. The dotted lines are produced by the Markov model and the circle markers are samples of the simulation results. The results show that the distribution of the convergence time matches the analysis. The discrepancy in the curve for 3 nodes is caused by the particularly small convergence times which are therefore sensitive to minor fluctuations. The simulation model determines whether convergence has been achieved at the beginning of each frame, so the reported convergence time is restricted to multiples of the frame length and usually larger than the estimated convergence time obtained from the Markov model. This discrepancy does not exist for larger number of nodes.

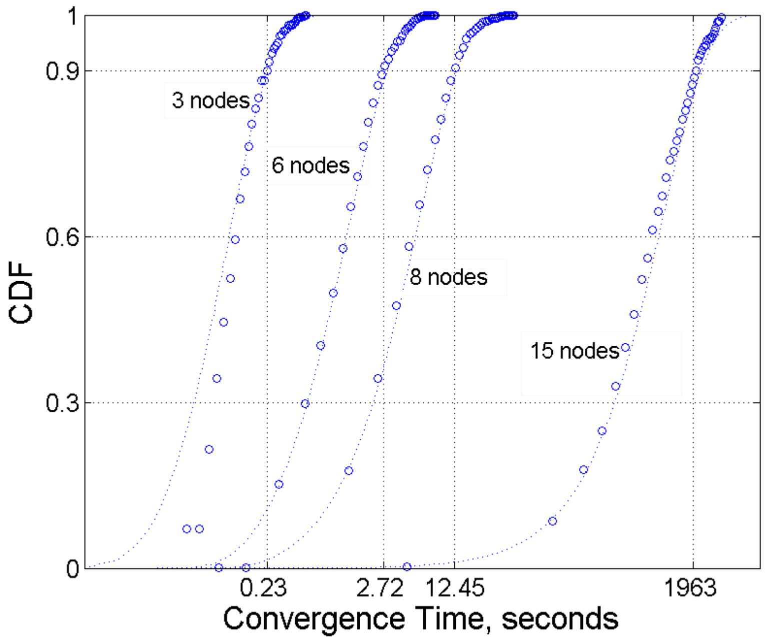


Fig. 5.5 CDF of convergence time measured through analysis and simulation

It can be seen that the convergence times increase rapidly with an increasing number of nodes in the network, but the observed times need placing in context. In the majority of sensor network applications, the intention is to deploy a set of nodes for long periods of times (potentially years). In this respect, an initial convergence time of minutes (or even hours) is not a significant problem. It is also important to note that the protocol operates in a perfectly adequate fashion and offers performance benefits prior to convergence. To demonstrate this, Figure 5.6 shows the real-time running throughput (the throughput from the start of the simulation to different time points) achieved from initialisation to each time step with different learning rates. Each curve represents the average of 100 simulations. The network comprises 200 nodes, the frame size is 200 slots and the generated traffic load is 0.7 Erlangs. The running throughput is calculated at the beginning of each frame. It can be seen that a learning rate of 0.5 performs worse than the smaller learning rates, with suboptimal throughput for about 400 seconds before rising and reaching 0.7 Erlangs after about 1500 seconds. Learning rates equal to or less than 0.1 provide much more rapid convergence and almost identical performance characteristics. Throughput increases with time and reaches 0.7 Erlangs about 100 seconds after initialization. The experienced throughput exceeds that obtainable with conventional slotted ALOHA after just 10 seconds. It is worth noting that Z-MAC requires a similar set up phase where nodes are unable to transmit data packets before it is completed [44]. According to the analysis in [45], DRAND requires approximately 600 seconds to determine the owner of each slot for a network with 200 nodes. ALOHA is able to provide substantial data throughput, approaching its maximum of 0.7 Erlangs after 200 seconds.

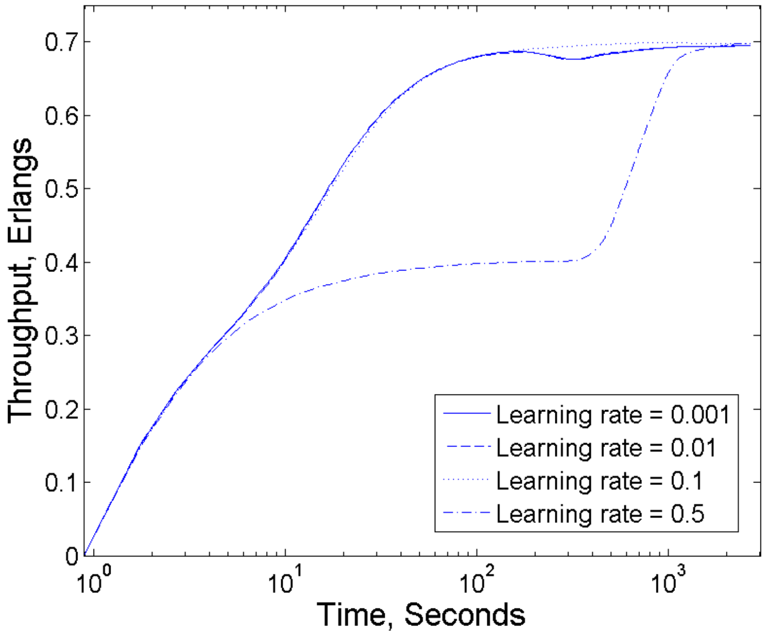


Fig. 5.6 Real-time throughput

Note that the first group of experiments (results in Figure 5.4 and 5.5) are based on the Markov model in section 5.3.1, and the second group of experiments (results in Figure 5.6) are acquired from the ALOHA-Q protocol described in section 5.2.1. To implement the analytical model, we made some assumptions and set certain parameters to the protocol. As the result, the convergence time of the analytical model is much larger than the capability of this scheme. For a network with 15 nodes, it requires on average 1000 seconds to converge, and the convergence time is doubled (approximately, by considering the trend) with each extra node in the network. On the other hand, the full ALOHA-Q protocol converges within 100 seconds with the network consisting 100 nodes, which is the capability of this scheme.

Compare the full ALOHA-Q protocol with the Markov model, the ALOHA-Q has much lower probability of returning to the previous state (the probability $p_{k,k-1}$), because once the Q values build up, it tolerates a few failures before hopping to another preferred slot, unless the nodes just hops to the current slot with low Q values. The probability of moving forward one state (the probability $p_{k,k+1}$) in ALOHA-Q is also higher because the Q values of the slots once have collisions are lower and the node can avoid using them. However it is very difficult to implement a similar Markov for ALOHA-Q because of the memory feature of the RL algorithm. The Q values of each node are determined by all prior actions of the network, it is difficult to quantise the network conditions to states because it could have infinite number of states, and the probabilities of the node and network actions vary through time. In each simulation, the RL algorithm has independent and different convergence process, resulting in the wide spreading convergence time mentioned in Chapter 6. For our knowledge, there are few analytical results on completed RL algorithms, because of the complexity of the analysis and requirements of manipulating wide range of mathematical skills. It can potentially be a future research topic to explore the understanding of RL and estimate the convergence capability not only by simulations.

5.4.2. Steady State Simulation Results

ALOHA-Q, Slotted ALOHA with exponential back-off [74], S-MAC and Z-MAC have been simulated in OPNET to evaluate and compare their performance in terms of throughput, end-to-end delay and energy cost per bit throughput. A single-hop network with 200 nodes is used in the simulations. A 10% duty cycle is used for S-MAC as commonly employed [36]. Each contention slot of Z-MAC has a length of 0.5 bits, and we use the same contention window size employed in [44] (8 contention slots for slot owners and extra 32 contention slots for non-owners) for the simulations of Z-MAC in this chapter. We turn off the LCL, HCL and Explicit Contention Notification (ECN) because they make no difference in a single-hop network, in accordance with [44], which assumes that all nodes in the network can transmit packets to and receive packets from each other, so that there are no hidden node problems and the channel sensing provides correct information. Note that this assumption (all nodes are within one-hop range) is not required for ALOHA-Q, and hidden nodes may exist even in a practical single-hop network, which will affect the channel performance. According to the experiments in [44], a 250 slots frame length is used for Z-MAC, which is typically larger than the number of nodes in the network.

Figure 5.7 shows the throughput performance. It can be seen that the throughput of ALOHA-Q with 200 slots/frame increases linearly with the generated traffic load and reaches a maximum close to 0.95 Erlangs, which is the theoretical maximum for the simulation parameters in Table 5.1 (1044 data bits/1100 bit slot length ≈ 0.95). S-MAC has a slightly lower maximum throughput capability because of the fixed overheads present in the listening period (10% in the simulations). The performance of ALOHA-Q with 250 slots/frame is shown to demonstrate the impact of overestimating the number of nodes in the network. The throughput exhibits a similar linear increase but to a lower maximum, because the frame is oversized and some slots remain unused. Slotted ALOHA achieves a maximum throughput of 0.27 Erlangs, one third of that achievable with

ALOHA-Q given the marginal increase in complexity, demonstrating the effectiveness of intelligent slot selection through learning. Z-MAC achieves a throughput of about 80% of the channel capacity, which is similar to the results observed in [44]. Its performance is limited by the contention windows (8 slots for owners and additional 32 slots for non-owners) used for channel sensing. When the network comprises a large number of nodes, the probability of two or more nodes applying channel sensing in the same slot is high under high traffic conditions, which causes more collisions and limits the throughput performance. Therefore, Z-MAC usually has better performance than S-MAC for multi-hop networks, but under this scenario it is not optimum.

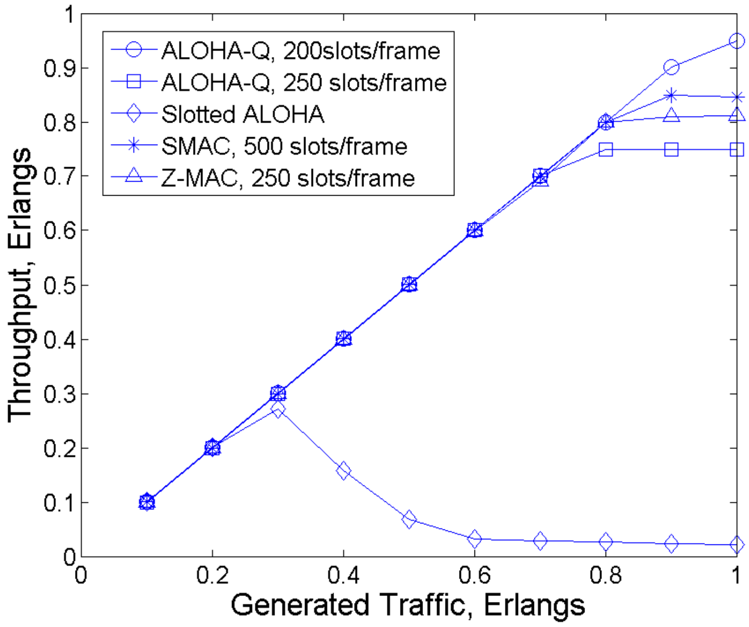


Fig. 5.7 Throughput

The average end-to-end delay experienced by packets is shown in Figure 5.8 as a function of generated traffic load. When the traffic load is low, Slotted ALOHA offers the lowest delay because nodes are able to access the channel almost immediately after packets are generated and relatively few experience a collision. Z-MAC has good delay performance under low traffic conditions, because it is similar to Slotted ALOHA in allowing a node to transmit a packet straight after it has been generated when the channel is clear. All ALOHA-Q schemes offer less than 3 seconds delay before reaching the maximum traffic level. The slightly higher delay of S-MAC is caused by the long frame structure and sleep period.

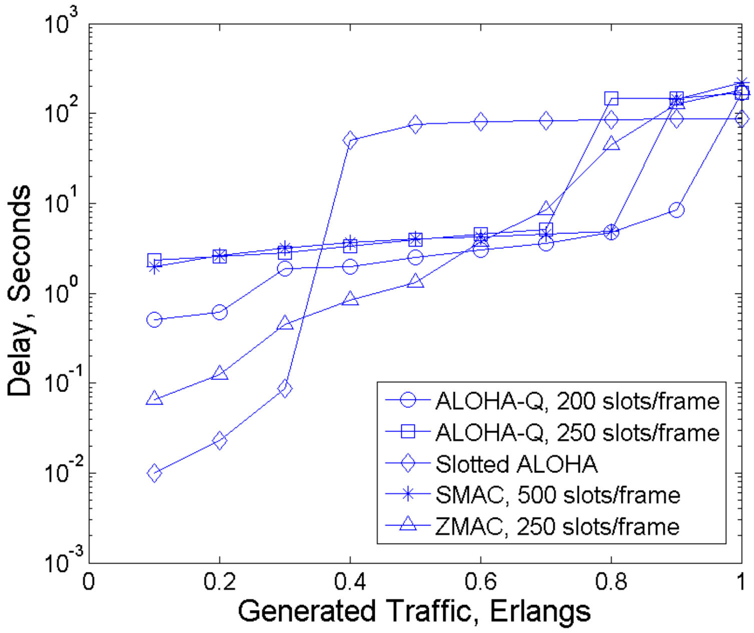


Fig. 5.8 End-to-end delay

Figure 5.9 presents the energy efficiency results. The total energy cost is the sum of energy consumed by transmitting/receiving packets, idle listening and overhearing. The energy cost per bit throughput is calculated by dividing the total energy cost (mJ) by the amount of data received at the sink (in bits). Slotted ALOHA has the best energy efficiency at low traffic loads because of the low probability of collision, but energy costs rises rapidly with increasing traffic load due to the overheads of retransmission. The two ALOHA-Q schemes have similar performance characteristics as a function of traffic load and better performance at higher traffic levels. S-MAC exhibits the same performance trend but generally higher energy costs resulting from the additional overheads in the frame and idle listening. Z-MAC has slightly higher energy cost than ALOHA-Q because of channel sensing (a node applies channel sensing in every Z-MAC slot when it has packets to send).

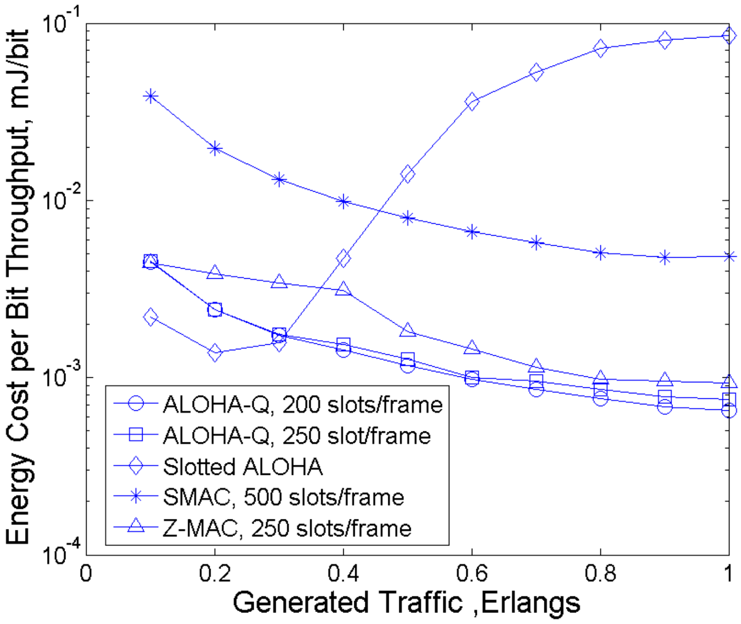


Fig. 5.9 Energy cost per bit throughput

5.5. Conclusions

In this chapter, ALOHA-Q is proposed as a low complexity MAC protocol capable of providing high energy-efficiency performance combined with high throughput and adequate delay. The main contribution of this protocol is its simplicity and low overheads compared with the majority of current schemes. Q-Learning is employed as an intelligent slot selection strategy in a frame based Slotted ALOHA scheme which results in migration from random access to perfect scheduling in steady state conditions. A detailed study of the convergence properties of the approach has been evaluated through a Markov model of the learning process. Based on the Markov model, we prove the convergence to the optimal steady state of ALOHA-Q and obtain the expected convergence time.

Results for a single hop network demonstrate that ALOHA-Q is able to provide rapid convergence to steady-state conditions and improved energy-efficiency, comparable throughput and delay to S-MAC and Z-MAC despite being much simpler.

6. Reinforcement Learning for Multi-hop WSNs

Contents

Chapter 4 introduced the principles of applying RL to WSNs based on a single-hop network topology and initially explored the potential performance of RL. A typical WSN usually has a multi-hop topology in which nodes cannot reach all other nodes in the network, especially in environmental monitoring tasks which have large sensing area (and limited communication range of sensor nodes). The scheme in Chapter 4 is based on the sink node constantly listening; however, in multi-hop networks the relay nodes must switch to sleep mode to save energy. In this chapter, techniques are introduced to solve these problems, extending the protocol for application to multi-hop WSNs.

6.1. Introduction

In single-hop networks, nodes directly communicate with the sink node which constantly listens to the channel, so packet loss can only be caused by collisions (at least in the simulations). However, in a typical multi-hop WSN, nodes have limited power resources and must switch between active and sleep modes to extend the network lifetime. This brings additional packet loss through channel closure due to unavailability of the intended receiver (since the intended receiver has switched into a sleep mode). To maintain low overheads and complexity, we aim to solve these problems without using traditional channel sensing and handshake mechanisms (eg. RTS/CTS).

This chapter introduces two methods to reduce packet loss caused by channel closure, while offering nodes as many opportunities to sleep as possible. Informed receiving (IR) uses information embedded in data packets to inform the receiver about future slot selection, so that the receiver can wake up when the transmitter sends packets. When the traffic load is low, nodes send data packets relatively infrequently, which makes the IR information out of date. Therefore the receiver does not know whether the absence of the traffic is due to collisions or silence of the transmitter. At low traffic levels, nodes send ping packets to the next hop receiver to keep them informed about their slot selection when they have no data packets to send, and nodes switch to sleep mode after receiving ping packets. A ping packet is tiny and it uses “owned” slots so that it only causes minimal overheads and no collision (in steady state).

This chapter extends the scheme introduced in Chapter 4 by introducing IR and ping packets and evaluates them in a linear chain multi-hop network and a random multi-hop network. Different RL algorithms are also applied to achieve better convergence performance. Results show that our schemes significantly improve throughput, delay and energy efficiency performance compared with the original ALOHA scheme, and considerable performance gain compared with other MAC protocols for WSNs. Under very light traffic conditions, our scheme achieves over 60% energy cost by transmitting/receiving data packets and at high traffic conditions it achieves nearly 100% energy efficiency (almost all the energy cost results from transmitting/receiving data packets). The initialisation time (convergence time) of the network remains at an acceptable level, from seconds to a few minutes, depending on the size of the network.

6.2. Network Topologies

In this chapter, we apply our scheme to two multi-hop topologies: a linear chain multi-hop network and a random multi-hop network. This section describes these topologies and associated assumptions.

6.2.1. Linear Chain Network

To adapt and apply our scheme to multi-hop networks, consider as the first topology case study a simple linear chain network. Figure 6.1 shows the network topology. The chain consists of 9 nodes, where node 1 generates data packets and node 9 is the sink. All other nodes are relay nodes. Data packets are transmitted via 8 hops from node 1 to node 9. A node can only receive packets from its one-hop neighbours, but packet reception can be interfered with based on the transmissions from two-hop neighbours. For example, while node 3 is sending a packet to node 4, simultaneous transmissions initiated by nodes 2, 5, or 6 can collide with the packet from node 3 and cause failures of its transmission. This interference situation indicates that one in four neighbour nodes can transmit in the same slot without any collisions, therefore the theoretical maximum throughput at the sink node will be 0.25 Erlangs (4 slots per frame) without counting any overheads. This theoretical maximum throughput can also apply to other similar chain networks (source and sink nodes at both ends of the chain) unless the number of nodes is less than 5. With 4 or less nodes in the network, the number of contenders in the neighbourhood of each node becomes less, therefore the maximum throughput of node numbers from 4 to 2 become 0.33 Erlangs, 0.5 Erlangs and 1 Erlang.

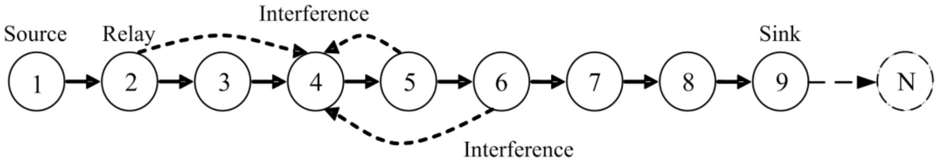


Fig. 6.1 Linear chain network

6.2.2. Random Network

In most current WSNs applications, nodes are randomly deployed (usually by vehicles or aircraft throughout the deployment region) and have an unknown number of neighbours and random locations. Figure 6.2 shows an example of a part of a random WSN. We assume that packets from nodes within the radius R_r can be correctly received and decoded and that simultaneous transmissions from nodes within the radius R_i can interfere the current reception and cause failures. We consider a simple Most Forward Routing (MFR, or shortest path) [75] for nodes to relay packets to the sink. Nodes select the neighbour within R_r with the shortest distance to the sink node as the next hop to transmit packets. All nodes generate packets (a node is both a source and a relay), which causes higher contention levels for nodes close to the sink. In the simulations in this chapter, we use randomly deployed nodes over a square sensing area. The sink node is located at the centre of the sensing area.

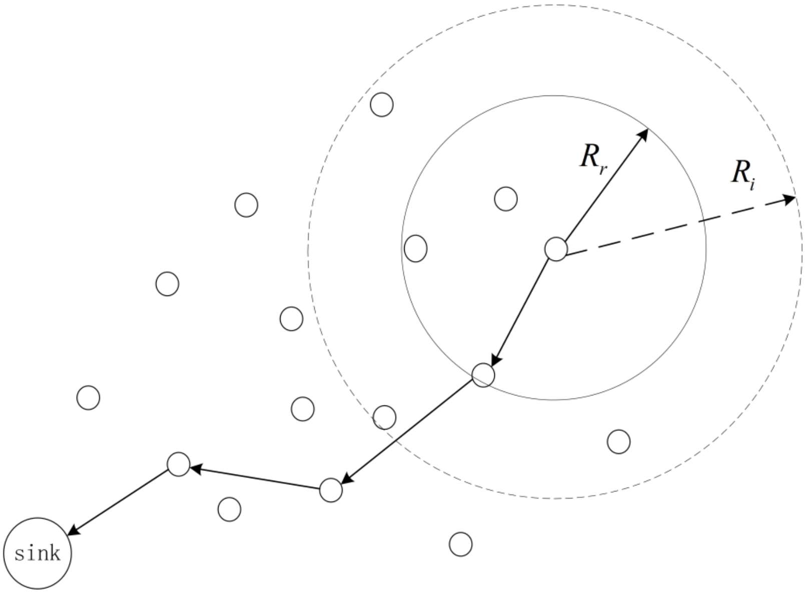


Fig. 6.2 Random network

6.2.3. Constraints of Multi-hop Networks

Multi-hop networks have much more complicated network topologies than the single-hop network considered in Chapter 4, which leads to difficulties in the design of MAC protocols. It is possible to deploy a more powerful node as the sink which is capable of constantly listening to the channel. However, most nodes in the network are still energy constrained and have to switch between active and sleep modes to save energy. Packet loss can be caused by collisions or radio closure at the receiver node. The previously considered RL algorithm can avoid the majority collisions (potentially all collisions in steady state), however it consumes prohibitive amount of energy because it requires the receiver node to listen to the channel constantly (because the transmission slot can change during operation) and the impact in single-hop networks is not obvious because the only receiver node is the sink node. Moreover, any form of global control has very high energy costs in multi-hop networks because of the limited communication range of each node, therefore frequent global information exchange is not practical. This chapter presents techniques to solve these problems while maintaining the favourable properties of ALOHA (distributed scheduling, absence of sensing and negotiation).

6.3. Techniques to Improve the Energy Efficiency

6.3.1. Informed Receiving (IR)

The RL algorithm always prefers to select slots with higher weights to send packets. When a preferred slot suffers a collision, the weight is reduced and the node may hop to other preferred slots. If the associated receiver node is not constantly listening to the channel, it fails to receive packets when the transmitter hops to another preferred slot. The information obtained by the learning process is unknown to the receiver node, but some of the learning information can be provided to the receiver node while transmitting data packets, allowing the receiver node to listen more efficiently. This section defines this process, which is known as Informed Receiving.

Assume that a node sends at most one packet in each frame. Considering the worst case that all transmissions fail, we can calculate the number of future frames that the node will keep using the current preferred slot from the highest weight and the second highest weight (and the associated learning algorithm). As part of the IR process, each data packet piggybacks this information and the receiver node uses this number to set up and update a timer. This timer indicates the number of future frames that the receiver should keep listening to the same preferred slot (and sleep in other slots). This timer is updated after receiving each data packet, or decremented by 1 if not updated in a frame. When the timer reaches zero, the receiver node switches to full listening mode, in which it listens constantly in every slot to locate the new preferred slot of the associated transmitter node and start a new listen/sleep schedule. The implementation details IR are provided in Sections 6.4.2 and 6.6.2.

6.3.2. Ping Packets

IR provides a basis for efficiently switching nodes between listening and sleeping, but it has one weakness. Under light traffic conditions, a transmitter node may remain silent for a few frames between two packet transmissions. During these frames, the IR timers (details are explained in Section 6.4.2) are not updated; the timers may reach zero and receiver nodes switch to full-listening mode, while the associated transmitter node still intends to use the same preferred slot. For the receiver nodes, the idle listening in the preferred slots of the associated transmitter nodes when they have no packet to send is energy consuming as well.

To avoid this frequent timeout (a timer reaches zero) problem under low traffic levels, ping packets are introduced, which carry the same IR information. When a node reaches its preferred slot in a frame and has no data packets to send, it transmits a ping packet to the next hop receiver node instead, indicating its current IR information. When a node receives a ping packet, it updates the timer, sends back an ACK and switches to sleep mode immediately. A ping packet is tiny (only 28 bits) and therefore only introduces minimal overheads. It uses the available channel capacity (preferred slots in which there is no data to send) to significantly reduce the energy consumption through idle listening. Non-acknowledged ping packets can be considered as failures and used to update the RL weights. The successful ping packets can also contribute to the RL algorithm (depending on the length of ping packets). More details of ping packets will be introduced in Sections 6.4.2 and 6.6.2.

6.4. RL-ALOHA for a Linear Chain Network

6.4.1. Protocol Design

A repeating frame structure is applied to Slotted ALOHA to implement RL. A frame consists of *N* slots, and each node is allowed to transmit at most one data packet per frame. The slots in the repeating frame are given unique weights which are initialised to zero and updated by successful transmissions (+1) and failures (-1). At the beginning of each frame, the node selects the slot with the highest weight as the current preferred slot and sends a packet in this slot if it has any queued. If multiple slots have the same highest weight the node will select one randomly. The number of slots per frame *N* is an important parameter for this scheme, and is usually set to the number of contenders. Therefore each node can own a unique slot in steady state conditions and collision can be avoided as long as the network has sufficient capacity. In Chapter 4 we considered Exponential Backoff (EB) for retransmissions, but here EB is removed because this work considers the optimal frame size (or larger) and the backoff is no longer needed to decrease the offered traffic of each node. EB is useful when the frame size is smaller than the optimum, however, it introduces extra delay and reduces convergence speed (less learning opportunities in the same amount of time) when frame size is equal to or larger than the optimum. Figure 6.3 shows the structure of the frames and the RL algorithm, and provides an example of the slot selection strategy of one node. In this example *N* equals 3.

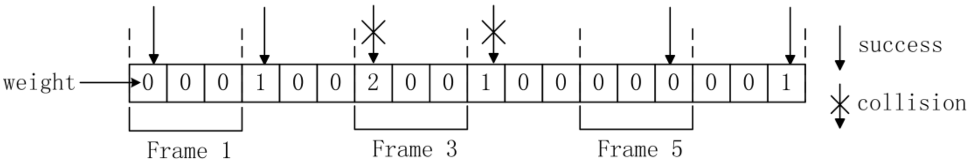


Fig. 6.3 Example of the Reinforcement Learning algorithm

Arriving packets are put in a first-in-first-out queue and only the packet at the queue head can be transmitted to the next hop. Once a packet arrives, it needs to wait at least until the next preferred slot to be sent (depending on the number of packets in the queue). If a collision occurs, the packet will be retransmitted based on the same slot selection strategy in future frames until it succeeds or reaches the maximum retry limit (*M=6* according to the IEEE 802.11 specification [74]). Every slot accommodates a data packet and the associated acknowledgement packet, therefore one data packet can only collide with another entire data packet as long as the system is synchronised.

In the steady state, one node intends to transmit packets in a fixed preferred slot to avoid collisions with the other nodes. However, this preferred slot assignment can possibly be impaired by sudden interference caused by other nodes being introduced to the network or external devices causing interference from outside the system. The weights are limited to the range of -1 to 10, to allow the scheme to adapt to environmental changes, because infinite weight values may cause a constant collisions Without these limits the system would underperform, for the following reason: once a node has had many successes in a preferred slot, if the current preferred slot became unavailable, it would take a huge amount of trials to reduce the weight sufficiently before the RL algorithm would hop to another alternative slot.

The RL algorithm can effectively reduce the probability of collision in the network. However this result is achieved based on the assumption that all nodes keep listening to the channel, leading to high energy consumption which is not acceptable in energy constrained WSNs. IR and ping packets are applied to avoid idle listening and overhearing, and allow nodes to sleep for as long as possible when they are not transmitting or receiving.

6.4.2. Informed Receiving (IR) and Ping Packets

A. IR

Given the frame-based structure of this protocol, a node can send at most one packet per frame. Therefore, for each transmitter and receiver pair the most energy efficient way for the receiver to operate is by only listening to the transmitter’s current preferred slot. However the nodes may switch to different slots during a task based on changes in the dynamic environment, which means that the nodes can have various preferred slots in different time periods. IR is applied to provide the receiver recent information about the associated transmitter so that the receiver can follow the transmission pattern and listen efficiently.

When a node switches on, it listens to all the slots and waits for arriving packets. The transmitter piggybacks the weight advantage (the weight of the current preferred slot minus the second highest weight) onto each data packet to give the receiver knowledge of its recent transmission pattern. The receiver has a timer which is decremented by 1 at the beginning of each frame, and its value is updated based on the weight advantage from the transmitter. The timer will be updated to the weight advantage plus 2 as its current value and this value indicates the number of future frames in which the transmitter will keep using the same slot. To explain the timer update: the weight advantage of the transmitter is the advantage before the transmission, and as long as the data packet is successfully received the weight of the current preferred slot will increase by 1. Since the timer decreases by one every frame, the +2 timer update can recover the offset caused by the two issues above and the value of the timer equals to the weight advantage in the next frame.

After switching on, the receiver listens to all the slots until its timer has a value equal or greater than 2 (the weight advantage of the transmitter is equal or larger than 1), indicating that the transmitter node will use the same slot for the next packet transmission. Then it only listens to the preferred slot and switches off the radio in other slots unless it has packets to transmit. The receiver will listen to all the slots again if the timer reaches zero, which indicates that the transmitter may start to use another preferred slot. This IR process offers the receivers a lot of sleep time with only very small additional overheads: an 8-bit field in each data packet which contains the weight advantage.

B. Ping Packets

The optimal potential performance of IR is that a node listens to one slot per frame and sleeps in all others. If the transmitter always has a packet to transmit in every frame, IR will work efficiency. However if the traffic level is low, the transmitter may keep quiet for several frames between two transmissions, which means that the receiver will waste energy listening to these empty slots in blank frames. Moreover, the timer will not stop in a blank frame, regardless of whether the blank frame is caused by a collision or by the idleness of the transmitter. Under low traffic levels the timer times out frequently because the transmitter rarely transmits. However, the receiver must then listen to all slots in a frame, which wastes energy.

Ping packets are applied to improve energy efficiency under low traffic levels. Once a transmitter reaches its preferred slot, if it has no data packet in its queue it sends a tiny ping packet which also carries the weight advantage to the associated receiver. The receiver then updates its timer, sends back an ACK packet and switches to sleep mode immediately. If the ping packet is not acknowledged, the transmitter will consider it as a failure and update its weights (-1). But the weights will not be updated if the ping packet succeeds, because the ping packet is tiny and practically its success does not indicate that the data packet can succeed in the same slot (ping packets from two neighbours can both success in the same slot if there is a certain synchronisation shift).

6.5. Performance of RL-ALOHA

6.5.1. Scenarios and Parameters

This scheme is implemented using the OPNET simulator. Two other different schemes are simulated for comparison: RL-ALOHA with IR (but not ping packets) and basic Slotted ALOHA with Binary Exponential Backoff (BEB). Table 6.1 shows the simulation settings of all schemes. The parameters of the radio transceiver are based on the IRIS nodes [31]. The power levels are calculated by multiplying the current levels by the average voltage 3V. The sleep power level is set to zero because the energy cost by sleeping is considered negligible. The idle listening power is considered to be the same as the receive power [36]. The source node 1 is the only node which generates data packets. The packet inter-arrival time is exponentially distributed which models typical traffic statistics in realistic scenario (eg. in long-term monitoring tasks all nodes constantly generate packets with rate [1]) and the generated traffic is calculated based on the average packet inter-arrival time. All nodes in the network are assumed to be synchronised. SMAC and Z-MAC are also simulated for comparison. SMAC is set to 50 slots per frame and a 10% duty cycle as commonly employed [8]. Z-MAC has 6 slots per frame, which is typically larger than the number of contenders in one neighbourhood [13]. Each contention slot of Z-MAC has a length of 0.5 bits, and 8 contention slots for slot owners and extra 32 contention slots for non-owners (same as [13]).

Parameters	*Value*
Transmitter Data Rate	250 kbits/sec
Receiver Data Rate	250 kbits/sec
Transmit Power	51mW
Receive Power	48mW
Data Packet Size	1044 bits
ACK Packet Size	20 bits
Ping Packet Size	28 bits
Slot Length	1100 bits
Number of Slots per Frame	4
Simulation Period	500,000 slots

Table 6.1 Simulation parameters

6.5.2. Delay Analysis

According to the scheme described in Chapter 4, nodes schedule a packet at the beginning of each frame, if there are packets ready to be sent. If no packets exist, a node sleeps in the current frame and all packets generated in this frame have to wait at least until the next frame to be transmitted. However, this introduces additional delay. Here we consider that a node determines the preferred slot at the beginning of each frame whether there are any packets ready or not. If a packet is generated before the preferred slot the node sends it, otherwise if no packets are generated it sleeps.

Considering the network shown in Figure 6.1 with 4 slots per frame, the nodes must have a slot selection pattern of reusing slots of nodes four hops away to reach the optimal steady state. As Figure 6.4 shows, there exists an optimal slot selection in which the packet can be relayed immediately in the next slot after it is received and the queuing time in each hop is minimised. There are 24 permutations (4!) of slot selections, and four of them are optimum. With the optimal slot selections, considering only one packet in the queue, each packet has an average of 2 slots queuing time at the source node and 1 slot relay time through each hop, so the total delay of the optimum slot selection is 10 slots, which is 0.044 seconds ($1100/250k \times 10$). However it is difficult to achieve this optimum slot selection without additional negotiation amongst several hops, so in the current scheme the slot selection is random (for each simulation) after convergence and the network has the probability of 16.7% (4/24) of convergence to the optimum.

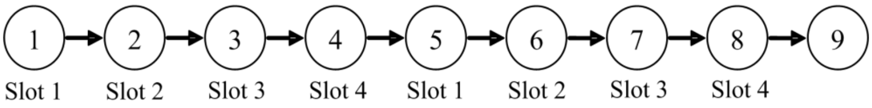


Fig. 6.4 Optimum slot selection

Considering another slot selection pattern of slots 1, 2, 4, 3, the estimated delay through 8 hops is:

$$\frac{1100}{250k} \times [2 + 2(1 + 3 + 3 + 2)] = 0.088 \text{ seconds} \tag{6.1}$$

For the slot selection pattern 1, 3, 2, 4, the delay is:

$$\frac{1100}{250k} \times [2 + 2(2 + 3 + 2 + 1)] = 0.079 \text{ seconds} \tag{6.2}$$

For the slot selection pattern 1, 3, 4, 2, the delay is:

$$\frac{1100}{250k} \times [2 + 2(2 + 1 + 2 + 3)] = 0.079 \text{ seconds} \tag{6.3}$$

For the slot selection pattern 1, 4, 2, 3, the delay is:

$$\frac{1100}{250k} \times [2 + 2(3 + 2 + 1 + 2)] = 0.079 \text{ seconds} \tag{6.4}$$

For the slot selection pattern 1, 4, 3, 2, the delay is:

$$\frac{1100}{250k} \times [2 + 2(3 + 3 + 3 + 3)] = 0.114 \text{ seconds} \tag{6.5}$$

So the average estimated delay of this topology is 0.085 seconds.

6.5.3. Channel Performance

The simulation results of throughput and end to end delay are shown in Figures

6.5 and 6.6 respectively. The results are collected after the simulations are initiated for 4000 slots, in order to precisely show the performance in steady state. The theoretical maximum throughput of the topology described at the sink node is 0.25 Erlangs. Based on the parameters given in Table 6.1, including the overheads, the maximum throughput in the simulation is approximately 0.2373 Erlangs. Figure 6.5 shows that the RL-ALOHA with IR provides good throughput, with all the input traffic successfully flowing to the sink until it reaches the limit. The basic Slotted ALOHA can only achieve 0.1 Erlangs throughput due to the blind transmission strategy, and the use of ping packets does not affect the throughput performance at all (the overheads caused by ping packets only exist when there is available channel capacity). The throughput stabilises at 0.237 Erlangs when node 1 generates more traffic, because the traffic on the chain is limited by the frame structure. SMAC has the maximum throughput of about 0.157 Erlangs, because its collisions avoidance through RTS/CTS and channel sensing (overhearing) is limited by the two-hop interference conditions. Z-MAC achieves less maximum throughput (0.204 Erlangs) than RL-ALOHA due to more overhead and more potential contention (nodes can potentially contend for their non-owned slots). Figure 6.6 shows that both RL-ALOHA schemes maintain an end-to-end delay less than or around 0.1 seconds with most traffic levels. The delay results for each marker are obtained from the average values of 50 simulations, and they are close to the estimated average delay when the traffic is lower than the capacity. The use of ping packets does not affect the delay performance. The delay of the original Slotted ALOHA increases significantly with the traffic level, due to the collisions caused by its blind slot selection strategy. SMAC has a higher overall delay than RL-ALOHA and Z-MAC according to the larger frame structure and packet losses/retransmissions caused by two-hop interference. Z-MAC achieves better delay performance at low traffic levels due to its random access feature at low contention level (LCL), however when the traffic load increases, Z-MAC has similar delay performance as RL-ALOHA.

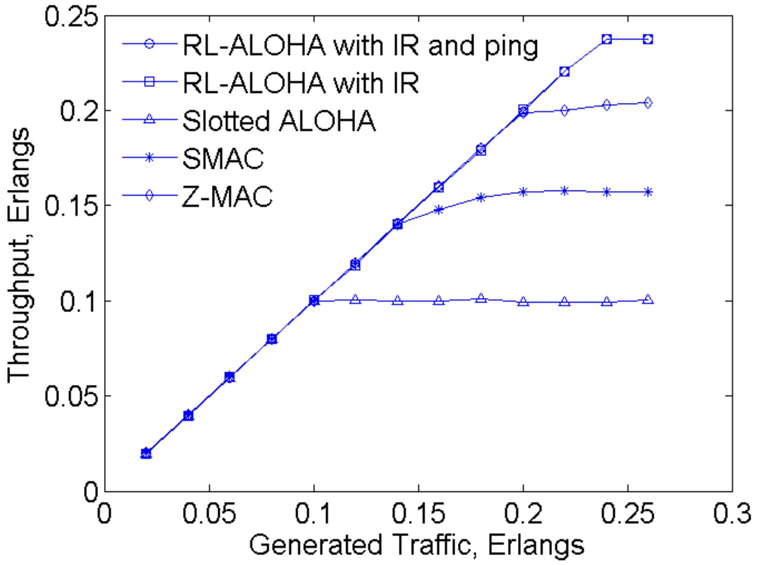


Fig. 6.5 Throughput Comparison

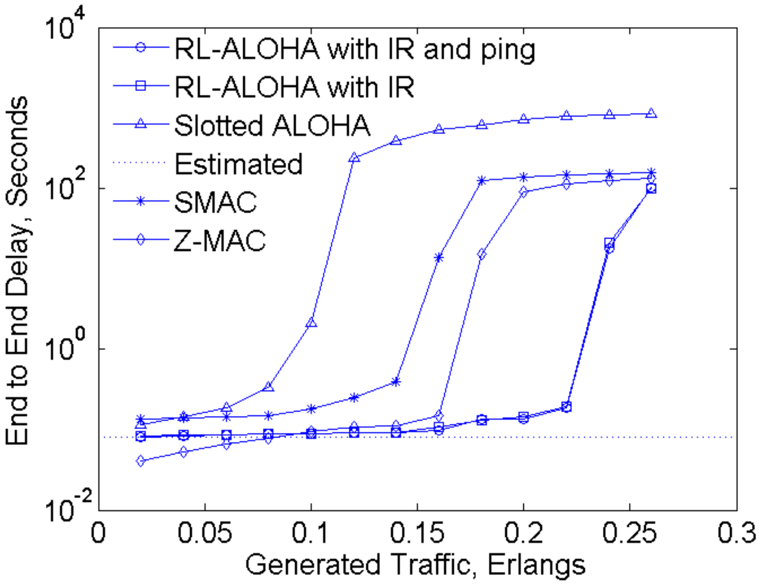


Fig. 6.6 End to end delay

6.5.4. Energy Efficiency

The average energy consumption per bit useful throughput, the average energy consumption per second and the detailed energy consumption of the radio transceivers are shown in Figure 6.7, Figure 6.8 and Figure 6.9 respectively. All the results are network-wide (i.e. not for a single node). Figure 6.7 and Figure 6.8 show that both RL-ALOHA schemes have much better performance in terms of energy efficiency than Slotted ALOHA, and they have similar energy efficiency when the generated traffic is close to the throughput limit (0.237 Erlangs). However huge differences are appeared between these two schemes when the traffic is low. The use of ping packets can save approximately 90% of the energy compared with RL-ALOHA with IR when the traffic level is very low (0.02 Erlangs). Figure 6.7 shows that RL-ALOHA with IR and ping packets maintains similar and low energy consumption per bit throughput with different traffic levels. Both SMAC and Z-MAC achieve stable energy efficiency across different traffic levels, but less efficient than RL-ALOHA due to more energy consumed by overheads, retransmissions and idle listening (SMAC has constant listening period and Z-MAC uses low power listening to ensure not missing any packets). Figure 6.8 shows that its energy consumption per second increases linearly with the traffic level and remains at a certain level when the input traffic is larger than the limit. The energy cost of Slotted ALOHA increases with the traffic level (transmitting is more energy consuming than receiving/listening) and reaches a limit when the generated traffic its maximum throughput 0.1 Erlangs. RL-ALOHA with IR has high energy consumption at low traffic levels because of the idle listening caused by the non-frequently updated IR information. The energy consumption caused by idle listening decreases with the increasing traffic level, and the total energy consumption becomes dominated by transmitting and receiving rather than listening, as Figure 6.8 shows. The performance of SMAC and Z-MAC has the similar trend as RL-ALOHA, but higher values.

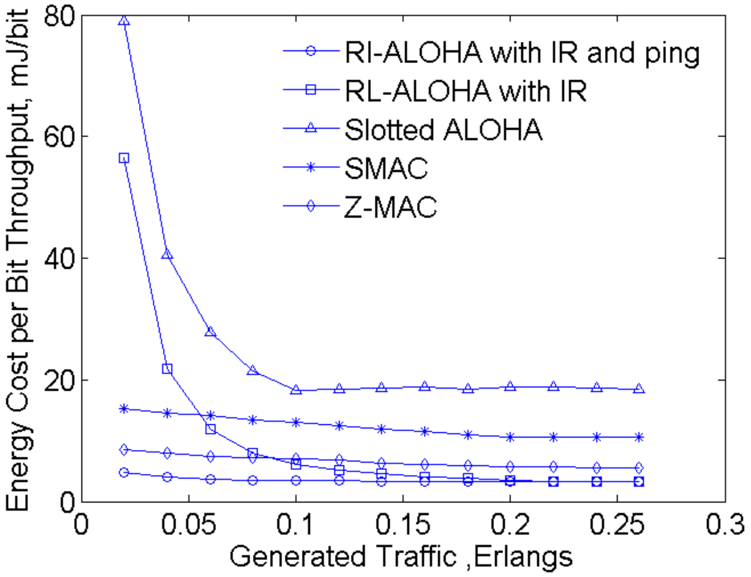


Fig. 6.7 Energy consumption per bit throughput

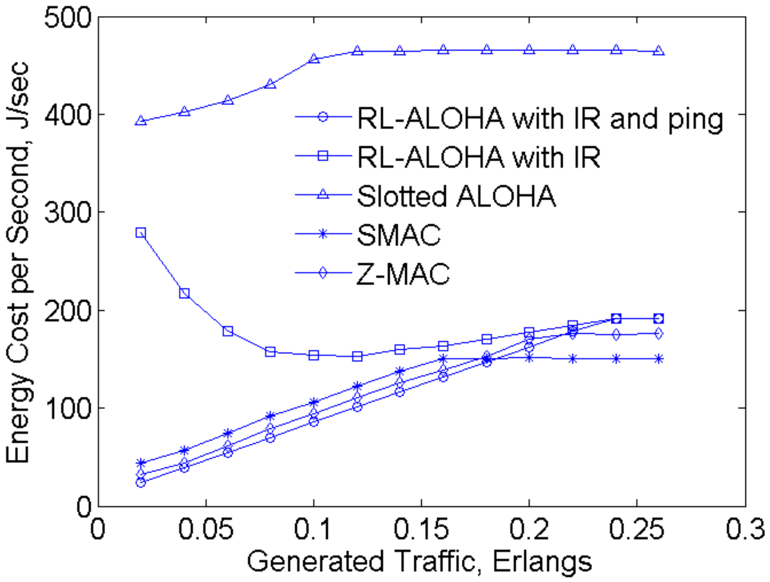


Fig. 6.8 Energy consumption per second

Figure 6.9 provides detailed energy consumption results and demonstrates the energy efficiency benefits of RL-ALOHA more clearly. The energy consumed by transmitting and receiving for both schemes is similar with different traffic levels, and the energy consumed in idle listening in both is very low when the traffic is high. This explains why the two schemes have similar energy efficiency with high traffic. The difference in energy efficiency at low traffic levels is caused by idle listening. In Figure 6.9, RL-ALOHA with IR and ping packets maintains low energy consumption due to idle listening with different traffic levels. However if ping packets are not used by this scheme, the network consumes a huge amount of energy on idle listening at low traffic levels. When the nodes rarely transmit packets, the receivers are seldom updated so their timers time out frequently and they have to listen during all the slots to make sure they will not miss any information. The ping packets solve this problem by frequently updating the weight advantages. The use of ping packets therefore creates significant improvements in energy efficiency, and only introduces 48-bits overhead (an ACK packet plus a ping packet) in each blank frame. Similar to RL-ALOHA, SMAC and Z-MAC have much lower proportion of energy cost on data at low traffic levels, due to the larger ratio of energy consumed by overheads and idle listening. When the traffic load increases to their maximum throughputs, SMAC and Z-MAC achieve the proportion of 82% and 91% respectively, but still lower than the performance of RL-ALOHA (which is over 97%).

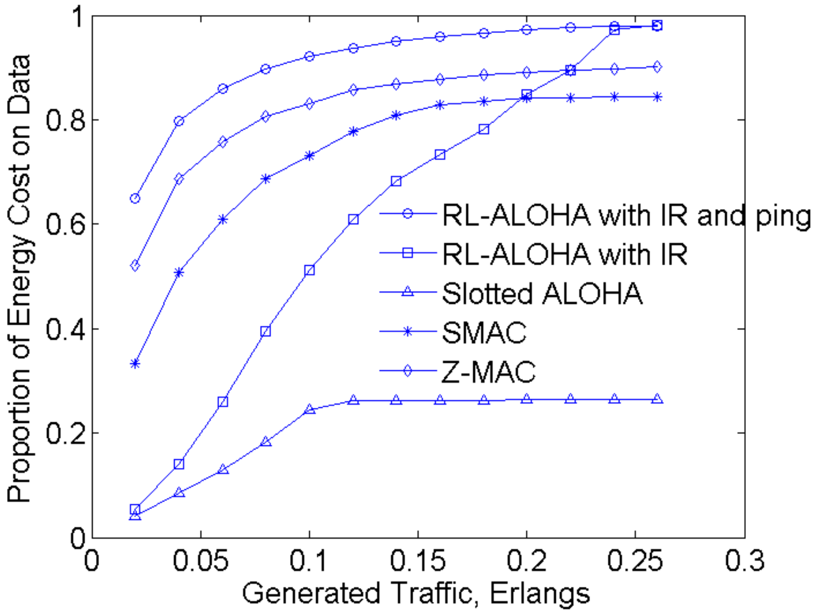


Fig. 6.9 Proportion of energy cost by transmitting and receiving data packets

6.5.5. Convergence Time

Recording the statistics under a steady state conditions is necessary to evaluate the long-term throughput, delay and energy performance. However, in realistic deployments the time that the system needs to converge to the steady state is also important. Figure 6.10 shows the Cumulative Distribution Function (CDF) of the number of frames that the system takes before converging to steady state (all nodes find their unique slots and never hop again) at three different traffic levels. The results of 100 individual simulations are included. The results are spread within a quite wide range due to the random nature of the learning process, and the system may take less than 20 frames or even more than 1000 frames to converge.

The results show that under medium and high traffic conditions approximately 90 percent of the simulations have converged within 1000 frames. This corresponds to 17.6 seconds, which is negligible time cost in most WSNs monitoring tasks. The results also indicate that the system converges faster with higher traffic loads, because more frequent packet transmissions give the nodes more information about the channel and thus accelerate the speed of learning.

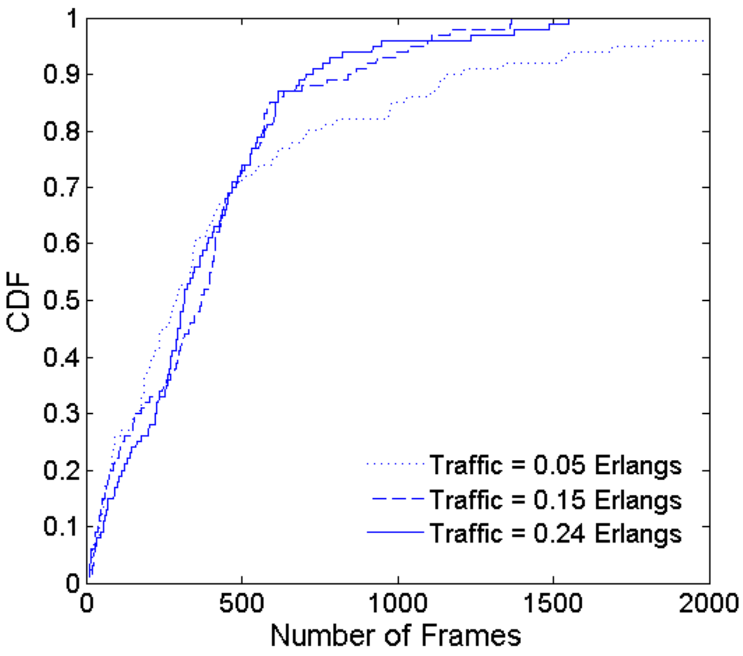


Fig. 6.10 CDF of the converging speed

6.6. ALOHA-QIR for Random Networks

6.6.1. ALOHA-QIR Protocol Design

In this section ALOHA with Q-Learning and Informed Receiving (ALOHA-QIR) is introduced and applied to a more practical network topology: a random deployed network with all nodes (except the sink node) operate as both a source and a relay node. To alleviate the potential problems caused by asymmetric traffic (nodes closer to the sink node have higher traffic levels due to more relay packets), nodes are allowed to use multiple slots per frame. Each frame contains *N* slots and node i is allowed to use n_i slots per frame. *N* is the same for all the nodes in the network but n_i is determined from the number of sources that the node needs to relay, plus one to account for its own packets. For example, if node i relays packets from two other nodes as well as sending packets itself, n_i will be 3.

Stateless Q-Learning [29] is used as the basis of the learning process. Compared with the RL algorithm used in Chapter 4 and the previous section, the weight values of Q-Learning are within the range of the rewards (+1 for success and -1 for failure) when the learning rate is between 0 and 1 [29]. The learning rate controls the size of the step of the Q value towards the current reward during each Q value update. If the learning rate is 1, the updated Q value is equal to the current reward. Each node has individual Q values for every slot, which represent the preference of the slot selections. The Q value is denoted $Q(i,k)$, which represents node *i* taking an action on slot *k*. The previous Q values and the current reward all contribute to the Q value update. The Q value is updated as below after the reward is returned (same to equation 5.1).

$$Q_{t+1}(i,k) = Q_t(i,k) + \alpha\big(r - Q_t(i,k)\big) \qquad (6.6)$$

In equation 6.6 α represents the learning rate and r the current reward. If the transmission succeeds, the current reward will be +1 and if the transmission fails, the current reward will be -1. The slots with higher Q values will always be preferred for use. If multiple slots have the same Q value, the node will randomly select one (or more) from them. All the Q values are initialised to zero at the very beginning. The pseudo code below describes this learning algorithm. *n* indicates the number of slots the node can access per frame.

```
loop sleep until the next frame
        set the Q values to decreasing order, as well as the slots
        if multiple slots have the same Q value
           randomly reset their order
        select the first n slots as scheduled slots
        loop sleep until the next scheduled slot
           if one or more packets in the queue
              send the packet at the queue head
           if ACK successfully arrives
              return +1 reward and update the associated Q value
              retry number = 0
           else
```

```
            return -1 reward and update the associated Q value
            retry number = retry number + 1
            if retry number > retry limit
                discard the packet
    end loop
end loop
```

Figure 6.11 shows an example of the frame structure and the operation of the Q-Learning algorithm. In this example, *N* equals 3 and n_i equals 2.

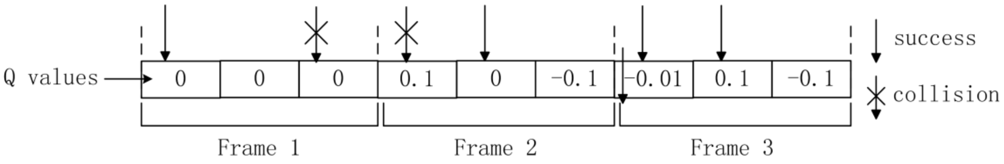


Fig. 6.11 Example of *Q* values and repeated frames

After a packet is generated by the upper layer or arrives from the previous hop, it will be put into a first-in-first-out queue and only the packet at the queue head can be transmitted. A packet needs to wait until the next preferred slot to be sent after its arrival or a collision occurring. A maximum retry limit $M = 6$ is defined according to the IEEE 802.11 specification [76]. A packet will be discarded if it exceeds the retry limit.

After the learning process, each node will find a unique slot (or multiple slots depending on the number of nodes it needs to relay) to transmit packets if *N* is set appropriately. *N* is an important parameter for this scheme, and its optimum value is related to the network topology, the density of the nodes and the number of source nodes. Generally, *N* needs to be set higher when the network has more source nodes and higher density. But if *N* is too large, some slots might be wasted and the delay will increase. To improve the performance of ALOHA-QIR, general knowledge of the network size and topology are essential. In next section, an estimation of the optimum frame size for a random deployed network is provided.

6.6.2. IR and Ping Packets

A. IR

A similar Informed Receiving (IR) as described in Section *6.4.2 A* is applied in this scheme, with some different details. When a node switches on, it listens constantly and waits for the packets from potential transmitters for a certain time period. After this time period, the receiver decides which slot (or slots) to listen in based on the information obtained from the transmitters. The transmitter piggybacks a number m on each data packet to inform the receiver its transmission pattern. m represents the number of future frames the transmitter will keep using the current preferred slot i, and it is calculated by the number of iterations of (6.1). An example is given below to clearly demonstrate the calculation of m.

Given that $N = 4, n_1 = 2$ and the Q values of node 1 are $Q(1,1) = 0.5$, $Q(1,2) = 0.1$, $Q(1,3) = 0.5$ and $Q(1,4) = 0.2$ Slot 1 and 2 are the current preferred slots for node 1 according to the Q values. Considering the worst case that the future transmissions will all fail, the returned reward r will always be -1. $Q(1,2)$ is the smallest value of the current preferred slots, and $Q(1,4)$ is the largest value of the slots which are currently not preferred. Suppose that after 5 iterations of (7.6) (with the reward of -1) $Q(1,2)$ falls less than $Q(1,4)$, then the current value of m will be 5. It represents the number of future frames that the node will keep using the same set of preferred slots. m is calculated before every data packet is transmitted. To avoid unnecessary calculation (when the Q value of the preferred slot is much larger than the second highest Q value), m is given a maximum value of 20.

The receiver has a group of timers T:

$$T = \{t_1, t_2, \dots, t_N\} \tag{6.7}$$

The timer group T has one timer every slot in the frame (the size of the timer group is equal to *N*), and the timers represent the number of future frames the associated transmitters of different slots will use (one node may receive data packets from multiple nodes). The timers are updated at the beginning of each frame (they are all reduced by 1) and after the arrival of the data packets in the associate slots (set to m of the transmitter plus 1 to accommodate the offset caused by the other update). Because each node knows the number of sources it relays, then it needs to listen to the same number of slots per frame n_i (if the node is a source the number will be $n_i - 1$). If the ${n_i}^{\text{th}}$ largest timer is larger than 1, the node will only listen to the n_i slots with larger timers. And if the ${n_i}^{\text{th}}$ largest timer is equal or lower than 1, the node will listen to all the slots to see if the transmitters will switch to other slots.

Based on the previous example, suppose node 3 is the next hop of node 1, and node 2 also selects node 3 as its next hop. Suppose $n_2 = 1$ and node 3 is not a source node, so $n_3 = 3$ and node 3 listens 3 slots per frame. Suppose the preferred slot of node 2 is slot 4 and m of node 2 is 3, and the preferred slots of node 1 are slot 1 and 2 and m is 5, so the timers of node 3 are $t_1 = 5$, $t_2 = 5$, $t_3 = 0$ and $t_4 = 3$. Node 3 will only listen to slot 1, 2 and 4 until anyone of t_1, t_3 and t_4 falls below 2.

B. Ping Packets

Similar ping packets as described in *6.4.2 B* are used in this scheme. The results in Figure 6.10 show that the convergence varies with generated traffic, and it takes more time to converge under low traffic levels because there are fewer learning opportunities. Here we apply a different ping packet to accelerate the learning process and reduce the convergence time at low traffic levels. For the original short ping packets, their collisions indicate that the data packets cannot succeed as well, but their success does not mean the data packets can get through the channel in the same slot because the ping packets are tiny and have a lower probability of collision. So, while updating the weights the original ping packets only contribute to negative reward but not positive reward. In this scheme we introduce long ping packets which have the same length as data packets. When a node is learning and not in relatively steady state (using the same preferred slots in less than 20 frames), it sends long ping packets instead of short ping packets. After receiving long ping packets, both positive and negative rewards can be returned just the same as after receiving data packets. So the weights can be updated every frame and nodes can find optimum transmission slots faster. The overheads caused by long ping packets only exist while learning, and the steady state performance is not affected.

To help understand the protocol design, Figures 6.12 and 6.13 provide to describe the protocol as a transmitter node and a receiver node respectively. The two processes operate simultaneously for each node, but a node does not listen or receive packets when it is scheduled for transmitting.

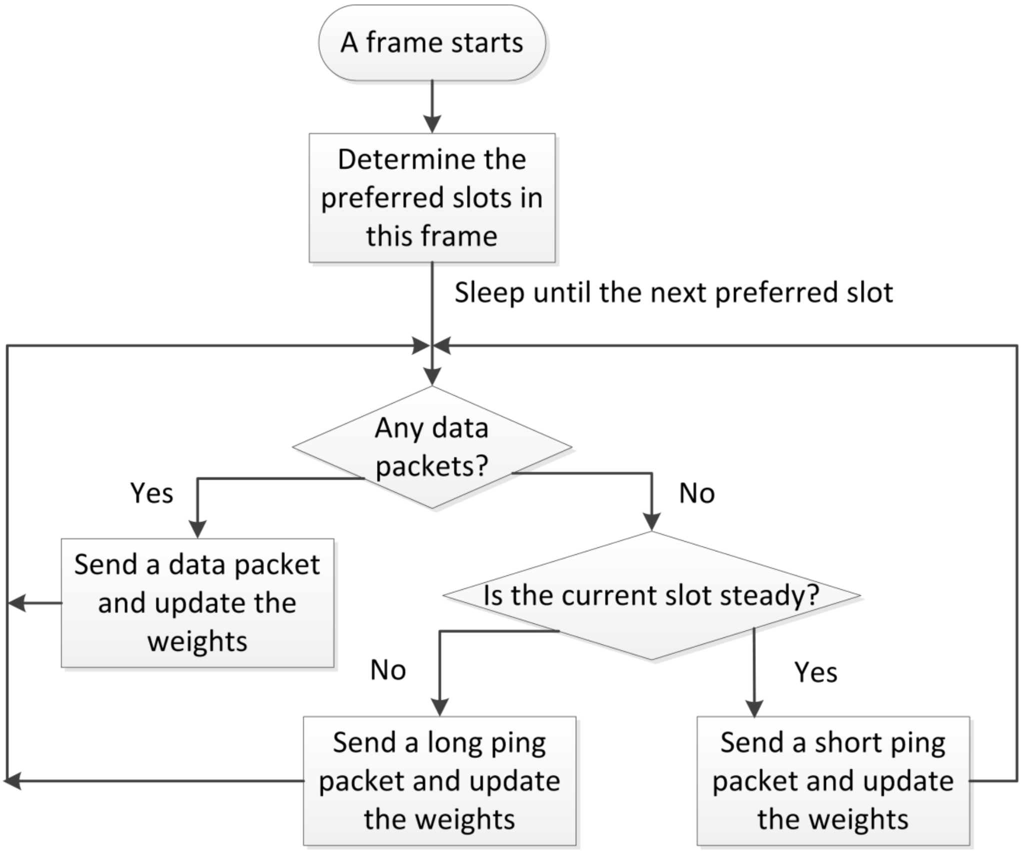

A frame starts
Determine the preferred slots in this frame
Sleep until the next preferred slot
Any data packets?
Yes
No
Send a data packet and update the weights
Is the current slot steady?
No
Yes
Send a long ping packet and update the weights
Send a short ping packet and update the weights

Fig. 6.12 Flow chart of the transmitter node

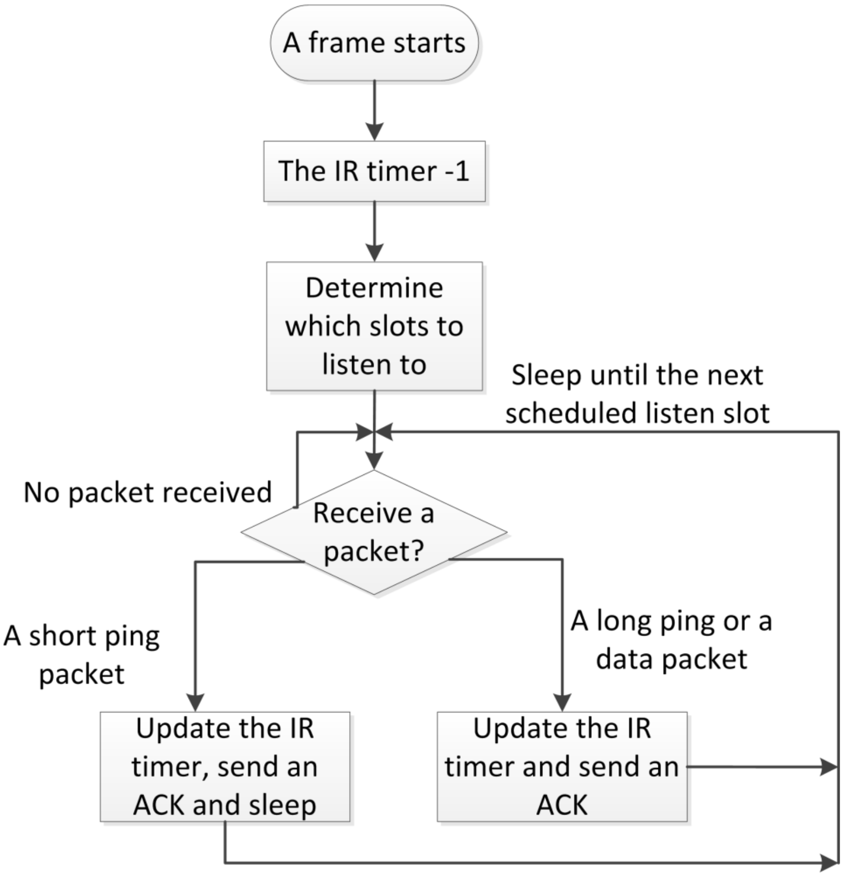

A frame starts
The IR timer -1
Determine which slots to listen to
Sleep until the next scheduled listen slot
No packet received
Receive a packet?
A short ping packet
A long ping or a data packet
Update the IR timer, send an ACK and sleep
Update the IR timer and send an ACK

Fig. 6.13 Flow chart of the receiver node

6.7. Performance of ALOHA-QIR

6.7.1. Scenarios and Parameters

This scheme ALOHA-QIR is simulated and evaluated in OPNET. 50 nodes are randomly deployed in a $50m \times 50m$ square area and all the nodes are sharing the same channel (they all transmit and receive on the same channel). All nodes sense the environmental data, fragment the data into packets with the same length and forward them to the sink. The scenario has one sink node and it is deployed at the centre of the sensing area. All the schemes in the simulations are assumed to be synchronised to slots.

All the nodes are given the same average packet inter-arrival time which is calculated from the overall generated traffic load (Erlangs) given as a simulation parameter. 1 Erlang generated traffic represents the amount of traffic when a node transmits with full channel capacity. The average packet inter-arrival time is obtained as below,

$$\tau = \frac{L \times S}{G \times R} \tag{6.8}$$

τ denotes the average packet inter-arrival time (seconds), L denotes the length of a data packet (bits), S represents the number of source nodes in the network, G is the generated traffic of the network (Erlangs) and R denotes the data rate of the channel (bits/second). In the simulations each node can store a maximum of 200 packets in the buffer, and any more incoming packets beyond this will be discarded directly. For example an IRIS node [31] has a flash memory of 128 Kbytes, which is able to save a maximum of about 1000 data packets. We set this buffer size to a relatively low value to avoid inaccurate performance in steady state caused by too many queued packets during the learning process.

Considering the complexity of implementing a full interference model, the radio transceiver in the OPNET models operates under following assumptions. During packet reception, any partial or entire overlap of the packet will be considered as a failure. The propagation delay is ignored because the nodes are close to each other, and the propagation delay is tiny compared with the queuing delay of a packet and the packet duration. Only packets within a certain radius of a node can be correctly received, but transmissions within a larger radius of the node can cause interference and collisions. Some radio transceiver parameters are obtained from the IRIS motes datasheet [31]. Slotted ALOHA with BEB, SMAC and Z-MAC are used for comparison. Simulation parameters are given in Table 6.2. SMAC uses 200 slots per frame and 10% duty cycle. Z-MAC has 120 slots per frame, 0.5 bits contention slots, 8 contention slots for slot owners and 32 slots for non-owners.

Parameters	***Values***
Channel bit rate	250 Kbits/s
Data packet length	1044 bits
ACK packet length	20 bits
Ping packet length	28 bits
Slot length	1100 bits
Transmit power	51 mW
Receive power	48 mW
Interference range	30 m
Receive range	15 m
Simulation length	5×10^6 slots

Table 6.2 Simulation parameters

6.7.2. Optimal Frame Size and Maximum Throughput Estimation

Figure 6.14 shows a simple overhead view of the network described in section 6.7.1. The square is the sensing area where nodes are randomly deployed, the inner circle is the reception range of the sink node and the outer circle is the interference range of the sink node. According to section 6.6.1 nodes can access multiple slots per frame based on the number of sources they relay. Therefore, nodes in the inner circle must have the heaviest contention in the network and the optimum frame size can be estimated based on these nodes.

According to the number of nodes and the area of the sensing field, the node density is $0.02\,nodes/m^2$. The nodes in the inner circle need 1 hop to the sink, nodes in the area between the inner circle and outer circle need 2 hops, and nodes outside the outer circle need 3 hops. We need to calculate the area to determine the average number of nodes with different hops to the sink. The area of the inner circle is $707m^2$, so the average number of nodes 1hop to the sink is 14.14 (28.28%). The area of the nodes 3 hops to the sink can be represented by $8S_c$

(the area of the four corners). We can obtain that:

$$S_c + S_a = 208m^2 \tag{6.9}$$

$$b = \arcsin\left(\frac{\sqrt{r_2^2 - r_3^2}}{r_2}\right) = \arcsin\left(\frac{16.58}{30}\right) = 0.5856 \tag{6.10}$$

$$a = \frac{\pi}{4} - b = 0.2 \tag{6.11}$$

$$S_a = r_2^2\pi\frac{a}{2\pi} = 180m^2 \tag{6.12}$$

$$8S_c = 224m^2 \tag{6.13}$$

So there are on average 4.48 (8.96%) nodes needing 3 hops to the sink, and 31.38 (62.76%) nodes needing 2 hops to the sink. Suppose all nodes generate 1 packet, then 3-hop nodes need to relay 4.48 packets, 2-hop nodes need to relay 35.86 packets, 1-hop nodes need to relay 50 packets. So there are 85.86 packets contending within the interference range of the sink node, and it is also the theoretical optimum frame size. However considering the random deployment (a part of the network could have larger density than the average) and our previous experience (the scheme performs worse with small frame size, but has acceptable performance with a larger frame size), we expect the optimum frame size in the simulations to be larger than the theoretical value 85.86. And the maximum throughput will be about 0.5 Erlangs (depending upon the frame size) without considering any overheads.

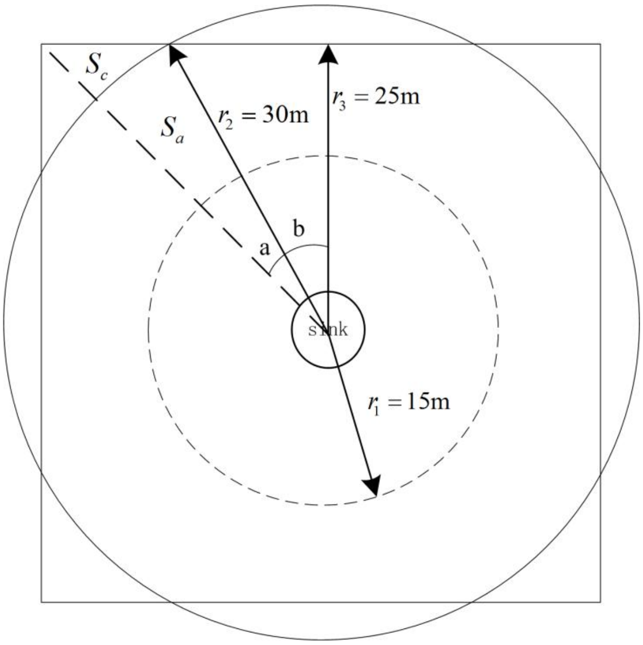


Fig. 6.14 Network structure

6.7.3. Channel Performance

Figure 6.15 compares the throughput performance of ALOHA-QIR to basic Slotted ALOHA with different frame sizes N and varying traffic levels. The throughput of the four schemes grows with the traffic linearly (which indicates that all generated data is delivered) before the generated traffic reaches certain levels. As the traffic increases, the throughput of the four schemes converges to different limits. ALOHA-QIR (N=100) is the optimum and has a maximum throughput of approximately 0.47 Erlangs, which is over twice that of the Slotted ALOHA (0.22 Erlangs) and close to the theoretical maximum of about 0.5 Erlangs. Frame size N is an important parameter for ALOHA-QIR, and affects the performance significantly. According to the learning algorithm each node i will own n_i unique slots of its next hop in steady state to avoid collisions. However if N is smaller than the optimum, some nodes may find their unique slots while others have to share, which causes collisions (it has been evaluated in Chapter 4). If N is larger than the optimum, the system can reach a steady state and have no collisions but some slots may never be used, which wastes the channel capacity.

Therefore ALOHA-QIR with 120 slots per frame has lower throughput than 0.47 Erlangs, and the throughput stays the same as the increasing generated traffic until about 0.39 Erlangs. SMAC achieves the throughput of about 0.36 Erlangs because the collision avoidance of SMAC suffers severely from two-hop interference. Z-MAC has less maximum throughput than ALOHA-QIR with 100 slots per frame but better than the ALOHA-QIR with same 120 slots per frame, due to its larger overheads but more flexible slot selection strategy.

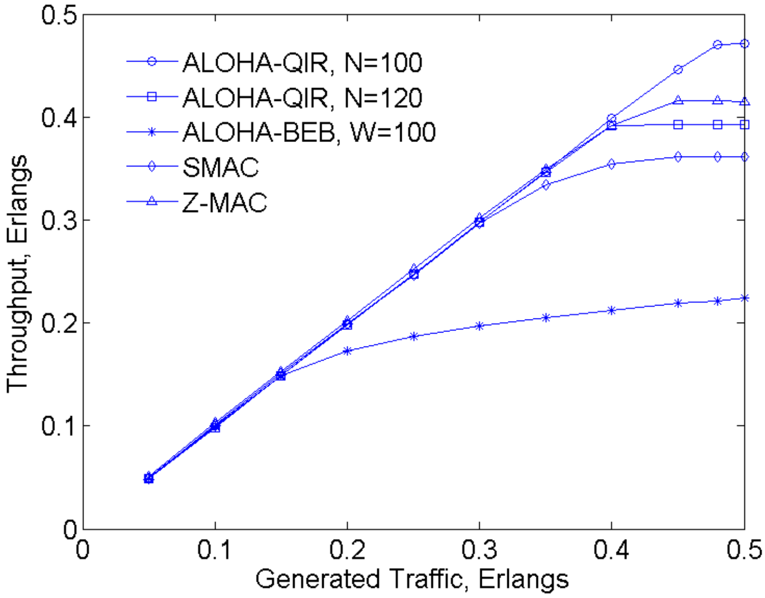


Fig. 6.15 Throughput under different traffic levels

Figure 6.16 shows the end to end delay of the four schemes. ALOHA-QIR with *N*=100 maintains a delay of less than 1 second under most traffic levels. *N*=120 has slightly higher delay because of the larger frame size. ALOHA-BEB has the lowest delay when the traffic is very low because it immediately sends the packet in the next slot after the packet is generated or received and does not have to wait until the next preferred slot. SMAC has high overall delay because of the larger frame and duty cycle. Z-MAC achieves best delay performance at low traffic levels, resulted by its immediate access to the channel at LCL. When the traffic load increases, Z-MAC has similar delay as the ALOHA-QIR schemes.

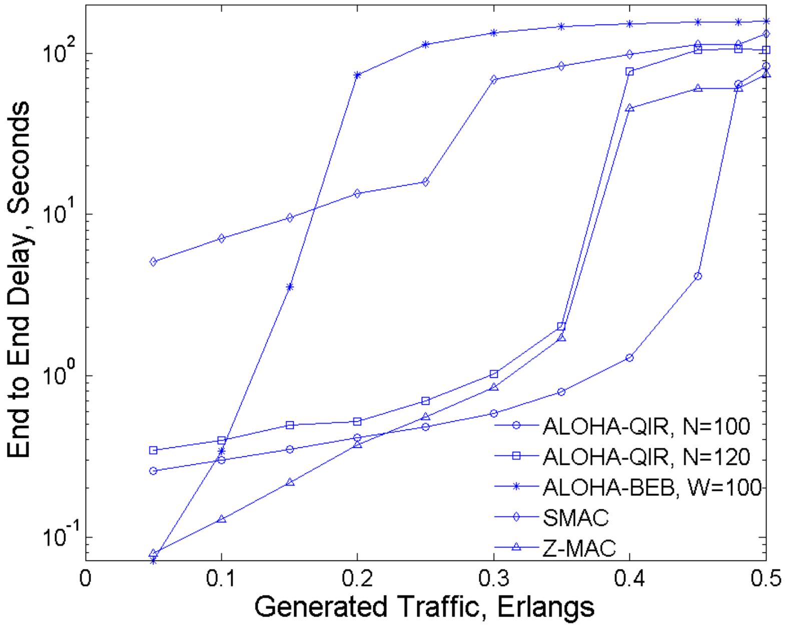


Fig. 6.16 Delay under different traffic levels

6.7.4. Energy Efficiency

Figure 6.17 to 6.19 demonstrate the energy efficiency of ALOHA-QIR. In Figure 6.17, ALOHA-QIR with *N*=100 and 120 has similar energy cost per bit because they all avoid most of the collisions and achieve perfect scheduling. The slightly higher energy cost at low traffic levels is caused by the ping packets, because nodes transmit many ping packets instead of the data packets under low traffic conditions. It is shown more clearly in Figure 6.19 that the network consumes 70% energy on data under low traffic and over 95% energy when the traffic is high. ALOHA-BEB has the highest energy cost because it consumes a large amount through idle listening, just as Figure 6.19 shows, only a tiny proportion of energy is consumed by data. SMAC and Z-MAC achieve steady energy cost per bit at different traffic levels, but higher cost than ALOHA-QIR due to their larger overheads and idle listening, especially SMAC suffers more collisions and retransmissions. Figure 6.18 shows the energy cost per second of the five schemes. The energy cost of ALOHA-QIR with *N*=100 and 120 is dominated by data so that their energy cost per second increases linearly with the traffic. SMAC and Z-MAC have the similar trend but higher values compared with ALOHA-QIR. Moreover, when the traffic is 0.5 Erlang, ALOHA-QIR with N=100 consumes about 95 mW and ALOHA-BEB 2476 mW, which means that the network can survive at least 25 times longer.

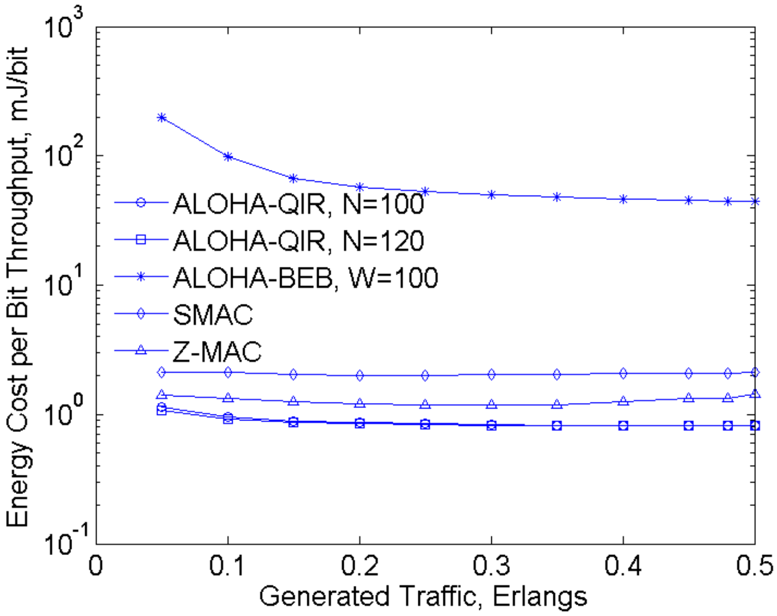


Fig. 6.17 Energy cost of the network per bit throughput

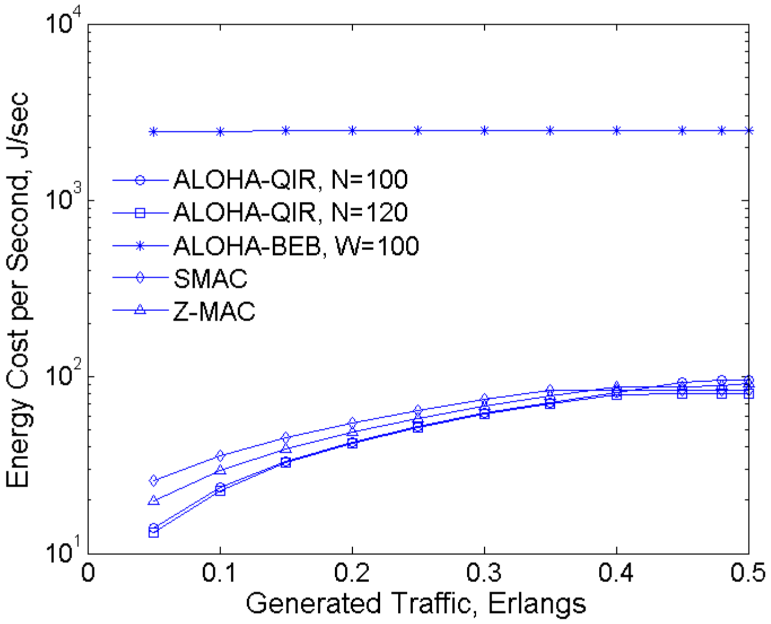


Fig. 6.18 Energy cost of the network per second

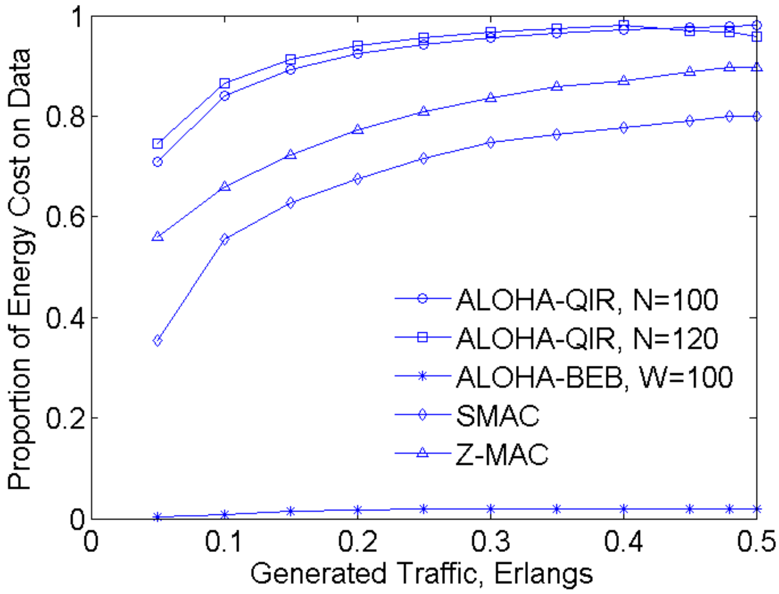


Fig. 6.19 Proportion of energy cost on data transmission and reception

6.7.5. Convergence Time

Figure 6.20 shows the CDF of the number of frames elapsed before converging to the optimum solution under different traffic levels, with each curve containing 100 simulation results. The scheme in this Figure does not include long ping packets. All the simulations converge within 3000 frames (300,000 slots), so simulation data collection is started after 500,000 slots to make sure the system is under steady state. Under medium traffic (0.2 Erlangs) about 90% of simulations converge within 500 frames (about 220 seconds) which is quite tolerable for the initialisation of a network. The scheme with light traffic (0.05 Erlangs) takes more time to converge due to the less learning opportunities under such low traffic level.

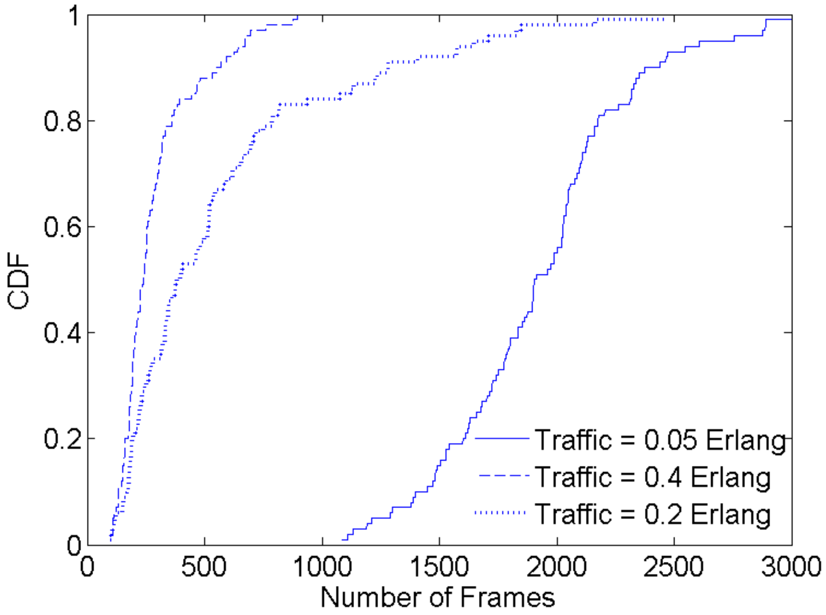


Fig. 6.20 CDF of the convergence time under different traffic levels without long ping packets

To solve the slow convergence problem under low traffic conditions, long ping packets are introduced. Figure 6.21 shows the CDF of the convergence time under different traffic levels with long ping packets. Each curve contains 400 simulation results. Results show that the two curves almost overlap, which indicates that the traffic level does not affect convergence time any more. Over 90% of the simulations converge within 500 frames (220 seconds) which is acceptable for the initialisation time of a WSN.

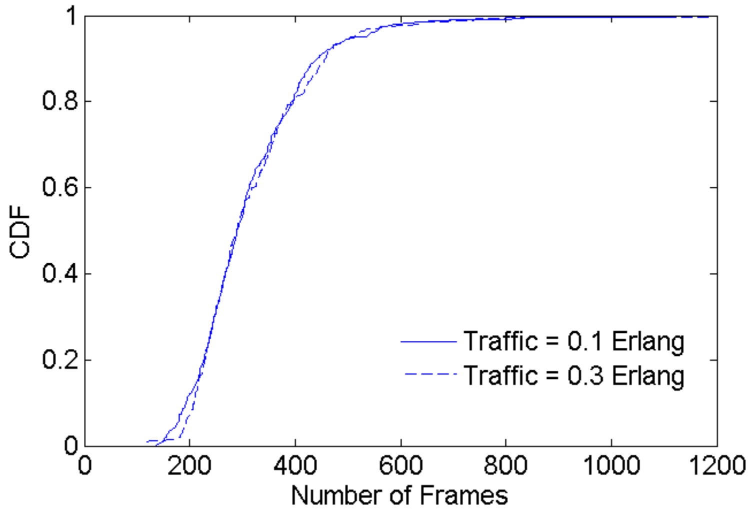


Fig. 6.21 CDF of the convergence time under different traffic levels with long ping packets

6.8. Conclusions

This chapter has provided a detailed study of applying RL and ALOHA to multi-hop WSNs. In a multi-hop network, it is important to consider not only the channel performance but also the energy constraint. Our original RL algorithm in Chapter 4 relies on constant listening of the receiver radio, which is not acceptable in multi-hop WSNs. RL and ping packets are applied to efficiently switch nodes between sleep and active modes by providing receiver nodes with the learning information of the associated transmitter nodes, thereby maintaining the channel performance and improve the energy efficiency.

Two protocols are proposed in this chapter. RL-ALOHA was first applied in a linear chain network, also exploring the idea of IR and ping packets. Then the RL algorithm was applied with IR and ping packets in a random network, presenting ALOHA-QIR. Results show that both schemes significantly improve the throughput and delay results compared with schemes without intelligence, demonstrating that the throughput approaches the theoretical maximum. Both schemes achieve over 60% energy cost on data under very low traffic and almost 100% energy cost on data under high traffic conditions.

This chapter focused on steady state performance rather than the learning process prior to convergence. For the schemes with only short ping packets, the traffic level has a significant impact on convergence time because it directly affects the learning opportunities. With the introduction of long ping packets, schemes with different traffic have similar convergence time, and 90% of the simulations converge within 220 seconds, which is acceptable given the initialisation time of a WSN.

7. Further Work

Contents

This chapter discusses potential further research related to the work in this thesis, to explore various capabilities of the protocols, and to analyse more complicated schemes.

7.1. Different RL algorithms

In Chapter 4 to Chapter 6, RL and Q-Learning have been applied as the intelligent slot selection algorithms. According to the experiments and simulations, both schemes have been shown to converge to a steady state and achieve significant improvements in channel performance compared with the non-intelligent schemes. However, the current RL algorithms are based upon single-agent learning where nodes learn independently from the environment without cooperation or information exchange. Multi-agent RL is a potential technique to improve the convergence time and steady state performance. For example in Chapter 6, optimum delay performance exists in certain slot selection patterns (packets are always relayed in the next slot). The current RL schemes ensure no collisions but are not optimised for delay performance. Through multi-agent RL, nodes can cooperatively select slots to achieve optimal steady state. On the other hand, the extra negotiation and information exchange increases overheads, so the trade off between the gain and costs would need to be considered.

7.2. Robustness to Synchronisation Errors

In Chapter 4, the robustness of the RL scheme to synchronisation errors was evaluated in a single-hop network. Results show that the maximum throughput decreases from 0.9 Erlangs to about 0.6 Erlangs. Global synchronisation was assumed when implementing protocols in multi-hop networks in order to obtain the capabilities of RL schemes. Practically, it may be difficult to achieve global synchronisation in a WSN. Instead of seeking global synchronisation, improving the robustness and adaptability of the protocols to synchronisation errors is another alternative, because it introduces less overhead and has lower complexity. The initial idea is to use the same RL algorithms for slot selection and packet transmission, and to adjust the nodes' sleep and wake up schedules to the clock of the associated transmitters to avoid idle listening and overhearing. Performance reduction is inevitable under unsynchronised conditions but it should be kept as low as possible.

7.3. Cross-Layer Protocols

MAC of the data link layer and routing at the network layer both have a significant impact on energy efficiency of WSNs. MAC protocols and routing protocols are highly related and can be integrated to one protocol with the functions of them both. Researchers have proposed some cross-layer protocols (for example LEACH) which aim to improve energy efficiency and channel performance from a different perspective. Except for collision avoidance at the MAC layer, cross-layer protocols also determine the topology of the network as the routing protocol. From the network layer point of view, network lifetime can be improved by evenly assigning relay routes to avoid draining energy of single nodes, and selecting routes based on nodes' remaining energy or traffic load. With a known network topology, the MAC protocol can work more efficiently and cooperatively. In the simulations of our protocols, we use MFR (most forward routing, or shortest path) for routing which is simple but offers no improvement to energy efficiency. With more topology information, selecting an optimal frame size will be easier, channel performance and energy efficiency can be improved by adjusting the MAC layer option to the current topology. Note that dynamic routing may cause inefficiency because nodes need to relearn to reach steady state. The trade-off between the performance loss before convergence and the gain of new routes needs to be considered.

7.4. Mathematical Analysis for RL

In Chapter 5 a Markov model was proposed for a certain RL scheme, the convergence was proved and the expected convergence time was obtained. However, to implement the Markov model we set constraints and limitations on the scheme so that it has no memory and time-relevance. The simulation results matched the analysis, which confirmed the correctness of the Markov model. Analytical models can be developed for other RL schemes based on the prior experience to obtain a more in depth understanding of learning.

7.5. RL on Frame Size

Frame size is an important parameter to our protocols and it significantly affects channel performance and energy efficiency. Note that nodes are allowed to use certain number of slots per frame (1 slot per frame in single-hop networks, the slot number in multi-hop networks depends on the number of sources each node relays for). The total number of slots which can be used in each frame is fixed (considering the nodes in a certain neighbourhood), such that an insufficiently short frame size causes collisions and a large frame size brings additional delay and reduction in maximum throughput. The optimal frame size is determined by many factors. In early simulations, the optimal frame size was based on the topology and set manually. It would be beneficial to apply RL to determine the frame size dynamically so that the protocol does not need to rely on prior frame size estimation. By considering the past throughput, delay and collision probability (could be a certain time window or certain number of frames), nodes can adjust their frame size individually, and a certain technique is required to make sure that nodes in the same neighbourhood all converge to the same frame size. RL applied to slot selection can work on parallel, but the change of frame size may affect the learning results and nodes need to relearn. The convergence time of two parallel RL schemes is expected to be larger, but the protocol can adapt to different topologies.

8. Summary and Conclusions

Contents

This thesis has presented the research work undertaking during Ph.D. study from 2009 to 2012 at the University of York. The early chapters introduced the purpose of the research, background knowledge and related work, and necessary research methodology. The later chapters demonstrate the detailed research work, which mainly focuses on developing energy efficient MAC protocols for WSNs, based on the ALOHA schemes and Reinforcement Learning.

According to the energy and hardware restrictions of WSNs, the research work in this thesis focuses on the principles of energy efficiency, low computational complexity and low signalling overheads. The ALOHA schemes meet the requirements of low complexity and overheads, but suffer low throughput and energy efficiency from the random transmission strategy. RL techniques are applied as intelligent slot selection strategies to avoid collisions and improve energy efficiency, and perfect scheduling is achieved in both single-hop and multi-hop WSNs without additional overheads. IR and ping packets are applied to multi-hop WSNs to avoid idle listening and overhearing, and further improve the energy efficiency. A Markov model is provided to estimate the convergence time of the learning process.

General background information of the WSNs is provided in Chapter 2. WSNs are usually self-organised distributed networks which can be used in a wide range of applications. This chapter briefly introduces the history, network structure, applications, design requirements, devices and constraints of WSNs, as well as general information of RL, which is the main technique applied in MAC protocol development in later chapters.

Background information and a related literature review of MAC protocols for WSNs are provided in Chapter 3. The basic MAC schemes are introduced, along with the specific design requirements of protocols for WSNs. Contention-based protocols are natural approaches for distributed networks, and usually have good performance under low traffic conditions. Schedule-based schemes are the alternative, which ensure no collisions and have good performance under high traffic conditions, but with greater overheads. Several representative MAC protocols for WSNs are introduced in this chapter and categorised into three main types.

To avoid collisions and improve energy efficiency of the ALOHA schemes, Chapter 4 introduces RL as an intelligent slot selection strategy to framed ALOHA. Nodes select transmission slots by considering their prior transmission history, as long as they have enough trials and on appropriate frame size, collisions can be avoided and perfect scheduling can be achieved. Results show that the maximum throughput is improved to almost three times compare to the random access scheme.

In Chapter 5, ALOHA and RL are applied to single-hop WSNs, Chapter 6 extends the work and explores it to multi-hop networks. Two network topologies are considered in this chapter: a linear chain network and a randomly deployed network. IR and ping packets are applied to avoid idle listening and overhearing, thereby switching nodes to a sleep state as long as possible without missing any information and improve the energy efficiency. Results show significant improvement in both channel performance and energy efficiency. Compared with S-MAC and Z-MAC, the RL schemes have achieved greater maximum throughput lower end to end delay over most traffic levels (except that Z-MAC has better delay performance under very low traffic levels).

RL algorithms provide perfect scheduling in steady state, but they require a certain time for convergence and the performance before convergence is not optimum. Chapter 5 focuses on the understanding to the convergence behaviour of RL algorithms. The convergence time is related to many factors including network size, network topology, routing protocols, node density and communication range. According to the uncertain environment in the learning period (where the environment is all other nodes which are also learning), the convergence time spreads over a large range in every individual trial. This chapter provides a Markov model to estimate the convergence time of a simple RL algorithm in a single-hop network, and validates it by simulations. Performance results during the learning period are also provided and discussed in this chapter.

8.1. Novel Contributions

As far as we know, this thesis has first explored the application of RL and ALOHA to MAC protocols for WSNs. The main contributions are summarised as follows:

8.1.1. Intelligent Slot Selection Strategy Based on RL

Applying RL as an intelligent slot selection strategy is a novel concept to solving the collision and low throughput problems in random access schemes. Nodes learn their own slot selections through trial-and-error interactions to avoid collisions. Perfect scheduling can be achieved in steady state with an appropriate configuration. No additional packet exchanges are introduced by the learning algorithm. Chapter 4 has evaluated the performance of RL algorithms in a single-hop network. Results show that the protocol achieves over 0.9 Erlangs throughput, which is almost three times (less than 0.36 Erlangs) compared to the scheme without intelligence. This chapter has also evaluated the performance of the RL scheme under high propagation delay and unsynchronised conditions. The throughput drops with longer propagation delay but is still close to the theoretical maximum. The maximum throughput decreases by about 30% when the network is not synchronised.

8.1.2. IR and Ping Packets for Multi-hop Networks

The work in Chapter 4 ensures no collisions in the steady state but it is based on the assumption that the sink node is always listening. In multi-hop WSNs, the relay nodes must switch to sleep state considering the energy and lifetime issues of WSNs. In Chapter 6 IR is applied for the transmitter nodes to provide the recent learning information to the associated receiver nodes through usual data packets, so that they can switch to active state necessarily and sleep as long as possible. Ping packets are applied to provide the receiver nodes IR information when the transmitter nodes have low traffic. The long ping packets can also add more trials under low traffic conditions to accelerate the learning process and improve the convergence time. These contributions have been presented at the ***IET Conference on Wireless Sensor Systems (WSS2012)*** and the ***International Symposium on Wireless Communication Systems (ISWCS2012)***.

8.1.3. Average Convergence Time of RL

Perfect scheduling can be achieved by the RL algorithms but they require a certain time to converge to the steady state. During the learning process the system performs suboptimal and this learning period can be considered as the initialisation stage of the network. Chapter 5 provides the research of the learning behaviour, based on the example of a simple RL process. A Markov model is proposed to describe this learning process. The convergence is proved through this model, and the average convergence time is obtained. The channel performance in learning stage is also provided, and to show the convergence behaviour of the RL algorithm in Chapter 6. These contributions have been **submitted to IEEE Transactions on Wireless Communications**.

8.2. Publications

Conference Presentations

Y. Chu, P. D. Mitchell and D. Grace, “ALOHA and Q-Learning based medium access control for wireless sensor networks”, International Symposium on Wireless Communication Systems (ISWCS 2012), pp. 511-515, 2012.

Y. Chu, P. D. Mitchell and D. Grace, “Reinforcement learning based ALOHA for multi-hop wireless sensor networks with informed receiving”, IET Conference on Wireless Sensor Systems (WSS 2012), pp.1-6, 2012.

Submitted

“Application of Reinforcement Learning to Medium Access Control for Wireless Sensor Networks”, Yi Chu, Paul D. Mitchell, David Grace and Tim Clarke, submitted to IEEE Transactions on Wireless Communications.

Appendices

I. Modelling Techniques and Validation Methods

This section provides an overview of the simulation tools and mathematical validation methods which are used in the thesis. This section also introduces the important performance measurements which are used in protocol performance analysis.

The primary simulation tool used is OPNET modeller. OPNET is an event-based network simulation tool which constructs the network through three levels of models: process models, node models and network models. Network topology, device functions and protocols of different layers of the Open System Interconnection (OSI) stack [1] can all be described by these three models. With its kernel procedures and transceiver pipeline stages, OPNET is capable of simulating a close-to-real communication environment [53]. The event-driven and interrupt based operations of OPNET are well suited the MAC protocol simulations.

MATLAB is well-known scientific computing software which is widely used in academic, research and industry. It is able to perform sophisticated matrix computation, implement algorithms and plot data [54]. With various extended toolbox functions, it can also directly control hardware and perform more tasks such as transmission, reception and analysis of acoustic signals, fragmentation and synthesis of images. The advantage of MATLAB is its potential for matrix calculation and the variety of tool box functions provided. It is widely used in communication research. The data analysis capability of MATLAB is very important to the work in this thesis. The analytical results generation and matrix computation in Chapter 5 are all performed by MATLAB.

II. OPNET Radio Transceiver Pipeline Stages

All the transmissions in the network can be subject to various adverse effects due to channel fading, interference, noise and transceiver hardware problems. The radio transceiver pipeline simulates the basic influences upon a transmission including physical phenomenon (such as distance and noise) and characteristics of the physical layer (such as the channel and error correction). In OPNET models, packets go through 14 stages from transmission to reception to determine if this packet is correctly received. Some stages are modified for simulations in this thesis.

Stage 0 is usually determined at initialisation state, and stages 1 to 13 are applied at every packet transmission to determine whether the packet is correctly received. The failure of certain stages will cancel the rest of the stages, which will be considered as the failure of the transmission.

- Stage 0 receiver group: this is not a real part of the dynamic pipeline stages [61]. It provides a virtual receiver group to the transmitter and determines the nodes that can receive packets from it. In practical systems this can correspond to frequency distinction or physical separation.
- Stage 1 transmission delay: calculates the time needed to complete the entire packet transmission.
- Stage 2 link closure: determines whether the signal can reach the receiver channel and whether the receiver channel can accept it.
- Stage 3 channel match: it is operated at each individual receiver channel and it classifies the transmission with respect to the receiver channel, one of three states will be assigned to the received packets, valid, noise or

ignored.

- Stage 4 transmitter antenna gain: executed for each destination channel and it computes the transmitter antenna gain based on the vector between the transmitter and the receiver.
- Stage 5 propagation delay: calculated from the time from the packet leaving the transmitter to arriving at the receiver.
- Stage 6 receiver antenna gain: executed for each destination channel and it computes the receiver antenna gain based on the vector between the transmitter and the receiver.
- Stage 7 received power: computes the received power of the signal by considering the distance and shadowing.
- Stage 8 interference noise: calculates the interaction among the transmission simultaneously arrived at each receiver channel.
- Stage 9 background noise: counts all the environmental noise except the interference caused by other transmissions.
- Stage 10 signal-to-noise ratio: calculates the average power SNR of the arriving packets by substituting the results obtained in the prior stages.
- Stage 11 bit error rate: derives the probability of bit errors during the past interval of constant SNR.
- Stage 12 error allocation: estimates the number of bit errors in a packet segment where the bit error probability is constant.
- Stage 13 error correction: the final stage to determine whether the arriving packet can be accepted and forwarded to the neighbour modules of the receiver nodes.

Modified pipeline stages are applied to meet assumptions in the simulations in later chapters. The default receiver group (stage 0) is all other nodes in the network. In the multi-hop network simulations in Chapter 6, one-hop reception and two-hop interference (nodes can only receive packets from their one-hop neighbours but the simultaneous transmissions from the two-hop neighbours can cause interference and collisions) are applied by modifying the receiver group. The receiver group of a node consists of other nodes within the distance of two-hop radius, which can physically receive its packets and potentially cause collisions. Each node also has a group of node addresses which consists of the address of all its one-hop neighbours. After receiving a packet the node first checks the address of the source node, if the source node is not a one-hop neighbour the packet will be destroyed directly to meet the one-hop reception and two-hop interference assumption described above. According to the short communication range of sensor nodes [31], the propagation delay of a packet is negligible to the packet length. So the propagation delay (stage 5) is set to zero in most of the simulations in this thesis, except for the simulations in Section 4.3.6 in which the robustness of the scheme against different propagation delays is tested. The default error correction (stage 13) is to determine whether the packet is correctly received by considering the number of error bits. To meet the theoretical assumption that any partial or entire overlap of multiple packets causes failure of all packet receptions, the error correction is modified to consider the packet reception failed if it has one or multiple number of collisions. This assumption considers the worst case during the packet reception to obtain the precise simulated performance, because the data part of a packet can potentially be correctly received if the packet is partially overlapped (for example the preamble is collided).

III. Performance Metrics

To quantify system performance, many performance metrics are collected and calculated during simulations. These metrics will be used in later chapters to evaluate the system QoS, energy efficiency and convergence performance.

- *Throughput*

Throughput is the successful traffic arriving at the sink and it is an important metric for the system QoS. In this thesis we consider the throughput including all overheads, which has the maximum of 1 Erlang (however, this is not achievable in a practical system because overheads are essential to ensure the correct reception of data). Where 1 Erlang means the throughput achieves the channel capacity. The throughput is calculated as below:

$$S = \frac{R}{C\,T} \tag{1}$$

R is the total amount of data (bits) received by the sink node. C is the data rate (bits/sec) of the receiver channel (the channel capacity), and T is the period of time (sec) during which to calculate the throughput S (Erlangs). In Chapter 5 we calculate the throughputs over different time periods, which are the throughputs of different time windows, where each window starts at the beginning of the simulation and ends at different time points.

- *Delay*

Delay is another important QoS metric for communication systems, which indicates the timeliness of the data. Delay is a critical measurement in systems with real-time applications such as cell phone networks, live video flows and security systems. In this thesis we use end-to-end delay, which is the time between the generation of the packet and the time that the packet successfully arrives at the sink node. Network size, density, routing and MAC all have a significant impact on delay performance.

- *Channel Capacity*

Channel capacity is the maximum amount of traffic that can be transmitted on the channel. In this thesis the same data rate is set for the transmitter and receiver channels, so the data rate becomes the channel capacity. In the results analysis we usually define channel capacity as 1 Erlang.

- *Offered Traffic*

Offered traffic is the amount of traffic that users transmit on the channel. For a single-hop network it is the total output traffic of all users including overheads. For a multi-hop network it is difficult to define due to many reasons. For example the packets have different destinations, the nodes have different communication coverage and the repeating packets in the network (relayed packets). We use generated traffic instead, which is the traffic consisting of new packets generated by the source users. These metrics use Erlangs as their unit as well.

- *Channel Efficiency*

Channel efficiency is the ratio of dividing the throughput by the offered traffic, which is relevant to packet delivery rate and shows the proportion of useful traffic successfully received by the receiver.

- *Energy Efficiency*

We use four metrics to show the energy efficiency of a system. The *number of transmissions per success* the number of transmission attempts (data packets) divided by the number of received packets at the sink node. It is used to indicate the energy efficiency in single-hop networks in Chapter 4 and is relevant to packet delivery rate. However this definition is not practical in multi-hop networks according to the same issues of measuring offered traffic, so three other metrics are defined. The *energy cost per bit useful throughput* (mJ/bit) is calculated by total energy consumption (use mJ as unit, including energy consumed by transmitting and receiving packets, idle listening and overhearing) divided by the amount of data (bits) received at the sink node, which directly shows the energy efficiency. The *proportion of energy cost on data* is obtained by energy consumed on transmitting/receiving data packets divided by total energy consumption, which indicates the amount of overheads during the transmissions. The *energy cost per second* (mW) is calculated by total energy consumption divided by simulation time, which shows the network lifetime. Note that all energy metrics are calculated network-wide, which includes the energy consumption of all nodes in the network.

- *Convergence Time*

Convergence time is an important metric to our learning schemes, with a detailed study of convergence behaviour provided in Chapter 5. Convergence time can be considered as the initialisation time of a network (the time the network requires to obtain optimal performance), but before convergence the network still provides a certain suboptimal QoS. Specifically for our learning schemes, convergence time is the time the network takes for all nodes to find their unique slots and transmit without collisions.

IV. Framed ALOHA with Exponential Backoff

Framed ALOHA applies repeating frames to slotted ALOHA, and it is the basis of applying Reinforcement Learning in this thesis. Each frame consists of a certain number of slots, and nodes can access different numbers of slots each frame to solve the asymmetric traffic problem (packets can be queued with equal access when the nodes have different traffic levels). A user checks the number of packets in its queue and randomly schedules slots for them at the beginning of each frame. The packets generated during a frame need to be queued at least until the next frame. To achieve better channel performance, the frame length and number of packets allowed to be transmitted in each frame need to be set properly. Generally, networks with a high density of nodes and contention require large frame lengths, and users with more traffic can access a greater number of slots per frame. Framed ALOHA is the basis of applying RL in Chapter 4, 5 and 6, simulations.

Figure 9.1 shows an example of framed ALOHA. User A has a packet generated before frame 1, and it is successfully transmitted in frame 1. User B has a packet generated in frame 1 and this packet is successfully transmitted in frame 2. Both A and B have packets generated in frame 2 and they collide when transmitted in the same slot in frame 3. A and B then select different slots in frame 4 to retransmit their packets. Framed ALOHA is also sensitive to synchronisation errors as described previously for slotted ALOHA.

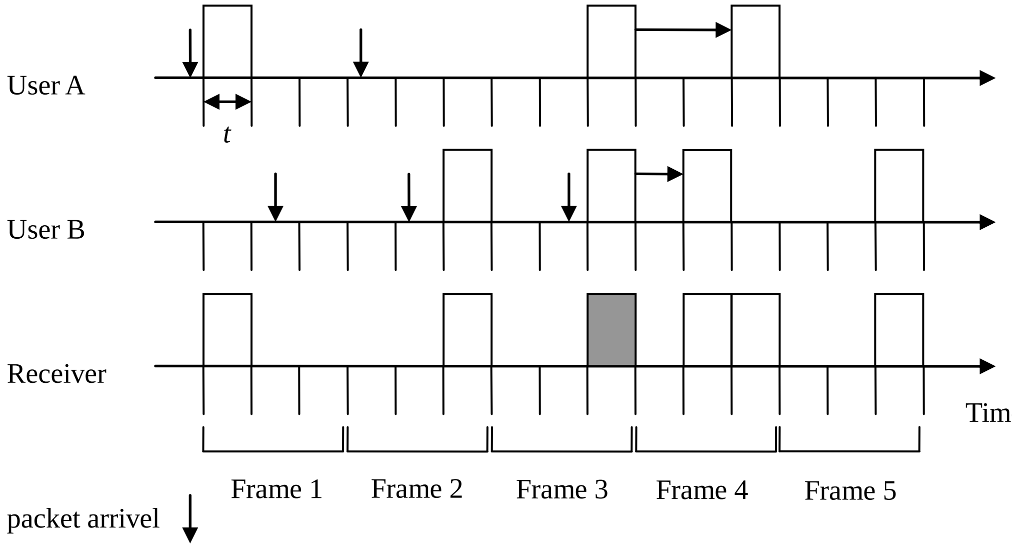


Fig. 9.1 An example of framed ALOHA

Framed ALOHA with Binary Exponential Backoff and Maximum Retry Limit (EB-F-M) is used for performance comparison in Chapter 4. Binary Exponential Backoff (BEB) is a retransmission strategy. Each node has an initial contention window. Once it suffers a collision, the window size is doubled to backoff its traffic flow, and the window size returns to the initial value one after successful retransmission. The offered traffic decreases according to the backoff, thereby reducing the probability of collision. To avoid large queuing time, packets that exceed a certain maximum retry limit are dropped and the window size returns to its initial value. For EB-F-M, the initial contention window size is set to the length of one frame, and the maximum retry limit is set to 6 according to the IEEE standard [74].

Figure 9.2 shows an example of EB-F. The frame length is 4 slots and the backoff factor is 2. In the first frame, nodes A and B do not schedule transmissions for arriving packets because the contention window cannot start in the middle of a frame. After the packets arrive (packets are kept in a first-in-first-out queue in each node) in the first frame, both of them start a contention window from the start of frame 2. The packets arrive in the middle of a frame have to wait at least until the next frame to be sent because nodes schedule transmissions at the beginning of each frame. As Figure 9.2 shows, the two transmissions in the second frame all succeed. As a result of the queuing packets, the two nodes start another contention window from frame 3. Unfortunately the transmissions in the third frame collide, so nodes double their current contention window to two frames. Node A schedules a retransmission in the fourth frame and node B schedules one in the fifth.

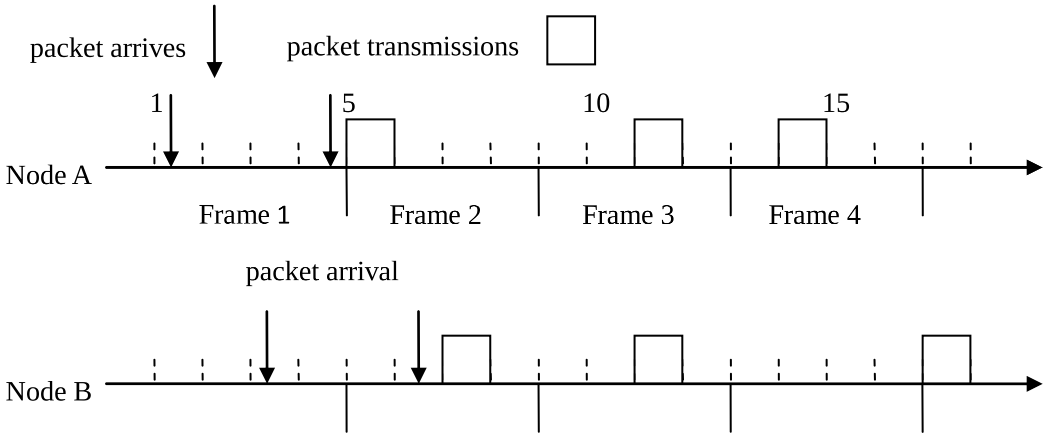


Fig. 9.2 An example of EB-F

V. Proof in 5.3.2 below (5.19)

To prove the diagonal elements of the Jordan form of $\boldsymbol{P}$ are all positive (or the eigenvalues of $\boldsymbol{P}$ are all positive), it needs to be proven that the diagonal elements of $\boldsymbol{P}$ are all larger than 0.5, so that according to the Gershgorin Circle Theorem the eigenvalues will lie within the disc centered between 0.5 and 1 with the radius of less than 0.5, which means the eigenvalues are positive. We need to prove:

$$p_{k,k} = \frac{k}{N}\left(\frac{N-1}{N}\right)^{N-k} + \frac{N-k}{N}\left(1 - \frac{N-k}{N}\left(\frac{N-1}{N}\right)^{N-k-1}\right) > 0.5 \qquad (2)$$

Where N is and integer larger than 2, and k is an integer between 0 and $N-1$. Move 0.5 to the left:

$$\frac{2k}{N}\left(\frac{N-1}{N}\right)^{N-k} + \frac{N-2k}{N} - 2\left(\frac{N-k}{N}\right)^2\left(\frac{N-1}{N}\right)^{N-k-1} > 0 \qquad (3)$$

Multiply $-N^{N-k+1}$on both sides:

$$2(N-k)^2(N-1)^{N-k-1} - 2k(N-1)^{N-k} - (N-2k)N^{N-k} < 0 \qquad (4)$$

We enlarge the first part of (4) by multiplying $\frac{N-1}{N-k}$ assuming $N > k \geq 1$, then we have:

$$LHS(4) \leq 2(N-k)(N-1)^{N-k} - 2k(N-1)^{N-k} - (N-2k)N^{N-k} < 0 \qquad (5)$$

Transform (5) to:

$$(N-2k)(N-1)^{N-k} - (2k-N)(N-1)^{N-k} + (2k-N)N^{N-k} < 0$$

$$(N-2k)(N-1)^{N-k} + (2k-N)(N^{N-k} - (N-1)^{N-k}) < 0$$

$$(2k-N)(N^{N-k}-(N-1)^{N-k}) < (2k-N)(N-1)^{N-k} \tag{6}$$

For the situation that $2k-N<0$, (6) becomes:

$$(N-1)^{N-k}-N^{N-k} < (N-1)^{N-k} \tag{7}$$

Which is obviously true. For the situation that $2k-N>0$, (6) becomes:

$$N^{N-k} < 2(N-1)^{N-k} \tag{8}$$

By using $x=e^{\ln x}$, (8) can be rewritten as:

$$e^{(\mathrm{N-k})\ln\frac{N}{N-1}-\ln 2} < 1 \tag{9}$$

To prove (9) we need to use the inequality that:

$$\frac{x}{1+x} < ln(1+x) < x \tag{10}$$

Where $x>-1$ and $x\neq 0$. (10) can be proved as follow:

Proof of the right side:

$$f(x)=ln(1+x)-x \ \rightarrow f'(x)=\frac{1}{1+x}-1 \tag{11}$$

When $x>0, f'(x)<0$, and $f(x)<0$ because $f(0)=0$. When $x<0, f'(x)>0$ and $f(x)<0$ because $f(0)=0$.

Proof of the left side:

$$f(x)=\frac{x}{1+x}-ln(1+x)\rightarrow f'(x)=\frac{-x}{(1+x)^2} \tag{12}$$

When $x>0, f'(x)<0$, and $f(x)<0$ because $f(0)=0$. When $x<0, f'(x)>0$ and $f(x)<0$ because $f(0)=0$.

Go back to the proof of (9), we assume $x = \frac{1}{N-1}$, then we have :

$$\frac{1}{N} < ln\frac{N}{N-1} < \frac{1}{N-1} \tag{13}$$

And (9) becomes:

$$LHS(66) < e^{\frac{N-k}{N-1}-\ln 2} < 1 \tag{14}$$

By using the minimum value of $k = \frac{N+1}{2}$, (14) becomes:

$$e^{\frac{1}{2}-\ln 2} < 1 \tag{15}$$

And $\ln 2 \approx 0.69$, (15) is proved. So we can say that (2) is true when $k \neq 0 \ or \ \frac{N}{2}$ (when N is even).

For the situation that $k = 0$, (2) becomes:

$$1 - 2\left(\frac{N-1}{N}\right)^{N-1} > 0 \tag{16}$$

$$2\left(\frac{N-1}{N}\right)^{N-1} = e^{\ln 2+(\mathrm{N}-1)\ln\frac{N-1}{N}} < e^{\ln 2+(\mathrm{N}-1)\frac{-1}{N}} < 1 \tag{17}$$

For the situation that $k = \frac{N}{2}$, (2) becomes:

$$\frac{1}{2}\left(\frac{N-1}{N}\right)^{\frac{N}{2}} - \frac{1}{4}\left(\frac{N-1}{N}\right)^{\frac{N}{2}-1} > 0 \tag{18}$$

Multiply both sides of (18) by $4\left(\frac{N-1}{N}\right)^{1-\frac{N}{2}}$:

$$2\left(\frac{N-1}{N}\right) - 1 > 0 \tag{19}$$

Which is obviously true because N is larger than 2.

Glossary

ACK	Acknowledgement
AEA	Adaptive Election Algorithm
AI	Artificial Intelligence
BEB	Binary Exponential Backoff
CA	Collision Avoidance
CAM	Channel Acknowledgment Message
CC-MAC	Correlation-Based Collaborative MAC
CDF	Cumulative Distribution Function
CDMA	Code Division Multiple Access
CRM	Channel Request Message
CSMA	Carrier Sense Multiple Access
CTS	Clear to Send
DARPA	Defence Advanced Research Projects Agency
DFA	Dynamic-Framed-ALOHA
DSMAC	Dynamic Sensor MAC
DSN	Distributed Sensor Network
DW-MAC	Demand Wakeup MAC
EB	Exponential Backoff
EB-F	Framed ALOHA with Exponential Backoff
EB-F-M	EB-F with Maximum Retry
ECN	Explicit Contention Notification
EH	Energy Harvesting
E-MAC	Event-MAC
FDMA	Frequency Division Multiple Access
FFT	Fast Fourier Transform

FRTS	Future Request to Send
FSM	Finite State Machine
GMAC	Global Time Synchronised MAC
G-MAC	Game-Theoretic MAC
HCL	High Contention Level
IEEE	Institute of Electrical and Electronics Engineers
IET	Institute of Electronic Engineers
INS	Iterative Node Selection
IR	Informed Receiving
ISWCS	International Symposium on Wireless Communication Systems
LCL	Low Contention Level
LEACH	Low Energy Adaptive Clustering Hierarchy
MAC	Medium Access Control
MEMS	Micro-Electro-Mechanical System
MFR	Most Forward Routing
MIT	Massachusetts Institute of Technology
MRP	Maximum Rendezvous Period
NAMA	Node Activation Multiple Access
NAV	Network Allocation Vector
NP	Neighbour protocol
N-MAC	Network-MAC
OFDMA	Orthogonal Frequency Division Multiple Access
Q-MAC	Quorum-Based MAC
QoS	Quality of Service
RF	Radio Frequency
RL	Reinforcement Learning
RPs	Rendezvous Periods

RTS	Request to Send
SCH	Scheduling Frames
SDMA	Space Division Multiple Access
SEP	Schedule Exchange Protocol
SMAC	Sensor MAC
SUA	Schedule Unifying Algorithm
SYNC	Synchronisation
TDMA	Time Division Multiple Access
TRAMA	Traffic-Adaptive medium access control
T-MAC	Timeout-MAC
VQ	Vector Quantisation
WSNs	Wireless Sensor Networks
WSS	Wireless Sensor Systems

References

[1]. I. F. Akyildiz, Y. Sankarasubramaniam, E. Cayirci and W. Su, "Wireless sensor networks: A survey", Computer Networks, vol.40, pp. 393-422, 2002.

[2]. I. Demirkol, C. Ersoy and F. Alagoz, "MAC protocols for wireless sensor networks: A survey", Communications Magazine, vol.44, pp. 115-121, 2006.

[3]. L. P. Kaelbling, M. L. Littman and A. W. Moore, “Reinforcement learning: A survey”, Artificial Intelligence Research, vol.4, pp. 237-285, 1996.

[4]. H. Li, D. Grace D and P. D. Mitchell, “Cognitive radio multiple access control for unlicensed and open spectrum with reduced spectrum sensing requirements”, International Symposium on Wireless Communication Systems (ISWCS), pp.1046-1050, 2010.

[5]. C.-Y. Chong and S. P. Kumar, "Sensor networks: Evolution, opportunities and challenges", Proceedings of the IEEE, vol.91, pp. 1247-1256, 2003.

[6]. Ahmed M. Mahdy, "Marine Wireless Sensor Networks: Challenges and Applications", Seventh International Conference on Networking, pp. 530-535, 2008.

[7]. G. Werner-Allen, K. Lorincz, M. Welsh, Omar, M. Ruiz and K. Lees, "Deploying a wireless sensor network on an active volcano", IEEE Internet Computing, vol.10, pp. 18-25, 2006.

[8]. A. Mainwaring, J. Polastre, R. Szewczyk, D. Culler and J. Anderson, "Wireless sensor networks for habitat monitoring", WSNA'02, Atlanta, USA, pp. 88-97, 2002.

[9]. Jelena Skulic and Kin K. Leung, “Application of Network Coding in Wireless Sensor Networks for Bridge Monitoring”, PIMRC 2012, pp 789-795, 2012.

[10]. Mohammad Alex, Simon Carlsen and Stig Petersen, “Applications of Wireless Sensor Networks in the Oil, Gas and Resources Industries”, 2010 International Conference on Advanced Information Networking and Applications, pp 941-948, 2010.

[11]. J. G. T. Anderson. Pilot survey of mid-coast Maine seabird colonies: an evaluation of techniques. Bangor, ME, 1995. Report to the State of Maine Dept. of Inland Fisheries and Wildlife.

[12]. Gao Baolu, Xiong Shibo and Xu Zhanwei, “The Application of Wireless Sensor Networks in Machinery Fault Diagnosis”, 2010 International Conference on Machine Vision and Human-machine Interface, pp 315-318, 2010.

[13]. E. Shih, S. Cho, N. Ickes, R. Min, A. Sinha, A. Wang and A. Chandrakasan, "Physical layer driven protocol and algorithm design for energy-efficient wireless sensor networks", ACM MobiCom'01, Rome, Italy, pp. 272-286, 2001.

[14]. J. M. Kahn, R. H. Katz and K. S. J. Pister, "Next century challenges: Mobile networking for smart dust", ACM MobiCom'99, Washington, USA, pp. 271-278, 1999.

[15]. A.-S. Porret, T. Melly, C. C. Enz and E. A. Vittoz, "A low-power low-voltage transceiver achitecture suitable for wireless distributed sensor network", ISCAS 2000, Geneva, Switzerland, pp. 56-59, 2000.

[16]. A. Perrig, R. Szewczyk, V. Wen, D. Culler and J. D. Tygar, "Spins: Security potocols for sensor networks", ACM MobiCom'01, Rome, Italy, pp. 189-199, 2001.

[17]. W. Ye, J. Heidemann and D. Estrin, "An energy-efficient mac protocol for wireless sensor networks", IEEE INFOCOM, pp. 1567-1576, 2002.

[18]. http://en.wikipedia.org/wiki/List_of_wireless_sensor_nodes.

[19]. http://www.btnode.ethz.ch/

[20]. http://www.archrock.com/technology

[21]. http://www.senceive.com/

[22]. http://envisense.org/glacsweb/tech/probe/probe

[23]. http://en.wikipedia.org/wiki/IMote

[24]. http://en.wikipedia.org/wiki/Iris_Mote

[25]. http://www.xbow.com/

[26]. http://www.csee.ltu.se/~jench/mulle.html

[27]. http://en.wikipedia.org/wiki/NeoMote

[28]. http://www.nanork.org/wiki/FireFly

[29]. Sutton, R. S., and Barto, A. G., “Reinforcement learning: An introduction”, Cambridge, MA: MIT Press, 1998.

[30]. C. Schurgers and M. B. Srivastava, "Energy efficient routing in wireless sensor networks", Proc. MILCOM for Network-Centric Ops.: Creating the Info. Force, Mclean, 2001.

[31]. Datasheet for IRIS nodes available at [Online] http://www.memsic.com/products/wireless-sensor-networks/wireless-modules.

[32]. Datasheet for Micaz nodes available at [Online] http://www.memsic.com/products/wireless-sensor-networks/wireless-modules.

[33]. L. Kleinrock and F. A. Tobagi, "Packet switching in radio channels: Part 1-carrier sense multiple-access modes and their throughput-delay characteristics", IEEE Transactions on Communications, vol.23, pp. 1400-1416, 1975.

[34]. N. Abramson, "The aloha system: Another alternative for computer communications", AFIPS '70 (Fall), pp. 281-285, 1970.

[35]. Czapski, P. P., "A survey: MAC Protocols for Applications of Wireless Sensor Networks", TENCOM 2006, 2006 IEEE Region 10 Conference,

2006.

[36]. Y. Wei, J. Heidemann and D. Estrin, “Medium access control with coordinated adaptive sleeping for wireless sensor networks”, IEEE/ACM Transactions on Networking, vol. 12, no. 3, pp. 493- 506, 2004.

[37]. LAN MAN Standards Committee of the IEEE Computer Society, “Wireless LAN medium access control (MAC) and physical layer (PHY) specification”, IEEE, New York, USA, IEEE Std 802.11-1997 edition, 1997.

[38]. Hayat, S., Javaid, N., Khan, Z. A., Shareef, A., Mahmood, A., Bouk, S.H., "Energy Efficient MAC Protocols", HPCC-ICESS, 2012 IEEE 14th International Conference on, pp. 1185-1192, 2012.

[39]. T. V. Dam and K. Langendoen, "An adaptive energy-efficient mac protocol for wireless sensor networks", SenSys’03, Los Angeles, California, USA, pp. 171-180, 2003.

[40]. P. Lin, C. Qiao and X. Wang, "Medium access control with a dynamic duty cycle for sensor networks", WCNC 2004 / IEEE Communications Society, pp. 1534-1539, 2004.

[41]. Yanjun Sun, Shu Du, Omer G. and David Johnson, “DW-MAC: A Low Latency, Energy Efficient Demand-Wakeup MAC Protocol for Wireless Sensor Networks”, MobiHoc’ 08, pp.53-62, Hong Kong, 2008.

[42]. V. Rajendran, K. Obraczka and J. J. Garcia-Luna-Aceves, "Energy-efficient, collision-free medium access control for wireless sensor networks", Wireless Networks, vol.12, 2006.

[43]. W. Heinzelman, A. Chandrakasan and H. Balakrishnan, "Energy-efficient communication protocol for wireless microsensor networks", Proc. 33rd Hawaii Int’l. Conf. Sys. Sci., Japan, 2000.

[44]. I. Rhee, A. Warrier, M. Aia, J. Min and M. L. Sichitiu, "Z-MAC: A hybrid mac for wireless sensor networks", IEEE/ACM Transactions on

Networking, vol.16, pp. 511-524, 2008.

[45]. I. Rhee, A. Warrier, J. Min and L. Xu, "Drand: Distributed randomized TDMA scheduling for wireless ad hoc networks", in ACM MobiHoc, pp. 190-201, New York, 2006.

[46]. Yongrui Chen, Weidong Yi and Yang Yang, “A Global Time Synchronised MAC for Wireless Sensor Network”, WiCOM’ 08, pp. 1-6, Dalian, 2008.

[47]. Y. Kim, H. Shin and H. Cha, “Y-MAC: An Energy-Efficient Multi-Channel MAC Protocol for Dense Wireless Sensor Networks”, IPSN International Conference, pp. 53-63, St. Louis, 2008.

[48]. Mehmet C. Vuran and Ian F. Akyildiz, “Spatial Correlation-Based Collaborative Medium Access Control in Wireless Sensor Networks”, IEEE Transactions on Networking, vol.14, no.2, 2006.

[49]. Chih-Min Chao and Yi-Wei Lee, “A Quorum-Based Energy-Saving MAC Protocol Design for Wireless Sensor Networks”, IEEE Transactions on Vehicular Technology

[50]. L. Zhao, L. Guo, J. Zhang, and H. Zhang, “Game-theoretic medium access control protocol for wireless sensor networks”, IET Communications, vol.3, pp. 1274-1283, 2009.

[51]. Woonsik Lee, Minh Nugyen, Arabinda Verma and Hwang Soo Lee, “Schedule Unifying Algorithm Extending Network Lifetime in S-MAC-Based Wireless Sensor Networks”, IEEE Transactions on Wireless Communications, vol. 8, pp. 4375-4379, 2009..

[52]. Fabio Iannello, Osvaldo Simeone and Umberto Spagnolini, “Medium Access Control for Wireless Sensor Networks with Energy Harvesting”, IEEE Transactions on Communications, vol. 60, pp. 1381-1389, 2012.

[53]. OPNET Documents, Introduction.

[54]. http://en.wikipedia.org/wiki/MATLAB.

[55]. Kwan-Wu Chin, “PairWise: A Time Hopping Medium Access Control

Protocol for Wireless Sensor Networks”, IEEE Transactions on Consumer Electronics, vol. 55, pp. 1898-1906, 2009.

[56]. http://en.wikipedia.org/wiki/OPNET.

[57]. OPNET Documents, Tutorial.

[58]. OPNET, Process Model Editor.

[59]. OPNET, Node Model Editor.

[60]. OPNET, Project Model Editor.

[61]. OPNET Documents, Wireless.

[62]. OPNET Documents, Tutorial on Transceiver pipeline stages.

[63]. O'Connor, John J., Robertson, Edmund F., "Markov chain", MacTutor History of Mathematics archive, University of St Andrews.

[64]. Norris, James R., Markov chains. Cambridge University Press, 1998.

[65]. S. P. Meyn and R.L. Tweedie, Markov Chains and Stochastic Stability. Second edition, Cambridge University Press, 2008.

[66]. Spitzer, Frank, "Interaction of Markov Processes". Advances in Mathematics 5, 246–290, 1970.

[67]. Usatenko, O. V. Apostolov, S. S. Mayzelis, Z. A. Melnik, S. S., Random finite-valued dynamical systems: additive Markov chain approach. Cambridge Scientific Publisher, 2010.

[68]. N. Abramson, "The ALOHA System - Another Alternative for Computer Communications". 1970 Fall Joint Computer Conference, 1970.

[69]. Frank F. Kuo. "The ALOHA system". ACM Computer Communication Review, 1995.

[70]. R. Binder; N. Abramson, F. Kuo, A. Okinaka, D. Wax, "ALOHA packet broadcasting - A retrospect". 1975 National Computer Conference, 1975

[71]. N. Abramson. "The ALOHAnet – Surfing for Wireless Data" (PDF). IEEE Communications Magazine 47 (12): 21–25, 2009

[72]. Robert M. Metcalfe and David R. Boggs, "Ethernet: Distributed Packet

Switching for Local Computer Networks". Comm. of the ACM 19 (7), 1976.

[73]. Y. C. JENQ, "Correspondence on the stability of slotted aloha system", IEEE Transactions on Communications, vol.28, pp. 1936-1939, 1980.

[74]. Byung-Jae, Nah-Oak Song, "Performance analysis of exponential backoff ", IEEE transactions on networking, vol.13, pp. 343-355, 2005.

[75]. N. AL-KARAKI and A. E. KAMAL, Routing techniques in sensor networks: A survey, IEEE Wireless Communications December, pp. 6-28, 2004.

[76]. IEEE Computer Society LAN MAN Standards Committee. IEEE std 802.11-1999m Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications, 1999.

[77]. D. T. Finkbeiner II, “Introduction to Matrices and Linear Transformations”, Freeman, 1978.

[78]. D. Poole, “Linear Algebra. A Modern Introduction”, Cengage Learning, 2005.

[79]. R. S. Varga, “Geršgorin and His Circles”, Springer-Verlag, 2004.

[80]. Zwillinger, Daniel, Kokoska, Stephen, "CRC Standard Probability and Statistics Tables and Formulae", CRC Press. p. 49, 2010.

[81]. J. Barcelo, B. Bellalta, C. Cano, A. Sfairopoulou, M. Oliver and K. Verma, “Towards a collision-free WLAN: dynamic parameter adjustment in CSMA/E2CA", EURASIP Journal on Wireless Communications and Networking, 2011.

[82]. M. Fang, D. Malone, K. Duffy and D. Leith, “Decentralised learning MACs for collision-free access in WLANs”, Wireless Networks, vol. 19, no. 1, pp. 83-98, 2013.