

UNIVERSIDADE DO ALGARVE

*QoS Improvement in TCP/IP-based Wireless Sensor
Networks using Cross-Layer Optimization*

(Master Thesis in Computer Science Engineering)

Naz Fouad Hasan

Master Thesis in Computer Science Engineering

Work done under the supervision of: Prof. Dra. Noélia Correia and Dr. José Coimbra
2015

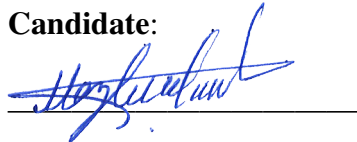
Statement of Originality

QoS Improvement in TCP/IP-based Wireless Sensor Networks using Cross-Layer Optimization

(Master Thesis in Computer Science Engineering)

Statement of authorship: The work presented in this thesis is, to the best of my knowledge and belief, original, except as acknowledged in the text. The material has not been submitted, either in whole or in part, for a degree at this or any other university.

Candidate:



(Naz Fouad Hasan)

Copyright ©Naz Fouad Hasan. A Universidade do Algarve tem o direito, perpétuo e sem limites geográficos, de arquivar e publicitar este trabalho através de exemplares impressos reproduzidos em papel ou de forma digital, ou por qualquer outro meio conhecido ou que venha a ser inventado, de o divulgar através de repositórios científicos e de admitir a sua cópia e distribuição com objetivos educacionais ou de investigação, não comerciais, desde que seja dado crédito ao autor e editor.



NETWORKING

Work done at Research Center of Electronics Optoelectronics and Telecommunications
(CEOT)

Abstract

Internet of Things (IoT) is becoming a reality and new and advanced applications are expected to emerge. For applications with reliability needs to work well in IoT environments, robust data transportation is required. Approaches like TCP are known for not being adequate in sensor network environments, while UDP has been included in the 6LoWPAN stack allowing low-power and limited processing devices to participate in the IoT. However, UDP provides no reliability. One way of providing reliability is to use link-layer acknowledgements but this mechanism may lead to an inefficient use of resources if used unconditionally throughout all the network. Another way is to request the confirmation of messages sent, done at the application layer, but this is an end-to-end process that can only be applied for specific message type transactions. If used for all data then there will be long delays and inefficient use of resources also. Here we address the design of a cross-layer reactive mechanism that improves reliability of data delivery, in order to support applications that require some reliability level when delivering data notifications. This mechanism introduces link layer reliability at specific nodes, gradually and only when needed, having no scaling problems. Results show that this mechanism can improve data delivery and improve the use of network resources.

Keywords: Internet of Things, 6LoWPAN, Cross-Layer Optimization, Reliability.

Resumo

A Internet das Coisas, ou *Internet of Things* (IoT), é um termo utilizado para nos referirmos a uma realidade em que vários tipos de objetos do nosso dia a dia estão ligados à Internet. Refere-se também à possibilidade de ligar o nosso mundo físico ao mundo digital através da Web. Este conceito tornar-se-á uma realidade em breve, e começa já a ter um forte impacto na nossa sociedade e na forma como vivemos o nosso dia a dia, estando a desencadear uma nova era de serviços inovadores. Esta onda de modernidade tecnológica está a ser vista por muitos como uma verdadeira revolução silenciosa.

A crescente utilização de sistemas incorporando sensores, que participarão na chamada Internet das Coisas, cujo funcionamento depende muitas vezes de baterias com tempo de vida limitado e que têm limitações na capacidade de armazenamento e processamento, fez com que se tornasse importante desenvolver protocolos que fossem energeticamente eficientes. O protocolo IEEE 802.15.4 foi um dos protocolos que foi standardizado com este objectivo. Este protocolo especifica as camadas física e de ligação em redes sem fios que têm taxas de transmissão baixas devido às suas limitações energéticas, de processamento e armazenamento. Mais recentemente foi também standardizada a utilização do IPv6 nestas redes, tendo sido proposta a pilha protocolar 6LoWPAN, permitindo assim uma melhor integração dos nós sensores na Internet e utilização das ferramentas e protocolos atualmente disponíveis nas redes IP. O trabalho desenvolvido nesta dissertação assume a utilização da pilha protocolar 6LoWPAN em redes de sensores sem fios. Uma rede de sensores sem fios é um sistema ad hoc composto por um conjunto de nós que têm capacidade de monitorar um determinado fenómeno, e que têm a capacidade de transmitir esta informação a nós de destino com mais capacidade de armazenamento e processamento. As redes de sensores sem fios são especialmente úteis em locais de difícil acesso ou áreas consideradas perigosas.

Com a Internet das Coisas surgirão, certamente, aplicações novas e mais avançadas que poderão ter necessidade de fiabilidade para poderem funcionar corretamente num ambiente IoT. Muitas das aplicações atuais em redes de sensores, por exemplo, não têm esses requisitos, ou têm menos, e operam apesar da perda de pacotes. Um dos protocolos de transporte mais conhecidos, para fornecer fiabilidade no transporte de dados em redes, é o TCP. Este tipo de abordagem não é, contudo, viável nas redes de sensores devido aos seus mecanismos de controlo de congestionamento e pelo facto de ser mais complicado fazer a compressão dos dados. A abordagem UDP é mais leve e, por este motivo, foi

adoptado na pilha protocolar 6LoWPAN permitindo que dispositivos com limitações de processamento e potência participem na IoT. O UDP não fornece, no entanto, fiabilidade. Ou seja, o UDP não garante retransmissão dos pacotes que se perdem, o que pode ser inaceitável para algumas aplicações com baixo nível de tolerância a perdas.

Uma das formas de fornecer fiabilidade, quando o protocolo de transporte por si não fornece, é utilizar confirmações, ou *acknowledgements*, na camada de ligação. Este mecanismo pode, no entanto, fazer com que os recursos sejam utilizados de forma ineficiente, caso seja usado incondicionalmente em toda a rede. Outra possibilidade será solicitar ao receptor que confirme as mensagens que recebeu, feito na camada de aplicação, mas este é um processo de extremo-a-extremo que apenas pode ser aplicado a alguns tipos de transações dado que aumenta o atraso e consumo de recursos, caso haja de necessidade de retransmitir pacotes. Ou seja, se utilizado para todos os tipos de pacotes de dados então existirão atrasos muito longos e os recursos serão utilizados de forma ineficiente. Outra abordagem passa pela adoção de mecanismos multi-camada. Esta abordagem explora as dependências e interações entre diferentes camadas para aumentar o desempenho da rede e, se for bem planeada, as suas vantagens superam a desvantagem de perda de modularidade (independência entre camadas). As abordagens multi-camada podem envolver interações entre duas ou mais camadas protocolares para que seja atingido um determinado objectivo. No caso das redes de sensores sem fios as preocupações energéticas estarão directa ou indirectamente ligadas a este objetivo.

Nesta dissertação o assunto abordado é o desenho de abordagens multi-camada para melhorar a fiabilidade no transporte de dados em redes de sensores sem fios. Mais concretamente, é proposto um mecanismo reativo multi-camada, que reage em função das condições da rede, que introduz fiabilidade no transporte de dados de forma a suportar aplicações que apenas funcionem com algum grau de fiabilidade no transporte através da rede. Este mecanismo introduz confirmações na camada de ligação de forma gradual e em nós específicos, apenas quando é necessário, não tendo problemas de escalonamento. Os nós selecionados são aqueles onde a atuação das confirmações pode reduzir mais a perda de pacotes das zonas consideradas críticas, melhorando assim a utilização dos recursos da rede. A abordagem proposta possui assim duas etapas: *i*) Seleção do conjunto de nós considerados críticos; *ii*) Introdução gradual do processo de confirmações nos nós selecionados. Para avaliar o desempenho da abordagem proposta foram feitas simulações usando o simulador Cooja, que assenta no Contiki OS, considerando diferentes cenários. Os resultados mostram que a seleção de um conjunto pequeno de nós considerados críticos, para introdução de confirmações na camada de ligação, pode reduzir a perda de pacotes de forma significativa sem aumentar muito o congestionamento na rede. No futuro poderão ser estudadas outras abordagens para a seleção do conjunto de nós considerados críticos, e também a influencia do protocolo de encaminhamento no desempenho da abordagem proposta.

Termos chave: Internet das Coisas, 6LoWPAN, Cooja, Optimização Multi-Camada, Transporte, Fiabilidade.

Acknowledgements

I dedicate this work and give special thanks to my Supervisors Prof. Dra. Noélia Correia and Dr. José Coimbra.

It is with immense gratitude that I acknowledge the help, support and patience of my supervisor Prof. Dra. Noélia Correia. Her inspirational discussions, professional critiques were invaluable to me personally and to the completion of this thesis. Big thank also to Prof. Dr. Álvaro Barradas who always provided unlimited support.

I would like to thank all professors from MEI (Mestrado em engenharia informatica) courses, those who contributed to my education.

I dedicate my dissertation work to my family. A special feeling of gratitude to my loving parents, Fouad Hasan and Sozan Omar whose words of encouragement and push for tenacity ring in my ears. My lovely grandmother, my uncle Saryas, my brothers Mohammed and Omar, my aunts Viyan and Sulave have never left my side and are very special, I owe you my deepest gratitude.

I also dedicate this dissertation to many friends who have supported me throughout the process. I will always appreciate all they have done, especially Mohammed Ali for helping me and my lovely friend Maymoni for being there for me throughout the entire master program.

I wish to thank Erasmus Mundus Iran Iraq Yemen (EMIIY) scholarship programme were funded me to get the chance to study the master degree.

Last, but by no means least, deepest thanks go to all people who took part in making this thesis real.

Contents

Statement of Originality	i
Abstract	iii
Resumo	iv
Acknowledgements	vii
Nomenclature	xiii
1 Introduction	1
1.1 Wireless Sensor Networks	1
1.1.1 Architecture	2
1.1.2 Protocol Stack	3
1.2 Internet of Things	4
1.3 Standardization	6
1.3.1 IETF Working Groups	6
1.3.2 IEEE 802.15.4	6
1.3.3 IPv6 over Low power Wireless Personal Area Networks	6
1.3.4 Routing Over Low power and Lossy networks	6
1.3.5 Constrained Restful Environments	6
2 The 6LoWPAN Adaptation Layer	7
2.1 Introduction	7
2.2 Characteristics of 6LoWPAN	8
2.3 6LoWPAN Protocol Stack	9
2.3.1 6LoWPAN vs Zigbee	9
2.3.2 Why IPv6?	10
2.4 6LoWPAN Details	11
2.4.1 The 6LoWPAN Architecture	11
2.4.2 6LoWPAN Format	12
2.4.3 Forwarding and Routing	13
2.4.4 Link Layer Adaptation and Frame Format	14

2.4.5	Header Compression	15
3	Reliability in 6LoWPAN-based Networks	17
3.1	Single-Layer Approaches	17
3.2	Cross-Layer Approaches	18
3.3	State of The Art	18
4	Cross-Layer Optimization Approach for Reliability Improvement	21
4.1	Introduction	21
4.2	Assumptions	22
4.3	Motivation and Definitions	22
4.4	Proposed Approach	23
5	Performance Evaluation	26
5.1	Experiment Setup with Contiki	26
5.1.1	Cooja Simulator	26
5.1.2	The Simulation Interface	27
5.1.3	Setting Mote Types	28
5.1.4	The Simulation Model	31
5.2	Simulation Results	34
6	Conclusions and Future Work	46
	References	47
	List of Publications	1

List of Figures

1.1	Basic components of a sensor node.	2
1.2	WSNs Protocol Stack.	3
1.3	Areas leading up to today's smart objects [27].	5
2.1	Protocol stack when using 6LoWPAN.	9
2.2	6LoWPAN architecture.	12
2.3	Typical LoWPAN Header Stacks.	12
2.4	Routing decision layer for both mesh-under and route-over routing schemes in 6LoWPAN.	14
2.5	General MAC frame format.	15
2.6	MAC frame control field (FCF).	15
2.7	Values of the frame type subfield.	15
2.8	6LoWPAN header compression example (L = LoWPAN header).	16
2.9	Standard IPv6/UDP headers (48 bytes).	16
2.10	6LoWPAN/UDP compressed headers (6 bytes).	16
3.1	Cross-layer interaction.	19
5.1	Creating a new simulation.	27
5.2	Cooja's window.	28
5.3	Create new mote type.	29
5.4	Select Contiki process source.	29
5.5	Compile the file.	30
5.6	Adding motes.	30
5.7	Randomly generated networks used in simulations.	35
5.8	The Cooja network window for Scenario I.	36
5.9	The Cooja network window for Scenario II.	36
5.10	Results for Scenario I with nodes generating a packet every 20s.	37
5.11	Results for Scenario I with nodes generating a packet every 15s.	38
5.12	Results for Scenario I with nodes generating a packet every 10s.	39
5.13	Results for Scenario I with nodes generating a packet every 5s.	40
5.14	Results for Scenario II with nodes generating a packet every 20s.	41
5.15	Results for Scenario II with nodes generating a packet every 15s.	42

5.16 Results for Scenario II with nodes generating a packet every 10s.	43
5.17 Results for Scenario II with nodes generating a packet every 5s.	44

List of Tables

- 2.1 Comparison between IPv6 and IPv4. 11
- 5.1 Results for Scenario I with nodes generating a packet every 20s. 37
- 5.2 Results for Scenario I with nodes generating a packet every 15s. 38
- 5.3 Results for Scenario I with nodes generating a packet every 10s. 39
- 5.4 Results for Scenario I with nodes generating a packet every 5s. 40
- 5.5 Results for Scenario II with nodes generating a packet every 20s. 41
- 5.6 Results for Scenario II with nodes generating a packet every 15s. 42
- 5.7 Results for Scenario II with nodes generating a packet every 10s. 43
- 5.8 Results for Scenario II with nodes generating a packet every 5s. 44

Nomenclature

6LoWPAN IPv6 over low-power wireless area networks
ACK Acknowledgement
ARP Address Resolution Protocol
ARQ Automatic Repeat reQuest
CCA Clear Channel Assessment
CLMHR Cross-Layer Multi-Hop Routing
CoAP Constrained Application Protocol
CoRE Constrained Restful Environments
CPU Central Processing Unit
CRC Cyclical Redundancy Check
CSMA-CA Carrier Sense Multiple Access with Collision Avoidance
DAG Directed Acyclic Graph
DCA Distributed Control Algorithm
EASYO Energy mAnagement and croSs laYer Optimization algorithm
ER edge router
FCF Frame Control Field
FCS Frame Check Sequence
FEC Forward Error Correction
HTTP HyperText Transfer Protocol
IEEE Institute of Electrical and Electronics Engineers
IETF Internet Engineering Task Force
IP Internet Protocol
IPv4 Internet Protocol version 4
IPv6 Internet Protocol version 6
IPsec Internet Protocol security
IPSO IP for Smart Objects (Alliance)
IoT Internet of Things
LAR Location-Aided Routing
LLN Low-power and Lossy Network
LoWPAN Low-power Wireless Personal Area Network
LR-WPANs Low-Rate Wireless Personal Area Networks
M2M Machine-to-Machine

MAC Medium Access Control
MFR MAC Footer
MHR MAC Header
MIPv6 Mobile IP version 6
MMSPEED Multipath and Multi-Speed Routing Protocol
MPR Multi-Packet Reception
MSDU MAC Service Data Unit
MTU Maximum Transmission Unit
NA Neighbor Advertisement
ND Neighbor Discovery
OS Operating System
OSI Open Systems Interconnection
PAN Personal Area Network
PCDST Power Control based Directed Spanning Tree
PHY Physical Layer
QoS Quality of Service
RA Router Advertisement
RDC Radio Duty Cycling
RF Radio Frequency
RFET Radio Frequency Energy Transfer
ROLL Routing Over Low-power and Lossy networks (IETFWG)
SIC Successive Interference Cancellation
SICS Swedish Institute for Computer Science
ST Spanning Tree
TCP Transmission Control Protocol
UDGM Unit Disk Graph Medium
UDP User Datagram Protocol
WG Working Group
WLAN Wireless Local Area Network
WPAN Wireless Personal Area Network
WSN Wireless Sensor Network

Introduction

1.1 Wireless Sensor Networks

The technological progress in microprocessors and embedded systems has enabled the development of smaller, lower-power and cheaper smart sensors. Such revolution empowers the use of *Wireless Sensor Networks* (WSNs) in many different scenarios and, consequently, increases the attention of the scientific community. WSNs are ad-hoc networks systems composed of thousands of smart sensing nodes capable of sensing the surrounding environment and communicating the sensed data to other nodes for storage and further analysis. Sensor nodes are the main components of every WSN and can be monitored and controlled wirelessly. These sensor nodes usually obtain power from batteries and, as such, the processing and wireless communication done at each node should be as energy efficient as possible [29]. Therefore, the development of energy efficient protocols is of uttermost importance..

A WSN has constraints such as limited energy, short communication range, low bandwidth and small memory size. Resource constraints impose limitations on the design and applications, which are related with the monitored environment. Thus, depending on the place and how it is to be used, the designer will plan the network size, network topology, and the deployment scheme. With those constraints in mind, researchers in WSNs try to create and improve existing protocols, allowing new applications and new algorithms to be developed.

Important characteristics of WSNs, like self-organization, fault-tolerance and rapid deployment, make them relevant in many fields and applications (e.g. military command, control, communications, intelligence, surveillance, and targeting systems) [1]. In addition to the WSN's wide range of applications, in the future new applications and services for wireless sensor networks will be unlocked.

The main challenges when designing WSNs can be summarized in:

- Fault tolerance
- Scalability

1.1 Wireless Sensor Networks

- Hardware constraints
- Sensor network topology
- Environment
- Transmission media
- Power management

1.1.1 Architecture

As shown in Figure 1.1, sensor nodes are composed by the following components: *i*) sensing unit, *ii*) processing unit with memory, *iii*) transceiver (communication unit), *iv*) power unit.

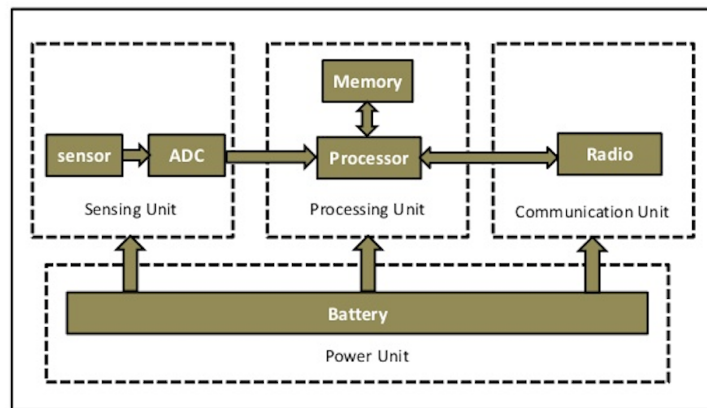


Figure 1.1: Basic components of a sensor node.

Sensing Unit

One or more sensor nodes to capture and reveal the environment. The sensing unit can be embedded in the mother board, or connected to an external port depending on the platform used.

Processing Unit

The processing unit consists of a microprocessor or microcontroller, and is known as the CPU of the node.

Transceiver Unit

To convert the physical world to the digital world by converting the sensed information to a form that can be stored, processed and acted upon.

Power Unit

This is the component that houses the batteries and provides energy to the unit. The nodes lifetime is heavily dependent on this unit.

1.1 Wireless Sensor Networks

1.1.2 Protocol Stack

In general, networks may be implemented using a variety of protocols. In WSNs the commonly used protocol stack of five layers is also adopted. This stack includes the following layers:

- **Application layer:** Includes a variety of application protocols. This layer may perform various sensor network tasks, such as node localization and time synchronization.
- **Transport layer:** It is responsible for end-to-end reliability between sensor nodes and the sink(s), packet retransmissions and may provide end-to-end security.
- **Network layer:** It takes care of routing the data sensed by source sensor nodes to sink(s). Power aware protocols may be used.
- **Data Link layer:** It provides the multiplexing of data streams, data frame detection, error control and *Medium Access Control* (MAC) that is responsible for fair and efficient share of the communication medium.
- **Physical layer:** Responsible for frequency selection, signal detection and power selection. Physical layer security techniques may also be implemented.

Besides the five protocol layers, as shown in Figure 1.2, there are three cross layer management planes in Wireless Sensor Networks, namely power management, connection management and task management.

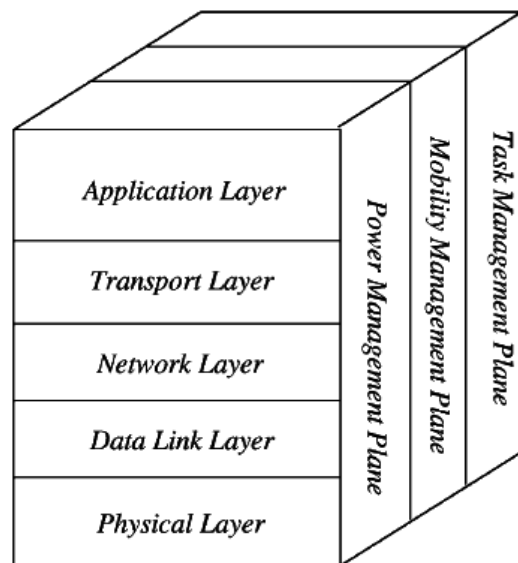


Figure 1.2: WSNs Protocol Stack.

1.2 Internet of Things

Power management plane

Manages the power of a sensor node required for sensing, processing and communication.

Connection management plane

Configuration/reconfiguration of sensor nodes. It is also responsible for registering the node's movement.

Task management plane

Responsible for schedules and task distribution between the sensor nodes to improve energy efficiency and extend network lifetime.

1.2 Internet of Things

Today's Internet is one of the largest engineered system that has been a great success over the past two decades. Until today the Internet has been growing, its contents have increased and new Internet-based applications have emerged. This network incorporates nearly 2 billion people, many servers, laptops, desktops and mobile units. At the present time, smart objects are also entering and becoming part of this heterogeneous network.

The smart object technology and its applications have names such as the Internet of Things, the Web of Objects, the Web of Things, and Cooperating Objects. Figure 1.3 show some of terms commonly used. Although there are slight differences in the connotations and definitions of those names, they represent the same fundamental type of technology [27]. We can say that the smart object is a physical object equipped with a form of sensor or actuator constrained in memory size, battery lifetime, and bandwidth provided. These objects have unique identifiers and the ability to transfer data over a network without being directly operated by humans. They might exist as components in buildings or vehicles, and have the ability to communicate with the outside world, and with other smart objects, in addition to communicating over wireless low power lossy networks. A large number of smart objects, possibly interconnected using the *Internet Protocol* (IP) and communicating with each other, is called Internet of Things. The IPv6 plays a fundamental role in the IoT, as it will be discussed later.

Smart object networks can be very large with a huge number of the nodes and generating huge amounts of data per node. One of the most important characteristics of smart objects is the ability to communicate. To guarantee a successful communication in such networks, it is important to respect the constraints of the smart objects such as power consumption, physical size, memory and CPU.

The aim of the IoT is to integrate, collect information and offer services to different groups of physical things used in different domains and, for this reason, IoT will be the most complex structure that has ever been created by humans. Things may communicate,

1.2 Internet of Things

collect information and collaborate with each other over the Internet, and sensors and actuators will be used to transform real things into virtual objects [4]. In 2009 the Internet of Things was still in its infancy but the capabilities of embedded devices (processor, power, and communication technologies) kept increasing, so has the complexity of communication standards, protocols and services. However, the embedded devices were not fully integrated with the Internet at the time.

The *Institute of Electrical and Electronics Engineers* (IEEE) released the 802.15.4 low-power *Wireless Personal Area Network* (WPAN) standard in 2003, which provides the first global low-power radio standard. ZigBee Alliance followed that standard and developed a solution for ad hoc network over IEEE.15.4. Although ZigBee and WPAN solve a small portion of the applications for wireless embedded networking, they still have problems with scalability and Internet integration. For this reason the 6LoWPAN was launched in 2011 allowing IEEE 802.15.4 to become IP-enabled, which allows a good interoperability between low-power devices and existing IP-based networks [24].

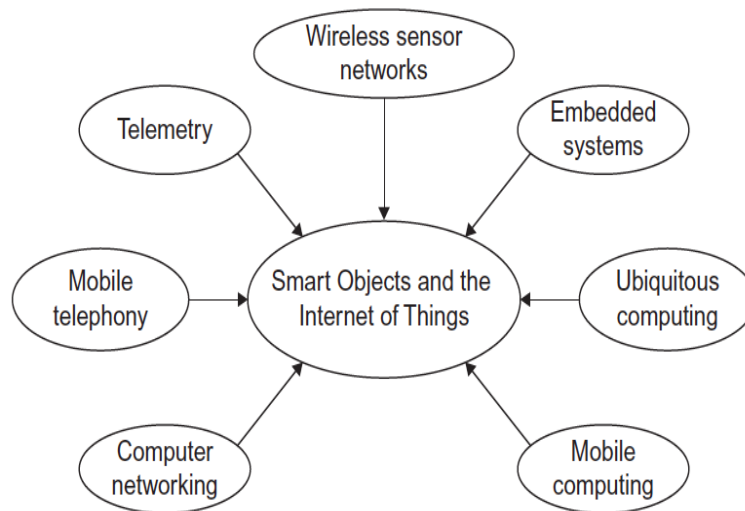


Figure 1.3: Areas leading up to today's smart objects [27].

1.3 Standardization

The IEEE and the *Internet Engineering Task Force* (IETF) are the standardization bodies that focus on the development of protocols and technologies for networks in general, including wireless sensor networks.

1.3.1 IETF Working Groups

The working groups are typically created to address a specific problem or to produce one or more specific deliverables (a guideline, standards specification, etc.). Upon completion of its goals and achievement of its objectives, the working group terminates. There are three IETF working groups of particular interest to WSN: *IPv6 over Low power Wireless Personal Area Networks* (6LoWPAN), *Routing Over Low power and Lossy networks* (ROLL) and *Constrained Restful Environments* (CoRE).

1.3.2 IEEE 802.15.4

IEEE 802.15.4 is a standard for low-power, low data rate wireless communication between small devices. It specifies the *physical layer* (PHY) and MAC layer on top of which the 6LoWPAN operates to build the wireless embedded Internet [17]. The physical layer supports different data rates: 250 kbps (2.4GHz), 40 kbps (915MHz), 20 kbps (868MHz). The MAC layer controls the access to the radio channel using *Carrier Sense Multiple Access with Collision Avoidance* (CSMA-CA) algorithm.

1.3.3 IPv6 over Low power Wireless Personal Area Networks

6LoWPAN working group started in 2007 to work on specifications for IPv6 to be used on IEEE 802.15.4 based networks.

1.3.4 Routing Over Low power and Lossy networks

ROLL working group specifies routing solutions for IP-based *Low-power and Lossy Networks* (LLN), which includes unreliable wireless networks. Although these routing solutions are not restricted to be used with 6LoWPAN, this was the main goal. RPL, the routing protocol for LLN, was the result of ROLL working group.

1.3.5 Constrained Restful Environments

CoRE working group has done the major standardization work for CoAP protocol. *Constrained Application Protocol* (CoAP) is an application layer protocol that is easily translated to HTTP for integration with the web, while meeting specialized requirements.

The 6LoWPAN Adaptation Layer

2.1 Introduction

WSNs are adhoc network systems composed of low power nodes capable of sensing the surrounding environment and communicating the sensed data to other nodes for storage and further analysis. These sensor nodes usually obtain power from batteries and, as such, the processing and wireless communication done at each node should be as energy efficient as possible [30]. Therefore, the development of energy efficient protocols is of uttermost importance.

With the aim of achieving energy efficiency in WSNs, IEEE developed the 802.15.4 standard, which specifies the physical and MAC layers for *Low-Rate Wireless Personal Area Networks* (LR-WPANs). This standard is widely used and forms the basis for all the protocols in the upper layers of WSNs, such as the network and transport layer protocols.

One of the most well known network protocols for WSNs is ZigBee, which can be used on top of 802.15.4 and can achieve a reliable and energy efficient network. However, ZigBee cannot successfully inter-operate with existing IP technology, unless there is some sort of gateway responsible for the translation between these two protocols. The perfect solution would be to use IP directly on the sensor networks, providing a complete interoperability with existing networks, so that existing IP based tools/protocols can be used and the IoT can become a reality. There are, however, a few problems with the use of IP directly at the network layer of WSNs. More specifically, sensors have limited processing power, small memory size and the 802.15.4 physical frame has a maximum payload of only 127 bytes [24]. Concerning the available payload size, it needs to be considered that the headers of IPv4 and IPv6 need 20 and 40 bytes, respectively, and if a transport protocol is also used there will be even less space available for actual data. For instance, an 802.15.4 frame with 127 bytes, and maximum 116 bytes of payload, using IPv6 and UDP will be left with only 68 bytes for data. Considering that the transmission of packets is the most energy consuming task of a sensor node, these headers pose a significant overhead on both data transmission and energy efficiency [11].

2.2 Characteristics of 6LoWPAN

Some lightweight implementations of IPv4 for sensor networks have been proposed, such as uIP and lwIP, which were able to be executed on devices with very small memory size and on very small 8-bit micro-controllers. However, none of these approaches tackles the issue of the very small payload size. The 6LoWPAN is able to reduce the size of the IPv6, UDP and TCP headers through stateless compression techniques. In the best case, 6LoWPAN is able to reduce the IPv6 header down to only 2 bytes.

2.2 Characteristics of 6LoWPAN

The characteristics of 6LoWPAN based networks that should be highlighted are:

- The maximum physical layer payload is 127 bytes, with 72-116 bytes of payload available after link layer framing, addressing and optional security. That is, packets have a very small packet size.
- Support for both 16-bit (short) and 64-bit (long) extended MAC addresses.
- Low bandwidth. Data rates of 250 kbps (2.4GHz), 40 kbps (915MHz), 20 kbps (868MHz).
- Topologies include star and mesh operations.
- Some or all devices are battery operated.
- Devices have low processing and memory capabilities.
- Large number of devices are expected to be deployed during the lifetime of the technology.
- Unreliable because of a variety of reasons: uncertain radio connectivity, battery drain, device lockups, physical tampering.

The purpose of using 6LoWPAN is to provide Internet-based communication to embedded devices and allow low power heterogeneous networks to be connected. Besides this, networks evolve more easily, becoming scalable, and infrastructures are designed having mobility in mind. The 6LoWPANs have a wide range of applications, for example:

- Facility, building and home automation
- Personal sports and entertainment
- Healthcare and well-being

2.3 6LoWPAN Protocol Stack

- Asset management
- Advanced metering infrastructures
- Environmental monitoring
- Security and safety
- Industrial automation

2.3 6LoWPAN Protocol Stack

To be able to improve energy efficiency and *Quality of Service* (QoS) in 6LoWPAN based networks, a good understanding of this protocol stack, and mechanisms incorporated in it, is required. The 6LoWPAN protocol stack follows the basic structure of the TCP/IP, with an additional adaptation layer between the MAC and network layer, as shown in Figure 2.1 [24]. This adaptation layer is responsible for header compression, fragmentation and reassembly of IPv6 packets that are too big to fit in an 802.15.4 frame, giving this protocol stack the needed flexibility that can be exploited to improve overall energy efficiency [24]. Conversion between full IPv6 to 6LoWPAN format is done by edge routers.

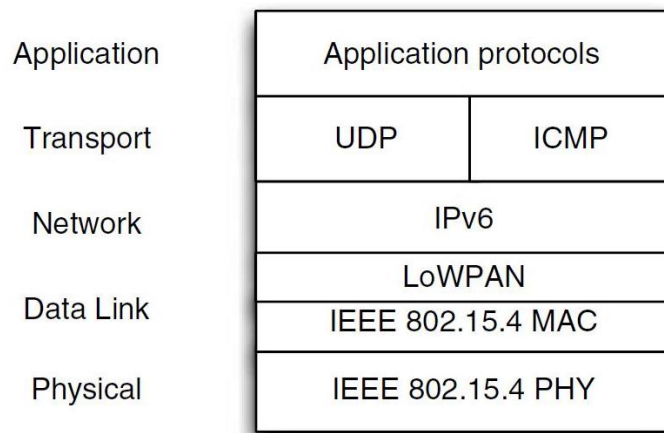


Figure 2.1: Protocol stack when using 6LoWPAN.

2.3.1 6LoWPAN vs Zigbee

When comparing the 6LoWPAN and Zigbee it is possible to state that:

- Zigbee has a complex middleware while 6LoWPAN has open development and portability.
- Above the IEEE 802.15.4, Zigbee uses entirely different protocols from Internet based ones.

2.3 6LoWPAN Protocol Stack

- In Zigbee, device discovery and other important features are performed at the application layer.
- Zigbee translates both address and messages between IP and Zigbee and, therefore, the IP stack vanishes at Zigbee network.

2.3.2 Why IPv6?

IPv6 is a new version of the IP used in the TCP/IP suite of protocols, and replaces the protocol IPv4 without changing the TCP/IP architecture. IETF decided to develop the new IPv6 standard when they found that the Internet is simply running out of IPs. The migration to IPv6 not only allows the expansion of the address space from 32 bits (IPv4) to 128 bits (IPv6) but also brings new features and enhancement. Table 2.1, from [12], presents the comparison between IPv4 and IPv6. Besides the benefits of working within an IPv6 network, e.g. Internet, the use of IPv6 in LoWPAN networks will allow a huge address space to be available and auto address configuration can be used.

The following list summarizes the features of the IPv6 protocol:

- New header format
- Large address space
- Stateless and stateful address configuration
- IPsec header support required
- Better support for prioritized delivery
- New protocol for neighbour node interaction
- Extensibility

2.4 6LoWPAN Details

	IPv4	IPv6	IPv6 Advantages
Address Space	2^{32} address space	2^{128} address space	More address space
Routing (packet fragmentation)	End station and routers	End station	Faster routing
Mobility	Need agent and used MIPv4	No agent and used MIPv6	Faster handover
Quality of Services	High latency and differentiated services	Low latency, Use traffic classes and flow labels	Enhanced support
Security	Site-to-site secure communications	End-to-end secure communications	More secure
Auto Configuration of Hosts	Need configuration	Plug-and-play	Faster configuration
Checksum in header	Included	No checksum	Faster routing
Header includes options	Required	Moved to IPv6 extension headers	Faster routing
Fragmentation	Routers and source node	Source node	Faster routing
IP configuration	Manually or DHCP	Auto-configuration or DHCP	Speed up connection
IPSec support	Optional	Required	Better security
Unicast, multicast and broadcast	Use all	Uses unicast, multicast and anycast	Less packet traffic
Address Resolution Protocol (ARP)	To resolve an IPv4 address	Replaced by neighbor Discovery	Less packet traffic

Table 2.1: Comparison between IPv6 and IPv4.

2.4 6LoWPAN Details

2.4.1 The 6LoWPAN Architecture

The architecture of LoWPANs, as shown in Figure 2.2 [24], can be classified into three types: simple, extended and ad-hoc. A simple LoWPAN uses a single edge router for Internet connection, while an extended LoWPAN can use multiple edge routers. Such routers connect the sensors to the Internet through a backhaul or a backbone link. As for the ad-hoc LoWPANs, they are not connected to the Internet and operate without any infrastructure.

2.4 6LoWPAN Details

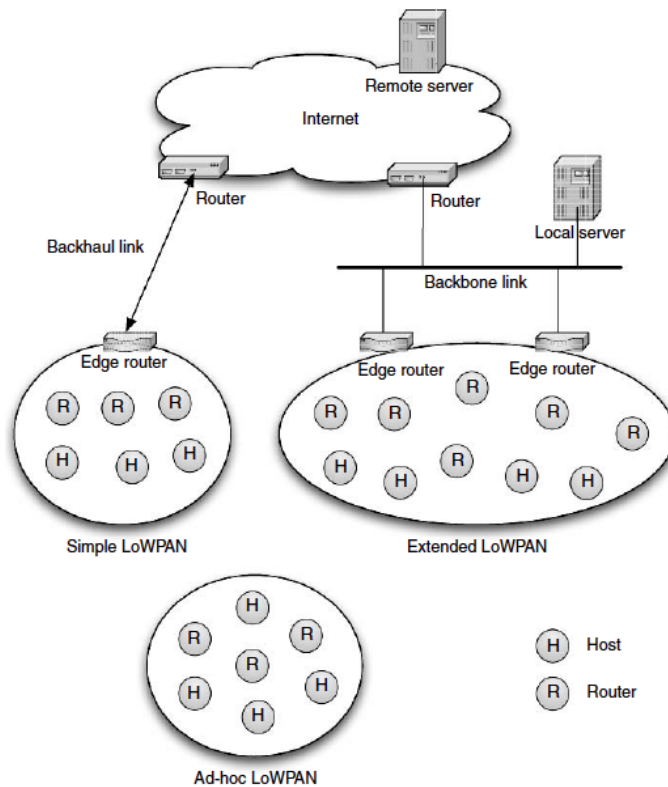


Figure 2.2: 6LoWPAN architecture.

2.4.2 6LoWPAN Format

The 6LoWPAN, defined in RFC 4944 [17], specifies several methods for an efficient transmission of IPv6 packets over IEEE 802.15.4 networks. This efficiency is achieved by means of an adaptation layer between the MAC and network layers, as previously shown in Figure 2.1 [24], which is responsible for header compression, fragmentation and reassembly of IPv6 packets that are too big to fit in a 802.15.4 frame.

All 6LoWPAN encapsulated datagrams are prefixed by an encapsulation header stack as shown in Figure 2.3 [7]. Each header in the stack starts with a header type field followed by zero or more header fields. The 6LoWPAN expresses each function in a self-contained subheader: mesh addressing, fragmentation, and header compression. Mesh addressing supports layer-two forwarding while fragmentation supports the IPv6 minimum MTU requirement.

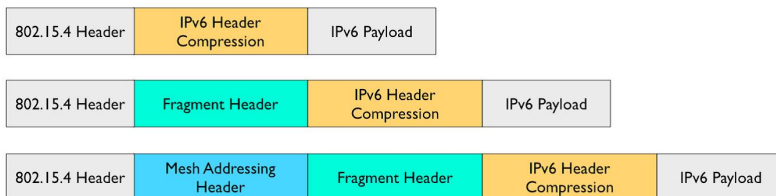


Figure 2.3: Typical LoWPAN Header Stacks.

2.4 6LoWPAN Details

Header compression

The compression of IPv6 header is done by not including information that is common knowledge among all nodes in the network.

Mesh addressing

The mesh addressing header is used to forward 6LoWPAN payloads over multiple radio hops and support layer-two forwarding [7].

Fragmentation

The fragmentation header is used if an entire payload datagram does not fit a single 802.15.4 frame [7]. In this case an IPv6 datagram is fragmented into multiple datagrams to accommodate the IPv6 minimum MTU requirement.

2.4.3 Forwarding and Routing

Forwarding and routing in 6LoWPAN networks can be done at the network layer, as with any other network, or it can be done at the adaptation layer, as shown in Figure 2.4 [24].

Mesh Under

In mesh-under scheme, routing and forwarding are performed at the link layer, based on 802.15.4 frame, or at the 6LoWPAN adaptation layer [5]. When sending a packet from a source to a destination, the MAC address (16-bit or 64-bit) is used to take forwarding decisions, instead of the usual network address. Using this method, different fragments of the same IP packet may travel through different paths, exploiting the path diversity in the mesh network, toward the same destination for reassembly.

Route Over

In route-over scheme, all routing decisions are taken in the network layer where each node acts as an IP router [5]. In route-over, each link layer hop is an IP hop. The IP routing supports the forwarding of packets with the help of IP routing tables and IPv6 hop-by-hop options. IP packets may be broken into fragments by the adaptation layer and sent to the next IP hop. At the next IP hop, all these fragments need to be reassembled by the adaptation layer and passed on to the network layer, which will check if the packet is destined for the node itself or not. If it is destined for the node itself, data is passed on to the next layer. If, on the other hand, the packet is destined to some other node, the routing table is checked and the packet is sent to the appropriate next hop.

2.4 6LoWPAN Details

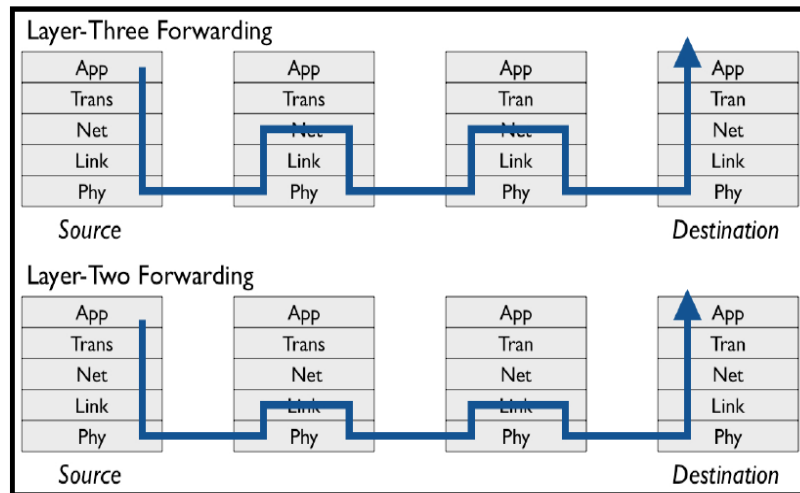


Figure 2.4: Routing decision layer for both mesh-under and route-over routing schemes in 6LoWPAN.

2.4.4 Link Layer Adaptation and Frame Format

IEEE 802.15.4 specifies the PHY and MAC layers that is assumed in 6LoWPAN standard to build the wireless embedded Internet [17]. In this specification there are four types of frames: beacon frames, MAC command frames, acknowledgement frames and data frames. MAC beacon frames are generated by the coordinator device to transmit beacons. The MAC commands are transmitted using a MAC command frame and are used to handle all MAC peer entity control transfers. Data acknowledgement frames are used to acknowledge successful reception of frames. Data frame is used for all transfers of data.

IEEE 802.15.4 MAC defines the data frame to transfer application data. The data frame format, as shown in Figure 2.5, consists of the *MAC Header (MHR)*, *MAC Service Data Unit (MSDU)* used to carry the information of IPv6, and *MAC Footer (MFR)*. The first field in the MHR is the *Frame Control Field (FCF)*, as shown in Figure 2.6, which indicates the type of MAC frame being transmitted, specifies the format of the address field and controls the acknowledgement. The frame type sub-field has 3 bits and shall be set to one of the nonreserved values listed in Figure 2.7 [6]. The size of the address field is between 0-20 bytes. The sequence number should match the acknowledgement frame with the previous data transmission. The payload field is variable in length, but the complete MAC frame may not exceed 127 bytes in length. The *Frame Check Sequence (FCS)* is a 16 bit *Cyclic Redundancy Check (CRC)*, and is used to verify the integrity of the MAC frame.

2.4 6LoWPAN Details

Octets: 2	1	0/2	0/2/8	0/2	0/2/8	Variable	2
Frame control	Sequence number	Destination PAN id	Destination address	Source PAN id	Source address	Frame payload	FCS
		Addressing Field					
MAC Header (MHR)						MAC Payload	MAC Footer (MFR)

Figure 2.5: General MAC frame format.

Bits: 0-2	3	4	5	6	7-9	10-11	12-13	14-15
Frame type	Security enabled	Frame pending	Ack request	Intra-PAN	Reserved	Dest-addressing mode	Reserved	Source addressing mode

Figure 2.6: MAC frame control field (FCF).

Frame type value $b_2 b_1 b_0$	Description
000	Beacon
001	Data
010	Acknowledgment
011	MAC command
100–111	Reserved

Figure 2.7: Values of the frame type subfield.

2.4.5 Header Compression

The header compression defined in RFC4944 is based on 802.15.4 16-bit and 64-bit addresses, shown in Figure 2.8 [24]. When a relatively large IPv6 packet needs to be sent, fragmentation is done first and then fragments are transmitted over IEEE 802.15.4 data frames, where each fragment carries a part of the original IPv6 packet. The 802.15.4 physical frame has a maximum payload of only 127 bytes, while the IPv6 and UDP header sizes are 48 bytes together. Header compression can reduce the IPv6 (40 bytes) to 4 bytes, and UDP (8 bytes) in the best case can be reduced to 2 bytes. Without header compression, 802.15.4 has its payload reduced considerably. Figure 2.9 [24] and Figure 2.10 [24] show IPv6/UDP header without and with header compression.

2.4 6LoWPAN Details

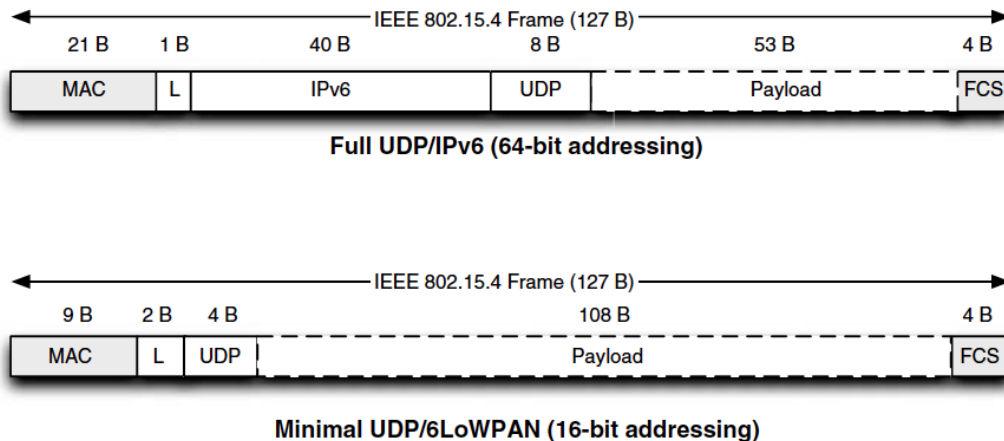


Figure 2.8: 6LoWPAN header compression example (L = LoWPAN header).

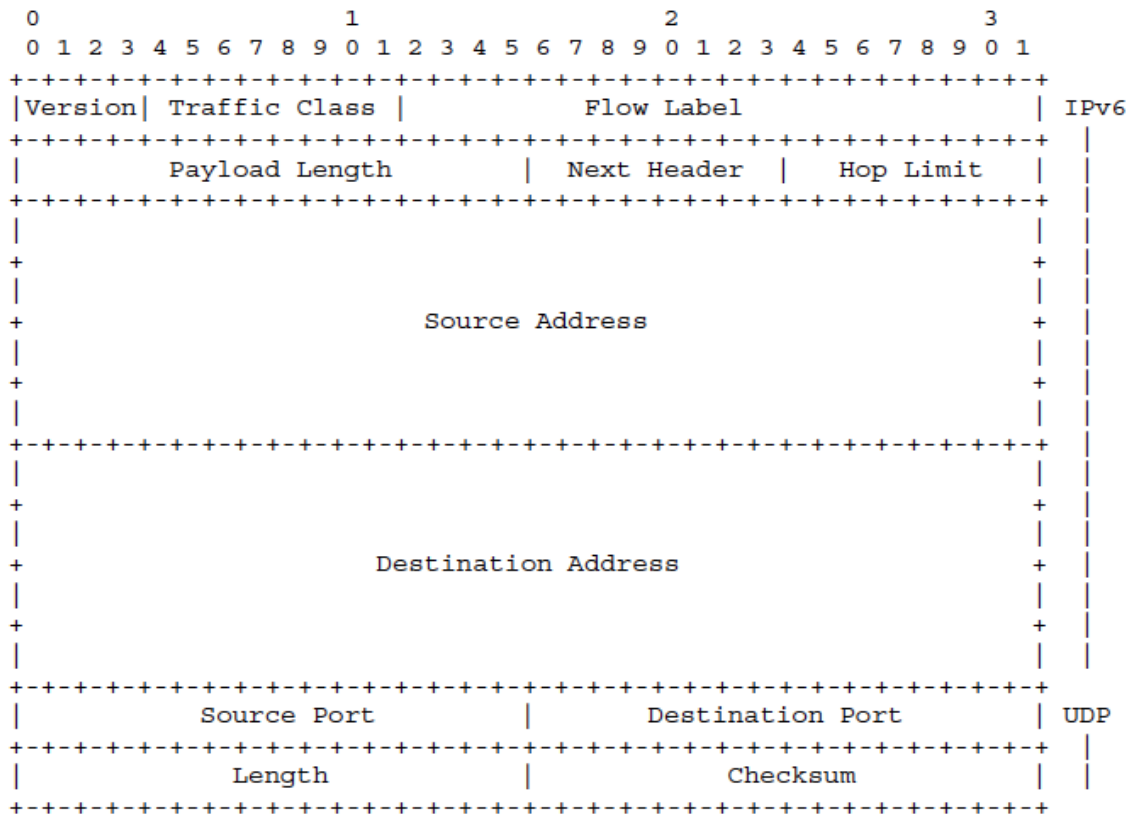


Figure 2.9: Standard IPv6/UDP headers (48 bytes).

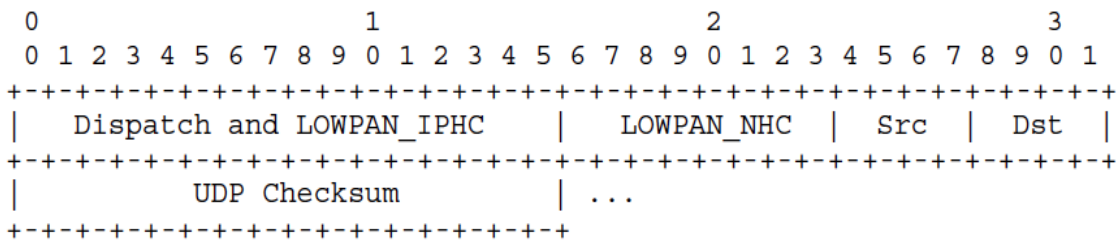


Figure 2.10: 6LoWPAN/UDP compressed headers (6 bytes).

Reliability in 6LoWPAN-based Networks

Over the last years most of the monitoring and tracking applications using sensor networks have been loss tolerant. However, management and re-tasking applications, requiring more reliable and robust data transportation, are emerging [3]. Also, as integration with mobile and ubicomp systems emerge, different classes of applications are expected meaning that mechanisms for reliability improvement become crucial.

3.1 Single-Layer Approaches

Transport protocols should be able to isolate applications from the unreliable nature of WSNs in an efficient and robust manner, and should be scalable. Approaches like *Transmission Control Protocol* (TCP) are known for not being adequate in sensor network environments, although being the most widely used transport protocol that ensures reliability of data transmission at the Internet.

We could think that TCP is a good choice for sensor node or *Machine-To-Machine* (M2M) reliable connections. However, TCP is not easy to compress, and is poorly suited for lossy wireless mesh networks because of its congestion avoidance design [24]. TCP is not commonly used with 6LoWPAN for performance, efficiency and complexity reasons while UDP has been included in the 6LoWPAN stack allowing low-power and limited processing devices to participate in the IoT. However, UDP is unreliable and does not ensure retransmission in case of packet drop at intermediate nodes, which may not be tolerable for applications that have low level or no packet drop requirements. Besides not being acceptable for some applications, packet forwarding will not be effective in the sense that resources are used at intermediate nodes and packets are not arriving to their destination.

Another way to provide end-to-end data reliability is through the data link layer. A data link layer reliable-delivery series is achieved by using *acknowledgements* (ACKs) and retransmissions. When sending a frame the sender has to request for acknowledge to know if the frame reached the destination successfully. If the source receives a positive ACK then the next frame will be sent, if an ACK is not received after a maximum period

3.2 Cross-Layer Approaches

of time, the frame has to be retransmitted. Retransmissions may create duplicate frames, if frames have been correctly received previously but the ACK reply was lost. In that case there must exist mechanisms to detect and eliminate such duplicates. This method guarantees data delivery but at the same time reduces the available bandwidth, which is not good.

3.2 Cross-Layer Approaches

Due to the characteristics of typical WSN applications, there are some constraints and requirements that must be tackled. Many WSN application have strict QoS requirements meaning that the required latency and reliability, for QoS to be achieved, must be met. With the aim to improve QoS provisioning, WSN designers and developers may resort to different cross-layer optimization techniques. Researchers have shown that it is possible to increase the performance, in certain scenarios of wireless networks, when the dependencies and interactions between different layers are exploited. The information exchange between different layers allows the design of advanced allocation and optimization algorithms but, on another hand, there is the disadvantage of missing the flexibility and modularity. Changes in one layer can result in changes in other layers (e.g. when cross layer is based on network and MAC layers, a change in the network layer might mean that the MAC layer needs to be adapted).

The main difference between single layer and cross-layer approaches is that single layer approaches investigate the optimization of protocols in individual layers, leading to the achievement of the required QoS provisioning in a specific layer, while cross-layer approaches provide QoS by jointly optimizing the interactions among two or more layer protocols to achieve an objective [2].

A survey on cross-layer QoS approaches in WSNs for delay and reliability-aware applications is presented in [2]. The authors summarize the previous works to achieve delay and reliability bounds in critical applications and systematise the different cross-layer interactions as shown in Figure 3.1. State of the art on cross-layer approaches is discussed in the following section.

3.3 State of The Art

With the creation of wireless sensor networks and increasing number of applications to be used for different purposes, different approaches have been proposed to overcome their constraints. Cross-layer optimization is a technique that can be used to improve network operation based on interactions between two or more WSN layers, namely based on information exchange. Cross-layer optimization for these networks has shown more

3.3 State of The Art

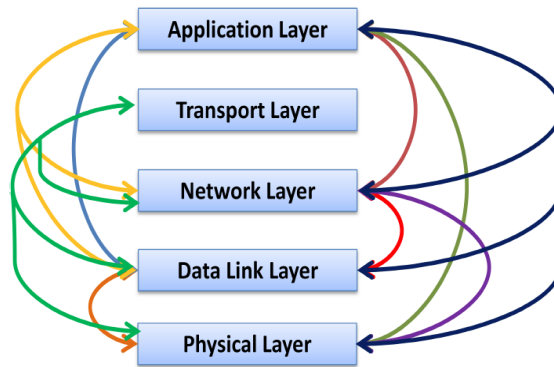


Figure 3.1: Cross-layer interaction.

efficient results than the traditional operation of single layer approaches. Many papers have studied cross-layer optimization among different layers, usually considering two layers, by taking into consideration the behavior of protocols at each layer together with WSN constraints. Most proposals aim to minimize the energy consumption, increase routing efficiency, and improve QoS provisioning. QoS requirements are closely related to energy consumption issues, as more power is needed to transmit the data and reduce channel errors, for example [16]. Moreover, retransmission will have effects on delay time and QoS.

In [25], a cross-layer model for a single-hop WSN with *Multi-Packet Reception* (MPR) was developed and achieved by using *Successive Interference Cancellation* (SIC) to increase the network capacity. MPR is a technology in the physical layer, but it has effects on the time slot of the data link layer. Results show that the model increases the network capacity in 100%.

In [15] a cross-layer approach has been discussed based on the transport and network layers. A game theory optimization model is used to reduce network congestion in wireless mesh networks.

WSN lifetime depends on the independent small batteries at each sensor node, since nodes without energy is essentially useless. Substantial research efforts have been spent on designing energy-efficient networking protocols to extend the network lifetime and solve the unbalanced energy waste among sensor nodes. In [10], a routing protocol *Cross-Layer Multi-Hop Routing* (CLMHR) has been developed based on the *Location-Aided Routing* (LAR) protocol that is proposed in [14]. LAR protocol uses location information to reduce the search space of routing, called request zone, based on the expected location of the destination node at the time of route discovery. On the other hand, in CLMHR the authors proposes the following: when a source node wants to send a packet to the destination, it must select the path according to distance and residual energy to reach the destination. Results show that CLMHR is more efficient than LAR protocol since the packet loss rate has been reduced and it prevents some nodes of becoming high power

3.3 State of The Art

consumers.

In [29], *Energy mAnagement and croSs laYer Optimization algorithm* (EASYO) is proposed so that the time-average utility of the source rate and energy management is maximized. The simulation results proof the efficiency of their algorithm regarding to the objective function used.

While the previous mentioned approaches discuss the interaction between two layers, other approaches propose three layer cross protocols, which is the case of [21] where cross optimization of PHY layer, MAC layer, and routing layer is presented. The authors propose an algorithm for a cross-layer routing protocol called *Power Control based Directed Spanning Tree* (PCDST), which is based on the traditional *Spanning Tree* (ST) routing protocol [13]. The PCDST reduces the total energy consumption of the network. The network throughput and the energy consumption is improved also.

Renewable sensor energy can be obtained through ambient energy sources, such as solar and wind, but those resources are still uncontrolled, unpredictable and not continuous. In [22] the authors studied an integrated network, MAC and PHY layer protocol based on the use of *Radio Frequency Energy Transfer* (RFET) approaches, which is presented as an alternative to the previously mentioned renewable sources. In RFET, a dedicated RF energy source is used to directly charge sensor nodes. They have concluded that using that technology improves the system performance and balances energy consumption among sensor nodes.

Interaction of application, network, MAC and physical layers has been presented in [23] to achieve a delay-aware framework. The authors present a cross-layer framework that employs cognitive radio communication, to circumvent the hostile propagation conditions in power systems, and supports QoS for smart grid applications. They have presented a suboptimal *Distributed Control Algorithm* (DCA) to support QoS through dynamic spectrum access, flow control, scheduling and routing decisions. The authors have shown that their protocol could reduce the delay when compared with a mechanism called *Multi-path and Multi-Speed Routing Protocol* (MMSPEED) [9], used for probabilistic QoS guarantee in WSNs.

Besides being used for optimization purposes, cross-layer approaches have also been used for analysis. In [28] the authors make a cross-layer analysis of three error control schemes: *Forward Error Correction* (FEC), *Automatic Repeat reQuest* (ARQ), and hybrid ARQ. This cross-layer analysis regards to multi-hop routing, energy consumption, and end-to-end latency. The results show that FEC and hybrid ARQ schemes are suitable for delay sensitive traffic in WSNs.

Cross-Layer Optimization Approach for Reliability Improvement

4.1 Introduction

Here we address the design of a reactive mechanism that improves reliability of data delivery, in order to support existing and future applications that require some level of reliability, in 6LoWPAN networks under the UDP transport protocol. This is a cross-layer mechanism that allows regions on the network experiencing data loss to be detected, activating link layer acknowledgement at specific nodes in these regions for loss reduction. This way, end-to-end confirmation for all notification messages is avoided since packet drops are reduced at some regions. The goals of this mechanism are:

Introduce reliability only when necessary: Packet drops experienced by users can vary widely over time or at different regions of the WSNs. Thus, the use of the same reliable end-to-end or link-based approach under any network conditions may be consuming. The reactive mechanism being proposed introduces reliability only when source nodes have frequent packet drops, which might be detected at the edge routers.

Gradually increase of link layer acknowledgements: Upon frequent packet drop detection, our mechanism selects a minimum set of nodes to implement local data link layer acknowledgement. Link layer acknowledgements are gradually introduced into the network. The chosen nodes are the ones that will most likely reduce losses. This gradual mechanism imposes minor delays while saving resources (e.g. energy) when compared with full end-to-end or full link layer acknowledgements.

Operate correctly under any environment: Since the approach is able to react according to the network conditions while using UDP light transport protocol, it has no scaling problems and is able to operate correctly under any environment. This aspect is important as large scale WSNs are emerging.

4.2 Assumptions

Therefore, the cross-layer reactive mechanism proposed involves IEEE 802.15.4/6LoWPAN stack layers with the aim of improving reliability at critical regions. This mechanism can be implemented by edge routers that may receive power from the grid.

4.2 Assumptions

1. **Neighbour Discovery in 6LoWPAN:** The *Neighbour Discovery* (ND) protocol [18] is used by nodes to discover other nodes and find their link-layer addresses, and to find routers and keep reachability information about the paths to neighbors that the node is actively communicating with. A node can be classified as host and router, depending on whether node is able to forward IP packets not addressed to itself. Thus, routers have additional functions in ND compared to hosts. In LoWPANs there is a third role, that of an edge router, because routers inside the mesh might have limited capabilities. An edge router will perform more complex tasks, relieving mesh non-edge routers, meaning that they must centralize some of the protocol state. *Router advertisement* (RA) messages are used to disseminate context information across the topology. The cross-layer approach is assumed to run on edge routers where full context information is available.
2. **Sequence Number in 802.15.4 Frame:** The 802.15.4 frame includes one sequence number byte for acknowledgement purposes, and this is unique for each outgoing frame. In Contiki this sequence number can remain unchanged over the multiple hops. The cross-layer approach at edge routers keeps track of arriving packet sequence numbers to control the packet drop level per source. This way critical nodes are detected.
3. **Acknowledgments:** The 802.15.4 includes acknowledgment frames that are used to ensure successful data transmission, if requested. The cross-layer approach being proposed activates acknowledgement at selected nodes, in a gradual way, for reliability improvement.

4.3 Motivation and Definitions

Errors accumulate exponentially over multiple hops. That is, assuming that there is a packet dropping probability of p at each hop, the chances for a message to arrive successfully to the destination (edge router in our case) are $(1 - p)^n$, where n is the number of hops. For large prone to error/drops regions the deliver of correct packets will be, therefore, very low. For better use of bandwidth it becomes, therefore, important to reduce

4.4 Proposed Approach

drops as resources used for packet forwarding at previous hops will not be productive if packets are dropped ahead. But, on another hand, involving all intermediate nodes in a fully hop-by-hop approach can be overwhelming since acknowledging all transmissions impose delays and data cache mechanisms are required.

In 6LoWPAN networks, registration and neighbour discovery allow edge routers to know who is reachable through who. Registration can be done directly to an edge router or using intermediate routers. This, together with a knowledge of the adopted routing algorithm, allows problematic areas of the network to be identified at edge routers if sequence numbers per source node are traced. That is, if gaps between sequence numbers are detected at edge routers then the problematic node(s) can be any node(s) traversed from source to the edge router, or their neighbours due to interference. Whenever problematic areas are identified, local link layer acknowledgements can be request for specific nodes. The goal is to request for link layer acknowledgement at nodes with a higher probability of improving reliability. This problem is called cross-layer data transportation improvement (XL-DTI) and is defined as follows:

Definition 1 (XL-DTI) *Assume a 6LoWPAN network $\mathcal{G}(\mathcal{N}, \mathcal{E}, \mathcal{I}_n, \mathcal{R}_n)$ where sensor data from the wireless sensor network nodes in \mathcal{N} is sent toward one of the edge routers in \mathcal{E} . Upon frequent packet drop detection, and having knowledge of the interference range of nodes $\mathcal{I}_n, \forall n \in \mathcal{N}$, and undergoing routing $\mathcal{R}_n, \forall n \in \mathcal{N}$, determine the set of most critical nodes, for further link layer acknowledgement activation, by crossing the following information: packet drops per source (by tracing sequence numbers of arriving frames), routes adopted by sources across the wireless network and interfering nodes.*

It is assumed that set \mathcal{I}_n includes the nodes interfering with n and \mathcal{R}_n gives the nodes at the route being used by n .

4.4 Proposed Approach

The proposed cross-layer approach for reliability increase includes two steps:

1. Selection of critical node set;
2. Gradual introduction of link layer acknowledgements into the network.

These two steps are discussed next. As previously said, it is assumed a 6LoWPAN network $\mathcal{G}(\mathcal{N}, \mathcal{E}, \mathcal{I}_n, \mathcal{R}_n)$, where \mathcal{N} includes all wireless sensor network nodes, \mathcal{E} refers to the set of edge routers, \mathcal{I}_n indicates nodes interfering with n and \mathcal{R}_n indicates nodes at the route being used by n . A frequency drop threshold, T^C , is also adopted at edge routers to determine if the source node in question is to be considered critical.

4.4 Proposed Approach

Step 1: Selection of Critical Node Set

At this stage a set of nodes is selected. Each node has a weight associated with it, which is related with its probability of increasing network packet delivery if it starts requesting for acknowledgement when forwarding packets. Node selection is done taking this weight into consideration. The following information is given:

- \mathcal{R}_n Set of nodes used by node $n \in \mathcal{N}$ in its way to an edge router in \mathcal{E} (route).
- \mathcal{I}_n Set of nodes at the interference range of node $n \in \mathcal{N}$, n included.
- \mathcal{N}^X Set of nodes with drops above the threshold T^C .
- \mathcal{N}^P Set of potential critical nodes will be $\mathcal{N}^P = \{i : i \in \mathcal{I}_j, \forall j \in \mathcal{R}_k, \forall k \in \mathcal{N}^X\}$.
Each $n \in \mathcal{N}^P$ has a weight defined by $w(n) = \min_{n' \in \mathcal{N}^X, n \in \mathcal{C}_{n'}} \{|\mathcal{C}_{n'}|\}$, such that $\mathcal{C}_{n'} = \{\mathcal{N}^P \cap \{i : i \in \mathcal{I}_j, \forall j \in \mathcal{R}_{n'}\}\}$.

Considering \mathcal{N}^X a set of source nodes whose packet drops are higher than T^C (traced at edge routers), a set of potential critical nodes, \mathcal{N}^P , is found by considering all nodes (route) used to forward data from source to an edge router, and their neighbour/interfering nodes. This is based on the idea that if a node is making packet drops then its neighbours might have problems too. As this approach is intended to improve global network efficiency, such neighbours should be considered as potential critical nodes too. The inclusion of neighbour nodes will also reinforce the weight of some nodes, influencing their probability of being selected, as will become clearer next.

The weight $w(n)$ is related with the following: for each node n' with drops above the threshold, node in \mathcal{N}^X , consider all the nodes used to forward its data, and interfering nodes, that are also potential critical nodes, $\mathcal{C}_{n'}$. The size of the smallest $\mathcal{C}_{n'}$ set including n will be the weight of n because the lowest the intersection with the set of potential critical nodes then the higher the certainty that n is the reason behind the non arrival of packets to edge routers, meaning that it should request for link layer acknowledgements when forwarding packets to the next node.

The goal is to select a minimum set of nodes whose interfering areas, together, cover all the critical region. That is, not all nodes considered potential critical nodes should be selected. Only the smallest subset, whose interfering areas cover the critical region, should be selected. That is, the nodes selected together with their neighbours/interfering nodes should cover all the critical region \mathcal{N}^P . This will allow nodes with low weight (high probability of reducing drops) to be selected while avoiding congestion at some areas of the network (high number of acknowledgements of many nearby nodes) that could lead to inefficient use of resources in some regions. The following variables are necessary:

4.4 Proposed Approach

- σ_n One if node $n \in \mathcal{N}^P$ is selected, zero otherwise.
- γ_n One if node $n \in \mathcal{N}^P$ interferes with at least one selected node, zero otherwise.

The following problem can be formalized:

– Objective Function:

$$\text{Minimize } \sum_{n \in \mathcal{N}^P} \sigma_n \times w(n) \quad (4.1)$$

– Interference coverage of a node:

$$\gamma_n \leq \sum_{i \in \mathcal{I}_n \cap \mathcal{N}^P} \sigma_i, \forall n \in \mathcal{N}^P \quad (4.2)$$

– Full critical area coverage:

$$\sum_{i \in \mathcal{N}^P} \gamma_i = |\mathcal{N}^P| \quad (4.3)$$

– Non-negative assignments

$$\sigma_n, \gamma_n \in \{0, 1\}. \quad (4.4)$$

After this problem is solved the variables σ_n set to 1 will be the ones where link layer acknowledgements should be set first. This is to be done in a progressive way as explained next.

Step 2: Gradual Introduction of Link Layer Acks

When setting link layer acknowledgements the following steps must be performed:

- Define $\Pi = \{n : \sigma_n = 1, \forall n \in \mathcal{N}\}$;
- Sort Π by increasing order of $w(n)$;
- For each $n \in \Pi$:
 - Apply link layer acknowledgement;
 - Evaluate dropping during Δ_t . If drops have decreased then stop.
- $\mathcal{N}^P = \mathcal{N}^P \setminus \Pi$;
- Perform Step 1 considering the new \mathcal{N}^P .

Performance Evaluation

5.1 Experiment Setup with Contiki

ContikiOS is an open source operating system for embedded systems and wireless sensor networks developed by Adam Dunkels from the *Swedish Institute of Computer Science* (SICS) and is in release 2.7, which is available as a VMware virtual machine, called Instant Contiki. This might be downloaded from the SICS¹ and requires the installation of VMware Player. The Instant Contiki is based on Ubuntu Linux and supports many hardware platforms, including MSP430 and AVR. Source code version is also available, which requires the use of a C compiler in Linux or Windows. Contiki provides three network mechanisms: *i*) the uIP TCP/IP stack, which provides IPv4 networking; *ii*) the uIPv6 stack, which provides IPv6 networking and the RPL routing protocol for low-power lossy IPv6 networks, and the 6LoWPAN header compression and adaptation layer for IEEE 802.15.4 links; *iii*) the Rime stack, which is a set of custom lightweight networking protocols designed specifically for low-power wireless networks. Therefore, Contiki is developed with IPv6 connectivity in mind, providing the compressed version of IPv6 called 6LoWPAN. The implementation of 6LoWPAN in Contiki is based on RFC 4944 to transmit IPv6 Packets over IEEE 802.15.4 Networks.

5.1.1 Cooja Simulator

Cooja [20] is a Java-based simulator developed for ContikiOS. Cooja is contained in Instant Contiki and allows the developers to test their code and systems before running it on the target hardware. In Cooja nodes can be simulated also, allowing real hardware platforms to be emulated. To use Cooja we must compile it and run it, through `ant run`. After compilation Cooja runs alone.

¹SICS Swedish ICT is a research institute for applied information and communication technology.

5.1 Experiment Setup with Contiki

The simulation can start with two plugins: a log listener that listens to the serial ports of all nodes, and a visualizer that shows the information about the nodes. There are other plugins, such as UDGM visualizer that allows us to analyze radio transmissions and change its range.

Creating A Simulation

First we will need to create a new simulation. To do this, we click *File > NewSimulation*. This will present a dialog box, as shown in Figure 5.1. Then a suitable title is provided to the simulation model and the the radio medium that best suits the simulation type is selected. *Unit Disk Graph Medium (UDGM)* is quite suitable for most simulations. When we are done, "Create" should be clicked.

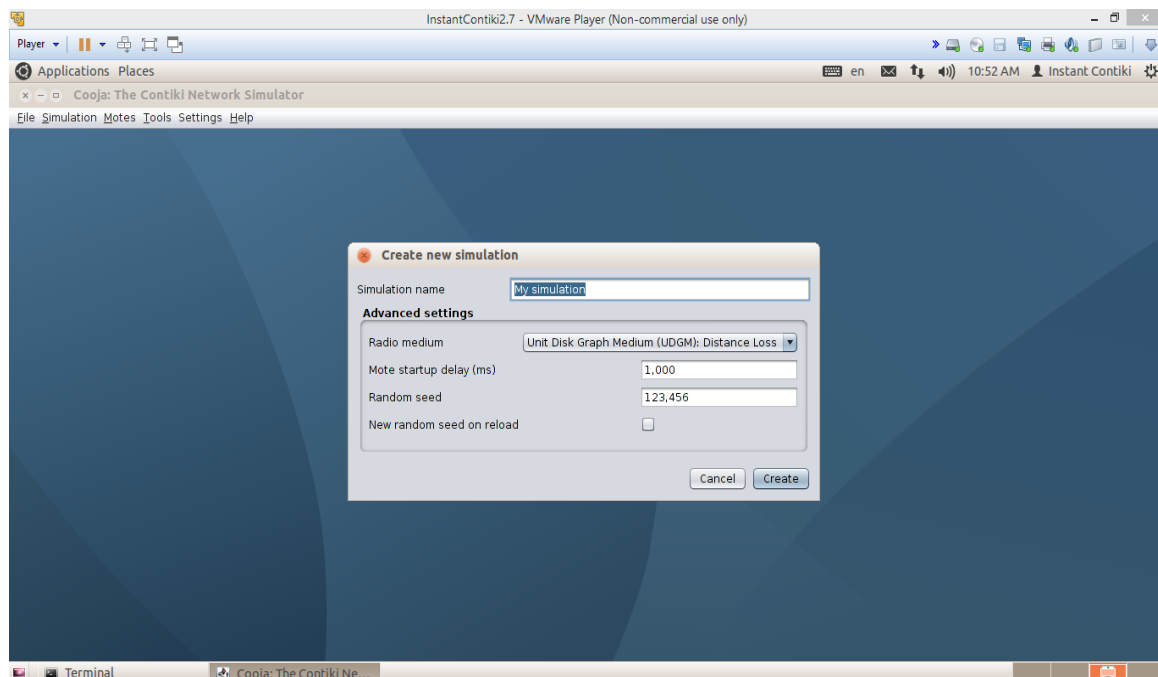


Figure 5.1: Creating a new simulation.

5.1.2 The Simulation Interface

The simulation interface, shown in Figure 5.2, consists of five windows. The Network window shows the physical layout of the network. We will be able to physically place motes here and move them around, as needed, in order to form the topology and layout we are interested in. The Simulation Control window lets us start, stop and reload the simulation. It also lets us control the rate at which the simulation proceeds. The Mote Output window shows any serial output generated by all the motes (e.g. the output from the `printf` command). We may filter the output shown based on the string we enter into the Filter field. For example, if we wish to filter the output such that it only shows output

5.1 Experiment Setup with Contiki

from mote 1, then we can enter ID:1 in this field. The Timeline window shows events that occur on each mote over the timeline of simulation. The Notes window can be used to take temporary notes in the simulation.

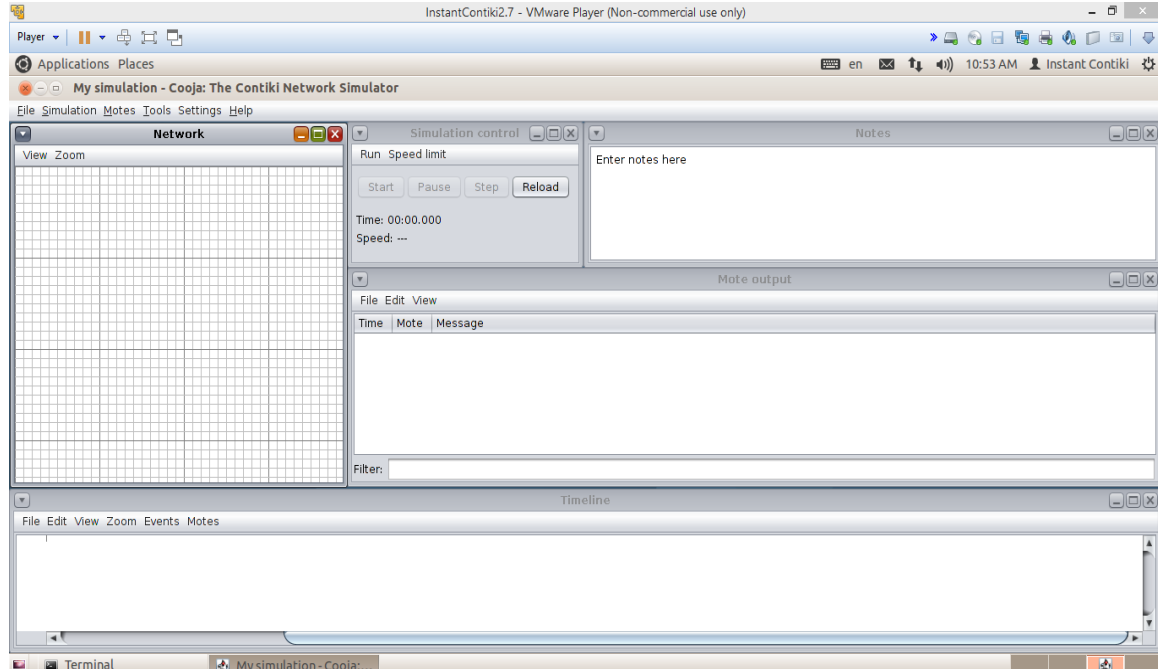


Figure 5.2: Cooja's window.

5.1.3 Setting Mote Types

The next step is to set mote type as shown in Figure 5.3. To do this, we click *Motes > AddMotes > CreateNewMoteType > SkyMote*. In the Contiki Process field, we should specify our source file (the .c file), Figure 5.4. If we specify the source code, then the compile commands field becomes active, as shown in Figure 5.5. We need to press Compile before creating the mote type. We can see the compilation output/results in the compilation output tab. If successful, the Create button becomes available. After creating a node, we can add one or more nodes in the simulation as shown in Figure 5.6, using the same mote type or creating more than one type. To start the simulation we can use start from the control panel. We can also save the configuration of the simulation, using the menu File then Save simulation. The simulations are saved in file extension .csc.

5.1 Experiment Setup with Contiki

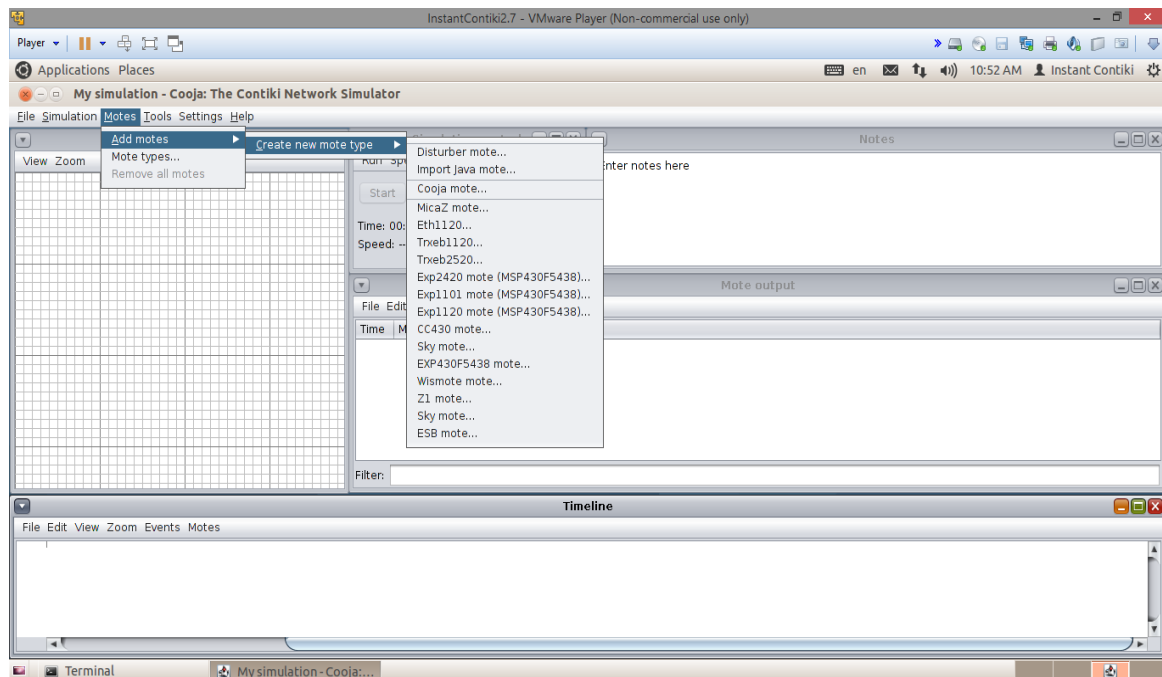


Figure 5.3: Create new mote type.

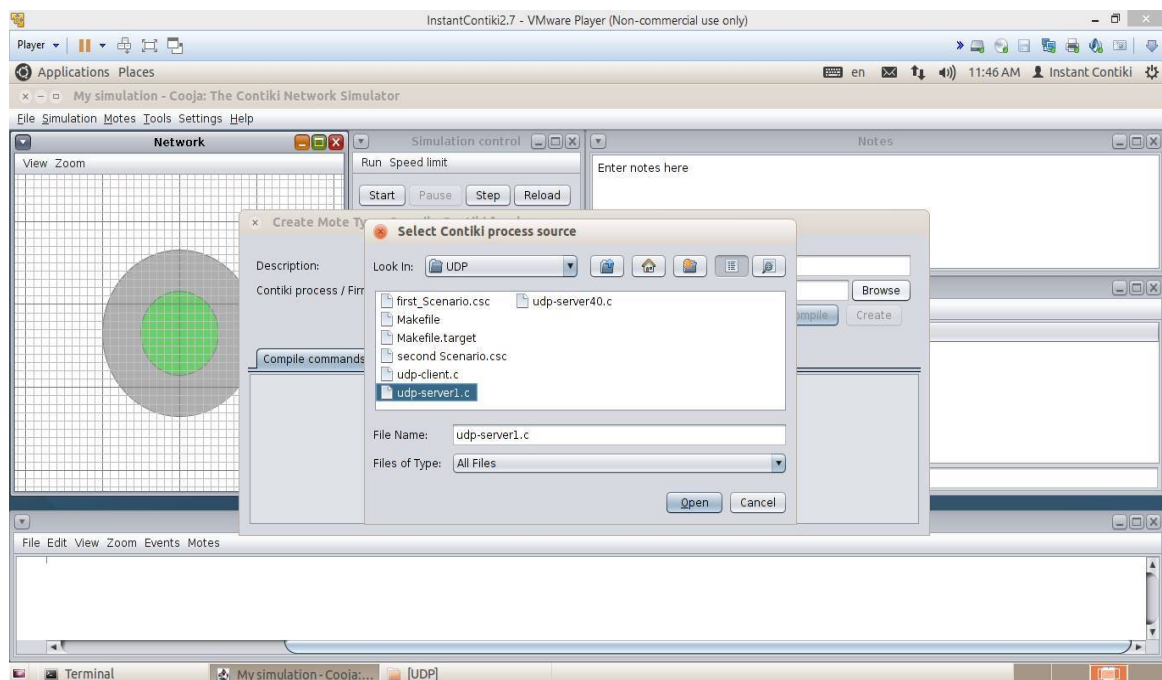


Figure 5.4: Select Contiki process source.

5.1 Experiment Setup with Contiki

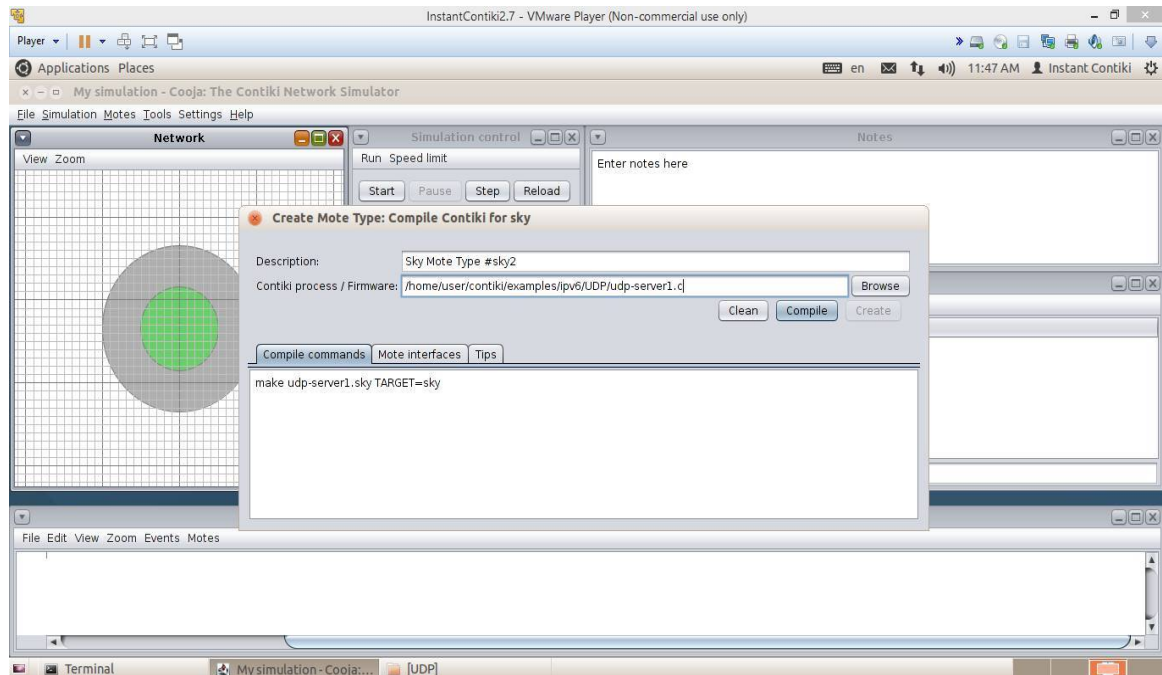


Figure 5.5: Compile the file.

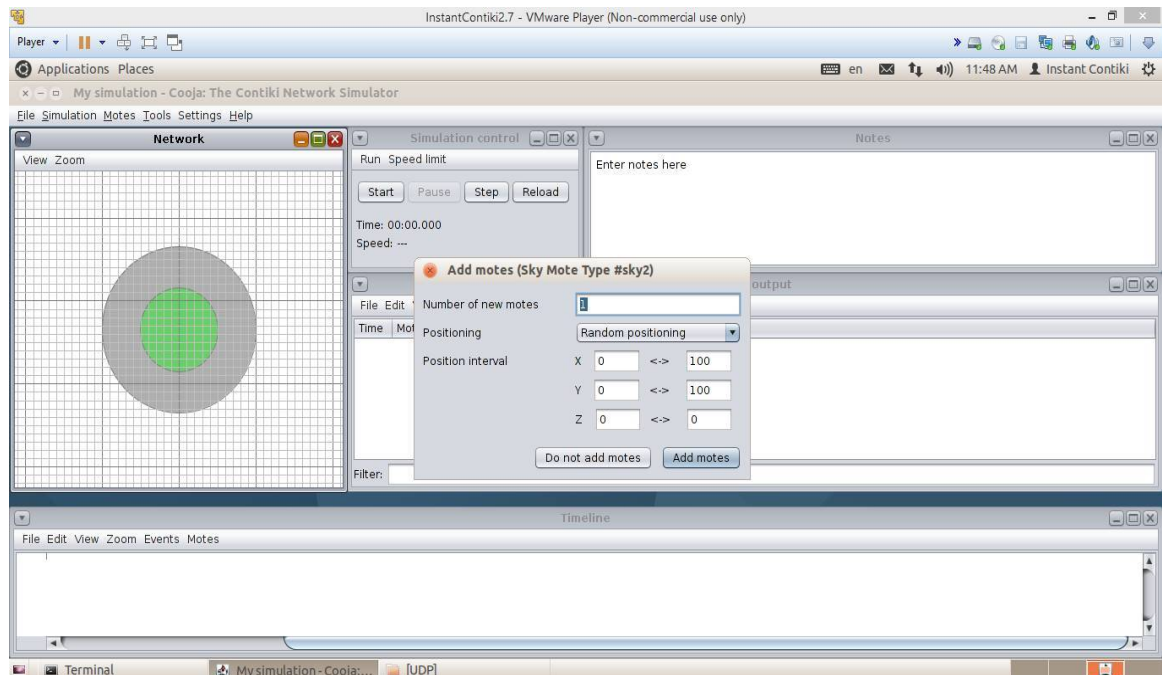


Figure 5.6: Adding notes.

5.1 Experiment Setup with Contiki

5.1.4 The Simulation Model

In this work, the RPL-UDP example was used as a basis. This example is allocated in folder */examples/ipv6/rpl-udp*, but we have modified those files in order to adapt them to our network and implement the cross-layer approach proposed, and discussed in the previous Chapter.

UDP is fully implemented in Contiki and implemented on top of RPL, and RPL is implemented on top of ContikiMAC radio duty cycling protocol. ContikiMAC and RPL are the default MAC and routing protocol used in Contiki. Experiments were carried out for different inter packet time intervals: 20 sec, 15 sec, 10 sec, and 5 sec. Experiment runs for a duration of 60 minutes. There are 40 nodes, 2 of them are edge routes and the other 38 are normal nodes. Each node sends the data packet towards one of the edge routers.

In order to implement the cross-layer approach, for reliability improvement, the default configuration of ContikiMAC had to be changed. By default all the senders ask for data link acknowledgements from the node receiving the packet, but code has been changed in order to set the data link acknowledgements only on the nodes selected by CPLEX when solving the mathematical formalization provided in Section 5.2 (page 34). CPLEX is a package from IBM/ILOG that is able to find the optimal solution for a mathematical problem formalization provided as input. The input information required to solve the mathematical problem, like nodes with drops above the threshold and set of potential critical nodes, were obtained by a first run of the simulation model, for both scenarios. Then a second run occurs after setting the data link acknowledgements at the appropriate nodes (nodes given by CPLEX after solving the mathematical problem), in order to observe the effect of acknowledgements on packet drop. The steps of the first and second runs are shown in Algorithms 1 and 2, respectively.

5.1 Experiment Setup with Contiki

```
/* General set up at edge router */
threshold ← 3;
simulation_time ← 3600 sec;
time_slot ← 360 sec;
for each arriving packet, denoted by p do
  /*record the packet arrival time*/
  t ← arrival time;
  /*record the sequence number and source ID of arriving packet */
  sn ← SequenceNumber(p);
  s_id ← SourceID(p);
  if sn = last_sequence_number[s_id] + 1 then
    /*there is no lost*/
    last_sequence_number[s_id] ← sn;
  end
  else
    /*accumulates drops for specific source node*/
    drops[s_id] ← (drops[s_id] + (sn - last_sequence_number[s_id]));
    last_sequence_number[s_id] ← sn;
  end
  if t > (last_time[s_id] + time_slot) then
    if drops[s_id] ≥ threshold then
      problematic_node[s_id] ← TRUE;
    end
    /*reset the timer and the drop counter*/
    last_time[s_id] ← t;
    drops[s_id] ← 0;
  end
end
end
```

Algorithm 1: Procedure to determine if a node is a problematic one. Runs at edge routers.

5.1 Experiment Setup with Contiki

*/*ACK activation is done just for the nodes been selected by the CPLEX; activation is done for all outgoing frames (current's node frames and neighbors' frames being forwarded) */*

```
for each outgoing frame f do  
    if current node ID belongs to set of nodes selected by CPLEX then  
        /*turns data link acknowledgment on*/  
        ActivateACK(f);  
    end  
    /*turns data link acknowledgment off*/  
    DeActivateACK(f);  
end
```

Algorithm 2: ACK activation/deactivation. Runs at every node.

RPL

A routing protocol is responsible for forwarding the packet from one node to another. That is, making the next step routing decision. Another important task of a routing protocol is to find the shortest possible path to reach the destinations and saving it in its routing table.

The objective of the ROLL *Working Group* (WG) was to design a routing protocol for LLNs supporting a variety of link layers, sharing the common characteristics of being low bandwidth, lossy and low power [26]. RPL was the result of this WG. RPL is a Distance Vector IPv6 routing for LLNs that builds the graph known as *Directed Acyclic Graph* (DAG) by using a set of metrics/constraints and an objective function. The objective function operates on a combination of metrics and constraints to compute the best path. Each node is assigned to a rank that is incremented as the node go far away from the sink.

Radio Duty Cycling

The MAC layer in contiki receives the incoming packet from the *Radio Duty Cycling* (RDC) and transmits the packet by using the RDC. The main challenge in WSNs is the energy, radio duty cycling mechanisms are used at the MAC layer to save energy consumption by reducing the idle listening time. Contiki provides many duty cycling mechanisms including ContikiMAC, and X-MAC.

5.2 Simulation Results

ContikiMAC

ContikiMAC is a radio duty cycling mechanism for low-power wireless networks. It is the default mechanism in Contiki and uses periodical wake-ups to listen for packet transmissions from neighbors. In ContikiMAC, nodes stay in sleep mode most of the time. If a packet transmission is detected, the receiver wakes up, receives the packet and then sends a link layer acknowledgement, if requested by the sender.

To guarantee the successful packet transmission, a sender periodically sends its packet until it receives a link layer acknowledgment from the receiver. ContikiMAC is very simple asynchronous mechanism, has a power-efficient wake-up mechanism and relies on different timing between transmissions. ContikiMAC uses the *Clear Channel Assessment* (CCA) in order to check the radio activity on the channel [8].

5.2 Simulation Results

Simulations were done using Cooja simulator, under Contiki OS, for the 40 wireless router networks in Figure 5.7 that were randomly generated using the weighted proximity algorithm presented in [19]. Figures 5.8 and 5.9 show both network scenarios in Cooja simulator. The ContikiMac and RPL are assumed for MAC and routing protocols. The first step of the approach was solved using the CPLEX optimization package, while the impact of the gradual implementation of link layer acknowledgements was evaluated in Cooja after setting at each step the acknowledgements for the nodes selected, which is determined by the first step in section 4.4 on page 24. The nodes selected by CPLEX for acknowledgement setting, and their order, were the following:

- **Scenario I:** 6, 10, 16, 22, 24, 36, 30
- **Scenario II:** 12, 7, 18, 34, 21, 26

5.2 Simulation Results

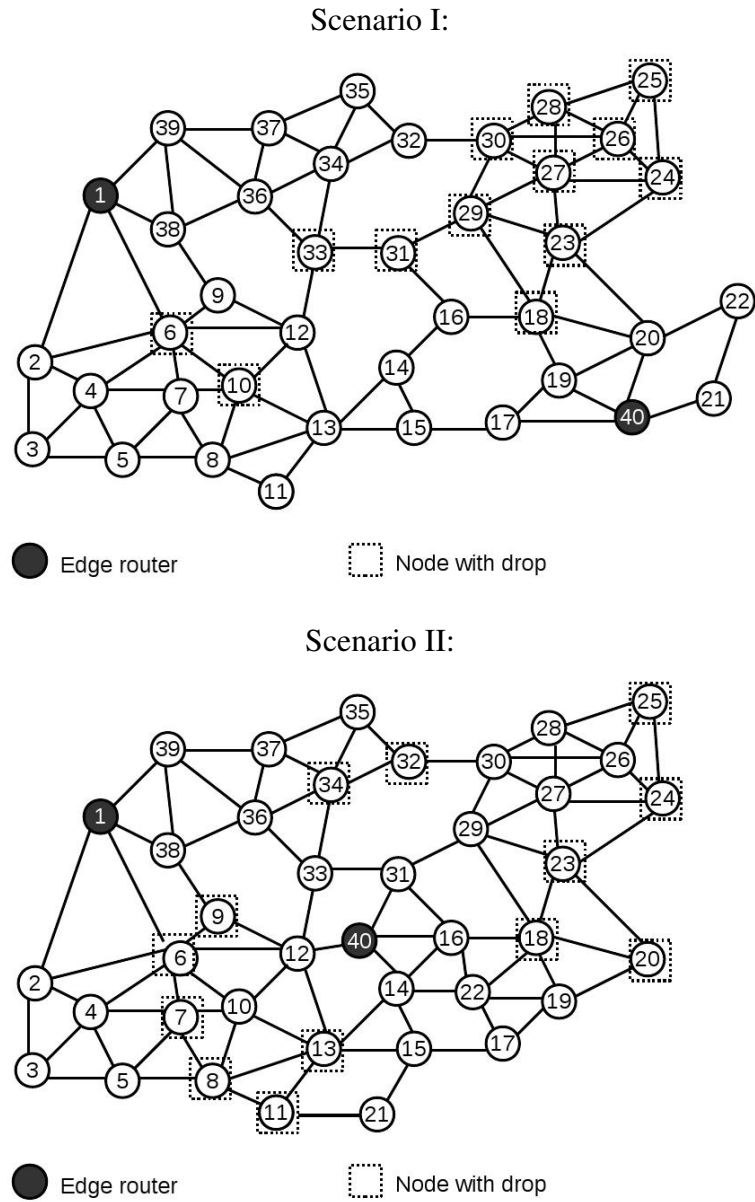


Figure 5.7: Randomly generated networks used in simulations.

5.2 Simulation Results

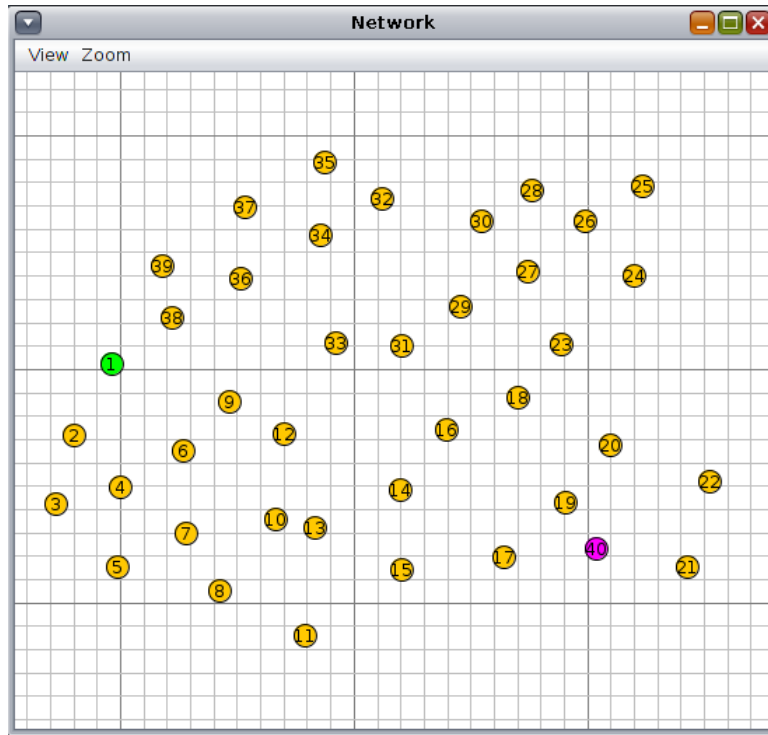


Figure 5.8: The Cooja network window for Scenario I.

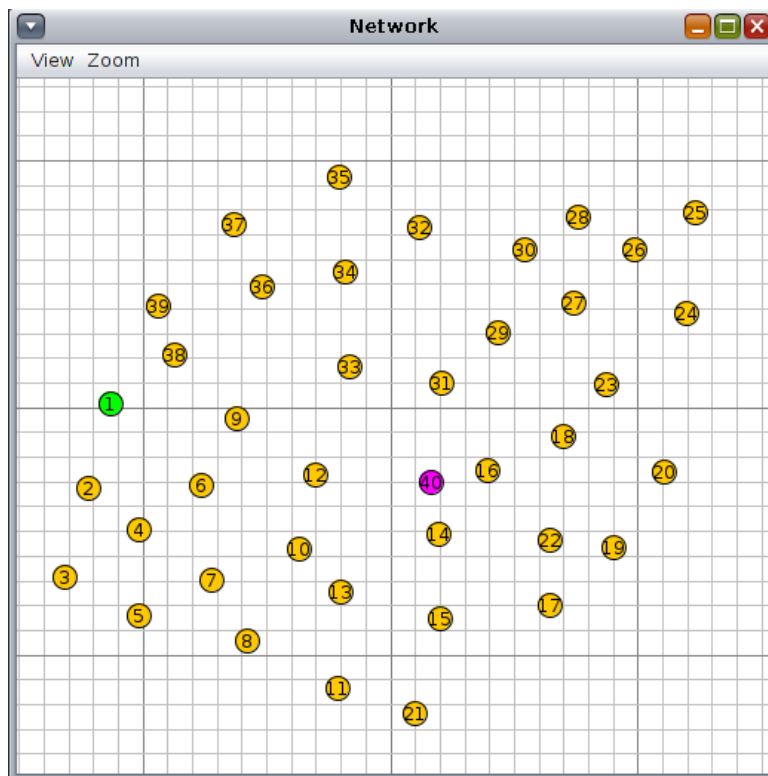


Figure 5.9: The Cooja network window for Scenario II.

The order for acknowledgement setting at nodes is related with the weight of nodes, as stated in section 4.4 on page 24. For both scenarios four load cases were tested: *i*) nodes

5.2 Simulation Results

generate a packet every 20 seconds; *ii*) nodes generate a packet every 15 seconds. *iii*) nodes generate a packet every 10 seconds. *iiii*) nodes generate a packet every 5 seconds. Simulation time took 1 hour, and a total of 6840, 9120, 13680 and 27360 packets were generated for the light and heavy load cases just mentioned, respectively. The results for Scenario I, considering four different loads, were the following:

Nodes with acknowledgement set	Edge Router 1	Edge Router 40	Edge Routers 1 and 40	Total sent
No ACK	1664	1737	3401	6797
6	2114	1339	3453	6799
6-10	2051	1323	3374	6799
6-10-16	1988	1345	3333	6798
6-10-16-22	2240	1413	3653	6800
6-10-16-22-24	2718	1326	4044	6797
6-10-16-22-24-36	2304	1452	3756	6792
6-10-16-22-24-36-30	2457	1592	4049	6794

Table 5.1: Results for Scenario I with nodes generating a packet every 20s.

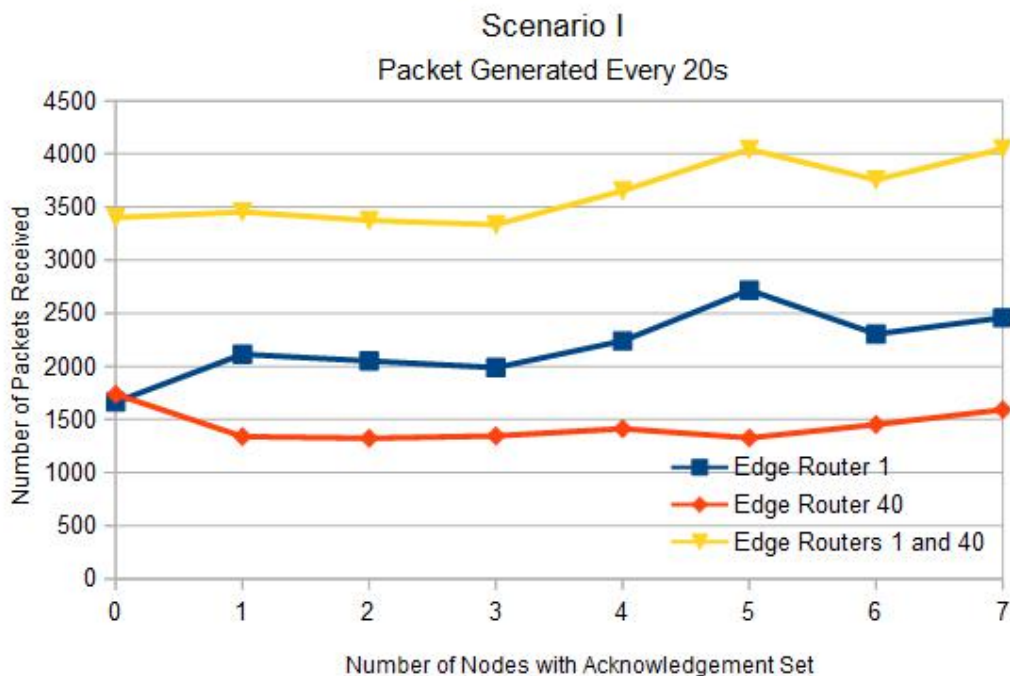


Figure 5.10: Results for Scenario I with nodes generating a packet every 20s.

5.2 Simulation Results

Nodes with acknowledgement set	Edge Router 1	Edge Router 40	Edge Routers 1 and 40	Total sent
No ACK	1672	1872	3544	9066
6	1980	1949	3929	9069
6-10	2651	1605	4256	9072
6-10-16	2061	1763	3824	9069
6-10-16-22	1942	1941	3883	9071
6-10-16-22-24	2084	1769	3853	9068
6-10-16-22-24-36	2406	1888	4294	9072
6-10-16-22-24-36-30	3279	1545	4824	9071

Table 5.2: Results for Scenario I with nodes generating a packet every 15s.

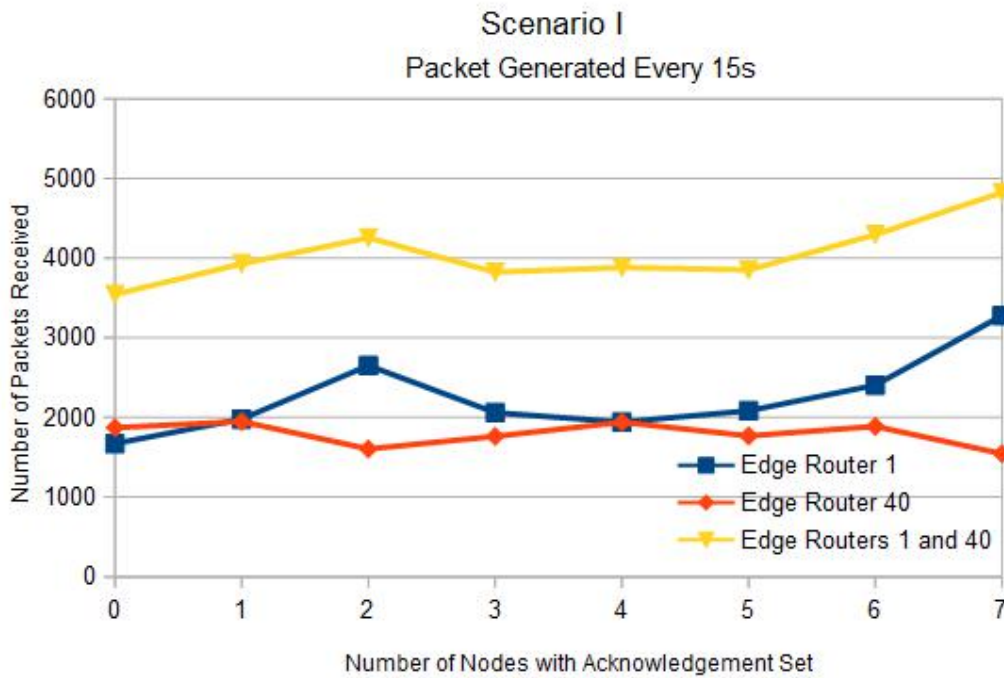


Figure 5.11: Results for Scenario I with nodes generating a packet every 15s.

5.2 Simulation Results

Nodes with acknowledgement set	Edge Router 1	Edge Router 40	Edge Routers 1 and 40	Total sent
No ACK	1767	1930	3697	13618
6	2205	2227	4432	13622
6-10	1964	2234	4198	13628
6-10-16	2935	2220	5155	13617
6-10-16-22	1972	2369	4341	13626
6-10-16-22-24	2175	2345	4520	13627
6-10-16-22-24-36	3194	2287	5481	13627
6-10-16-22-24-36-30	2313	2318	4631	13621

Table 5.3: Results for Scenario I with nodes generating a packet every 10s.

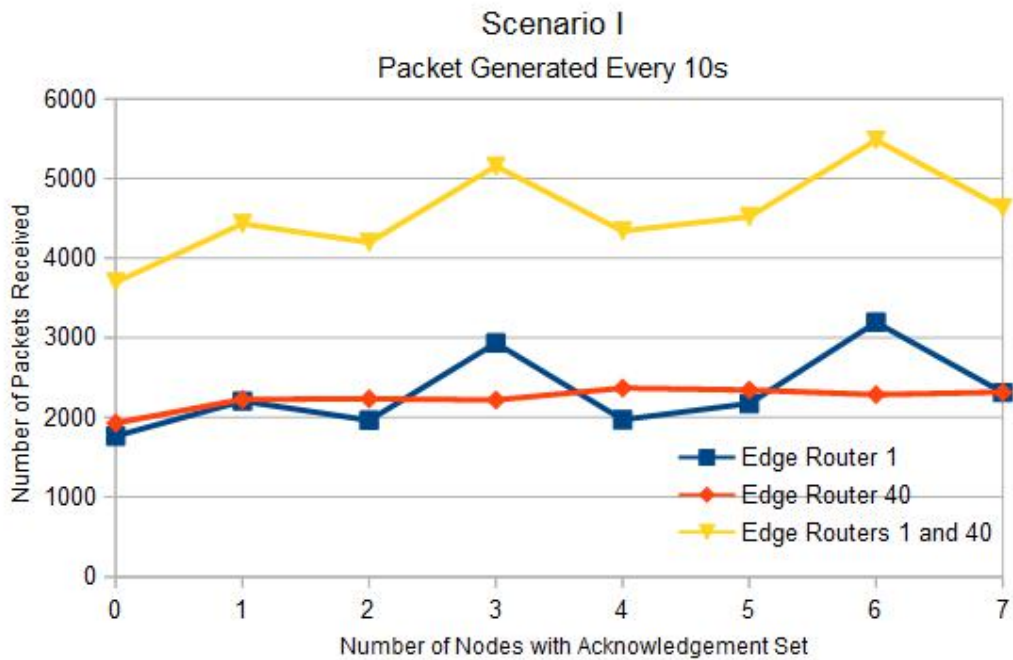


Figure 5.12: Results for Scenario I with nodes generating a packet every 10s.

5.2 Simulation Results

Nodes with acknowledgement set	Edge Router 1	Edge Router 40	Edge Routers 1 and 40	Total sent
No ACK	1920	2834	4754	27240
6	2256	2686	4942	27254
6-10	2148	2945	5093	27252
6-10-16	3825	2583	6408	27255
6-10-16-22	2076	3254	5330	27259
6-10-16-22-24	5488	2828	8316	27267
6-10-16-22-24-36	4092	3192	7284	27277
6-10-16-22-24-36-30	4095	2651	6746	27275

Table 5.4: Results for Scenario I with nodes generating a packet every 5s.

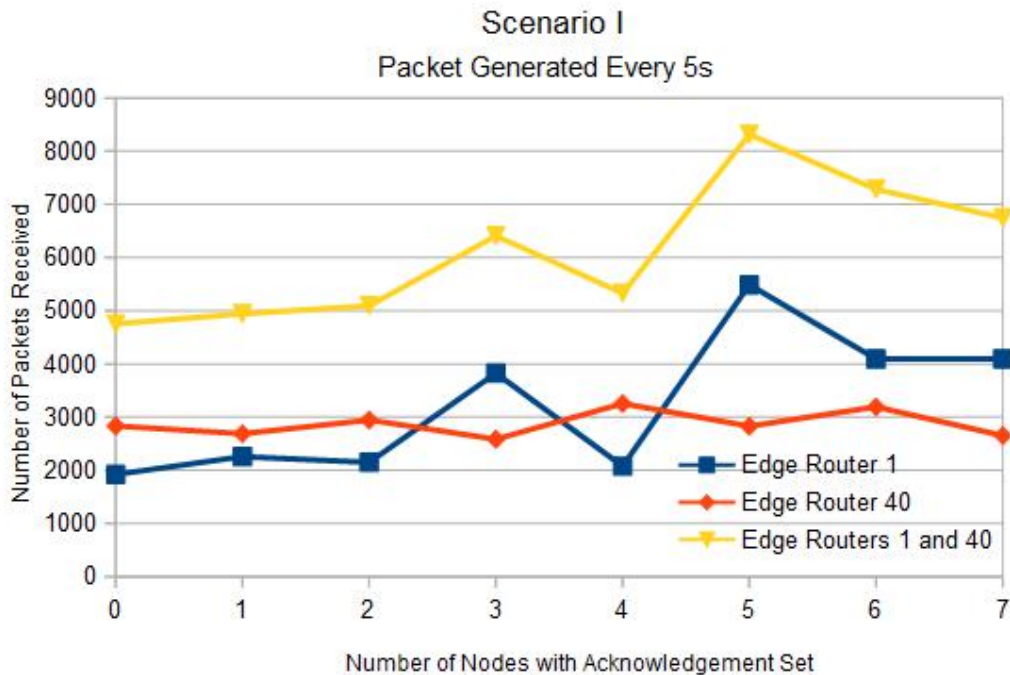


Figure 5.13: Results for Scenario I with nodes generating a packet every 5s.

Figures 5.10, 5.11, 5.12 and 5.13, relating Scenario I, consider four different loads. These plots show that the number of packets received at edge routers 1 and 40 increase when sparse acknowledgement is used by setting acknowledgement at the selected nodes, when compared with no ack (number of nodes performing acknowledgement is equal to zero). Such increase is not the same over time, but it was possible to get a maximum total increase of 10% for the more lightly loaded (packet every 20s) case and 13% for the other case (packet every 5s). It is also possible to see that there was a higher positive effect on

5.2 Simulation Results

traffic sent toward edge router 1, while traffic sent to edge router 40 remains more or less the same. The results concerning Scenario II follow.

Nodes with acknowledgement set	Edge Router 1	Edge Router 40	Edge Routers 1 and 40	Total sent
No ACK	2253	2141	4394	6802
12	1103	3875	4978	6800
12-7	967	4117	5084	6803
12-7-18	898	4381	5279	6805
12-7-18-34	544	5267	5811	6799
12-7-18-34-21	555	5314	5869	6802
12-7-18-34-21-26	551	5767	6318	6799

Table 5.5: Results for Scenario II with nodes generating a packet every 20s.

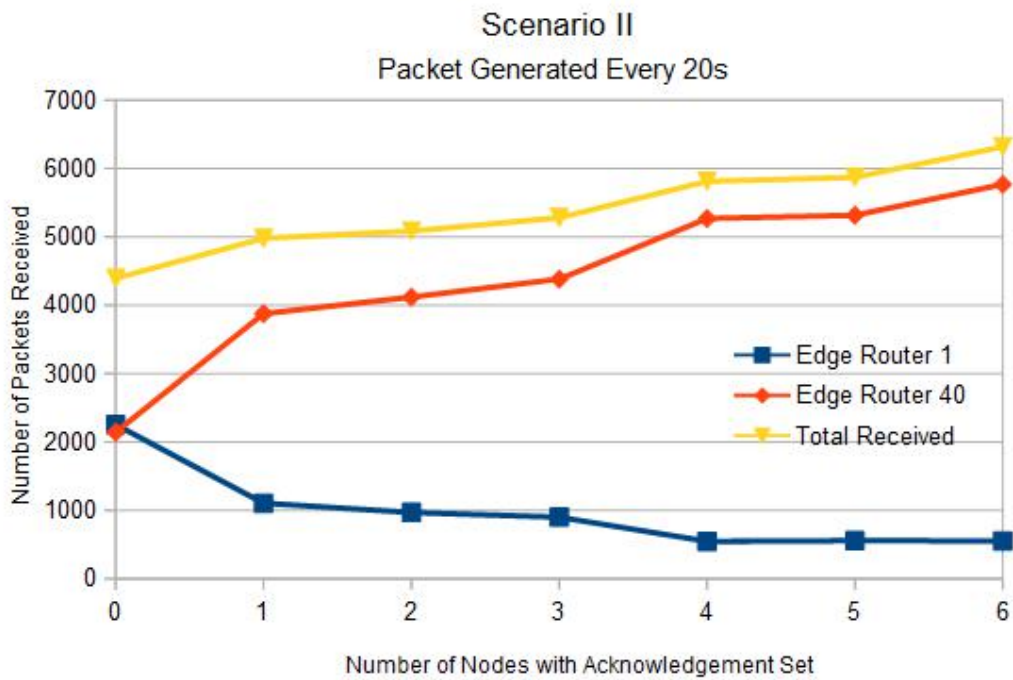


Figure 5.14: Results for Scenario II with nodes generating a packet every 20s.

5.2 Simulation Results

Nodes with acknowledgement set	Edge Router 1	Edge Router 40	Edge Routers 1 and 40	Total sent
No ACK	1701	1852	3553	9067
12	1442	4872	6314	9068
12-7	1062	5206	6268	9067
12-7-18	1232	5521	6753	9070
12-7-18-34	730	6314	7044	9069
12-7-18-34-21	734	6247	6981	9072
12-7-18-34-21-26	718	6723	7441	9071

Table 5.6: Results for Scenario II with nodes generating a packet every 15s.

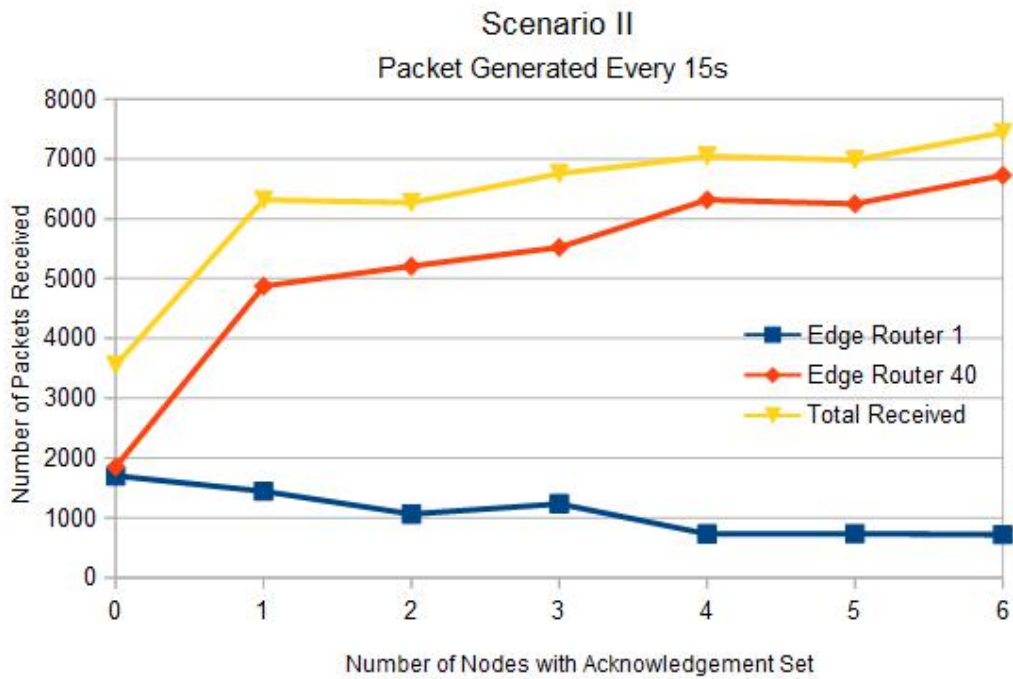


Figure 5.15: Results for Scenario II with nodes generating a packet every 15s.

5.2 Simulation Results

Nodes with acknowledgement set	Edge Router 1	Edge Router 40	Edge Routers 1 and 40	Total sent
No ACK	1853	2138	3991	13623
12	1114	6314	7428	13622
12-7	1218	6906	8124	13618
12-7-18	1551	7219	8770	13622
12-7-18-34	1133	8148	9281	13627
12-7-18-34-21	1106	8524	9630	13627
12-7-18-34-21-26	1090	8831	9921	13626

Table 5.7: Results for Scenario II with nodes generating a packet every 10s.

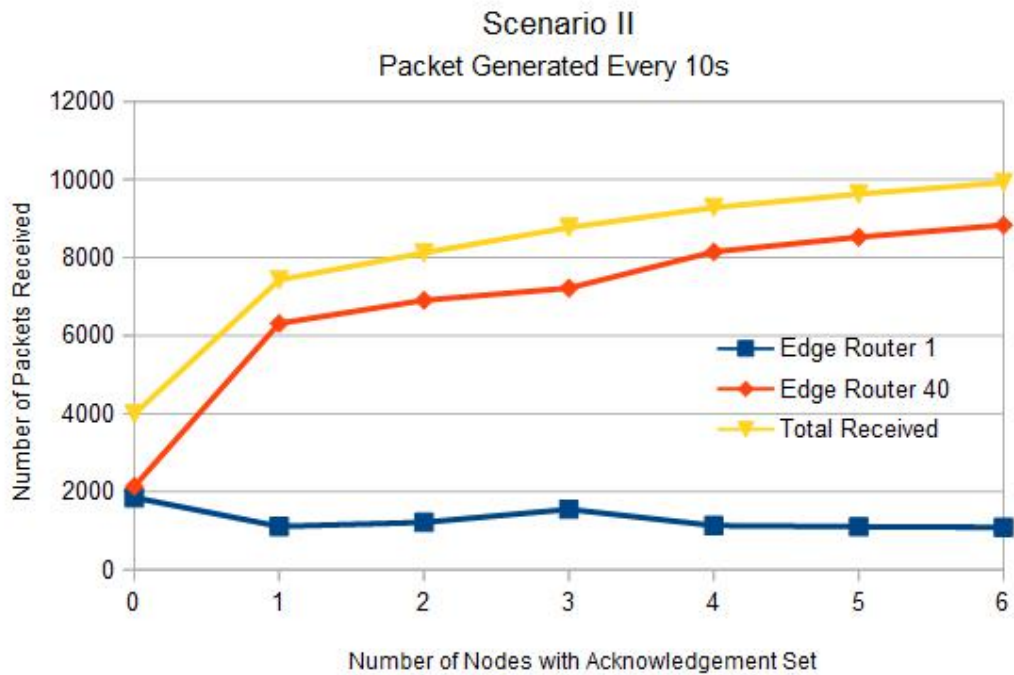


Figure 5.16: Results for Scenario II with nodes generating a packet every 10s.

5.2 Simulation Results

Nodes with acknowledgement set	Edge Router 1	Edge Router 40	Edge Routers 1 and 40	Total sent
No ACK	1902	2173	4075	27260
12	1760	8037	9797	27241
12-7	1757	10273	12030	27261
12-7-18	1582	10556	12138	27255
12-7-18-34	2143	13144	15287	27267
12-7-18-34-21	2101	13150	15251	27282
12-7-18-34-21-26	1911	13418	15329	27276

Table 5.8: Results for Scenario II with nodes generating a packet every 5s.

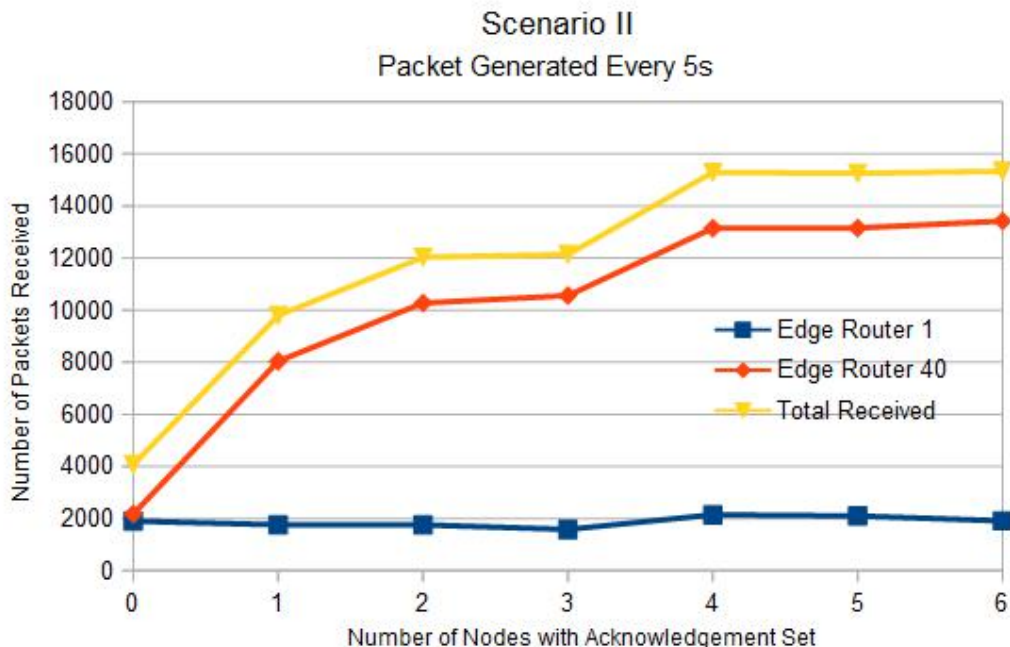


Figure 5.17: Results for Scenario II with nodes generating a packet every 5s.

Figures 5.14, 5.15, 5.16 and 5.17 relate to Scenario II considering the four different loads also used for Scenario I. These plots also show that sparse acknowledgement increases the number of packets received at edge routers, and in this case the maximum total increase was of 28% for the more lightly loaded (packet every 20s) case, 43% for (packet every 15s) case, 44% for (packet every 10s) case and 41% for the other case (packet every 5s). However, contrarily to Scenario I the increase on packet delivery was higher for traffic send toward edge router 40, while traffic send to edge router 1 remains more or less the same or even decreases. This might have to do with the fact that most of

5.2 Simulation Results

packet drops relate to traffic being routed to edge router 40, while in the Scenario I packet drops relate to traffic being routed equally to edge routers 1 and 40.

From all the plots, and since nodes generate equal amount of traffic meaning that the load would be the same throughout the network, we can point out that the location of critical areas in a network and the routing have effect on the overall performance of this approach. Also, the weight assigned to nodes, which determines the order for acknowledgement setting, is also a relevant issue. In scenario I traffic is sent to edge routers in an equal way, but benefits were more visible at traffic directed to edge router 1. This might be influenced by the order of acknowledgement setting done, which can be influenced by the weight besides the strategy chosen (critical area coverage approach).

In summary, we can say that the approach used, which does not select neighbour nodes to implement acknowledgement in order to avoid congestion at specific areas, can be a good approach since it was possible to deliver more packets. However, other strategies might be explored in the future. Note that this approach is not intended to full network acknowledgement, which might be required by some applications.

Conclusions and Future Work

Cross-layer design is strongly recommended as a new methodology for designing and optimizing the performance of wireless networks. This work presents a cross-layer optimization approach to improve reliability in networks using 6LoWPAN. 6LoWPAN implementation is based on RFC4944 Transmission of IPv6 over IEEE 802.15.4 networks, where IEEE 802.15.4 is a standard that specifies the physical and MAC layer for WPAN. Simulations were done using Cooja simulator, under ContikiOS. Results show that selecting a small set of critical node, where link layer acknowledgement requests are to be introduced, can reduce packet drops significantly while not imposing too much congestion into the network. As the benefit changes over time, with increases and decreases taking place, it is still necessary to analyse different rules for node selection, and their usefulness under different network scenarios. The influence of routing protocols can also be analysed in the future.

References

- [1] Ian F Akyildiz, Weilian Su, Yogesh Sankarasubramaniam, and Erdal Cayirci. A survey on sensor networks. *Communications magazine, IEEE*, 40(8):102–114, 2002.
- [2] Irfan Al-Anbagi, Melike Erol Kantarci, and Hussein T Mouftah. A survey on cross-layer quality of service approaches in wsns for delay and reliability aware applications.
- [3] Luigi Atzori, Antonio Iera, and Giacomo Morabito. The internet of things: A survey. *Computer networks*, 54(15):2787–2805, 2010.
- [4] Nik Bessis and Ciprian Dobre. *Big Data and Internet of Things: A Roadmap for Smart Environments*. Springer, 2014.
- [5] Aminul Haque Chowdhury, Muhammad Ikram, Hyon-Soo Cha, Hassen Redwan, SM Shams, Ki-Hyung Kim, and Seung-Wha Yoo. Route-over vs mesh-under routing in 6lowpan. In *Proceedings of the 2009 International Conference on Wireless Communications and Mobile Computing: Connecting the World Wirelessly*, pages 1208–1212. ACM, 2009.
- [6] LAN/MAN Standards Committee et al. Part 15.4: wireless medium access control (mac) and physical layer (phy) specifications for low-rate wireless personal area networks (lr-wpans). *IEEE Computer Society*, 2003.
- [7] David Culler and Berkeley Samita Chakrabarti. 6lowpan: Incorporating ieee 802.15.4 into the ip architecture. 2009.
- [8] Adam Dunkels. The contikimac radio duty cycling protocol. 2011.
- [9] Emad Felemban, Chang-Gun Lee, Eylem Ekici, Ryan Boder, and Serdar Vural. Probabilistic qos guarantee in reliability and timeliness domains in wireless sensor networks. In *INFOCOM 2005. 24th Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings IEEE*, volume 4, pages 2646–2657. IEEE, 2005.
- [10] Fei Gao, Hongli Wen, Lifan Zhao, and Yuebin Chen. Design and optimization of a cross-layer routing protocol for multi-hop wireless sensor networks. In *Sensor*

References

- Network Security Technology and Privacy Communication System (SNS & PCS), 2013 International Conference on*, pages 5–8. IEEE, 2013.
- [11] Coalesenses Gmbh. iSense Core Module 3 Datasheet. pages 1–13.
- [12] Nurul Halimatul Asmak Ismail, Rosilah Hassan, and Khadijah Wan Mohd Ghazali. A study on protocol stack in 6lowpan model. *Journal of Theoretical and Applied Information Technology*, 41(2):220–229, 2012.
- [13] Peng Ji, Chengdong Wu, Yunzhou Zhang, and Xiaozhe Wang. A cross-layer power controlled mac protocol in wireless sensor networks. In *Wireless Communications, Networking and Mobile Computing, 2008. WiCOM'08. 4th International Conference on*, pages 1–4. IEEE, 2008.
- [14] Young-Bae Ko and Nitin H Vaidya. Location-aided routing (lar) in mobile ad hoc networks. *wireless networks*, 6(4):307–321, 2000.
- [15] Xianhu Ma, Li Xu, and Geyong Min. Congestion control based on cross-layer game optimization in wireless mesh networks. In *Mobile Ad-hoc and Sensor Networks (MSN), 2013 IEEE Ninth International Conference on*, pages 41–46. IEEE, 2013.
- [16] Tommaso Melodia and Ian F Akyildiz. Cross-layer qos-aware communication for ultra wide band wireless multimedia sensor networks. *Selected Areas in Communications, IEEE Journal on*, 28(5):653–663, 2010.
- [17] Gabriel Montenegro, Nandakishore Kushalnagar, J Hui, and D Culler. Transmission of ipv6 packets over ieee 802.15. 4 networks. *Internet proposed standard RFC*, 4944, 2007.
- [18] Thomas Narten, William Allen Simpson, Erik Nordmark, and Hesham Soliman. Neighbor discovery for ip version 6 (ipv6). 2007.
- [19] Furuzan Atay Onat and Ivan Stojmenovic. Generating random graphs for wireless actuator networks. In *World of Wireless, Mobile and Multimedia Networks, 2007. WoWMoM 2007. IEEE International Symposium on a*, pages 1–12. IEEE, 2007.
- [20] Fredrik Osterlind and Adam Dunkels. Contiki cooja hands-on crash course: Session notes.
- [21] Ji Peng, Jiang Jingqi, Sun Qiushuo, and Zhang Songyang. A noble cross-layer protocol for qos optimization in wireless sensor networks. In *Control and Decision Conference (2014 CCDC), The 26th Chinese*, pages 2430–2434. IEEE, 2014.

References

- [22] Hee-Tae Roh and Jang-Won Lee. Cross-layer optimization for wireless sensor networks with rf energy transfer. In *Information and Communication Technology Convergence (ICTC), 2014 International Conference on*, pages 919–923, Oct 2014.
- [23] Ghalib Asadullah Shah, Vehbi Cagri Gungor, and Özgür B Akan. A cross-layer qos-aware communication framework in cognitive radio sensor networks for smart grid applications. *IEEE Trans. Industrial Informatics*, 9(3):1477–1485, 2013.
- [24] Zach Shelby and Carsten Bormann. *6LoWPAN: The wireless embedded Internet*, volume 43. Wiley. com, 2011.
- [25] Lei Shi, Jiang-Hong Han, Yi Shi, and Zhen-Chun Wei. Cross-layer optimization for wireless sensor network with multi-packet reception. In *Communications and Networking in China (CHINACOM), 2010 5th International ICST Conference on*, pages 1–5. IEEE, 2010.
- [26] J Vasseur, Navneet Agarwal, Jonathan Hui, Zach Shelby, Paul Bertrand, and Cedric Chauvenet. Rpl: The ip routing protocol designed for low power and lossy networks. *Internet Protocol for Smart Objects (IPSO) Alliance*, 2011.
- [27] Jean-Philippe Vasseur and Adam Dunkels. *Interconnecting smart objects with ip: The next internet*. Morgan Kaufmann, 2010.
- [28] Mehmet C Vuran and Ian F Akyildiz. Error control in wireless sensor networks: a cross layer analysis. *Networking, IEEE/ACM Transactions on*, 17(4):1186–1199, 2009.
- [29] W. Xu, Y. Zhang, Q. Shi, and X. Wang. Energy management and cross layer optimization for wireless sensor network powered by heterogeneous energy sources. *Wireless Communications, IEEE Transactions on*, PP(99):1–1, 2015.
- [30] Jennifer Yick, Biswanath Mukherjee, and Dipak Ghosal. Wireless sensor network survey. *Computer networks*, 52(12):2292–2330, 2008.

List of Publications

1. N. Hasan, M. Ali, A. Barradas and N. Correia, "Cross-Layer Optimization for Reliability Improvement of Data Delivery in 6LoWPAN-based Networks", IEEE Med-Hoc-Net' 15, 17–19 June, Vilamoura, Portugal (accepted).
2. M. Ali, N. Hasan and N. Correia, "Fairness for CoAP/Observe Based Wireless Sensor Networks With Aggregation Deployment", IEEE Globecom' 15, San Diego, USA (submitted).