Practical Evaluation of Low-complexity Medium Access Control Protocols for Wireless Sensor Networks

Selahattin Kosunalp

Ph.D.

University of York

Electronics

March 2015

Abstract

This thesis studies the potential of a novel approach to ensure more efficient and intelligent assignment of capacity through medium access control (MAC) in practical wireless sensor networks (WSNs), whereby Reinforcement Learning (RL) is employed as an intelligent transmission strategy. RL is applied to framed slotted-ALOHA to provide perfect scheduling. The system converges to a steady state of a unique transmission slot assigned per node in single-hop and multi-hop communication if there is sufficient number of slots available in the network, thereby achieving the optimum performance.

The stability of the system against possible changes in the environment and changing channel conditions is studied. A Markov model is provided to represent the learning behaviour, which is also used to predict how the system loses its operation after convergence. Novel schemes are proposed to protect the lifetime of the system when the environment and channel conditions are insufficient to maintain the operation of the system.

Taking real sensor platform architectures into consideration, the practicality of MAC protocols for WSNs must be considered based on hardware limitations/constraints. Therefore, the performance of the schemes developed is demonstrated through extensive simulations and evaluations in various test-beds. Practical evaluations show that RL-based schemes provide a high level of flexibility for hardware implementation.

Contents

List of Figures

List of Tables

Acknowledgements

I would like to thank my supervisors, Dr Paul Mitchell and Dr David Grace, and my thesis advisor Mr Tim Clarke, for their great help and support given throughout the research.

I would like to give my special thanks to the Republic of Turkey Ministry of National Education for its financial support for this project and for giving me the opportunity to study abroad.

I am also grateful to my colleagues in Communications and Signal Processing Group for sharing their knowledge and Ali Bostancioglu, Ozgur Culfa, Yilmaz Kaplan, Oguzhan Bilgin, Mutlu Cukurova, Mehmet Emin Bakir, Tayfun Kok and Muhammet Arucu for creating a friendly and helpful environment.

I would like to dedicate this thesis to my family and my fiancé, Hatice Yalman, for their endless love and support.

Finally, I would like to express my deepest thankfulness to my cousin, Mrs. Arife ORAK, her husband Mr. Mehmet Naci Orak and their lovely kids, Muhammed Orak, Ridvan Orak, Humeyra Orak and Hamza Orak, who has been sharing happiness with me from the moment I arrived in the UK.

Declaration

Some of the research presented in this thesis has resulted in international journals and conference publications. All contributions presented in this thesis are as original to the best knowledge of the author. References and acknowledgements to other researchers have been given as appropriate. A list of the publications is provided as follows:

Journal (SCI)

S. Kosunalp, P.D. Mitchell, D. Grace and T. Clarke, "Experimental Study of the Capture Effect for Medium Access with ALOHA," ***ETRI Journal***, 37(2), pp. 359-368, 2015.

S. Kosunalp, Y. Yan, P.D. Mitchell, D. Grace and T. Clarke, " Practical Implementation and Stability Analysis of ALOHA-Q for Wireless Sensor Networks," ***IEEE Sensors***, (under review)

S. Kosunalp, Yi Chu, P.D. Mitchell, D. Grace and T. Clarke, "Use of Q-Learning Approaches for Practical Medium Access Control in Wireless Sensor Networks," ***International Journal of Distributed Sensor Networks*** , (under review)

Conference

S. Kosunalp, P.D. Mitchell, D. Grace and T. Clarke, "Practical Implementation Issues of Reinforcement Learning Based ALOHA for Wireless Sensor Networks," **International Symposium on *Wireless Communication Systems (ISWCS)***, pp. 360-364, 2013.

1. Introduction

Contents

1.1. Motivation

Medium access control (MAC) protocols are designed to regulate and control the users in accessing a shared medium, affording significant improvement in energy efficiency and channel performance of wireless sensor networks (WSNs). Due to most sensor nodes having limited-energy sources, MAC protocols are specifically required to provide energy-efficient operation in order to significantly prolong the lifetime of WSNs.

In the design of most MAC protocols, one main assumption is that packets which have collided (two or more, partly or fully) are discarded and have to be retransmitted. An effective collision avoidance scheme is required to design an ideal protocol. This thesis shows that collisions do not necessarily result in packet loss and the failure of all the collided packets, which refers us to the capture effect. It is shown that the first-arriving packet in a collision can capture the radio, at the same transmission power level, and may be decoded depending on the amount of overlapping length. Therefore, the performance of the MAC protocols can be enhanced with the benefit of the capture effect.

The key advantage of the ALOHA-based schemes is simplicity and low overheads, but they provide poor throughput capability because of the collisions and retransmissions resulting from their blind transmission technique. Recently, Reinforcement Learning (RL) has been applied to improve the inefficient transmission strategy of ALOHA-based schemes. RL techniques have increased the channel performance significantly in both single-hop and multi-hop communication, whereby an intelligent slot selection strategy is employed. The performance evaluations of the RL-based schemes are however restricted to simulation-based tools where a static environment is considered. This thesis evaluates the performance of the RL-based schemes through real-world test-beds. Also, the behaviours of the schemes against dynamic channel and environment conditions are studied. The main contributions are summarised as follows:

- To the best of our knowledge, this thesis has first studied the capture effect practically for equal power transmissions. A new 3-packet-capture scenario is introduced and tested, extending prior work for realistic systems. In order to add the capture effect to the throughput performance of MAC protocols, a capture coefficient is derived in association with packet length which can be used analytically to predict the impact of capture through simulation and analytical models (see chapter 4).
- Reinforcement learning based ALOHA (RL-ALOHA) scheme potentially solves the problems of collision and poor throughput capability in random access schemes in an intelligent manner. The throughput performance of RL-ALOHA is first practically studied in this thesis through a real-world test-bed. During the practical implementations, numerous limitations and

constraints that impact on the maximum achievable throughput have been observed and novel schemes are proposed to deal with these practical implementation issues (see chapter 5).

- This thesis provides a practical evaluation of ALOHA and Q-Learning based MAC (ALOHA-Q) protocol for the first time. The resilience level of ALOHA-Q to loss of convergence is explored in order to consider the weakness of the scheme in the presence of packet losses in the steady state. The resilience level to loss of convergence is presented according to various packet loss probabilities. A Markov model is derived to estimate the time to loss of convergence for a single user. Then, a novel technique is proposed to protect the convergence lifetime in the presence of packet loss (see chapter 6).
- This thesis also extends the implementation of ALOHA-Q in both simulation and realistic test-beds, to linear-chain, grid and random topologies. Practical implementation issues of ALOHA-Q are studied based upon hardware limitations and constraints. The performance of ALOHA-Q in simulation in comparison to ZMAC is validated with practical results. In order to strengthen the merits of the ALOHA-Q against dynamic the channel and environment conditions, the epsilon-greedy strategy is integrated (see chapter 7).

1.2. Thesis Outline

The chapters comprising this thesis and the contents of the chapters are summarised as follows:

- Chapter 2 Wireless Sensor Networks

This chapter provides a brief overview of sensor networks, physical layer components, operating systems, the capture effect and RL techniques. An introduction to wireless sensor networks, including applications, challenges, constraints and network architecture, is presented. The fundamental structure of the TinyOS operating system and its features are described in details. The capture effect is introduced with example scenarios and existing work summarised.

- Chapter 3 Medium Access Control

A wide range of MAC protocols proposed in the literature and the main four multiple access techniques and their advantages and disadvantages are presented. Basic features and design requirements of the protocols as well as their advantages and drawbacks are summarised. A discussion of the feasibility of MAC protocols for practical implementation and common approaches to performance evaluation are given.

- Chapter 4 Experimental Study of the Capture Effect for Medium Access Control with ALOHA

The main focus of this chapter lies in exploring the capture effect when two or more packets collide. First, the test-bed and its purpose are introduced. The practical results of the scenarios are given. This chapter proposes the capture coefficient to be used in assessing the throughput performance of the MAC protocols. ALOHA networks, pure and slotted are then introduced. The theoretical throughput analysis of pure and slotted Aloha with a finite number of users is analytically derived. Following this, the throughput calculation of pure Aloha with the capture effect is explored using the capture coefficient. The experimental result of pure ALOHA is presented and compared with the theoretical value derived.

- Chapter 5 Practical Implementation Issues of RL-ALOHA for WSNs

A description of RL-ALOHA scheme is presented with the fundamentals of the slot selection strategy. Then, the practical implementation of RL-ALOHA protocol with acknowledgement packet loss caused by the receiver is presented. In order to mitigate the practical issue, a new slot selection strategy and its implementation is provided.

- Chapter 6 Practical Implementation and Stability Analysis of ALOHA-Q for WSNs

A detailed description and practical performance of ALOHA-Q protocol as well as the experimental setup are introduced. Then, a stability analysis of ALOHA-Q with the Markov model derived is provided. The behaviour of ALOHA-Q in the presence of packet loss after convergence is studied. The time taken to lose convergence is estimated using the test-bed and the Markov model. A new punishment modification scheme is proposed to protect the convergence, analysed using a Markov model and implemented on test-beds.

- Chapter 7 ALOHA-Q for Practical Multi-hop WSNs

A brief description of *exploration/exploitation* methods is introduced, resulting in the proposal of two new slot selection strategies using these methods. The performance evaluations of ALOHA-Q in three realistic test-beds scenarios (linear chain, grid and random topology) are presented. Then, ALOHA-Q is experimentally evaluated in response to two real-world events. ALOHA-Q with *exploration/exploitation* strategies is implemented to provide better performance in these two events.

Chapter 8 describes the potential future work to extend the research reported in this thesis. This thesis is summarised in chapter 9, highlighting the novelty and original contributions.

2. Wireless Sensor Networks

Contents

Wireless sensor networks (WSNs) are a class of *ad hoc* networks which can be used in many different areas, ranging from environmental monitoring to industrial, military and health applications [1]. A typical WSN is expected to consist of a potentially large number of inexpensive sensor nodes with the capabilities of sensing, computation and communications, each of which is likely to be battery-powered, small in size and able to communicate over short distances. Recent advances in wireless communications and digital electronics have resulted in the development of inexpensive, low-power, multifunctional sensor nodes. In many cases, a distinctive feature of WSNs is that sensor nodes are randomly deployed in unreachable areas which often make recharging or replacing batteries difficult. A typical WSN needs to be able to self-organize and be robust to environmental changes such as failure of nodes.

Sensor nodes in a WSN usually share and contend for the same communication medium, which may result in a packet transmission failure through multiple concurrent accesses. Medium access control (MAC) protocols have the responsibility of controlling and regulating users in accessing a shared transmission medium. Sensor nodes are usually powered by energy-limited batteries, and it may be impractical to charge the exhausted battery. Therefore, the primary objective is to maximize the lifetime of the network which brings the necessity to design energy-efficient MAC protocols. It is well-known that the communication in a sensor network is the most energy-consuming mechanism [2]. The main sources of energy wastage include packet collisions, idle listening, overhearing and control packet overheads. A well-designed MAC scheme should be able to mitigate the overall energy consumption to an acceptable level, while affording good channel performance.

In the recent past, MAC protocols for WSNs have been broadly explored with significant improvements on performance (throughput, latency and energy-efficiency). Although most proposed protocols have provided significant performance enhancements, the design of the protocols has resulted considerable complexity and overheads. Taking real sensor platform into consideration, simple devices with limited power, memory and wireless communication ability, the practicality of MAC protocols must be considered when designing. Many of the proposed schemes have only been evaluated through simulation which may not reflect the actual performance of the protocols because of the unrealistic assumptions. Therefore, it is important to develop simpler protocols to provide flexibility for practical implementation. ALOHA based schemes have a key benefit of simplicity but suffer from the inefficient transmission strategy. For the purpose of simplicity, Reinforcement learning (RL) strategy has been applied to slotted-ALOHA for developing an effective transmission strategy [65-68]. In this thesis, RL-based ALOHA schemes are practically evaluated in chapters 5 to 7.

2.1. Fundamentals of WSNs

In this section, a brief introduction to WSNs is provided including basic features and components of a sensor node, WSNs application domains, and design challenges is presented.

2.1.1. Sensor Nodes Architecture

The literature typically understands a sensor node to refer to a device sensing a real-world parameter, storing and processing, and transfering it out of network. Each node of a sensor network is composed of four basic sub-units:

- A sensing unit: It is composed of a number of sensors to detect real-world parameters which are produced in analog unit and converted to digital signals for processing in the processor unit.
- A Processor: In general, it performs the data processing and organizes the functionality of other tasks in a sensor node.

- A radio transciever: A transciever unit makes the nodes' connection to the network, typically operating in 4 different modes: (1) transmit, (2) receive, (3) idle, and (4) sleep. Most sensor nodes exhibit a similiar behaviour of power consumption in these 4 modes in which the power consumption is almost equal in idle and receive modes. Therefore, sensor nodes are inherently required to be shut down rather than leaving them in idle mode when they are not transmitting or receiving.
- A finite power source: This is the most critical component of the sensor nodes. Energy consumed for data communication is always much more than sensing and processing. Minimizing the energy consumption is vital.

Fig 2.1 depicts the structure of a typical sensor node platform.

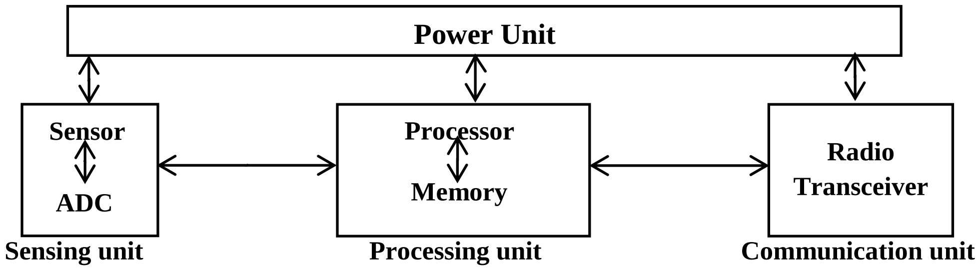


Fig. 2.1 Sensor node architecture

2.1.2. WSN Applications

Since a sensor node can be composed of many different types of sensor, sensor networks can be employed in a variety of applications. Environmental monitoring, military, health and industry applications are a few examples which aim to monitor ambient conditions such as temperature, target detection and air quality.

Environmental monitoring enables the collection of environmental data from various sensor nodes deployed at various sensing locations. For example, WSNs can be deployed in places where forest fires happen very frequently and an estimated origin of a fire can be relayed to the end users before the fire can spread out of control. Therefore, research into the detection and prevention of forest fires is being carried out in a practical way throughout the world [3]. In 2005, researchers deployed a WSN on an active volcano to observe the behaviour of volcanoes as well as to collect volcanic data [4]. They observed 230 volcanic events (eruptions) in 3 weeks through multi-hop communication. In order to assist to monitor air pollution, a WSN-based system has been developed, the Wireless Sensor Network Air Pollution System (WAPMS). In this system, the level of air pollution is produced on a daily or monthly basis. They categorise the various levels of air pollution based on an air quality index [5]. Another WSN-based system has been proposed to control a traffic system which measures the number of vehicles and the vehicle speeds [6].

In recent years, the use of low-cost sensor nodes has been a good approach in battlefield contexts to detect enemy attacks as well as reconnaissance of movements of the enemy. In [7], researchers have proposed a novel algorithm to track moving objects which can move fast, so that they may not be detected along a group of sensor nodes. The coordinates and speed of the objects are estimated to predict the trajectory of the objects. A wide range of military applications of WSNs have been recently surveyed in [8] based on application scenarios and sensor node types.

Recent technologies have enabled the new generation of medical sensor nodes that are capable of sensing and collecting data about people's conditions and behaviour such as physical, psychological and cognitive [9]. WSN-based approaches in healthcare have gained a rising interest due to the availability of the integration of environmental and medical sensors. A lot of sensor nodes can form a Wireless Body/Personal Sensor Network (WBAN) which is critical for early detection of medical conditions [10]. For example, a WSN-based healthcare system is developed to control the patients' health status in a hospital such as controlling a pregnant woman' heart rate and blood pressure [11]. In this system, 4 sensor nodes are placed on a patient to collect the psychological signals. A constant number of relay nodes forwards the information to a base-station. The information is then analysed and presented graphically. This is then sent to the healthcare provider or patient's family in emergency conditions via an SMS using GSM.

Industrial applications have long been based on wired communication which has a cost of installation, maintenance and scalability limitations. In order to reduce these costs, WSNs have been a good approach and can be greatly adopted in industrial environments [12]. A practical WSN-based approach has eliminated the costly installation and maintenance of the communication cables [13]. A transmitter is installed in a Motor to measure the motor energy usage. This is to estimate the motor efficiency.

2.1.3. Protocol Stack for WSNs

A WSN protocol stack is made up of 5 basic architectural layers: *application layer, transport layer, network layer, data link layer and physical layer* [1]. The main tasks of the protocol stack are the combination of power and routing awareness, energy-efficient communication through the wireless medium and integration of data with networking protocols. The application layer specifies the relevant sensing data and tells it to the lower layers, depending on the application types. The transport layer is used to connect WSNs to other networks if it is needed by a specific application. The network layer routes the data through the given directions to the sink nodes and takes care of the data provided by the transport layer. Medium Access Control (MAC) protocols in the data link layer are responsible to manage channel access among neighbouring nodes. The physical layer provides the needs of transmission, reception, modulation and coding techniques. The protocol stack for a WSN is presented in fig. 2.2.

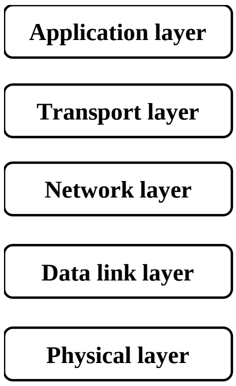


Fig. 2.2 Protocol layers of a WSN

This research focuses mainly on the MAC protocols in the data link layer for developing and implementing low-complexity schemes while achieving high channel throughput and energy efficiency.

2.1.4. Challenges and Constraints

Sensor nodes are equipped with a limited supply of energy. Since sensor nodes can be deployed in inaccessible areas, replenishment of energy sources may not be practical or possible. Node failure can cause significant topological changes and affect the whole sensor network. WSN lifetimes depend mainly on the number of active nodes and connectivity of the network [14]. In order to maximize the network lifetime, the first and often most important design challenge for a WSN is energy efficiency.

Given the inaccessibility of sensor nodes, careful handling of topology maintenance is a challenging task because sensor nodes are prone to frequent failures. This inherent feature brings about the necessity to design robust operation of the networks without external intervention and access. During the deployment phase, sensor nodes may be thrown from a ship, dropped from a plane or placed one by one by a robot. After deployment, new nodes may be deployed in the network depending on the application. The designs should be able to work with varying numbers of nodes. The question of scalability must be taken into consideration when systems are being designed. WSNs should be scalable enough to respond to changes in the network.

The cost of a single sensor node is an important issue in terms of the overall cost of networks because WSNs can include huge numbers of sensor nodes. A well-designed application deployment can fail to commercialize itself if it is not cost-effective. There is a great deal of competition in the commercial market and a WSN designer can expect to find some competitors selling the same sensor network at a cheaper rate. As a result, the cost of each sensor node should be kept at an acceptable low level.

WSNs can collect confidential information across a given area. A typical WSN is unable to resist threats and risks. It is easy for an adversary to compromise a sensor node, disrupt the integrity of the data, eavesdrop on sensor transmission, inject fake messages, and waste network resource. Unlike wired networks, wireless nodes broadcast their messages to the medium. Hence, the issue of security may need to be addressed in WSNs.

2.2. WSNs Programming and Operating Systems

The previous section presented a brief conceptual introduction to wireless sensor networks. To design advanced real-world WSN applications, it is important to understand how sensor network programming plays an important role through the use of custom hardware. Node programming is required to remain creative as a WSN may include hundreds of nodes and it is often impossible to access the nodes deployed to be re-programmed. Therefore, sensor programming should provide a good and reliable system for maintaining the operation and life-time of nodes using the deepest low-power mode. The most significant and potentially used operating systems (OS) for WSNs are TinyOS [15], Contiki [16], Mantis [17] and LiteOS [18]. The choice of operating system for a particular node depends on the adequacy of system support because there are many different sensor nodes being designed. It is worth noting that some versions of an operating system may not support some sensor nodes' requirements. TinyOS-1.x, for example, does not support IRIS nodes but TinyOS-2.x does.

An operating system (OS) supports a programming model which impacts on the application development. OSs for WSNs typically provide two programming models, event-driven and multi-threading model. Multi-threading is an ability of OS to allow multiple threads to occur in a single process. This is not well-suited for resource-constrained devices. Therefore, the current design of OSs for WSNs focuses on developing a light-weight multi-threading model. Event-driven process is more useful because flow of the program is managed by events such as sensor reading output. A WSN OS must provide an efficient memory management strategy for processes and threads. Dynamic memory management provides more flexible process for run-time memory allocation. The correct functioning (reliability) of WSNs after deployment is an important issue for complex application goals as WSNs typically operate unattended for a long time once deployed.

2.2.1. Hardware Devices

There is an increasing trend in producing low-cost and tiny WSN hardware to be effectively disposable. Many of the nodes use RF channels to communicate with other nodes. Most of the RF transceivers operate in the Industrial, Scientific and Medical (ISM) bands due to the free usage of the corresponding spectrum. The characteristics of some existing sensor nodes, the typical features, similarities and notable trends, are illustrated as a list in Table 2.1.

Node name	Microcontroller	Transceiver	Data Memory	External Memory	Operating system
BT node [19]	ATmega 128L	Chipcon CC1000	180K RAM	128KB Flash	TinyOS
EPIC mote [20]	Texas Instruments MSP430	Chipcon CC2420	10KB RAM	48KB Flash	TinyOS
EyesIFX [21]	MSP430F1611	Infineon TDA5250	-	8 Mbit	TinyOS
GWnode [22]	PIC18LF8722	BIM(173 MHz) FSK	64KB Ram	128KB Flash	Custom OS
IMote 2.0 [23]	ARM 11- 400 MHz	Chipcon CC2420	32 MB SRAM	32 MB Flash	TinyOS
IRIS [24]	ATmega 1281	Atmel AT86RF230	8KB RAM	128KB Flash	LiteOS TinyOS
Mica [25]	ATmega 103	RFM TR1000	4KB RAM	512KB Flash	TinyOS
Mica2 [26]	ATmega 128	Chipcon CC1000	4KB RAM	128KB Flash	TinyOS, Mantis
Mica2dot [27]	ATmega 128	Chipcon CC1000	4KB RAM	128KB Flash	TinyOS, Mantis
MicaZ [28]	ATmega 128L	Chipcon CC2420	4KB RAM	128KB Flash	LiteOS TinyOS, Mantis
Mulle [29]	Renesas M16C	Atmel AT86RF230	31KB RAM	384 KB + 4 KB Flash	Contiki, TinyOS
NeoMote [30]	ATmega 128L	Chipcon CC2420	4KB RAM	128KB Flash	TinyOS, Mantis
Shimmer [31]	MSP430F1611	Chipcon CC2420	10KB RAM	2 GB microSD Card	TinyOS
TelosB [32]	Texas Instruments MSP430	Chipcon CC2420	10KB RAM	48KB Flash	Contiki, TinyOS, Mantis
Vesna [33]	ARM Cortex-M3	Chipcon CC1101	100KB RAM	-	Contiki

Table 2.1 Typical features of the existing sensor nodes.

2.2.2. *Operating Systems*

WSNs include severely restricted hardware capabilities in terms of processing power, memory, storage and energy. An operating system should efficiently manage the allocation of the constrained resources to users in a controlled manner. The operating systems for WSNs are inherently less complex than traditional operating systems since WSNs perform a particular application. TinyOS is perhaps the earliest operating system specifically designed for WSNs. The following section focuses on the principles of TinyOS in details as it was the operating system used in this thesis. Then, other popular operating systems for WSNs will be briefly highlighted.

2.2.2.1. TinyOS

TinyOS is an open-source operating system, richly documented and specifically designed for tiny low-power wireless devices. It was conceived by researchers at the University of California (Berkeley), and is aimed at the use of low-cost and low-power operation as a set of components. TinyOS is written in an extension of the C programming language named NesC and its programs are compiled with NesC.

TinyOS provides a large number of independent components that are reusable and linked statically together through their interfaces. Each application specifies the set of components that it uses. In some ways, the components are very similar to objectives in traditional object-oriented programming. A NesC application consists of a number of components that provide and use interfaces. These interfaces have a set of *commands* and *events* that are functions to be implemented by the interface's provider and user. A *command* is basically a demand to a component to operate some service. The *events* are the results of this service, signalling completion of a service.

Additionally, the *events* can be signalled externally, such as reception of a packet. The main interface to transmit a packet is given in next code segment as an example:

```
interface AMSend
{
        command error_t send(am_addr_t addr, message_t* msg, uint8_t len);

        command error_t cancel(message_t* msg);

        command uint8_t maxPayloadLength();

        command void* getPayload(message_t* msg, uint8_t len);

        event void sendDone(message_t* msg, error_t error);
}
```

Fig. 2.3 The interface for packet transmission.

If a component is interfaced with AMSend, the component is able to transmit a packet using the *send* command. After the transmission has been attempted, the *sendDone* event is signalled with a return (error) showing whether the packet was transmitted successfully or not.

In NesC, there are two types of component: *modules* and *configurations*. The modules are the actual implementation of the commands and events where codes are developed. Each module has to declare the interfaces that it provides or uses. Configurations are the place where the components that are going to be used in the application are wired together, connecting the interfaces used by components to interfaces provided by others. Therefore, each application must have a top-level configuration. An example of the top-level configuration is given in fig. 2.4.

```
#include "transmitter.h"
configuration transmitterAppC {}
implementation {
/* Declare used components. */
  components MainC, transmitterC as App, LedsC;
  components new AMSenderC(AM_RADIO_COUNT_MSG);
  components new TimerMilliC() as Timer0;
  components ActiveMessageC;
/* Declare used components. */
/* Wire components together. */
  App.Boot -> MainC.Boot;
  App.AMSend -> AMSenderC;
  App.AMControl -> ActiveMessageC;
  App.Leds -> LedsC;
  App.Timer0 -> Timer0;
  App.Packet -> AMSenderC;
/* Wire components together. */
}
```

Fig. 2.4 The top-level configuration for packet transmission.

TinyOS-2.x ensures a message buffer abstraction called *message_t*. It consists of four fields: header, data, footer and metadata. Each of these is opaque so the source code cannot directly access it. Instead, we use the interfaces to access it as the length and sub-fields of each field may vary depending on the radio platforms. Figure 2.5 depicts the message frame and its fields for AT86RF230 radio transceiver as an example.

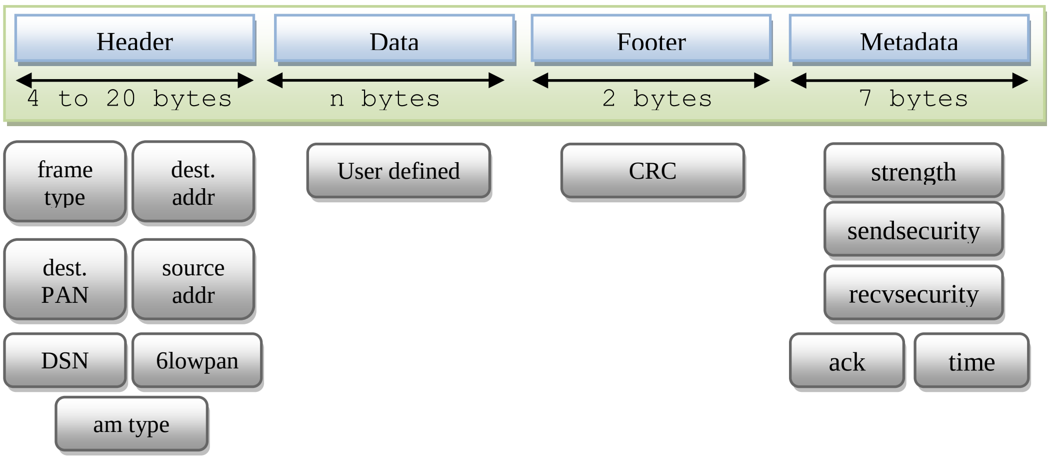


Fig. 2.5 Message frame for IRIS nodes.

- **Header:** The *header* field is an array of bytes which is intended to provide information about the packet such as where the packet is going to or where it was created. It contains a 2-byte frame type, 1 byte sequence number (DSN), 1 byte group id (dest PAN), 2-byte destination address, 2-byte source address, 1 byte network type for 6lowpan and 1 byte message type. The header is 10-bytes long.

- **Data:** Each application can define its data length. The default length of data is small (29-bytes). The maximum value can be 110-byte for a 128-byte long full message because 18 bytes are left for a 6-byte preamble and synchronization, 10-byte header and 2-byte CRC.

- **Footer:** The *footer* field is the cyclic redundancy check (CRC) that indicates corrupted data. It can be automatically generated before transmission and evaluated after reception. It has a length of 2 bytes and is appended in the last 2 bytes of a message.

- **Metadata:** This field is used to store packet information such as RSSI and timestamps. It is not actually transmitted or received. It is intended for internal accounting use only.

2.2.2.2. Others

Contiki is an open-source operating system for tiny networked sensors which is implemented in C language. It has an event-driven model providing multi-tasking facilities to individual processes. In order to minimize the need for memory, Contiki introduces a proto-thread technique where threads are driven by events. It is hard to program as programs have to be implemented as state-machines and long-term computations are performed which may make it unsuitable for all applications.

The Multimodal system for networks of in-situ wireless sensors (MANTIS) is a lightweight and energy-efficient multi-threaded embedded operating system for WSNs. The architecture of MANTIS lies on the traditional layered multithreaded design. The system consists of a scheduler and device drivers. The scheduler provides a pre-defined subset of priority-based thread package. A timer is set to call the scheduler periodically. The microprocessor switches to sleep mode when no event is waiting to be performed. MANTIS has a complex scheduling mechanism which incurs in a memory overhead of a stack per thread.

LiteOS is a Unix-like and multithreaded operating system for WSNs. It provides a number of interfaces to interact users with Unix-like commands. Due to the high complexity and new approaches, other operating systems may require user longer time to effectively learn how to use them. However, LiteOS aims to be an easy-to-use platform thereby reducing the learning time by introducing a more familiar environment, including a hierarchical file system and a wireless shell interface. LiteOS has a larger code size because it employs more complicated scheduling algorithms. It supports only few sensor platforms, MicaZ and IRIS.

TinyOS is used in the all experiments as it has many advantages. TinyOS has a rich library including ready-made components for common tasks, support for a wide range of sensor platforms and documentation to save much programmer work. TinyOS is most used OS in the existing MAC protocol implementations as table 3.1

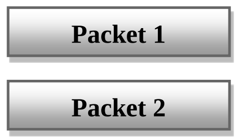
summarizes in the next chapter. Also, it can be seen from the table 2.1 that TinyOS supports more devices than other OSs.

2.3. Capture Effect

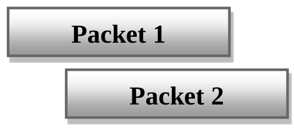
This section presents background information relevant to the work done on *capture effect* in this thesis. The capture effect phenomenon is practically explored. The throughput performance of pure ALOHA scheme is investigated both practically and analytically with the capture effect, presenting significant performance enhancements.

It is commonly assumed in protocol research that when two or more senders start transmitting simultaneously over the same channel, a packet collision occurs at the receiver. Therefore, concurrent transmissions may result in packet loss and the failure of all the collided packets. A collision may happen to different extents either fully or partly, as depicted in Fig. 2.6.

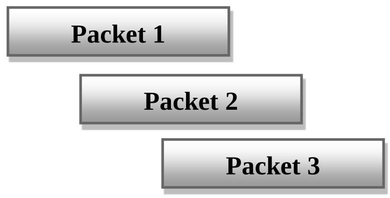
The capture effect is defined as an ability of some radios to correctly receive a packet among simultaneous transmissions. Through the capture effect, packets from the strongest user may be successfully received in the presence of the simultaneous reception (collision) of packets. A strong packet can be successfully decoded if its power is sufficiently larger than the sum of the powers of the other simultaneous packets. In other words, If the *signal-to-interference-plus-noise-ratio* (SINR) of a packet is above a critical threshold, the packet is captured (received).



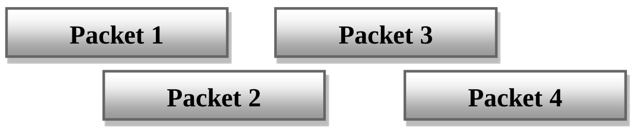
(a) Fully-overlapping



(b) Partly-overlapping



(c) More packets involved in a collision

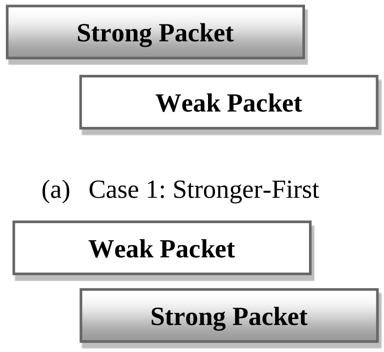


(d) Consecutive packet collisions

Fig. 2.6 Some examples of packet collisions.

2.3.1. Capture Scenarios

Much of the prior work on the capture effect assumes two overlapping packets in one of two situations. In Case 1, the strong packet arrives first and the radio transceiver synchronizes to it. As long as it has sufficient power, it is received normally. In Case 2, the stronger packet arrives later, the radio transceiver synchronizes to the weak packet and reception fails, resulting in both packets being lost because the strong packet corrupts the weak packet. Fig. 2.7 demonstrates the two possible capture cases.



(a) Case 1: Stronger-First

(b) Case 2: Stronger-Last

Fig. 2.7 Common capture cases.

In practice, the capture features of radio transceivers depend on their hardware design and implementation, particularly the modulation and decoding scheme. For example, in stronger-last, it is possible that a radio allows the stronger packet to resynchronize with itself. Therefore, as in stronger-first, the strong packet is received as long as it carries sufficient power. For example, [39] and [42] empirically show that the strong packet can be captured if it arrives later but within the preamble of the first arriving packet. This case is illustrated in Fig. 2.8. Moreover, as in Case 2, it has been shown that the strong packet can be captured regardless of the timing relation with the first arriving packet [43].

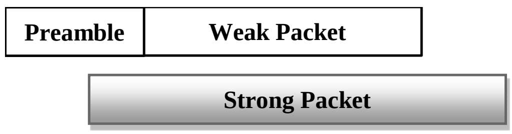


Fig. 2.8 Stronger-last, but within the preamble of the weak packet

2.3.2. Related Work

Theoretical studies of the capture effect have been undertaken to understand packet reception in the presence of interference and to improve the performance of sensor networks. An early study [34] utilised a perfect capture model, in which the packet with the highest received power is always received, thereby achieving dramatic improvements in throughput and delay reduction. Later, researchers [35] proposed a *signal-to-interference-plus-noise-ratio* (SINR)-based capture model. If the SINR of a received packet is above a specified threshold, the received packet is captured.

The capture effect has been thoroughly studied through theoretical work on the throughput of ALOHA networks. Throughput can be measured in Erlangs, where one Erlang corresponds to a single channel in use 100% of the time, or equivalently ten channels in use 10% of the time each. For example, the maximum throughput of Slotted ALOHA network is around 0.36 Erlangs assuming a finite-user network and no capture effect. This implies that 36% of the channel capacity can be used for useful data reception, but it has been markedly increased to nearly 0.53 Erlangs by exploiting the capture effect using two power groups; one transmitting at a relatively high power and the other at a relatively low power [36]. In [37], the maximum throughput of Slotted ALOHA was shown to be increased to 0.65 Erlangs, assuming that one of the colliding packets can survive a collision and be received. In the case of pure ALOHA, the maximum throughput can be increased from 0.18 Erlangs to 0.26 Erlangs by exploiting the capture effect where users randomly select from one of two power levels from a given set [38]. The higher power level is chosen according to a pre-defined power capture threshold of the receiver in which the packets with the higher power can be successfully received. However, this is a theoretical study which did not consider real-world sensor node characteristics.

In the literature, there have been few practical studies [39-41] of the capture effect. Ref. [39] empirically observed the capture effect from real-life measurements using

Prism2 chipset wireless cards with IEEE 802.11b. They showed that the strong packet can be decoded either when it arrives first or last but within the preamble of the first arriving packet. In [40], the packet reception rate for a single sender and interferer is presented; the measurements were conducted using Mica2 nodes equipped with CC1000 radios. They identified a critical threshold value such that if the SINR exceeds the threshold value, there is a high probability that capture will be achieved. Also, the measured threshold value increases with varying numbers of interferers. The capture effect in IEEE 802.15.4 networks is presented in [41], providing a mathematical model using 13192-SARD and 13192-EVB boards equipped with MC13192 radio transceivers. They give the packet capture probability (PCP) versus C/I ratio, the carrier power (C), and the sum of the interfering carrier powers (I), with up to four interferers. They carried out both simulation and test-bed validation for comparison and achieved a very good agreement.

In this thesis, the capture effect is systematically studied through experiments conducted with IRIS nodes equipped with AT86RF230 radio transceiver. The packet reception rate in collisions, using equal power transmissions, is determined based on the degree of packet overlap, up to the packet size. In order to estimate the level of interference in high-density networks, a new capture scenario, namely 3-packet-capture (see section 4.2.4), is introduced and tested, extending prior work for realistic systems.

2.4. Reinforcement Learning

Reinforcement learning (RL), an individual learning approach, is a fully distributed algorithm and works based only on local observations which makes it well-suited to

WSNs. RL, a sub-area of machine learning, is a method by which a learning agent, using reward and punishment, learns from the consequences of its actions by interacting with a dynamic environment through trial-and-error [44]. It allows determination of an optimum transmission strategy from the consequences of a device's action on its environment. In particular, an agent is given a set of actions, and the agent chooses actions according to reward signals received in the form of numerical values. The objective of an agent is to find the best actions that maximize its long-term benefits. Basically, the agents receive reward/punishment after each action in order to influence only on the current action and state pair. A weight value is assigned to each state and action which shows the preference. Upon a reception of reward, the associated weight value is updated.

Reinforcement learning has been potentially studied in many disciplines, such as game theory, probability theory and statistics, genetic algorithms and artificial intelligence. More recently, RL has been considered as one of the best strategy to improve the performance of cognitive radio systems [45-46]. Reinforcement learning has been considered as an efficient and highly suitable approach for MAC protocol design in WSNs in this thesis. It allows users to individually learn based only upon local observations. Since it works on a trial-and-error basis, no environment and radio model are required. Also, it has a certain level of robustness to the environmental changes. This work applies RL as an intelligent transmission strategy, requiring minimal overhead, computation and complexity. The RL techniques are described in details in Chapters 5 to 7.

Discovering solutions by trial and error is very popular in almost all areas of optimization, as well as engineering practice. In the sense of wireless networking, the collective and coordinated behaviour of users is of paramount importance resulting in the self-organizing behaviour of the network. Firefly algorithm is another example solution of providing a collective behaviour within self-organized networks [47]. As in RL method, it is a population-based strategy through trial and error within a search space to find the best solution. Therefore, the firefly algorithm could be applied to design communication protocols in WSN domain but it requires a precise time synchronisation. This is a complex issue in distributed systems and hard to achieve.

2.5. Summary

This chapter has presented a brief introduction to the area of wireless sensor networks. The fundamentals of WSNs including applications, system structure, challenges and constraints are introduced. Then, practical issues of WSN programming are discussed, and the main operating systems designed for WSNs have been highlighted. Packet collisions as one of the major energy consumption reasons in WSNs is described and the capture effect and its scenarios have been presented. This chapter has also introduced the reinforcement learning strategy which is employed as the basis technique of intelligent slot selection in later chapters.

3. Medium Access Control

Contents

This chapter presents a general overview of Medium Access Control (MAC) protocols for WSNs, and introduces an extensive literature review. Basic multiple access techniques are briefly described in order to understand how users share a common medium. The main features of popular and widely used MAC protocols proposed for WSNs are presented.

3.1 Introduction

In a WSN, sensor nodes usually share and contend for the medium, and a packet transmission may fail through multiple concurrent accesses. Medium access control (MAC) protocols are responsible for coordinating user access to a shared medium, affording significant improvement in energy efficiency and channel performance. Due to the unique characteristics and application requirements of WSNs, traditional MAC protocols can be impractical for direct application to WSNs. The primarily goal of such protocols is to provide better Quality of service (QoS) and higher utilisation [1]. During the last two decades, researchers have focused on developing energy-efficient MAC protocols for WSNs. To design a good MAC protocol for WSNs, energy-efficiency is often of paramount importance. Since WSNs may experience continuous change in topology, the scalability and adaptability are other important metrics. Latency, throughput and bandwidth utilization are usually secondary design criterions in WSNs. However, as WSNs have many application scenarios, these secondary attributes can be as important as the primarily ones depending on the intended goals.

A tremendous number of MAC protocols have been proposed for WSNs to deal with various related issues. A typical feature of WSNs is the lack of a centralised controller which makes provision of a perfect coordination a challenge. Accordingly, channel contention and collision could take place because of the simultaneous access from several nodes. Channel sensing can provide collision avoidance, incurred at the expense of higher energy consumption. Reservation-based techniques can prevent most collisions, incurred at the expense of greater overheads. Also, constrained hardware platforms may lead to channel inefficiency due to the lack of infrastructure support. Consequently, a well-designed MAC protocol should comfortably handle these issues and provide a good level of self-organization in order to accommodate network changes.

3.2 Multiple Access Techniques

One of the main problems in multi-user communication systems is the concurrent use of the medium by a large number of users. The capacity of a channel is limited and should therefore be assigned in a sensible way to the users. Multiple access techniques define how the users share a common medium. There are four classic multiple access techniques and these are described next.

3.2.1. Frequency Division Multiple Access (FDMA)

FDMA is the earliest multiple access technique. In this technique, the bandwidth is divided into slices based on frequency so users simultaneously transmit on different frequencies. It is crucial to provide a sufficient guard band between the frequency slices to minimize adjacent channel interference. FDMA is flexible and simple because each user has its own frequency band on which they can start transmission whenever they want. However, the number of frequency slices is constrained. Also, the guard bands are a waste of channel bandwidth. Figure 3.1 illustrates the basis of FDMA.

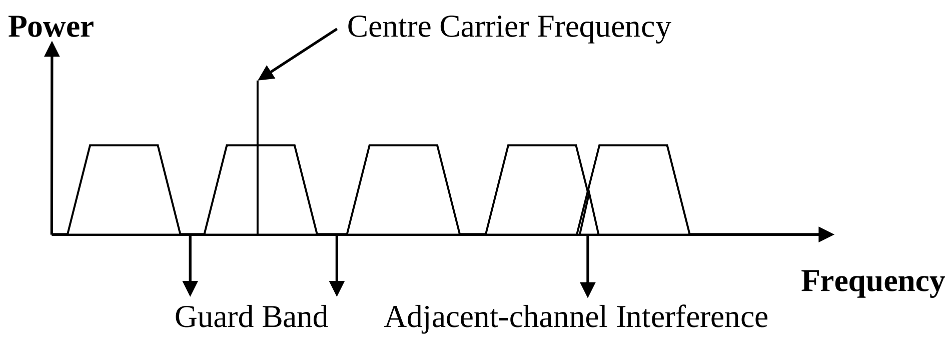


Fig. 3.1 Frequency Division Multiple Access (FDMA)

3.2.2. Time Division Multiple Access (TDMA)

TDMA is a most popular technique particularly suited to digital data that allows the users to transmit on the same frequency, separated in time. TDMA requires accurate time synchronization to ensure that users do not overlap in time. To do this, there is generally a centre station which synchronizes other users' timing and allows new users to enter the network. Also, guard periods are required to make sure that different propagation delays do not cause unintended overlap at a receiver. Figure 3.2 presents the TDMA concept.

.

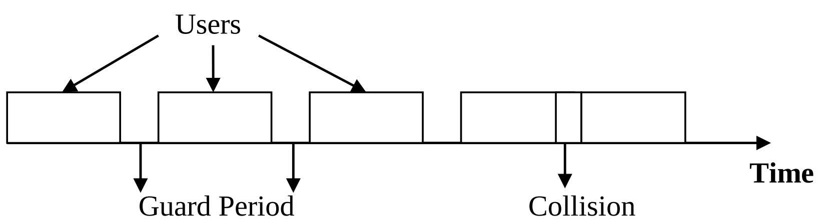


Fig. 3.2 Time Division Multiple Access (TDMA)

3.2.3. Code Division Multiple Access (CDMA)

CDMA allows the users to transmit on the same frequency and at the same time. Each user has a unique spreading code that is multiplied by the user signal to generate a signal for transmission. The receiver uses the same spreading code, multiplied by the incoming signal to reproduce the original signal. All the other decoded signals are considered as background noise because of the orthogonality of the spreading codes. CDMA does not require timing synchronization among the users. CDMA works well when the number of users is low as increasing the number of users will cause higher noise levels at the receiver. CDMA is practically difficult and least used by MAC protocols as it may not be possible to decode spreading signals in the limited sensor platforms.

3.2.4. Random Multiple Access

In this technique, users attempt to transmit in an uncoordinated or minimally coordinated manner. If more than one user tries to access the medium simultaneously, it results in collisions. The collisions are detected by sending out an acknowledgement packet to the source after a packet is correctly received. To avoid collisions, the users listen to the medium to detect possible transmissions from other users in CSMA-based schemes. If the medium is not occupied, the users start transmitting. Random Multiple Access is the simplest technique and easy to implement. However, it suffers from collisions as there is no coordination between the users.

3.3 MAC Protocols for WSNs

Current MAC protocols can be broadly divided into *contention-based* and *schedule-based*. The majority of the proposed schemes are contention-based and inherently distributed, but they introduce energy waste through *overhearing, collisions, idle-listening and re-transmissions*. Schedule-based protocols can alleviate these sources of energy waste by dynamically assigning transmission schedules, but these benefits are incurred at the expense of higher complexity and overheads. On the other hand, there are a number of MAC schemes which combine the features of both contention-based and schedule-based approaches, called *hybrid* protocols. The main drawback of the hybrid schemes is their complexity which may make them suitable for only a limited number of applications. A few Reinforcement Learning (RL) based MAC protocols exist and they are examined as a subsection. In the literature, a large number of MAC protocols, probably hundreds by now, have been developed for WSNs. This thesis surveys the well-established and representative MAC protocols. Taxonomy of the MAC protocols surveyed is illustrated in Fig. 3.3.

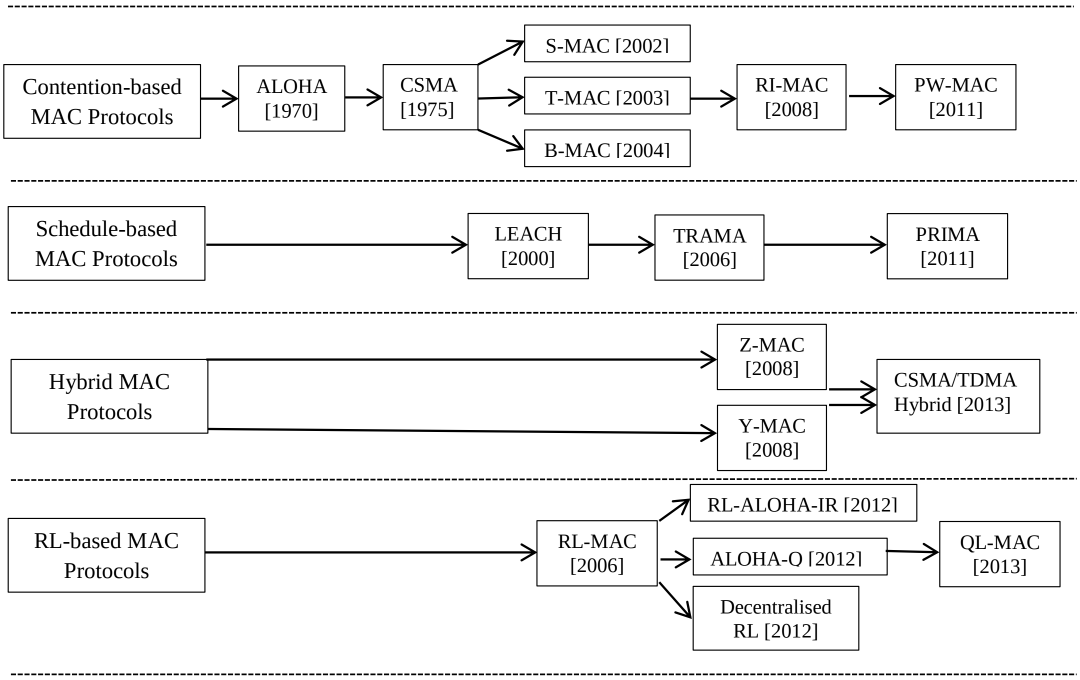


Fig. 3.3 Taxonomy of WSN MAC protocols.

3.3.1. Contention-based MAC Protocols

ALOHA is the earliest contention-based protocol which allows a user to access the medium as soon as it is ready for transmission [48]. Naturally, if more than one user starts transmission simultaneously, packet collisions occur at a receiver. A collision, either partly or fully, results in corruption of all packets involved and collided packets have to be retransmitted. ALOHA is a very simple scheme and easy to implement but not suitable under high channel loads. A variation of ALOHA, called slotted ALOHA, doubles the throughput performance of ALOHA by introducing time slots. Nodes are allowed to transmit their packets at the beginning of the time slots. A time slot accommodates a single packet and a guard period, ensuring a sufficient gap between the slots. Slotted ALOHA requires time synchronization among the nodes.

Carrier Sense Multiple Access (CSMA) is a widely used contention-based scheme with two popular variations, Collision Detection (CSMA/CD) and Collision Avoidance (CSMA/CA) [49]. The main purpose of this approach is to listen to the medium to determine whether anyone else is already transmitting. If the medium is found to be free, the sender initiates transmission. In CSMA/CD, a user keeps listening to the medium while it is transmitting to find whether other users started to transmit simultaneously. If so, the user immediately stops transmitting. CSMA/CD is not easy for hardware implementation as it requires full-duplex communication. Therefore, CSMA/CD is more suitable for wired environments as it is used most notably in local area networking. In CSMA/CA, if a user has sensed that the medium is available, it waits for a random back-off time to alleviate the possibility of collision with other users before it transmits. To avoid collisions, a four-way handshaking technique (RTS/CTS/DATA/ACK) is used. Before sending a packet, a user sends out a special RTS short frame to make a connection with the receiver and reserve the transmission medium. Upon the receipt of the RTS, the receiver acknowledges by sending a CTS packet. Then, the normal data transmission (DATA) begins, followed by the immediate response of an acknowledgement (ACK). The drawback of CSMA/CA is the collision of concurrently-sent RTS frames.

Sensor MAC (S-MAC), perhaps the most studied MAC scheme, is a well-known RTS-CTS based MAC protocol that introduced the concept of a duty-cycle in which the nodes in the network periodically sleep, wake up, listen for potential transmissions, and return to sleep [50]. A large number of existing protocols have used S-MAC as a basis to further schedule the nodes' sleep and active periods as the duty-cycle period in S-MAC is of a fixed duration. The nodes form virtual clusters to determine a common schedule among the neighbouring nodes. A small SYNC packet is exchanged among neighbours at the beginning of each active period to ensure that they wake up concurrently to reduce the control overheads. After the SYNC exchange, the data packets can be transmitted using the RTS-CTS mechanism until the end of the active period. S-MAC adopts a *message passing* technique to reduce latency. A long message is fragmented into many small fragments which are sent in bursts. The main drawback of S-MAC is the predefined constant listen-sleep periods which can cause unnecessary energy consumption, particularly when some nodes such those are located near the sink, may require a higher duty-cycle in order to serve the traffic passing through them.

Timeout MAC (T-MAC) [51] extends S-MAC by introducing an adaptive duty-cycle to lower energy consumption while maintaining reasonable throughput. Packets are sent in bursts of variable lengths and the transmitting nodes switch to a sleep mode between the bursts. The active period is dynamically ended if there is nothing heard and has no data to transmit after a timeout period TA and its duration is longer than the sum of RTS packet and propagation time. The TA is used to assign the minimum amount of idle listening in a duty-cycle. The nodes wake up, as in S-MAC, at the beginning of each active period and listen to the medium to sense any activity and go back to sleep mode if no activation event occurs. T-MAC allows the nodes to remain awake after completion of a packet transmission or reception to observe potentially incoming traffic. T-MAC introduces future request-to-send (FRTS) to solve the early sleeping problem which means that if a node loses the contention, the destination will switch to the sleep mode. By using FRTS packets, the intended destination is informed of the future packet reception time so the destination will wake up at the appropriate time. Compared with S-MAC, T-MAC has better performance in terms of energy efficiency and similar delay performance.

Berkeley Media Access Protocol (B-MAC) is a CSMA-based carrier sense protocol that aims to provide low-power processing, effective collision avoidance, high channel utilization, small code size and RAM usage, and scalability to large number of nodes [52]. B-MAC uses an adaptive preamble sampling scheme to achieve low power operation by reducing the duty cycle and minimizing idle listening. A source node transmits a preamble which is long enough to enable the destination to sense the channel activities. Once an activity has been sensed on the channel, the destination remains awake to receive the incoming message, then it will go back to sleep. When a node wakes up, it expects to detect any preamble. If preamble is sensed, the node waits for the completion of preamble and if the data packet is destined to the node itself, it receives the full packet, otherwise it goes to sleep. However, all the nodes in the range of a sender wake up and wait for packet (overhearing). B-MAC provides bi-directional interfaces to re-configure its functionalities such as acknowledgements, CCA and backoff. These functionalities can be enabled or disabled depending on the application, resulting in changes in both throughput and energy consumption. Therefore, B-MAC is a link layer protocol and any protocol can be built on the top of B-MAC.

Receiver-Initiated MAC (RI-MAC) introduces receiver-initiated data transmissions [53]. RI-MAC tries to minimize the time required for data transmission between a sender and its intended receiver. In this scheme, each node broadcasts its wake-up time with a short *beacon*. A beacon message can also be used to acknowledge the received packets, and to invite new packet transmissions. A sender can only send data when it hears a beacon from the receiver, otherwise it has to wait. When a packet is ready for transmission, waiting for a beacon may require a long duty cycle until the receiver wakes up. RI-MAC is an efficient scheme for unicast traffic, but it is inefficient for broadcast messages.

Predictive-Wakeup MAC (PW-MAC) is a receiver-initiated MAC protocol based on asynchronous duty-cycling [54]. It reduces the energy consumption by allowing senders to accurately predict the wakeup times of receivers. In this way, a sender will only wake up slightly before the intended receiver. Therefore, PW-MAC decreases the duty cycle for both senders and receivers. It introduces an on-demand prediction method in which each node is required to compute its wakeup times using an independent pseudo-random wakeup-schedule generator instead of a fixed or random schedule. This is to prevent neighbouring nodes to concurrently wake up (collision avoidance). PW-MAC uses the similar beacon packets as used in RI-MAC to exchange the parameters for pseudo-random number generation among neighbouring nodes. This allows each node to properly calculate the wakeup times of any neighbouring node. When a node wakes up, it has to send a beacon to broadcast that it is up for transmission. Use of beacon packets introduces high overhead which worsens at higher network densities.

3.3.2. Schedule-based MAC Protocols

The traffic-adaptive medium access protocol (TRAMA), a TDMA-based collision-free algorithm, has been proposed for increasing channel utilization and energy-efficient free channel access [55]. It uses a distributed election algorithm to avoid collisions in a 2-hop neighbourhood (hidden terminal problem). TRAMA introduces low-power operation to mitigate the energy consumption by allowing nodes to go to sleep mode if they have no packet to send or are not intended receiver in a particular time slot. It employs an adaptive scheme to arrange listen/sleep schedule based on the current traffic level.

The nodes arrange common schedules by exchanging their 2-hop neighbourhood information and transmission schedules. To do this, TRAMA includes three components: (1) the Neighbour Protocol (NP) for 2-hop neighbourhood information, (2) the Schedule Exchange Protocol (SEP) for transmission schedule information and (3) the Adaptive Election Algorithm (AEA) for the selection of transmitters and receivers in a current time slot through the information obtained from NP and SEP.

In TRAMA, the time is divided into single slots where each slot has random and scheduled periods for both data and signalling transmissions. Here, random slots are used for signalling and schedule periods are used for data transmission. NP uses signalling slots to broadcast 1-hop neighbourhood information among the neighbouring nodes in order to extract 2-hop information. SEP uses transmission slots to exchange schedules among nodes. The overhead associated with neighbourhood extraction and schedule exchange is a significant burden. Random periods occupy at least 12.5% of channel capacity.

Low Energy Adaptive Clustering Hierarchy (LEACH) is a self-organizing, adaptive clustering-based MAC and routing protocol which achieves energy savings in sensor networks [56]. The concept of LEACH is to divide nodes into local clusters in which one node is nominated as the clusterhead in each cluster. LEACH has a set up phase in which the nodes decide to select with which node will be nominated as clusterhead. The clusterhead is responsible for coordinating the cluster and forwarding the data from nodes in its cluster to the sink (base-station). To equalize the energy dissipation among the nodes, the role of clusterhead node is randomly rotated among the nodes within a cluster based on the amount of energy left in the current clusterhead. LEACH assumes that each node in a cluster can directly communicate with the clusterhead. The nodes in a cluster transmit their data using a TDMA schedule created by the clusterhead. Hence, the nodes can switch the radio off when they are not scheduled to transmit or receive thus saving energy consumption in these nodes. In each cluster, a different CDMA code is used for transmission to avoid interference with a nearby cluster (inter-cluster).

LEACH is highly complex as each node is given a probability of being clusterhead and each clusterhead is required to directly connect the base-station. Each node can be elected the clusterhead based on a probability that is uniformly distributed. Therefore, nodes that are located on one side of the network can appear as a clusterhead, resulting in some nodes on the other side of the network having no clusterhead in their communication range. Since, sensor nodes have limited resources, generating different CDMA codes for clusters might be impractical.

Priority-Based MAC (PRIMA) is an energy-efficient MAC protocol that combines the advantages of TDMA and CSMA [57]. It uses CSMA for control packet transmission while data packets are sent in TDMA slots. PRIMA consists of two components; a clustering algorithm as in LEACH and a channel access mechanism. Time is divided into rounds and a clusterhead is selected in each round. The clusterheads selected broadcast a message to invite the other nodes to choose a cluster. Then, each node joins a clusterhead that is reached by using the minimum energy. A channel access mechanism is performed through 2 steps; Classifier MAC (C-MAC) and Channel Access MAC (CA-MAC). In C-MAC, data packets are appended 2 extra bits in order to show the degree of priority level of the packets by the application layer. CA-MAC uses scheduled slots (TDMA) for data transmission and random slots for periodic control packets (CSMA). As in LEACH, if a clusterhead dies, the nodes in the associated cluster will be totally useless until a new round of clusterhead selection takes place and control packets adds extra overhead.

3.3.3. Hybrid MAC Protocols

Zebra MAC (Z-MAC) [58] is a hybrid protocol that combines the advantages of TDMA and CSMA. It uses a CSMA scheme at low traffic loads and a TDMA scheme at higher traffic loads. It has a preliminary set-up phase if there is neighbour discovery. Neighbour discovery is performed by sending ping packets to one-hop neighbours. Each node generates a list of two-hop neighbours. Using the two-hop neighbourhood, Z-MAC applies a distributed slot assignment algorithm to make sure that any two nodes in the two-hop neighbourhood are not given the same slot, thereby reducing the potential for collision between two-hop neighbours [59]. In Z-MAC, each user has its own slot but if the user does not have any data to transmit, other users may borrow the slot through competition (CSMA).

Z-MAC considers nodes in 2 modes; Low Contention Level (LCL) and High Contention Level (HCL). In LCL, nodes are allowed to contend for any transmission slot while only the owner nodes of slots and their one-hop neighbours are allowed to contend in HCL. Explicit Contention Notification (ECN) packets inform all nodes within 2-hop neighbourhood not to send in associated slot. When a node receives ECN packet, it switches to HCL mode. Compared to CSMA, Z-MAC has a similar performance under low traffic levels and has a better performance with an increasing contention level.

Y-MAC is a TDMA-based energy-efficient multi-channel MAC protocol designed to decrease latency [60]. It divides time into frames consisting of a broadcast and unicast period with each of which containing a number of slots. In the broadcast period, neighbouring nodes exchange the neighbourhood information in order to allocate a slot for each user which avoids collisions and the hidden terminal problem. If some nodes do not hear from a neighbour within a pre-defined certain time, this neighbour is removed from the neighbourhood list and its slot becomes available. At the beginning of each slot, if multiple senders are present, the successive packets are sent on different available slots according to a pre-determined hopping sequence (frequency hopping). Several available frequencies are assigned to each node and one of them is called the basic frequency in which control packet and data transmission normally occurs on the basic frequency. However, if more than one packet is destined to the same destination, the transmitting node can hop to another available frequency and then contend for it. In this scheme, each node is guaranteed to receive at least one packet on the common channel.

A hybrid MAC protocol based on IEEE 802.15.4 standard [61] is proposed to reduce energy consumption and increase channel throughput which combines the benefits of CSMA/CA and TDMA methods [62]. The basic idea of this protocol is to incorporate a dynamic TDMA scheme into the contention access period of 802.15.4 period. Assignment of the TDMA slots is handled by a coordinator which is also responsible for adaptively arranging the contention access period between CSMA/CA and TDMA with respect to the nodes' data queue and the level of collisions on the network. Therefore, coordination of nodes is overcome by the coordinator through sending a beacon frame at specific time intervals. To distinguish the border between TDMA and CSMA, channel utilization level and the number of packets waiting in nodes' queues are considered. This scheme is shown to perform better than 802.15.4 at high channel loads in terms of throughput, energy and delay performance.

3.3.4. RL-based MAC Protocols

Reinforcement learning (RL) has been recently applied to design new MAC protocols in WSNs. Many developed RL-based schemes aim to adaptively adjust the duty cycle of the nodes which is best illustrated by S-MAC. These protocols provide the nodes with an intelligent way of predicting other nodes' wake up times based upon the transmission history of the network. RL-based protocols significantly reduce the energy consumption due to both idle listening and overhearing in the context of duty cycling.

The reinforcement learning-based MAC protocol (RL-MAC) is inspired by S-MAC to determine the duty cycle of nodes in an efficient way [63]. It adaptively adjusts the sleeping schedule based on local and neighbouring observations. For local observation, each node records the number of successfully transmitted and received packets to be a part of the determination of a duty cycle. As for neighbouring observations, the number of failed attempts is added to the header to inform the receiver, which saves energy by minimizing the number of missed packets (early sleeping). The key property of this scheme is that the nodes can infer the state of other nodes using a Markov Decision Process (MDP). RL-MAC outperforms S-MAC and T-MAC in both channel throughput and energy efficiency at high channel loads.

A decentralised RL approach is proposed in [64] to schedule the duty-cycle of sensor nodes based only on local observations. The main point of this study is to shift active periods in time based on transmission paths and ranges along a route. Similar to that of S-MAC, the wake-up periods of the nodes which need to communicate with each other are synchronised, whereas the schedules of nodes on neighbouring branches are desynchronised to avoid interference and packet losses. The active periods are further divided into time slots and the nodes are allowed to wake for a predefined number of slots in each period. The slots where a node will wake up are decided by the learning process. Each slot within a frame is given a quality value which shows the efficiency of the slots. The quality values of slots are updated by rewarding successful transmissions and penalising the negative interactions. As a result of the learning process, the quality values of one group of slots become strictly higher than the rest.

Another similar approach combining the Slotted-ALOHA and Q-Learning algorithms achieves high channel utilisation while mitigating energy consumption, namely ALOHA-Q [65]. Each node repeats a frame structure which includes a certain number of time slots. The packets are transmitted in these time slots. Each node stores a Q-value (equivalent to the quality value above) for each time slot within a frame. The Q-value of a slot is updated individually when a transmission happens and this is used to explore each slot more frequently. Successful transmissions are denoted by small acknowledgement packets which are immediately sent upon the correct reception of the packets. The nodes generate a positive outcome (reward) when the acknowledgement packet is received; otherwise a punishment is applied to update the Q-value. Consequently, the slots with higher Q-values are preferred for data communication and this behaviour repeats the same actions. This process continually returns rewards which serve to decrease the probability of unsuccessful transmission. Eventually, the learning process leads the network to a steady state condition where unique slots are assigned to the nodes. Although this approach results in what appears to be schedule-based access, it does not require any schedule information exchange. However, it is critical to set a sufficient number of slots within a frame. Redundant slots in a frame will result in achieving lower channel throughput. None of the work described above has addressed optimum frame size assignment problem. Instead, a distributed frame size selection algorithm is presented for ALOHA-Q in a single-hop scenario [66]. Furthermore, the selection of the frame size has been discussed for a multi-hop wireless sensor networking [67].

In [68], Reinforcement Learning based ALOHA with Informed Receiving (RL-ALOHA-IR) is proposed which uses the same concept as in ALOHA-Q. RL-ALOHA-IR implements a simple reinforcement learning technique where each slot has a weight value (equivalent to the Q-value above) which is updated with a reward of +1 for successful transmission and a punishment of -1 for failure. ALOHA-Q and RL-ALOHA-IR are the baseline protocols in this thesis.

QL-MAC, Q-Learning-based MAC, is proposed to deal with the issue of finding an efficient way of scheduling the duty-cycle of sensor nodes [69]. Each node considers its own traffic load and the network load of its neighbours. The basic underlying concept of QL-MAC resembles that of a decentralised RL approach, whereby time is divided into time slots (*frame*s) which are further divided into smaller time units (slots). Every node using the Q-Learning algorithm individually limits the total number of slots in which the node wakes up. The frame length and the number of slots constituting the frame remain constant.

3.4 Practicality of the MAC Protocols

Most of the MAC protocols in WSN domain are only evaluated through simulation tools, where practical considerations of unrealistic assumptions can be avoided. Various assumptions either realistic or relatively unrealistic are made to ensure that the design of a MAC protocol is simple and to avoid complexity. It is well-understood that simulation tools allow researchers to make assumptions easily. However, these assumptions, such as the capture effect, the use of two signalling channels, processing power and memory, full-duplex communication and even the use of short control packets, cannot be met on real sensor hardware due to resource limitations. Therefore, implementation and evaluation of a MAC protocol on a real test-bed may result in unexpected performance results compared with simulation results. It may not be possible to re-configure the properties of sensor platforms such as transmission rate, length and structure of a packet, or modulation type, as the design of hardware architectures is fixed. For realism and reliability, future MAC protocols will have to take the hardware specifications of sensor platforms into consideration and will have to be practically implemented and evaluated before being proposed.

Fortunately, implementation of some of the protocols described above is evaluated on test-bed environments. Table 3.1 presents the type of performance evaluation (simulation or test-bed) and the sensor node used in the experiments. Existing performance evaluation in simulation centre around the OMNeT++ [70] and ns-2 [71] simulators as free software. These simulation tools are object-oriented, discrete event-driven simulators which use two programming languages, C++ and OTcl.

Protocol	Simulation	Test-bed Evaluation	Node
Aloha	Analytical	-	-
CSMA	Analytical	-	-
S-MAC	-	Yes	Rene motes in TinyOS
T-MAC	OMNeT++	-	-
B-MAC	-	Yes	Mica2 in TinyOS
RI-MAC	ns-2	Yes	Micaz in TinyOS
PW-MAC	-	Yes	Micaz in TinyOS
TRAMA	Qualnet network Simulator [72]	-	-
Leach	Matlab [73]	-	-
PRIMA	OMNeT++	-	-
Z-MAC	ns-2	Yes	Mica2 in TinyOS
Y-MAC	-	Yes	TmoteSky in RETOS OS [74]
Hybrid CSMA/TDMA	OMNeT++	-	-
RL-MAC	ns-2	-	-
RL-ALOHA-IR	Opnet	-	-
Decentralised RL	OMNeT++	-	-
ALOHA-Q	Opnet [75]	-	-
Q-MAC	OMNeT++	-	-

Table 3.1 Protocols and their implementation type

3.5 Summary

This chapter has presented an overview of the MAC protocols, introducing some of the significant protocols to give an insight. Existing MAC protocols are traditionally classified as contention-based and schedule-based on the basis of channel access mechanisms. Contention-based protocols are inherently distributed and provide high channel utilisation and low energy consumption under low traffic levels. The performance of contention-based protocols degrades with increasing channel traffic load due to the frequent of collisions, unnecessary usage of the channel (idle-listening) and overhearing. S-MAC is a representative contention-based MAC protocol and a large number of protocols are designed based on the S-MAC.

Schedule-based protocols are appealing due to their collision avoidance through assignment of transmission schedules. Nodes exchange information about their neighbourhood to create collision-free schedules. However, these schemes introduce significant complexity and overheads at particularly low traffic loads. LEACH and TRAMA are the representative schedule-based protocols which provide no collisions after the schedules are created as a result of 2-hop neighbourhood information exchange among the nodes. Hybrid protocols have the benefits of both contention-based and schedule-based approaches but high complexity and overheads are still problem. Z-MAC is a practical hybrid protocol that takes the advantages of TDMA at high traffic levels and CSMA at low traffic levels.

Computationally complex algorithms may render the MAC protocols infeasible. It is concluded that low complexity and overheads are important for many practical deployments owing to the limitations and constraints of low-cost, simple sensor devices. ALOHA-Q therefore represents a good example of simplicity while providing perfect scheduling in an intelligent way with minimal additional overheads. It only has an initial poorer performance phase based on a Q learning algorithm in which each node learns to explore a unique transmission slot which will be dedicated to the node at the end of the phase. Eventually, all nodes will find a unique transmission slot and keep transmitting in this slot, if there is sufficient number of slots.

4. Experimental Study of the Capture Effect for Medium Access Control with ALOHA

Contents

This chapter provides a practical study of the capture effect through experiments conducted with IRIS nodes. ALOHA schemes, pure and Slotted ALOHA, considering a finite number of users are then introduced and the performance of the ALOHA schemes is evaluated in analytical form. The impact of the capture on the performance of pure ALOHA is then presented through both an analytical model and practical implementation.

4.1 Introduction

Medium access control (MAC) protocols for Wireless Sensor Networks (WSNs) are designed to regulate and control user access to a shared medium. In the design of most MAC protocols, a prominent assumption about the integrity of the received packets is that they will be corrupted and will not be correctly received when there is concurrent reception. One main function of the MAC protocol is to detect, resolve and avoid collisions. Collision avoidance is a crucial task in wireless sensor networks, since all packets lost in a collision have to be discarded and retransmitted. An effective collision avoidance scheme is a prerequisite for an ideal protocol, since a good MAC scheme will be able to sense a clear channel before attempting a transmission. In a poor scheme, collisions cause a waste of channel resources, with the consequence of low throughput and high delay. It is also important to develop simple schemes to support more basic devices. High complexity of the schemes may bring additional overheads and make the practicality of the schemes difficult.

It is commonly assumed in practical sensor platforms that packets from the strongest user, through the capture effect, may be successfully received in the presence of the simultaneous reception (collision) of packets. WSNs can benefit from the capture effect which would enhance the throughput performance of the MAC protocols. This thesis explores this phenomenon practically. A capture coefficient, the packet reception rate in collisions, is derived based on packet length. This enables the capture effect to be incorporated into the performance evaluation of wireless sensor networks through simulation and analytical models. The capture coefficient is determined in terms of the degree of packet overlap and probability of any value of overlap occurring, up to the packet size. It will be shown that the impact of capture is dependent on packet length because the capture effect does not occur beyond a certain overlap length. The capture coefficient is then adapted so it can be used numerically to generate further results and predict the impact of capture on larger networks.

ALOHA schemes are a good example of simple MAC protocols which are appropriate for lightly loaded networks but suffer from the drawbacks of employing a blind transmission strategy. ALOHA-based techniques are important for certain categories of Wireless Personal Networks (WPNs) [76] and WSNs such as those based on Radio Frequency Identification (RFID) [77] systems which have limited memory and power capabilities. One of the main assumptions in modelling these schemes is that the colliding packets (either fully or partly) will be lost. A packet transmission can be started at any time so packet overlap can occur to different extents. The capture coefficient is applied to pure ALOHA as a case study. Using analytical and practical implementations of the capture effect on ALOHA, a very good match in channel throughput performance enhancement is demonstrated over the non-capture effect case.

4.2 Exploring the Capture Effect

4.2.1. Experimental Setup

IRIS nodes are the platform used, manufactured by Crossbow Technology, which are IEEE 802.15.4-compliant devices operating in the 2.4 GHz (ISM) band and are used specifically for making up low-power wireless sensor networks. They feature an

ATmega1281 low-power microcontroller and an AT86RF230 radio transceiver [78]. TinyOS-2.0 provides software support for the design specifications of IRIS nodes. The AT86RF230 has an internal 128-byte Frame Buffer which is shared between transmission and reception and it can only keep track of one TX or RX frame at a time (half-duplex communication). The modulation scheme is offset quadrature phase-shift keying (O-QPSK). The IEEE 802.15.4 standard employs clear channel assessment (CCA) and Back-Off at the physical layer to ascertain whether or not the medium is occupied. In this study, CCA and Back-off were disabled to allow concurrent transmissions and to enable ALOHA to be implemented.

The complete packet structure provided by IEEE 802.15.4 is shown in Fig. 4.1. The length of a packet can be varied up to 127-bytes by adjusting the payload length. The length of the header depends on the specific radio platforms. A 2-byte CRC follows the last field in the packet format and is automatically generated by the hardware. These three fields form MAC Protocol Data Unit (MPDU). The MPDU is automatically prefixed with a preamble and start of frame delimiter by the radio and frame length by the microcontroller when transmitting a packet.

Bytes:	4	1	1	10	n	2
	Preamble	Start of frame delimiter	Frame Length	Header	Payload	CRC
	Synchronisation Header (SHR)		PHY Header (PHR)	MAC Header (MHR)	MAC Payload (PAYLOAD)	MAC Footer (MFR)

Fig. 4.1 IEEE 802-15.4-compliant packet format used in all the experiments

Successful reception of a packet is achieved based on two stages at the radio transceiver: valid preamble detection for synchronization and the CRC check for accuracy of the packets which includes the header and data payload fields. When the radio detects the preamble of a packet, it synchronizes and locks onto this packet. If the data has not been corrupted, it is passed to the CRC check stage. In terms of the two capture scenarios implemented, the experiments conducted with IRIS nodes confirm that reception of a packet in a collision condition is only successful in the case where a stronger packet arrives first (Case 1). It will be shown that the first-arriving packet in a collision can capture the radio channel, for equal power transmissions, and may be decoded depending on the amount of overlap.

All the experiments were conducted in an indoor, closed-office environment with line of sight communication where the surrounding objects were stationary. All the nodes were in range of each other and were deployed at the same distance (15cm) from the receiver, transmitting at the same power level. A view of the test-bed is shown in Fig. 4.2. The purpose of deploying the nodes at a short-range from the receiver was to maximize the signal-to-noise-ratio (SNR) of the received packets at the receiver and to minimize multipath effects. A computer linked to the base-station was used to observe data-exchange.

Fig. 4.2 Capture effect application environment, three senders, one receiver in the middle and a base-station.

4.2.2. Calculating the Capture Coefficient

In most existing simulation and analytical models for throughput calculation, researchers would make the assumption that all packets involved in a collision can be completely lost. However, this is not a realistic assumption. Therefore, if the probability of collision can be analytically expressed, depending on the system, the effect of capture can be simply incorporated into the overall system performance (throughput). If the probability of collision is known and the packet reception rate under collision conditions is also known, the capture effect can be estimated theoretically and incorporated into the prediction of the throughput performance by multiplying the probability of collision by the packet reception rate in collisions. This is referred to as the capture coefficient in association with a packet length.

To derive the capture coefficient precisely, we assume that the number of overlapping bytes in collisions is uniformly distributed. This means that the probability of having any number of bytes of overlap up to the packet length is equal. C_p is defined as the array of capture probabilities for each overlap length. The individual contribution of each overlap amount to the capture coefficient is given as:

$$\frac{C_p[k]}{L} \tag{4.1}$$

Where k denotes the number of bytes of overlap and L is the length of a packet in bytes. Therefore, the capture coefficient (C) is the sum of all of the individual contributions and is given by:

$$C = \frac{C_p[1]}{L} + \frac{C_p[2]}{L} + \frac{C_p[3]}{L} + \cdots + \frac{C_p[L]}{L} \tag{4.2}$$

Therefore, the capture coefficient is the arithmetic average of the capture probabilities:

$$C = \frac{\sum_{k=1}^{L} C_p[k]}{L} \tag{4.3}$$

This coefficient can be used analytically to add the impact of capture to the throughput performance of MAC schemes in simulation. It depends only on the packet length and considers any overlap length. In order to demonstrate this, new throughput figures are obtained by applying the capture coefficient and are compared with practical results obtained from the test-bed for the ALOHA protocol in Section 4.3.

4.2.3. 2-packet-capture Scenario

A packet transmission can start at any time, so packet overlap can vary. In this study, the number of overlapping bytes in a collision is systematically increased until the capture effect does not occur. In order to create packet collisions, the time between the colliding packets must be carefully adjusted. The minimum overlapping length was chosen as 5-bytes and it was increased in 5-byte steps until the capture effect did not occur. The reason for choosing 5-byte steps is to precisely arrange the time between the colliding packets as the time for transmitting a single bit is tiny. A similar work which uses controlled collisions through 2 packet transmissions with a precise time difference is carried out to detect collisions and recover the first-arriving packet in collisions in [79]. It experimentally showed that the first-arriving packet can be received if the second packet transmission starts after the preamble detection of the first-arriving packet as in our implementation.

The number of observed collisions has to be sufficiently large to give meaningful results. It was found that 6000 or more collisions would enable calculation of the average number of successful captures reliably. Also, the variability of experiment results is considered. Fig. 4.3 represents the graph of the results with maximum and minimum values (average value in the middle of blue marker) for each overlapping length. To see the possible differences when the system restarts, each overlapping ratio is run one hundred times. Table 4.1 presents the practical results: the mean number of packets captured over the hundred runs, with respect to various overlapping lengths as well as the capture probabilities for each overlapping length.

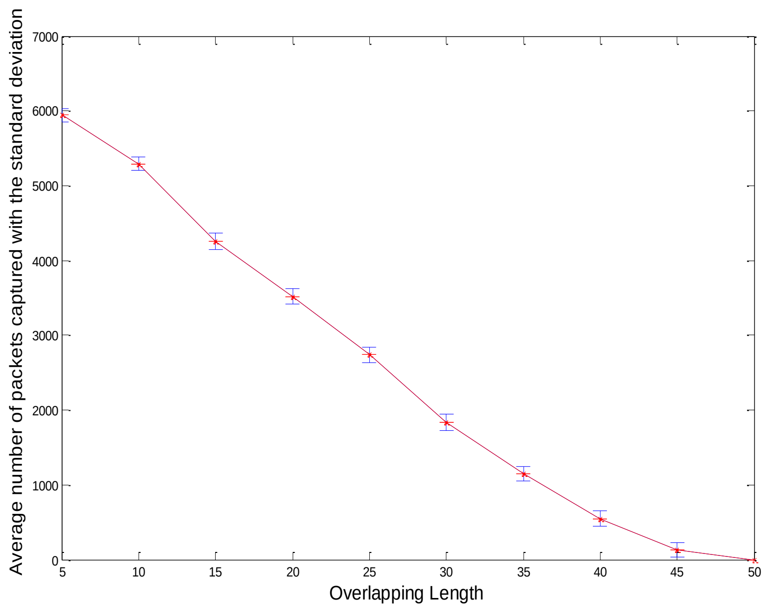


Fig. 4.3 The average number of captured packets with the variability.

The results given in Table 4.1 show that the capture of the first-arriving packet always occurs if there is no overlap between packets. However, there is no capture when the length of overlap exceeds 50-bytes. The reason is that the *SINR* is below the required threshold of the radio transceiver. Between these limits, as the overlap length increases, the number of capture events reduces, as noted by the mean values. This is because the power level of the first-arriving packet is lower than power of the interfering packet as demonstrated in [80], given that the nodes are stationary and power level during the packet reception remains same.

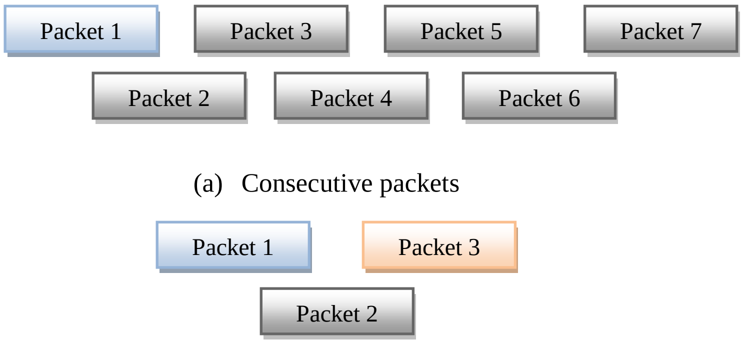
Number of overlapping bytes	0	5	10	15	20	25	30	35	40	45	50
Mean	6000	5945	5290	4266	3605	2744	1836	1155	540	136	0
Capture probability	1	0.99	0.88	0.71	0.60	0.45	0.30	0.19	0.09	0.02	0

Table 4.1 2-packet-capture scenario, number of captured packets for each overlapping ratio and capture probabilities.

4.2.4. 3-packet-capture Scenario

For the case of a higher traffic density in a network, a new capture scenario, practically very likely, is studied. Successive packets may arrive with small time differences, as depicted in Fig. 4.4a. In this model, the possible reception scenarios at the receiver are:

- Reception of a third packet (detection of preamble and CRC check) in the presence of second-packet interference.
- If reception of the third packet fails, the possibility of detecting the preamble of a fourth packet arises.
- If reception of the fourth packet fails, the possibility of detecting the fifth packet arises, and so on.



(b) Simplified model

Fig. 4.4 3-packet-capture scenario.

To make the implementation of this scenario more practical, we used the simplified model shown in Fig. 4b, which is referred to as the 3-packet-capture scenario. In this model, the first and second packets comprise the 2-packet-capture model, as described previously. The overlap between the first and second packets is a small fixed length and the radio always detects the preamble of the first packet. Therefore, the second packet causes distortion on the detection of the third packet preamble. The overlap length between the second and third packet is created by adjusting the transmission time of the third packet. The focus is now only on the reception of the third packet using the same overlapping technique implemented earlier. The outcome of this scenario is presented in Table 4.2, where the mean of number of packets captured is calculated over one hundred runs.

Number of overlapping bytes	0	5	10	15	20	25	30	35	40
Mean	6000	3102	2960	2264	1554	873	401	35	0
Capture probability	1	0.51	0.49	0.37	0.25	0.14	0.06	0.005	0

Table 4.2 3-packet-capture scenario, result of third packet, and number of captured packets for each overlapping ratio and capture probabilities.

It can be seen from these results that nearly half of the packets are missed with only a 5-byte overlap of packet 2 and packet 3. This is because packet 3 has its entire preamble and synchronisation fields distorted by the tail end of packet 2 (since these segments are 5-bytes long). When the overlap length exceeds 40 bytes, no packets are captured. Note that this is a lower threshold than the experiment with 2-packet-capture because half of the packets have already been lost at the preamble detection stage and the initial 5-bytes of overlap.

4.2.5. Adapting the Capture Coefficient

To adapt the capture coefficient to this study, we approximated the length of a packet to be a multiple of 5 bytes. The overlap length was increased in 5 byte increments and, similarly, the probability of all 5-byte increment overlaps is equal. The capture probabilities are given in Tables I and II. Finally, the capture coefficient, *C*, is given by:

$$C = \frac{\sum_{k=1}^{L/5} C_p[5k]}{L/5} \tag{4.4}$$

This coefficient has been used to add the impact of capture on the throughput of pure ALOHA in next section.

4.3 ALOHA NETWORKS

In order to demonstrate and quantify the impact of the capture effect on the throughput of pure ALOHA, a pure ALOHA network with a finite number of users is initially considered. The throughput calculation with a finite number of users through the binomial formula is presented. Then, the probability of the two capture scenarios occurring in pure ALOHA is calculated. These probabilities are then multiplied by the capture coefficient derived to incorporate the impact of capture on pure ALOHA throughput. Finally, the pure ALOHA throughput formula with a finite number of users and the capture effect is presented. The throughput of the two systems, one with four users and one with twelve users, are systematically analysed, providing both analytical and practical results. Later, the impact of packet size on the throughput of the 12-user system is rigorously analysed, considering three different packet sizes. Finally, the throughput with a high number of users is studied.

4.3.1. Introduction

The ALOHA system was the first multiple access scheme in the world; it was developed at the University of Hawaii, invented by Abramson. There are two standard ALOHA techniques, pure and slotted ALOHA. The pure ALOHA scheme, essentially the easiest and simplest technique based on the application requirements, allows users to transmit their packets as soon as they have a packet for transmission, requiring no pre-coordination with other users accessing the transmission medium. This results in packet collision if more than one user starts transmission simultaneously assuming equal propagation delay for each user. The main assumption in evaluating this scheme is that the colliding packets (either fully or partly) will be lost. Hence, in order to receive a packet correctly, it should be guaranteed that no packets start t_p seconds before or after the start time of a packet, where t_p is the time taken to transmit a packet. Fig. 4.5 indicates the packet reception and a set of packets as overlapped.

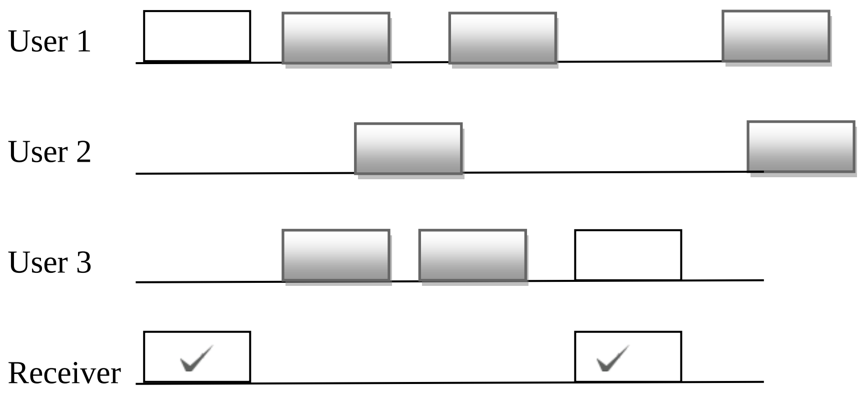


Fig. 4.5 An example of pure Aloha packet reception.

The slotted ALOHA system improves the throughput of pure Aloha by dividing the time into equal length slots and each user is required to transmit at the beginning of a slot. Therefore, complete packet collisions only occur when more than one user transmits in the same slot. Fig. 4.6 depicts an example scenario of the slotted ALOHA packet reception and collisions.

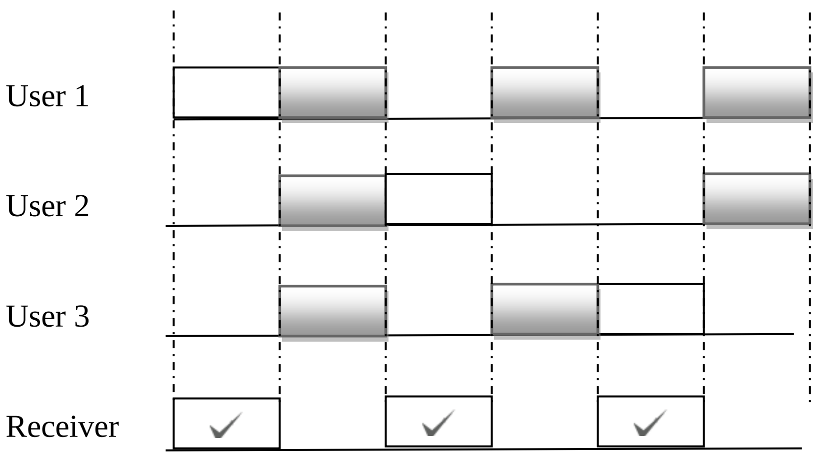


Fig. 4.6 Slotted-Aloha packet reception and rejection.

The throughput analysis of ALOHA is a very simple mathematical calculation and relies on a number of assumptions. These assumptions provide a straightforward piece of mathematics for the analysis. We assume that:

- There is a single receiver and a large number of transmitters, tending to infinity.
- Start times of packets comprise a random Poisson arrival process [81] with an average generation rate of λ packets/second.
- In pure ALOHA, all packet loss is caused by packet collision.
- Any packet overlapping (either fully or partly) causes complete packet loss.
- Until a packet transmission is finished, no new packet can be transmitted.
- All packets have the same length.
- Capture effect is not considered.

Ref. [82] has given the mathematical throughput calculation and plotted the channel throughput *versus* channel traffic load as reproduced in Matlab and shown in Fig. 4.7.

The throughput of pure Aloha:

$$T = G\mathrm{e}^{-2G} \tag{4.5}$$

The throughput of slotted Aloha:

$$T = G\mathrm{e}^{-G} \tag{4.6}$$

Where T denotes the throughput and G is the offered channel traffic.

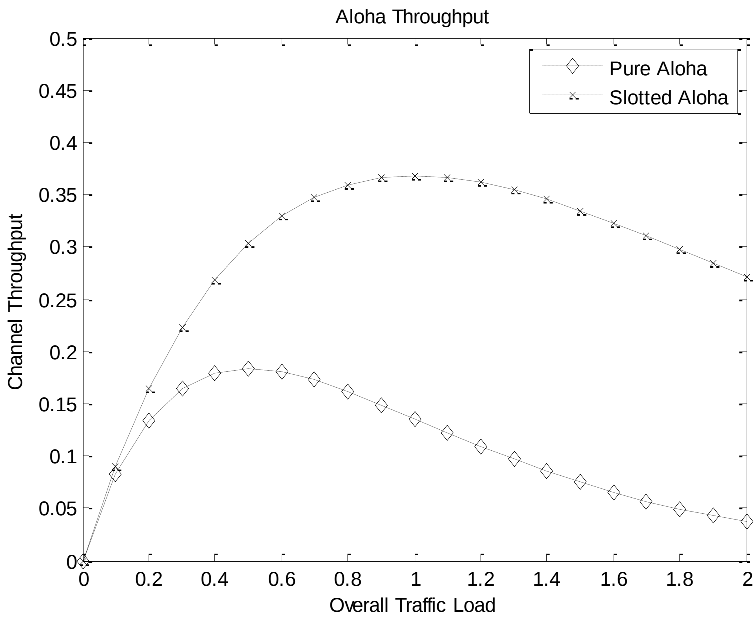


Fig. 4.7 Theoretical throughput of pure and slotted ALOHA.

4.3.2. Throughput Analysis of Pure and Slotted ALOHA

4.3.2.1. Pure ALOHA with Finite Users

A large body of prior work has studied the throughput of ALOHA networks assuming an infinite number of users [48]. However, for practical deployments, it is essential to recalculate the theoretical throughput with a small number of users. Some research has addressed the throughput of Slotted ALOHA with a finite user numbers in [83] as well as through binomial distribution in [84]. However, an extensive review of the literature has not revealed the same analysis for the pure ALOHA case. In this section, the throughput calculation of pure ALOHA with a finite number of users is presented based on the binomial distribution. Consider n users where each user transmits a packet with the same probability (p) based upon the channel traffic level. For a successful transmission, there should be no overlapping transmissions before or after the start time of a packet, as illustrated in Fig. 4.8.

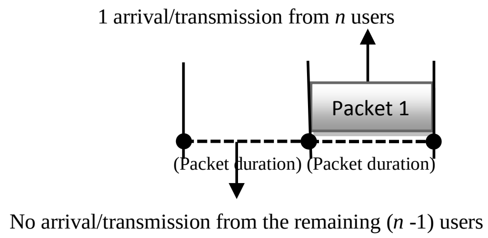


Fig. 4.8 A Successful transmission in pure ALOHA

The probability of k arrivals/transmissions from n users in the time duration of one packet can be given by the binomial distribution as:

$$P(k \text{ in } n) = \frac{n!}{k!(n-k)!} p^k (1-p)^{n-k} \tag{4.7}$$

Therefore, the probability of only one arrival/transmission from the n users is:

$$P(1 \text{ in } n) = \frac{n!}{1!\,(n-1)!} p^1 (1-p)^{n-1}$$

$$= np(1-p)^{n-1} \tag{4.8}$$

The probability of no arrivals/transmissions before the start time of Packet 1 is:

$$P\big(0 \text{ in } (n-1)\big) = \frac{(n-1)!}{0!\,(n-1-0)!} p^0 (1-p)^{n-1}$$

$$= (1-p)^{n-1} \tag{4.9}$$

The probability of a packet arriving and being transmitted in a period equal to the packet transmission duration (p) is related to the channel traffic by the following equation:

$$p = \frac{1}{n} G \tag{4.10}$$

Where G denotes the channel traffic in Erlangs (equivalent to the average number of packets arriving/being transmitted in the packet transmission time) and n is the total number of users in the system, the throughput can be calculated as:

$$\begin{aligned} \text{Throughput} &= P(1 \text{ in } n)P(0 \text{ in } (n-1)) \\ &= n\mathrm{p}(1-p)^{n-1}(1-p)^{n-1} \\ &= np(1-p)^{2(n-1)} \qquad (4.11) \end{aligned}$$

Fig. 4.9 presents the theoretical throughput and the throughput for 5, 10 and 100 users of pure ALOHA. The graph shows that the throughput increases as the number of users reduces because low number of users decreases the probability of packet collision. For example, the maximum throughput is around 0.21 Erlangs with five users whereas the theoretical value was about 0.18 Erlangs for infinite users.

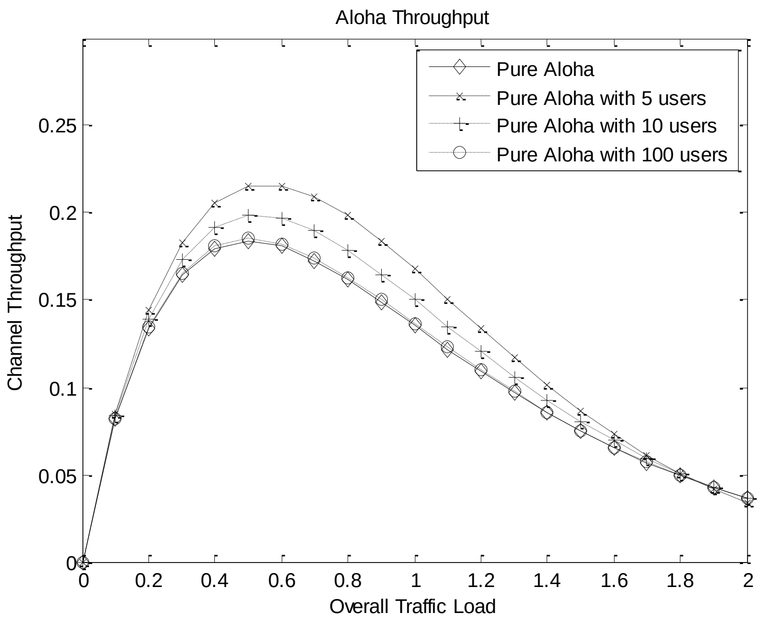


Fig. 4.9 Pure Aloha throughput for various numbers of users.

4.3.2.2. Slotted ALOHA with Finite Users

The throughput analysis of slotted ALOHA is quite similar to that of pure ALOHA. The focus is now on the probability of a user transmitting in a particular slot, shown in Fig. 4.10.

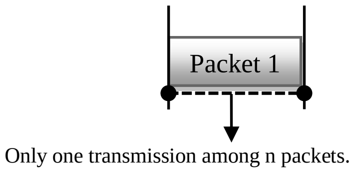


Fig. 4.10 A successful transmission in slotted Aloha

Therefore the probability of one arrival among n packets is:

$$P(1 \text{ arrival in } n \text{ packets}) = \frac{n!}{1!\,(n-1)!} p^1 (1-p)^{n-1}$$

$$= np(1-p)^{n-1} \tag{4.12}$$

Similarly, the throughput of slotted ALOHA is:

$$\text{Throughput} = np(1-p)^{n-1} \tag{4.13}$$

Fig. 4.11 presents the theoretical throughput and the throughput for 5, 10 and 100 users of slotted Aloha.

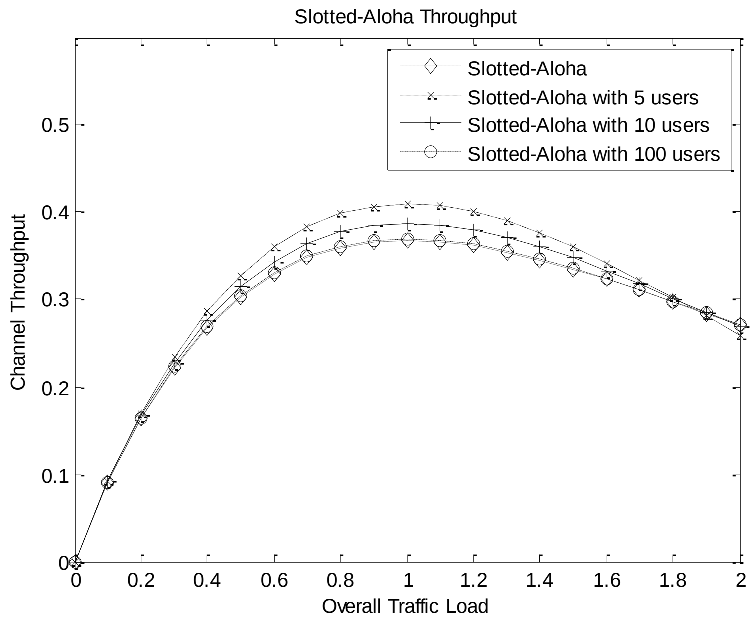
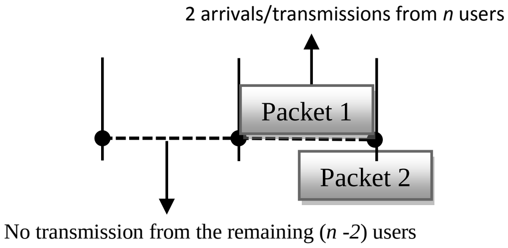


Fig. 4.11 Slotted Aloha throughput for various numbers of users.

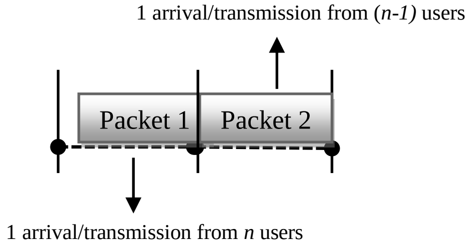
As with pure Aloha, the graph shows that the throughput increases as the number of users decreases due to low probability of packet collision. The maximum theoretical throughput of slotted Aloha is nearly 0.36 Erlangs but it increases to about 0.41 Erlangs with five users. When the traffic load is low, between 0.1- 0.4, and too high, between 1.6 - 2.0, there is not much difference in throughput. However, when the traffic load is between 0.4 - 1.6, the throughput changes moderately.

4.3.3. Throughput Analysis of Pure ALOHA with the Capture Effect

The probabilities of 2-packet-capture and 3-packet-capture based on the scenarios described previously are derived for the pure ALOHA scheme. Then, these probabilities with the capture coefficient are added to the throughput of pure ALOHA. There are two cases for the collision of two packets, depicted in Fig. 4.12.



(a) Case 1



(b) Case 2

Fig. 4.12 Two possible cases for the collision of two packets.

The probability of a two packet collision is the summation of the two cases:

$$P\ (\text{collision of two packets}) = P(\text{case 1})\ +\ P(\text{case 2}) \tag{4.14}$$

The probability of a 2-packet collision in Case 1 is:

$$P(\text{case 1}) = P(2 \text{ in } n\,)P(0\ \text{ in } (n\ -2)\,)$$

$$= \frac{(n)(n-1)}{2} p^2(1-p)^{2(n-2)} \tag{4.15}$$

The probability of a 2-packet collision in Case 2 is:

$$P(\text{case 2}) = P(1 \text{ in } n\,)P(1 \text{ in } (n-1))$$

$$= n(n-1)p^2(1-p)^{2n-3} \tag{4.16}$$

Combining the two cases, the probability of a 2-packet collision is:

$$n(n-1)p^2\left(\frac{(1-p)^{2(n-2)}}{2}+(1-p)^{2n-3)}\right) \quad (4.17)$$

Having calculated the probability of 2-packet-capture, we now derive the probability of the 3-packet-capture scenario which again includes two cases shown in Fig. 4.13.

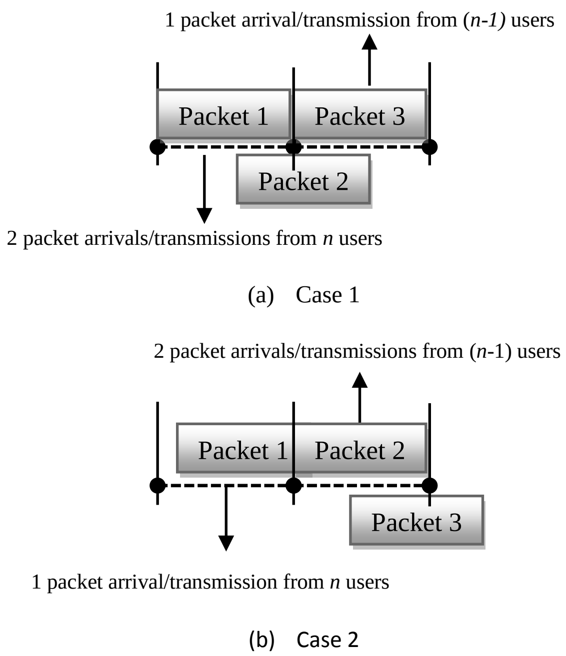


Fig. 4.13 Two possible cases for the collision of three packets.

Similarly, the probability of 3-packet collision is the summation of the two cases:

$$P(\text{collision of three packets}) = P(\text{case 1}) + P(\text{case 2}) \quad (4.18)$$

The probability of 3-packet collision in Case 1 is:

$$P(\text{case 1}) = P(2 \text{ in } n)P(1 \text{ in } (n-1))$$

$$= \frac{n(n-1)^2}{2}p^3(1-p)^{2(n-2)} \quad (4.19)$$

And the probability of 3-packet collision in Case 2 is:

$$P(\text{case } 2) = P(1 \text{ in } n)P(2 \text{ in } (n-1))$$

$$= \frac{n(n-1)(n-2)}{2} p^3 (1-p)^{2(n-2)} \quad (4.20)$$

Combining the two cases, the probability of a 3-packet collision is:

$$\frac{n(n-1)p^3(1-p)^{2(n-2)}}{2} \big((n-1) + (n-2)\big) \quad (4.21)$$

Using the capture coefficient derived (see section 4.2.2), the throughput accounting for the capture effect (combining the equations 4.11, 4.17 and 4.21) is given by:

$$\text{Throughput} = np(1-p)^{2(n-1)} + n(n-1)p^2\left(\frac{(1-p)^{2(n-2)}}{2} + (1-p)^{2n-3)}\right)C_2 + \frac{n(n-1)\mathrm{p}^3(1-\mathrm{p})^{2(n-2)}}{2}\big((n-1) + (n-2)\big)C_3 \quad (4.22)$$

Where C_2 denotes the capture coefficient for 2-packet collision and C_3 for 3-packet collision.

4.3.4. Performance Evaluation

The test-bed consists of a certain number of transmitters (4 and 12) and a single receiver, all in range of each other. Fig. 4.14 presents a view of the test-bed. Each node generates a packet with exponentially distributed inter-arrival time and immediately transmits it. Each node has the same average inter-arrival time which is given as below:

$$I = = \frac{L.N}{G.D} \quad (4.23)$$

Where I denotes the average packet inter-arrival times, L is the packet length, N is the number of nodes in the network, G is the generated traffic and D is the data rate of the channel. The transmit power of all packets is set to the same level. Table 4.3 summarizes the implementation parameters.

Parameters	Values
Channel Bit Rate	250 Kbits/s
Packet Length	50, 75, 100, 125 bytes
Transmit Power	2 mW
Number of Transmitters	4 and 12

Table 4.3 ALOHA implementation parameters.

(a) 4-user ALOHA system

(b) 12-user ALOHA system

Fig. 4.14 ALOHA implementation test-bed, a receiver in the middle and transmitters.

Fig. 4.15 and 16 show the practical throughput characteristics for the 4- and 12-node systems respectively, which are compared to the finite user analytical model with and without the capture coefficient. The standard infinite user curve is also included for comparison. For the practical implementation, we have chosen a 50-byte packet size, as the capture effect did not occur beyond an overlap of this length.

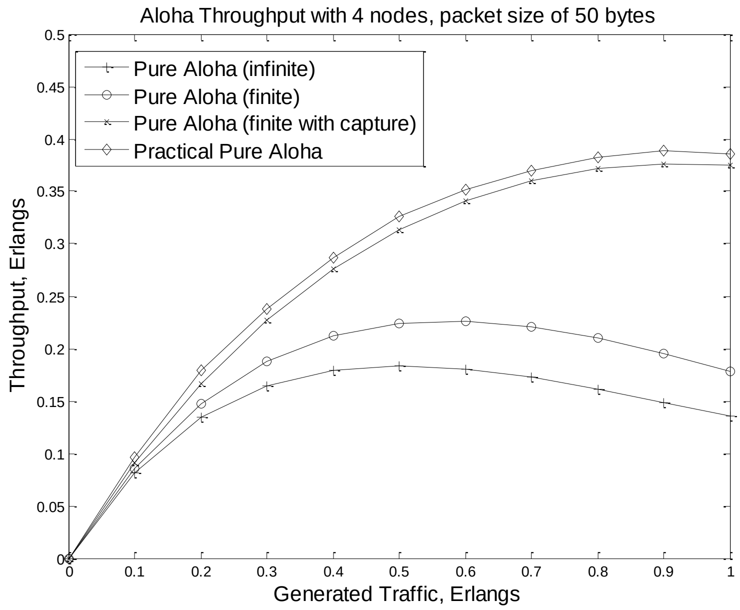


Fig. 4.15 ALOHA throughput with 4 users.

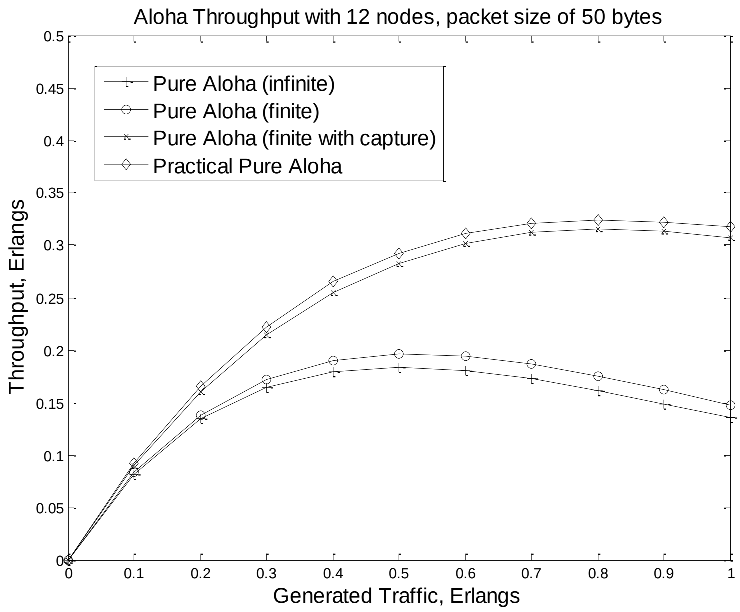


Fig. 4.16 ALOHA throughput with 12 users

The throughput with a finite number of nodes is higher than the infinite node case and, as expected, as the number of nodes increases, the throughput difference decreases. With the capture effect model, the maximum throughput of the 4-user scenario is slightly greater than double that predicted by the standard infinite node theory model and for the 12-user case it is enhanced by two-thirds. The practical results show that our capture effect model closely matches the practical results. The small discrepancy is because it is impossible to model and implement every capture scenario and we only modelled two of the most common scenarios. It can be concluded that the two capture scenarios represent the majority of the capture effect.

The impact of packet length on throughput has been studied since the capture coefficient is dependent on packet size (see Equation 4.3). Therefore, an important purpose of this study is to demonstrate the capture effect in association with varying packet lengths. Three different packet lengths (75, 100 and 125-bytes long) have been implemented with 12 nodes in the network. Fig. 4.17 presents the throughput performance of pure ALOHA with both the analytical (A) and practical (P) results.

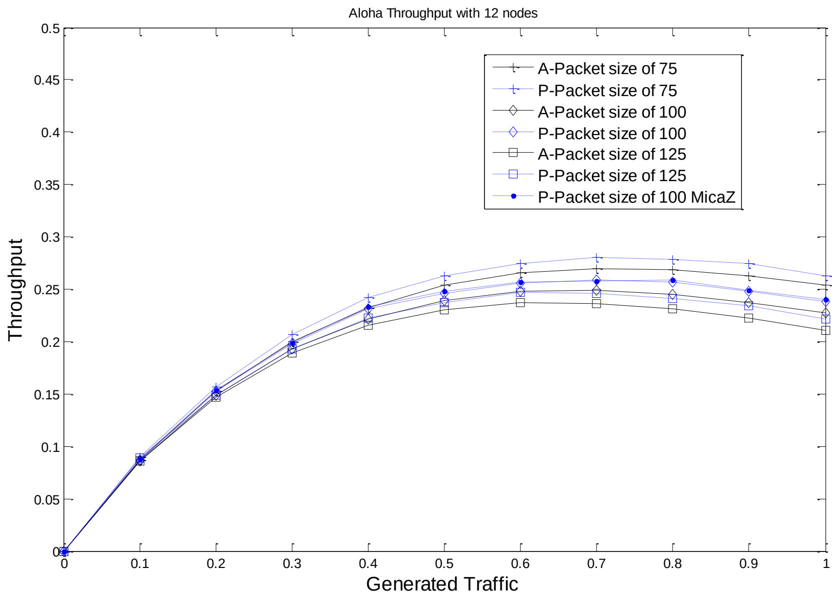


Fig. 4.17 ALOHA throughput with 12 users in association with packet lengths.

Fig. 4.17 illustrates that the throughput has an increasing trend as the packet size decreases. The analytical model effectively predicts the practical results with a small difference of approximately 0.01 Erlangs. A maximum throughput of around 0.28 Erlangs is achieved with a 50-byte packet when the traffic load is 0.7 Erlangs. On the other hand, a packet size of 125-bytes can achieve around 0.245 Erlangs throughput. The packet size plays an important factor when considering the influence of the capture effect on performance. The capture effect has greater influence with smaller packets where the capture coefficient has a high value, resulting in higher throughput. In order to test the capture model on a different sensor platform, a practical implementation using MicaZ nodes with a packet size of 100-bytes has been carried out and the results of this implementation are presented in fig 4. 17. It is seen that similar throughput results have been obtained. This is because MicaZ and Iris nodes have similar characteristics such as the same modulation scheme (see sections 4.2.1 and

6.3.1).

A 100-node system is now considered in order to demonstrate the effectiveness of the finite user capture model in predicting the performance that would be observed practically with a much larger test-bed. The impact of packet size is again considered. Fig. 4.18 presents the analytical throughput predictions.

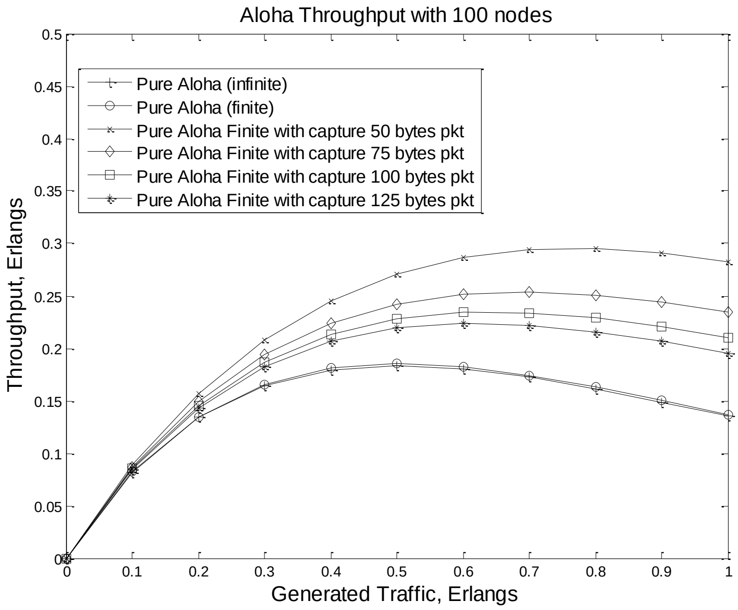


Fig. 4.18 ALOHA throughput with 100 users in association with packet lengths.

It is clear that the theoretical throughput with 100 nodes approaches the infinite node theoretical value. A maximum throughput of 0.295 Erlangs is achieved with a traffic load of 0.8 Erlangs for a packet size of 50-bytes, whereas it is around 0.223 Erlangs with a traffic load of 0.6 Erlangs for a packet size of 125-bytes. It can therefore be concluded that if the packet size is decreased, the throughput increases as an overall trend.

4.3.5. Summary

In this chapter, the capture effect has been investigated to clarify its impact based on real-world measurements. Contrary to theory, it is shown that using the same transmission power level at the same distance, the capture effect is seen. The impact of capture is modelled and incorporated into the performance evaluation of wireless sensor networks through simulation. The throughput of pure ALOHA with a finite-user case has been modelled for practical implementation. The increase in the throughput of pure ALOHA due to the capture effect has been presented. Two factors affecting ALOHA performance have been investigated, packet size and number of users, using both mathematical analysis and practical implementations. Results show that the throughput capacity of pure ALOHA with the capture effect is better than that predicted by the standard assumptions. The maximum throughput in a 4-user scenario is slightly more than double that predicted by theory (37.62%). It can be predicted that a high-user-density system would tend towards the theoretical limit based on the validated model. It can be concluded that the throughput of existing MAC schemes may be greater than currently predicted as a result of the capture effect described here.

5. Practical Implementation Issues of RL-ALOHA for WSNs

Contents

This chapter presents the basic concepts behind Reinforcement Learning (RL) strategy which is applied to Framed ALOHA whereby a repeating frame structure within Slotted ALOHA is employed. A novel MAC protocol for single-hop communication, Reinforcement Learning Based ALOHA (RL-ALOHA), is described. The practical implementations of RL-ALOHA are studied. A potential real-world phenomenon that impacts on performance is brought to light and a new scheme is proposed to deal with the phenomenon which reinstates the ideal throughput performance of RL-ALOHA. The practical results show that the proposed scheme improves the convergence properties to the steady state of a unique transmission slot assigned per node.

5.1 Introduction

With a key benefit of simplicity, ALOHA based schemes have been explored. Combining them with intelligent transmission strategies can mitigate the drawbacks of employing a blind transmission strategy. In recent years, researchers have exploited RL techniques in the development of MAC protocols, thereby achieving high channel throughput (see section 3.3.4). Here, a WSN node determines an optimum transmission strategy based on its past experience. RL-ALOHA assigns a weight value to each slot in the repeating frame and the weight values are updated after transmissions based on the reward signal. A typical example of the reward signal is (+1) for successful transmission and (-1) for failure, as implemented in [66] which results in enhancements in the essential requirements such as channel throughput and energy efficiency, assuming a perfect simulation environment. The advantage of this learning process for Framed Slotted ALOHA is that the nodes quickly learn to select the best transmission slot as a function of the accumulated reward history.

This chapter aims to provide a practical complement to the development of RL-based ALOHA protocols for single-hop communication, bringing to light a real-world phenomenon that impedes the learning algorithms. Since the transmission medium and sensor nodes are not perfect, there will be some loss of reward signals even though the best action may have been chosen with success. The learning process in the presence of reward loss results in some nodes failing to properly operate the learning system effectively in practice. It causes the nodes without a unique transmission slot to jump around other users' slots. An intelligent slot selection technique is introduced which accounts for this practical issue and minimizes the effect of reward loss, in particular when the channel load is high.

5.2 RL-ALOHA Protocol Description

A repeating frame structure is introduced within Slotted ALOHA. A frame comprises a fixed number of slots which is, optimally, equal to the number of nodes in the WSN so that each user can have a unique slot. A node is only allowed to send one packet per frame. The generated packets are queued first-in-first-out with the packet at the head being transmitted. For each node, every slot in a frame has a weight value which is cumulatively updated with respect to the reward received. The slot with the highest weight is preferred and, if ready for transmission, a packet is sent in this slot. If more than one slot has the same weight, one of them is chosen at random. Weight values are denoted by W(i, s) as shown in (5.1) after each packet transmission:

$$W_{t+1}(i, s) = W_t(i, s) + r \tag{5.1}$$

Where i indicates the present node, s is the preferred slot and r is the current reward. Successful transmissions (an acknowledgement packet received) will be rewarded with r = +1 and failures will be rewarded with r = -1. This strategy performs well in terms of throughput, delay and energy consumption because it significantly reduces the likelihood of collisions.

Fig. 5.1 presents an example of the RL-ALOHA strategy in which a frame includes 5 slots. In this example, the node randomly selects slot 1 among all slots and the transmission is successful in this slot. The weight value of slot 1 is immediately increased by a reward of +1 and the node transmits a packet in slot 1 of frame 2 as slot 1 has the highest weight. The transmission in frame 2, however, fails and the weight value is now reduced by a reward of -1. In frame 3, all the slots have the same weight, which results in the node randomly choosing the slot 5. The transmission fails again and the weight value of slot 5 is reduced. In frame 4, the node picks the slot 3 at random among the slots 1 to 4. The transmissions in slot 3 continue to be successful for the next N-3 frames.

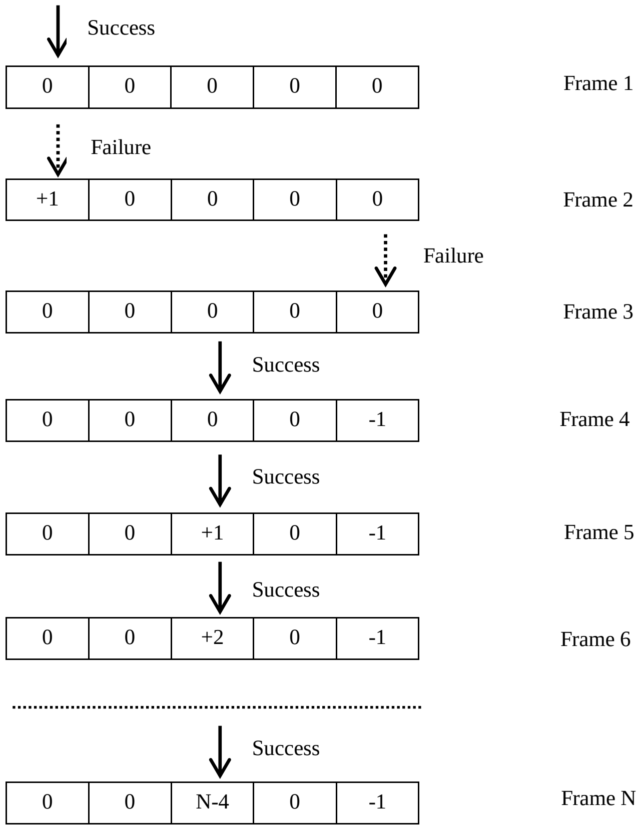


Fig. 5.1 An example of RL-ALOHA process.

5.3 RL-ALOHA Protocol Implementation

In this section, practical performance validation of the RL-ALOHA protocol with the consideration of the practical issue is presented. A common single-hop topology is considered and used for the performance evaluation. This topology is simple but can be considered as a one-hop sub-network in complicated network scenarios, such as a cluster in LEACH protocol [56].

5.3.1 Performance Evaluation

5.3.1.1 Experimental Setup

The IRIS nodes, IEEE 802.15.4-compliant devices, are the hardware platforms used in all the experiments. Tinyos-2.x is used to program the nodes and to observe data-exchange on a computer through a base-station. In our implementation of a packet, the SHR, PHR, MHR, PAYLOAD and CRC fields comprise 5-bytes, 1-byte, 10-bytes, 110-bytes and 2-bytes respectively (see section 4.2.1 for definitions of these fields). The control packets are normally expected to be very small without PAYLOAD. Another packet type is therefore created, the acknowledgement packet, which has a 5-byte SHR, a 1-byte PHR, a 10-byte MHR and a 2-byte CRC. This packet is only aimed to confirm the successful reception of the transmitted packets at the receiver so that no PAYLOAD is required.

Performance is evaluated under a star topology which comprises 10 users as depicted in Fig. 5.2. All the nodes are in range of each other, are equidistant from the receiver and transmit at the same power level. Each node generates a packet with an exponentially distributed inter-arrival time. All the nodes have the same mean packet inter-arrival time and are synchronised by the receiver. After the deployment of the nodes, the transmitters wait in the receive mode for a specific packet called a *Hello* packet sent from the receiver. Once the receiver is powered up, and after a certain time, it transmits the *Hello* packet to all nodes. This synchronises the transmitters to enable their packet generation process to run concurrently. Reception of the *Hello* packet is indicated by one of three LEDs lighting. Table 5.1 summarises the experimental parameters.

In order to avoid the overhearing problem as all the nodes are in the range of each other, after a packet is transmitted, the transmitting node switches its radio status from transmit mode to receive mode in order to receive an ACK packet. Upon expiration of ACK packet waiting time, the node switches its radio back to transmit mode not to overhear. Also, retransmissions of the lost packets are not taken into consideration, meaning that a packet is removed from the queue after its transmission.

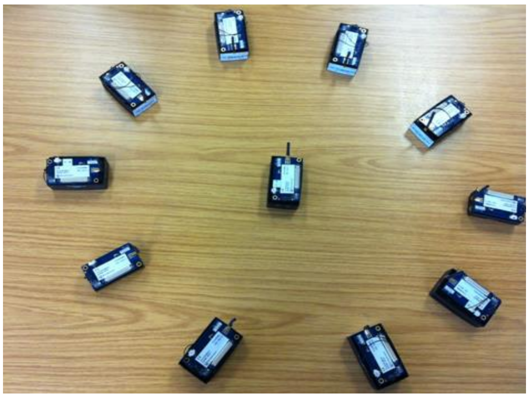


Fig. 5.2 Network topology, a receiver in the middle and ten transmitters.

Parameters	Values
Channel Bit rate	250 kbits/s
Transmit power	2 mW
Data packet length	1024 bits
ACK packet length	144 bits
Slot length	1200 bits
Number of slots per frame	10
Number of nodes	10
Experiment Period	500,000 slots

Table 5.1 Experiment parameters for RL-ALOHA.

Channel throughput is the percentage usage of the channel capacity. For data reception, the theoretical maximum throughput for the experiment parameters given in Table 5.1 is close to 0.85 Erlangs (1024 bits / 1200 bits). The slot length is determined to sufficiently include data and ACK packet lengths. One Erlang represents continuous use of the channel. A small guard period is left between slots in order to mitigate the propagation delays. Steady state occurs when all nodes have found an empty slot.

5.3.1.2 Acknowledgement Packet Loss

In simulation, sensor hardware is able to reply with an ACK packet (if requested) as soon as it receives a data packet. In slotted-based transmission strategies, for a successful packet reception/transmission, all the packet receptions/transmissions (data packet and control packets) occur in a single slot. In order to avoid channel wastage, slots are designed as small as possible where the transmissions in a slot happen in succession. In practice, IRIS nodes have various modes (transmission, reception, idle or off) as they can only stay in one of these modes at a time which is known as half-duplex communication (see section 4.2.1). For example, particularly in slotted-based systems, after a packet transmission, a transmitting node may need to switch its radio to the reception mode to receive an ACK packet from the receiving node while the receiving node switches its radio to transmission mode to send an ACK packet back after the successful packet reception. The practical observations conducted with IRIS nodes show that the sensor platform does not switch between transmission and reception modes very quickly. In particular, the receiving nodes which have more traffic can fail to switch the radio between the modes. Consequently, a more substantial guard period to allow for mode change must be employed. Since the sensor nodes are simple devices and have limited capacity, this wastes channel resources.

In order to test this phenomenon, a practical implementation on the test-bed described in the previous section was carried out. The transmitters in the single-hop communication network are given unique transmission slots manually at the beginning. A packet transmission happens in each slot and the receiver responds with a small ACK packet without guard periods employed. The transmitters make a decision as to whether a transmission has succeeded or not based upon the reception of ACK packet. In this implementation, all the packets are successful at the receiver side but the receiver is not able to send all the ACK packets. The ratio of the number of ACK packets received to the number of packets transmitted at all transmitters is recorded. The results will show that the guard period plays an important role in increasing the proportion of ACK packets sent. Without it, as Fig. 5.3 shows, using the hardware platform, 55% of the ACK packets are lost at a channel load of 1 Erlang.

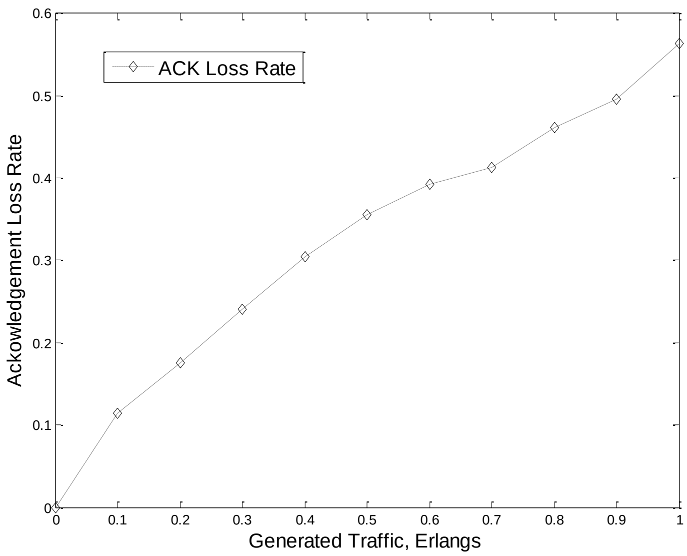


Fig. 5.3 Acknowledgement packet loss rate without guard period.

These results are obtained in the steady state where each user has a unique slot so there is no collision. The loss rate increases with higher generated traffic levels because the receiver has increasingly insufficient time to switch to the transmission mode because the receiver has lots of incoming traffic which results in the receiver continually switching between modes.

5.3.1.3 Results

In order to establish the ideal performance of the protocol, RL-based Slotted ALOHA is first simulated with 10 users in OPNET, using the same parameters as in the practical system and, assuming no ACK packet loss. Then, using hardware, practical RL-based slotted ALOHA is implemented, firstly without a guard period and then with sufficient guard period to enable the receiver to send all the ACK packets. The practical conventional Slotted ALOHA is then implemented with 10 users (see the analytical analysis in section 4.3.2.2). Fig. 5.4 presents the results of all these scenarios. In all cases, the results are captured from the moment the network run starts to demonstrate the overall performance.

The simulation results show that the throughput increases linearly and reaches a maximum of 0.85 Erlangs which corresponds to every node finding a unique slot, since there is an ACK packet overhead of 0.15 Erlangs.. The practical loss of ACK packets impedes the learning process, resulting in some nodes failing to find a unique slot. The residual contention reduces the throughput on the channel. Provision of the guard period resolved the problem of ACK loss, but incurs a significant cost. Although the guard period ensures more effective learning such that each user finds a dedicated transmission slot, the guard period overhead means that throughput is not at an acceptable level. At higher traffic loads, additional contention during the learning phase delays the onset of the steady state which serves to reduce the throughput experienced from the start of the run. It is even worse than conventional slotted ALOHA.

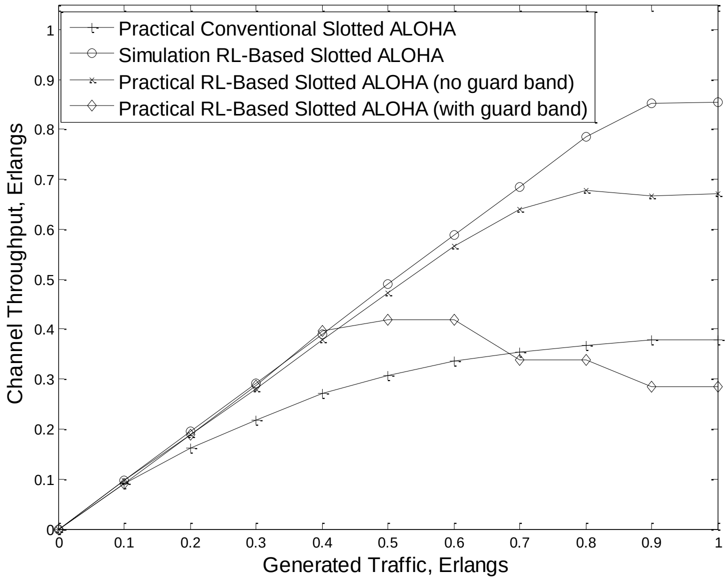


Fig. 5.4 Channel Throughput for RL-ALOHA.

5.3.2 Dynamic Punishment Strategy

In the previous section, it is shown that ACK packet loss has a significant effect on the learning process and channel throughput. Use of a guard period provides an efficient learning process but the channel throughput reduces dramatically with increasing channel traffic due to the increased overhead. This section describes the proposed scheme with its implementation which provides the ideal channel throughput in the presence of ACK packet loss.

5.3.2.1 Description

The purpose of the proposed scheme is to reduce the magnitude of the punishment during the period in which nodes search for an empty slot. When the ACK loss rate is beyond a certain value, 50% based on the experiments, the effect of the reward (positive ACK) is insufficient to support an effective learning process. Therefore, the learning punishes more at higher loss rates, even though the packets are being received successfully by the receiver.

It is proposed to replace the fixed numerical value of the punishment (-1) with the probability of success in each slot as table 5.2 shows. If the probability of success is high, collisions rarely happen and an ACK is rarely lost. The punishment has a high magnitude. However, if the probability of success is low, either a packet collision is likely or the hardware does not respond (practical issue), so that the punishment has a low magnitude. Therefore, this strategy will ensure an effective learning process which minimizes the packet loss regardless of its reason.

Reward	Punishment
1	$-\left(\text{Probability of success of the preferred slot}\right)$

Table 5.2 Value of weighting signals.

This variation has a key benefit of maintaining the unique slots, even under the worst case where the receiver is unable to send ACK packets or the transmitters cannot receive them. In this case, the transmitters will ideally keep transmitting in their dedicated slots. The punishment magnitude will keep decreasing as a result of the

ACK packet loss. The weight value of the unique slot reduces more slowly compared to the original scheme. It will eventually result in exploration for new unique slots.

At the beginning, all the weight values are initialized to 0. Fig. 5.5 shows an example of the reinforcement learning algorithm with the proposed slot selection technique.

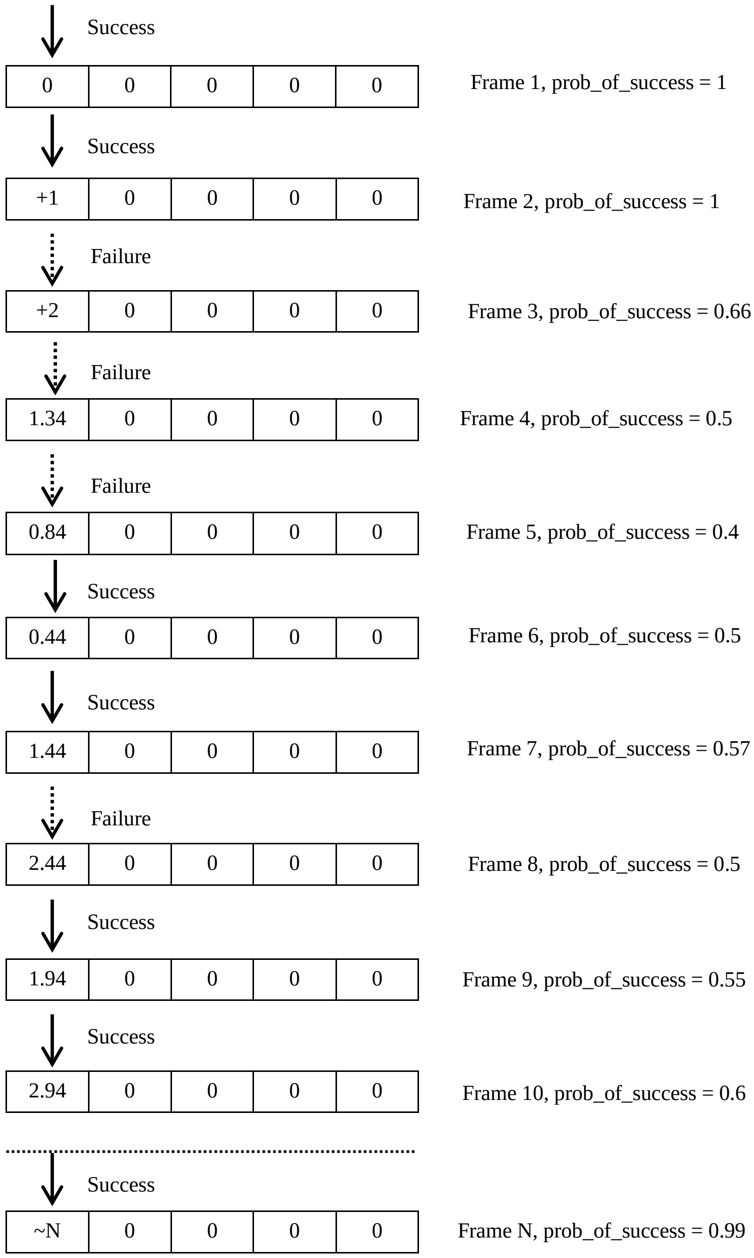


Fig. 5.5 An example of the proposed algorithm with the probabilities.

In this example, slot 1 is randomly chosen and two transmissions succeed in this slot. Then, 3 consecutive failures occur which reduces the probability of success in slot 1. The slot 1 is still preferable as it has the highest weight value. In the original scheme, slot 1 would be lost after 2 failures but the proposed scheme protects the willingness level of slot 1 to be selected based on the transmission history. The proposed scheme protects the current preferred slot from successive failures.

It is worth mentioning that a fix lower magnitude of the punishment instead of probability of success would protect the throughput performance at a particular loss rate. However, the packet loss rate can vary significantly considering dynamic environment and changing channel conditions, so that the use of probability of success for punishment value ensure more protective property to the system in achieving the steady-state at any loss rate as presented in the following section.

5.3.2.2 Implementation of Punishment Variation

The RL-ALOHA protocol has been implemented, using the proposed punishment variation, on the test-bed described in Section 5.3.1.1. Fig. 5.6 shows the throughput performance under varying traffic loads. It can be clearly seen that our learning scheme achieves the optimum performance that is obtained in simulation under perfect conditions. All the nodes end up with a unique slot after the learning process and keep transmitting in it.

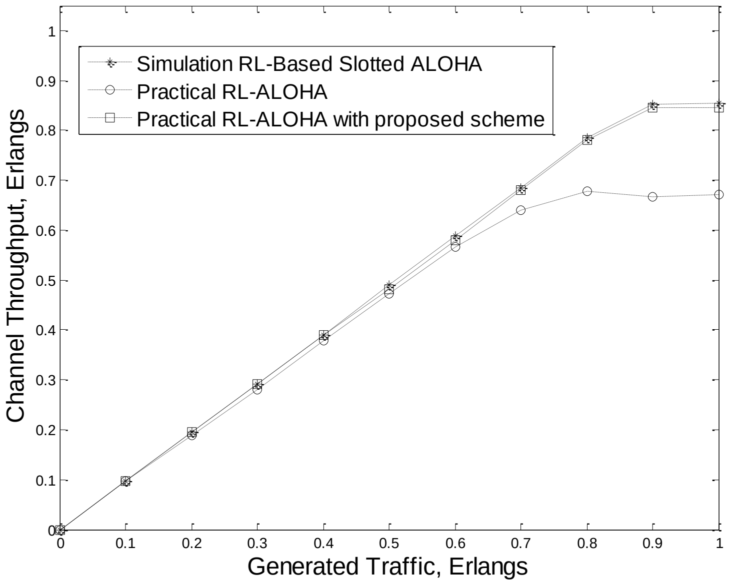


Fig. 5.6 Channel Throughput with the proposed scheme.

The ACK loss ratio varies at different periods of time depending on the hardware. It is therefore worth observing the *increasing/decreasing* trend of the weight value. Fig. 5.7 shows that the weight value of the preferred slot of a node at a channel loads of 1, 0.6 and 0.4 Erlangs. Regardless of the loss ratio, the weight value linearly increases. Although we showed the weight value of only one node, observations demonstrated that the weight values of the rest of the users have a similar trend.

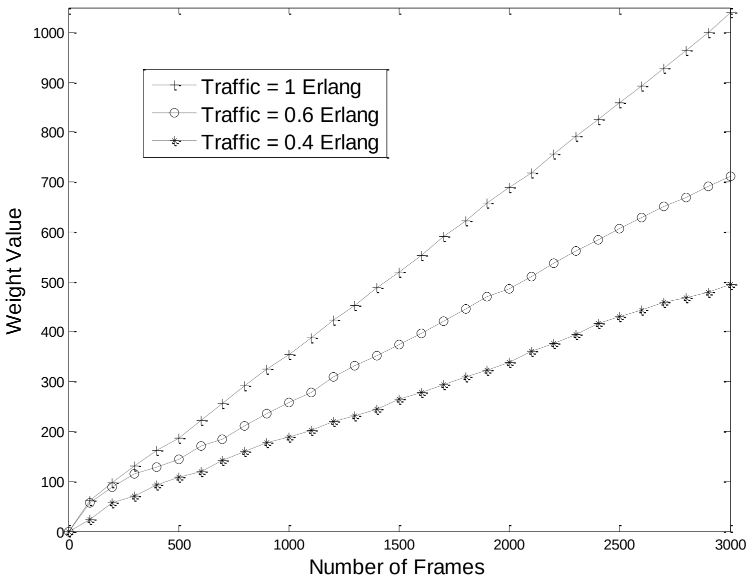


Fig. 5.7 Increase trend of the weight value of the preferred slot.

5.4 Summary

ALOHA has a key advantage of simplicity and low overheads, but it has a drawback of a blind transmission strategy, resulting in low throughput under high channel loads. This chapter presented the RL-ALOHA protocol which employs reinforcement learning as an intelligent slot selection strategy, thereby achieving high channel performance. RL-ALOHA only requires small ACK packets which represent a relatively low overhead. Practical results demonstrate that a maximum throughput of 0.85 Erlangs due to the ACK packet (18-byte) overhead of 0.15 Erlangs. The length of the ACK packet depends on the radio platform, so that different radio platforms may provide smaller ACK packets. For example, a CC1101 radio platform [85] can provide a 9-byte ACK packet.

Due to the practical issue of ACK packet loss, the learning process does not work as effectively inducing lower throughput performance. A new slot selection punishment strategy has been proposed to resolve this issue, based on the probability of success in each transmission slot. It achieves the optimum throughput performance. The steady state results show that the system converges in a period of a few seconds. The complete process can be readily implemented, incurring minimal resources on a potentially highly resource-limited WSN node. It is worth to mention that the issue of ACK packet loss depends highly on nodes platform. For example, implementations using MicaZ nodes, as presented in the next 2 chapters, show that MicaZ nodes switch quickly between the transmission and reception modes requiring no guard time. Therefore, original RL-ALOHA scheme achieves its optimum performance with the fix punishment value of -1.

RL-ALOHA implements a repeating frame structure based on the conventional Slotted ALOHA. A weight value is assigned to each slot in a frame which is updated by success and failure. The slot with the highest weight is preferred. The main drawback is the continuous increase of the weight value after the system converges. A large value of weight value can cause the network not to adapt quickly when the environment changes (possible some slots become unavailable). Setting the right frame size which must allow each user to find a unique slot is crucial as it has a significant impact on the channel performance. A large frame size would cause unnecessary channel wastage, and a small frame size causes the network not to achieve an optimum steady-state condition. In a single-hop scenario, the optimum frame size is equal to the number of users in the network.

6. Practical Implementation and Stability Analysis of ALOHA-Q for WSNs

Contents

This chapter presents a description, practical implementation and stability analysis of a recently-proposed, energy-efficient, Medium Access Control (MAC) protocol for Wireless Sensor Networks (WSNs), ALOHA-Q, which employs a reinforcement learning framework as an intelligent transmission strategy. The channel performance is evaluated through simulation and experiments conducted with a real-world test-bed. The stability of the system against possible changes in the environment and changing channel conditions is studied with a discussion of the resilience level of the system. A Markov model is derived to represent the system behaviour and to estimate the conditions under which the system loses its steady-state operation. A novel scheme is proposed to serve to protect the lifetime of the system when the environment and channel conditions do not sufficiently maintain the operation of the system.

6.1 Introduction

ALOHA and Q-Learning have been integrated to establish a new MAC protocol, namely ALOHA-Q which uses the same slot selection algorithm presented in the RL-ALOHA protocol [68]. Q-Learning was the RL algorithm used as intelligent slot selection strategy. The ALOHA-Q scheme inherits the merits of contention-based and schedule-based approaches while offsetting their drawbacks. ALOHA-Q uses slotted-ALOHA as the baseline protocol with a key benefit of simplicity. It allows users to find unique transmission slots in a fully distributed manner, resulting in a scheduled outcome. ALOHA-Q aims to reach an optimal steady state where each user within an interference range has unique (interference free) transmission slots. Therefore, ALOHA-Q works like a contention-based scheme gradually transforming into a schedule-based scheme, which can be fully achieved in the steady-state without the need for centralised control and scheduling information exchange in steady state conditions. In order to demonstrate the effectiveness of this approach, performance evaluations in both single-hop and multi-hop communication scenarios have been carried out [65-68]. These studies show that ALOHA-Q can achieve a high channel utilisation while minimising the energy consumption. However, these evaluations are restricted to simulation-based evaluation, where the practical implementation issues are not apparent.

This chapter provides a practical evaluation of ALOHA-Q protocol for the first time. The performance of ALOHA-Q is compared with a well-known MAC protocol Z-MAC. Z-MAC has been chosen for performance comparison because it is an effective MAC scheme which provides high channel utilisation and energy-efficiency. Another reason of choosing Z-MAC is that Z-MAC uses a similar frame structure as in ALOHA-Q which makes it appropriate for performance comparison. The resilience level of ALOHA-Q to loss of convergence is explored in order to consider the weakness of the scheme in the presence of packet losses in the steady state. The resilience level to loss of convergence is presented according to various packet loss probabilities. A Markov model is derived to estimate the time to loss of convergence for a single user. Then, a novel technique is proposed to protect the convergence lifetime in the presence of packet loss.

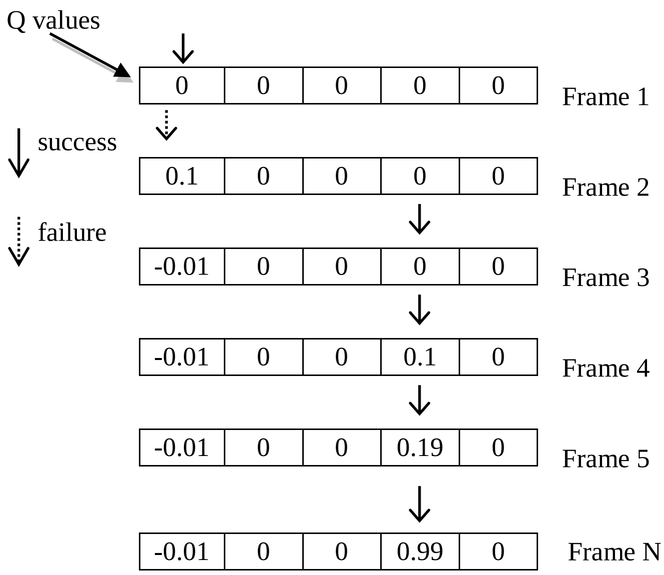
6.2 ALOHA-Q Protocol Description

A similar scheme to that of RL-ALOHA is used, whereby time is divided into repeating frames. Each frame consists of a certain number of slots (N) for data transmission, which should be appropriately set in order to allow each node to have a unique slot. In a single-hop scenario, N is optimally set to the number of nodes in the system. However, in a multi-hop scenario, N is determined by local transmission and interference range of the nodes, network topology, the density of the nodes, and the number of source nodes along the route. Nodes are restricted to access only one slot per frame for their generated packets and they can transmit in multiple slots in a frame for relaying the received packets. The generated packets are queued first-in-first-out with the packet at the head being transmitted first. Each slot is initiated with a Q-value to represent the willingness of this slot for reservation, which is initialised to 0 on startup. Upon a transmission, the Q-value of corresponding slot is updated, using the Q-learning update rule given by Eq. (6.1):

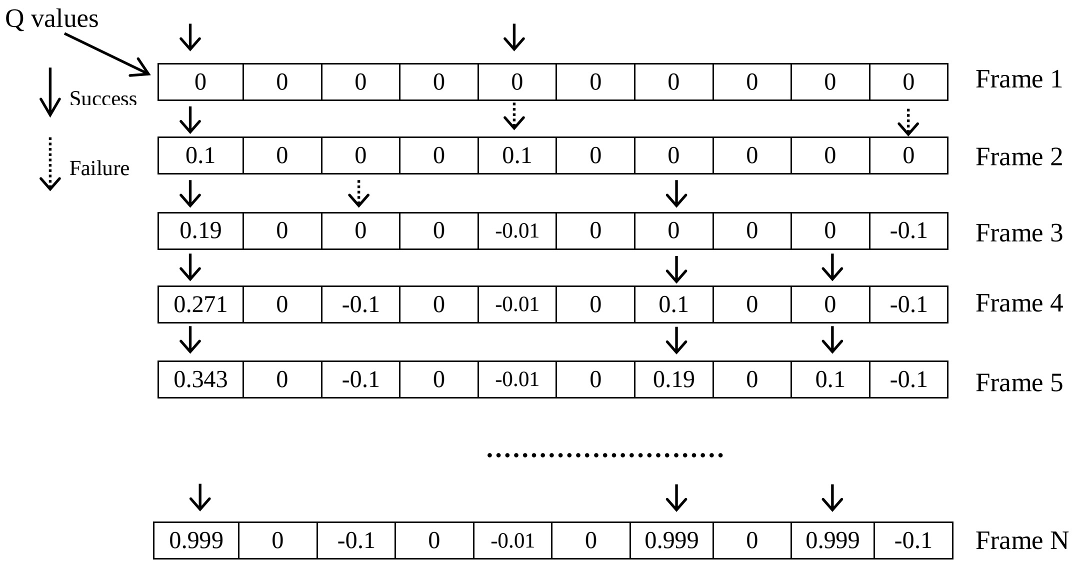
$$Q_{t+1}(i, s) = Q_t(i, s) + \alpha(R - Q_t(i, s)) \tag{6.1}$$

Where *i* indicates the present node, *s* is the preferred slot, *R* is the current reward and *α* is the learning rate. A transmitter will always choose the slot within a frame with the highest Q-value. If more than one slot has the same Q value, the transmitter randomly chooses one of them. If the packet transmission is successful, R takes a value of $r = +1$ which constitutes a reward. If the packet transmission fails, then R takes a punishment value of $p = -1$. Consequently, a sequence of successful transmissions using the same slot will cause the associated Q-value to increase, finally converging on a value very close to +1. There is no consensus on the choice of values for *r* and *p* but it has been shown in [65, 68] that +1 and -1 respectively produce convergence to +1 for successful slot choice and 0 for all other unchosen slots in that frame. We define this condition as being a steady state since for a particular slot; a node will always choose the high Q slot. The learning rate α is an important parameter which controls the speed of convergence. It determines the extent to which recently acquired information will be considered.

An illustrative example is now presented for updating the Q-values for 5 slots/frame in a single-hop scenario and 10 slots/frame in a multi-hop scenario. In Fig. 6.1a, the first packet is transmitted in slot 1 of the frame. This is randomly chosen as all Q values are equal to 0. As that packet was successfully received (as indicated by receipt upon acknowledgement), the Q value for slot 1 is incremented according to equation 1. However, in this case, the next packet transmission in slot 1 fails. The Q value falls immediately to -0.01 and another slot is selected randomly. In this scenario, slot 4 continues to be successful for the next N packets. It is seen that the Q value approaches a value of +1. It will be later shown that it takes many more successful transmissions for a Q value to approach +1 than it takes to reduce back towards 0 due to successive packet transmission failures. In fig. 6.1b, slot 1 and slot 5 of the frame 1 are randomly chosen for the first 2 packet transmissions because all Q values are equal to 0. As these packets were successfully transmitted, the Q-values for slot 1 and slot 5 are incremented according to Eq. (1). In frame 2, there are now 3 packets to be transmitted; slot 1 and slot 5 are certainly selected as they have higher Q-values and slot 10 is randomly selected among others. However, the transmissions in slot 5 and slot 10 fail and the associated Q-values fall immediately. In the next round, slot 1 will be the first preferred and two slots among the unchosen slots are selected at random. This process will continue to explore 3 slots which have Q-values approaching +1 for the next N packets. It will be later shown that it takes many more successful transmissions for Q value to approach +1 than it takes to reduce back towards 0 due to successive packet transmission failures.



(a) An example of single slot exploration.



(b) An example of multi slots selection.

Fig. 6.1 Example of the slot selection process, α = 0.1.

6.3 ALOHA-Q Performance Evaluation

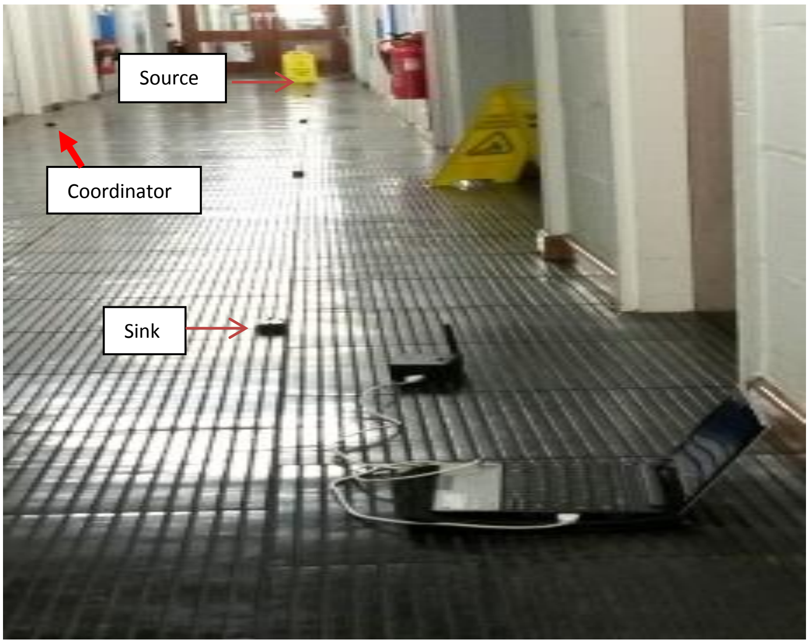
6.3.1 Experimental Setup

We use MicaZ nodes running TinyOS, IEEE 802.15.4-compliant devices, which feature an ATmega128L low-power microcontroller and a CC2420 radio transceiver, operating in the 2.4 GHz band. They have 4Kbytes of data memory, 128Kbytes of programmable flash and provided data rate is 250 kbits/s. In our implementation, the SHR, PHR, MHR, PAYLOAD and CRC fields comprise 5-bytes, 1-byte, 11-bytes, 114-bytes and 2-bytes respectively. The acknowledgement packet has a 5-byte SHR, a 1-byte PHR, an 11-byte MHR and a 2-byte CRC.

For the single-hop scenario, performance is evaluated for an indoor topology, as depicted in Fig. 6.2a, in an unobstructed area with line of sight communication which comprises 12 users. All the nodes are in range of each other, are equidistant from the receiver, and transmit at the same power level. Each node generates packets and sends them directly to the receiver. All the nodes have the same mean packet inter-arrival time which is exponentially distributed and synchronised by the receiver.

For a multi-hop scenario, performance is evaluated under a linear network topology comprising 5 nodes as presented in Fig. 6.2b. Here, packets are generated by node 1 (source) to be transferred through the line hop-by-hop to node 5 (sink). Each node transmits at the minimum transmission power level that allows them to receive the packets from only one-hop neighbours. The interference range is also one-hop (this is achieved by setting the transmission power to a low value). In order to synchronize the nodes at the onset of the repeating frames simultaneously, the *Hello* packet strategy (described in Section 5.3.1.1) is used via a *network coordinator* broadcasting at the maximum transmission power level which therefore covers all nodes.

(a) Single-hop scenario



(b) Multi-hop scenario

Fig. 6.2 Application environments for ALOHA-Q.

The theoretical maximum throughput of a single-hop scenario for the experimental parameters given in Table 6.1 is close to 0.85 Erlangs (1064 bits / 1250 bits). In a 5-hop scenario, with the transmission and interference ranges of one-hop, the optimum frame size is 3 slots. Therefore, the theoretical maximum throughput at the sink is 0.33 Erlangs.

Parameters	Values
Channel bit rate	250 Kbits/s
Data packet length(ALOHA-Q)	1064 bits
Data packet length(Z-MAC)	840 bits
ACK packet length (simulation)	20 bits
ACK packet length (experiment)	152 bits
Slot length (simulation)	1100 bits
Slot length (experiment)	1250 bits
Experiment Period	100,000 slots
Learning rate (α)	0.1

Table 6.1 Experiment parameters for ALOHA-Q.

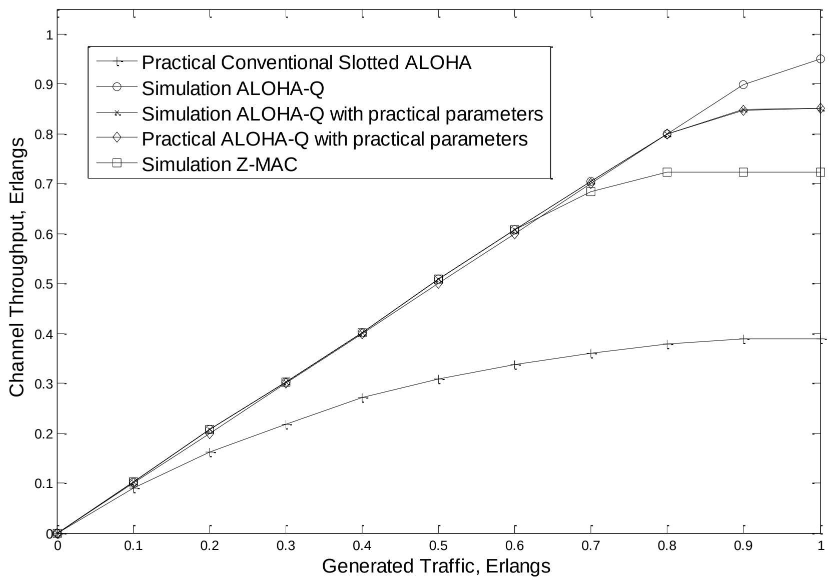
6.3.2 *Steady State Results*

In order to evaluate the performance of ALOHA-Q, we implemented it in both OPNET and MicaZ/TinyOS using the parameters given above. Also, the ideal performance of the ALOHA-Q and Z-MAC protocols is simulated based on a very small acknowledgement packet length. We also implement conventional slotted-ALOHA on the testbed. Fig. 6.3 presents the results of all these scenarios for varying traffic levels.

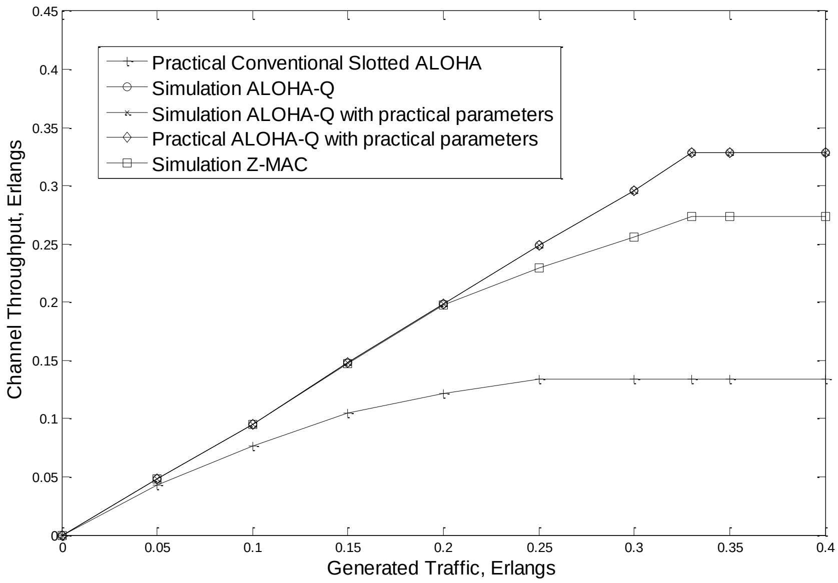
In the single-hop scenario, the simulation results show that the ideal throughput of ALOHA-Q increases linearly and reaches a maximum of 0.95 Erlangs which corresponds to every node finding a unique slot. The practical and simulation results of the throughput, using the same parameters as the practical system, exhibit a similar increasing trend but to a lower maximum throughput of approximately 0.85 Erlangs, since there is an ACK packet overhead of 0.15 Erlangs. Z-MAC achieves a lower maximum throughput than ALOHA-Q due to greater overheads and the potential for contention (nodes can potentially contend for the non-owned slots). On the other hand, the practical maximum throughput of the conventional slotted-ALOHA (nearly 0.39 Erlangs) with 12 users is slightly higher than the theoretical achievable throughput (0.368 Erlangs based on the assumption of an infinite number of nodes).

In the multi-hop scenario, the throughput using ALOHA-Q grows linearly and reaches its maximum limit. All the transmitted packets are transferred to the sink node in the steady state condition. Slotted-ALOHA can only provide 0.13 Erlangs throughput due to its inefficient transmission strategy. The throughput of the

ALOHA-Q stabilises at 0.33 Erlangs with increasing traffic levels as it depends on the frame size. Fig. 6.3b shows that ALOHA-Q achieves much higher throughput when the traffic load is heavy. This is because the large contention windows used for channel sensing limit the performance of Z-MAC.



(a) Single-hop scenario



(b) Multi-hop scenario

Fig. 6.3 Channel Throughput for ALOHA-Q.

6.4 Stability Properties of ALOHA-Q

6.4.1 Packet Loss in WSNs

Wireless sensor networks can have a reputation for unpredictable quality of wireless communication, since they are often fairly densely deployed in harsh, inaccessible environments. A lot of factors govern the performance of wireless communication. These focus around the environment, the network topology and the devices. We note that three important reasons for packet loss are multi-path interference, hardware architecture and change in network size.

Depending on the environmental characteristics, multi-path interference can occur which results in duplicate packets being received over small time differences that may result in their destruction. For instance, multi-path reflections caused by walls can produce significant interference. Sensor devices are constrained with low-power radios and energy cannot tolerate multi-path effects having insufficient frequency diversity [86].

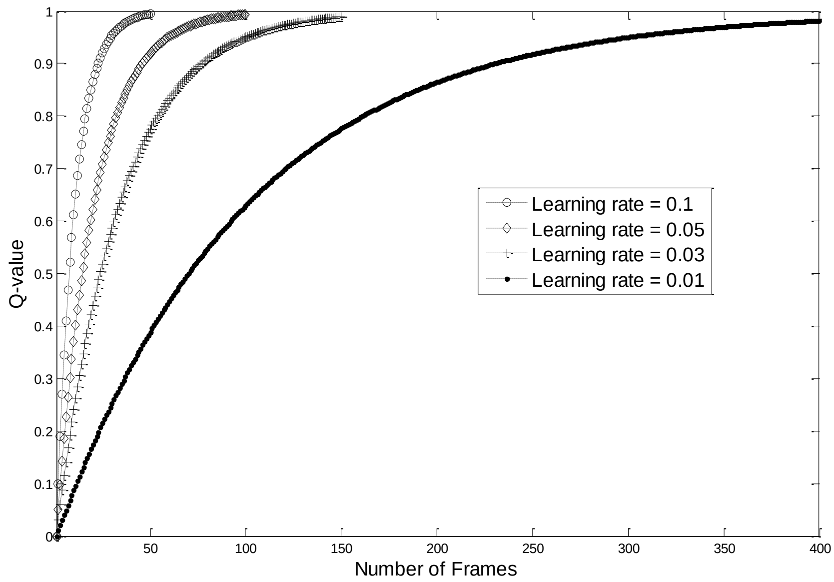
Our previous study (see Section 5.3.1.2) demonstrated that a typical popular sensor node (the IRIS node) cannot effectively operate under high traffic loads, as it cannot switch quickly from reception mode to transmission mode to send back acknowledgement packets. Consequently, depending on the traffic load level, a certain proportion of the acknowledgement packets may not be sent. Hence, even though a packet is received successfully, the transmitting node will assume it to be lost as no acknowledgement packet is received. To overcome this problem, employing a guard period between the transmission and reception modes is proposed, but this wastes channel resources. It is therefore concluded that there might always be a possibility of losing some packets in WSNs because of the sensor hardware and this may not be predictable.

An important desirable attribute of MAC protocols is scalability with network size; some new nodes may need to be deployed later on. A good MAC scheme must comfortably meet such change. However, during the addition of new nodes to the network, some packets might be lost due to the arrangement of new transmission schedules. Depending on the application, this process may need to protect existing users' current schedules.

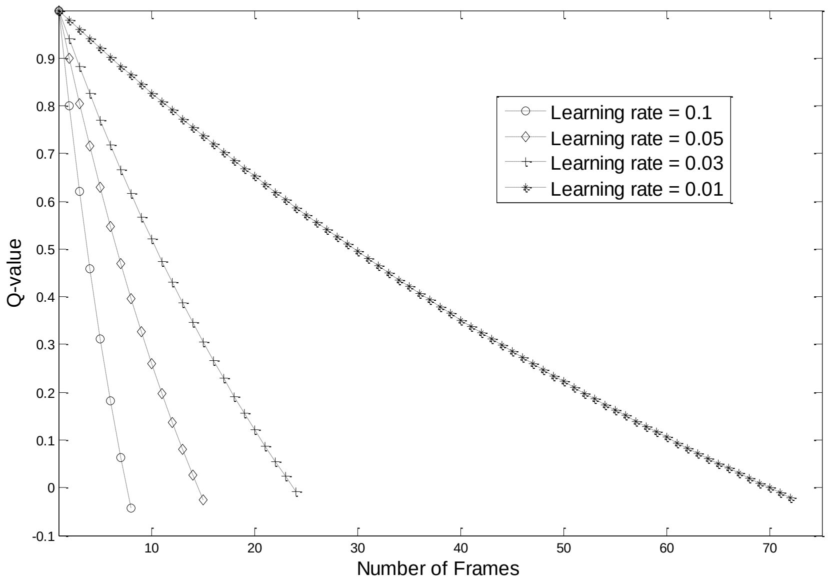
6.4.2 ALOHA-Q Level of Resilience to Loss of Convergence

Although ALOHA-Q provides perfect scheduling, allowing no packet loss due to collision after convergence, as validated in simulation and in a real-world testbed, packet loss can still occur in practice due to the reasons described above. This section systematically analyses the level of resilience to loss of convergence in the presence of packet loss. The learning rate (α) is an important parameter as it has a significant effect on the Q value updates. Therefore, various learning rates are simulated to demonstrate the behaviour of the Q-value of a single node unique slot as shown in Fig 6.4. Establishing the best case where all the packets are successful in a particular slot from the initialisation of the system is very important to deeper understanding of the behaviour of the Q-value updates. Fig. 6.4a shows the Q-values of a slot with consecutive successful transmissions.

It is seen that the learning rate determines, as expected according to equation 6.1, the accrual of the Q-value. Smaller values result in a longer time to converge to the Q-value of 1. The numbers of consecutive successful transmissions required to achieve convergence for a single user with respect to the learning rates of 0.1, 0.05, 0.03 and 0.01 are 50, 100, 150 and 400 respectively. However, as the negative reward has more impact on the Q-value when the Q-value is positive, the number of successive failures required to result in the Q-value reducing to 0 is therefore significantly fewer. It is assumed that the rest of the Q-values are set to 0 after convergence so that once the Q-value of a unique slot falls back to 0, it will lead the associated user to seek to find a new slot. It is seen from Fig. 6.4b that only 7 consecutive failures cause Q-value to return to 0 (loss of the convergence) at a learning rate of 0.1. Considering that in the real-world, there will be packet loss, the risk of this rapid decline of Q value is significant, leading to loss of convergence and subsequent quality of service. The system would not have a good level of robustness and would not be protected from infrequent collisions and small changes in the environment and channel conditions.



(a) Best case; all the packets successful.



(b) Worst case; all the packets lost.

Fig. 6.4 Behaviour of the Q-value update.

6.4.3 Markov Model

A Markov model is now derived to represent the behavior of the system after convergence. Each node has a unique slot and the Q-value of this slot is very close to 1 where the rest of the slots have Q-values of 0. Each state represents a particular Q-value. Given that Q-value increments based on successful transmissions are much smaller than Q-value decrements based on failed transmissions. Upward transitions between neighboring states correspond to a single success. A single failure results in downward transition across multiple states. The total number of states therefore depends only on the learning rate. After a transmission, the Q-value is updated and a state transition occurs. If a transmission succeeds, the process moves forward. If not, the process moves backward and chooses the state which has the closest Q-value. An example of the Markov model is given in Fig. 6.5 for a learning rate of 0.1. There are 50 states which are required for convergence as previously noted for a learning rate of 0.1.

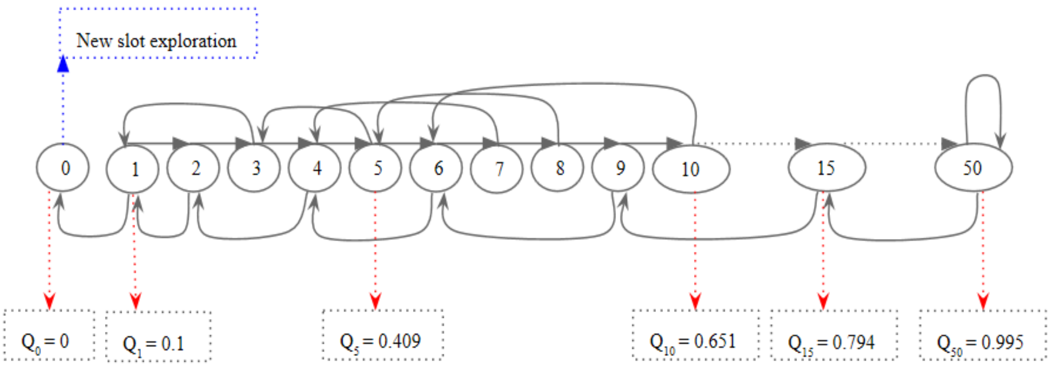


Fig. 6.5 Markov model, learning rate of 0.1.

Let p denote the probability of successful packet transmission, based upon the factors previously outlined. This will be probability of moving forward, $p_{k,\ k+1}$, k = 0, 1, 2…49. It will also be the probability of staying in the last state, $p_{50,\ 50}$. The probability of moving backwards, $p_{k,\ l}$, will be (1-p) where l is the corresponding state after an unsuccessful transmission (e.g. k=15, l=9). These state transition probabilities can be formulated as:

$$p_{k,k+1} = p \tag{6.2}$$

$$p_{k,l} = 1 - p \tag{6.3}$$

6.4.4 Loss of Convergence Time Estimation

In a practical deployment, packet loss can occur at different rates. We will not necessarily observe sustained sequence of consecutive failures. Therefore, the relationship between packet loss ratio and the time to lose convergence is an important one to establish. The approach presented in [87] for ALOHA-Q provides the convergence time of a whole network through an analytical model. In this model, a state transition probability matrix, ***P*** which is a sparse matrix, is considered. Using the notation $\boldsymbol{P^2}$ to denote the multiplication of ***P*** by itself, the elements of $\boldsymbol{P^2}$ are:

$$p_{\mathrm{i,j}}{}^{(2)} = \sum_{m=0}^{N} p_{\mathrm{i,m}}\, p_{\mathrm{m,j}} \tag{6.4}$$

$P_{\mathrm{i,j}}{}^{(2)}$ represents the transition probability from state *i* to state *j* in one transition. Similarly, the elements in $\boldsymbol{P^3}$ are:

$$p_{\mathrm{i,j}}{}^{(3)} = \sum_{m=0}^{N} p_{\mathrm{i,m}}{}^{(2)}\, p_{\mathrm{m,j}} \tag{6.5}$$

which is the probability from state *i* to state *j* via all possible states after two transitions. $\boldsymbol{P^n}$ is referred as the matrix of state transition probabilities after *n* transitions (n slots), so that $p_{i,j}{}^{(n)}$ is the probability from state *i* to state *j* after n transitions. In order to calculate the time of the convergence loss, we need the expected number of transitions (slots) from the last state to achieve to all states except state 0 which is obtained as:

$$E\{convergence\ loss\ time\} = \sum_{n=1}^{\infty} \sum_{j=\mathrm{N}}^{1} p_{\mathrm{N,j}}{}^{(\mathrm{n})} \tag{6.6}$$

It is the expected convergence loss time starting with state N. The detailed derivation and proof of the model can be found in [87].

Using our Markov model based simulation, for a given learning rate, the total number of states required to the convergence is initially calculated. Then, the Q-value of each state is calculated and state transitions, up or down, are determined. By using uniformly distributed random number generation, different packet loss ratios are artificially created. Here, a number is randomly generated in the range from 0 to 1. This is then compared with a predefined threshold that is determined to create a particular packet loss rate. If the number is greater than the threshold, the process moves forward, otherwise it moves backward. To create 40% of loss rate, for instance, the threshold is set at 0.4. The simulation is initialized in the final state as the system is assumed to have converged, following state transitions are undertaken. The process is stopped when the process has reached to the state 0. The required number of iterations, which is equivalent to the number of frames, is then recorded. The simulation is run 100 times and the average is presented.

Using the test-bed described previously for single-hop scenario, the time (number of frames) to lose convergence for a particular node is observed, for a given packet loss rate. The receiver sends a certain amount of acknowledgement packets using the random number generation strategy described above. In this case, some of the packet receptions will not be acknowledged, despite these packets being received in success. At the start of the trial, each node learns a unique slot. When a node then tries to change that slot (this is referred to as convergence loss), it sends a message containing the number of frames taken from the beginning to a base station, this is connected to a computer to monitor data packets. The implementation is again run 100 times and the average of them is presented.

Fig. 6.6 presents the time (number of frames) before convergence loss with different probability of failure. Three results are presented: (1) the simulation of the Markov model described above, (2) theoretical convergence loss time obtained from the equation 6.6, and (3) the test-bed results. It is seen that practical result of convergence loss time matches the result of the Markov model and the simulation. The convergence can be lost within 100 frames below to the probability failure level of 0.3, whereas the node lost convergence within 600 frames at the probability fail level of 0.2. However, below a level of 0.1, convergence is never lost. Here, the state 0 is never achieved in the simulation of the Markov model and no packet is lost at the receiver in practice. Therefore, to provide efficient operation of the protocol, the probability of failure must be less than 0.1 which will be referred as *convergence loss point (CLP)*. The simulation has been run for sufficiently long time (theoretically infinite as seen in equation 6.6).

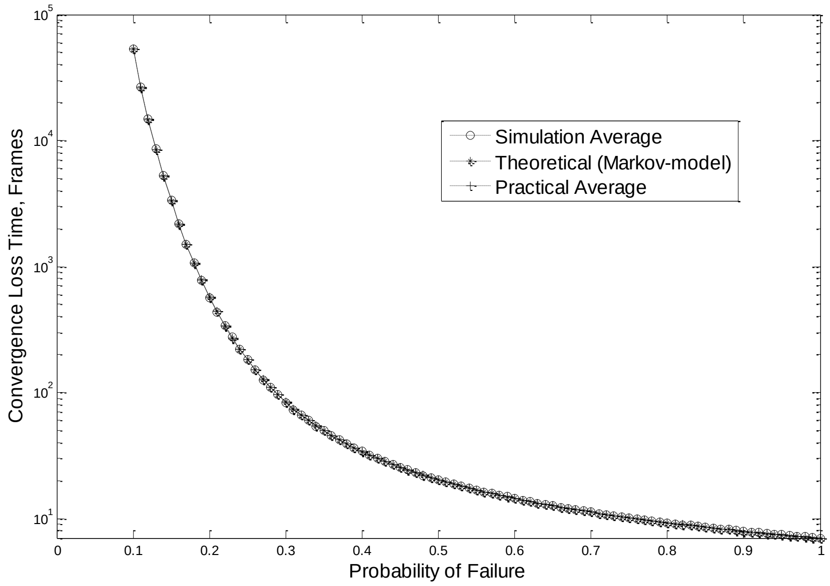
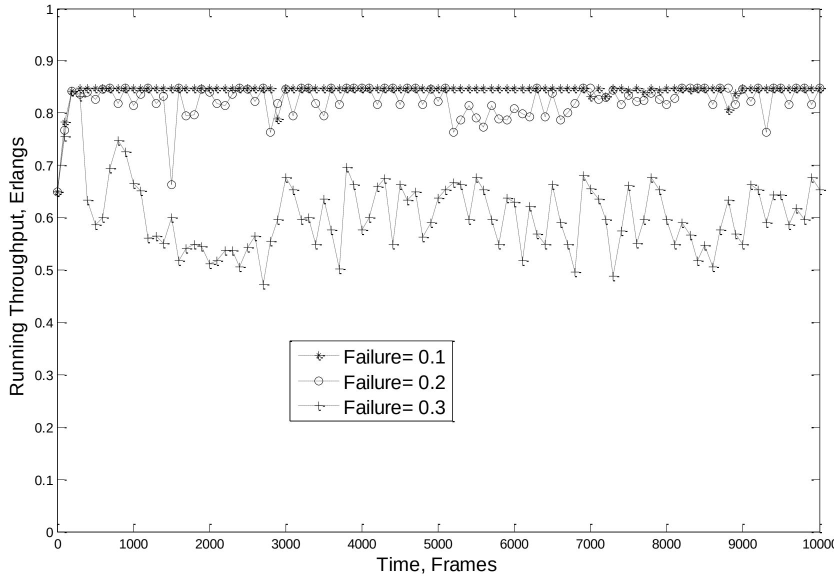


Fig. 6.6 Average time of convergence loss.

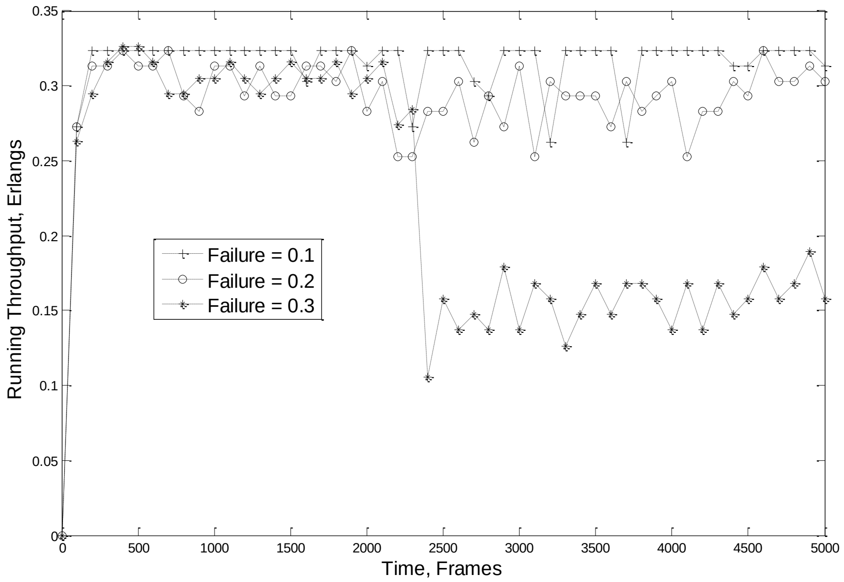
6.5 Performance Evaluation of ALOHA-Q with Loss of Convergence

6.5.1 Performance Results

In order to understand how the packet loss impacts on the throughput performance of ALOHA-Q, given that the system is in steady state, three packet loss ratios are created at the receiver. To allow the system to achieve steady state without acknowledgement packet loss, the receiver acknowledges all the successful receptions at the beginning. After convergence, a certain amount of acknowledgement packets are not sent upon successful receptions. Fig 6.7 presents the running throughput of the 2 scenarios as described in section 6.3.1.



(a) Running throughput, single-hop.



(b) Running throughput, multi-hop.

Fig. 6.7 System behaviour against packet loss running on MicaZ nodes.

The running throughput is obtained from initialisation through each time step (10 frames for single-hop and 100 frames for multi-hop) in which each curve represents an average of 100 runs. In both topologies, the users lose convergence rarely with a failure rate of 0.1. However, the real-time running throughput decreases faster with increasing packet loss rates because users lose convergence more frequently. The fluctuations on the throughput performance are due to the artificial creation of packet loss through not sending acknowledgement packets for some packets being received in success.

6.5.2 Reformulation of the Punishment Magnitude

It has been found that the convergence would not be lost if the packet loss rate does not exceed 10%. The main objective is to maximize the CLP to protect the convergence from an instant unknown or a long-term change. According to equation 6.1, the punishment value, assuming a fixed learning rate, plays an important role in updating the Q-value. In particular, use of a fixed punishment value (-1) reduces the Q-value more quickly when the Q-value is positive. It is therefore clear that use of a reformulated numerical value of the punishment can serve to protect the convergence loss. It is intended to dynamically change the magnitude of the punishment when a packet loss occurs after convergence. This will result in the Q-value reducing more slowly.

As pointed out previously, the number of consecutive failures required to lose the convergence is 7, whereas the number of consecutive successes required to achieve to the convergence is 50. It is proposed to equate this imbalance by updating the punishment value when a packet transmission fails. In this case, 50 consecutive failures will cause the loss of convergence. After a packet failure, the punishment value is re-calculated to update the Q-value, so that the process will take the previous state in the Markov model. Let us consider the 2 neighbouring states to demonstrate the modification of the punishment value as depicted in Fig 6.8.

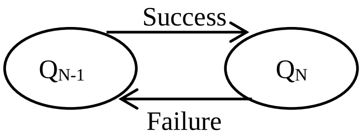


Fig. 6.8 2 neighbouring states in the Markov model.

Here, Q_{N-1} represents the Q-value of the state *N-1* and the Q_N is the Q-value of the state *N*. If the packet transmission fails when the process is in the state *N*, the new Q-value will be Q_{N-1}. If the packet transmission succeeds when the process is in the state *N-1*, the new Q-value will be Q_N.

$$Q_{N-1} = Q_N + \alpha(p - Q_N) \tag{6.7}$$

$$Q_N = Q_{N-1} + \alpha(r - Q_{N-1}) \tag{6.8}$$

Then we substitute (7) in (8) to obtain the new punishment value which will take the process to the previous state:

$$p = \frac{Q_N\,(2-\alpha) - r}{1-\alpha} \tag{6.9}$$

Which is the new punishment equation based on the current Q-value. Therefore, after an unsuccessful transmission, the punishment value is calculated and then the Q-value is updated.

Similar to the results shown in Fig. 6.6, the results of the new punishment scheme are obtained from the Markov model and the simulation which is presented in fig 6.9. The Markov model results match the average results of the simulation. It can be clearly seen that our scheme achieves better results, improving the time of convergence loss. The CLP is now 0.47 meaning that the network operates in steady-state condition where users keep transmitting data packets in their unique transmission slots as long as the around half of the packets are successfully received.

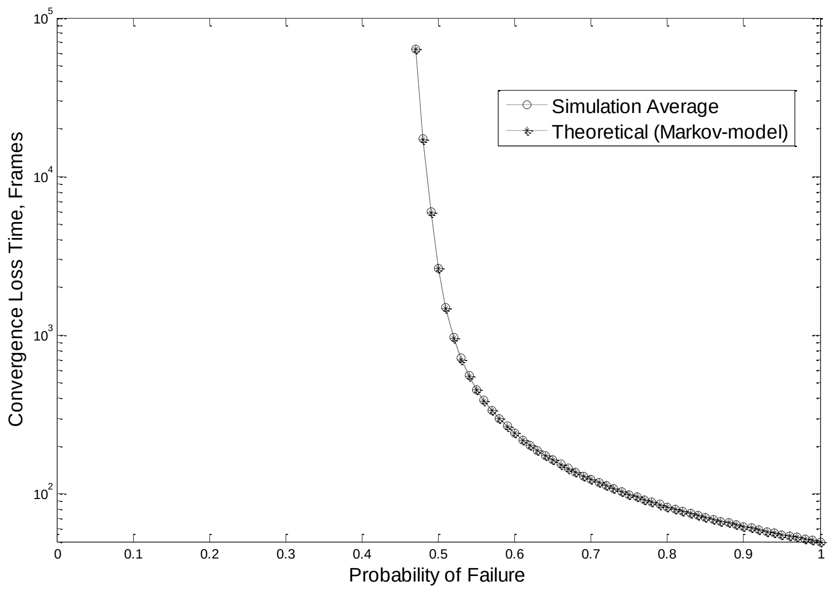
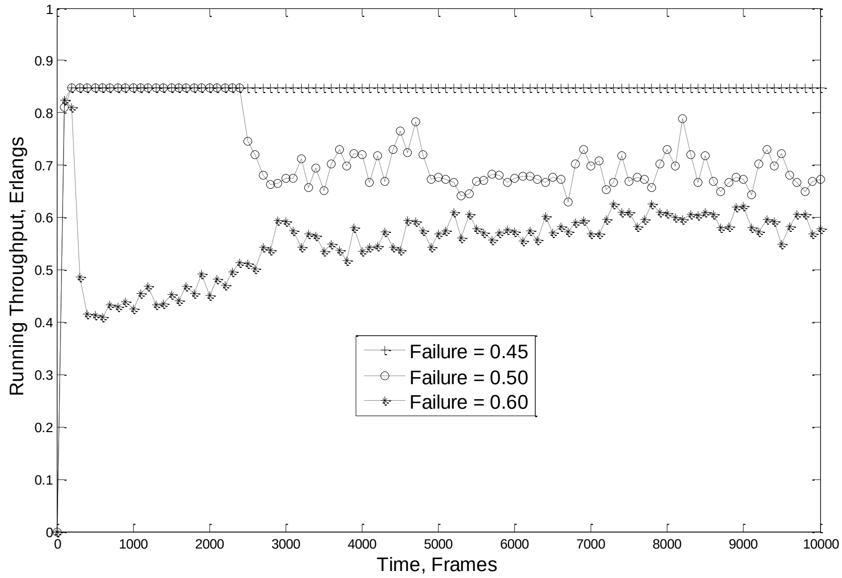


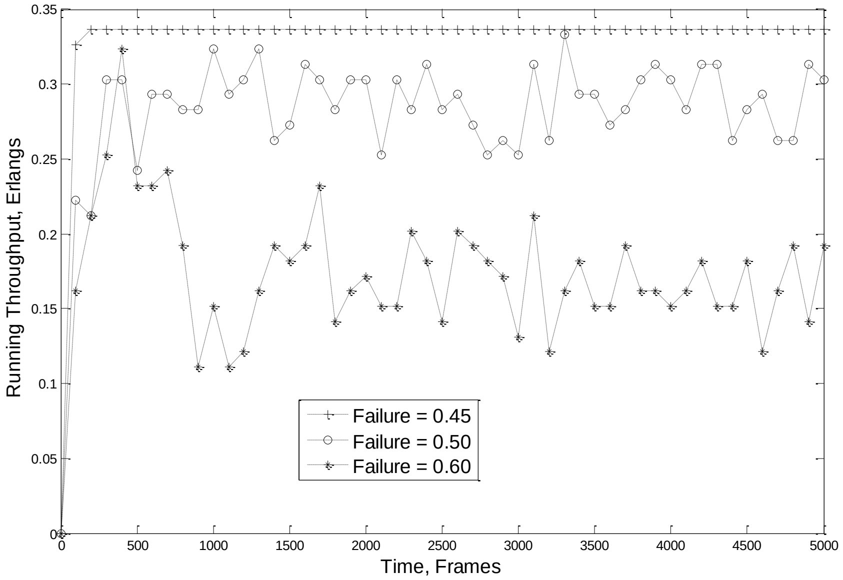
Fig. 6.9 Average time of convergence loss with the new scheme.

6.5.3 Implementation of the Proposed Punishment Modification

Here, the analytical results are experimentally validated, through implementing the proposed scheme in the test-bed. Again, the running throughput is evaluated for 3 loss rates with convergence already having been achieved prior to the start of the test. The practical results show that the system does not lose convergence beyond the loss rate of 0.47. However, as the loss rate increases, the system loses convergence quickly. All of these results are presented in Fig. 6.10.



(a) Running Throughput, single-hop.



(b) Running Throughput, multi-hop.

Fig. 6.10 Overall system behaviour against packet loss.

6.6 Summary

This chapter has thoroughly analysed the stability properties of a recently-proposed, energy-efficient MAC protocol for single-hop and multi-hop communication, named ALOHA-Q combining slotted-ALOHA with its benefits of simplicity and low computation, and Q-Learning, providing an intelligent slot selection strategy. Starting with the practical implementation issues of ALOHA-Q provided a perfect scheduling in steady state that is rapidly achieved. It is then shown that ALOHA-Q is prone to the loss of convergence in the presence of packet loss due to the changes in the environment and radio conditions. A Markov model to represent the behaviour of a user has been provided and used to estimate the time taken to lose the convergence. It has been shown through the Markov model and the test-bed that the convergence can be quickly lost because of the high punishment level. A novel punishment technique has been proposed to deal with low packet failure in order to protect the operation of the network. The proposed scheme serves to protect the lifetime of the convergence by dynamically adjusting the punishment level.

7. ALOHA-Q for Practical Multi-hop WSNs

Contents

This chapter extends the implementation of ALOHA-Q in both simulation and realistic test-beds, to linear-chain, grid and random topologies. Practical implementation issues of ALOHA-Q are studied based upon hardware limitations and constraints. The performance of ALOHA-Q is simulated in comparison to ZMAC and is validated with practical results. In order to strengthen the merits of the ALOHA-Q against dynamic channel and environment conditions, the epsilon-greedy strategy is integrated.

7.1 Introduction

Chapter 6 evaluated the ALOHA-Q protocol based on a single-hop and a linear chain topology and studied the stability issues against packet loss in steady state conditions. These two topologies are simple and the key properties of ALOHA-Q are tested. A typical WSN covers a large area which is usually built around multi-hop communication. Therefore, the scale of the experiments in chapter 5 is small and more extensive experiments are highly important, in order to demonstrate the applicability level of ALOHA-Q on a larger scale. The performance of ALOHA-Q is tested on three multi-hop topologies which are commonly seen in the performance evaluation of existing MAC protocols.

In reinforcement learning strategies, is the knowledge gained to a point in time enough to exploit, or when should an agent start exploiting its current knowledge?. This is one of the main challenges. In parallel with this issue, the next section discusses the exploration and exploitation in the scope of MAC protocol design. A few of the existing *exploration/exploitation* techniques which can be applied to MAC protocol design are described in details. Based on these techniques, two new approaches are incorporated into ALOHA-Q which achieves a good balance between the exploration and exploitation for real-world implementations. To decide on the exploration levels in the transmission slots, the Q-values of the slots are used as the transmission history is recorded by Q-values. The performance of ALOHA-Q is experimentally evaluated in two real-world events: (1) packet losses in the steady state and (2) participation of new nodes in the network. ALOHA-Q with *exploration/exploitation* is implemented to provide better performance in the presence of these two events.

7.2 Exploration and Exploitation

7.2.1 Trade-off between Exploration and Exploitation

A reinforcement learning agent needs to explicitly explore its environment in order to find the best action. Exploitation is used to capitalise on the experience already available. The balance between the exploration and exploitation is a fundamental issue in reinforcement learning, and has long been an important challenge. Exploration is, of course, particularly crucial when the environment is non-stationary. Therefore, an agent must adapt to environmental changes.

Several methods have been proposed to ensure a good balance between exploration and exploitation. This study will focus only on the methods used in the context of ad-hoc techniques. The first strategy is *greedy* selection. The agent always chooses the action with the highest action-value as in ALOHA-Q. This is, however, not an efficient exploration method because it does not control the exploration time. Hence, it might take too long to adapt to any change. A variation on the *greedy* approach is *Є-greedy* selection. In this method, at each time step the agent generates a random value between 0 and 1 and if this value is smaller than a fixed value $0 < \epsilon < 1$, the agent explores, otherwise the agent exploits. Therefore, the duration of exploration is controlled by the ϵ value. In exploration, the agent randomly selects an action. A derivative of *Є-greedy* for mainly selecting the ϵ value is the *decreasing-Є* method.

Basically, ϵ is set to a high value at the beginning to allow more exploration and it is reduced at each time step for more exploitation.

As stated above, ALOHA-Q naturally uses the *greedy* method in which the exploration and exploitation are always performed together, since the slot with the highest Q-value is preferred and the Q-value of this slot is immediately updated after transmission. It can be clearly seen from the Eq. (6.1) that the Q-value increments based on successful transmissions are extremely small after it converges close to 1 as the Q-value has a limit of one. However, in this case, a single failure results in a significant decrement of the Q-value. More specifically as discussed in section 6.4.2, only seven consecutive failures cause a Q-value to return to 0 at a learning rate of 0.1, as the punishment has more impact on the Q-value when the Q-value is positive. We see that a few failures would lead the associated user seeking to find a new slot despite the possibility of thousands or even millions of successful transmissions previously. Therefore, the Q-learning algorithm actually considers the short-term channel history in order to obtain enough knowledge. This inherent tradeoff can have a significant influence on network performance in dynamic channel and environment conditions. In the previous studies of ALOHA-Q, this issue has not arisen due to the simulations being carried out in a static environment and with time-invariant channel conditions.

7.2.2 ALOHA-Q with ϵ-greedy: ALOHA-Q-EPS

Using the *ϵ-greedy* policy, nodes select a transmission slot with the highest Q-value with probability 1- ϵ and select a random slot with probability ϵ. The main drawback is that the selection of ϵ value is unclear. An epsilon value of 1.0 will result in exploring constantly, while a value of 0.0 will result in always exploiting. Therefore, the ϵ value is ideally set to a small value in order to allow more exploitation while providing sufficient exploration. In all experiments of this work, it is typically set to 0.1. Also, during exploration, an equal probability is given for the non-ideal slots to be selected which may have low Q-values. Due to the constant value of ϵ, the transmissions in randomly selected slots in the steady-state can cause collisions which may reduce the maximum achievable throughput. The ALOHA-Q protocol with *ϵ-greedy* is called the ALOHA-Q-EPS.

7.2.3 ALOHA-Q with decreasing-ϵ greedy: ALOHA-Q-DEPS

The efficiency of knowledge obtained is represented by Q-values which also represent the goodness of the exploration level. Intelligent control of the tradeoff between exploration and exploitation is investigated with respect to the behaviour of the Q-value. A *decreasing-ϵ* method is developed to allow nodes to explore more until they achieve a certain level of exploration. The duration of ϵ is controlled by Q-value as shown in Eq. (2). Basically after convergence, the Q-value is updated after transmission if in exploration, but it is not updated after exploitation. As the Q-value of a slot increases, the exploitation in this slot occurs more frequently and vice versa. In ALOHA-Q, the term *convergence* in a slot occurs when the Q-value of this slot approaches to one. However, this would not allow exploration in this strategy after convergence. To solve this problem, we define $Q_{convergence}$ in which the slots will be accepted as converged when the Q-values of them exceed $Q_{convergence}$. This will decide the ratio of the exploration after convergence. The value of $Q_{convergence}$ depends on the application scenario such as topology, density of the nodes and environment conditions.

$$\epsilon = \begin{cases} 1 - Q_{value} & \text{before convergence} \\ 1 - Q_{convergence} & \text{after convergence} \end{cases} \qquad (7.1)$$

We denote this modification with a new protocol name, ALOHA-Q-DEPS. The benefits of the ALOHA-Q-DEPS will be practically presented in the following section. The value of $Q_{convergence}$ is set to 0.9, so that the nodes will explore 10% of the time after convergence. In exploration, the slots are randomly chosen as in ϵ-*greedy*. However, this random selection in the steady state is not efficient as the packets with random slots would potentially collide with others' unique slots. Therefore, the nodes reasonably select the slots with higher Q-values in exploration after convergence.

7.3 ALOHA-Q Performance Evaluation

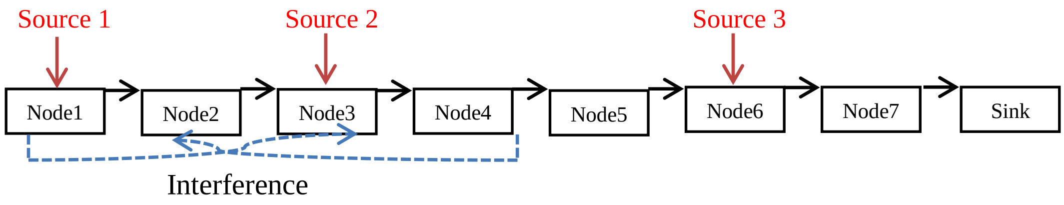
In order to evaluate the performance of ALOHA-Q in comparison to that of Z-MAC, several simulations have been carried out under three main topologies which are described in detail in the following sections; *linear-chain, grid and random*. The simulation results are validated experimentally using MicaZ and IRIS nodes running TinyOS. All the experiments were conducted in an unobstructed area with line of sight. To enable multi-hop networking, each node reduces the transmission power level to a low value. In all simulations, the default values of Z-MAC (8 contention slots for slot owners and an extra 32 contention slots for non-owners) are used. Table 7.1 summarises the simulation and experimental parameters. It is worth noting that the length of ACK packets and the slots are larger than in practice because of the preamble, synchronisation, header and CRC bits sent from the radio chip. ALOHA-Q and ALOHA-Q-DEPS will have the same throughput performance in the steady state. Therefore, we run ALOHA-Q in simulations. In all practical experiments, saturated traffic conditions are considered where each node always has data to transmit.

Parameters	Values
Channel bit rate	250 Kbits/s
Data packet length(ALOHA-Q)	1024 bits
Data packet length(Z-MAC)	840 bits
ACK packet length (simulation)	20 bits
ACK packet length (experiment)	144 bits
Slot length (simulation)	1050 bits
Slot length (experiment)	1200 bits
Experiment Period	500,000 slots
Learning rate (α)	0.1

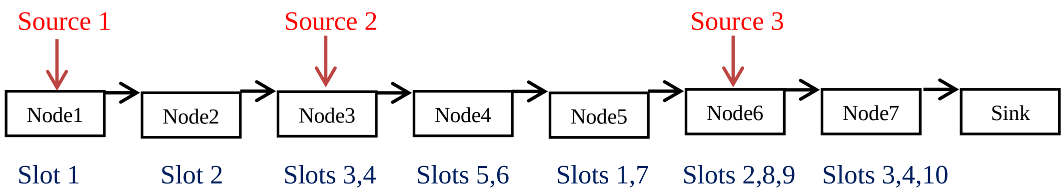
Table 7.1 Experiment parameters for multi-hop ALOHA-Q implementations.

7.3.1 Linear Chain Network

A linear network topology is created with 8 nodes lined up hop by hop, where the sink node is placed at the end of the chain as depicted in fig. 7.1a. The packets are generated at the source nodes and forwarded to the sink by the intermediate nodes. The number of source nodes can be varied up to the number of nodes in the network (excluding the sink node). For our implementation, networks with 1-source, 2-sources, 3-sources and all-source topologies are evaluated respectively. According to the practical observations, packet transmission can be successful within a one-hop neighbourhood (transmission range of 1-hop) but can be interfered with within a 2-hop neighbourhood (interference range of 2-hops). Therefore, four neighbour nodes have to select different transmission slots to avoid collisions. It is very difficult to get a sharp boundary of interference range in real-world environments, but this model is commonly employed as a baseline with which to understand and compare protocol performance. The frame size should be appropriately set in order to allow every node in the network to find a unique slot which can be theoretically calculated with respect to the network topology. For a 1-source topology (node 1), the optimum frame size is 4 slots/frame, whereas it is 7 slots/frame for 2-source topology (node1 and node5) as the intermediate nodes along the source 2 route receive a packet and transmit 2 packets in a frame. The 3-source topology requires 10 slots/frame as source 2 receives a packet and sends 2 packets in a frame and source 3 receives 2 packets and transfers 3 packets in a frame. When all nodes act as sources and intermediate relays, 22 slots per frame is estimated to be sufficient. To understand how the slots are allocated among the nodes, an example slot allocation calculation for 3-source topology is presented in fig. 7.1b.



(a) 7-hop network topology and source nodes



(a) An example slot allocation for 3-source topology.

Fig. 7.1 Linear chain network with multiple source nodes

Fig. 7.2 demonstrates the channel performance of ALOHA-Q and Z-MAC in steady state. The channel throughput exhibits an increasing trend and achieves its maximum value for all scenarios. The throughput stabilises at its maximum when the source nodes generate more traffic, because the traffic passing through the chain is limited by the frame structure. For example, the maximum throughput in the all-source scenario is 0.318 Erlangs (7 packets / 22 slots) as the sink node can receive seven packets at most in a frame. When the traffic load is low, Z-MAC achieves similar throughput performance. However, under a high level of contention, Z-MAC achieves a lower maximum throughput than ALOHA-Q due to greater overheads and the potential for contention (nodes can potentially contend for their non-owned slots).

Fig. 7.3 shows in all scenarios that the maximum throughput is achieved when the frame size is set to its optimum value as estimated above. As the frame size increases beyond its optimum value, the channel throughput reduces because some slots are unused after each node finds a unique slot in the frame. The behaviour of the channel throughput dependent on the frame size is observed.

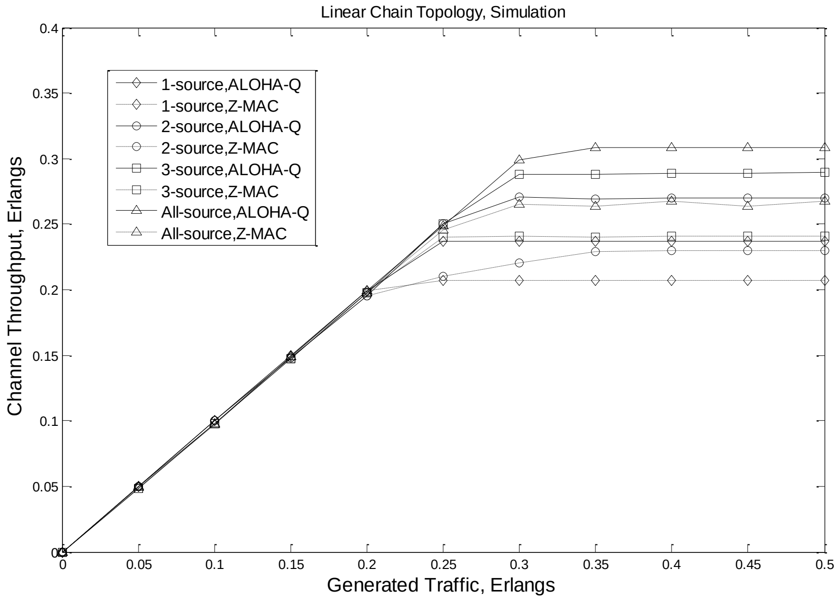


Fig. 7.2 Throughput comparisons for linear chain network

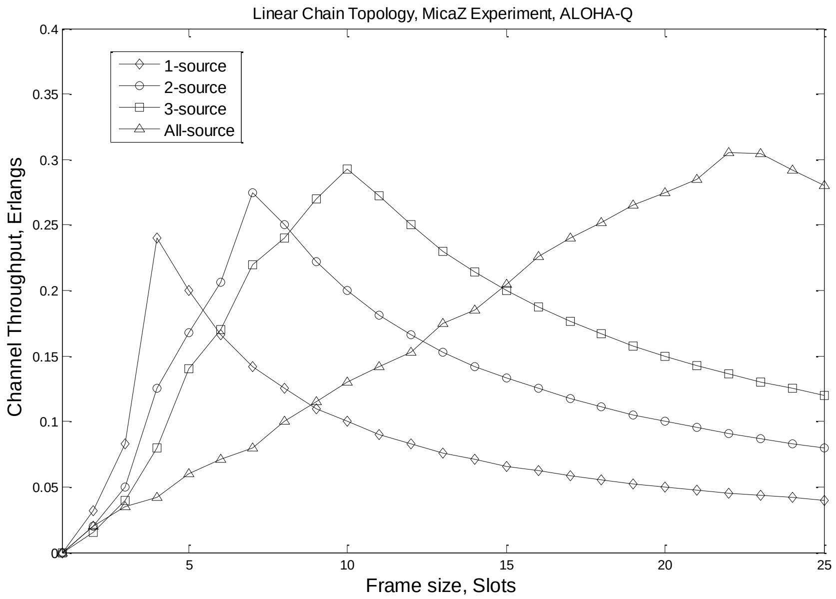


Fig. 7.3 Practical experiments of the channel throughput for linear chain network.

7.3.2 Grid Network

A 12-node grid topology is considered as presented in fig. 7.4. Each node acts as a source, aiming to deliver all the generated and the received packets to the sink. Each node routes the packets using the shortest path. In all implementations, fixed routing paths as shown in fig. 7.4 are used and as in the chain topology, all nodes are always assumed to have packet to send in practical implementation.

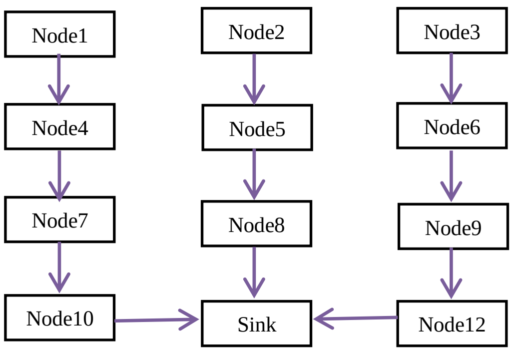


Fig. 7.4 Grid topology and the routing paths.

ALOHA-Q and Z-MAC have been simulated with various frame sizes to compare their performance in terms of channel throughput. Fig. 7.5 shows that ALOHA-Q achieves much higher throughput when the traffic load is heavy. This is because the large contention windows used for channel sensing limit the performance of Z-MAC. The performance of ALOHA-Q and Z-MAC is almost identical at lower traffic loads. In all cases, the channel throughput of both schemes grows linearly with increasing traffic and reaches its maximum.

In simulation, ALOHA-Q with the frame size of 20 slots/frame is the optimum and has a maximum throughput of approximately 0.5 Erlangs. However, practical experiments as presented in fig. 7.6 indicate that the maximum throughput can be achieved at a frame size of 23. The reason is the irregularity of the interference range in practice, so that practical observations experience lower throughput, but the system nonetheless can achieve the steady state if frame size is set 23 slots/frame.

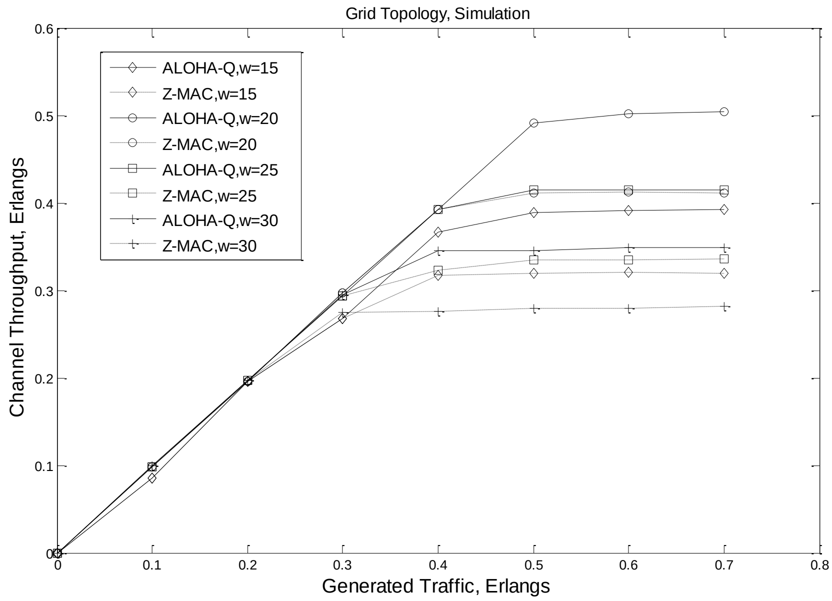


Fig. 7.5 Throughput comparisons for grid network.

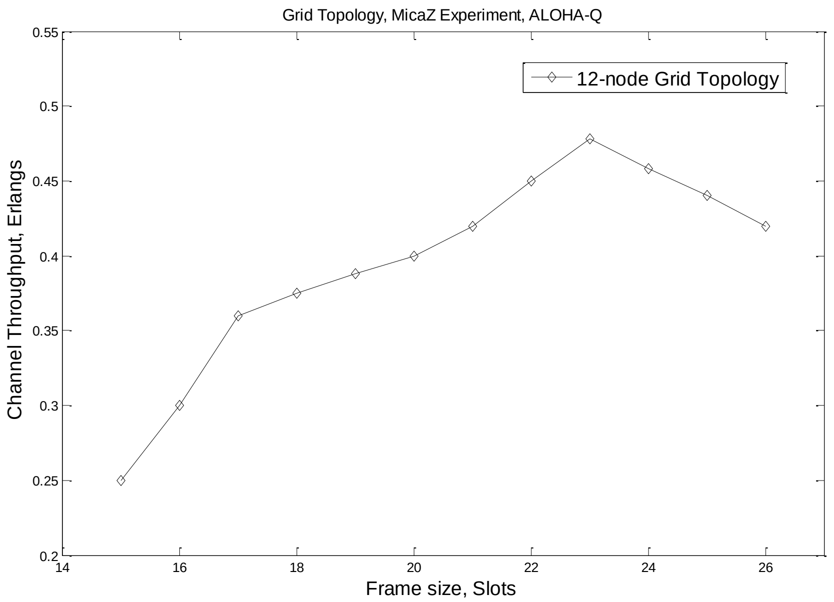


Fig. 7.6 Practical experiments of the channel throughput for grid network.

7.3.3 Random Network

A fully-distributed network of 21 nodes as a more realistic deployment is constructed as presented in fig. 7.7. Shortest path is used for routing, and fixed routing paths are set in the nodes. The sink is located in the leftmost-bottom of the network. All nodes generate traffic and also operate as relay nodes.

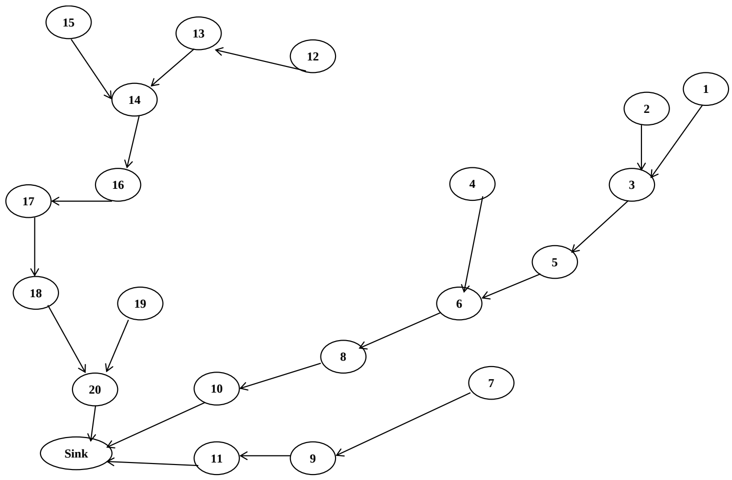


Fig. 7.7 Random topology.

We evaluate and compare the performance of ALOHA-Q with the Z-MAC protocol with different frame sizes and increasing traffic levels in terms of the channel throughput, delay and energy-efficiency. Based on the simulation results presented in fig. 7.8, it can be seen that the throughput of the two schemes grows linearly with increasing traffic load and converges to different limits. In all cases, ALOHA-Q has the same or superior channel throughput to Z-MAC at the same offered traffic level. The throughput performance of the both schemes depends on the frame size (w). ALOHA-Q with 40 slots/frame achieves the steady state with a maximum throughput of 0.48 Erlangs which is close to the theoretical limit of 0.5 Erlangs (20 packets / 40 slots) because the sink node can receive 20 packets at maximum in a frame because each node is allowed to transmit only one packet that is generated by itself. The performance of ALOHA-Q with 50 slots/frame exhibits a similar increasing trend but at a lower maximum because of the overestimate in the frame size. Z-MAC provides similar throughput performance at low traffic loads, but lower throughput due to the high overheads at high traffic loads.

Fig. 9 shows the energy efficiency results of ALOHA-Q and ZMAC. In this figure, ALOHA-Q provides better energy consumption performance than Z-MAC. This is because Z-MAC uses clear channel assessment (CCA) in every slot when a packet is ready for transmission and some control messages such as synchronisation are periodically transmitted. ALOHA-Q with 40 and 50 slots/frame has a similar energy cost per bit because nodes avoid all collisions and achieve perfect scheduling.

Fig. 7.10 shows the end to end delay performance of the 2 schemes. Z-MAC has slightly better delay performance at lower traffic loads because the generated packets are immediately transmitted when the channel is clear. However, when the traffic load increases, only the owners and their one-hop neighbours can compete to transmit in the current slot. Therefore, Z-MAC has similar delay characteristics as the ALOHA-Q scheme at high traffic loads. Both schemes with less than 40 slots/frame have higher delay because of the collisions and retransmissions.

In fig. 7.11, the practical results validate the simulation results in which the system achieves the steady state with 40 slots/frame. Also, ALOHA-Q-EPS and ALOHA-Q-DEPS are implemented with 40 slots/frame. The running throughput (see section 6.5.1 for definition) in association with the frame size is presented as a function of experiment time. The system with 50 slots/frame can converge earlier because the frame is oversized and some slots are unused. Each point in the running throughput curves is obtained from the start to the end of a window which has a length of 50 frames. ALOHA-Q-DEPS converges earliest because it provides more exploitation with the increasing number of successful transmissions, resulting in it being more protective of a chosen slot. As expected, ALOHA-Q-EPS experience a lower maximum throughput due to random access in exploration. It takes a longer time to reach its maximum throughput value because of the random transmissions in exploration.

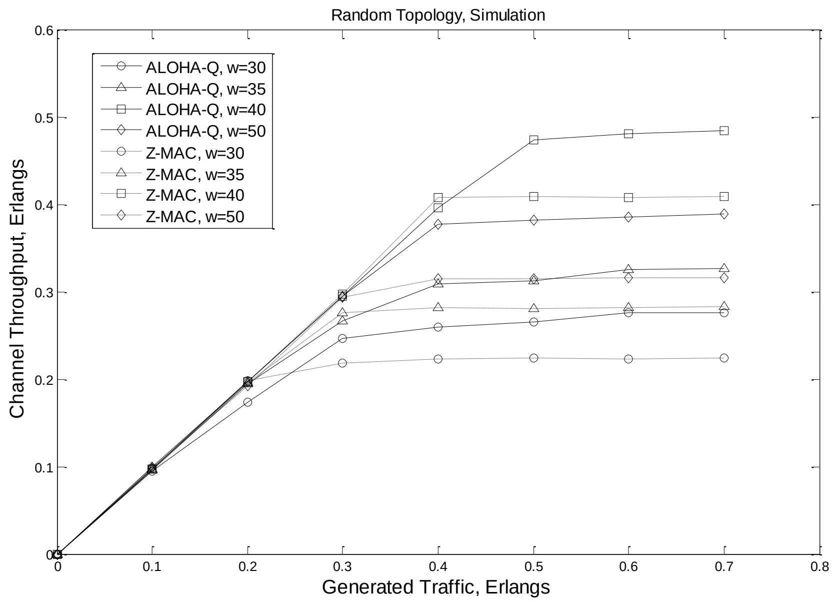


Fig. 7.8 Channel throughput with different frame sizes for random network.

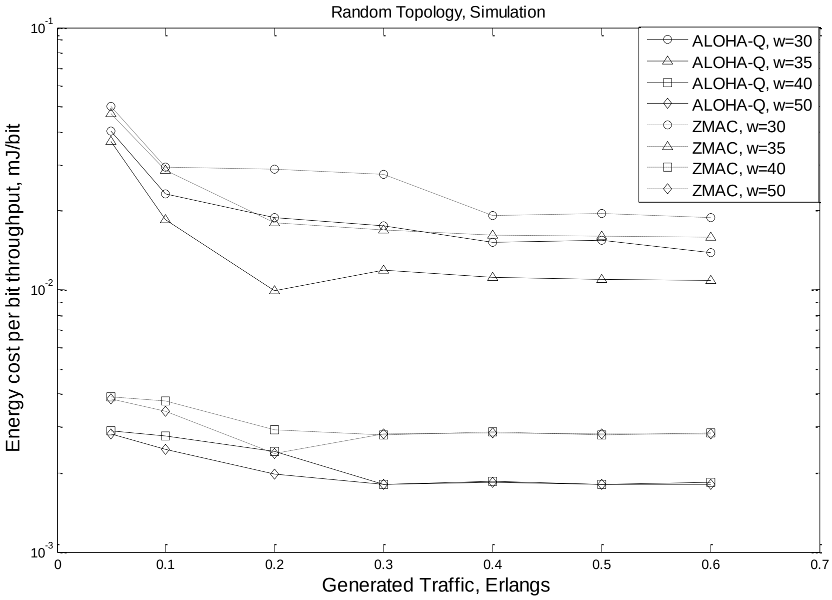


Fig. 7.9 Energy cost per bit throughput.

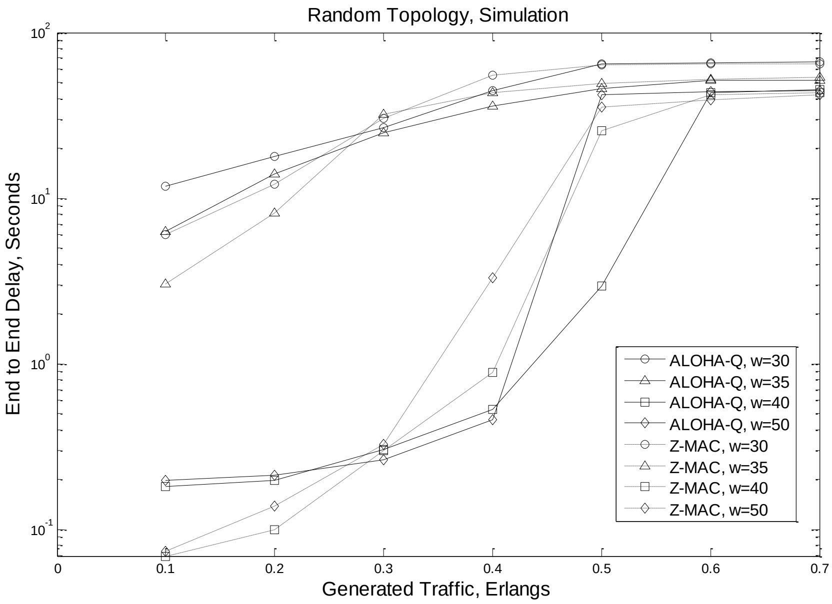


Fig. 7.10 End-to-end latency.

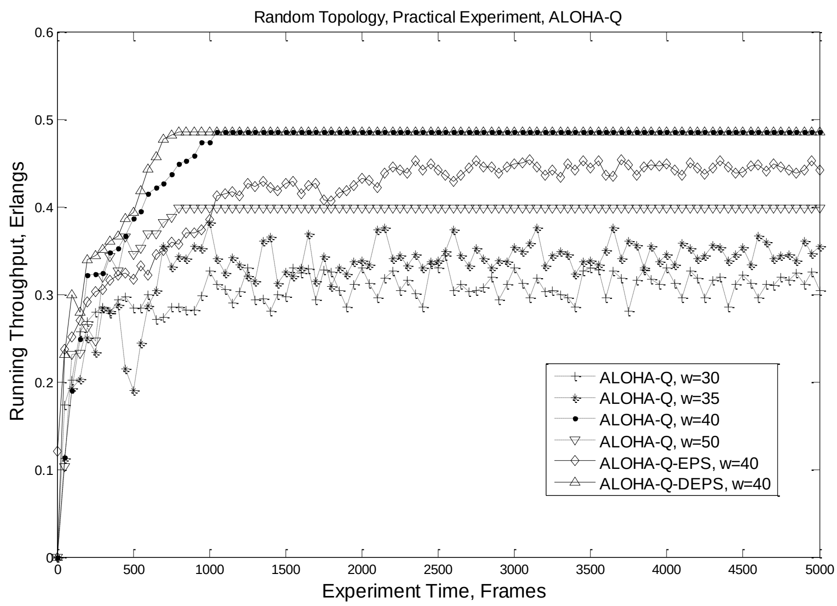


Fig. 7.11 Running throughput with different frame sizes.

7.4 ALOHA-Q in Dynamic Environments and with Variable Channel Conditions

7.4.1 Practical Issue of Acknowledgement Packet Loss

The practical loss of ACK packets which impedes effective operation of the learning algorithm resulting in the nodes not using unique transmission slots and causing collisions was studied in Section 5.3.1.2. In this section, ALOHA-Q is first implemented without guard periods in the presence of ACK loss using the random test-bed. Then, ALOHA-Q-EPS and ALOHA-Q-DEPS are implemented. Also, ALOHA-Q-DEPS with the punishment modification proposed in section 5.3.2.1 is implemented.

Fig. 7.12 presents the results of all these scenarios. ALOHA-Q and ALOHA-Q-EPS exhibit similar throughput performance at lower traffic loads, but ALOHA-Q-EPS has slightly lower throughput at higher traffic loads. Given that consecutive successful transmissions against infrequent failures will cause more frequent exploitation, the Q-value update will occur less frequently. Therefore, ALOHA-Q-DEPS serves to increase the channel throughput as the nodes have more willingness to protect the sequence of successful transmissions using the same slot. This is, however, not sufficient to enable all nodes to converge because of more substantial ACK packet loss when the channel load is high. ALOHA-Q-DEPS with the punishment modification provides the system with an extra level of protection in the transmission slots. Here, the system has a robust level of protection against the ACK loss. As a result of the modified learning process, all the nodes find unique slots and keep transmitting in these slots.

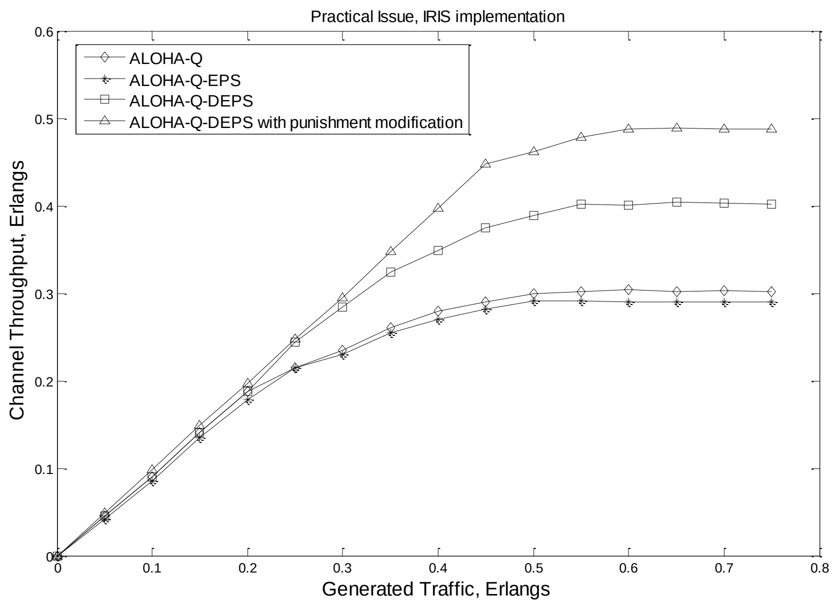


Fig. 7.12 Channel throughput.

7.4.2 Extending the Network with Participation of New Nodes

A typical WSN needs a high level of self-organization and needs to be robust to environmental changes such as failure of nodes which may result in the addition of new nodes. Fundamentally, sensor nodes need to operate uninterruptedly for long periods from limited-capacity batteries. Taking the constraints of sensor platform architecture into consideration, a well-designed MAC protocol should gracefully accommodate such network changes. From this perspective, the behaviour of ALOHA-Q and ALOHA-Q-DEPS is tested by adding new nodes as an event as depicted in fig. 7.13.

The results of both schemes highlights the importance of setting the frame size to an appropriately high level. The channel performance drops considerably as new nodes introduce more packet transmissions which requires the window size to be reset. In the programming of the sensor nodes, pre-defined timers are set to update the frame size at specific moments as shown in fig. 7.14. The channel throughput, as expected, increases with a higher frame size. Practical experiments showed that both schemes with 50 slots/frame operate in steady-state condition because addition of new nodes introduce more channel traffic to the network requring a higher frame length. Therefore, the frame length increases until the network converges to the steady-state again. ALOHA-Q-DEPS provides better channel throughput than ALOHA-Q until the network converges. ALOHA-Q-DEPS in large networks can protect the channel performance significantly while the nodes adapt to network changes. However, ALOHA-Q is prone to network failures due to the continuous exploration.

Discussion: The main aim of a typical WSN deployment is to cover large areas with hundreads or even thousands of sensor nodes. The participation time of the new nodes into the network will be usually unknown as it depends on the application requirements. Therefore, pre-defined timers in updating the frame size would not be a good idea in practice. *Code dissemination* [88] has recently been shown as a means of propagating new code in order to add new functionalities to the sensor networks, and can be applied to update the frame size as required here.

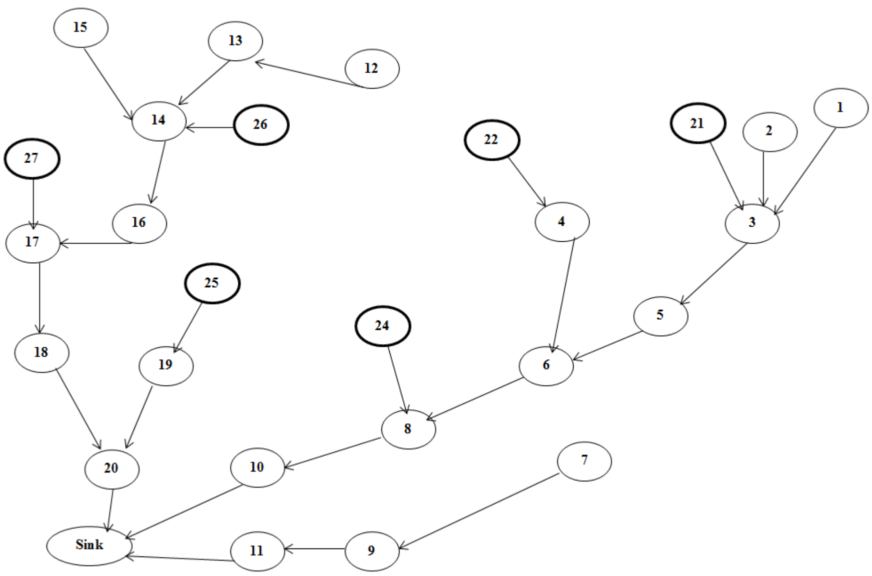


Fig. 7.13 Random topology with new nodes.

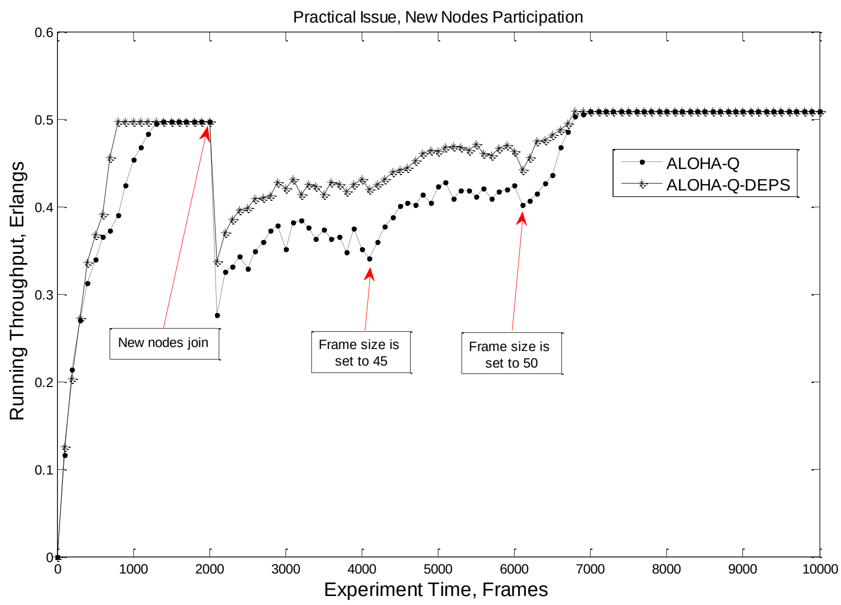


Fig. 7.14 Running throughput with new nodes.

7.5 Summary

This chapter has presented an extended implementation, applying ALOHA-Q to multi-hop WSNs, comparing its performance with Z-MAC through extensive simulations. Experimental evaluations have been carried out for the validation of simulation results. It has been shown that ALOHA-Q outperforms Z-MAC protocol in multi-hop networking. A problem with the ALOHA-Q protocol has been identified (the trade-off between exploration and exploitation) and an original solution is proposed, ALOHA-Q-DEPS. The channel performance of both ALOHA-Q and ALOHA-Q-DEPS has been evaluated for 2 practical real-world events: (1) packet losses caused by the sensor node hardware and (2) addition of new nodes to the network later on. ALOHA-Q-DEPS is more robust than ALOHA-Q in protecting and maintaining the channel performance in dynamic environments.

8. Further Work

Contents

8.1 Protocol Implementation on Larger Networks

The key properties of the protocols and modifications integrated with the protocols have been tested sufficiently on single-hop and multi-hop networks. However, the scale of experiments is rather small, so that it would be better if the performance evaluations of the protocols can be observed on larger scales. [67] has shown that the throughput performance of ALOHA-Q is significantly reduced due to the length of the transmission route and interference range. Hidden nodes in a high-density network can have a significant impact on the throughput performance. Therefore, the throughput of the RL-ALOHA and ALOHA-Q may not be stable for large-scale multi-hop networks.

8.2 Frame Size Decision

Frame size plays a significant role in the performance of RL-based protocols. As discussed in section 3.3.4, existing RL-based protocols use a fixed frame size which is determined based on the topology. However, as discussed in chapter 7, use of the fixed frame size is inefficient when environment or topology faces any change. A frame size algorithm is proposed in [66] for single-hop networks. For multi-hop networks, there is a need to develop a specific technique which determines the frame size dynamically. In all implementations in this thesis, all nodes are assumed to be active with a pre-defined constant frame size. However, the active number of nodes on a larger network can vary dynamically, while the frame size may not change for a relatively long time. Therefore, if the number of slots per frame is smaller than the number of active nodes, there could be a high instability in the network. Setting the right frame size is an important issue and should be handled directly by the protocol itself.

8.3 Reinforcement Learning with Energy Harvesting

Recent advances in energy harvesting (EH) technology have resulted in the design of new types of sensor node which are able to extract energy from the surrounding environment [89]. The major sources of EH include solar, wind, sound, vibration, thermal and electromagnetic. The concept of extracting ambient energy is to convert the harvested energy from existing environmental sources into electricity to power the sensor nodes, and an energy storage device accumulates the harvested energy. EH sensor nodes have the potential to provide an infinite lifetime of a battery through continuous energy harvesting. This has changed the fundamental design criterion of MAC protocols for EH-WSNs. However, the amount of ambient energy that can be harvested is time-variable and heavily dependent on environment conditions [90]. Various medium access control (MAC) protocols [91-97] with the objective of exploiting the ambient energy have been proposed for WSNs, and this has been termed ‘energy harvesting wireless sensor networks’ (EH-WSNs). The main objective of new MAC protocols on EH-WSNs is to increase the performance of the network in association with the available rate of energy to be harvested [98].

The rate of energy harvesting depends highly on the environment conditions. However, constant EH rates are assumed in the performance evaluations of the protocols. This is, however, not a realistic assumption as practical environments are inherently dynamic. If the average rate of energy harvesting can be analytically expressed depending on the environment properties, the effect of a time-variable energy harvesting rate can be simply determined and incorporated into the overall system performance. It is therefore believed that future MAC protocol design should consider more practical scenarios and define the dynamics of the environment. It can be concluded that intelligent methods (possibly RL methods) are required to carefully adapt the unpredictability of the EH process in order to maintain the performance of the protocol at an acceptable level. Q-learning can be used to represent the past energy harvesting behaviour through the Q-value. It can be then considered to predict the future energy availability.

9. Summary and Conclusions

This thesis presents a detailed description of the work undertaken during the PhD research from 2011 to 2014. The main point of this work is concerned with real-world implementations into medium access control (MAC) protocols in order to observe the performance of the MAC protocols based on the ALOHA schemes and reinforcement learning onto a set of sensor nodes. The early chapters presented the scope of the research, background knowledge and related work. The later chapters provided the detailed research work, which mainly deals with the practical implementation issues relating specially to dynamic environment and channel conditions.

A brief summary and conclusions for each chapter are given below:

Chapter 2 provided a general background to the whole work. Background information related to wireless sensor networks, physical-world components, operating systems and sensor network programming, the capture effect and reinforcement learning strategy have been presented. It summarized the basics of a wireless sensor network with its applications, depicting an example of sensor network architecture. Sensor network programming and main operating systems have been described briefly. The main features of many popular sensor platforms have been highlighted. The capture effect along with its scenarios and related work has been studied to understand packet collisions in WSN domain. Finally, an overview of reinforcement learning which is the main strategy used in developing efficient MAC protocols in later chapters has been presented.

Chapter 3 thoroughly surveyed the existing MAC protocols, describing their basic requirements, constraints and challenges. The protocols were categorized into contention-based, schedule-based, hybrid and RL-based protocols. A very wide range of MAC protocols from the past to the present such as ALOHA proposed in 1970 and QL-MAC in 2013 was introduced. The feasibility of the MAC protocols for real-world implementations, taking the architecture of real sensor nodes platforms into consideration, has been discussed. It is suggested that future MAC protocols need to consider the hardware specifications of sensor platforms for realism and reliability before being proposed.

Chapter 4 investigated the capture effect to provide an understanding of its impact; in a sense, a real-world measurement was carried out. Contrary to the theory, it is proved that use of the same transmission power level at the same distance creates the capture effect due to the nature of radios. A new and very practical capture scenario that may have a significant effect was studied and tested. Also, the capture coefficient to be simply added onto overall sensor network throughput performance has been derived. It is concluded that the capture effect is a radio-dependant property. Our measurements show that successful capture of a packet in a collision is performed through two stages: a valid preamble detection to be synchronized and the frame check sequence (FCS), which is in CRC-bytes and includes only the headers and the data payload. The pure and slotted ALOHA schemes have been studied with a finite number of users. The capture effect has been added to the throughput of pure ALOHA which significantly increased the maximum throughput. The implementation results of throughput for various packet lengths were presented and compared with the results of the analytical model derived.

The RL-ALOHA scheme has been practically evaluated in chapter 5. A hardware problem which has a great impact on the throughput performance has been faced. The problem is the inability of the receiver to send the acknowledgement (ACK) packet back after packet reception. The amount of the missing ACK packets depends on the traffic load in which increasing channel load would cause high ACK packets missing. The performance of the RL-ALOHA with this issue has been evaluated in a single-hop scenario. The results showed that the learning scheme does not work as effectively with increasing channel load, resulting in inducing lower throughput performance. In order to deal with this issue, a new punishment scheme has been proposed which enables the system to converge to the steady state.

The ALOHA-Q protocol has been extensively studied in chapter 6 and 7. The ALOHA-Q has been implemented practically in both simulation and test-beds. The chapter 6 focuses on the stability properties of ALOHA-Q in a single-hop and a linear-chain network in the presence of packet loss after the convergence of the system. In order to protect the convergence, a new punishment strategy has been proposed and implemented. Chapter 7 extends the implementations of ALOHA-Q to multi-hop networks. Three network topologies have been considered in this chapter: a linear-chain, a grid and a random network. Also, the performance of ALOHA-Q has been observed in 2 real-world conditions: the acknowledgement packet loss described in chapter 5 and the new nodes addition to the network after convergence. To further improve the performance of ALOHA-Q in these events, *exploration/exploitation* technique has been integrated into the learning strategy. The results showed that ALOHA-Q has achieved better throughput performance when compared with Z-MAC.

References

[1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam and E. Cayirci, “A survey on sensor networks”, *IEEE Communications Magazine*, vol. 40, no. 8, pp. 102-114, 2002.

[2] I. Demirkol, C. Ersoy and F. Alagoz, “MAC protocols for wireless sensor networks: a survey”, *IEEE Communications Magazine*, 2006, vol. 44, no. 4, pp. 115-121, 2006.

[3] K. Bouabdellah, H. Noureddine and S. Larbi, “Using wireless sensor networks for reliable forest fires detection”, *Procedia Computer Science* 19, pp. 794-801, 2013.

[4] G. Werner-Allen, K. Lorincz, M. Welsh, O. Marcillo, J. Johnson, M. Ruiz and J. Lees, “Deploying a wireless sensor network on an active volcano”, *IEEE Internet Computing*, vol. 10, no. 2, pp. 18-25, 2006.

[5] K.K. Khedo, R. Perseedoss and A.Mungur, “A Wireless sensor network air pollution monitoring system”, *International Journal of Wireless & Mobile Networks*, vol. 2, no. 2, pp. 31-45, 2010.

[6] C. Wenjie, C. Lifeng, C. Zhanglong, and T. Shiliang. “A realtime dynamic traffic control system based on wireless sensor network”, *International Conference on Parallel Processing Workshops*, pp. 258-264, 2005.

[7] A. Alaybeyoglu, K. Erciyes, A. Kantarci and O. Dagdeviren, “Tracking fast moving targets in wireless sensor networks”, *IETE Technical Review*, vol. 27, no. 1, pp. 46-53, 2010.

[8] M. P Durisic, Z. Tafa, G. Dimic, and V. Milutinovic, “A survey of military applications of wireless sensor networks”, *Mediterranean Conference on Embedded Computing* (MECO), pp. 196-199, 2012.

[9] J. Ko, C. Lu, M.B. Srivastava, J.A. Stankovic, A. Terzis and M. Welsh, “Wireless sensor networks for healthcare”, *Proceedings of the IEEE*, vol. 98, no.11, pp. 1947-1960, 2010.

[10] A. Milenkovic, C. Otto and E. Jovanov, "Wireless sensor networks for personal health monitoring: Issues and an implementation", *Computer Communications*, vol. 29, no. 13, pp. 2521-2533, 2006.

[11] M. Aminian and H.R. Naji, "A hospital healthcare monitoring system using wireless sensor networks", *J Health Med Inform*, vol. 4, no. 2, 2013.

[12] I. Silva, , L.A. Guedes, P. Portugal and F. Vasques, " Reliability and availability evaluation of wireless sensor networks for industrial applications", *Sensors*, vol. 12, no. 1, pp. 806-838, 2012.

[13] J. A Gutierrez, B. David, B. Lu, and T. G Habetler"Applying wireless sensor networks in industrial plant energy evaluation and planning systems", *Pulp and Paper Industry Technical Conference, Conference Record of Annual*, pp. 1-7, 2006.

[14] J. Yick, B. Mukherjee and D. Ghosal, "Wireless sensor network survey", *Computer Networks*, vol. 52, no. 12, pp. 2292-2330, 2008.

[15] P. Levis, S. Madden, J. Polastre, R. Szewczyk, K. Whitehouse, A. Woo, D. Gay, J. Hill, M. Welsh, E. Brewer, and D. Culler, "TinyOS: An operating system for wireless sensor networks", *Ambient Intelligence*, Springer-Verlag, pp. 115-148, 2005.

[16] A. Dunkels, B. Gronvall, T. Voigt, "Contiki:A lightweight and flexible operating system for tiny networked sensors", *IEEE International Conference on Local Computer Networks*, pp. 455-462, 2004.

[17] S. Bhatti, J. Carlson, H. Dai, J. Deng, J. Rose, A. Sheth, B. Shucker, C. Gruenwald, A. Torgerson and R. Han, "MANTIS OS: An embedded multithreaded operating system for wireless micro sensor platforms", *Mobile Networks and Applications*, vol. 10, no. 4, pp. 563–579, 2005.

[18] Q. Cao, T. Abdelzaher, J. Stankovic and T. He, "The LiteOS operating system: towards unix-like abstractions for wireless sensor networks", *International Conference on Information Processing in Sensor Networks*, pp. 233-244, 2008.

[19] Datasheet for BT node wireless measurement system available at [online] http://www.btnode.ethz.ch/Documentation/BTnodeRev3HardwareReference, accessed October 2014.

[20] Datasheet for EPIC mote wireless measurement system available at [online] http://moteware.com/pub-docs/datasheet-epic.pdf, accessed October 2014.

[21] Datasheet for EyesIFX wireless measurement system available at [online] http://www.twist.tu-berlin.de/wiki/WSN/EyesIFXv2Node, accessed October 2014.

[22] Datasheet for GWnode wireless measurement system available at [online] http://envisense.org/glacsweb/tech/probe/probe, accessed October 2014.

[23] Datasheet for IMote 2.0 wireless measurement system available at [online] http://wsn.cse.wustl.edu/images/e/e3/Imote2_Datasheet.pdf, accessed October 2014.

[24] Datasheet for IRIS wireless measurement system available at [online] http://www.memsic.com/userfiles/files/Datasheets/WSN/IRIS_Datasheet.pdf, accessed October 2014.

[25] Datasheet for Mica wireless measurement system available at [online], www.cs.berkeley.edu/~prabal/teaching/cs294-11-f05/slides/day6a.ppt, accessed October 2014.

[26] Datasheet for Mica2 wireless measurement system available at [online], http://www.eol.ucar.edu/isf/facilities/isa/internal/CrossBow/DataSheets/mica2.pdf, accessed October 2014.

[27] Datasheet for Mica2dot wireless measurement system available at [online], http://www.eol.ucar.edu/isf/facilities/isa/internal/CrossBow/DataSheets/mica2dot.pdf, accessed October 2014.

[28] Datasheet for MicaZ wireless measurement system available at [online], http://www.openautomation.net/uploadsproductos/micaz_datasheet.pdf, accessed October 2014.

[29] Datasheet for Mulle wireless measurement system available at [online], http://www.eistec.se/mulle.php, accessed October 2014.

[30] Datasheet for NeoMote wireless measurement system available at [online], http://metronomesystems.com/download/NeoMote%20Data%20Sheet.pdf, accessed October 2014.

[31] Datasheet for Shimmer wireless measurement system available at [online], http://www.capsil.org/capsilwiki/index.php/SHIMMER_Wireless_Sensor_Platform, accessed October 2014.

[32] Datasheet for TelosB wireless measurement system available at [online], http://www.willow.co.uk/TelosB_Datasheet.pdf, accessed October 2014.

[33] Datasheet for VESNA wireless measurement system available at [online], http://sensorlab.ijs.si/hardware.html, accessed October 2014.

[34] B. Ramamurthy, A. A. M. Saleh, and D. J. Goodman, "Perfect-capture ALOHA for local radio communications", *IEEE J. Sel. Areas Commun,* vol. 5, no. 5, pp. 806-814, 1987.

[35] C. Lau and C. Leung, "Capture models for model packet radio networks", *IEEE Trans.Commun.,* vol. 40, no. 5, pp. 917–925, 1992.

[36] J.J. Metzner, "On improving utilization in ALOHA networks", *IEEE Transactions on Communications*. vol. 24, pp.447-448, 1976.

[37] C. Namislo, "Analysis of Mobile Radio Slotted ALOHA Networks", *IEEE Journal on Selected Areas in Communications*, vol. 2, no. 4, pp.583-588, 1984.

[38] R.L. Borchardt and T.T. Ha, "Capture ALOHA with two random power levels", *Computer Communications*, vol. 17, no. 1, pp. 67-71, 1994.

[39] A. Kochut, A. Vasan, A. Shankar and A. Agrawala, "Sniffing out the correct physical layer capture model in 802.11b", *IEEE International Conference on Network Protocols*, pp. 252-261, 2004.

[40] D. Son, B. Krishnamachari and J. Heidemann, "Experimental study of concurrent transmission in wireless sensor networks", *ACM International Conference on Embedded Networked Sensor Systems*, pp. 237-250, 2006

[41] C. Gezer, A. Vasan, C. Buratti and R. Verdone, "Capture effect in IEEE 802.15.4 networks: modelling and experimentation", *IEEE International Symposium on Wireless Pervasive Computing*, pp. 204-209, 2010.

[42] A. Kiryushin, A. Sadkov and A. Mainwaring, "Real-World performance of clear channel assessment in 802.15.4 wireless sensor networks", *IEEE International Conference on Sensor Technologies and Applications*, pp. 625-630, 2008.

[43] J. Lee, W. Kim, S.-J. Lee, D. Jo, J. Ryu, T. Kwon and Y. Choi, "An experimental study on the capture effect in 802.11a networks", *ACM International Workshop on Wireless Network Testbeds, Experimental Evaluation and Characterization*, pp. 19-26, 2007.

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

[45] T. Jiang, D. Grace and Y. Liu, "Two stage reinforcement learning based cognitive radio with exploration control", *IET Communications*, vol. 5, no. 5, pp. 644-651, 2011.

[46] T. Jiang, D. Grace and P.D. Mitchell, "Efficient exploration in reinforcement learning based cognitive radio spectrum sharing", *IET Communications*, vol. 5, no. 10, pp. 1309-1317, 2011.

[47] I. Fister, X.S Yang and J. Brest, "A comprenhensive review of firefly algorithms", *Swarm and Evolutionary Computation* vol. 13, pp. 34-46, 2013.

[48] N. Abramson, "The aloha system: Another alternative for computer communications", *ACM Fall Joint Computer Conference*, pp. 281-285, 1970.

[49] G. Bianchi, "Performance analysis of the IEEE 802.11 distributed coordination function", *IEEE J. Sel. Areas Commun.*, vol. 18, no. 3, pp. 535–547, 2000.

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

[51] T. V. Dam and K. Langendoen, "An Adaptive energy-efficient MAC protocol for wireless sensor networks", *ACM International Conference on Embedded Networked Sensor Systems*, pp. 171–180, 2003.

[52] J. Polastre, J. Hill, and D. Culler, "Versatile low power media access for wireless sensor networks", *ACM International Conference on Embedded Networked Sensor Systems*, pp. 95–107, 2004.

[53] Y. Sun, O. Gurewitz, and D. B. Johnson, "RI-MAC: A receiver initiated asynchronous duty cycle MAC protocol for dynamic traffic loads in wireless sensor networks", *ACM International Conference on Embedded Networked Sensor Systems*, pp. 1-14, 2008.

[54] T. Lei, Y. Sun, O. Gurewitz, and D. B. Johnson, “PW-MAC: An energy-efficient predictive-wakeup MAC protocol for wireless sensor networks”, *IEEE INFOCOM*, pp. 1305-1313, 2011.

[55] 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, no. 1, pp. 63-78, 2006.

[56] W. Heinzelman, A. Chandrakasan and H. Balakrishnan, “Energy-efficient communication protocol for wireless microsensor networks”, *IEEE Hawaii International Conference on System Sciences*, pp. 1-10, 2000.

[57] B.O. Jalel, L. Mokdad, and B. Yahya, “An energy efficient priority-based QoS MAC protocol for wireless sensor networks”, *IEEE International Conference on Communications*, pp. 1-6, 2011.

[58] 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, no. 3, pp. 511-524, 2008.

[59] I. Rhee, A. Warrier, J. Min and L. Xu, “Drand: Distributed randomized TDMA scheduling for wireless ad hoc networks”, *ACM International Symposium on Mobile Ad Hoc Networking and Computing*, pp. 190-201, 2006.

[60] Y. Kim, H. Shin and H. Cha, “Y-MAC: An energy-efficient multi-channel MAC protocol for dense wireless sensor networks”, *International Conference Information Processing in Sensor Networks*, pp. 53-63, 2008.

[61] LAN-MAN Standards Committee of the IEEE Computer Society, Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (LR-WPANs), IEEE, 2003.

[62] M. H. S. Gilani, I. Sarrafi, and M. Abbaspour, “An adaptive CSMA/TDMA hybrid MAC for energy and throughput improvement of wireless sensor networks”, *Ad Hoc Networks*, vol. 11, no. 4, pp. 1297-1304, 2013.

[63] Z. Liu, and I. Elhanany, “RL-MAC: a reinforcement learning based MAC protocol for wireless sensor networks”, *International Journal of Sensor Networks*, vol. 1, no. 3, pp. 117-124, 2006.

[64] M. Mihaylov and Y. L. Borgne Elhanany, “Decentralised reinforcement learning for energy-efficient scheduling in wireless sensor networks”, *Int. J.*

Communication Networks and Distributed Systems, vol. 9, no. 3, pp. 207-224, 2012.

[65] 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*, pp. 511-515, 2012.

[66] Y. Yan, P.D. Mitchell, T. Clarke and D. Grace, "Distributed frame size selection for Q Learning based Slotted ALOHA protocol", *International Symposium on Wireless Communication Systems*, pp. 733-737, 2013.

[67] Y. Yan, P.D. Mitchell, T. Clarke and D. Grace, "Adaptation of the ALOHA-Q protocol to Multi-hop Wireless Sensor Networks", *European Wireless Conference*, pp. 921-926, 2014.

[68] Y. Chu, P. D. Mitchell, and D. Grace, "Reinforcement Learning based ALOHA for Multi-Hop wireless sensor networks with Informed Receiving", *IET Wireless Sensor Systems*, pp. 1-6, 2012.

[69] G. Stefano, A. Liotta, and G. Fortino, "QL-MAC: A Q-Learning based MAC for wireless sensor networks", *Algorithms and Architectures for Parallel Processing*, pp. 267-275, 2013.

[70] V. András, "The OMNeT++ discrete event simulation system", *European Simulation Multiconference* , vol. 9, pp. 185, 2001.

[71] The ns-2 simulation environment available at [online], http://www.isi.edu/nsnam/ns/, accessed October 2014.

[72] The Qualnet network simulator environment available at [online], http://web.scalable-networks.com/content/qualnet, accessed October 2014.

[73] The Matlab environment available at [online], http://www.mathworks.co.uk/, accessed October 2014.

[74] The RETOS OS environment available at [online], http://retos.yonsei.ac.kr/, accessed October 2014.

[75] The OPNET network simulator environment available at [online], http://www.riverbed.com/products/performance-management-control/opnet.html, accessed October 2014.

[76] D. C. Cox, "Wireless personal communications: What is it?", *IEEE Personal Communications,* vol. 2, no. 2, pp. 20-35, 1995.

[77] Z. H. E. N. Bin, M. Kobayashi, and M. Shimizu, "Framed ALOHA for multiple RFID objects identification", *IEICE Transactions on Communications,* vol. 88, no. 3, pp. 991-999, 2005.

[78] Data Sheet for AT86RF230 2.4 GHz IEEE 802.15.4-compliant RF Transceiver. http://www.atmel.com/Images/doc5131.pdf, accessed October 2014.

[79] K. Whitehouse, A. Woo, F. Jiang, J. Polastre and D. Culler, "Exploiting the capture effect for collision detection and recovers", *IEEE Workshop on Embedded Networked Sensors,* pp. 45–52, 2005.

[80] X. Li and Q. A Zeng, "Performance analysis of the IEEE 802.11 MAC protocols over a WLAN with capture effect", *Information and Media Technologies,* vol. 1, no. 1, pp. 679–685, 2006.

[81] L. Kleinrock, "Queuing Systems – Volume 1: Theory", Wiley-Interscience, 1975.

[82] N. Abramson, "The throughput of packet broadcasting channels", *IEEE Transactions on Communications*, vol. 25, no. 1, pp. 117-128, 1977.

[83] R. Rom, and M. Sidi, "Multiple access protocols: performance and analysis", Springer-Verlag, New York, Inc., 1990.

[84] A. Brand, and H. Aghvami, "Multiple access protocols for mobile communications GPRS, UMTS, and beyond", Wiley, Chichester, East Sussex, UK, 2012.

[85] Datasheet for CC1101 radio platform available at [online], http://www.ti.com/lit/ds/symlink/cc1101.pdf, accessed October 2014.

[86] J. Zhao and R. Govindan, "Understanding packet delivery performance in dense wireless sensor networks", *ACM International Conference on Embedded Networked Sensor Systems,* pp. 1–13, 2003.

[87] Y. Chu, "Application of reinforcement learning to medium access control for wireless sensor networks", Ph.D. dissertation, Dept. Elect. Eng., York Univ., York, United Kingdom, 2013.

[88] H. Sangwon, and P. Ning. "Secure code dissemination in wireless sensor networks", *Encyclopedia of Cryptography and Security,* Springer US, pp. 1099-1102, 2011.

[89] S. Sudevalayam, and P. Kulkarni, "Energy harvesting sensor nodes: Survey and Implications", *IEEE Commun. Surv. Tuts.*,vol. 13, no. 3, pp. 443-461, Third Quarter 2011.

[90] H.F. Rashvand, A. Abedi, J.M. Alcaraz-Calero, P.D. Mitchell, and S.C. Mukhopadhyay, "Wireless sensor systems for space and extreme environments: A review", *IEEE Sensors,* vol. 14, no. 11, pp. 3955-3970, 2014.

[91] Z.A. Eu, H.-P. Tan, and W.K.G. Seah, "Design and performance analysis of MAC schemes for wireless sensor networks powered by ambient energy harvesting", *Ad Hoc Networks,* vol. 9, no. 3, pp. 300-323, 2011.

[92] Z.A. Eu, and H.P. Tan, "Probabilistic polling for multi-hop energy harvesting wireless sensor networks", *IEEE International Symposium on Ad-hoc and Sensor Networking*, pp. 271-275, 2012.

[93] C. Fujii, and W.K.G. Seah, "Multi-Tier probabilistic polling in wireless sensor networks powered by energy harvesting", *IEEE International Symposium on Intelligent Sensor, Sensor Networks and Information Processing* , pp. 383-388, 2011.

[94] X. Fafoutis, and N. Dragoni, "ODMAC: An on-demand MAC protocol for energy harvesting – wireless sensor networks", *ACM Symposium on Performance Evaluation of Wireless ad hoc, sensor, and ubiquitous networks,* pp. 49-56, 2011.

[95] K. Nguyen, V.H. Nguyen, D.-D. Le, Y. Ji, D.A. Duong, and S. Yamada, "ERI-MAC: An Energy-Harvested Receiver-Initiated MAC protocol for wireless sensor networks", *International Journal of Distributed Sensor Networks*, 2014.

[96] S.C. Kim, J.H. Jeon, and H.J. Park, "QoS aware energy-efficient (QAEE) MAC protocol for energy harvesting wireless sensor networks", *In Convergence and Hybrid Information Technology,* Springer Berlin Heidelberg, pp. 41-48, 2012.

[97] S. Kosunalp, "MAC protocols for energy harvesting wireless sensor networks: a survey", to be published in *ETRI Journal*, 2015.

[98] W.K.G. Seah, Z.A. Eu, and H.P.Tan, "Wireless sensor networks powered by ambient energy harvesting (WSN-HEAP) – survey and challenges", *International Conference on Communication, Vehicular Technology, Information Theory and Aerospace & Electronic Systems Technology*, pp. 1-5, 2009.

[99] S. Kosunalp, P.D. Mitchell, D. Grace and T. Clarke, “Practical implementation issues of reinforcement learning based ALOHA for wireless sensor networks”, *International Symposium on Wireless Communication Systems*, pp. 360-364, 2013.

[100] S. Kosunalp, P.D. Mitchell, D. Grace and T. Clarke, “Experimental study of the capture effect for medium access with ALOHA”, *ETRI Journal*, 37(2), pp. 359-368, 2015.