



Cape Peninsula
University of Technology

**CONTENT DISTRIBUTION FRAMEWORK FOR WIRELESS MESH
NETWORKS : An Information-Centric Approach**

by

THOMAS ALWALA GONE

Thesis submitted in fulfilment of the requirements for the degree

Master of Technology: Information Technology

in the Faculty of Informatics and Design

at the Cape Peninsula University of Technology

Supervisor: Dr. Boniface Kabaso

Cape Town

May 2017

CPUT copyright information

The dissertation/thesis may not be published either in part (in scholarly, scientific or technical journals), or as a whole (as a monograph), unless permission has been obtained from the University

Declaration

I, **Gone Thomas Alwala**, declare that the contents of this dissertation/thesis represent my own unaided work, and that the dissertation/thesis has not previously been submitted for academic examination towards any qualification. Furthermore, it represents my own opinions and not necessarily those of the Cape Peninsula University of Technology.

--	--

Signed

Date

Abstract

Information Centric Networking (ICN) is an emerging research field that aims to replace the current host-centric model of Content distribution on the Internet. In ICN architectures, network devices not only forward Content, but also store it thus effectively distributing the traffic load and making access to Content faster. In this thesis, the ability to utilize ICN as the content distribution scheme in WMNs is explored.

The main contribution is the development of IcnMesh – a simulator that integrates a Content-Centric scheme in Wireless Mesh Networks thus allowing the study of ICN Content distribution schemes in WMNs. Through simulation experiments, it is shown that ICN schemes can significantly enhance the performance of a WMN and eliminate some of their existing shortcomings.

Acknowledgements

My sincere gratitude to my supervisor, Dr. Boniface Kabaso, for his endless support and encouragement.

Dedication

To my parents, Victoria and Francis Gone.

Table of Contents

Declaration.....	i
Abstract.....	ii
Acknowledgements.....	iii
Dedication.....	iv
List of Figures.....	x
List of Tables.....	xii
List of Appendices.....	xiii
List of Abbreviations.....	xiv
Publications.....	xvi
CHAPTER 1.....	1
1 INTRODUCTION.....	1
1.1 Objectives.....	4
1.2 Research Aims.....	5
1.3 Problem Statement.....	5
1.4 Research Question.....	5
1.4.1 Research Sub-questions.....	5
1.5 Research Methodology.....	6
1.6 Thesis Organization.....	6
1.7 Summary of Chapter 1.....	7
CHAPTER TWO.....	8
2 LITERATURE REVIEW.....	8
2.1 Wireless Mesh Networks.....	8
2.1.1 Introduction.....	8
2.1.1.1 IEEE 802.11 Standard.....	8
2.1.1.2 Wireless Mesh Networks Architectures.....	10
2.1.1.3 Applications of Wireless Mesh Networks.....	12
2.1.2 WMN Physical Layer.....	12
2.1.2.1 Radio-Frequency Systems.....	12
2.1.2.2 Antenna Systems.....	13
2.1.2.3 MultiChannel Systems.....	14
2.1.3 WMN Medium Access Layer.....	15
2.1.4 Network Layer.....	15
2.1.4.1 Optimized Link-state Routing Protocol (OLSR).....	17
2.1.4.2 Better Approach To Mobile ad hoc Networking (BATMAN).....	18

2.1.5	Wireless Mesh Network Capacity.....	20
2.1.6	Standards on WMNs.....	20
2.1.6.1	IEEE 802.11s Standard.....	21
2.2	Content Distribution Schemes.....	22
2.2.1	Centralized File Distribution.....	22
2.2.2	Content Distribution Networks.....	23
2.2.3	Multicast Communication.....	25
2.2.4	Peer-to-peer Schemes.....	26
2.2.5	Network Coding Schemes.....	27
2.3	Information-Centric Networking.....	27
2.3.1	Content Naming.....	27
2.3.2	Content Routing.....	28
2.3.3	Advantages of Information-Centric Networks.....	29
2.3.4	Main ICN Architectures.....	30
2.3.4.1	Data-Oriented Network Architecture (DONA).....	30
2.3.4.2	Content-Centric Networking (CCN).....	31
2.3.4.3	Publish-Subscribe Internet Routing Paradigm (PSIRP).....	32
2.3.4.4	Network of Information (NetInf).....	33
2.4	Content Distribution Schemes in WMNs.....	33
2.4.1	Overlay Networks/Peer-to-Peer schemes.....	33
2.4.2	Replication Schemes.....	34
2.4.3	Caching Schemes.....	36
2.4.4	Other WMN Content Distribution Schemes.....	37
2.4.4.1	Content Prefetching.....	37
2.5	Summary of Related Work.....	37
2.6	Summary of Chapter.....	39
	CHAPTER THREE.....	41
3	RESEARCH METHODOLOGY AND DESIGN.....	41
3.1	Research Approach.....	41
3.1.1	Conceptual framework.....	43
3.1.2	Overview of Simulation As A Research Tool.....	43
3.1.2.1	Purpose of Simulation.....	44
3.1.3	Research Approach Using Simulation.....	45
3.1.3.1	Conceptual Design.....	45
3.1.3.2	Code Development.....	45
3.1.3.3	Validation.....	45
3.1.3.4	Experimental Design.....	46

3.1.3.5	Implementation.....	46
3.1.3.6	Analysis.....	46
3.1.3.7	Interpretation.....	46
3.2	Research Design.....	46
3.2.1	Design Requirements.....	46
3.2.1.1	Network Design Requirements.....	47
3.2.1.2	Content Model Requirement.....	47
3.2.1.3	Client Behaviour Requirement.....	47
3.2.1.4	User Interface Requirement.....	47
3.2.1.5	Data capture and display requirement.....	48
3.3	Research Metrics.....	48
3.3.1	Hit Ratio.....	48
3.3.2	Network Load Reduction.....	48
3.3.3	Average Throughput.....	49
3.3.4	Hop-Count.....	49
3.3.5	Server Load Reduction.....	50
3.3.6	Latency.....	50
3.3.7	Communication Overhead.....	50
3.4	Data collection/gathering method.....	51
3.5	Data analysis methods.....	51
3.6	Summary of chapter 3.....	51
	CHAPTER FOUR.....	52
4	SYSTEM IMPLEMENTATION.....	52
4.1	Development Environment Selection.....	52
4.1.1	User Interface.....	52
4.1.2	Network Simulator.....	52
4.1.2.1	OMNET++.....	52
4.1.2.2	GloMoSim.....	53
4.1.2.3	Network Simulator 3.....	53
4.2	Development Process.....	53
4.3	Resource Requirements.....	54
4.4	Architectural Overview.....	55
4.5	Application Classes.....	56
4.5.1	IcnMesh Class.....	56
4.5.2	IcnSimulator Class.....	59
4.5.3	MeshCreator Class.....	60
4.5.4	MeshNode Class.....	60

4.5.5 Client Node Class.....	61
4.5.6 ServerNode Class.....	61
4.5.7 Content Object.....	62
4.6 Network Simulator 3 API.....	62
4.6.1 ndnSIM.....	63
4.7 IcnMesh User Interface.....	65
4.7.1 Map Display.....	66
4.7.2 MenuBar.....	67
4.7.3 ToolBar.....	67
4.7.4 Status bar.....	67
4.8 IcnMesh Network Model.....	68
4.8.1 Wireless Channel.....	68
4.8.2 Client behaviour.....	69
4.8.3 Content model.....	69
4.8.4 Routing.....	69
4.9 Measurement System.....	71
4.9.1 Data Collection System.....	71
4.9.1.1 Packet Capture.....	71
4.9.1.2 Packet Tracing.....	72
4.9.2 Data Analysis Tools.....	72
4.9.3 Wireshark.....	72
4.9.4 R-Script.....	72
4.10 Running a Simulation.....	73
4.11 Summary of Chapter 4.....	74
CHAPTER FIVE.....	76
5 RESULTS AND EVALUATION.....	76
5.1 Validation of Simulator.....	76
5.1.1 Simple Host-Centric Network.....	76
5.1.1.1 Objectives.....	76
5.1.1.2 Network Setup.....	77
5.1.2 Simulator Configuration.....	78
5.1.2.1 Results and Discussion.....	78
5.1.3 Simple Information-Centric Network.....	81
5.1.3.1 Objectives.....	81
5.1.3.2 Network Setup.....	81
5.1.3.3 Simulator Configuration.....	82
5.1.3.4 Results.....	82

5.2 Routing in Information-Centric Wireless Mesh Network.....	88
5.2.1 Objectives.....	88
5.2.2 Network Setup.....	88
5.2.3 Routing with 802.11 MAC.....	89
5.2.3.1 Simulation Configuration.....	89
5.2.3.2 Results and Discussion.....	90
5.2.4 Routing with Modified 802.11s MAC.....	94
5.2.4.1 Simulation Configuration.....	94
5.2.4.2 Results and Discussion.....	95
5.3 Content Distribution to Multiple Consumers with Single Producer.....	98
5.3.1 Objectives.....	98
5.3.2 Network Setup.....	98
5.3.3 Simulator Configuration.....	99
5.3.4 Results and Discussion.....	100
5.4 Distribution of Content with Varying Popularity in IC-WMN.....	104
5.4.1 Objectives.....	104
5.4.2 Network Setup.....	104
5.4.3 Simulator Configuration.....	106
5.4.4 Results and Discussion.....	107
5.5 Summary of Results.....	112
5.5.1 Average Hop Count.....	112
5.5.2 Server Load Reduction.....	112
5.5.3 Hit Ratio.....	113
5.5.4 Network Load Reduction.....	113
5.5.5 Latency.....	114
5.5.6 Communication Overhead.....	114
5.6 Summary of Chapter Five.....	114
CHAPTER SIX.....	116
6 CONCLUSION AND FUTURE WORK.....	116
6.1 Future Work and Research Directions.....	117
REFERENCES.....	119
APPENDIX A.....	125
APPENDIX B.....	126

List of Figures

Figure 1.1: Example of Hybrid WMN Architecture.....	2
Figure 2.1: Infrastructure Wireless Mesh Network.....	10
Figure 2.2: Client Wireless Mesh Network.....	11
Figure 2.3: Hybrid Wireless Mesh Network.....	11
Figure 2.4: Multiple-Input Multiple Output Antenna System.....	14
Figure 2.5: IEEE 802.11s mesh components.....	21
Figure 2.6: Centralized File Distribution.....	22
Figure 2.7: Content Distribution Network.....	24
Figure 2.8: Multicast Communication.....	25
Figure 2.9: Peer to peer network.....	26
Figure 2.10: Data Oriented Network Architecture.....	31
Figure 2.11: Content Centric Networking (CCN).....	31
Figure 2.12: Publish Subscribe Internet Routing Paradigm (PSIRP).....	32
Figure 2.13: Network of Information(NetInf).....	33
Figure 3.1: Conceptual Framework.....	43
Figure 4.1: IcnMesh Architecture Overview.....	55
Figure 4.2: Dependency chart for IcnMesh.....	56
Figure 4.3: IcnMesh class collaboration diagram.....	58
Figure 4.4: MeshNode class diagram.....	60
Figure 4.5: Client node class diagram.....	61
Figure 4.6: Server Node class diagram.....	61
Figure 4.7: NS-3 Simulator modules.....	63
Figure 4.8: IcnMesh User Interface.....	65
Figure 4.9: Key of icons used in IcnMesh Application.....	66
Figure 4.10: IcnMesh mesh settings dialog.....	67
Figure 4.11: Pcap file structure.....	71
Figure 5.1: Simple host-centric WMN topology.....	77
Figure 5.2: Pcap on wireless interface of meshNode0.....	79
Figure 5.3: Pcap on Ethernet interface of producer0.....	79
Figure 5.4: Pcap on Ethernet interface of consumer0.....	80
Figure 5.5: Simple ICN network topology simulated on IcnMesh.....	81
Figure 5.6: Graph of interest packets logged by consumer0.....	83

Figure 5.7: Graph of data traffic received by consumer0.....	83
Figure 5.8: Graph of Interest packets logged by consumer1.....	84
Figure 5.9: Graph of data received by consumer1.....	84
Figure 5.10: Graph of Interest packets logged by producer0.....	85
Figure 5.11: Graph of data traffic sent by producer0.....	85
Figure 5.12: Graph of interest traffic logged by mesh nodes.....	86
Figure 5.13: Graph of data traffic logged by mesh nodes.....	86
Figure 5.14: IcnMesh network topology for hidden node routing.....	89
Figure 5.15: Graph of interest packets logged by consumer0.....	90
Figure 5.16: Graph of data transfer at consumer0.....	91
Figure 5.17: Graph of interest packets logged by meshNodes.....	91
Figure 5.18: Graph of data transfer at producer0.....	92
Figure 5.19: Graph of Interest packets logged by producer0.....	92
Figure 5.20: Graph of Interest packets logged by consumer0.....	95
Figure 5.21: Graph of data packets transferred to consumer0.....	95
Figure 5.22: Graph of interest packets logged by mesh nodes 0,1,3 and 4.....	96
Figure 5.23: Graph of interest packets logged by producer0.....	96
Figure 5.24: Graph of data transferred by producer0.....	97
Figure 5.25: Single producer, multiple consumers network setup.....	99
Figure 5.26: Graphs of interest packets logged by consumers 0-3.....	101
Figure 5.27: Graphs of data received by consumers 0-3.....	101
Figure 5.28: Graphs of Interest packets logged by consumers 4-7.....	102
Figure 5.29: Graphs of data received by consumers 4-7.....	102
Figure 5.30: Graph of interest packets received by producer0.....	103
Figure 5.31: Graph of Producer0 data transfer.....	103
Figure 5.32: Network topology of simulated medium sized WMN.....	105
Figure 5.33: Graph of interest packets logged by consumer nodes 0-3.....	108
Figure 5.34: Graph of data received at consumer nodes 0-3.....	108
Figure 5.35: Graph of interest packets logged by consumer nodes 4-7.....	109
Figure 5.36: Graph of data received by consumer nodes 4-7.....	109
Figure 5.37: Graph of interest packets at consumer nodes 8-11.....	110
Figure 5.38: Graph of data received by consumer nodes 8-11.....	110
Figure 5.39: Graph of interest packets logged by producer nodes.....	111
Figure 5.40: Graph of data transfer at producer nodes.....	111

List of Tables

Table 1.1: Summary of Related work.....	7
Table 2.1: Modulation and coding schemes in 802.11.....	15
Table 3.1: Differences between Quantitative and qualitative research methodologies.....	41
Table 4.1: IcnMesh Resource requirements.....	52
Table 4.2: IcnMesh Wireless Channel Properties.....	64
Table 5.1: Simulator configuration for simple host-centric network.....	74
Table 5.2: IcnMesh configuration for simple ICN network.....	78
Table 5.3: IcnMesh configuration with IEEE 802.11 MAC.....	85
Table 5.4: IcnMesh configuration with IEEE 802.11s MAC Layer.....	90
Table 5.5: IcnMesh configuration for single source multiple clients.....	96
Table 5.6: Content prefixes published by Producer nodes.....	102
Table 5.7: Consumer node Content prefix subscriptions.....	103
Table 5.8: IcnMesh configuration for multiple producers and consumers.....	103

List of Appendices

APPENDIX A Sample R script for graph generation.

APPENDIX B Sample Trace file

List of Abbreviations

AODV	Ad hoc On-demand Distance Vector Routing
API	Application Programming Interface
BATMAN	Better Approach to Mobile ad hoc Networking
CCN	Content-Centric Networking
CD-PAN	Content Distribution for Personal Area Networks
CDN	Content Distribution Network
DHT	Distributed Hash Table
DNS	Domain Name Server
DSSS	Dynamic Sequence Spread Spectrum
HTTP	Hyper Text Transfer Protocol
HWMP	Hybrid Wireless Mesh Protocol
IC-WMN	Information-Centric Wireless Mesh Network
ICN	Information Centric Networking
ICNRG	Information-Centric Networking Research Group
IEEE	Institute Electrical and Electronic Engineers
IETF	Internet Engineering Task Force
IP	Internet Protocol
ISM	Industrial-Scientific and Medical
MAC	Medium Access Control
MANET	Mobile Ad hoc Network
MAP	Mesh Access Point
MBSS	Mesh Basic Service Set
MIMO	Multiple Input Multiple Output
MU-MIMO	Multi-User MIMO
NAT	Network Address Translation
NDN	Named-Data Networking
NDO	Named-Data Object
NS-3	Network Simulator 3
OFDM	Orthogonal Frequency Division Multiplexing
OGM	Originator Messages
OLSR	Optimized Link-state Routing protocol
OSI	Open Systems Interconnection
P2P	Peer-to-peer
PHY	Physical layer

QoS	Quality of Service
RTT	Round Trip Time
RWIN	Receive Window.(TCP)
SNMP	Simple Network Management Protocol
TCP	Transmission Control Protocol
TQ	Transmission Quality
TTL	Time To Live
UDP	User Datagram Protocol
UI	User Interface
UPAC	Unified Peer-to-peer and Cache
URL	Universal Resource Locator
WISP	Wireless Internet Service Provider
WMN	Wireless Mesh Network

Publications

CHAPTER 1

1 INTRODUCTION

Community Networks are a new phenomenon where members of a community collaborate and satisfy common objectives. Recent technological revolutions have enabled communities to build their own networks especially in places with inadequate Internet access or underserved by Telecommunication networks. Wireless mesh Networks (WMNs) can provide an excellent framework for delivering broadband services to such areas (Berkoben, 2011). In the Web 2.0 era, where Internet Service Provider (ISPs) networks struggle to deliver traffic that is dominated by the sharing of bandwidth-intensive multimedia Content, WMNs have drawn greater appeal even to urban populations that seek alternatives to ISPs.

WMNs have their origins in the exploitation of IEEE 802.11 (Wi-Fi) technology to offer widespread access to the Internet. The size of the hotspots created by single Wi-Fi node was often too small to serve entire neighbourhoods. This, combined with the cost of installation of backbone wired networks led to the development of community WMNs that covered greater regions and were cheaper to install.

There are different architectures for WMNs depending on the intended application (Akyildiz & Wang, 2005). In Infrastructure wireless mesh networks, dedicated nodes participate in the mesh network and provide gateways for conventional client nodes through existing wireless networks or through wired networks such as Ethernet. In Client WMNs, the client nodes themselves form a wireless mesh network and participate in routing of traffic among themselves. Hybrid WMNs combine the approaches of Infrastructure and Client WMNs and may feature dedicated backhaul links within the mesh network.

The prevalent design for a WMN designed for providing Internet access is the hybrid architecture (Akyildiz et al., 2005). The hybrid-WMN is split into a two-tier architecture, in which an access tier connects ordinary consumer devices such as laptops, tablets and mobile phones (mesh clients) to stationary infrastructure nodes called mesh nodes, mesh routers or mesh access points. The nodes form a wireless mesh backhaul to route data packets between different mesh clients in the WMN and between mesh clients and the wider Internet through gateways with interface connected to the Internet (Akyildiz et al., 2005). A mesh node can also be equipped

with multiple radios thus allowing it to transmit and receive on multiple radio bands or channels simultaneously, for example, on 2.4 GHz and 5.8 GHz Industrial, Scientific Medical (ISM) bands or different channels within the same band.

Some of the WMN projects used for providing Internet access include Roofnet from MIT (Aguayo *et al.*, 2003), Kenya and Afghanistan's Fabfi (Berkoben, 2011), Rice University's TAP (Robinson and Knightly, 2007) and Berlin Freifunk (freie Netzwerke eV, F.). The main attractiveness of WMNs lies in the relatively low cost of installation as off-the-shelf hardware is used, and in the ability to scale the networks to cover large areas.

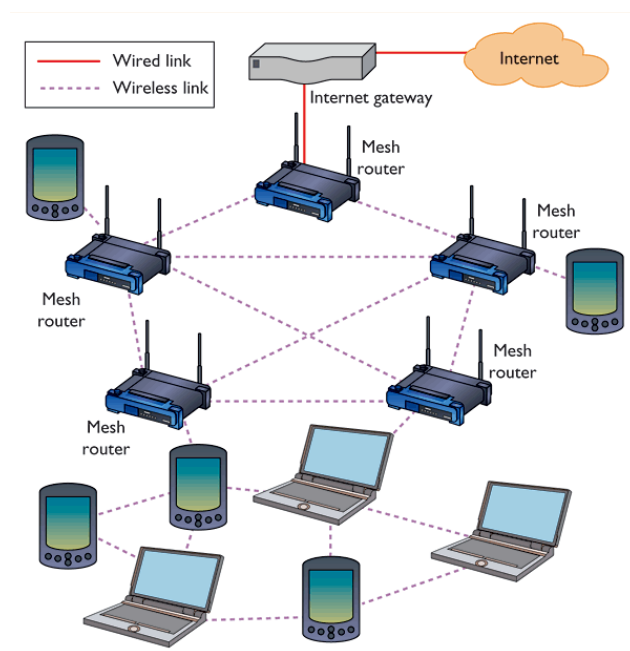


Figure 1.1: Example of Hybrid WMN Architecture

Apart from facilitating wireless access to the Internet, WMNs hold great potential in services that depend on sourcing Content from within the mesh network itself. Services such as P2P Content sharing, IPTV streaming, can benefit from the plenty in-network bandwidth that is available locally that is often underutilized.

Despite the many potentials of WMNs, they are yet to be adopted in the mainstream owing to two major limitations described as the bottleneck problem (Wu *et al.*, 2006, Aoun *et al.*, 2006) and the workload locality problem (Das *et al.*, 2007, Maggs, B.M. *et al.*, 1997). As most WMNs are used to provide Internet access, most of the Content accessed by network users resides in external networks. Since the dominant scheme for Content distribution in the present-day Internet is host-centric, traffic

from the WMN has to be routed to servers residing in the wider Internet. This is often through a single node in the WMN known as gateway node. Network clients have to share the fixed bandwidth at the gateway node which limits the number of clients that can simultaneously use it. In addition, a WMN becomes inefficient as the hop-count towards the destination increases (Passos *et al.*, 2006). Further to that, the whole network's access to the Internet is dependent on the gateway node thus creating a single point of failure and erodes all the advantages of deploying a WMN. Several schemes have been proposed to overcome this disadvantage. Solutions to mitigate the bottleneck problem are sometimes referred to as multi-homing schemes (da Silva *et al.*, 2010).

Another shortcoming of WMNs is when in a given population of clients, several users in the network request the same Content over time or the same Content is requested by several users at the same time. This is referred to as the Workload-locality problem. The Workload-locality problem arises in WMNs as many of them are used to provide Internet access. This has been published by Kroeger (1997), where the workload locality problem shows itself for a given client population accessing the Internet. Solutions to mitigate the Workload-locality problem are sometimes referred to as Content Distribution or Content Outsourcing Schemes.

It can be argued that the bottleneck and workload-locality problems arise from flaws in the current architecture of the Internet. In the traditional Internet, Content is stored in servers which client computers connect to in order to retrieve it. While this approach has worked for long, it has limitation in that Servers or connections to them could go down thus making content inaccessible to clients, overwhelmed when multiple clients simultaneously download content from them and low bandwidth and higher latencies when retrieving content stored in a server located several hops from the client.

In order to overcome some of these limitations of traditional Internet architecture, Content-Caching, Peer-to-peer networking and Network Coding have been proposed. These approaches have been used successfully in wired networks as the network's availability and stability are usually higher. As Wireless mesh networks have a different structure and less guarantees of availability, they require a different approach to Content distribution. It can also be argued that different kinds of Content require different distribution schemes owing to the differences in QoS guarantees and

bandwidth requirements. Thus an alternative approach to Content distribution in WMNs is to be investigated.

Recent developments in networking propose Information-Centric networking (ICNs) as a potential replacement for present-day Host-Centric network model. ICN is an approach to Internet content distribution that focuses on content rather than on addresses of the Content hosts. In ICNs, clients specify only *what* content they require and not *where* or *how* to retrieve the content as is the case with host-centric networks. The routing and forwarding subsystem in an ICN handles content retrieval. ICN leverages on content-caching and replication in addition to decoupling senders and receivers. The approach of ICN is similar to P2P protocol where focus is on the Content rather than on the location of the Content.

Currently, the prevailing approaches to ICN as outlined by Ahlgren et al.(2012) are:

1. Data-Oriented Network Architecture (DONA) (Koponen et al., 2007)
2. Content-Centric Networking (CCN) by (Jacobson, 2007)
3. Publish-Subscribe Internet Routing Paradigm (PSIRP) (Tarkoma, 2009)
4. Network of Information (NetInf) (Dannewitz, 2013)

The overall aim of ICN is to establish schemes for efficient access and distribution of Content. Compared to host-centric networks, ICN networks inherently cope better with disconnections, disruptions and flash-crowd effects. ICNs promise better Content distribution over traditional host-centric networks. If implemented in WMNs promise to offer superior performance compared to traditional WMNs.

In this Chapter, the concept of WMNs has been introduced, some of their limitations are discussed and ICN is introduced as a promising solution to WMN Content distribution limitations. This Chapter concludes by discussing the research objectives, related work, research contributions, an overview of the methodology used and the organization of this thesis.

1.1 Objectives

The main objective of this research was to develop an Content distribution scheme for wireless mesh networks that mitigates the bottleneck problem experienced in present-day WMNs.

This research proposes the use of ICN as a Content distribution scheme for WMNs. A suitable ICN scheme was to be identified and integrated in the designed WMN.

Another objective of this thesis was to understand the limitations of ICN schemes when used in WMNs. As ICNs have yet to gain widespread use, this research aims to predict potential shortcomings of ICNs and their integration in WMNs.

For this thesis, the NS-3 simulator was used to simulate a WMN and an ICN-scheme was built on top of the simulated WMN. To understand the integration and limitations of ICN in WMN, an analysis was conducted on the generated simulation data in order to get average throughput, network loading, average hop-count, convergence time and the communication overhead of the proposed scheme.

1.2 Research Aims

This research aims to develop an alternative architecture for WMNs that overlays an Information-Centric content distribution scheme on the WMN in order to mitigate the bottleneck problem that exists in present day WMNs. The proposed system integrates Information-Centric Networking into a Wireless Mesh Network architecture.

1.3 Problem Statement

Wireless Mesh Networks (WMN) have great potential in accommodating the increasing number of Internet-enabled devices. However, due to bottleneck problem, Content distribution in WMNs remains poor and that has led to a low adoption rate of WMN-based networks. This research will try to solve the Content distribution problem in WMNs through Information-Centric Networking.

1.4 Research Question

How can Information-Centric networking alleviate the Content-distribution problems that exists in present day WMNs?

1.4.1 Research Sub-questions

What is the most efficient ICN scheme for WMNs?

What WMN architecture supports ICN Content distribution best?

What are the challenges of the proposed ICN scheme in WMNs?

1.5 Research Methodology

In this section, the methodology used for the research conducted is described. The research requirement is to investigate ICN architectures and their applicability to WMNs. In order to conduct experiments on large network architectures and allow for reconfiguration as required, the simulation approach was preferred. Simulation offers a convenient way to examine large networks which would otherwise be complex, expensive and time consuming to study physically and can also be used to analyze system performance prior to physical design or to compare multiple alternatives over a wide range of conditions.. A custom Simulator, IcnMesh, was developed based on the NS3 simulator with ndnSIM ICN extensions. ndnSim is a flexible overlay network simulation framework based on NS3. It includes several structured and unstructured ICN protocols.

Data is collected through observation. The metrics of analysis in this research are network loading, hit ratio, average throughput, latency, communication overhead and server load. These are described in detail in Chapter 3. Data collection was performed through experimentation and observation.

1.6 Thesis Organization

The remainder of this thesis is organized as follows.

Chapter 2 explores the background and literature review. It gives an in-depth introduction to WMNs, Content-outsourcing schemes used in WMNs and Content Distribution schemes used in the wider Internet. It also introduces Information-Centric networks and discusses proposed ICN schemes.

In Chapter 3, the methodology used in this research is discussed in detail. The chapter describes the methods used in data collection, how data was collected and the metrics used for analysis. The design parameters for the Simulator are then presented.

Chapter 4 describes and discusses the design of our proposed Simulator for evaluating the Content distribution scheme and evaluates it against other comparable schemes.

In **Chapter 5**, After the system design and carrying out of the experimental processes, the results and/or findings are then analysed and presented in a clear and readable manner in this chapter.

Chapter 6 concludes this thesis by summarizing the contributions of our study and outlining directions for future work

1.7 Summary of Chapter 1

This chapter is gives a brief introduction to WMNs and also introduces the Content distribution limitation in WMNs. As the demand for Internet connectivity grows, WMNs have been seen as a potential to connect more users and devices to the Internet. However, because of the bottleneck and workload-locality limitations, WMNs have not yet been adopted in a large scale. The chapter then gives a brief introduction to Content distribution in traditional networks and also introduces Information-Centric networking. It also outlines the objectives of this research, gives the motivation for the research, states the problem to be solved, outlines related work and how they differ with this thesis.

CHAPTER TWO

2 LITERATURE REVIEW

2.1 Wireless Mesh Networks

2.1.1 Introduction

Wireless mesh networks (WMNs) consist of wireless devices organized in a mesh topology and that do not rely on any centralized support infrastructure. WMNs may comprise of mesh nodes and mesh clients. The key difference between WMNs and Mobile ad hoc Networks (MANETs) is that mesh nodes in MANETs have mobility and in most cases, MANETs do not feature mesh clients. WMNs provide network access for both mesh and conventional clients. Key to the development of WMNs was the development of the IEEE 802.11 standard popularly known as Wi-Fi. As the hotspot provided by a single 802.11 device was small, WMNs enable larger area coverage by connecting several Wi-Fi devices together in mesh configuration. Most modern devices have inbuilt Wi-Fi and would readily connect to WMNs for Internet access. The integration of WMNs with other networks such as the Ethernet, cellular, IEEE 802.11, IEEE 802.15, IEEE 802.16 can be accomplished through the gateway and bridging functions in the mesh routers.

Although other wireless standards such as IEEE 802.15.4 (bluetooth), IEEE 802.16 (ZigBee) can be configured to form WMNs, this thesis focuses on IEEE 802.11-based WMNs as they are the most common, most devices readily support the standard and 802.11 tends to have greater throughput than the rest. This section will view WMNs from an OSI model approach.

2.1.1.1 IEEE 802.11 Standard

The 802.11 standard, popularly known as Wireless LAN or Wi-Fi, was first published in 1997 (Crow *et al.*, 1997), defining a Medium Access Layer and Physical specification for Wireless Connectivity (Bernardos *et al.*, 2008). Owing to great success of the initial specification, the demand for higher bandwidth and the proliferation of Wi-Fi-enabled devices, the standard has since been extended to support new functionality and features such as security, higher data-rates and Quality of Service (QoS). All 802.11 standards operate in the 2.4 GHz and 5.8 GHz Industrial, Scientific and Medical (ISM) radio bands.

Although the role of IEEE 802.11 was initially limited to mobile client to Access Point (AP) communication, economies of scale made it a desirable alternative to interconnect these APs to create larger networks. IEEE 802.11 supports two additional modes of operation: the ad hoc mode for forming a single-hop ad hoc network where nodes communicate with each other directly without the use of an AP; and the wireless distribution system (WDS) mode for forming point-to-point AP relay links where each AP acts not only as a base station, but also as a wireless relay node. Through these modes, it is possible to form a WMN using standard 802.11 devices.

Below is a summary of the evolution of ratified IEEE 802.11 standards (Bernardos et al., 2008).

- IEEE 802.11a was originally described as clause 17 of the 1999 specification. 802.11a standard operates in the 5 GHz band with a maximum net data rate of 54 Mbit/s which yields a net achievable throughput in the mid-20 Mbit/s after error correction codes have been applied. It uses an OFDM-based physical layer. The OFDM waveform at 5.8 GHz is now defined in clause 18 of the 2012 specification, and provides protocols that allow transmission and reception of data at rates of 1.5 to 54 Mbit/s.
- IEEE 802.11b - The 802.11b standard has a maximum raw data rate of 11 Mbit/s, and uses the same media access method defined in the original standard since 802.11b is a direct extension of the modulation technique defined in the original standard. The dramatic increase in throughput of 802.11b (compared to the original standard) along with simultaneous substantial price reductions led to the rapid acceptance of 802.11b as the definitive wireless LAN technology. 802.11b products appeared on the market in the early 2000s.
- IEEE 802.11g - 802.11g was ratified in June 2003. It works in the 2.4 GHz band like 802.11b, but uses the same OFDM-based transmission scheme as 802.11a. It operates at a maximum physical layer bit rate of 54 Mbit/s exclusive of forward error correction codes, or about 22 Mbit/s average throughput after error correction is applied. 802.11g hardware is fully backward compatible with 802.11b hardware, and therefore is encumbered with legacy issues that reduce throughput by approximately 21% when compared to 802.11a.
- IEEE 802.11n – 802.11n was published by the IEEE in October, 2009. 802.11n is an amendment that improves upon the previous 802.11 standards by adding multiple-input multiple-output (MIMO) antennas. 802.11n operates on both the 2.4 GHz and

the 5 GHz bands. It operates at a maximum net data rate ranging from 54 Mbit/s to 600 Mbit/s depending on the radio and antenna configuration.

- IEEE 802.11ac - IEEE 802.11ac-2013 is an amendment to IEEE 802.11, published in December 2013, that builds on 802.11n. Changes compared to 802.11n include wider channels (80 or 160 MHz versus 40 MHz) in the 5 GHz band, more spatial streams (up to eight versus four), higher-order modulation (up to 256-QAM vs. 64-QAM), and the addition of Multi-User MIMO (MU-MIMO).

2.1.1.2 Wireless Mesh Networks Architectures

Akyildiz *et al.* (2005) classify WMNs into three categories namely Infrastructure or Backbone WMNs, Client WMNs and Hybrid WMNs. In Infrastructure WMNs, dedicated nodes participate in the mesh network and provide gateways for conventional client nodes through existing wireless networks or through wired networks such as Ethernet. In Client WMNs, the client nodes themselves form a WMN and participate in routing of traffic among themselves. Hybrid WMNs have been categorized as any network that features approached used in Infrastructure and Client WMNs. Hybrid WMNs tend to have a complex structure.

Infrastructure WMNs.

In this architecture, mesh nodes form an infrastructure for clients. The WMN infrastructure/backbone can be built using various types of radio technologies, in addition to the mostly used IEEE 802.11 technologies. The mesh nodes form a mesh of self-configuring, self-healing links among themselves. With gateway functionality, mesh nodes can be connected to the Internet. This approach, also referred to as infrastructure meshing, provides a backbone for conventional clients and enables

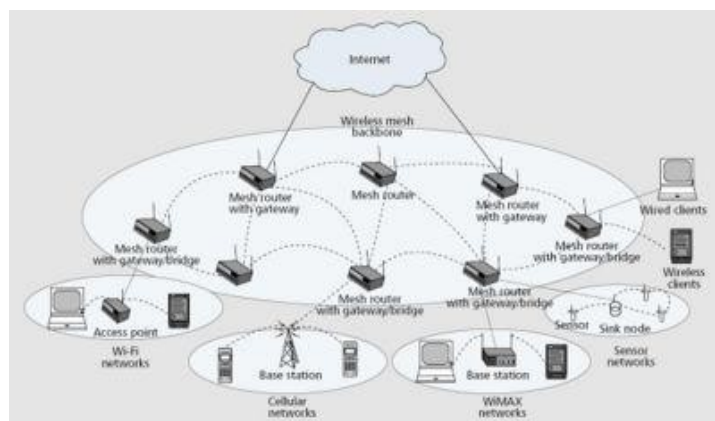


Figure 2.1: Infrastructure Wireless Mesh Network

integration of WMNs with existing wireless networks, through gateway/bridge functionalities in mesh routers. Conventional clients with an Ethernet interface can be connected to mesh nodes via Ethernet links. For conventional clients with the same

radio technologies as mesh routers, they can directly communicate with mesh nodes. If different radio technologies are used, clients must communicate with their base stations that have Ethernet connections to mesh routers.

Client WMNs.

In this type of architecture, client nodes constitute the actual network to perform routing and configuration functionalities as well as providing end-user applications to customers without going through a mesh router. Client WMNs are usually formed using one type of radios on devices and are similar a conventional ad hoc network. However, the requirements on end-user devices is increased when compared to infrastructure meshing, since in Client WMNs the end-users must perform additional functions such as routing and self-configuration.

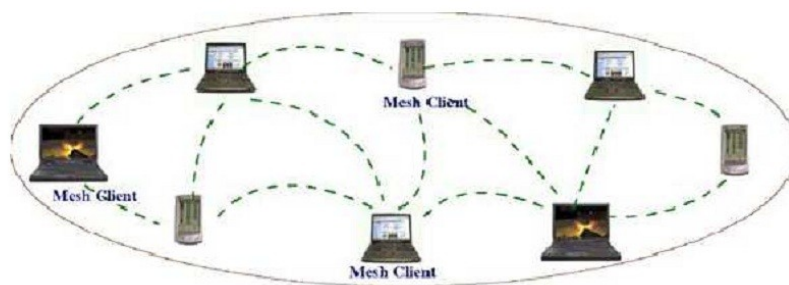


Figure 2.2: Client Wireless Mesh Network

Hybrid WMNs.

This architecture is the combination of infrastructure and client meshing. Mesh clients can access the network through mesh routers as well as directly meshing with other

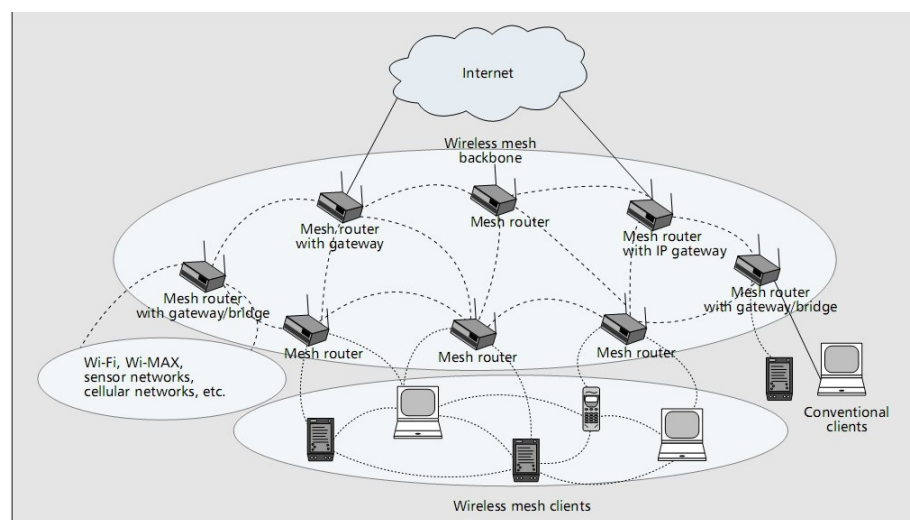


Figure 2.3: Hybrid Wireless Mesh Network
mesh clients. While the infrastructure provides connectivity to other networks such as

the Internet, Wi-Fi, WiMAX, cellular, and sensor networks, the routing capabilities of clients provide improved connectivity and coverage inside WMNs.

2.1.1.3 Applications of Wireless Mesh Networks

The most significant application of WMNs is to provide broadband Internet access to static or mobile hosts in areas where wired infrastructure is difficult or economically infeasible to deploy. WMNs can also be used in the following scenarios.

- Community and neighbourhood networking
- Enterprise networking
- Metropolitan Area Networks
- Security and Surveillance
- Emergency and Disaster response networks - If infrastructure is destroyed or overburdened, an WMN can quickly be established to facilitate rescue and emergency response. Similarly, police, parametric and fire fighters all responding to the same call should have information disseminated amongst them.

2.1.2 WMN Physical Layer

The physical (PHY) layer is the first layer in the OSI model (Stallings, 1987). It defines the specifications of the interface and the connector as well as the requirements of the medium. Key to the improvements in capacity of wireless networks are advancements in radio and antenna technologies. Physical layer techniques advance rapidly as communication theories, digital signal processing algorithms, RF technologies, and circuit design for wireless communications quickly evolve. These techniques mainly focus on three directions:

- Increase in data transmission rate.
- Improvements in error resilience capability in the wireless environment.
- Enabling software controllability of radio physical interface and reconfiguration while in operation.

2.1.2.1 Radio-Frequency Systems

The wireless medium that WMNs rely on is not as resilient as the wired counterpart. This is because radio propagation in the wireless medium is affected by weather conditions, interference and noise. Because of variations of channel quality, if the same coding and modulation scheme is used all the time, then the bit error rate (BER) in a channel varies significantly, which equivalently reduces the channel capacity and degrades the performance of upper layer protocols.

WMNs nodes are therefore required to perform adaptive channel coding and modulation. Link adaptation refers to the ability to adapt the modulation scheme and the coding rate of the error correction according to the quality of the radio link. If the conditions of the radio link are good, a high-level efficient modulation scheme and a small amount of error correction is used.

Modulation schemes such as OFDM and DSSS have been key to increases in capacity of WMN nodes. IEEE 802.11 devices implement techniques such as Adaptive modulation and link adaptation to respond to variations changing characteristics of the wireless medium. Software control of the physical interface allows configurations to be changed in response to changes in the wireless medium. Error control and correction is also performed at this stage in order to ensure packet integrity.

Table 2.1 shows how the selected modulation and coding scheme affects the overall data transmission rate. Configurations are to be changed in response to changes in the wireless medium. Error control and correction is also performed at this stage in order to ensure packet integrity. The table illustrates how the selected modulation and coding scheme affects the overall data transmission rate.

Table 2.1: Modulation and coding schemes in 802.11

Modulation Scheme	Bits per symbol	Coded bits per symbol	Coding rate	Transmission rate (Mbps)
BPSK	24	48	1/2	6
BPSK	36	48	3/4	9
QPSK	48	96	1/2	12
QPSK	72	96	3/4	18
16 QAM	96	192	1/2	24
16 QAM	144	192	3/4	36
64 QAM	192	288	2/3	48
64 QAM	216	288	3/4	54

2.1.2.2 Antenna Systems

Antennas radiate signals from wireless node to the environment by converting electric power into radio waves when transmitting, and radio waves to electric power when receiving. The two functions of an antenna are:

- For transmission of a signal, radio frequency electrical energy from the transmitter is converted into electromagnetic energy by the antenna and radiated into the surrounding environment.
- For reception of a signal, electromagnetic energy impinging on the antenna is converted into radio-frequency electrical energy and fed into the receiver.

Therefore, an Antenna performs a crucial role in a WMN node. There are two general classifications of antennas: Directional antennas and Omnidirectional antennas. Directional antennas radiate in a specific direction while omnidirectional antennas radiate over 360 degrees.

One of the biggest developments that improved the capacity of IEEE 802.11 networks was the use of antenna diversity through a scheme referred to as Multiple Input Multiple output (MIMO). MIMO allows a WMN node device to transmit and receive from multiple receivers simultaneously using multiple antennas.

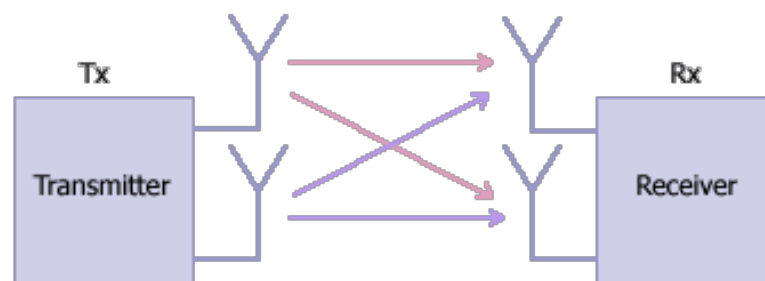


Figure 2.4: Multiple-Input Multiple Output Antenna System

By using multiple antenna techniques, the capacity and throughput of the WMNs could be enlarged, and the routing performance be improved. In addition, the WMNs benefit from the multiple antenna techniques in the aspects such as increased capacity and throughput, improved routing performance, increased energy efficiency, better quality of service (QoS), and improved location management (Zhang et al., 2006).

2.1.2.3 MultiChannel Systems

Multi-radio systems refer to nodes that can communicate on different frequencies simultaneously. A WMN node can be equipped with multiple wireless interfaces operating in different bands (e.g. 2.4 GHz and 5 GHz) or within the same band but on separate frequencies. Such systems are usually useful in hybrid WMNs.

2.1.3 WMN Medium Access Layer

The medium access layer (MAC) is the second layer of the OSI model (Stallings, 1987). The MAC layer interfaces the physical and higher layer protocols in order to interpret bit streams and convert them into packets or vice versa. The operation mechanisms and algorithms required to coordinate transmission and reception of packets among many nodes with the objective of improving network performance are also implemented in the MAC layer. In addition to these, additional error control checking is performed at this stage because errors can still occur because of variations in link quality from interference and other factors. In 802.11 networks, the MAC algorithm used is the Carrier Sense Multiple Access, Collision Avoidance (CSMA/CA).

The notion of proportional fairness is used in WMN where each node gets channel share proportional to their load. Some of the QoS related parameters that may be quantified are end-to-end delay, available bandwidth and probability of packet loss. In WMNs, the lack of centralized control, limited bandwidth, error-prone wireless channels, node mobility, and power or computational constraints makes it very difficult to provide effective QoS in such networks (Kumar *et al.*, 2006).

2.1.4 Network Layer

The Network Layer is the third layer of the OSI model (Stallings, 1987). In this section some routing protocols developed for WMNs are discussed, how network topology information can be discovered, and what routing metric are used and how these components are integrated with the routing path selection schemes in various routing protocols.

The most important role of the network layer is routing. Routing is the process by which given a source and a destination, a path can be found that achieves the best performance. A suitable routing protocol for WMNs has to consider the network topology and interference (Akyildiz & Wang, 2009).

The network topology in a WMN can be variable and inconsistent owing to effects such as interference and fading on the wireless medium. Link variations therefore cause an inconsistent view of network topology by different nodes in the same network. The network topology can also be changed due to node mobility or other nodes joining or leaving the network.

In WMNs, it may not be possible to determine a routing path solely based on the network topology. A WMN routing protocol has to consider delays in transmission between nodes, interference from nodes in the selected routing paths, traffic distribution and load balancing (Akyildiz & Wang, 2009).

In order to ensure good performance, routing metrics must satisfy four requirements (Yang et al., 2005).

- The routing metrics must not cause frequent route changes to ensure the stability of the network.
- The routing metrics must capture the characteristics of mesh networks to ensure that minimum weight paths have good performance.
- The routing metrics must ensure that minimum weight paths can be found by efficient algorithms with polynomial complexity.
- The routing metrics must ensure that forwarding loops are not formed by routing protocols.

The following are the routing metrics that a WMNs routing protocol has to rely on to determine the “distance” between any source node and a destination node (Akyildiz & Wang, 2009 ; Yang et al., 2005).

- Hop count
- Per-Hop Round Trip Time (RTT)
- Per-Hop Packet Pair Delay - Per-hop packet pair delay (PPD) is measured by sending two back-to-back probe packets from a node to its neighbour.
- Expected Transmission Count (ETX) - The ETX of a link is the expected number of transmissions before a packet is successfully delivered on a link.
- Expected Transmission on a Path (ETOP)
- Expected Transmission Time (ETT) and Weighted Cumulative ETT (WCETT)
- Effective Number of Transmissions (ENT)
- Bottleneck Link Capacity (BLC)
- Expected Data Rate (EDR)

WMNs share many common features with ad hoc networks. Thus, the routing protocols developed for ad hoc networks can usually be applied to WMNs. (Akyildiz & Wang, 2009). Several routing schemes have been developed for ad hoc networks, MANETs and WMNs. In this section, two popular routing schemes used in host-

centric WMNs are discussed; OLSR and BATMAN. In section 2.1.6, the IEEE 802.11s standard which enables mesh networking on MAC level is discussed.

2.1.4.1 Optimized Link-state Routing Protocol (OLSR)

Clausen, Thomas, and Philippe Jacquet (2003) defined Optimized Link-State Routing Protocol (OLSR). OLSR reduces the overhead of flooding link state information by requiring fewer nodes to forward the information. A broadcast from node X is only forwarded by its multi point relays. Multi point relays of node X are its neighbours such that each two-hop neighbour. of X is a one-hop neighbour. of at least one multi point relay of X. Each node transmits its neighbour. list in periodic beacons, so that all nodes can know their 2-hop neighbours, in order to choose the multi point relays (MPR). The OLSR routing process is described in the following sub-sections.

Packet Format and Forwarding

A universal specification of the packet format and an optimized flooding mechanism serves as the transport mechanism for all OLSR control traffic.

Link Sensing

Link sensing is the process through which a node detects changes to its neighbourhood (Clausen *et al.*, 2001). Link Sensing is accomplished through periodic emission of HELLO messages over the interfaces through which connectivity is checked.

Neighbour detection

The HELLO-message emitted in the link-sensing stage contains the emitting node's own address and the list of neighbours known to the node, including the status of the link to each neighbour. (e.g. symmetric or asymmetric). A node thereby informs its neighbours with which neighbours, and in what direction(s), communication has been confirmed. Upon receiving a HELLO-message, a node gathers information describing its neighbourhood and two-hop neighbourhood, as well as detect the quality of the links in its neighbourhood. The link from a node to its neighbour. is symmetric if in the HELLO-message from the neighbour-node contains the original node's own address, else the link is asymmetric (Clausen *et al.*, 2001). In a network with multiple interface nodes, additional information is required in order to map interface addresses to main addresses of the nodes. This additional information is acquired through multiple interface declaration (MID) messages.

MPR Selection and MPR Signalling

MPR selection is the process by which a node selects a subset of its neighbours such that a broadcast message retransmitted by these selected neighbours will be received by all nodes 2 hops away. The MPR set of a node is computed such that it, for each interface, satisfies this condition. The information required to perform this calculation is acquired through the periodic exchange of HELLO messages.

Topology Control Message Diffusion

Once mechanisms for neighbour sensing and for flooding messages to all nodes in the network in place, the OLSR algorithm diffuses a sufficient set of topological information to all nodes in the network. Topology Control messages (TC-message) are diffused with the purpose of providing each node in the network with sufficient link-state information to allow route calculation. A TC-message contains the address of the node generating the TC-message, as well as the addresses of all the MPR selectors of that node. This TC-message is diffused to all nodes in the network thus a node effectively announces reachability to all its MPR selectors.

Route Calculation

All nodes will receive a partial topology graph of the network, made up by all reachable nodes in the network and the set of links between a node and its MPR selectors. Using this partial topology graph, it is possible to apply a shortest path algorithm for computing optimal routes from a node to any reachable destination in the network. The topological information in each node is valid for a limited period of time, and must be refreshed periodically to remain valid. Expired information is purged from the topology graph. Given the link state information acquired through periodic message exchange, as well as the interface configuration of the nodes, the routing table for each node can be computed..

2.1.4.2 Better Approach To Mobile ad hoc Networking (BATMAN)

BATMAN is a simple and robust algorithm for establishing multi-hop routes in ad hoc networks. As explained by Johnson *et al.*(2008), a node in a WMN that employs BATMAN routing does not maintain the full route to the destination but only the information about the next link through which the node can find the best route.

The objective in BATMAN is to maximize the probability of delivering a message. A mesh node does not check the quality of each the link but only just checks for the existence of a link to neighbouring nodes. These checks are done by broadcasting

originator messages (OGM) periodically to neighbouring nodes. When a mesh node receives an OGM, it adds a new entry for the sender node in its routing table if it did not exist. The sender node is added as a one hop neighbour to it and its count incremented. If the originator is in the routing table and the sender is a new, the sender is added as a one hop neighbour to the originator and count incremented. If the originator is in the routing table and the sender is not new the senders count is incremented. The count is the amount of received OGMs of an originator through a specific one hop neighbour.

The structure of the OGM packet periodically sent is as follows:

- Originator address
- Sending node address: this is changed by receiving nodes and then the packet is re-broadcasted
- Unique sequence number: The sequence number is used to check the concurrency of the message
- Bidirectional link flag: used when the OGM packet received is its own and the sender is someone else
- Time to live (TTL)

The links are compared in terms of the number of originator messages that have been received within the current sliding window this value is called the transmission quality (TQ) and is the routing metric used by BATMAN. The sliding window is a fixed value that defines a range of the unique sequence numbers afforded to each OGM packet sent by a node. BATMAN is therefore in essence a proactive routing protocol as it pre builds its routing table. The BATMAN routing algorithm has two implementations:

- BATMAN daemon (BATMANd) which is implemented as a layer 3 protocol and runs as a daemon in Unix operating systems.
- BATMAN advanced (BATMAN-adv) which is implemented as a layer 2 protocol. BATMAN-adv only uses the MAC address for addressing its neighbours and emulates an Ethernet bridge such that nodes in the WMN appear to be connected directly. In effect, protocol layers above the MAC layer are not aware of the multi-hop links between mesh nodes since all nodes appear to be connected directly.

BATMAN's routing technique causes low processing and traffic cost. This makes it ideal for use on devices that have limited processing power such as the IEEE 802.11 embedded devices commonly used in WMNs.

2.1.5 Wireless Mesh Network Capacity

Jun and Sichitiu (2003) show that for WMNs the throughput of each mesh node decreases by the inverse of the total number of mesh nodes (n) in the network ($O(1/n)$). In their work, they defined the bottleneck collision domain as the geographical area of the network that bounds from above the amount of data that can be transmitted in the network. This sets an upper limit on the throughput of a mesh node in the WMN and shows that the larger the network, the more inefficient a node becomes.

Jun and Sichitiu (2003) proposed that the limitation can be addressed by adding multiple gateway nodes in the network in order to reduce the distance between the source and destination nodes. However, da Silva et al. (2010) note that in NAT-enabled WMNs, multi-homing is a challenge owing to NAT address domains assigned by the gateway nodes. da Silva et al. proposed DynTun to address the challenge of multi-homing in WMNs.

For large WMN implementations such as Berlin Freifunk, a multi-channel design is used to enable scalability while retaining good network throughput. Jiang et al. (2007) note that routing in large-scale multi-radio WMNs faces two limitations in achieving a high throughput. The long path between the source and the destination, and the high routing overhead. They propose a new routing metric for selecting multi-channel routes with maximum end-to-end capacity and a feedback based algorithm to maximize the control message broadcasting interval is proposed to minimize the routing overhead.

2.1.6 Standards on WMNs

Despite the potential offered by WMNs, there are hardly few open standards supporting it. Most implementations of WMNs still rely on IEEE 802.11 devices which although capable of forming a mesh configuration, IEEE 802.11 was not intended for mesh applications. It has been noted that the CSMA/CA MAC used by IEEE 802.11 is not ideal for WMNs. Akyildiz & Wang (2009) mention 3 general schemes for fine-tuning the CSMA/CA MAC to improve performance for WMNs. These are:

- Adjust physical carrier sense to eliminate the hidden node problem.
- Improve virtual carrier sense to reduce the number of exposed nodes.

- Dynamic tuning of back-off procedure to prevent collisions.

In response to the need for an open mesh networking standard, the IEEE 802.11s standard was developed and is described in section 2.1.6.1 below.

2.1.6.1 IEEE 802.11s Standard

IEEE 802.11s is an amendment to add mesh networking capabilities to the 802.11 standard. IEEE 802.11s allows multiple wireless nodes to connect to each other without requiring an access point between them. 802.11s implements a MAC protocol such that the formed WMN supports both broadcast/multicast and unicast frame delivery at the MAC layer using radio-aware metrics. 802.11s defines the concept of a mesh station which is simply a station that supports the mesh facility and is capable of participating in a mesh cloud, or Mesh Basic Service Set (MBSS). The mesh facility is simply the set of features, functions, and frame formats that enable mesh operation.

Mesh Basic Service Set (MBSS) is the official name of the mesh cloud. An MBSS is an 802.11 LAN consisting of autonomous stations. These stations establish peer-to-peer wireless links and transfer messages mutually. For stations participating in an MBSS to be able to detect each other, they must be on the same channel. Access to the channel is a key component of the mesh infrastructure, and a new coordination function is created for mesh networks; the mesh coordination function is a coordination function that combines aspects of the contention-based and scheduled access methods. The IEEE 802.11s standard defines a routing scheme based on AODV called Hybrid Wireless Mesh Protocol (HWMP). Figure 2.5 illustrates the IEEE 802.11s mesh architecture.

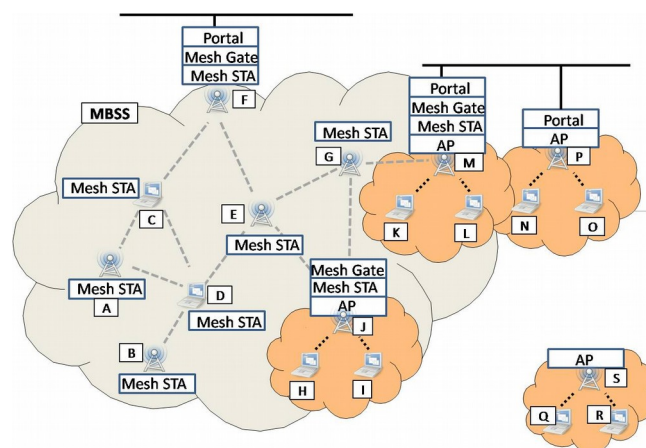


Figure 2.5: IEEE 802.11s mesh components

2.2 Content Distribution Schemes

Different kinds of content require different distribution schemes owing to the differences in QoS guarantees, bandwidth requirements. It is for these reasons that there are different approaches to content distribution, thus alternatives have been sought to the Server-Client model of the traditional Internet. Some of them are outlined below.

2.2.1 Centralized File Distribution

This is the approach taken by traditional Internet networks where a single copy of a file is stored on a single server. Clients have to connect to the server in order to download the file. This scheme is also known as end-to-end file distribution.

There are three characteristics of the host-centric architecture that are barriers to satisfy the requirements of Content distribution:

- There is no guarantee for quality of service
- There is no guarantee for end-to-end security
- There are no scalable forwarding mechanisms.

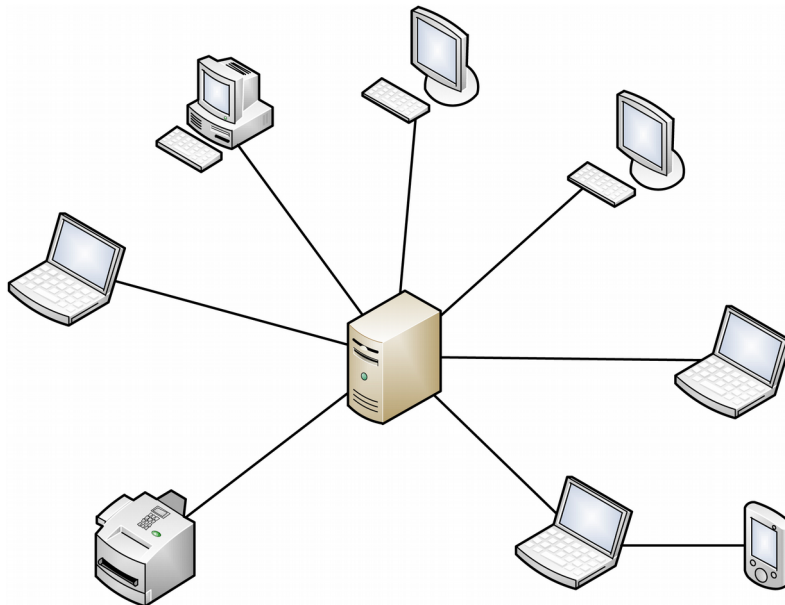


Figure 2.6: Centralized File Distribution

2.2.2 Content Distribution Networks

Content Distribution Networks (CDNs) are a mechanism for distributing content on behalf of the origin Web Sites. CDNs were proposed to increase the efficiency and scalability of the client-server model (Passarella, 2012). CDNs work by replicating content on different servers thus increasing its availability and communication efficiency. CDNs enable clients to experience a high delivery rate and low latency because of the decrease in congestion.

CDNs are composed of a set of distributed servers which are interconnected through the Internet and work cooperatively to distribute Content. A variety of schemes are used by CDNs to balance the load on Content servers. The most commonly used schemes are:

- URL rewriting – This uses the redirection mechanism originally offered by the Hypertext transfer protocol (HTTP). All requests to HTTP objects are performed by Web browsers running on client hosts. When the origin server receives a request, it sends back to the requester an HTTP redirect message with the address of the best replica server.
- DNS redirection – In this scheme, Content requests are redirected to the nearest server by exploiting the DNS scheme. A DNS server within the CDN receives messages from client hosts requesting the address associated with the name of the origin server and then sends back to the client the address of the best replica server.

HTTP and DNS redirection can select a replica server based on the number of hops or round-trip time (RTT) between clients and replica servers and may also consider the server load during selection. The redirection schemes cannot, however, guarantee Content persistence because if the owner, domain or any other property of a given Content changes, users may not be able to retrieve the Content by using the original URL.

Thus for every change in Content storage location, users have to query centralized structures in order to obtain the new place of the content, which may increase the Content delivery time (Koponen et al., 2007).

There are two main building blocks at the core of a CDN (Passarella, 2012):

- i. The distribution and replication service
- ii. The request redirection service.

The distribution and replication service is used to find proper servers, allocate storage capacity and to allocate Content to the selected servers. The request redirection service is the CDN interface with content consumers. Basically, this service receives content requests and then forwards each request to the more suitable CDN server to satisfy it.

There are two types of servers in a CDN: the Origin server and Replica/Edge server. The origin server attributes the content identifier, stores and announces the content. Replica servers then forward the content to clients. Clients send requests to the origin server who redirects these messages to the replica server closer to the client and that stores the desired content.

CDN performance is affected by server placement algorithms, capacity planning of servers and cache replacement policies. The more the number of replica servers there are in a CDN , the higher is the probability of finding a server close to the client. Redirection mechanisms in CDNs must select the best replica server in real time in order to have less impact on the delivery time, but it implies high computational costs.

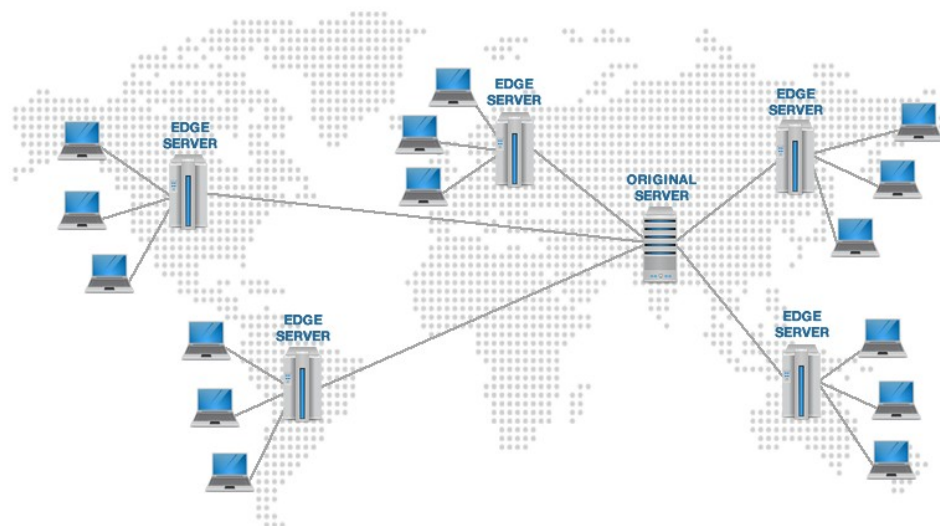


Figure 2.7: Content Distribution Network

CDNs cannot be considered a general solution to satisfy the different content distribution applications on the Internet. This is because most CDNs are proprietary and specific for a given application and are therefore not interoperable (Passarella,

2012). Examples of commercial CDN services include Akamai networks, Amazon Web services, Facebook and YouTube.

2.2.3 Multicast Communication

Multicast communication was one of the first proposals to increase the content distribution efficiency on the Internet (Lin & Ni, 1993). Multicast is implemented by an IP multicast in the network layer (Deering, 1988).

In multicast, Content sent by a host can reach multiple hosts simultaneously. Receiving hosts are aggregated in a group identified by only one IP address. When a datagram is sent to the multicast IP address, all nodes in the multicast group receive a copy. The network layer in a multicast network forwards and replicates this datagram over the entire distribution tree that covers all the hosts interested in the group content. The main advantage of multicast is bandwidth saving by not forwarding unnecessary copies of the same datagrams over the same link.

Although IP multicast was proposed in the 1990s, it is not widely adopted. The main reason is the complexity to configure and manage the set of protocols needed by IP multicast. This complexity comes from the service model proposed by the IP multicast itself. A host is able to join and leave a multicast group at any time and can be a member of more than one group simultaneously. It is also important to note that a host does not need to be a member of a group to send datagrams to the group (Costa *et al.*, 2006).

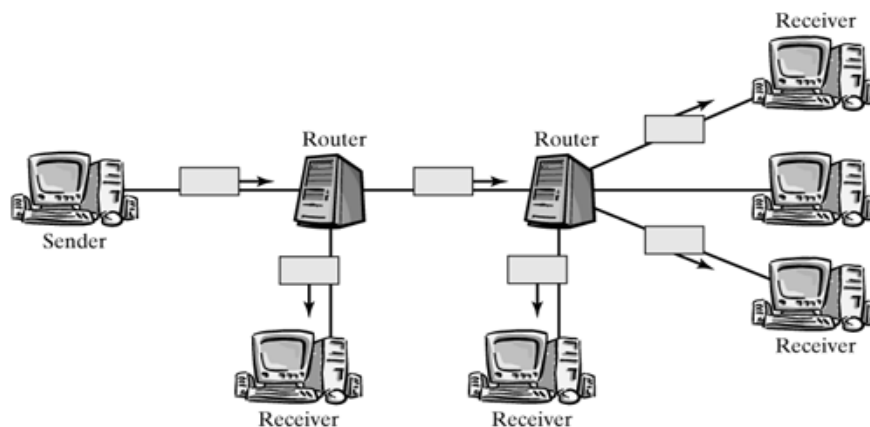


Figure 2.8: Multicast Communication

2.2.4 Peer-to-peer Schemes

Peer-to-peer (P2P) systems aim at increasing content distribution efficiency by promoting content sharing among the users of the system. Basically, nodes interested in the same content, referred to as peers, create an overlay network at the application layer and altruistically share bandwidth, the process and storage capacity and are therefore able to exchange Content.

The key idea of P2P is that each peer contributes to a given amount of its resources and uses the service offered by the system (Passarella, 2012). Consequently, the more peers there are in the system, the more is the capacity of the system to satisfy the user requirements by improving Content delivery time and availability. P2P systems therefore intrinsically provide the the scalability needed by content distribution applications.

P2P systems do not require changes in the network core as IP multicast does. The key aspect is that users in a P2P network are interested in receiving given content (a file or a multimedia streaming) regardless of who sends it. For instance, in Bittorrent, a new peer in the system randomly chooses its partners, that is the nodes allowed to exchange content chunks with it. These partners are selected at random from a subset of peers who are interested in the same content and no information about location or identification of peers is taken into account during the selection process.

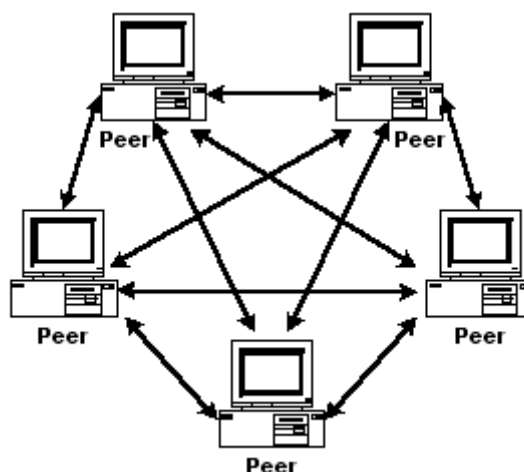


Figure 2.9: Peer to peer network

The huge success of both P2P file-sharing and P2P streaming systems with millions of users indicates that the paradigm of the Internet application is changing. It can therefore be argued that the success of P2P systems is the basis for the

development of Information-Centric Networks discussed in section 2.3. The success of P2P networks shows that users are more interested in the Content regardless of its sender.

2.2.5 Network Coding Schemes

Network coding is a scheme where routers not only forward but also store packets. In networks implementing network coding schemes, routers not only forward packets but also mix packets from different sources in order to increase the information content of each transmission.

Network coding has been shown to reduce file download times by up to 30% (Gkantsidis *et al.*, 2005). Network Coding has also been implemented in WMNs where it has been shown to increase the network throughput (Katti Sachin *et al.*, 2006). An architecture for network coding in WMNs is proposed by Katti Sachin *et al.*, 2008.

2.3 Information-Centric Networking

Information-Centric networking (ICN) refers to network architectures that are based on named data objects (NDOs) (Ahlgren *et al.*, 2013 , de Brito *et al.*, 2013). An NDO is any type of object that can be stored and accessed on a Computer, for example web-pages, images, videos, music files etc.

ICN networks leverage in-network caching, multi-party communication through replication, and interaction models decoupling senders and receivers. The main goal of ICN research is to provide a network infrastructure service that is better suited to today's use and that is more resilient to disruptions and failures. (Ahlgren *et al.*, 2013).

2.3.1 Content Naming

The NDO in an ICN is unique and independent of location, storage method, application program, and transportation method. This means that an NDO keeps its name, and thus its identity, regardless of its location and how it is copied, stored, and communicated. It also means that any two copies of an NDO are equivalent. According to de Brito *et al.*(2013), an ideal naming scheme for an ICN should provide names with the following set of features:

- Uniqueness: This ensures assure unique Content identification with the smallest possible false positives and negatives.

- Persistence: This guarantees the validity of the content name in synchronization with the content validity itself.
- Scalability: This is in order to allow different namespace scales, serving tiny and huge namespaces the same with no limitations regarding content nature, storage location or any other characteristic.

de Brito et al., (2013) further give three basic content naming techniques can be used in ICNs: flat naming, hierarchical naming and attribute-based naming.

- Flat naming – This is when there are no explicit rules to bind information into the content identifier format or meaning to the NDO. NDO names can be derived from cryptographic hashes of the file and appear to be random (de Brito et al., 2013)
- Hierarchical naming – This is when different string name components or any unique identifiers can be formed and assigned to content. Hierarchical names have intrinsic semantics since their structures and components reflect information related to the nature of the content itself, for example, property, version, date of creation and format.
- Attribute-based names – This is when Attribute-value pairs (AVP) of the form [attribute= value] are assigned to Content and make it possible to identify them.

2.3.2 Content Routing

ICNs should be able to deliver content requested by name, without any information regarding its storage location. There are two key functions that name resolution and routing must achieve when there is a request for a specific NDO. The first is to find a node that holds a copy of the NDO and deliver the request to that node. The second is to find a path from that node back to the requester over which the NDO can be delivered (Ahlgren et al., 2013). This Content or name-based routing should have the following characteristics (de Brito et al., 2013) :

- Content oriented: packets should be addressed to content names, without information about source or destination.
- Robustness: routing should be fault tolerant and quickly recover from discontinuities, avoiding sending data to faulty nodes.
- Efficiency: control information should have low impact on network traffic.

- Scalability: routing should be flexible so as to be deployed in a variety of scenarios, serving dense network topologies and large namespaces and tiny local networks as well.

Content routing can be hierarchical or non-hierarchical. In hierarchical routing, or structured routing, network routers are connected in a hierarchical structure thus ensuring deterministic flows of routing information and data. There are two hierarchical routing concepts in ICNs : tree-based and distributed hash tables (DHT) architectures. Hierarchical tree-based network topologies require knowledge of the location of the desired destination node. DHTs are structures adopted for cryptographic hash keys distribution among participating nodes. Processing and caching costs involved in key mapping are shared among nodes, ensuring protection against single points of failure

In Non-hierarchical routing, or unstructured routing, there are no dedicated structures for storing routing information. The network does not organize routers in hierarchical structures and allows multiple paths to be used for the same content. This increases the availability of the network

2.3.3 Advantages of Information-Centric Networks

Ahlgren et al., 2013 outline the following advantages of the ICN approach over existing host-centric networks.

- Scalable and Cost-Efficient Content Distribution. According to recent predictions, global IP traffic is set to increase tremendously owing to proliferation of user generated Content and more users getting connected to the Internet. Although P2P networking and CDNs alleviate some of the disadvantages of host-centric networks, they represent a shift to Content-Based networking as users and Content providers are more interested in accessing named content, regardless of its location. ICNs promise a more architecturally sound way of addressing these requirements in a manner that does not require individual amendments for specific domains and architectures while alleviating the shortcomings of CDNs and P2P networks.
- Persistent and Unique Naming for Internet Content. Currently, Content on the Internet is accessed by its URI. The URIs are object locators which, after DNS resolution, exhibit the IP address of a web server that is serving requests by resolving the local part of URI. This can easily break name-object bindings when an object is moved, the website changes domain, or if the website is unreachable. If replicas of the same object are placed at different web servers, they will be

accessible using different URIs, and essentially appear as different objects to the system (including caches). The ICN approach overcomes these problems with persistent and unique naming of NDOs, and with its service model that decouples producers from consumers.

- Inbuilt Security and data validation Model – Because Content is uniquely named in an ICN network, a name-data integrity and origin verification of NDOs system that is independent of the immediate source is built into the network. This also enables ubiquitous caching with retained name-data integrity and authenticity. In the host-centric model, Transport Layer Security is required to protect the communication channel between the client and the server. This is not required in an ICN network as data can be easily verified.
- Ability to provide mobility and multi-homing. The host-centric model of current networks require end-to-end connectivity between clients as servers. This limits the mobility of a client and in the case of routers it limits multi-homing. choosing which path or interface to use for these connections. The ICN approach does not have end-to-end connections that require this kind of connection management and therefore provides
- Inbuilt tolerance to disruption - Because Content in an ICN can be stored in multiple locations, the network has an inbuilt tolerance to disruption. This is unlike the the host-centric model which is very susceptible to various types of disruptions such as link breakages and deliberate denial of service (DoS) attacks.

2.3.4 Main ICN Architectures

2.3.4.1 Data-Oriented Network Architecture (DONA)

In a data-oriented network, Content producers publish the NDOs into the network (Koponen et al., 2008). NDOs are then registered by Data-serving nodes on a Resolution Handler (RH) which helps in routing requests for the Content to the specific host it is stored in. To request for Content, Consumers issue a FIND request which contains the NDO. Once the data is found in the network, it is sent back through the reverse RH path that the FIND request took to reach the storage location. Figure 2.10 illustrates the DONA network architecture.

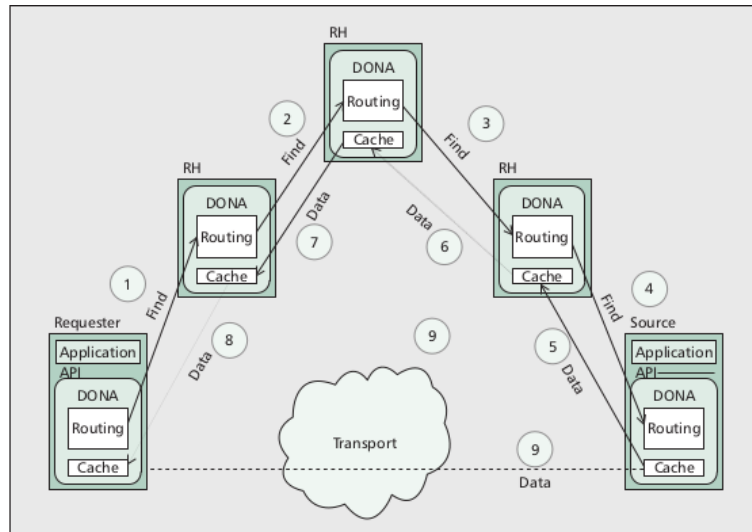


Figure 2.10: Data Oriented Network Architecture

2.3.4.2 Content-Centric Networking (CCN)

CCN communication is driven by the consumers of data. There are two CCN packet types, Interest and Data. A consumer asks for content by broadcasting its interest over all available connectivity. Any node hearing the interest and having data that satisfies it can respond with a Data packet. Data is transmitted only in response to an Interest and consumes that Interest.

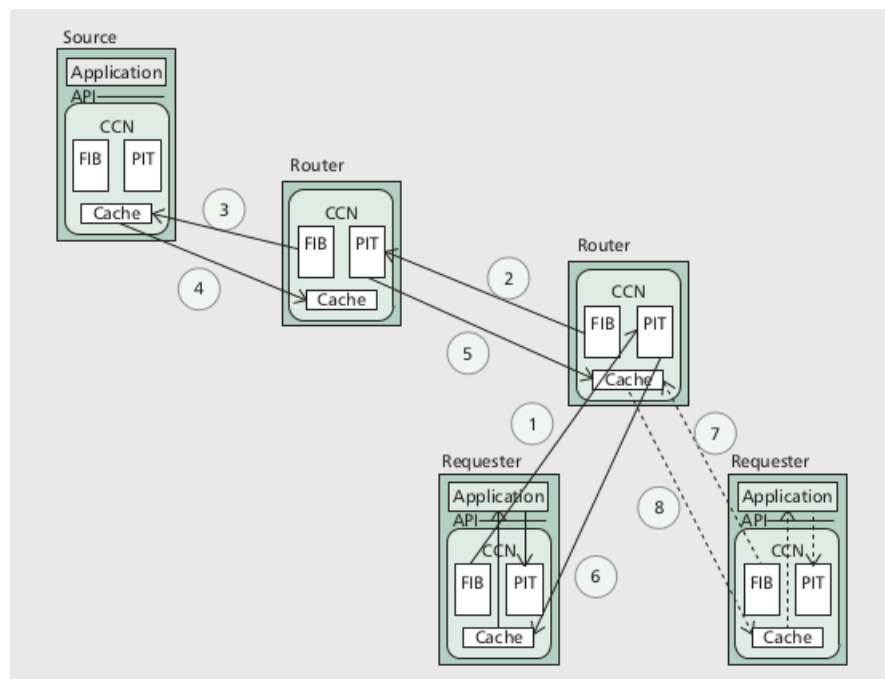


Figure 2.11: Content Centric Networking (CCN)

Since both Interest and Data identify the content being exchanged by name, multiple nodes interested in the same content can share transmissions over a broadcast

medium using standard multicast suppression techniques . Figure 2.11 illustrates the CCN network architecture.

2.3.4.3 Publish-Subscribe Internet Routing Paradigm (PSIRP)

In PSIRP, NDOs are also published into the network by the Content producers. Receivers can subscribe to NDOs. The publications and subscriptions are matched by a rendezvous system. The subscription request specifies the scope identifier and the rendezvous identifier which together name the desired NDO. The identifiers are input to a matching procedure resulting in a forwarding identifier, which is sent to the NDO source so that it can start forwarding data. The FI consists of a bloom filter that routers use for selecting the interfaces on which to forward an NDO. This means that routers do not need to keep forwarding state. The use of Bloom filters results in a certain number of false positives; in this case this means forwarding on some interfaces where there are no receivers. Figure 2.12 illustrates the PSIRP network architecture.

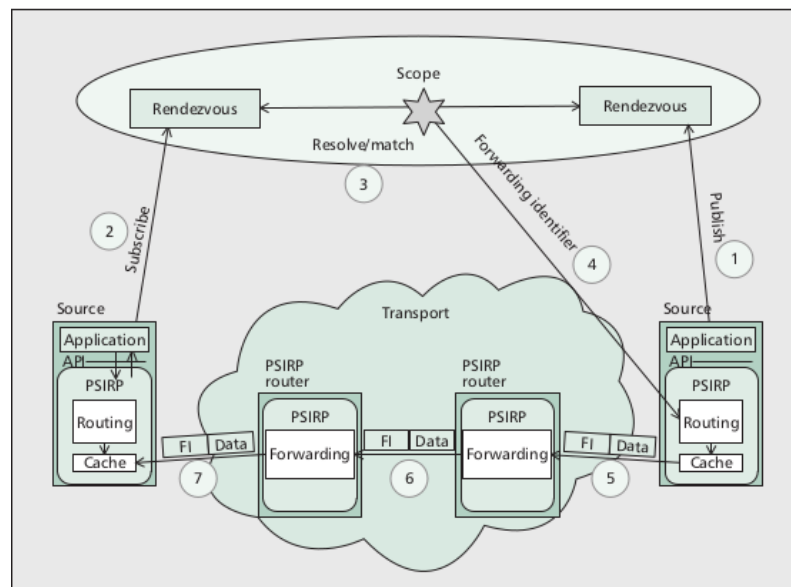


Figure 2.12: Publish Subscribe Internet Routing Paradigm (PSIRP)

2.3.4.4 Network of Information (NetInf)

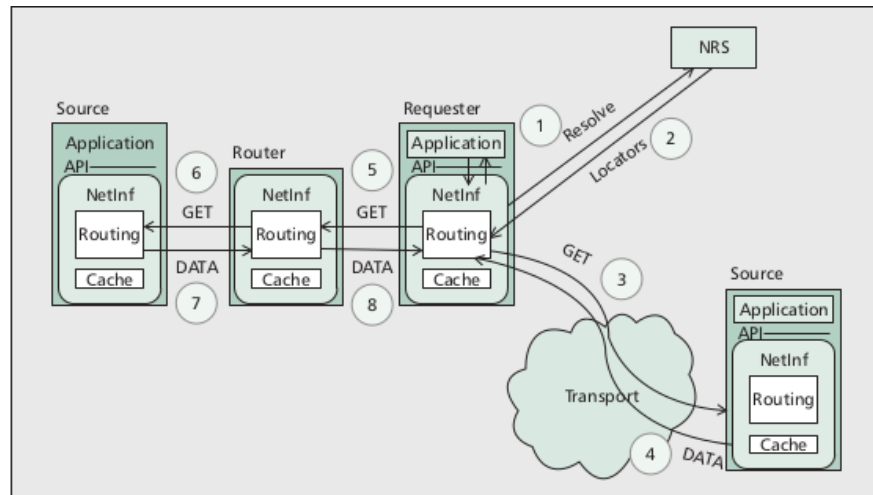


Figure 2.13: Network of Information(NetInf)

NetInf defines two models of retrieving NDOs: via name resolution, and by name-based routing. Content sources publish an NDO by registering a name/locator binding with a Name-Resolution Service(NRS) (Ahlgren B. *et al.*, 2010). Clients issue GET command to retrieve content of interest from the network. The name-resolution service and routing scheme employed will then retrieve content from any of the locations it is stored. Figure 2.13 illustrates NetInf architecture.

2.4 Content Distribution Schemes in WMNs

There have been attempts to solve both the bottleneck problem that causes poor Content distribution in WMNs. The most popular approaches use Overlay networks, Caching or replication.

2.4.1 Overlay Networks/Peer-to-Peer schemes

Overlay networks refer to network architectures where the underlying layers (i.e., the MAC, PHY and Network layers) are transparent to the client device. An overlay network is built on top of the physical network. Nodes in an overlay network are connected via virtual/logical links.

da Silva *et al.*(2010) proposed DynTun which enables multi-homing. in WMNs thus allowing multiple gateways to operate in the network. Their solution creates dynamic tunnels that are based on logical packet marking and routing policies. This allows load balancing between multiple gateways and also reduces unfairness because clients can access the Internet by using the closest gateway. However, during

gateway changes, applications running on clients may suffer with packet losses and disruptions. (da Silva *et al.*, 2010)

Most of the peer-to-peer schemes for Content distribution rely on derivatives of the Bit-torrent protocol which is a peer to peer protocol for distributing large amounts of data across networks.(Chokkalingam & Riyaz, 2004).

Bit-hoc (Sbai *et. al*) is an adaptation of the BitTorrent protocol for ad hoc WMNs. Cooperative Content Distribution protocol (CCD) (Michiardi, 2009) is another such scheme utilizing BitTorrent for mesh nodes downloading the same content in parallel. In both schemes, user end-devices participate in the mesh network and also store portions of the data to be shared. In reality, users are apprehensive about installation of foreign software in their computers and sharing content from their computers with others over the network, thus such schemes are not only difficult to implement, but are also impractical. Peer-to-peer schemes have however been shown to perform poorly when the number of peers is small. There are also issues of trust when sharing files in a peer-to-peer network (Aber , 2001).

Peer-to-peer schemes combine the approaches of centralized distribution with peer-to-peer distribution. It offsets the disadvantage of few seeds in peer-to-peer networks, and also relieves load from the centralized server. Peer-to-peer schemes are also used as overlay networks in replication and caching schemes described in the following sections.

2.4.2 Replication Schemes

Replication is a technique that allows improvements the quality of distributed services by creating copies of a site's Content, and placing these copies at well-chosen locations. Replication has been increasingly applied to Web services and is the basis for CDNs discussed in section 2.2.2 above.

Replication algorithms designed for the Internet are centralized and incur a high computation cost. The replica placement problem can be expressed in two forms (Sivasubramanian *et al.*, 2004)

- Replica server placement: This is the problem of finding suitable locations for replica servers.
- Replica content placement: This is the problem of selecting replica servers that should host replicas of an object.

P2PMesh was proposed by Al Asaad *et al.*, 2009. The scheme aims to reduce the number of failed lookups and the file lookup latency. In P2PMesh, a peer with a file to share sends a request to upload the file to the mesh node it is connected to. The request is acknowledge and the file is uploaded to the mesh node which registers the file's descriptor, the key of the host and the mesh node's IP address.

When a file is requested by a peer, the file's metadata is hashed to obtain the key which is then sent to the connected mesh node. The P2PMesh protocol then tries to locate the file within the WMN. The file is served from the mesh node if it was cached before. If, however, the file is not found in the connected mesh node, the key is sent to the mesh router responsible for the key using a distributed hash table DHT routing protocol. If the file is not present in the WMN, the key is sent to the wider internet through the gateway node to be obtained from the Internet.

If the P2PMesh system locates the requested file in more than one node, a list of the nodes is sent to the requester which then selects this which file providers to retrieve different chunks of Content from.

File providers are selected based on minimizing the following routing metrics:

- Route coupling that results in interference between neighbouring routes destined to the same receiver.
- Hop distance between requesting node and file provider.
- Number of disjoint nodes on a route from the requester to the provider.

Jin and Wang (2005) evaluated different replication strategies for Content and service replication in multi-hop WMNs. Their objective was to minimize object access cost in large decentralized and unstructured 2-D mesh networks and determining the optimal numbers of replicas for a set of objects with distinct access probabilities. They show that the optimal minimum-access strategy departs significantly from the commonly used proportional replication strategy, and has huge performance gains. This strategy can be implemented using a local replacement algorithm modified from the Greedy-dual approach.

MeshChord was proposed by Canali *et al.*(2010) and is an extension of the Chord protocol (Stoica *et al.*2003). In the Chord protocol, most messages are exchanged between a peer and its neighbouring peers. MeshChord utilizes locality by assigning

peers close in the physical network with close-by IDs in the Chord ring. Peers in the same sub-region of the deployment area are mapped to the same segment of the Chord ring. The Chord ring is formed by converting physical proximity to proximity in the Chord ring. A peer in MeshChord must therefore be aware of its physical position and neighbour positions. This can be accomplished by having every peer's GPS coordinates.

MeshChord also utilizes a MAC cross-layering technique which aims to speed up lookup operations by exploiting the information available at the MAC layer due to the 1-hop broadcast communication that occur in WMNs. A peer resolves a lookup request that is physically close to the peer issuing the lookup, while they are far away in the Chord ring. Packets are received at the MAC layer and then sent to the application layer for further processing. If the packet was not destined to it, then it checks if it may resolve the lookup operation by sending a message containing its own ID to the peer that invoked the lookup. This accelerates the lookup operation.

2.4.3 Caching Schemes

Caching has been used to hasten internet access speeds and reduce latency in traditional Internet networks. In WMNs, caching is often done at the gateway since most of the content accessed resides outside the mesh network. Niesen *et al.* propose a scheme for effective caching in wireless networks, with a view to improve bandwidth utilization

Das *et al.* (2006) propose MeshCache: a transparent cooperative caching scheme as a means to mitigate the gateway bottleneck problem in WMNs. MeshCache exploits the locality in client request patterns in a WMN thus providing better client throughput by enabling content downloads from closer high-throughput mesh routers. MeshCache is loosely coupled with the underlying transport and routing protocols to maximize deployability and exploit cross-layer information awareness. This way, the congestion bottleneck that exists at the gateway node is alleviated.

Ditto was proposed by Dogar *et al.* (2008). It is a system for opportunistic caching in multi-hop WMNs. Content is cached by a mesh node when it is sent to a client, or when the mesh node overhears it being transferred by other nodes. Content is divided into chunks of that can be cached in the mesh nodes along the data path and in mesh nodes overhearing the wireless transmissions.

The Ditto system is similar to hierarchical web caching, where each mesh node serves data to its previous hop, either from its cache or by requesting it from its next-hop mesh node. Each node in the data path caches all chunks passing through it and overheard chunks.

The hit ratio in Ditto is increased by having every mesh node cache overheard objects and by caching chunks passing through it. Ditto lacks cooperation between caches as on failure to find the chunk in the caches, the content chunk is looked up along the path to the data source and redundant cached copies are not eliminated thus leading to inefficient utilization of the caches' storage capacity.

2.4.4 Other WMN Content Distribution Schemes

2.4.4.1 Content Prefetching

In Prefetching, software running in user devices autonomously fetch content from storage servers or from peers thus making it available when needed. Venkataramani *et al.*(2001) argue that prefetching of content reduces latencies and cost of delivery of content. An example of a scheme that utilizes prefetching in wireless networks is CDPAN (Chaitanya, 2008).

2.5 Summary of Related Work

Previous research on the Content distribution problem in WMNs can be divided into three major approaches: Overlay-networking, Content-Caching and Content-Replication schemes. Overlay networks refer to where the underlying layers of the network (PHY,MAC) are transparent to the client device. Both Content-Caching and Content-Replication rely on duplicating objects and placing them closer to the client.

A Content placement and replication scheme for WMNs was designed and simulated Al-Arnaout (2014). This objective of this scheme was to increase Content availability within the WMN, discover what Content to be replicated, determine number of replica servers within the WMN, find an efficient way for selecting the best replica server to serve clients' requests and alleviate the bottleneck problem by fetching Content from within the WMN.

Canali *et Al.* (2010) proposed MeshChord which makes use of Peer-to-peer (P2P) distributed hash tables (DHT) and is an adaptation of the CHORD protocol (Stoica *et Al.*, 2003) for WMNs. It works by assigning location-based IDs to mesh peers .

Al Asaad, *et Al.* (2009) proposed P2PMesh which establishes a P2P network over a WMN. When a file is requested by a client, the file's metadata is hashed to obtain a key which is sent to the peer's current mesh router. The overlay network, P2PMesh, tries to locate the requested file within the WMN. If the file is not in the peer's mesh router, the key is routed to the mesh router responsible for that key using a DHT routing protocol. If the file is not found in the WMN, the key is sent to the gateway fetch the file from the Internet.

The work by ElRakabawy and Lindemann (2007) proposes a peer-to-peer Cooperative file transfer protocol. It works by incorporating a peer discovery as well as a peer selection mechanism.

Caching based solutions where mesh routers cache data sourced by mesh clients have also been proposed by Ditto (Dogar *et al.*, 2009). In their solution, data is cached by a mesh node when it is sent to a client, or when the mesh node overhears it being transferred by other nodes. The mesh node divides Content into chunks of data that can be cached by the mesh nodes along the data path and in mesh nodes overhearing the wireless transmissions.

Al Hamra *et al.*, 2006 proposed Network Coding which broadcast capacity of the network by allowing intermediate nodes inside the network to code and decode the information carried by the different flows.

Liu *et al.*, 2007 proposed UPAC, a unified peer-to-peer and cache framework for high quality video-on-demand services over infrastructure multi-hop wireless mesh networks. In this scheme, nodes in the WMN cooperatively download chunks of video content in a best effort manner to reduce the workload imposed on the servers and networks. The video Content is cached at selected wireless mesh access nodes in the WMN. The client device then establish a Client-Server relationship with the mesh node content cache servers and downloads the Video.

The main limitations with these schemes is that most rely on P2P or are adapted to specific Content types or traffic types. These schemes are still limited by inherent design problems of traditional host-centric networks.

The most similar scheme to our work is CHANET (Amadeo, 2011) and CroWN (Amadeo, 2012). These schemes explore the use of CCN in mobile-ad hoc Networks (MaNETs) and in Vehicular Networks (VANETs).

Table 2.2 gives a summary of related work.

Table 2.2: Summary of Related work

Scheme	Approach	Main Feature
Al-Arnaout (2014)	Content Replication	In-network Content replication
MeshCHORD (Canali <i>et Al.</i> , 2010)	P2P Content Replication	Peer-to-peer network
P2PMesh (Al Asaad <i>et Al.</i> ,2009)	P2P	Peer-to-peer network
P2P cooperative file transfer , (EIRakabawy and Lindemann ,2007)	P2P	Peer-to-peer network
Ditto (Dogar <i>et al.</i> , 2009)	Caching	Overhearing
CHANET (Amadeo, 2011)	Content-Centric Networking (CCN)	Uses CCN
CroWN (Amadeo, 2012)	Content-Centric Networking (CCN)	Uses CCN
UPAC (Liu <i>et al.</i> , 2007)	P2P Overlay Network	Adapted for video streaming
Network Coding (Al Hamra <i>et Al.</i> 2006)	Network Coding	Adapted for multicast applications

2.6 Summary of Chapter

This chapter gives an in-depth introduction to WMNs and their characteristics through the OSI model approach. The chapter starts by describes the IEEE 802.11 standard and its derivatives. It then describes the foundations of WMNs as an extension to traditional IEEE 802.11 Access points. It describes the physical and MAC layer techniques that advanced IEEE 802.11, and consequently, WMN data throughput capabilities.

Content distribution strategies on both wire-line Internet and in WMNs are then described. This section starts by describing the limitations of the host-centric Content distribution model and efforts made to address these limitations: Content-Distribution Networks, Peer-to-peer networks, multicast networking and network coding strategies.

The concept of Information-Centric Networking (ICN) is then introduced along with proposed schemes for Information-Centric Networking. The advantages of ICN over host-centric networks are discussed.

The chapter concludes by discussing Content distribution strategies developed for WMNs. In the next chapter, the research approach and design are discussed.

CHAPTER THREE

3 RESEARCH METHODOLOGY AND DESIGN

Research methodology refers to the means with which Research work is carried out, and in its sequential logical manner. Kothari (1990) and Mouton (1996) describe methodology as ways, means or methods with which something is done.

Methodology consists of problem enunciation, hypothesis formulation, collecting data, fact analysis and finally laying down the drawn conclusions (Kothari, 1990). The research process is taken within a framework of a set of philosophies, uses a procedures, methods and techniques that have been tested for their validity and reliability. A good research approach is also designed to be unbiased and objective.

This chapter explains in detail the research methods used in this research study; the work plan of the research (Rajasekar, Philominathan & Chinnathambi, 2013). The research approach took was the particularistic (Welman, Kruger & Mitchel, 2005) type, where the study took an interpretive approach to research and by specifically making use of an simulation research method to put together different phenomena and develop the system.

The deductive nature of the study required me to use an experiment whereby a simulation was used to study Content distribution in a WMN.

To have control over the factors that could cause deviation from the findings or the validity of the research study, a blueprint of the study needs to be constructed. According to Burns and Grove (2001), a well-designed study increases the chances of researchers to finding and producing intended information. This helps and guides the researcher from somehow not deviating from the intended study.

3.1 Research Approach

Babbie (2006) explains and differentiates between two known research approaches: quantitative and qualitative techniques. The former refers to a systematic and objective use of numerical data, while the latter findings are on measures in terms of quality; rigorous presentation in textual form using words (Maree, 2007). Table 3.1 below summarizes the differences between these approaches.

Table 3.1: Differences between Quantitative and qualitative research methodologies

Orientation	Quantitative	Qualitative
Assumption about the world	A single reality, i.e., can be measured by an instrument.	Multiple realities
Research purpose	Establish relationships between measured variables	Understanding a social situation from participants' perspectives
Research methods and processes	<ul style="list-style-type: none">- Procedures are established before study begins;- A hypothesis is formulated before research can begin;- Deductive in nature.	<ul style="list-style-type: none">- Flexible, changing strategies;- Design emerges as data are collected;- A hypothesis is not needed to begin research;- Inductive in nature.
Researcher's role	The researcher is ideally an objective observer who neither participates in nor influences what is being studied.	The researcher participates and becomes immersed in the research/social setting.
Generalisability	Universal context-free generalizations	Detailed context-based generalizations

To discover answers to any questions relating to research information, with the use of scientific procedures, there are a number of groupings with which objectives of research fall into and can be used. Babbie (2010) names those objectives or purposes as: to explore a topic, describing an event, a process etc., or an explanation of causality between events or variables. The latter can be seen as hypothesis testing where a hypothesis is tested for a causal relationship between variables.

Bouma and Atkinson (2005), and Neuman (2007) explain the two types of research: applied and fundamental research types. Since the applied type aims at presenting a discovery to solutions from a practical problem, fundamental research type is then aiming at solving qualitative solutions. Furthermore on applied research type, an empirical research is said to be applied in nature, and an experimental research method is an example of an empirical type of research.

The first thing done in this research was to provide a working hypothesis or rather an assumption or a guess that using multiple positioning methods will improve accuracy of the positioning system that is desired to be built. The second step was then to gather information or collect data on the various positioning methods by use of literature analysis method. The gathered data was to find out how these methods can be should and could be used in combination to form a multi-positioning system.

The third step was then the experimentation stage where the collected data was manipulated and methods used in combination to test the assumption and bring fourth the desired information. In conclusion, this study attempts to meet the explanatory purpose of the research, and test an assumption by use of an experimental research design.

3.1.1 Conceptual framework

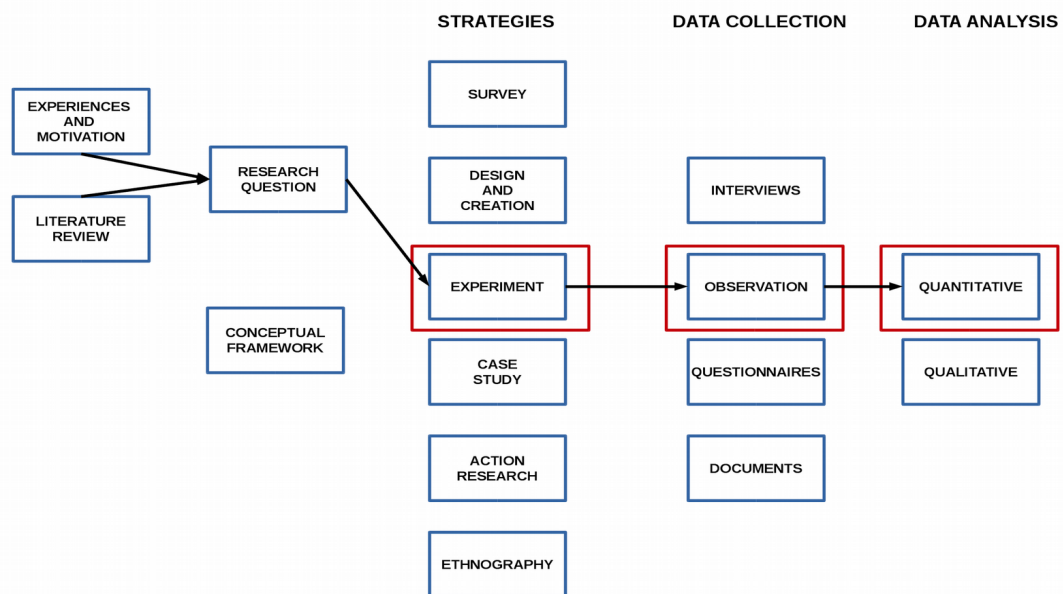


Figure 3.1: Conceptual Framework

3.1.2 Overview of Simulation As A Research Tool

Simulation is growing in popularity as a methodological approach. While other research methods make various assumptions about the exact cause and effect of the system under study, Simulation allows researchers to assume the inherent complexity of organization systems as a given. Simulation answers the question “what if?” while other research methods answer the question “what happened?, and how, and why?” (Dooley, 2002). Simulation enables studies of more complex systems because it creates observations by “moving forward” into the future, whereas

other research methods attempt to look backwards across history to determine “what happened, and how”.

There are three main types of simulation (Dooley, 2002)

1. Discrete event simulation - Discrete-event Simulation can be defined as the utilization of a mathematical or logical model of a physical system that portrays state changes at precise points in simulated time. Each event in a simulated system occurs in an instant of time and marks a change in the state of the system. The start-time and end-time of each event in the system must be defined.
2. System Dynamics - This involves identifying the key state variables that define the behaviour of the system, and then relating those variables to one another through coupled differential equations.
3. Agent-based simulation - This involves agents that attempt to maximize their fitness (utility) functions by interacting with other agents and resources.

In this research study, the Discrete event simulation strategy is utilized. Discrete event simulation may adopt several forms.

1. A single, large, relatively static model that serves over a protracted period of use, e.g. a weather simulation.
2. A single model which evolves rapidly during experimentation for system design or optimization, e.g. a cache model.
3. A model which consists of a synthesis of results from several existing models in an effort to answer questions on a metasystem level.
4. Models used for analysis.
5. Models used to animate and visualize systems.
6. Models used to provide an interactive training environment.
7. Models used to stimulate hardware prior to operational deployment.

3.1.2.1 Purpose of Simulation

Axelrod, (1997) outlines seven different purposes of simulation: prediction, performance, training, entertainment, education, proof, and theory discovery. These are outlined below.

1. Prediction - The Simulation takes a model, composed of a structure and rules that govern that structure and produces output (observed behaviour). The validity of the prediction depends on the validity of the model. Simulation for prediction is a

substitute for experimentation and intervention on the actual system. It is undertaken when such experimentation is too dangerous, costly, untimely, or inconvenient.

2. Performance - Simulation can be used to perform real tasks such as diagnosis or decision making.
3. Theory discovery - Phenomena that were previously unknown can be uncovered in a simulation study and may lead to the development of new theories.
4. Training - A simulation environment makes it quick, easy, and safe for users to make decisions that mimic the decisions they will make in reality. "Flight simulators" for example, fall into this realm.
5. Education - As opposed to using simulation for training purposes, simulation can also provide users more general education about how complex systems work. Users can gain a deeper conceptual, and perhaps metaphorical and symbolic understanding of mechanisms such as feedback, noise, reciprocity, self-organization, non-linearity, etc.
6. Entertainment - Simulators can also provide entertainment. An example is the flight simulator games where people fly virtual aircraft.
7. Proof - Simulation can be used to prove existence of a possible solution to a problem

3.1.3 Research Approach Using Simulation

The implementation of a simulation model involves the following steps (Dooley, 2002, Gilbert and Troitzsch 2005, Davis *et Al.* 2007)):

3.1.3.1 Conceptual Design

The purpose of this stage is to identify a research question that is suitable for study by simulation. The user must determine what is to be modelled, what questions are to be asked via such a model, who will be using the model, and what their requirements are. The questions to be answered at this stage are:

- What is being modelled?
- Who will be using the model?
- What are the requirements of the model?

3.1.3.2 Code Development

The computational model is implemented in software code and tested. High level languages such as C++ , Java or Python can be used to develop the simulator.

3.1.3.3 Validation

Verification is the process of ensuring that the model behaves as intended, usually by debugging or through animation. Verification is necessary but not sufficient for validation, that is a model may be verified but not valid. Validation ensures that no significant difference exists between the model and the real system and that the

model reflects reality. Validation can be achieved through statistical analysis. Additionally, face validity may be obtained by having the model reviewed and supported by an expert.

3.1.3.4 Experimental Design

The user designs a set of experiments, indicating particular initial and run-time parameter values that will be used to answer the questions posed.

3.1.3.5 Implementation

This is the execution of the experiment design, which will likely include replicates; a replicate is a multiple run of a single experimental condition, but using a different stream of inputs. Harrison *et Al.* (2007) identify five elements to such an experiment: the initial conditions, the time structure, outcome measurement, the number of iterations and any variation in model parameters or initial conditions. Variation allows different assumptions to be tested in order to answer the research questions and also to test the sensitivity of the model to changes in parameters.

3.1.3.6 Analysis

Replicates are averaged over an experimental condition for subsequent analysis, and the standard error of replicates can be used to determine confidence intervals for other subsequent work. The user analyses the output data according to the model being hypothesized. Gilbert and Troitzsch (2005) note that providing enough detail for the study to be replicated while avoiding burying the reader in detail can be a particular challenge when reporting simulation research.

3.1.3.7 Interpretation

Observations from analysis are noted, and results are discussed in order to make sense.

3.2 Research Design

The purpose of this section is to present not only the design requirements for the simulation application, but also to give the rationale behind the design requirements. The research's main objective was to develop a simulator to investigate ICN content distribution in a WMN.

3.2.1 Design Requirements

In this section, design requirements for the Simulator application developed in Chapter 4 are given.

3.2.1.1 Network Design Requirements

This refers to requirements pertaining to the simulated network components and topology. For the simulation to as close as possible to real-world, the following requirements must be met.

- Ability to accurately represent physical channel characteristics, MAC characteristics. This is important because WMNs, unlike wired networks suffer from degradation in the physical channel resulting from distance or fading. This in turn affects network performance.
- Ability to accurately represent network components, network interfaces and the connections between different nodes. The simulated WMN will comprise of different types of nodes with different network interfaces. Nodes that form the WMN (mesh nodes) must connect to other nodes through a wireless Interface. Connections to nodes not participating in the WMN must be done in a manner that does not interfere with the WMN.
- Flexibility in defining network topology. The developed simulator should not confine the researcher to specific network topologies. The user should be able to define and simulate custom WMN topologies.
- Ability to simulate host-centric networks. The reason for this requirement is because host-centric WMNs are well studied and can therefore be used to gauge the accuracy of the simulator.

3.2.1.2 Content Model Requirement

The primary purpose of the simulator is to experiment with ICN. Content in the simulated networks

- Accurately represent NDOs as described in section 2.3. This involves identifying Content objects with a unique identifier.
- Accurately represent NDO size. The NDO representing Content should have a realistic data size in order for Content requests and transfer to be realistic.

3.2.1.3 Client Behaviour Requirement

The role of clients or content consumers in the network is to initiate data transfers by requesting for Content. In the simulated network, content request patterns should approximate the behaviour of real-world Internet users.

3.2.1.4 User Interface Requirement

The developed simulator application should have an Interface through which the user issues commands to the simulator core. The requirements for the user interface are:

- User Interface design should be consistent with Multiple document interface (MDI) structure that most computer users are familiar with. This requirement should make the application easy to use.
- The user interface should accurately represent network nodes, respective positions and connections between nodes of different types. This is to give the user an indication of the network under simulation.
- Ability to change simulation settings from the user interface. This requirement is in line with the ease of use requirement.

3.2.1.5 Data capture and display requirement

The final set of requirements of the developed application pertains to data collection and analysis. These are:

- Ability to capture data listed in the Research Metric's section below.
- Ability to capture network traffic at different layers of the network.
- Ability to present gathered data in a user friendly format. This refers to the ability to display data in graphical form that users are able to easily understand.

3.3 Research Metrics

Research metrics refers to what or whom is to be studied and is referred to as metrics or unit of analysis. Babbie (2010) notes that although the study of units of analysis may seem a bit slippery, distinguishing between the units of analysis and the aggregates is considered very crucial in research. Units of analysis may also be called the units of observation. Thus, to study a certain thing one would have to study the behaviour of that particular thing. In any case, the conclusions wished to be drawn from these units of observation have to be well anticipated. In this research, the metrics used for evaluation are:

3.3.1 Hit Ratio

This refers to the ratio between the total requests satisfied to the total number of requests sent. The hit ratio is averaged on a client node that requests Content from a server. The formula for hit ratio can be expressed as

$$\text{Content Hit ratio} = \left[\frac{\text{Content Hits}}{\text{Content Hits} + \text{Content Misses}} \right] \times 100 \%$$

3.3.2 Network Load Reduction

Network Load refers to the total number of packets transmitted by the network layer for a single transfer. This accounts for all control as well as data packets. The network load is averaged across all the transfers initiated in the network. Thus, the

network load is expressed as packets transmitted per file download. Network load reduction can be expressed as follows:

$$\text{Network Load Reduction} = 1 - [(\text{Total packets transmitted in ICN WMN}) / (\text{Total packets transmitted in host-centric WMN})]$$

3.3.3 Average Throughput

Throughput of any transfer is the ratio of the size of the transfer to the time taken to complete the transfer. Most network transfers utilize the transmission control protocol (TCP) where the receiving node acknowledges reception of data before the transmitting node can send more. There are many overheads accounted for in throughput in addition to transmission overheads, including latency, TCP Receive Window size and system limitations, which means the calculated throughput does not reflect the maximum achievable throughput. Network throughput can therefore be expressed as:

$$\text{Network Throughput} \leq \text{RWIN} / \text{RTT}$$

RWIN (TCP Receive Window) is the amount of receive data (in bytes) that can be buffered at one time on a connection. The sending host can send only that amount of data before waiting for an acknowledgement and window update from the receiving host.

RTT (Round Trip Time) is the time it takes from the time the request for Content is made to the time the Content is received at the Client node.

3.3.4 Hop-Count

This is the total number of hops between the requesting nodes (consumers) and the serving nodes (producers) divided by the total number of served requests.

$$\text{Hop Count} = (\text{Number of hops between consumers and producers}) / (\text{Total number of served requests})$$

3.3.5 Server Load Reduction

This is defined as 1 minus the ratio between the number of Content objects generated by producers and the number of Content objects generated by producers when all requests are served from the producer node rather than from the cache of any other node. In this work, the server load reduction is calculated by comparing the ratio between Content objects generated in the ICN WMN versus the Content objects that would have been generated in the equivalent host-centric WMN.

Server Load Reduction = $1 - [(\text{Content objects generated in ICN WMN}) / (\text{Content objects generate in host-centric WMN})]$

3.3.6 Latency

This is the time interval between the sending of a packet and its receipt at the target node. Network communications can be affected by delays on the physical channel or during the routing of traffic. In Network communications, Latency can be expressed as:

Latency = (Frame Serialization Time) + (Link Media Delay) + (Queueing Delay) + (Node Processing Delay)

3.3.7 Communication Overhead

This is the difference between the total bytes sent in a single transfer and the actual amount of data contained in the transfer. Sending data reliably over the network requires sending more than the actual data itself. The overhead mainly comprises of control messages.

(Communication Overhead) = (Total Bits in Network transfer) – (Actual bits of data Contained in transfer).

Communication overhead can also be expressed as a percentage.

(Communication Overhead)% = $100 \times [(\text{Total Bits in Network transfer}) - (\text{Actual bits of data Contained in transfer})] / (\text{Total Bits in Network transfer})$

3.4 Data collection/gathering method

Data gathered through experimentation as well as observation and recording of events during the simulation experiments.

3.5 Data analysis methods

Data analysis is an important stage in research where data is inspected, transformed and manipulated in ways which to help extract only the needed and useful information since the collected data can be too large and sometimes comprises of wanted and most likely unwanted data. In this study, a quantitative data analysis approach was taken because of the study's precise nature of statistical techniques. The analysis method used was the statistical analysis method (Rajasekar, Philominathan & Chinnathambi, 2013), which is said to be a quiet involved analysis technique.

The initial analysis of the data collected was in different techniques. The analysis techniques used included, percentages, measures of variability and that of central tendencies like mean, median and deviation. The information is then presented using graphs and tables.

3.6 Summary of chapter 3

This chapter explains in detail how the research was conducted and the methodology followed. It also lists the design requirements for the Simulator to be implemented in Chapter 4. These requirements do not actually specify an implementation but serve to restrict the implementation to one within a range of acceptable limits. Chapter 4 explains how these requirements are instantiated in the implementation.

CHAPTER FOUR

4 SYSTEM IMPLEMENTATION

The simulator application developed for this thesis is named IcnMesh. This chapter briefly describes the current implementation of IcnMesh primarily with respect to its class structure and interfacing with external APIs. The description includes references back to the design requirements in the previous chapter. This chapter also describes IcnMesh's user interface and the data measuring system.

4.1 Development Environment Selection

The primary goal of software development is to deliver high-quality software efficiently and in the least amount of time possible. To achieve these goals, programmers use existing frameworks, sections of code, templates, functions, and and libraries instead of developing similar code artefacts from scratch.

The challenge in using existing frameworks or libraries is to understand the usage of Application Programming Interfaces (APIs) exposed by those frameworks or libraries. There may be existing frameworks or libraries are not well-documented or may have outdated documentation. Therefore, in implementing IcnMesh simulator, the first challenge was to identify suitable APIs for its development.

4.1.1 User Interface

There are several UI libraries in existence. The QT development environment was chosen due to its flexibility and constant development. Qt is a cross-platform application development framework widely used for the development of application software that can be run on various software and hardware platforms with little or no change in the underlying codebase, while having the power and speed of native applications.

4.1.2 Network Simulator

There are a number of Open-source network simulators available for research. Some of these are outlined below. The challenge was to identify a well-featured and easy to integrate simulation library.

4.1.2.1 OMNET++

OMNET++ has been in development since 1997. It was designed for modelling of multiprocessor hardware and software systems. This modelling environment is a

modular environment with a set of libraries for modelling. The main advantage of OMNET++ system is built-in graphical system for development of simulation models and modelling process. All main models in OMNET++ are coded in C++.

4.1.2.2 GloMoSim

Global Mobile Information System Simulator is a simulation environment, that's being developed for modelling of massive wireless networks. GloMoSiM was designed with a parallel discrete event simulation, that allows to lower modelling time. For this purpose Parsec language was developed, that was based on C language with prioritization of task paralleling. Currently it's not under active development or improvement.

4.1.2.3 Network Simulator 3

Network Simulator 3 (NS-3) is a discrete-event network simulator for Internet systems, targeted primarily for research and educational use. The NS-3 project was started in 2006 as a continuation of Network Simulator 2 (NS2) and was developed from ground-up, using parts that was previously developed in NS2, to create a complete software solution. The NS-3 environment is supplied in the form of libraries and examples of models, that help in creation of custom simulation models. NS-3 comes with is a graphical visualiser PyVis and system for simulation recording and playing Network animations.

For this thesis, NS-3 was selected because being provided in the form of libraries, it has great capabilities for advancements and modification and can output results of simulation in several forms.

4.2 Development Process

The development method used to create IcnMesh was iterative prototyping. That is, the system was original conceived in a "sketch" form which only outlined the major subsystems, such as the nodes, Content, NS-3 simulator back-end and user interface. These subsystems were then mapped into classes. Relationships between these classes was then defined. "Dummy" implementations where written to create a runnable ,but not functional, system. As each of the subsystems were gradually refined, their function and relationships could be confirmed or re-worked. Much of the design and code was discarded when strategies did not pan out. In this way, working models were created very early in the development period, which allowed for very early informal usability testing.

IncMesh was developed over a five month period, including design, implementation, testing, debugging, and learning the NS-3 Simulator API. The IncMesh application is about 6000 lines of code. The largest portion of the time was taken by iterative design and testing of the practicality of ideas in the context of the development environment.

The core of the IncMesh application was written in C++ and the user interface was written in QML.

4.3 Resource Requirements

IncMesh was designed to run on an Intel X86 compatible computer running the Linux operating system. Table 4.1 summarises the hardware and software tools that were used to develop IncMesh.

Table 4.1: IncMesh Resource requirements

Tool	Description
HP 15-r237ne Intel Core i7 8 GB RAM Nvidia GeForce GPU	Base hardware
Fedora Linux version 25	Base Operating System
QT 5.8	QT 5.8 Software development Kit
NS-3-3.26	NS-3 network simulator
WireShark	Traffic analysis software
RScript	Front-end software for R language – statistics ,data analysis and graphing.
Gcc-6.1	Compiler suite
Doxygen	Tool to generate UML from C++

4.4 Architectural Overview

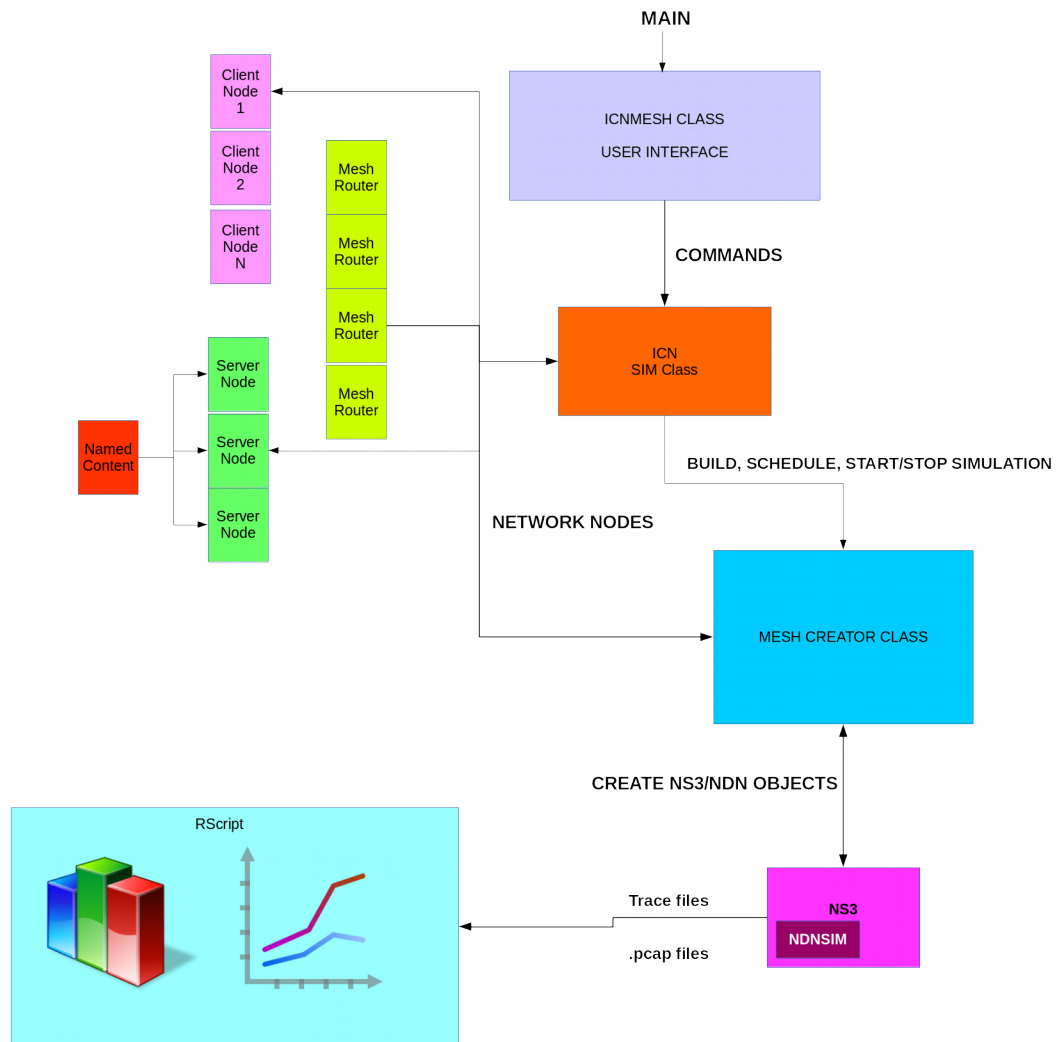


Figure 4.1: IcnMesh Architecture Overview

IcnMesh consists primarily of a set of classes which are accessed by an outer interface, or application window. An Object-oriented programming approach was used whereby the programming language model is organized around objects rather than “actions” and data rather than logic. Therefore components making up the simulator were represented using classes. The program flow of the IcnMesh application is represented in figure 4.1. When the program is started, the main function creates an IcnMesh object whose main function is to call the libraries required for the user interface, open the user interface QML file and links its functions with the IcnSimulator class.

The IcnSimulator class is the bridge between the UI and the core program. All commands from the UI functions are handled by an object from this class. It handles functions such as creation and deletion of network nodes, scheduling of simulator

events, starting and stopping the simulation. This class also holds an array of network nodes and links with the MeshCreator class. The mesh creator class creates equivalent NS-3 objects and topologies for the the simulation.

A simulation is started by calling the Run() function of the IcnSimulator class. If all conditions were set and there are no errors in the topology, simulation events are run as scheduled. While the simulation runs, traffic to and from the interfaces of all nodes is captures in the pcap packet capture format. When the simulation is completed, the user can view the pcap files generated graphs and reports using Wireshark.

4.5 Application Classes

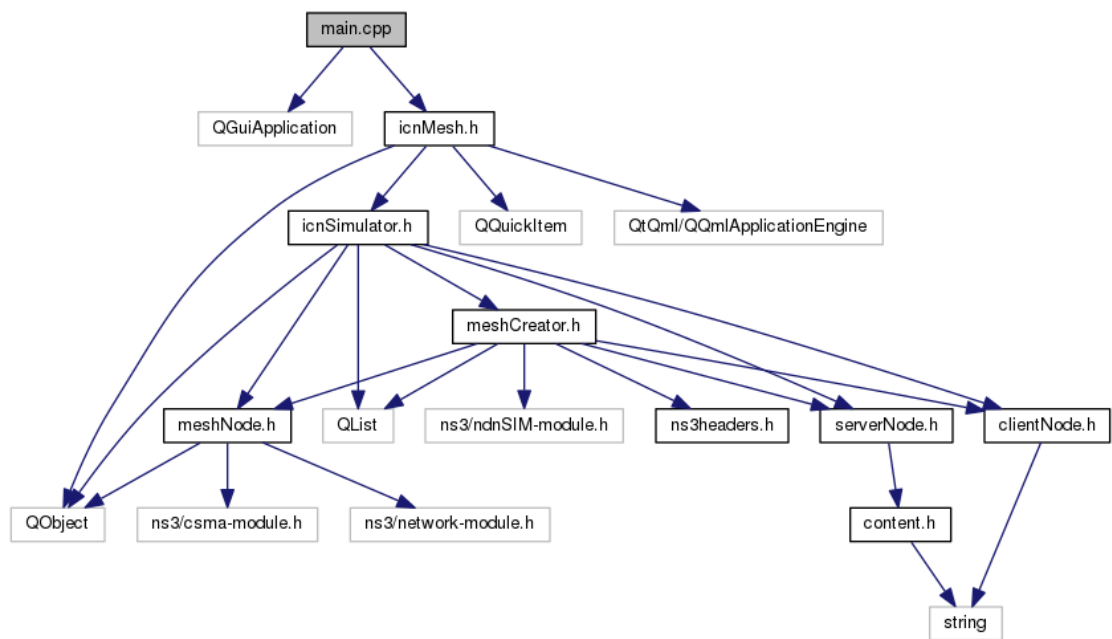


Figure 4.2: Dependency chart for IcnMesh

In the IcnMesh source code, each class is defined in a separate header file. Figure 4.2 above shows the interdependency between header files, and thus the interdependency between classes. The classes are described in the sections below.

4.5.1 IcnMesh Class

This is the first class instantiated when the program is started. The main role of this class is to create the user interface and to link it with the IcnSimulator object. The user does not interact directly with this class. This class inherits from

QQmlApplicationEngine class which allows a QML object (m_qmlItem) to be instantiated from a C++ class. The class constructor opens a QML file from which the user interface is loaded.

The IcnMesh class also creates an IcnSimulator object (m_sim) and links it with the UI. This class contains no functions other than the constructor and destructors. All the work in this class is done in the constructor. A diagram representing this class is given in the figure 4.3.

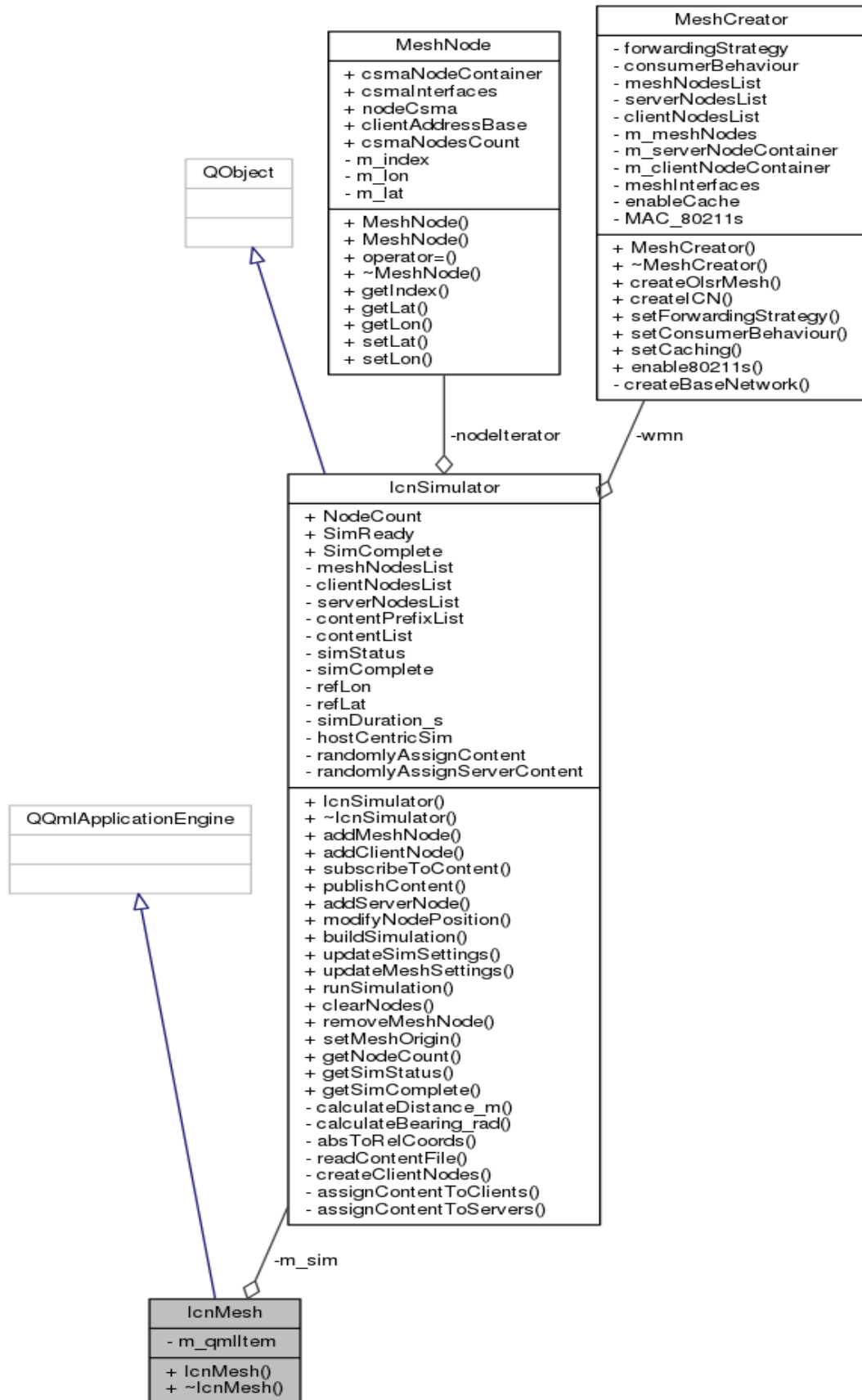


Figure 4.3: IcnMesh class collaboration diagram

4.5.2 IcnSimulator Class

The role of this class is to receive commands from the user Interface, interpret them, and call required functions from underlying classes. The connection between the QML UI and the C++ class is accomplished through the QT Meta-Object System which provides signals and slots mechanism for inter-object communication, run-time type information, and the dynamic property system.

This class holds arrays of various nodes created on the UI. When a topology object is created, it is held in a corresponding array of its type that is, meshNodesList array holds meshNode objects, serverNodesList holds serverNode objects and finally clientNodesList holds clientNode objects. It is through this class that simulation events are scheduled in addition to starting and stopping the simulator.

This class is also responsible for creating virtual content through methods implemented in the class. Content prefixes are loaded from a text file through the readContentFile() function. Through the assignContentToServers() function, an object of this class will randomly assign content of different prefixes and to producer nodes in the simulated mesh network, and through the assignContentToClients() function, content is assigned to clients following a uniform distribution. This is especially important for large simulations where it would be tedious to manually assign Content to producer and consumer nodes.

This class is also responsible for reading positions of meshNodes as created on the UI map, converts read GPS locations to relative coordinates which are conveyed to the NS-3 positioning module when building the simulation model. This is done through the absToRelCoords(), calculateDistance_m() and calculateBearing_rad() functions implemented in the class.

The IcnSimulator class also instantiates the MeshCreator class and calls functions from MeshCreator in response to commands from the UI. MeshCreator class builds NS-3 objects from the topology objects created in IcnSimulator class.

4.5.3 MeshCreator Class

An object of this class is instantiated in the `IncSimulator` class. The main role of this class is to convert `IncMesh` objects to equivalent NS-3 representations and to directly call NS-3 libraries. This class has two main public functions that are called depending on the simulation settings:

- `createOlsrMesh()` - This function is called for a host-centric simulation. It in turn installs an IP stack and the OLSR routing protocol on mesh nodes.
- `createIncMesh()` - This function is called for information-centric simulations. It installs the NDN stack on all nodes under simulation.

The `createBaseNetwork()` function is a private function from this class that is called by both `createOlsrMesh()` and `createIncMesh()` functions. The role of this class is to create the physical topology and MAC layers of the network. These two layers are independent of whether the simulation is host or information-centric. When creating the topology, client and server nodes are connected to the mesh nodes through Ethernet (CSMA/CD) links.

4.5.4 MeshNode Class

This class represents a wireless mesh router. It holds the location of the node, mesh

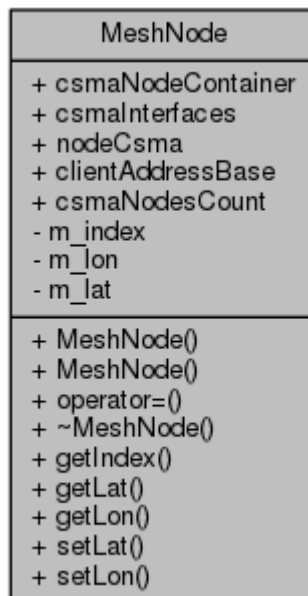


Figure 4.4: MeshNode class diagram

clients and mesh servers that are connected to it. For every WMN node added by the user to the UI, an object of this class is created and pushed in its corresponding array in the `IncSimulator` class.

A MeshNode object also holds the position of the node in GPS coordinates. These coordinates are passed to the NS-3 positioning module during the simulation.

4.5.5 Client Node Class

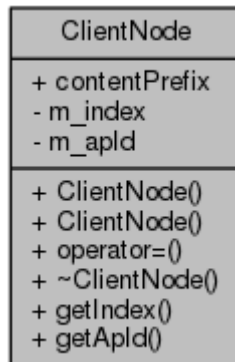


Figure 4.5: Client node class diagram

This class represents a clientNode also known as a consumerNode. It holds the ID of the Mesh Router that the node is connected to. In a host-centric simulation, it holds the IP address of the server to connect to and in the ICN context, it holds the Prefix of content to be fetched from this node.

4.5.6 ServerNode Class

This class represents a Server node or Content producer node.

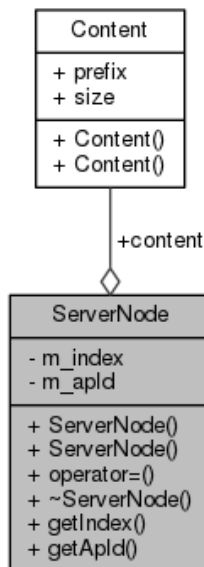


Figure 4.6: Server Node class diagram

The serverNode object holds the ID of the mesh router that the node is connected to and a list of Content objects published by the node.

4.5.7 Content Object

In the ICN context, Content in the network must be uniquely identified. Content is represented through a Content Object. IcnMesh does not use actual Content for simulations but utilizes Content objects with virtual data. The Content object is represented in a Struct with two fields: contentName and contentSize.

contentName is a human-readable identity for the Content and is used to identify a data chunk.

contentSize represents the size of virtual data contained in the Content object in kilobytes.

4.6 Network Simulator 3 API

The structure of NS-3 consists different modules. These are:

- Core Mobility module
- Propagation module
- Data transmission protocol module
- Traffic generation module
- Statistics IO module

Core is the central module of the simulation environment. This module connects to each module through itself and also connects to the model directly. This module is responsible for interaction of modules and order of execution of discrete events in simulation model. Core is responsible for making sure, that simulation is working and calculating things in right order.

The Mobility module is a module that is responsible for locating positions of devices in virtual space. This module consists of algorithms that imitate close to reality movement of devices. This module includes several built-in algorithms for node mobility:

- Constant position
- Gauss-Markov
- Random direction
- Random way

- Random way-point

As mobility is not considered in this work, the ConstantPositionModel is used for all nodes created in IcnMesh.

Propagation module is a module that is responsible for the modelling of wireless signal propagation. This module consists of 2 parts: propagation delay module and propagation loss module. Propagation module in more difficult circumstances may include signal recovery module. Propagation module in its default configuration includes several algorithms of propagation modelling:

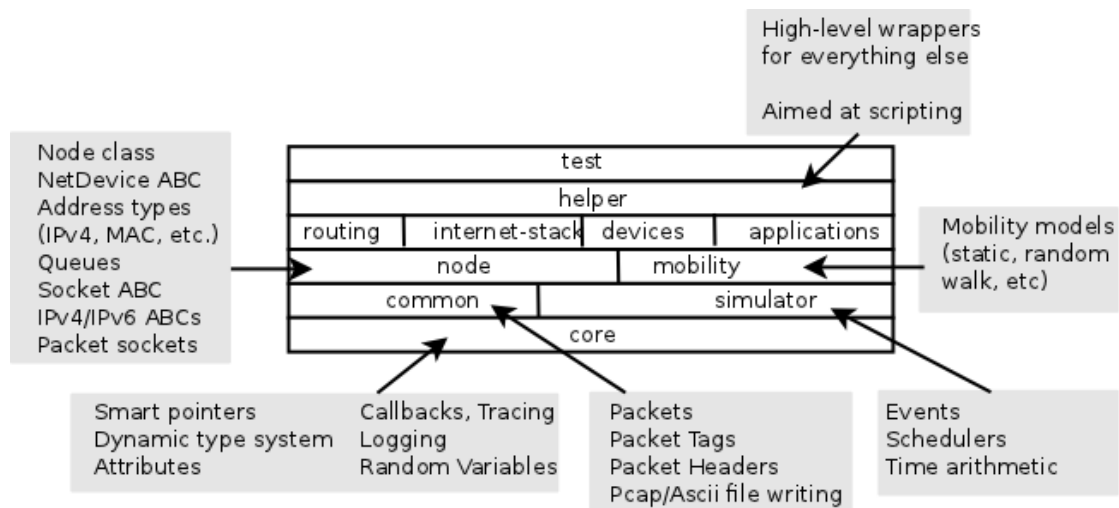


Figure 4.7: NS-3 Simulator modules

4.6.1 ndnSIM

ndnSIM is an extension to the NS-3 simulator that provides Information-Centric networking capabilities. The first public release of ndnSIM took place in June 2012 (Afanasyev, *et al.*, 2012). Since then, ndnSIM has become a popular tool used by many named-data networking researchers worldwide.

The main role of ndnSIM in IcnMesh is to install the ndn stack into nodes and to implement named-data routing. IcnMesh then lets the users simulate different scenarios.

For this thesis, ndnSIM 2.0 which is an updated version of the initial release of ndnSIM was used. NdnSIM 2.0 features the ndn-cxx library ((NDN C++ library with eXperimental eXtensions) and the modular NDN Forwarding Daemon (NFD).

The main classes of ndnSIM 2.0 are listed below.

- `ndn::L3Protocol` - This is the NS-3 abstraction of the NDN stack implementation. Its main task is initialization of the NFD instance of each node that participates in the simulation scenario and provides tracing sources to measure NDN performance.
- NFD: This is the implementation of the Named Data Networking Forwarding Daemon. It includes the following subclasses:
 - `nfd::Forwarder` – This is the main class of NFD, which owns all faces and tables of the NDN router node and implements NDN forwarding pipelines.
 - `nfd::Cs` – This is the cache of Data packets that is used by NFD.
 - `nfd::PIT` – This is the Pending Interest Table (PIT) of NFD keeps track of Interest packets that were forwarded upstream toward one (or more) content source(s). In this way, Data can be sent downstream to one (or more) requester(s).
 - `nfd::Fib` - The Forwarding Information Base (FIB) is used to forward Interest packets toward one or more potential sources.
- `nfd::fw::strategy` – This is the forwarding strategy in NFD makes the decisions regarding whether, when, and where the Interest packets will be forwarded. `nfd::fw::Strategy` is an abstract class that needs to be implemented by all the built-in or custom forwarding strategies.
- `ndn::AppFace` – This is the realization of the `nfd::Face` abstraction to enable communication with applications.
- `ndn::NetDeviceFace` – This is the realization of the `nfd::Face` abstraction to enable communication with other simulated nodes.
- Basic NDN Applications – These are implementations of built-in NDN consumer and producer applications that can generate and sink NDN traffic. These applications include parameters that can be configured by the user in the simulation scenario and thus generate NDN traffic according to a user-defined pattern.
- Trace Helpers - a collection of trace helpers that simplify collection and aggregation of various necessary statistical information about the simulation and write this information in text files.

4.7 IcnMesh User Interface

The IcnMesh UI was developed using the QML mark-up language. It is designed to be user friendly, intuitive and follows the design of most computer software that users are already familiar with. Figure 4.8 shows the IcnMesh UI and the following sections describe its components. Figure 4.9 describes the icons used in the IcnMesh application.

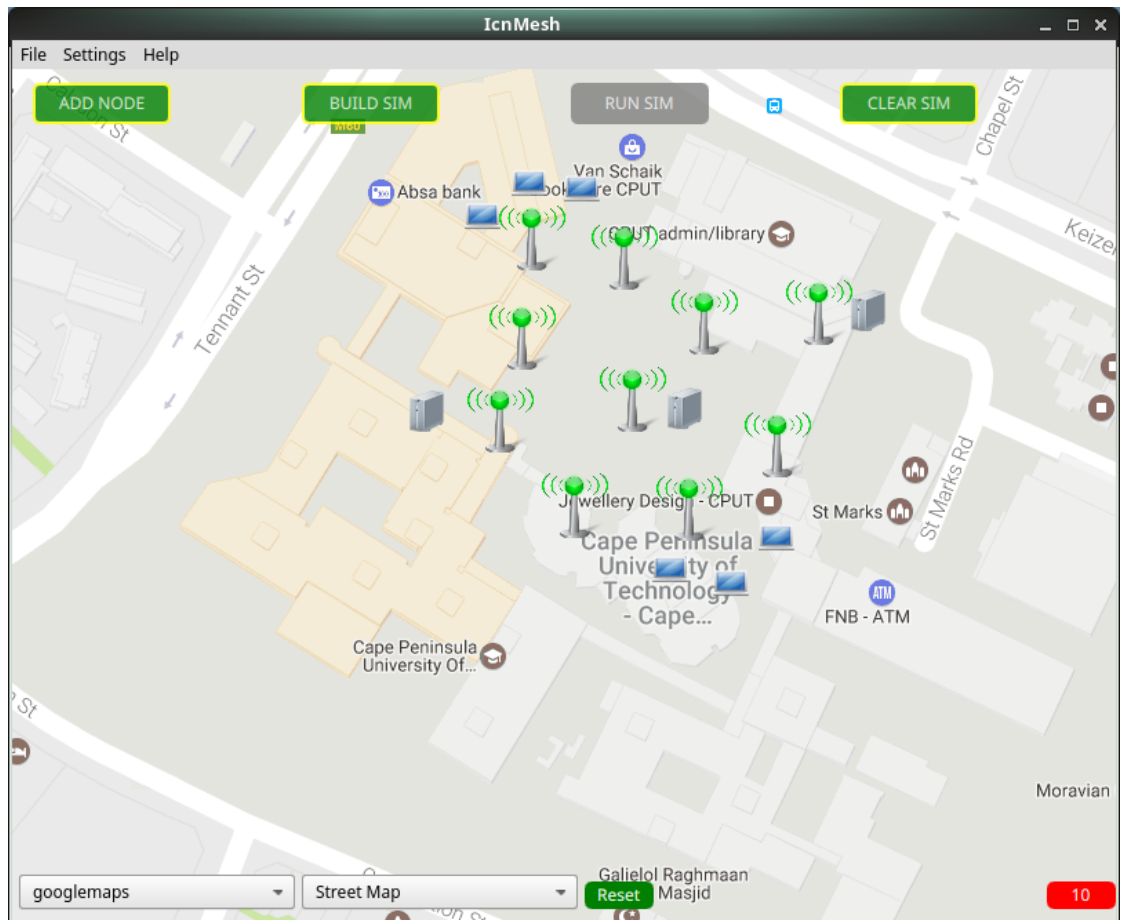


Figure 4.8: IcnMesh User Interface

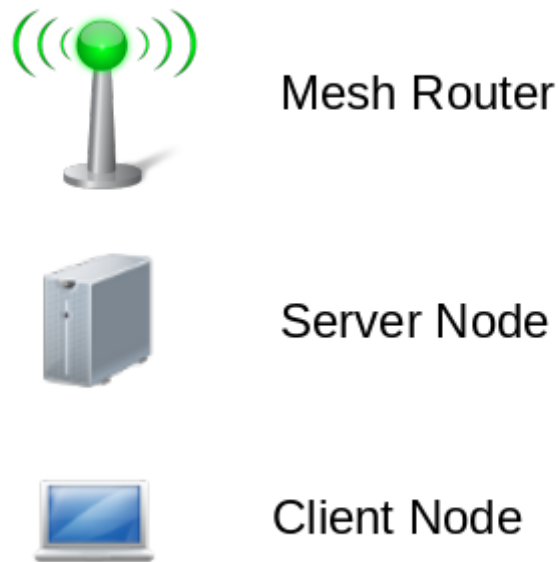


Figure 4.9: Key of icons used in IcnMesh Application

4.7.1 Map Display

The Map display is the central component of the user interface. In order to make simulations as realistic as possible, mesh node positions and distance between WMN nodes must be accurately represented as distance affects signal propagation and thus the strength of received signals.

When a node is added to the map, it's GPS coordinate is passed to the MeshNode constructor. The node position can be changed by dragging and dropping the node icon with the mouse. Consequently, a signal is passed to the corresponding MeshNode object thus changing its position.

A Client/consumer node is added by right clicking on a mesh node and selecting the menu option "Add Client Node". Similarly, a server/producer node is added by selecting the menu option "Add Server Node".

All nodes on the map are named as they are added to the map. MeshNodes are named with the prefix “meshNode” followed by a number (e.g. meshNode0, meshNode1 etc.). Clients are named with the prefix “consumer” followed by the client number and servers are named with prefix “producer”. The naming of nodes is very important when analysing log files.

4.7.2 MenuBar

The IcnMesh menu-bar is kept as standard as possible. Under the File menu, the user can open existing simulation topologies, save the current simulation or close the simulation. Under the settings menu, the user can modify global settings such as simulation run time and the simulation type. Under mesh settings, the user can modify mesh network settings. Figure 4.10 shows the mesh settings dialog.

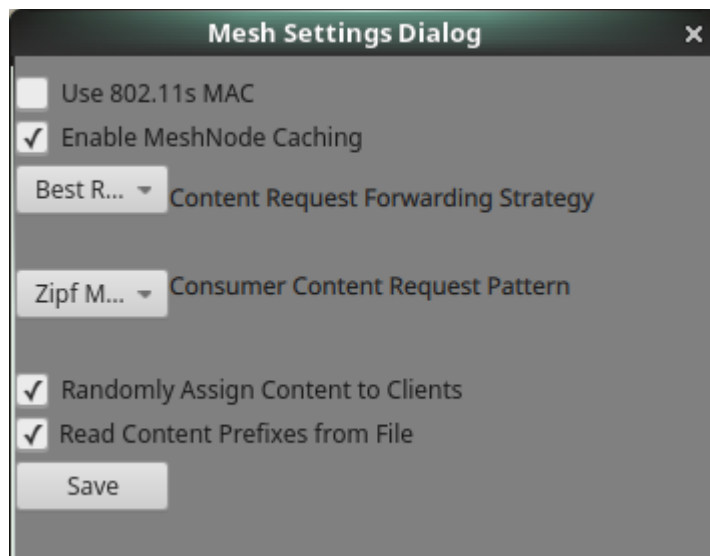


Figure 4.10: IcnMesh mesh settings dialog

4.7.3 ToolBar

The tool-bar consists of buttons and icons that can be used to create network topology objects, start and stop the simulation.

4.7.4 Status bar

This shows the status of the simulation and also displays messages to the user.

4.8 IcnMesh Network Model

This section explains the underlying structure of a network created with IcnMesh that is abstracted from the user. This section explains the properties of the wired and wireless channels used to connect different nodes in IcnMesh.

Each mesh node in IcnMesh is assumed to have one radio interface with an omnidirectional antenna. It is through this interface that communication with other nodes in the WMN is achieved. Each mesh node is also assumed to have an Ethernet Interface for connection to non-mesh nodes, that is, the Client and Server nodes.

4.8.1 Wireless Channel

Table 4.2 summarizes the default parameters of the wireless channel in IcnMesh.

Table 4.2: IcnMesh Wireless Channel Properties

Parameter	Value
Frequency Band	2.4 GHz
Physical Layer standard	IEEE 802.11n
Modulation Scheme	Ofdm
Channel bandwidth	20 Mhz
Propagation Delay	Constant speed propagation delay
Path loss model	Three log distance path loss
Stochastic fading model	Nakagami propagation loss
Transmission power	20 dBm (adjustable)
Mobility model	Constant position

A two-ray path loss propagation model is used to evaluate performance under Rayleigh fading and path loss between mesh nodes. In order to simulate path loss, the three log distance path loss model is applied. The three log distance model is a variation of the log distance model that applies different factors to the logarithmic path loss for different distance intervals (Stoffers, M. and Riley, G., 2012).

In order to simulate the effects of multipath propagation (Rayleigh fading), the Nakagami propagation loss model which is similar to the Rayleigh model, but describes different fading equations for short-distance and long-distance transmissions (Stoffers & Riley, 2012) is applied.

Mesh nodes in IcnMesh are considered static and therefore the constant position mobility model is used. The transmission power from a meshNode can be varied in the settings menu but is by default set to 10 dBm.

4.8.2 Client behaviour

The default behaviour for Client nodes in IcnMesh is to request Content following the Zipf-Mandelbrot Distribution (Montemurro, 2001). Client nodes can also request Content at a "constant" rate or poisson process. For every received interest by a producer node, a data packet is sent to the requester containing bytes of virtual payload.

Several studies have shown that the popularity of Content on the Internet follows the Zipf distribution (Gummadi, *et al.*, (2003) and that Zipf's distribution is the discrete distribution that best represents the request frequency of Content for different Content request patterns, for example, web pages, file Sharing, video on demand and user generated content (Fricker, *et al.*, 2012).

The logic of the Zipf's distribution is that the frequency of a content request is inversely proportional to the rank of the Content itself that is, the smaller the rank the higher the request frequency.

Client nodes connect to mesh routers via Ethernet links. This eliminates complexities arising from wireless link interference. This also ensures a stable connection to client/server node that the mesh router is connected to.

4.8.3 Content model

Content in IcnMesh is identified with a prefix and virtual payload size. Every producer node is assumed to serve only one type of Content. CCN, described in section 2.3.4.2, is the ICN content model employed in IcnMesh.

4.8.4 Routing

In the host-centric scenario, mesh nodes are assigned IP addresses in the 10.1.1.0/24 address space which holds a maximum of 254 mesh nodes. Mesh node Ethernet interfaces, client and server nodes are assigned addresses in the

172.16.x.0/24 address space, where x denotes the mesh node the client or server is connected to, for example, a client node connected to meshNode3 would be assigned an address like 172.16.3.2.

In the host-centric scenario, routing in the mesh network is performed using the OLSR protocol. Routing between nodes not participating in the mesh (client and server nodes) is performed through OLSR's host-network association (HNA). Each mesh node broadcasts the prefix of its HNA network to neighbouring nodes thus enabling data routing to non-OLSR nodes.

In the ICN scenario, routing is different from traditional IP routing in two ways:

- (a) ICN's perform routing on names: Content producers register name prefixes rather than address prefixes.
- (b) Multipath: The routing protocol in an ICN network is expected to provide multiple paths to each name prefix (if such paths exist).

IcnMesh uses ndnSim (Afanasyev, Moiseenko, & Zhang, 2012) to implement ICN. Communication in an ICN network is initiated by the consumer and is based on two types of messages: Interest messages and Data. In pure named-data networks, the routing layer needs to provide routes to name prefixes rather than address prefixes as in host-centric networks. Therefore, when ICN nodes forward Interest packets, they set up state information (Wang, *et al.*, 2012).

An interest message is generated by a client that wants to fetch a particular content. The data message encapsulates the requested Content and is generated by the producer node and from any node that had previously cached the requested content.

An Interest is therefore said to be satisfied by a Data when the Content name in the Interest is either the same or a prefix of the name contained in the generated Data packet.

There are 3 main data structures are used in routing in ICN node:

- The Content Store (CS) – This is a memory location used to store and cache Content.

- The Forwarding Information Base (FIB) – This data structure stores the next-hop information for every known Content prefix.
- The Pending Interest Table (PIT) – This data structure which keeps track of the arrival faces associated to the Interest packets which have been previously forwarded but that are still unsatisfied.

This architecture allows routing and forwarding operations to be made only on Interest packets. Data packets are transmitted to the requester by following the PIT entries created along the path by forwarded interests.

Interests loops are avoided through this stateful routing. Interest packets for the same content can also be aggregated in the same PIT entry by recording the respective incoming faces and forwarding only the first one.

4.9 Measurement System

The main objective of running simulations is to collect data. IcnMesh uses two schemes to gather generated data. These are Packet Capture system and the packet tracing system.

4.9.1 Data Collection System

4.9.1.1 Packet Capture

All the data entering and leaving any network interface is captured and stored in a file in the pcap file format. The structure of a pcap packet is shown in figure 4.11.

offset(h)	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
00	Magic Number			Version_Major		Version_Minor		Timezone			accuracy of timestamps					
01	Max length of captured packet			Datalink Type				Timestamp seconds			Timestamp microseconds					
02	packet(frame) length			actual packet(frame) length				packet data								
03	packet data															
04	packet data			Timestamp seconds			Timestamp microseconds			packet(frame) length			actual			
05	packet(frame) length			packet data												

Figure 4.11: Pcap file structure

The pcap file can be read using any packet tracing application, for example, tcpdump or wireshark.

4.9.1.2 Packet Tracing

In addition to the packet capture system, IcnMesh also offers packet tracing which is particularly useful for ICN traffic analysis. This is because unlike host centric networks where the source and destination of packets are primary and network protocols are well documented, in ICN, the network address source of Content is not important. The other reason is that the ICN protocols have yet to be implemented in pcap and therefore the pcap files will not display the data of interest.

For this reason, packet tracers are used in IcnMesh to capture ICN traffic information. Packet trace helpers collect and aggregate requested statistical information in text files. Unlike pcap, trace files capture traffic in human-readable ASCII format and can be read without special tools.

IcnMesh uses the L3RateTracer defined in ndnSim to track Interests and data packets at every node in the network.

4.9.2 Data Analysis Tools

4.9.3 Wireshark

Wireshark is a network packet analyser. A network packet analyser is used to capture network packets and to display captured packet data as detailed as possible. Wireshark is open source software built on top of the libpcap library. Wireshark is a convenient user interface program that allows direct and easy access to the network packets and makes the information acquisition simpler. In this work, Wireshark was used to read pcap files generated after every simulation.

4.9.4 R-Script

In this thesis, R-Script was used to generate graphs from IcnMesh simulations. R-script is the front interface processor for R language.

R is a language and environment for statistical computing and graphics. It is derived from the S language and environment which was developed at Bell Laboratories by John Chambers and colleagues. R provides a wide variety of statistical techniques, for example linear and non-linear modelling, classical statistical tests, time-series

analysis, classification, clustering etc. R also provides graphical techniques, and is highly extensible. The S language is often the vehicle of choice for research in statistical methodology. R can be considered to be an Open Source implementation of the S language.

R is an integrated suite of software facilities for data manipulation, calculation and graphical display. It includes

- An effective data handling and storage facility.
- A suite of operators for calculations on arrays and matrices.
- A large, coherent integrated collection of intermediate tools for data analysis.
- Graphical facilities for data analysis and display.
- A well-developed, simple and effective scripting language which includes conditionals, loops, user-defined recursive functions, data input and output facilities.

R's main strength is the ease with which well-designed publication-quality plots can be produced, including mathematical symbols and formulae where needed.

4.10 Running a Simulation

The process of running a simulation on IcnMesh and collecting data is summarized in this section.

First, the IcnMesh application is started by invoking the program name on a command line terminal or by clicking in the IcnMesh icon on the desktop.

The next action would be to set simulation and mesh parameters. This is done by clicking on the settings menu and changing settings as appropriate.

The third step is to add mesh nodes. Mesh Nodes are created by clicking "Add node" button on the tool-bar. The mesh node icon appears in the map once the node is added. The default location of a mesh node is at the centre of the map. The user can drag and drop the node at the desired location.

Once mesh nodes have been added, the user adds client or consumer nodes. Client nodes are created by right clicking on the parent mesh node and selecting “Add client node”. The client node icon appears next to the mesh node that it is connected to. Multiple clients can be added using the same procedure.

Like client nodes, server nodes are also added by right clicking on the parent mesh node and selecting “Add Content producer”. Each server node publishes only one type of Content. Multiple server nodes must therefore be added for different Content types.

The final step in setting up the simulation is to publish Content on server nodes and to subscribe client nodes to published Content. By default, IcnMesh automatically assigns Content to producer and consumer nodes following a random pattern. If, however, the user prefers to manually publish Content on producer nodes, “Assign Content to nodes” must be disabled on the settings dialog.

The simulation is then build by clicking the “Build Simulation” button which checks for errors in the topology and creates corresponding NS-3 network. If Status bar is green, the user clicks the “Run Simulation” button.

The final step is to view the collected data. If the simulation run without errors, pcap and trace files are saved in the application directory. The user can generate graphs using Rscript. An example script to generate graphs has been given in Appendix A.

4.11 Summary of Chapter 4

This chapter describes the IcnMesh implementation with respect to the design requirements given in Chapter 3. The chapter starts by discussing development tools and the development process used in creation of IcnMesh. IcnMesh was developed through an iterative process where sub-components were developed and tested separately and then combined to form the whole program. The architecture of IcnMesh is then described in detail from the viewpoint of its classes. The role of each class and how it interfaces with other classes is shown. The IcnMesh User Interface is then described.

Since IcnMesh uses NS-3 at its core, the NS-3 simulator API is described together with the ndnSIM API which is used to implement named data networking in IcnMesh. The measuring system of IcnMesh is then described, and it is shown how traffic flow between various nodes in the simulation is captured and analysed using the R statistical and data analysis language.

The chapter then concludes by describing in detail the process of running a simulation on IcnMesh.

CHAPTER FIVE

5 RESULTS AND EVALUATION

The developed IcnMesh simulator was used to evaluate the performance of ICN Content distribution in considered mesh topologies. This chapter starts by giving results of simple simulations that validate the functionality of the Simulator. A simulation to investigate routing in the WMN is then run. The final simulation is of a medium-sized WMN with multiple Content producers and consumers. In each simulation, the gathered data is analysed. The chapter concludes with a discussion on overall performance of information-centric Content distribution in WMNs.

5.1 Validation of Simulator

In order to prove functionality of IcnMesh, simple simulations with predictable results were run and data gathered then observed. The first simulation was of a simple host-centric WMN.

5.1.1 Simple Host-Centric Network

The first simulation conducted on IcnMesh was of a host-centric WMN. In this experiment, the simulation type, number of mesh nodes, server nodes and client nodes were set by the researcher. The number of packets sent and received by server and client nodes were then measured.

5.1.1.1 Objectives

- Verify basic communication between nodes in the WMN.
- Verify fading in the datalink as distance between the nodes increases.
- Verify mesh routing between mesh nodes that are not in direct association.
- Verify data transfer between nodes not participating in the mesh network, that is, the Client and Server nodes.
- Verify logging of network data and that correct log file is produced for each node in the WMN.

If the above objectives were achieved, then the simulator would be confirmed as operating correctly and more advanced simulations could then be carried out.

5.1.1.2 Network Setup

The network topology in figure 5.1 was set-up and simulated on IcnMesh. The network consists of 5 mesh routers with two Client Nodes connected to Node 5 and one server node connected to Node 3. As described in Chapter 4, mesh nodes connect to non-mesh nodes via a CSMA wired interface. Producer0 was automatically assigned the IP address 172.16.5.2 and consumer0 was assigned 172.16.3.2 . The Ethernet interfaces of meshNode5 and meshNode3 were assigned the IP addresses 172.16.5.1 and 172.16.3.1 respectively. The default routes for consumer0 and producer0 were set to their respective meshNodes.

The server node (producer0) was set to run a simple UDP server that was scheduled to send UDP packets to the Client node (consumer0) 15 seconds after the simulation was started. To verify the objectives of the experiment, UDP packets should be captured coming from producer0's network interface, captured on meshNode5's Ethernet interface, forwarded to the mesh network, received by meshNode3 and finally forwarded to consumer0's network interface.

To observe OLSR routing, HELLO messages should be captured on the wireless interfaces of the mesh nodes.

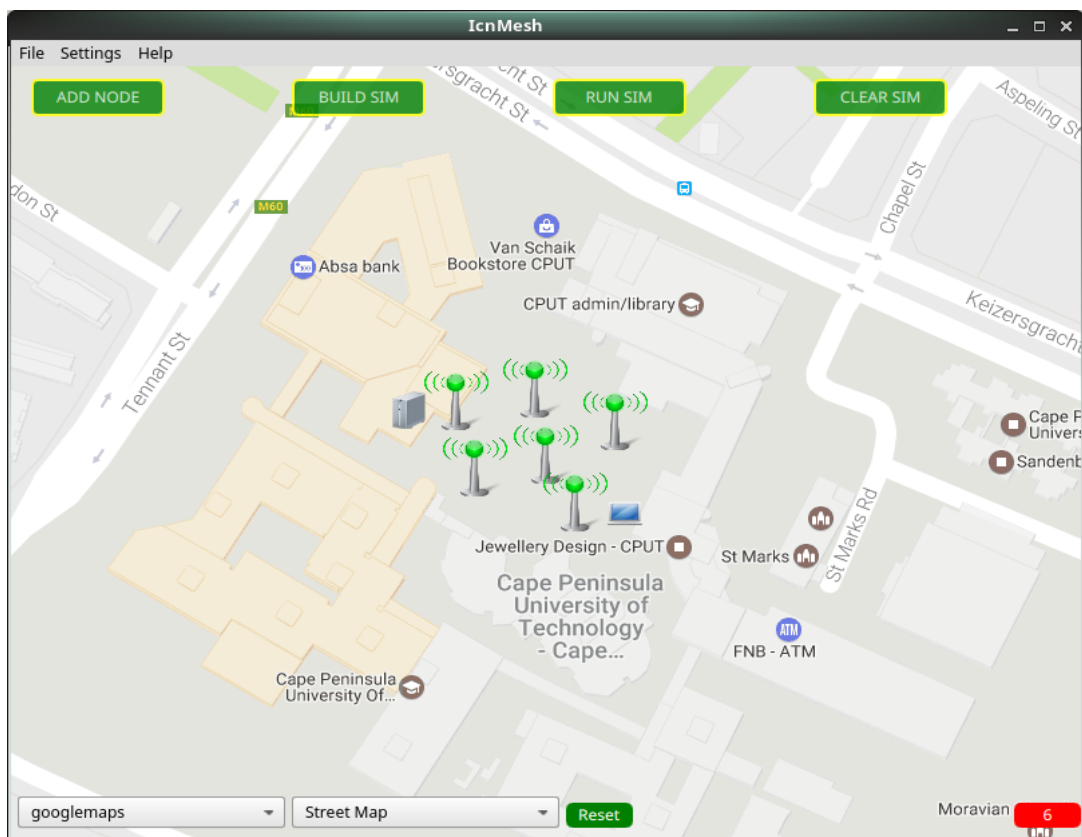


Figure 5.1: Simple host-centric WMN topology

5.1.2 Simulator Configuration

Table 5.1 summarises the simulator configuration for the simulation experiment.

Table 5.1: Simulator configuration for simple host-centric network

Parameter	Value
Number of Mesh Nodes	6
No. of Content Producers /Servers	1
No. of Content Consumers /Servers	1
Radio Interface	IEEE 802.11n
Carrier Frequency	2.4 GHz
Tx power level	20 dBm
Propagation Model	ThreeLog Distance Propagation Loss Model
Channel Fading model	Nakagami Propagation Loss Model
Link Rate	65 Mbps
Medium Access Layer	IEEE 802.11
Network Type	Host-Centric
Mesh Routing Protocol	OLSR
Transport Protocol	UDP
Content Request Frequency	Single request
Simulation Time	30 seconds

5.1.2.1 Results and Discussion

In this section, results of the simulation are presented and discussed according to the performance metrics described in each subsection. The metric for success of this experiment is successful routing of a packet from the server node to the client node. This is determined by the mesh nodes physical, datalink and network layers.

Mesh IP Routing

The simulation experiment used the OLSR protocol for routing of data in the WMN. In this subsection, a pcap file of one of the mesh routers is viewed and analysed. Figure 5.2 shows packets captured from the wireless network interface of meshNode0. In the figure, OLSR protocol message packets from mesh routers are observed being broadcasted to all mesh nodes in the network. These are neighbour discovery packets used by the OLSR protocol to announce mesh nodes reachable. This is consistent with the OLSR protocol as described in chapter 2. In order for non-OLSR

nodes to participate in the WMN, OLSR defines Host-Network Association (HNA) messages which mesh nodes broadcast network prefixes of non-OLSR networks that they are connected to. At 15 seconds, meshNode5 broadcasts an ARP message to identify meshNode3. MeshNode3 acknowledges and meshNode5.

```

tgone@localhost:~/Dev/Masters/src/icnMesh/data/host-centric
File Edit View Search Terminal Help
12.385524 IP 10.1.1.6.698 > 10.1.1.255.698: OLSRv4, seq 0x0007, length 60
12.413011 IP 10.1.1.5.698 > 10.1.1.255.698: OLSRv4, seq 0x0006, length 60
14.030547 IP 10.1.1.6.698 > 10.1.1.255.698: OLSRv4, seq 0x0008, length 60
14.112091 IP 10.1.1.2.698 > 10.1.1.255.698: OLSRv4, seq 0x0007, length 60
14.234691 IP 10.1.1.4.698 > 10.1.1.255.698: OLSRv4, seq 0x0008, length 60
14.283567 IP 10.1.1.5.698 > 10.1.1.255.698: OLSRv4, seq 0x0007, length 60
14.431124 IP 10.1.1.3.698 > 10.1.1.255.698: OLSRv4, seq 0x0007, length 60
14.446067 IP 10.1.1.1.698 > 10.1.1.255.698: OLSRv4, seq 0x0007, length 60
15.017584 ARP, Request who-has 10.1.1.4 (ff:ff:ff:ff:ff:ff) tell 10.1.1.6, length 32
15.017688 ARP, Reply 10.1.1.4 is-at 00:00:00:00:00:04, length 32
15.018002 Acknowledgment RA:00:00:00:00:00:04
15.018346 IP 172.16.5.2.49153 > 172.16.3.2.80: UDP, length 1000
15.018660 Acknowledgment RA:00:00:00:00:00:06
15.324764 IP 10.1.1.4.698 > 10.1.1.255.698: OLSRv4, seq 0x0009, length 24
15.465377 IP 10.1.1.6.698 > 10.1.1.255.698: OLSRv4, seq 0x0009, length 24
16.003847 IP 172.16.5.2.49153 > 172.16.3.2.80: UDP, length 1000
16.004971 IP 172.16.5.2.49153 > 172.16.3.2.80: UDP, length 1000
16.005285 Acknowledgment RA:00:00:00:00:00:06
16.039410 IP 10.1.1.2.698 > 10.1.1.255.698: OLSRv4, seq 0x0008, length 60
16.068407 IP 10.1.1.5.698 > 10.1.1.255.698: OLSRv4, seq 0x0008, length 60
16.077849 IP 10.1.1.3.698 > 10.1.1.255.698: OLSRv4, seq 0x0008, length 60
16.271965 IP 10.1.1.1.698 > 10.1.1.255.698: OLSRv4, seq 0x0008, length 60
16.358021 IP 10.1.1.4.698 > 10.1.1.255.698: OLSRv4, seq 0x000a, length 60

```

Figure 5.2: Pcap on wireless interface of meshNode0

```

tgone@localhost:~/Dev/Masters/src/icnMesh
File Edit View Search Terminal Help
[tgone@localhost icnMesh]$ tcpdump -nn -tt -r csmaInterface-producer0-0.pcap
reading from file csmaInterface-producer0-0.pcap, link-type EN10MB (Ethernet)
15.026769 ARP, Request who-has 172.16.3.2 (ff:ff:ff:ff:ff:ff) tell 172.16.3.1, length 50
15.026769 ARP, Reply 172.16.3.2 is-at 00:00:00:00:00:08, length 50
15.032547 IP 172.16.5.2.49153 > 172.16.3.2.80: UDP, length 1000
16.007723 IP 172.16.5.2.49153 > 172.16.3.2.80: UDP, length 1000
17.007723 IP 172.16.5.2.49153 > 172.16.3.2.80: UDP, length 1000
18.007723 IP 172.16.5.2.49153 > 172.16.3.2.80: UDP, length 1000
19.007723 IP 172.16.5.2.49153 > 172.16.3.2.80: UDP, length 1000
20.007723 IP 172.16.5.2.49153 > 172.16.3.2.80: UDP, length 1000
21.007723 IP 172.16.5.2.49153 > 172.16.3.2.80: UDP, length 1000
tcpdump: pcap_loop: truncated dump file; tried to read 1046 captured bytes, only got 1042
[tgone@localhost icnMesh]$

```

Figure 5.3: Pcap on Ethernet interface of producer0


```
tgone@localhost:~/Dev/Masters/src/icnMesh
File Edit View Search Terminal Help
[tgone@localhost icnMesh]$ tcpdump -nn -tt -r csmaInterface-consumer0-0.pcap
reading from file csmaInterface-consumer0-0.pcap, link-type EN10MB (Ethernet)
15.005000 ARP, Request who-has 172.16.5.1 (ff:ff:ff:ff:ff:ff) tell 172.16.5.2, l
length 50
15.009205 ARP, Reply 172.16.5.1 is-at 00:00:00:00:00:09, length 50
15.009205 IP 172.16.5.2.49153 > 172.16.3.2.80: UDP, length 1000
16.000000 IP 172.16.5.2.49153 > 172.16.3.2.80: UDP, length 1000
17.000000 IP 172.16.5.2.49153 > 172.16.3.2.80: UDP, length 1000
18.000000 IP 172.16.5.2.49153 > 172.16.3.2.80: UDP, length 1000
19.000000 IP 172.16.5.2.49153 > 172.16.3.2.80: UDP, length 1000
20.000000 IP 172.16.5.2.49153 > 172.16.3.2.80: UDP, length 1000
21.000000 IP 172.16.5.2.49153 > 172.16.3.2.80: UDP, length 1000
tcpdump: pcap_loop: truncated dump file; tried to read 1046 captured bytes, only
got 1042
[tgone@localhost icnMesh]$
```

Figure 5.4: Pcap on Ethernet interface of consumer0

Figure 5.3 shows the transmissions on producer0's interface and figure 5.4 show the these transmissions received at consumer0. In this experiment, the simplest simulation on IcnMesh simulator was tested. From the results above, the following can be deduced.

- Basic communication between nodes in the WMN takes place. This verifies that the physical and MAC interfaces of the meshNodes function correctly.
- The mesh nodes connect in a mesh configuration.
- The network layer routing protocol functions correctly and data transfer between mesh nodes is possible. OLSR protocol messages can be seen in figure 5.2
- The OLSR HNA protocol works correctly. It is possible to transfer data between nodes not participating in the mesh network.
- Pcap data logs are produced at every network interface of every node in the simulated network.

In the next sections, ICN experiments are carried out on IcnMesh.

5.1.3 Simple Information-Centric Network

Having verified basic communication, the next step was to simulate an ICN network. The main objective of this experiment was to confirm simple name-based routing.

5.1.3.1 Objectives

- Verify basic ICN communication between producer and consumer nodes
- Verify correct routing of Interest and Data packets within the WMN.
- Verify data transfer between producer and consumer nodes.
- Measure the timeout interval of an interest packet sent without response.
- Verify logging of ICN network data in trace files.
- Set a benchmark for ICN experiments in subsequent experiments.

5.1.3.2 Network Setup

The network comprises of three mesh nodes, meshNode0 and meshNode1, with a producer node (producer0) connected to one of the mesh nodes and two consumer nodes (consumer0 and Consumer1) connected to each to the other mesh nodes.

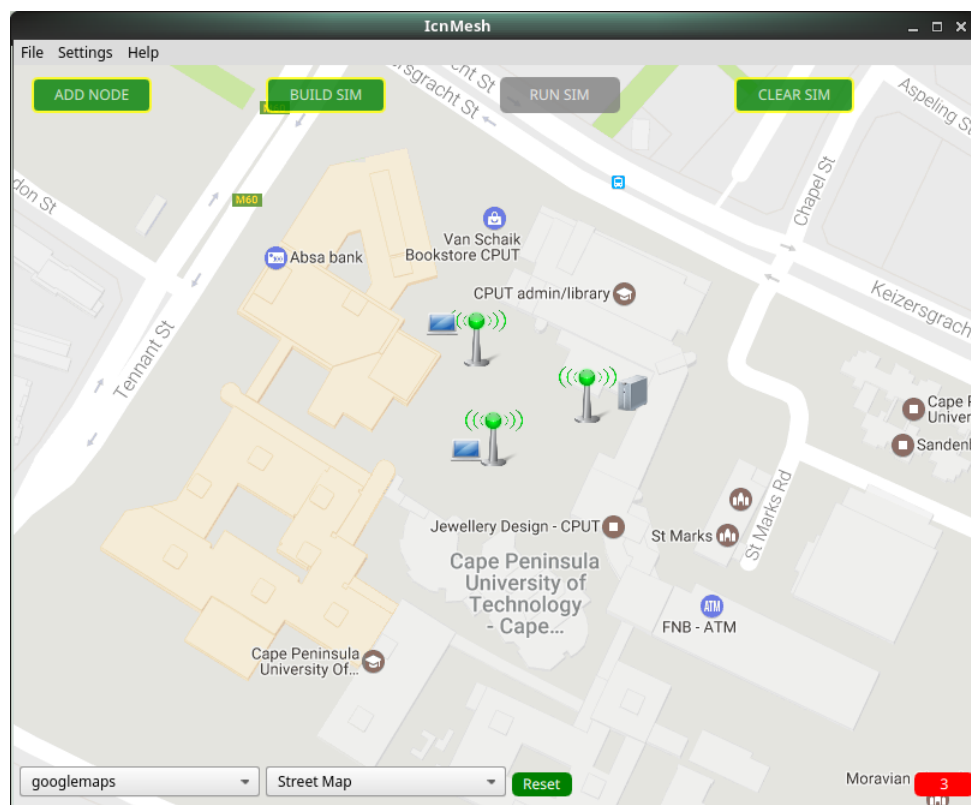


Figure 5.5: Simple ICN network topology simulated on IcnMesh

The producer0 was configured to publish Content with the prefix “/Content0” and of 1000 Kb size. Consumer0 was configured to fetch content with prefix “/Content0”. Consumer1 was configured to fetch “/Content1” which was not published on the network.

5.1.3.3 Simulator Configuration

Table 5.2 summarises the configuration of IcnMesh for the simulation experiment.

Table 5.2: IcnMesh configuration for simple ICN network.

Parameter	Value
Number of Mesh Nodes	6
No. of Content Producers /Servers	1
No. of Content Consumers /Servers	2
Radio Interface	IEEE 802.11n
Carrier Frequency	2.4 GHz
Tx power level	20 dBm
Propagation Model	ThreeLog Distance Propagation Loss Model + Nakagami Propagation Loss Model
Link Rate	65 Mbps
Medium Access Layer	ad hoc
Network Type	Information-Centric
Mesh Routing Protocol	Multicast
Transport Protocol	N/A
Content Request Pattern	Zipf-Mandelbrot Distribution
Content-Prefix assignment	Manual
Simulation Time	30 seconds

5.1.3.4 Results

The metric for success in this experiment is successful routing of an NDOs from the producer node to the consumer node. This is measured by observing the number of interest packets sent by consumer0 against the number of satisfied interests on the same node. Data sent from producer0 should also be observed on consumer0. The delay between the time a data packet is sent from producer0 and received at consumer0 is also noted. Tracing is used to collect data and R-script are used to generate graphs of interest and data packets on all nodes in the network.

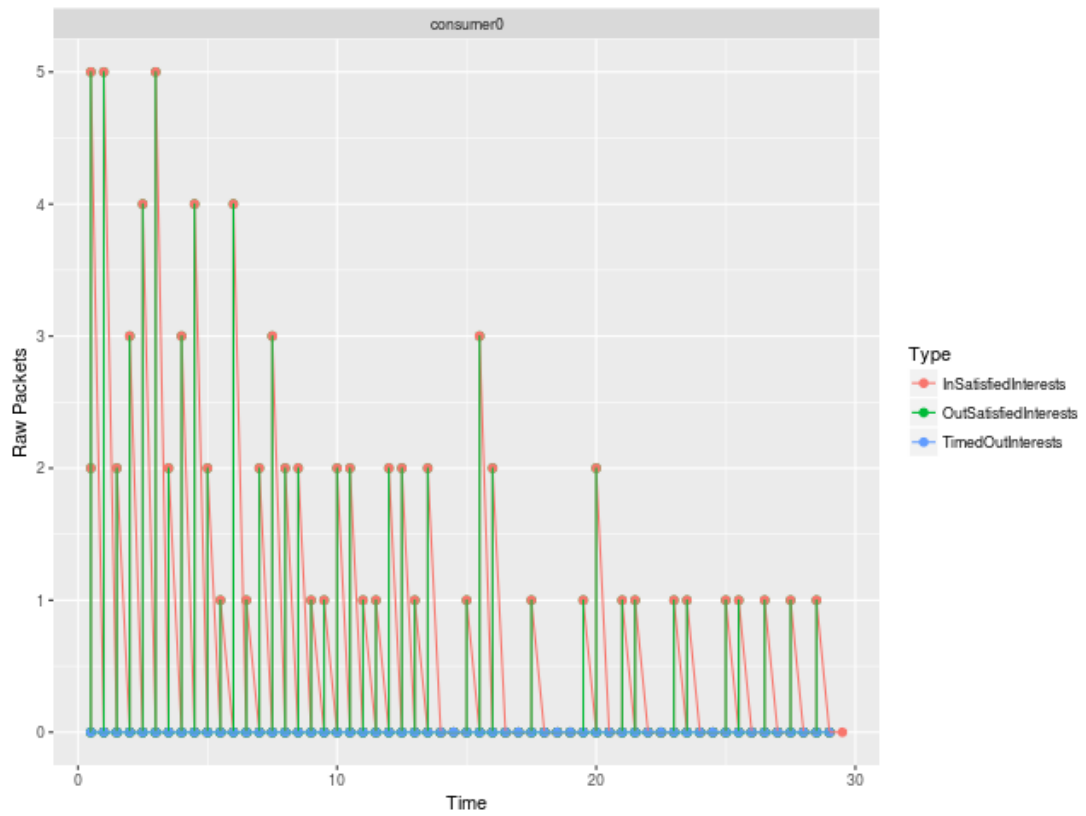


Figure 5.6: Graph of interest packets logged by consumer0

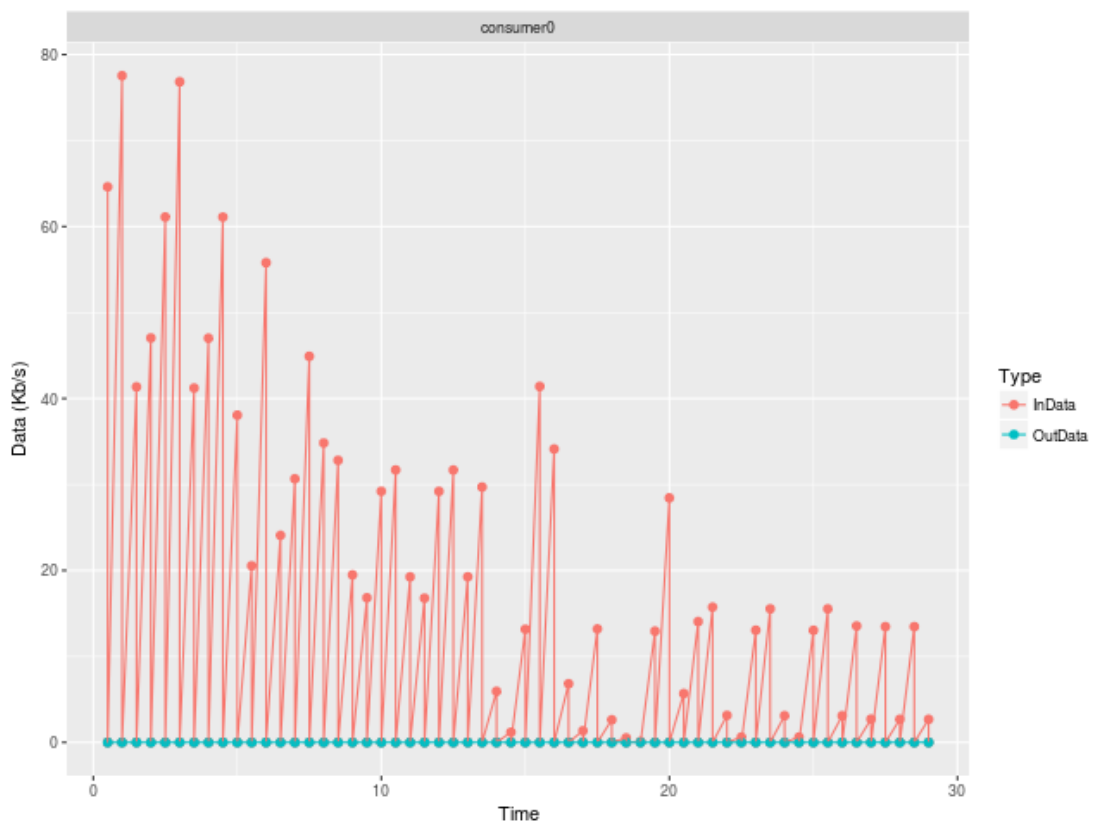


Figure 5.7: Graph of data traffic received by consumer0

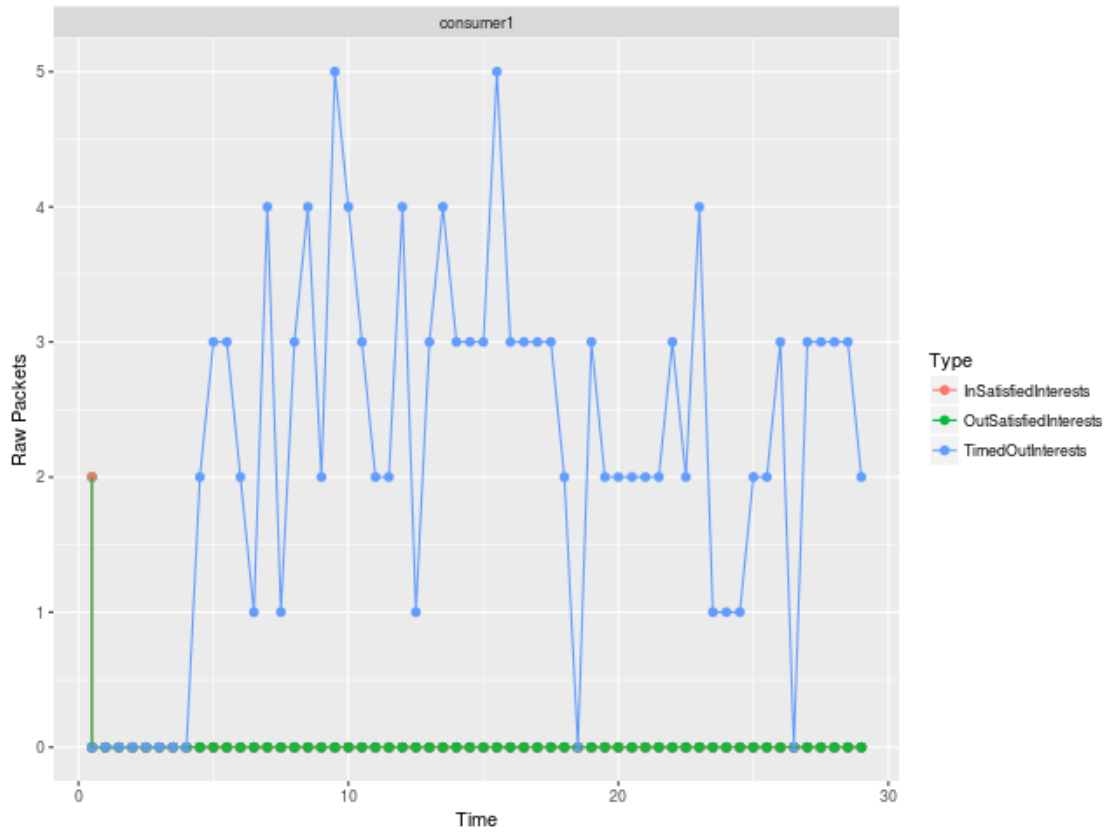


Figure 5.8: Graph of Interest packets logged by consumer1

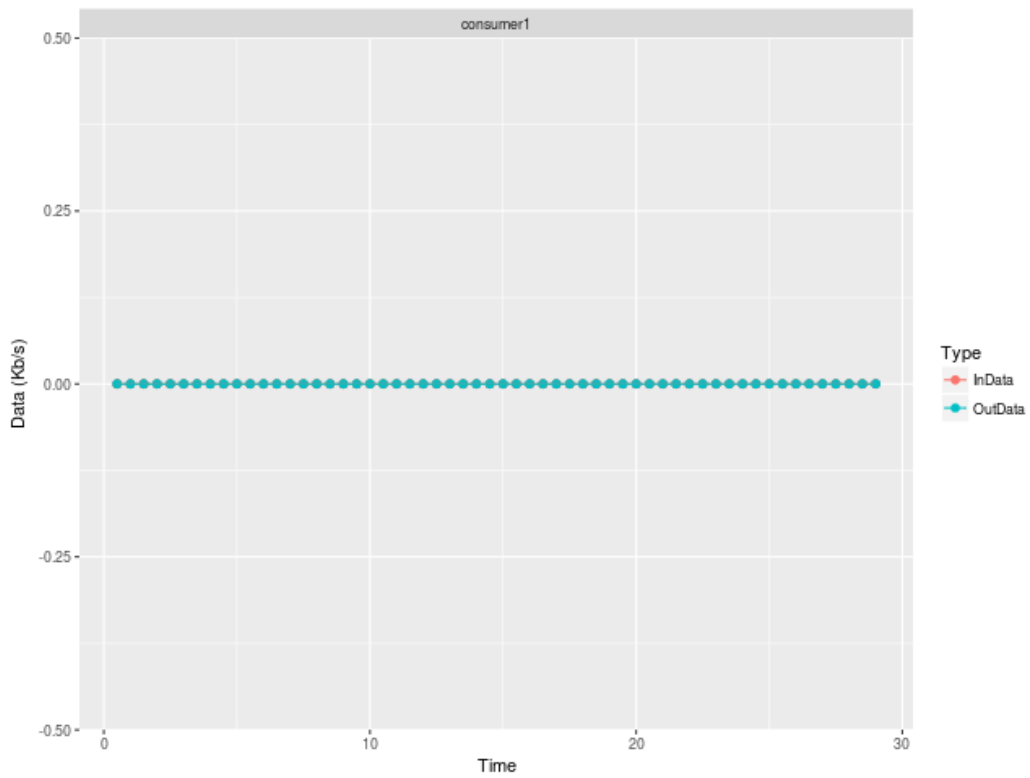


Figure 5.9: Graph of data received by consumer1

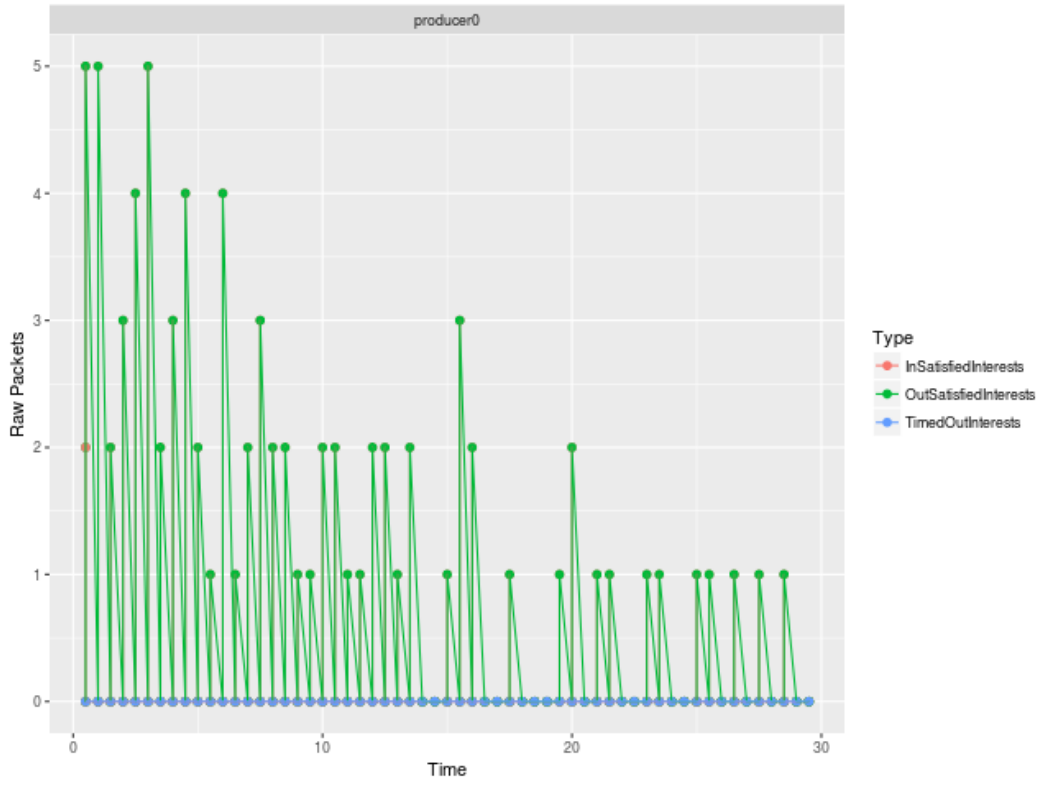


Figure 5.10: Graph of Interest packets logged by producer0

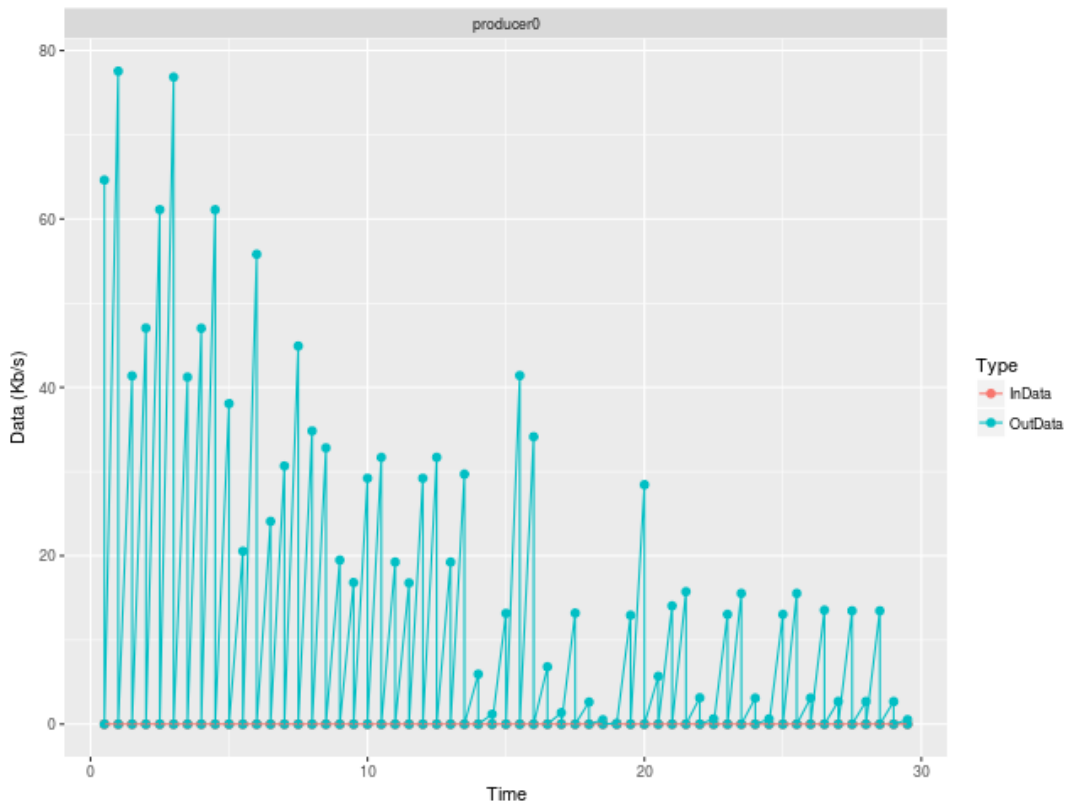


Figure 5.11: Graph of data traffic sent by producer0

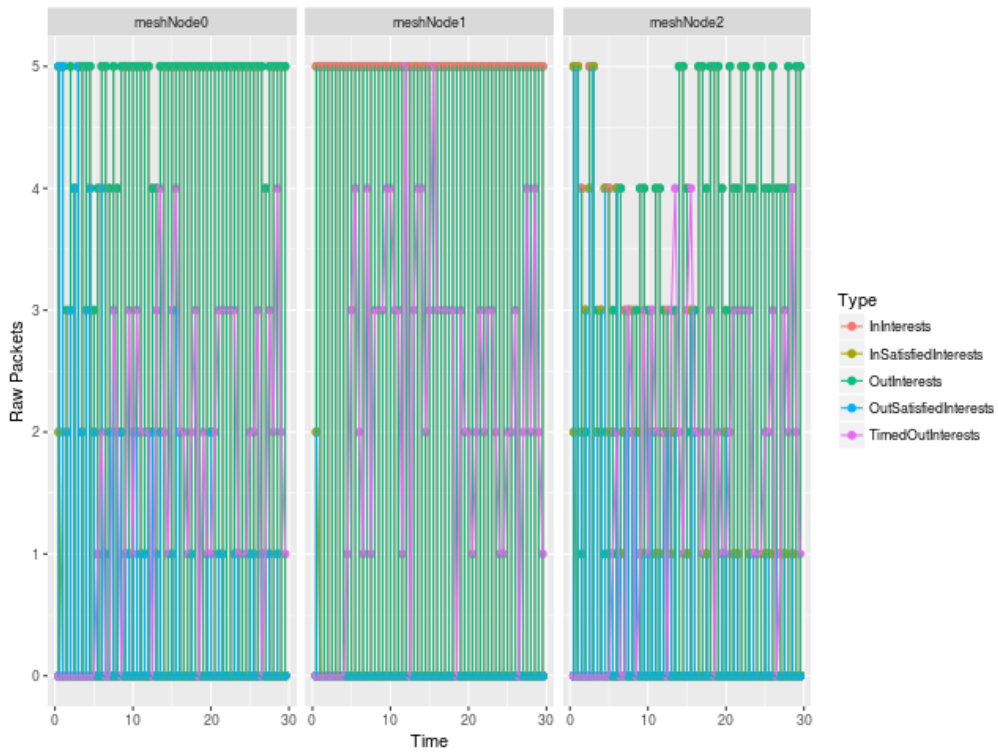


Figure 5.12: Graph of interest traffic logged by mesh nodes

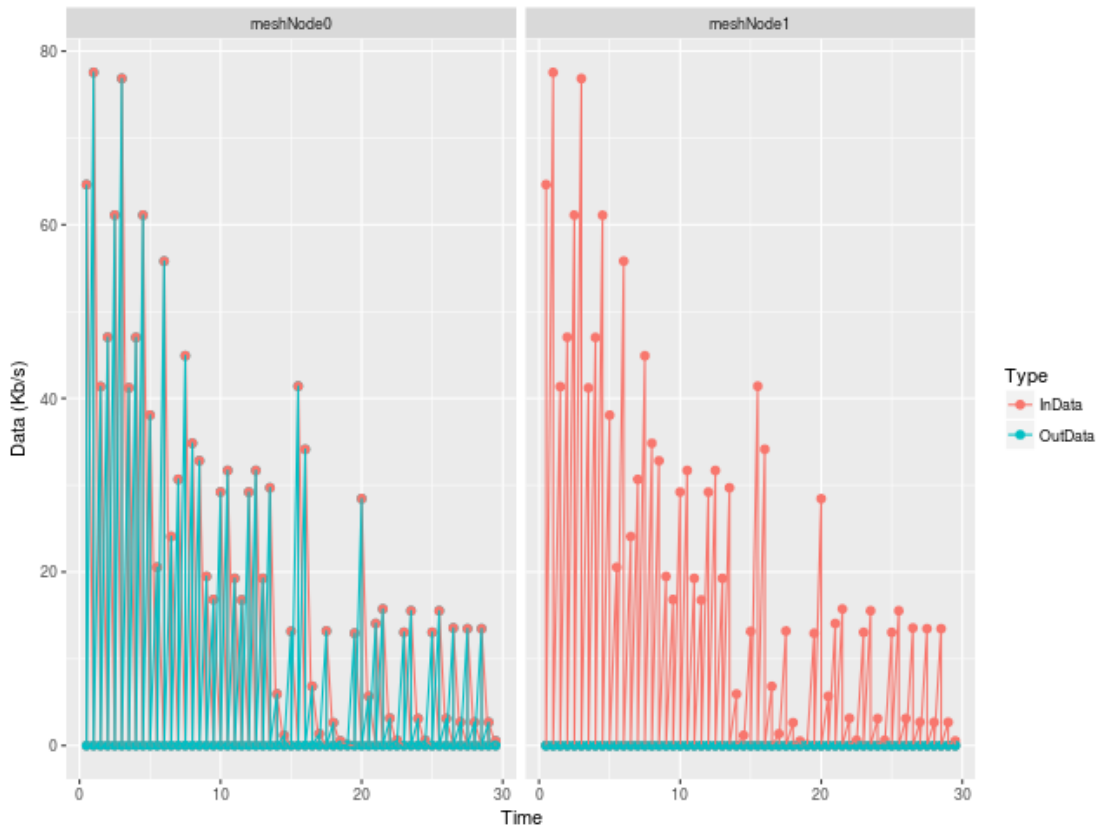


Figure 5.13: Graph of data traffic logged by mesh nodes

From figure 5.6, it can be seen that consumer0 had all its interests satisfied and data was transferred successfully. The graph of interests sent by consumer0 shows that all interests were satisfied and none timed out.

The data-out graph of producer0 (figure 5.11) perfectly matches the data-in graph of consumer0 (figure 5.7). It can therefore be deduced that data from all interests sent was received by consumer0. It should be noted that there is no observable delay in the transfer of data packets from producer0 to consumer0.

On the other hand, consumer1 had all its interest packets timed out as in figure 5.8. From consumer1 interest graph, it can be seen that it takes about 3 seconds for the interest packets sent from a consumer node to time-out. The data transfer graph in figure 5.9 also shows that no data packets were received at this node. From these results, it can therefore be seen that named-data routing works correctly on the IcnMesh simulator.

In the next experiment, routing between mesh nodes is investigated.

5.2 Routing in Information-Centric Wireless Mesh Network

In this section, 2 experiments are carried out to investigate routing in the Information-Centric WMN by deliberately introducing a hidden node. The Hidden-node problem is a well-known phenomenon in IEEE 802.11 wireless networks (Ng, P.C., *et al.*, 2005) , and by extension, in IEEE 802.11 WMNs. This simulation deliberately separates the mesh node connected to the consumer node from the mesh node connected to the producer node such that there is no direct association between the two mesh nodes.

5.2.1 Objectives

- Observe characteristics of different MAC protocols on WMN hidden node problem
- Identify best MAC protocol for use with IcnMesh

5.2.2 Network Setup

A simple topology consisting of 5 mesh nodes, one producer node and one client node was set-up as in figure 5.14. For this experiment, the transmission power of mesh nodes was set to the lowest level possible so that there would be no direct connectivity between meshNode0 and meshNode4 and therefore intermediate meshNodes would facilitate routing between them.

In the first case, the standard IEEE 802.11 MAC is used. In the second case, IEEE 802.11s MAC, which is the basis for IEEE 802.11s WMNs, is used. To facilitate the experiment, the transmission power in each mesh node is set to the lowest level possible so that the transmission range of each mesh node is scaled down. The simulated network is illustrated in figure 5.14. The network topology is similar to the one used to investigate the 802.11 hidden node problem by Ng, P.C., *et al.*, (2005).

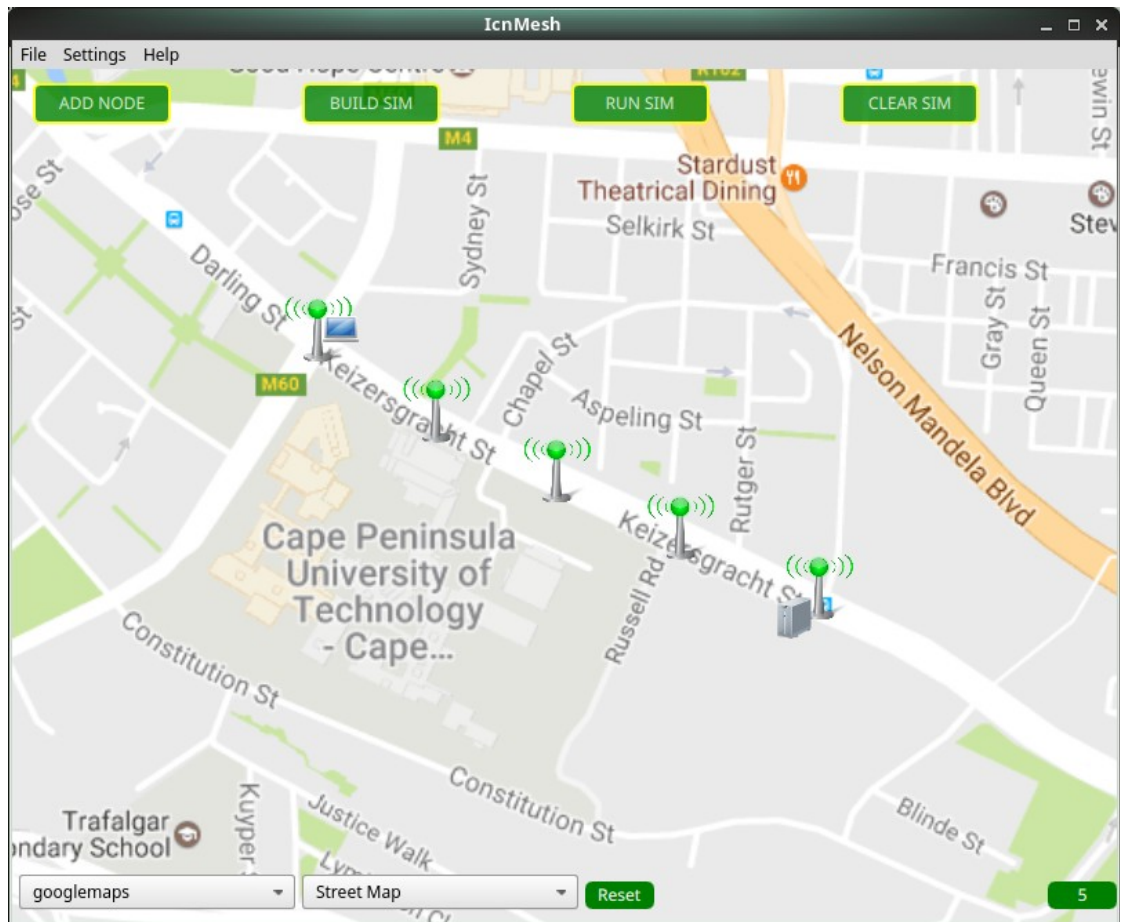


Figure 5.14: IcnMesh network topology for hidden node routing

5.2.3 Routing with 802.11 MAC

The key configuration parameter in this set-up was the use of the standard IEEE 802.11 MAC.

5.2.3.1 Simulation Configuration

Table 5.3 summarizes the configuration of the network under simulation.

Table 5.3: IcnMesh configuration with IEEE 802.11 MAC

Parameter	Value
Number of Mesh Nodes	5
No. of Content Producers /Servers	1
No. of Content Consumers /Servers	1
Radio Interface	IEEE 802.11n
Carrier Frequency	2.4 GHz

Tx power level	1 dBm
Propagation Model	ThreeLog Distance Propagation Loss Model + Nakagami Propagation Loss Model
Wireless Link Rate	65 Mbps
Medium Access Layer	IEEE 802.11 (ad hoc mode)
Network Type	Information-Centric
Mesh Routing Protocol	IEEE 802.11s
Transport Protocol	N/A
Content Request Frequency	Constant rate, 10 interests per second
Simulation Time	30 seconds

5.2.3.2 Results and Discussion

Figure 5.15 shows content requests from consumer0 against time. It shows that all

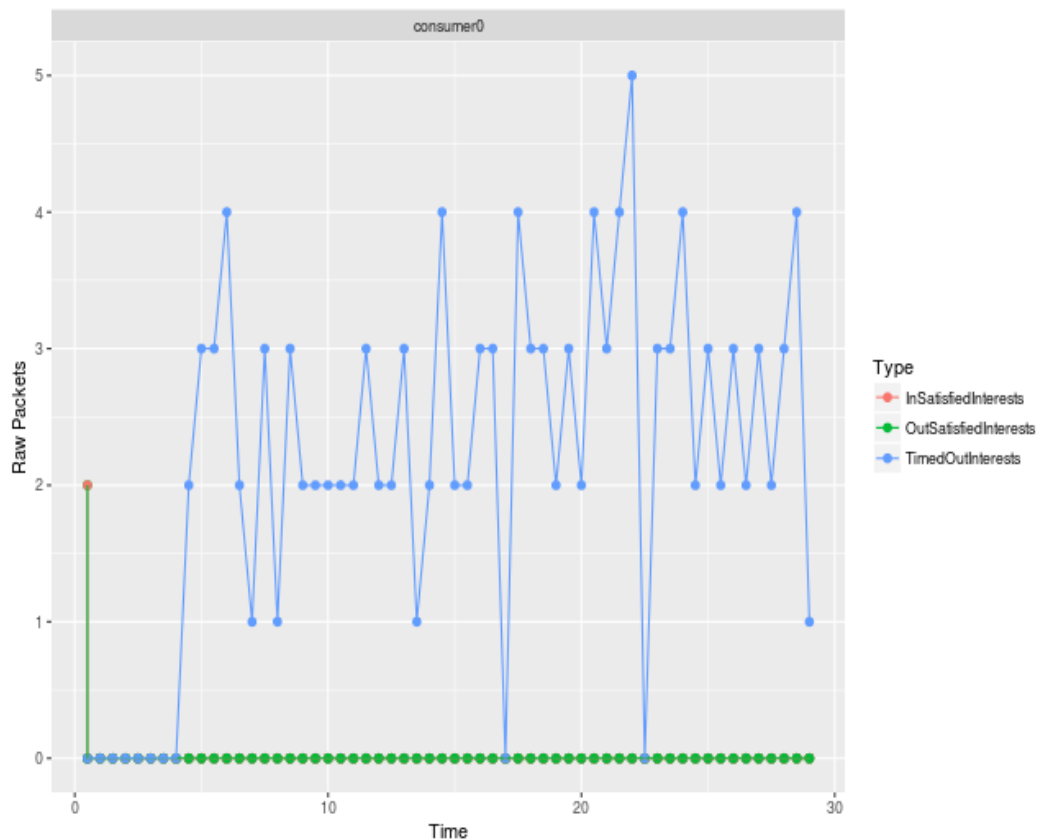


Figure 5.15: Graph of interest packets logged by consumer0 content requests from consumer0 timed out and none was satisfied and consequently, no data from the producer was received.

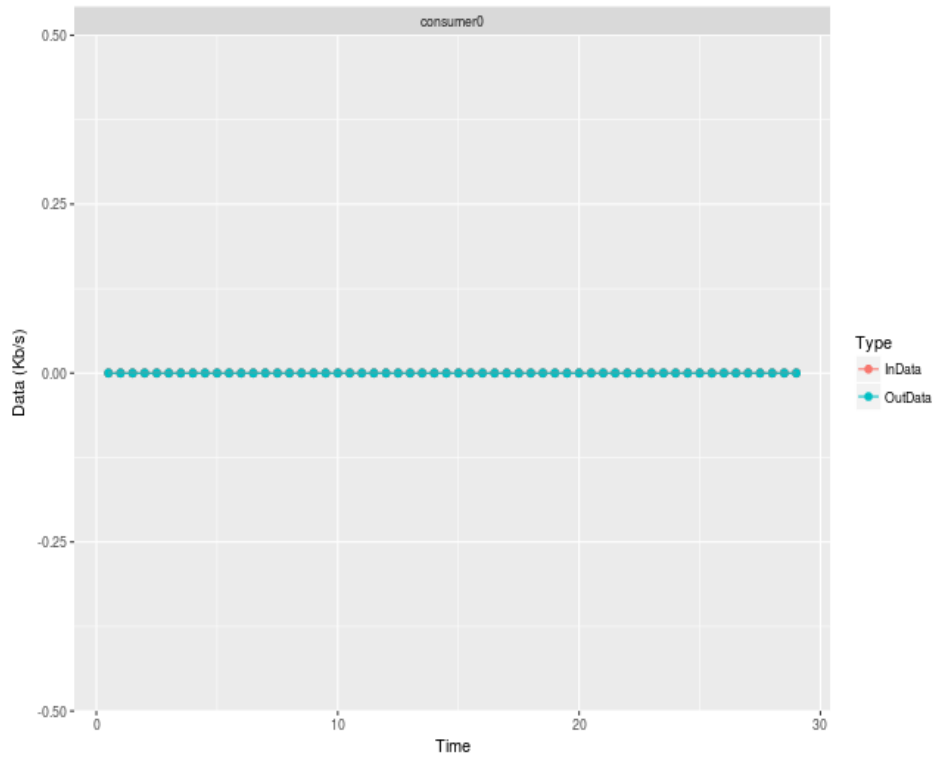


Figure 5.16: Graph of data transfer at consumer0

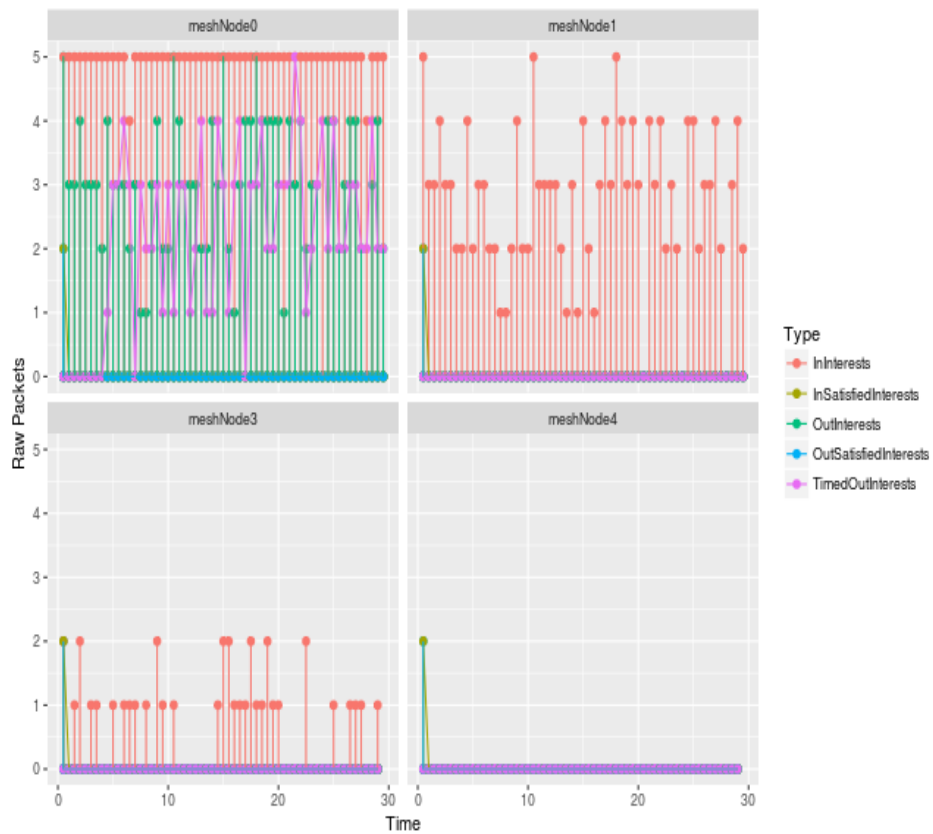


Figure 5.17: Graph of interest packets logged by meshNodes

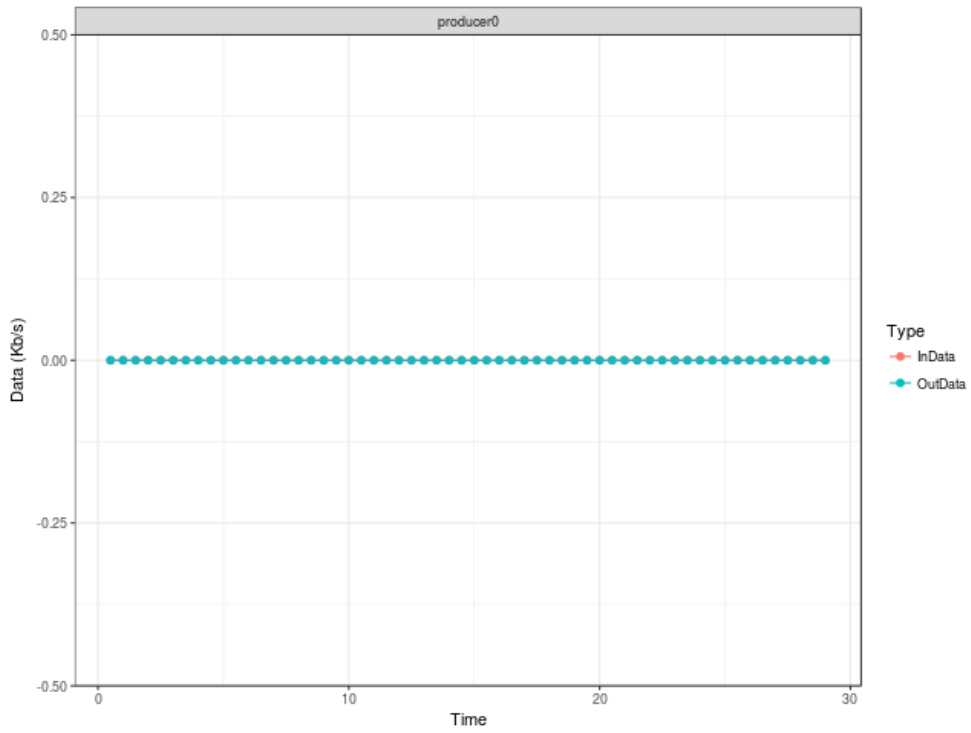


Figure 5.18: Graph of data transfer at producer0

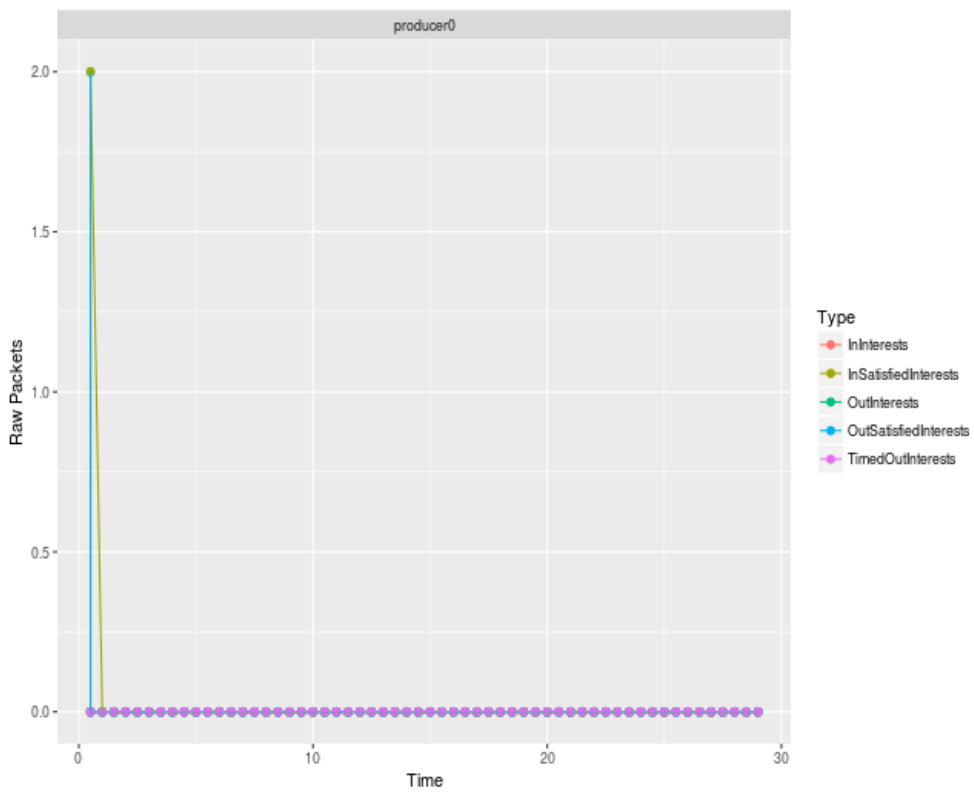


Figure 5.19: Graph of Interest packets logged by producer0

In this experiment, the metric for success was successful data transfer between producer0 and consumer0. It would indicate that the MAC protocol in use eliminates the hidden node in the WMN.

From figures 5.18 and 5.16, it can be seen that no data transfer took place between the producer and consumer nodes. It can be seen that producer0 did not receive any interest packets.

From the mesh node interest graphs (figure 5.17), it can be seen that meshNode0, to which consumer0 is connected, receives interest packets from consumer0 and broadcasts them on the network. MeshNode1 receives a significant portion of these interest packets. MeshNode3 which receives a weak signal from meshNode0 receives some packets but it can be noted that they are considerably fewer than packets received at meshNode1. MeshNode4 does not receive any interest packets.

What can be deduced from the results above is that meshNode4 is effectively hidden from meshNode0. Data is not forwarded to the hidden node by the other nodes in the WMN and therefore producer0 does not receive any interest packets from consumer0. This shows that a routing algorithm that eliminates the hidden node problem should be employed in the WMN.

5.2.4 Routing with Modified 802.11s MAC

The network is set up with similar parameters as the previous case. The significant configuration change in this setup is that the IEEE 802.11s MAC is used in place of the standard IEEE 802.11 MAC used in the previous section. IEEE 802.11s performs routing at the MAC layer unlike mesh routing protocols which perform routing at the network layer. It was hoped that by utilizing a layer2 routing protocol, the hidden node problem could be eliminated.

It should be noted that the ndnSim stack modifies layer2 in addition to higher layers therefore it was also important to note what effect this had on the performance of IEEE 802.11s.

5.2.4.1 Simulation Configuration

Table 5.4 summarizes the configuration of the network under simulation.

Table 5.4: IcnMesh configuration with IEEE 802.11s MAC Layer

Parameter	Value
Number of Mesh Nodes	5
No. of Content Producers /Servers	1
No. of Content Consumers /Servers	1
Radio Interface	IEEE 802.11n
Carrier Frequency	2.4 GHz
Tx power level	1 dBm
Propagation Model	ThreeLog Distance Propagation Loss Model + Nakagami Propagation Loss Model
Link Rate	65 Mbps
Medium Access Layer	IEEE 802.11s
Network Type	Information-Centric
Mesh Routing Protocol	IEEE 802.11s
Transport Protocol	N/A
Content Request Pattern	Zipf-Mandelbrot distribution
Simulation Time	30 seconds

5.2.4.2 Results and Discussion

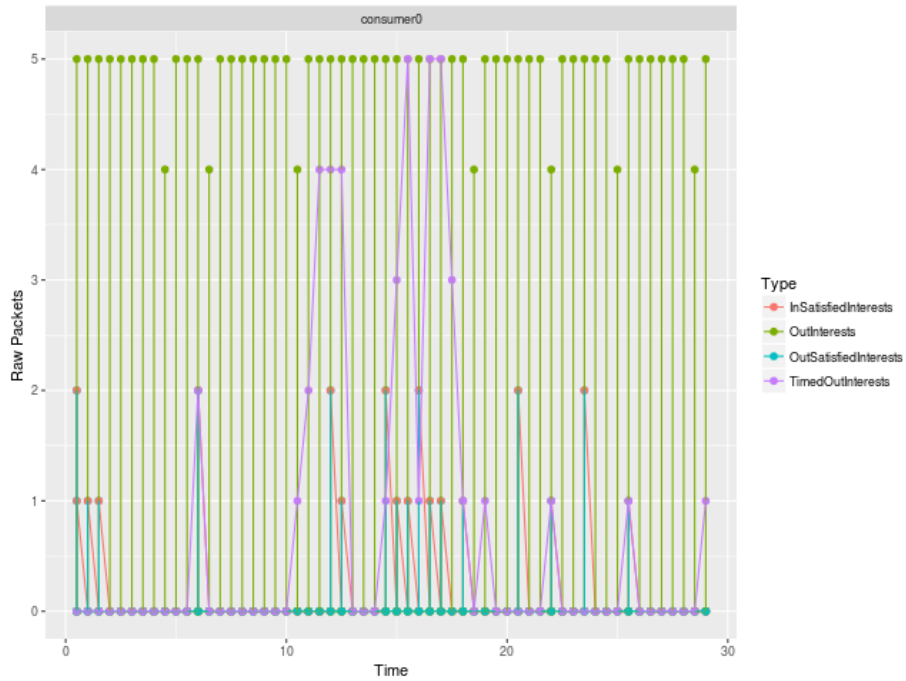


Figure 5.20: Graph of Interest packets logged by consumer0

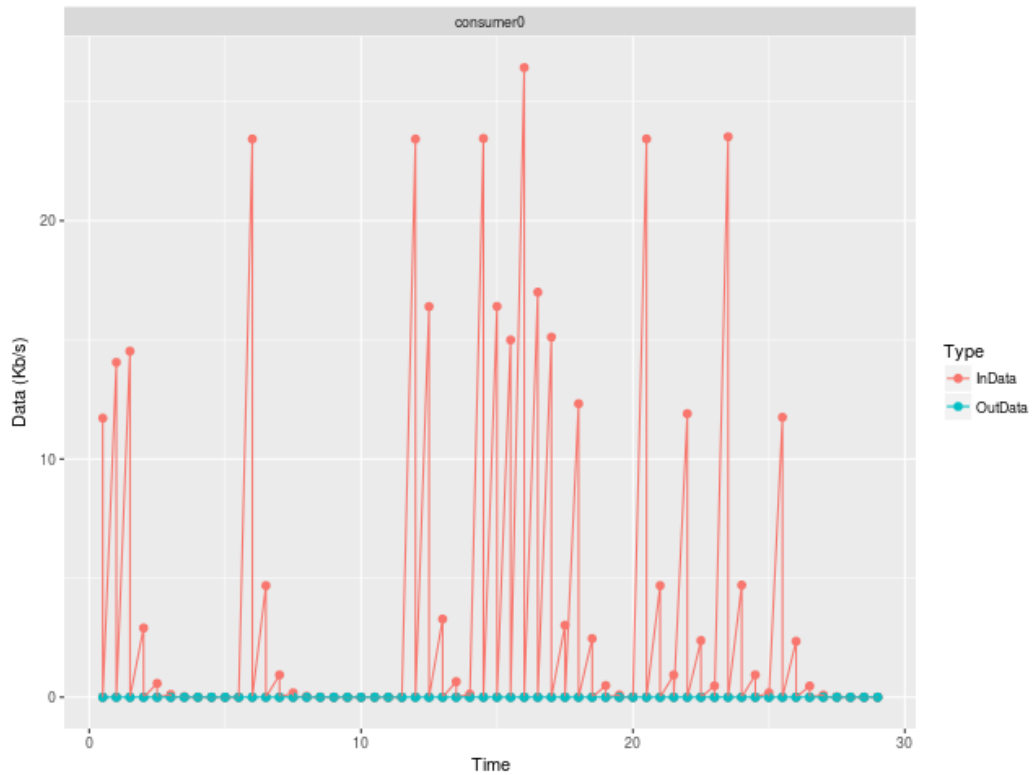


Figure 5.21: Graph of data packets transferred to consumer0

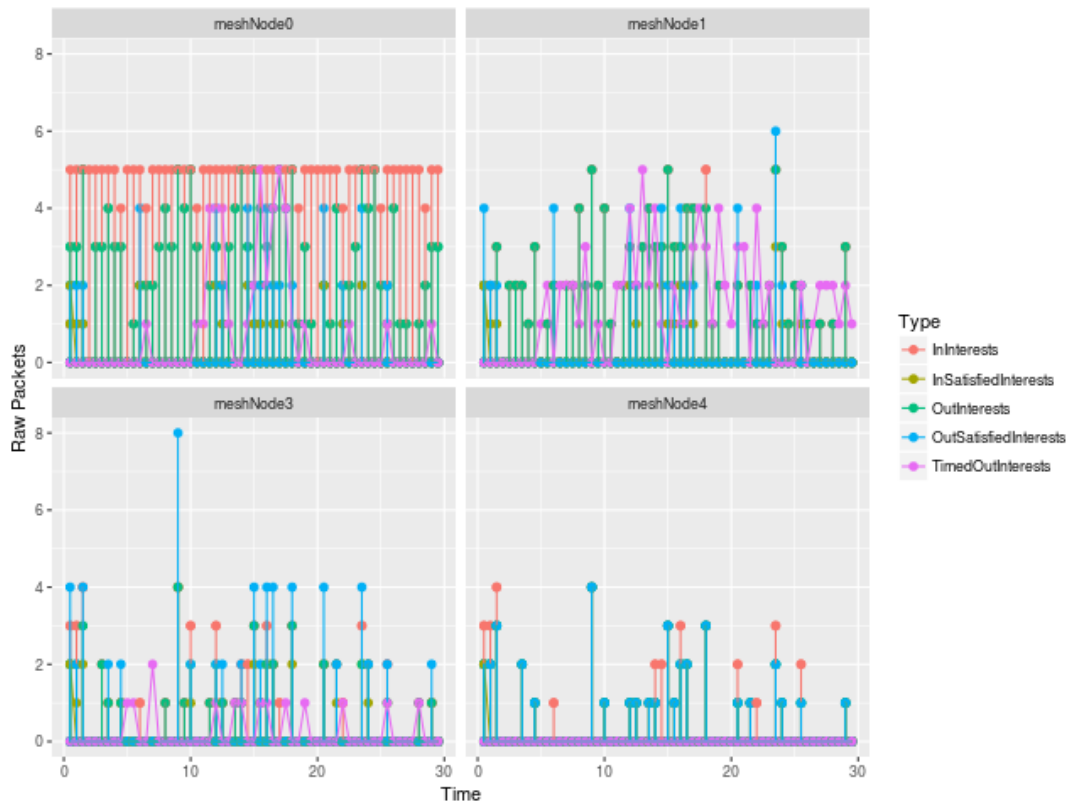


Figure 5.22: Graph of interest packets logged by mesh nodes 0,1,3 and 4

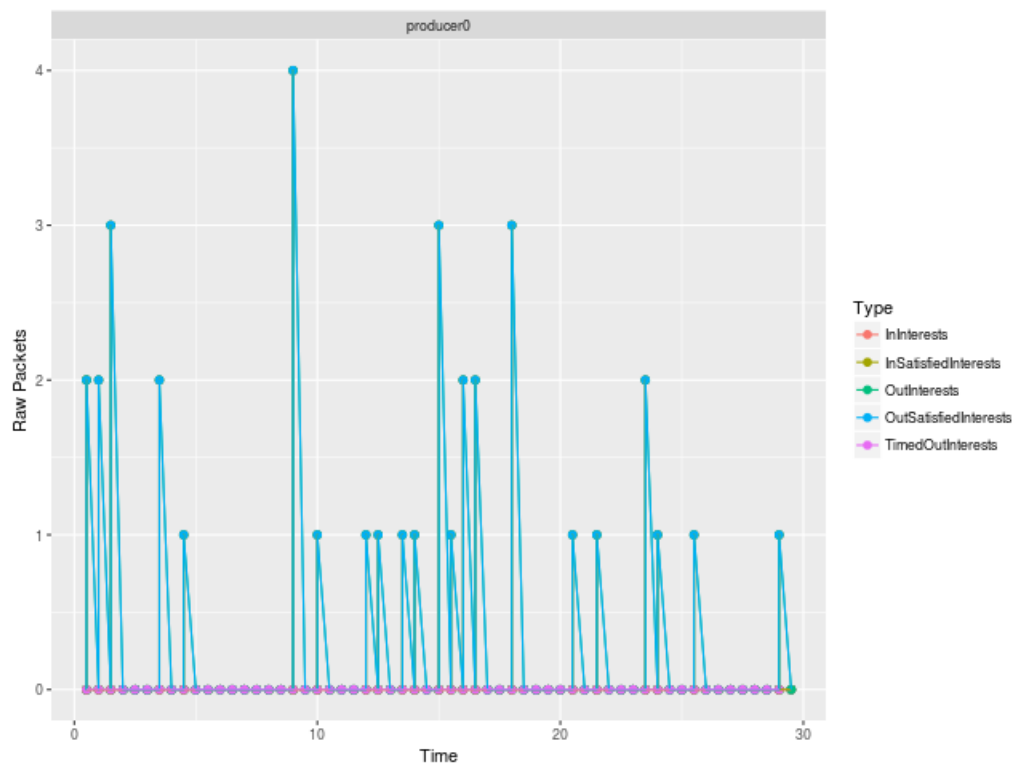


Figure 5.23: Graph of interest packets logged by producer0

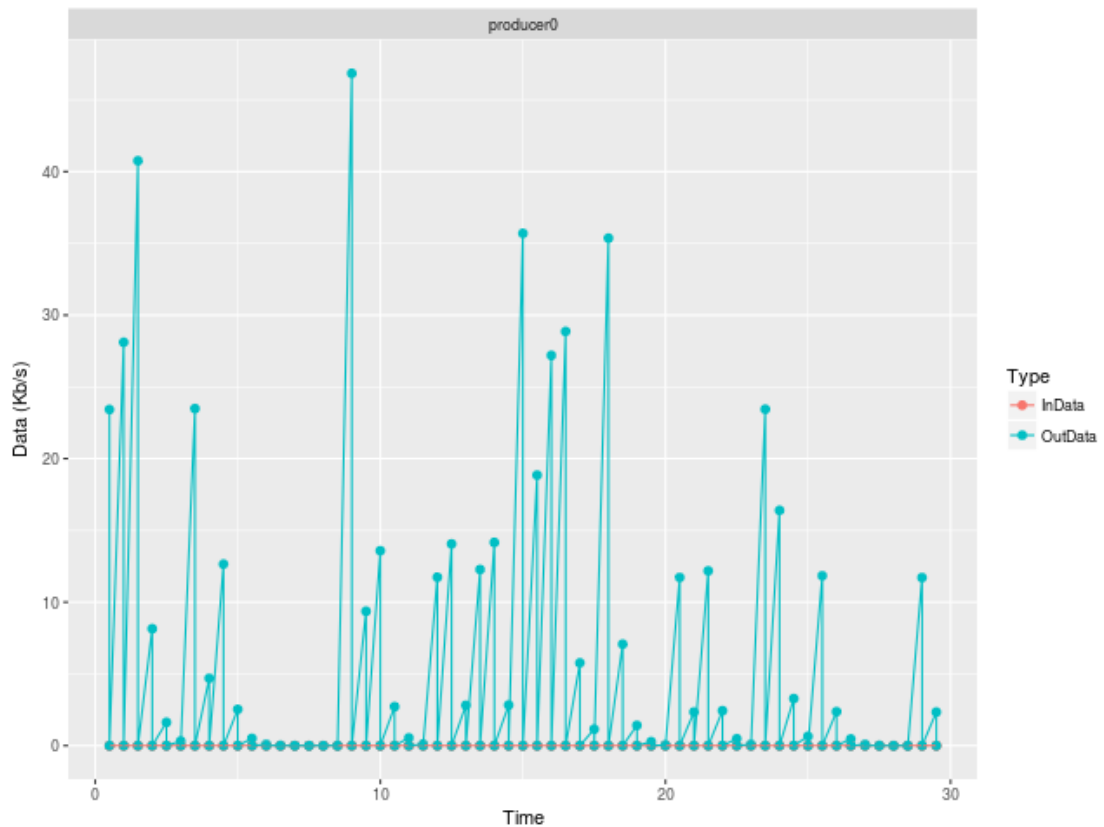


Figure 5.24: Graph of data transferred by producer0

The metric for success was successful data transfer between producer0 and consumer0 which would indicate that the MAC protocol in use eliminates the hidden node in the WMN. From the results above, the following can be inferred. With the IEEE 802.11s MAC, it was possible to transfer traffic between the producer and consumer nodes. However, it can be seen that several interest packets sent by consumer0 timed out. All interest packets that did not time out were fulfilled by producer0.

It should also be noted from the graphs (figures 5.21 and 5.24) that all the data transferred from producer0 to consumer0 is successful and no packets are dropped along the path to consumer0. However, there is a delay of between 2 and 3 seconds in data sent by producer0 to consumer0. This delay between transfers satisfactorily explains the drops in interest packets sent by consumer0. Data packets do not time out hence the reason why all data sent is received by consumer0. Even though the hop counts between the consumer and producer nodes have increased, the delay is too high compared to the nearly instantaneous transfers observed in section 5.1.3 above.

It can therefore be deduced that the IEEE 802.11s protocol, though successful in eliminating the hidden node problem, results in a high delay in data transferred between mesh nodes. This causes timing out of interest packets and the not all interests are satisfied. A suitable MAC protocol for use in ICN mesh network should therefore be developed.

In the next section, a medium-sized WMN is simulated. The standard IEEE 802.11 MAC is used.

5.3 Content Distribution to Multiple Consumers with Single Producer

In this experiment, a WMN with a single content producer and multiple consumers was simulated.

5.3.1 Objectives

The main objective of this experiment was to observe the effects on the server load in a WMN with multiple clients and one server. The distances between the mesh clients and the producer node are kept almost equal in order to eliminate any bias brought about by poor signal reception.

5.3.2 Network Setup

The network comprises of nine mesh nodes arranged in a square pattern with a producer node connected to the central mesh node and eight consumer nodes connected to the other mesh nodes as shown in figure 5.25. The client nodes were configured to request Content at a constant rate of 10 requests every second.

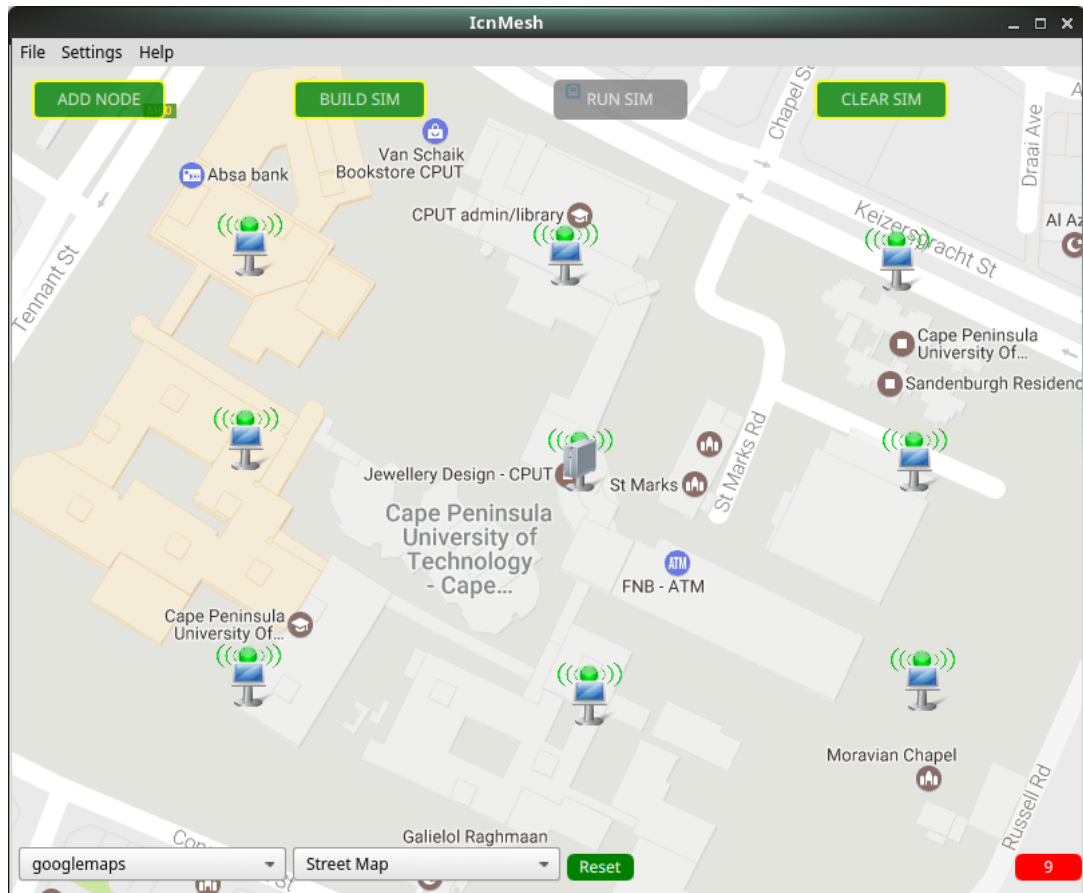


Figure 5.25: Single producer, multiple consumers network setup

5.3.3 Simulator Configuration

Table 5.5 summarizes the configuration of the simulator for this experiment.

Table 5.5: IcnMesh configuration for single source multiple clients

Parameter	Value
Number of Mesh Nodes	9
No. of Content Producers /Servers	1
No. of Content Consumers /Servers	8
Radio Interface	IEEE 802.11n
Carrier Frequency	2.4 GHz
Tx power level	20 dBm
Propagation Model	ThreeLog Distance Propagation Loss Model + Nakagami Propagation Loss

	Model
Link Rate	65 Mbps
Medium Access Layer	IEEE 802.11
Network Type	Information-Centric
Mesh Routing Protocol	Multicast
Transport Protocol	N/A
Content Request Pattern	Constant rate (10 requests/second)
Content-Prefix assignment	Automatic
Simulation Time	30 seconds.

5.3.4 Results and Discussion

In this experiment, one producer node publishes Content and eight consumer nodes are configured to request for the same Content at a constant rate. The main objective was to observe the effects on the server load. Another objective was to observe the patterns of data received by the consumer nodes in order to ascertain whether the content distribution scheme was fair: meaning that there should be no significant differences in the data-in graphs of the consumer nodes.

Figures 5.27 and 5.29 show the pattern of data traffic received by the consumer nodes and transmitted by the producer nodes. It can be seen that there are no significant differences between the interest and data graphs for all consumer nodes. The amount of data received by each consumer node is roughly the same. It can therefore be deduced that absent other factors, Content distribution to the consumer nodes is fair.

The interest and data graphs of producer0 (figures 5.30 and 5.31), however, shows that despite the consumer nodes sending several requests at once, the interest graph of the producer node matches the request pattern of an individual node. The explanation for this is that mesh routers aggregate Content requests and only send forward a single request to the producer. This has an effect of reducing the server load experienced at the producer load.

It can also be seen that the data sent from the producer node is much less than the sum of data received by the consumer nodes; and is also equal to the data received by a single consumer. In the host-centric scenario, eight consumers requesting data

of size X would result in a load of size $8X$ at the server node. In the ICN case, it can be seen that the server load is reduced by a factor of 8.

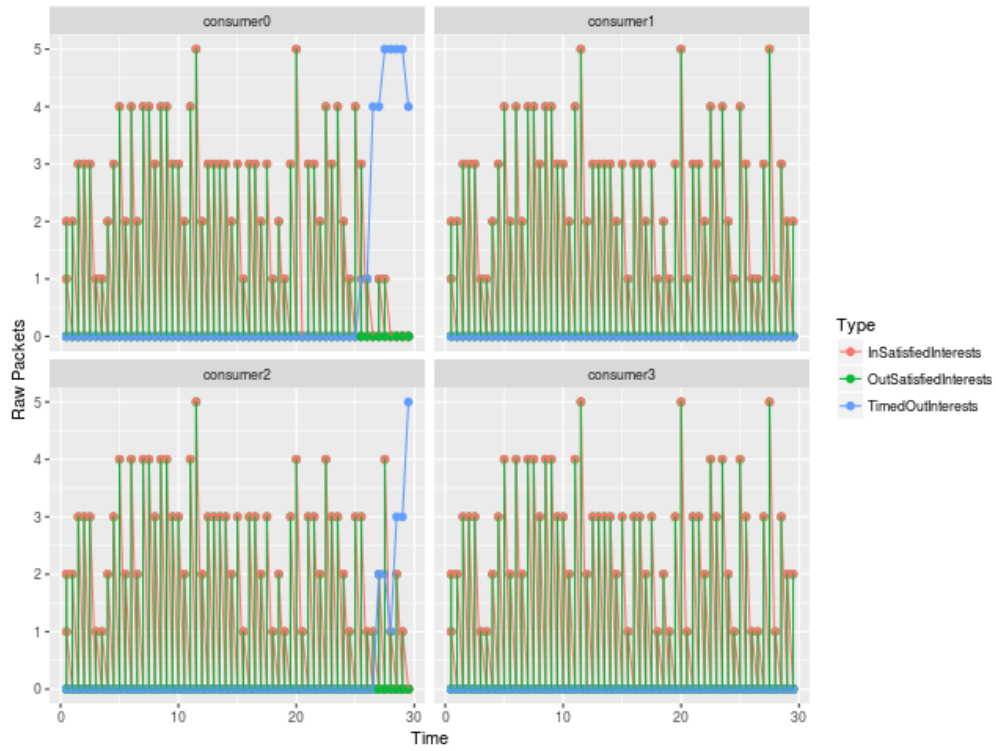


Figure 5.26: Graphs of interest packets logged by consumers 0-3

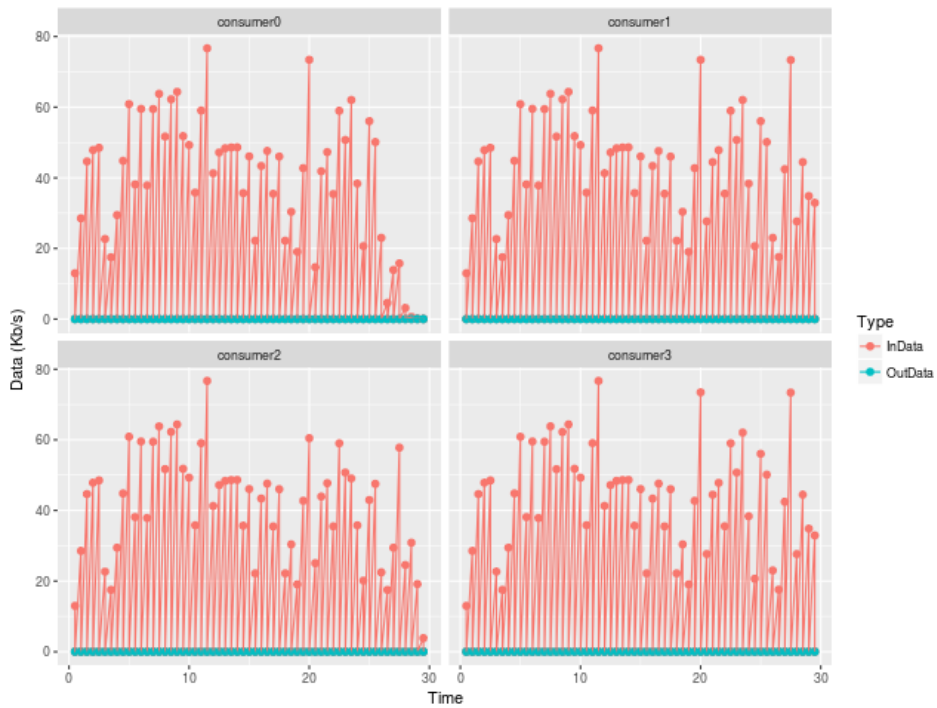


Figure 5.27: Graphs of data received by consumers 0-3

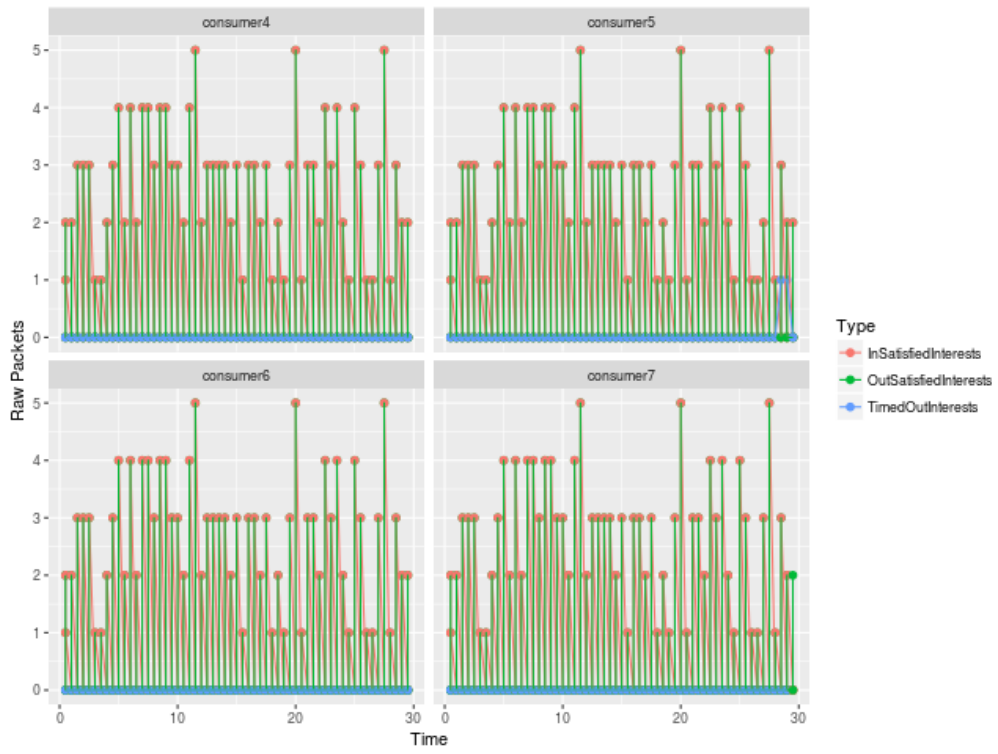


Figure 5.28: Graphs of Interest packets logged by consumers 4-7

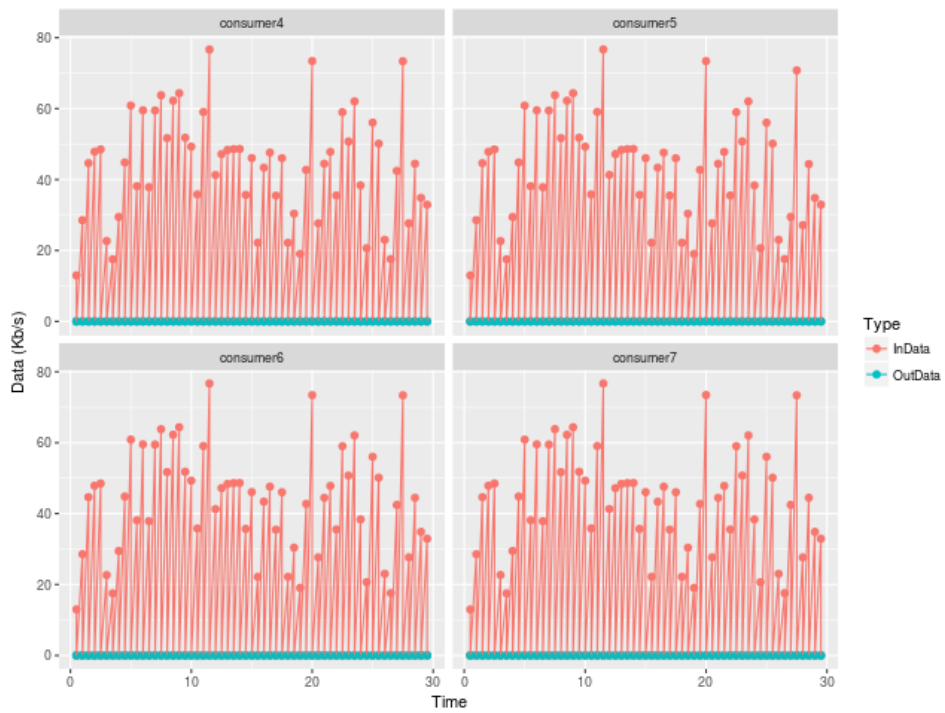


Figure 5.29: Graphs of data received by consumers 4-7

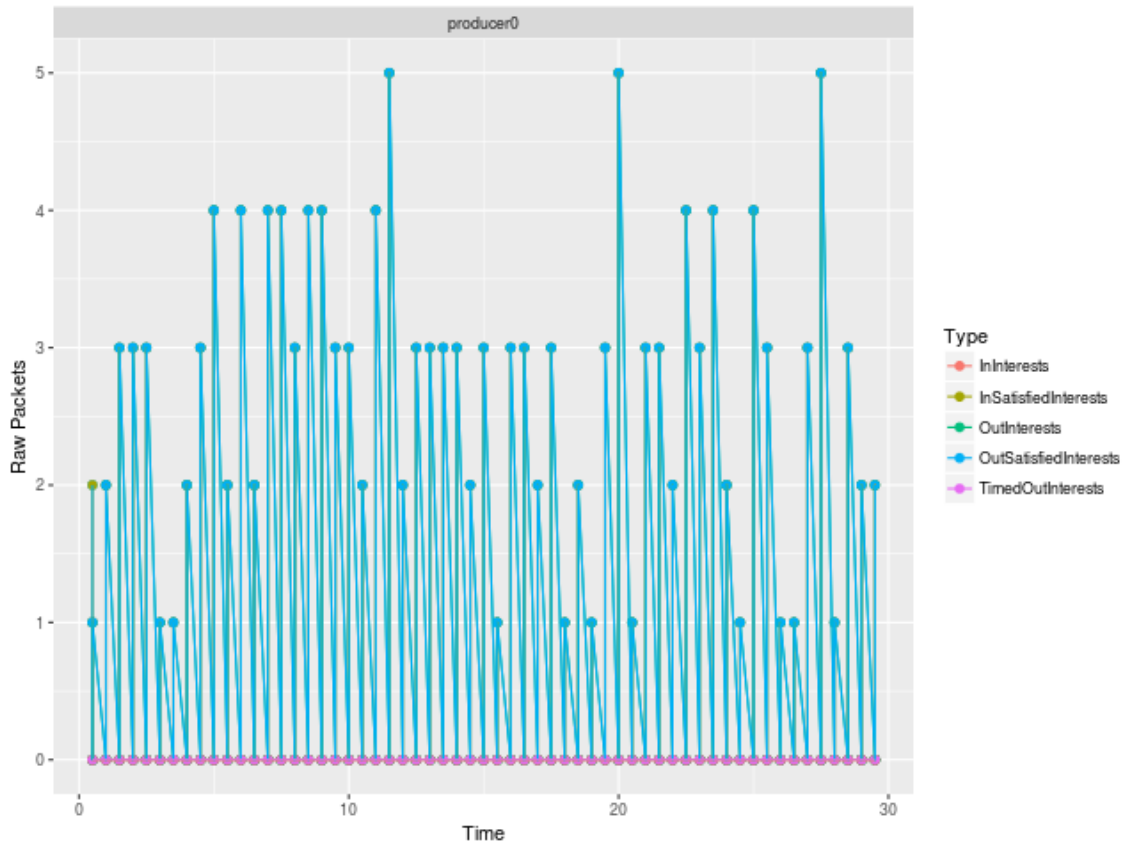


Figure 5.30: Graph of interest packets received by producer0

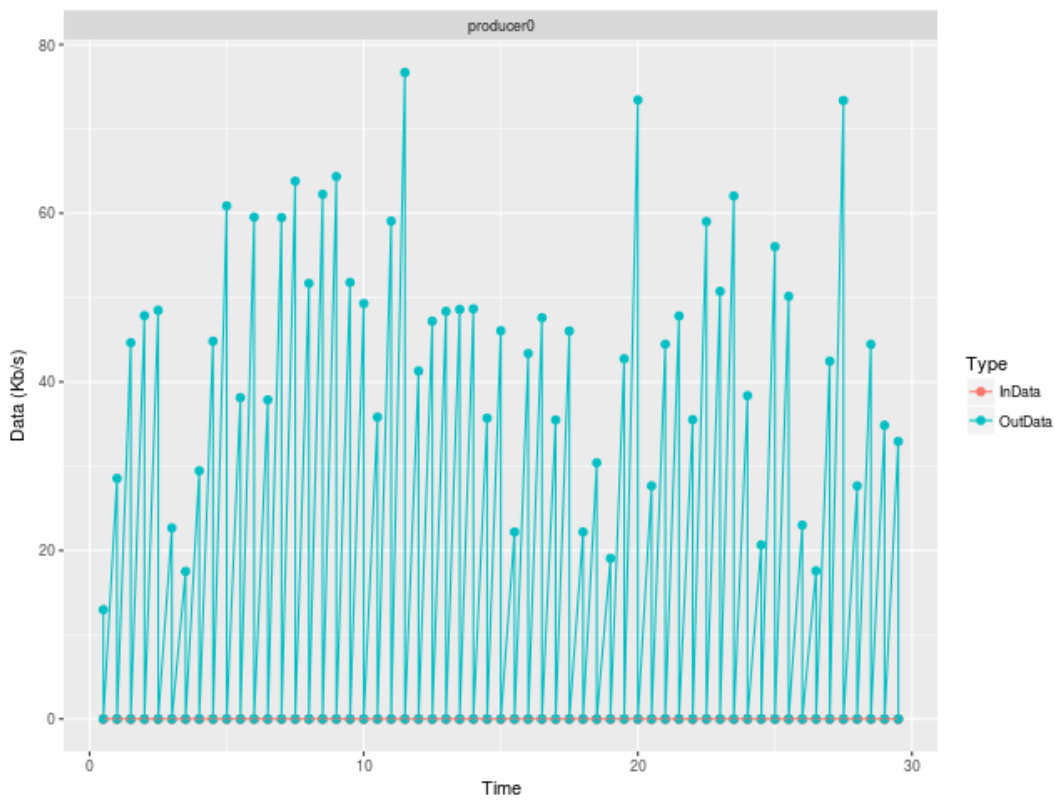


Figure 5.31: Graph of Producer0 data transfer

5.4 Distribution of Content with Varying Popularity in IC-WMN.

In this section a medium-sized WMN with multiple Content producers and consumers is simulated. The main objective of this experiment was to observe the Content distribution patterns at producer and consumer nodes in the WMN when the Content objects have different request popularities. This experiment was run multiple times in order to get an average result. All the re-runs of the experiment produced similar results.

5.4.1 Objectives

- Observe Content distribution patterns on consumer and producer nodes in the simulated WMN for multiple Content sources with varying popularity.
- Predict real-world performance of ICN scheme in WMN with multiple consumers and producers

5.4.2 Network Setup

The network topology comprised of 16 mesh nodes arranged in a square pattern with 4 producer nodes connected to mesh nodes along the diagonal of the square. One consumer node was connected to each of the remaining mesh nodes. This topology is represented in figure 5.32 below.

Content was randomly assigned to producer and consumer nodes in the network. A random algorithm was used to subscribe consumers to Content published in the WMN. This resulted in a disproportionate distribution in Content subscription by the consumers. Table 5.6 shows the prefixes published by the producer nodes and table 5.7 shows the prefixes that the consumer nodes subscribed to. 5 nodes subscribed to “/macbeth”, 3 nodes each subscribed to “/atonement” and “/enchantment” and only one consumer node subscribed to “/dissolution”.

All Content objects were of the same size (1000 Kb).

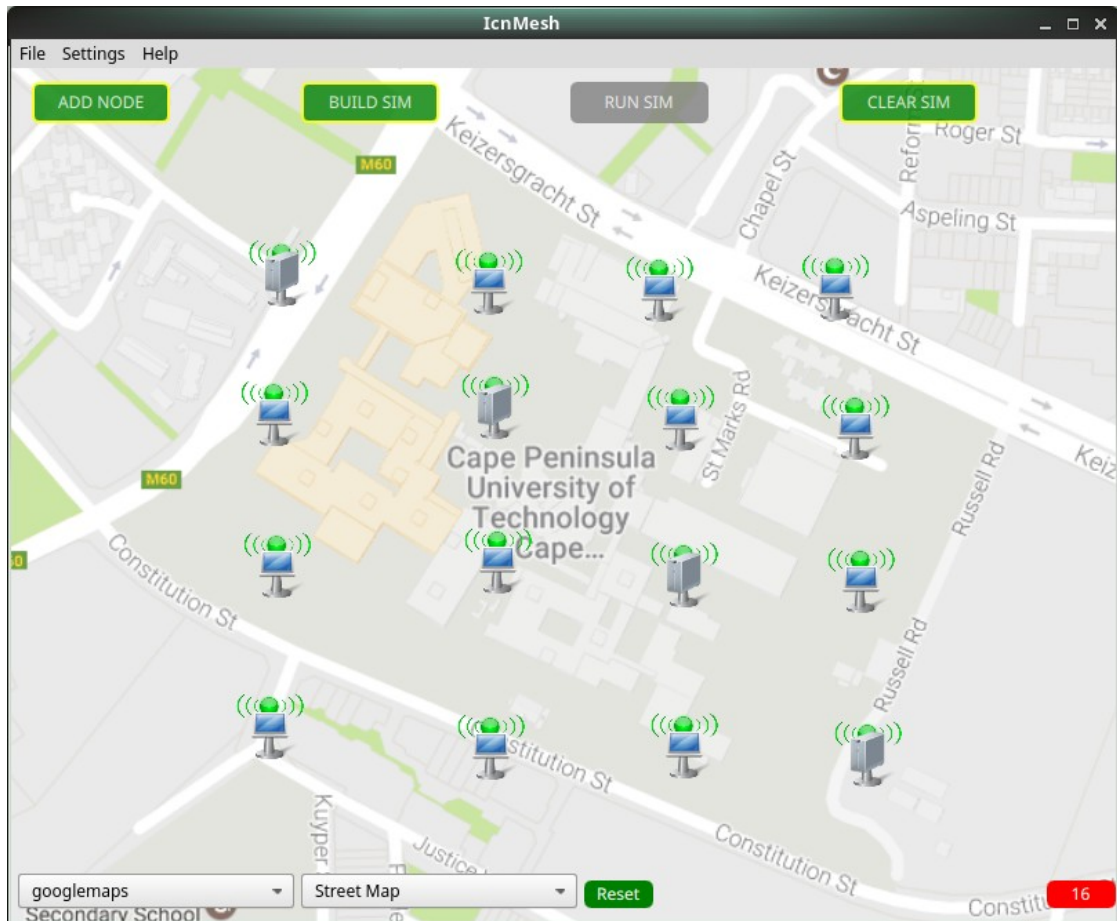


Figure 5.32: Network topology of simulated medium sized WMN

Table 5.6: Content prefixes published by Producer nodes

Producer Node	Content Prefix
Producer0	/macbeth
Producer1	/enchantment
Producer2	/atonement
Producer3	/dissolution

Table 5.7: Consumer node Content prefix subscriptions

Consumer Node	Content Prefix
Consumer0	/macbeth
Consumer1	/atonement
Consumer2	/enchantment
Consumer3	/macbeth
Consumer4	/macbeth
Consumer5	/atonement
Consumer6	/enchantment
Consumer7	/macbeth
Consumer8	/atonement
Consumer9	/enchantment
Consumer10	/dissolution
Consumer11	/macbeth

5.4.3 Simulator Configuration

Table 5.8: IcnMesh configuration for multiple producers and consumers

Parameter	Value
Number of Mesh Nodes	16
No. of Content Producers /Servers	4
No. of Content Consumers /Servers	12
Radio Interface	IEEE 802.11n
Carrier Frequency	2.4 GHz
Tx power level	20 dBm
Propagation Model	ThreeLog Distance Propagation Loss Model + Nakagami Propagation Loss Model
Link Rate	65 Mbps
Medium Access Layer	IEEE 802.11
Network Type	Information-Centric

Mesh Routing Protocol	Multicast
Transport Protocol	N/A
Content Request Pattern	Zipf-Mandelbrot Distribution
Content-Prefix assignment	Automatic
Simulation Time	30 seconds.

5.4.4 Results and Discussion

The results in the result graphs (figure 5.33 to figure 5.40) show disproportionate Content distribution among consumers in the WMN.

Nodes that subscribed to popular Content (“/macbeth”) show the highest proportion of satisfied requests and the lowest proportion of timed-out interests. Nodes that subscribed to unpopular Content had the lowest hit-rates and the highest number of timed-out requests, for example, consumer10, the only one to subscribe to “/dissolution” has most of its requests timed out and received a lower hit rate. It can therefore be concluded that the Content distribution scheme favours popular Content.

Because of the nature of Content distribution in the network, the load on the producer nodes is reduced. Producer0 served the most popular Content and received the highest number of Interest packets. Mesh nodes aggregate Content requests as seen in the previous section. This reduces the effective load on the producer (server load).

The network throughput in this setup is observed to be low. This is because all nodes in the WMN are on the same channel. The IEEE 802.11 MAC in use broadcasts every transmission sent to all nodes in the network. Packet collisions occur when several mesh nodes attempt to transmit at the same time resulting in poor use of bandwidth.

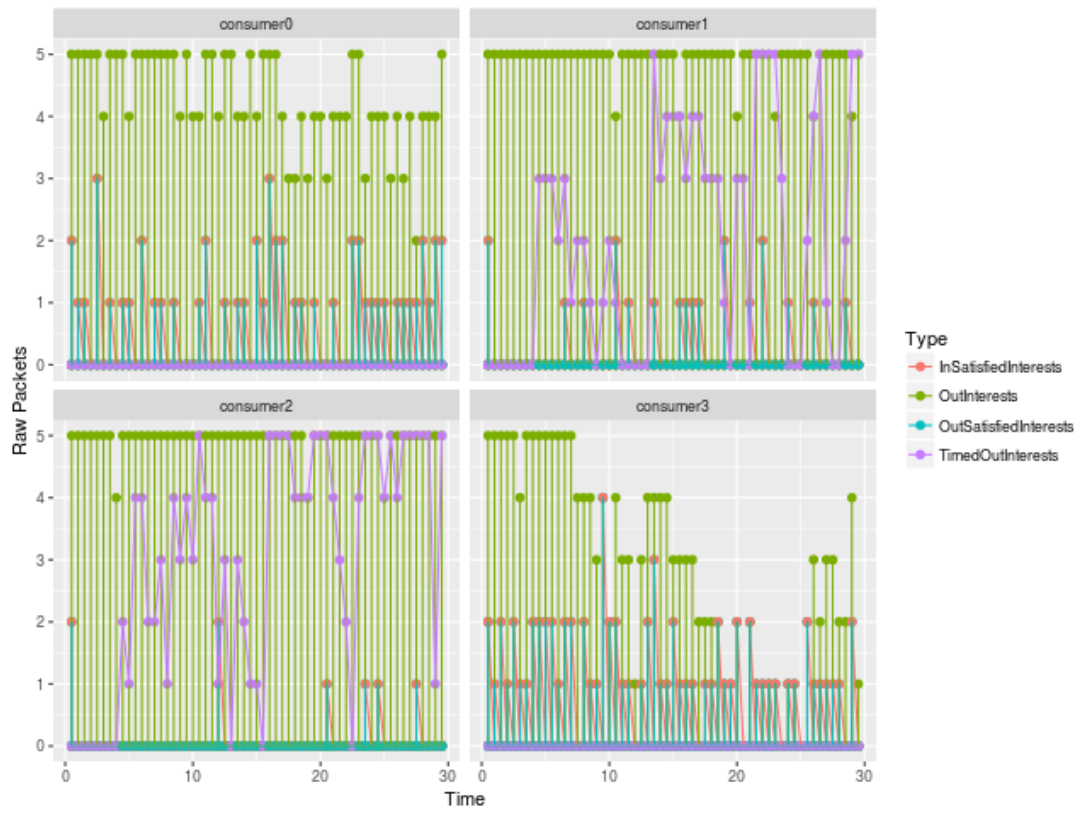


Figure 5.33: Graph of interest packets logged by consumer nodes 0-3

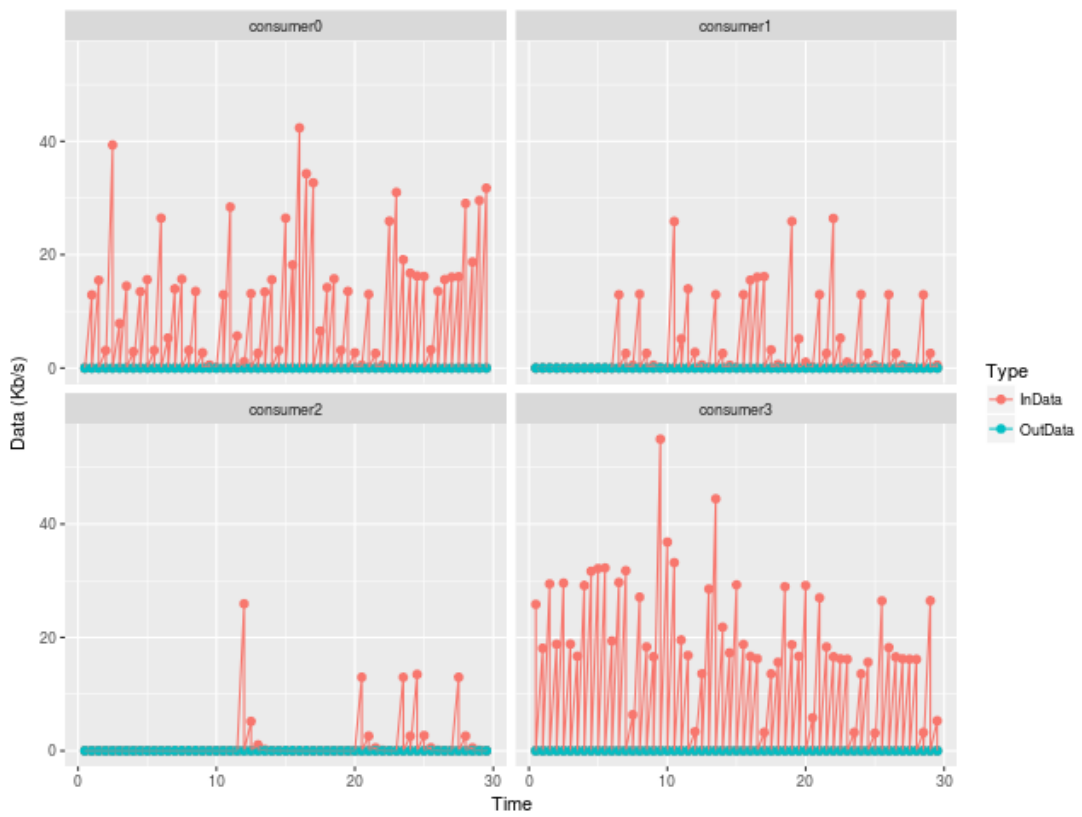


Figure 5.34: Graph of data received at consumer nodes 0-3

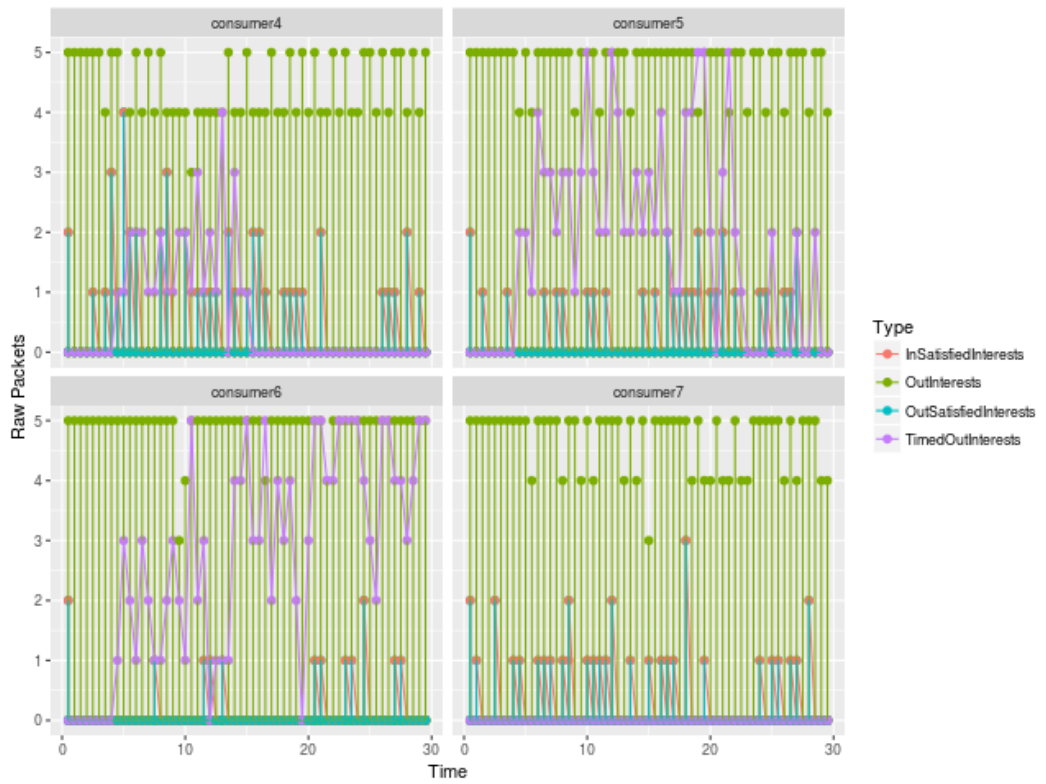


Figure 5.35: Graph of interest packets logged by consumer nodes 4-7

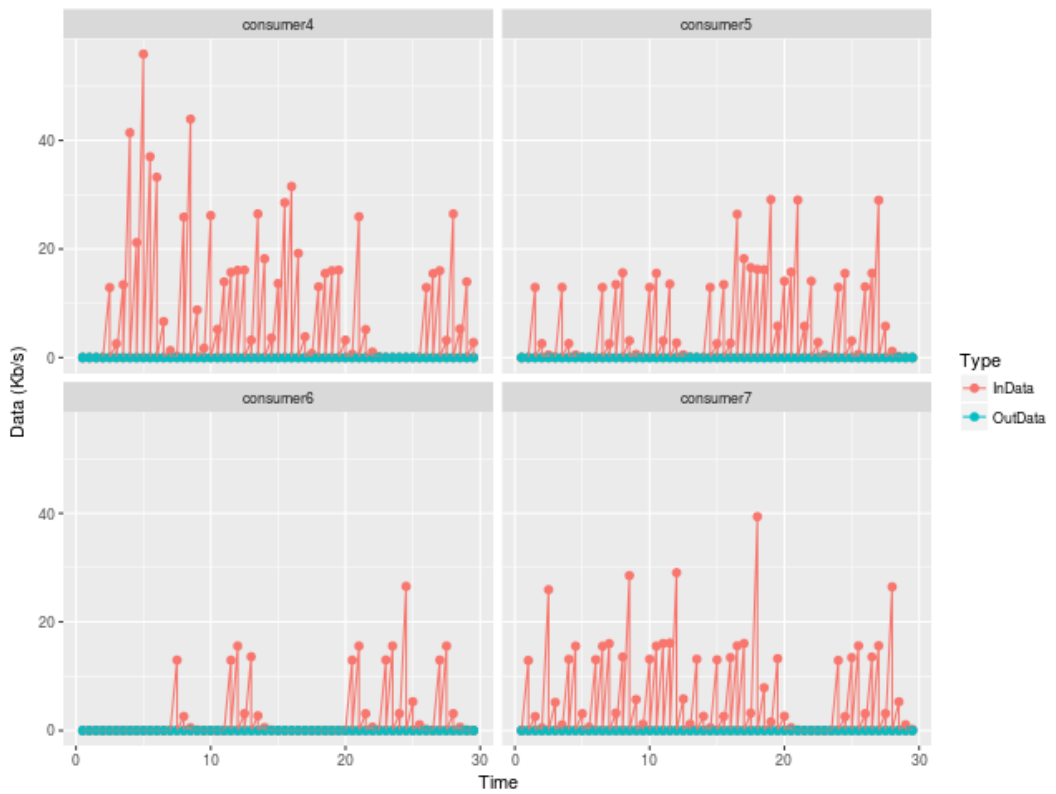


Figure 5.36: Graph of data received by consumer nodes 4-7

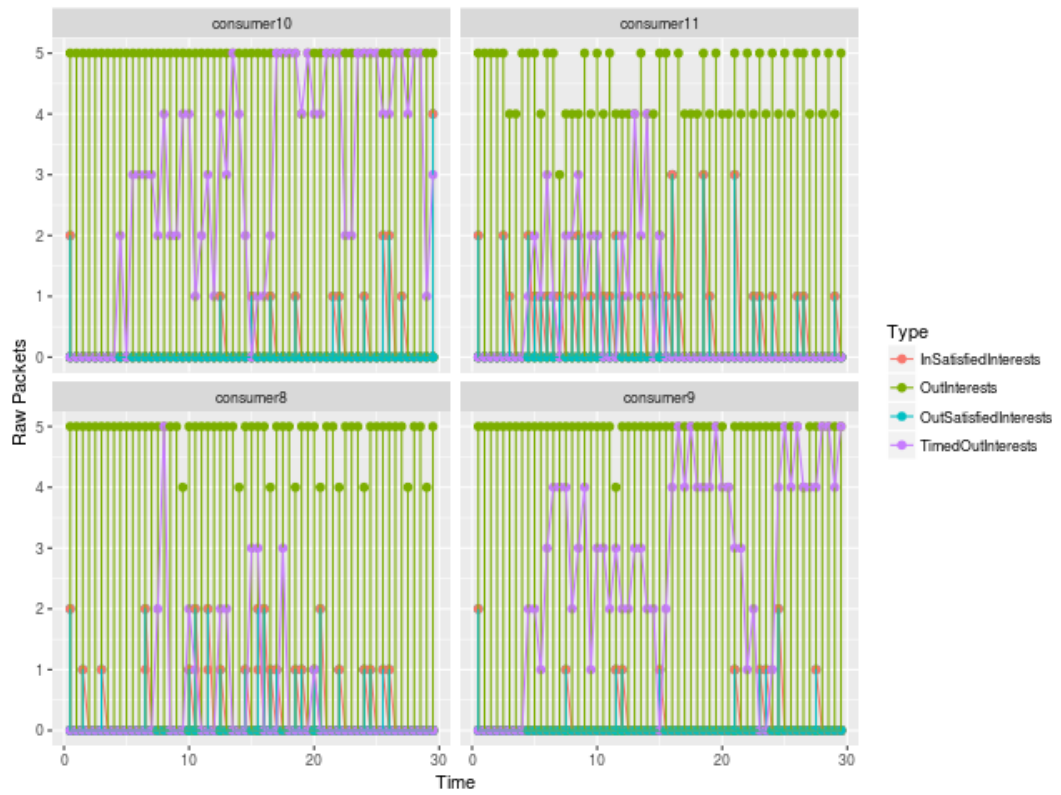


Figure 5.37: Graph of interest packets at consumer nodes 8-11

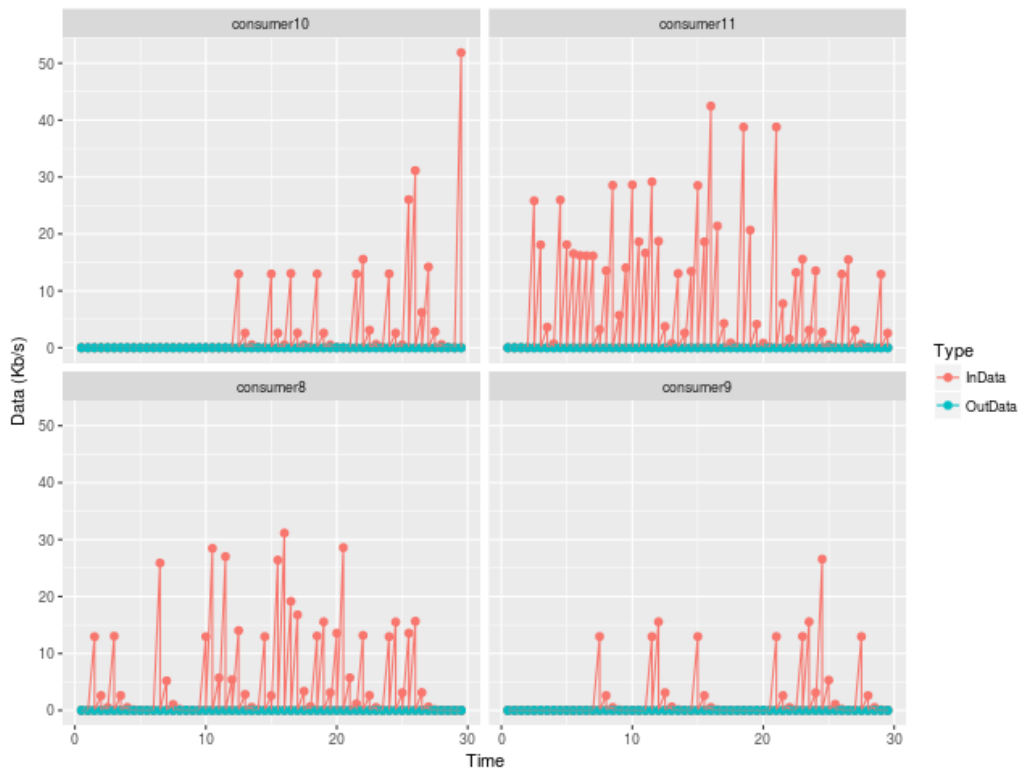


Figure 5.38: Graph of data received by consumer nodes 8-11

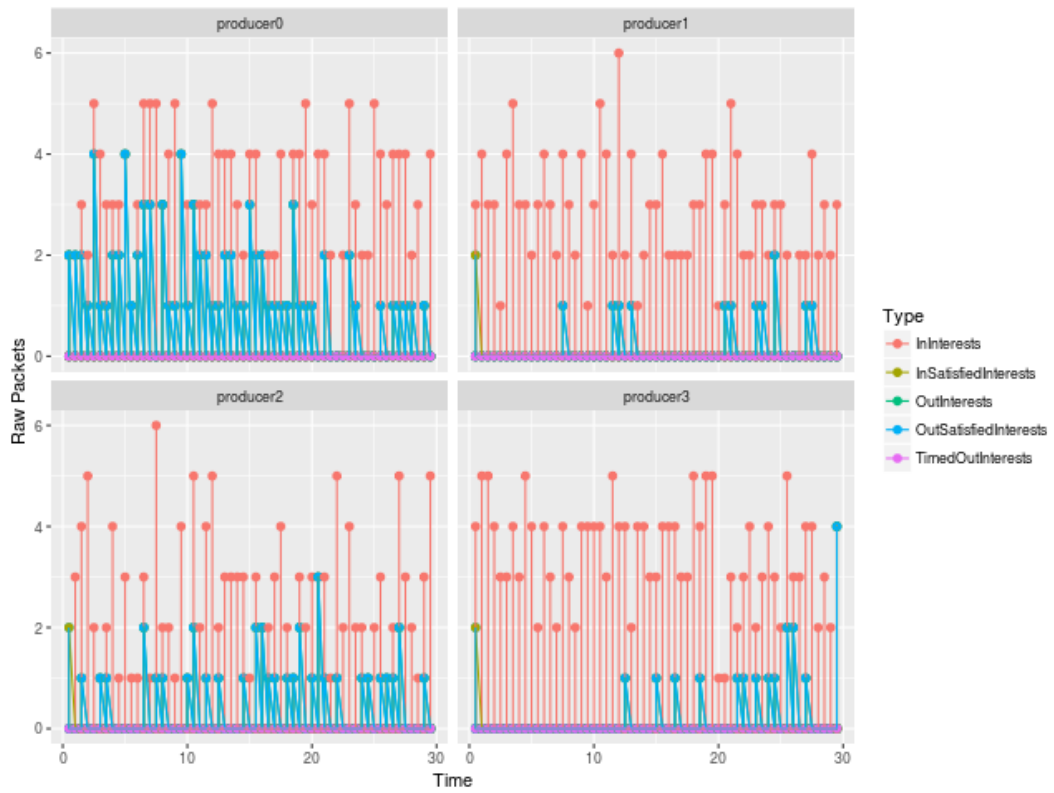


Figure 5.39: Graph of interest packets logged by producer nodes

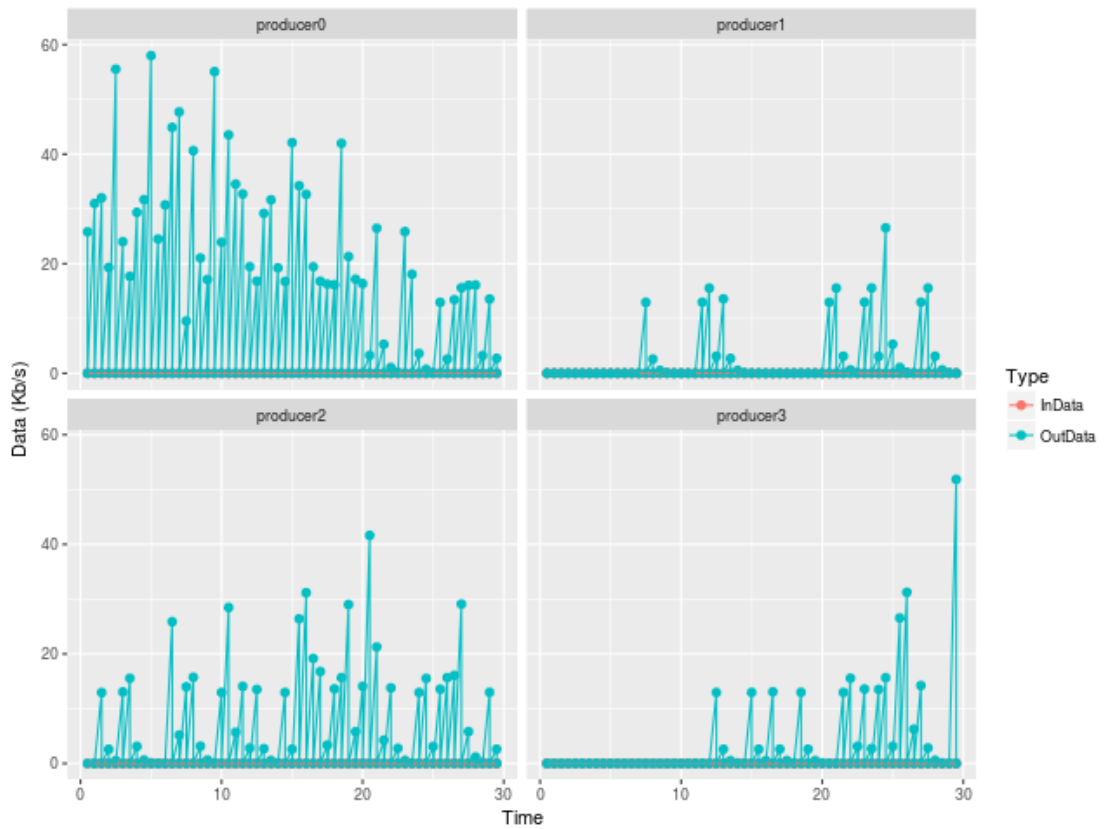


Figure 5.40: Graph of data transfer at producer nodes

5.5 Summary of Results

This section began with simple simulations of a simple host-centric WMN and a simple information-centric WMN. The experiments verified that the IcnMesh simulator was capable of simulating WMNs and ICN.

5.5.1 Average Hop Count

This is the total number of hops between the requesting nodes (consumers) and the serving nodes (producers) divided by the total number of served requests.

When the IEEE 802.11 MAC is used, the network operates as an ad hoc network. This requires all mesh nodes in the WMN to be directly connected to each other in order to exchange traffic. In this scenario, the average hop-count is 1 : from the consumer's mesh node to the producer's mesh node. The problem with this setup is that nodes that are far from each other experience poor throughput or no throughput at all.

The IEEE 802.11s MAC is able to route packets through the most efficient path which could be through multiple mesh nodes for example, in the hidden-node experiment, it can be seen that data packets traversed 5 nodes to get to their destination. However, this was not investigated further as the standard IEEE 802.11 MAC scheme was used in subsequent experiments.

5.5.2 Server Load Reduction.

This is defined as 1 minus the ratio between the number of Content objects generated by producers and the number of Content objects generated by producers when all requests are served from the producer node rather than from the cache of any other node. In a pure host-centric network, the the server responds to every request sent by clients. This can overwhelm the server when multiple clients request for Content at the same time and is the basis for Distributed Denial (DDoS) of Service attacks.

In the simulation of multiple consumers with a single producer, it was noted that the effective load on the server is significantly reduced when multiple Consumers simultaneously request for Content from the producer. The producer node responded

with a single packet of data for every request of the same type received. The mesh nodes then route data to consumers that requested. The server load reduction on the IC-WMN can therefore be expressed as:

$$\text{Server Load Reduction} = 1 - \left[\frac{\text{Content objects generated in ICN WMN}}{\text{Content objects generate in host-centric WMN}} \right]$$

From the results generated, it can be seen that the server load is reduced by $1 - [(X-1)/X]$, where X is the total number of Content requests.

5.5.3 Hit Ratio

This refers to the ratio between the total requests satisfied to the total number of requests sent. The hit ratio is averaged on a client node that requests Content from a server.

In the simple ICN mesh network, it can be seen that all interests sent by the node are satisfied thus indicating a hit ratio of 1. In the single producer, multiple consumer experiment, the hit ratio for all nodes is also 1.

$$\text{(Content Hit ratio) \%} = \left[\frac{\text{Content Hits}}{\text{Content Hits} + \text{Content Misses}} \right] \times 100$$

5.5.4 Network Load Reduction

The refers to the total number of packets transmitted by the network layer for a single transfer. This accounts for all control as well as data packets. The network load is averaged across all the transfers initiated in the network. Thus, the network load is expressed as packets transmitted per file download.

In the single-producer multiple-consumer experiment, it can be seen that the network experienced a lower load compared to an equivalent host-centric scenario where the producer would respond to every request. Therefore, if X is the total network load for one request, and Y is the total number of simultaneous requests, a host-centric network would experience a load of XY, whereas in the ICN scenario, the network load would only be slightly higher than X to account for the higher number of interest packets generated. This is because of the role mesh nodes play in aggregating Content requests and data transfers.

Although this experiment did not consider caching by mesh nodes, it can be predicted that if the network load would be further reduced if the requests were served from mesh node caches rather than directly from the consumer.

Network Load Reduction = $1 - [(\text{Total packets transmitted in ICN WMN}) / (\text{Total packets transmitted in host-centric WMN})]$

5.5.5 Latency

This is the time interval between the sending of an Interest packet and receipt of data. Network communications can be affected by delays on the physical channel or during the routing of traffic. In the ideal case, zero latency is preferred; that is, there should be no time difference between when a packet is sent between network nodes.

The simple ICN mesh simulation showed of less than 100mS between packet transfers. In that simulation, the standard IEEE 802.11 MAC was installed on mesh nodes. The time taken for a producer to serve requests received also contributes to latency. It can be seen from the producer interest graphs that the the InInterests and outSatisfiedInterest graphs align perfectly. This shows that very little delay is contributed by the producer node responding to requests.

In the hidden-node experiment, a much greater latency of 2 Seconds was observed when the IEEE 802.11s MAC protocol was used. This resulted in interest packets sent from the consumer node being timed out and was the basis of ruling out the MAC scheme. Latency can be expressed as:

Latency = (Frame Serialization Time) + (Link Media Delay) + (Queueing Delay) + (Node Processing Delay)

As the network parameters are the same for both the standard and 802.11s WMNs, the increase in latency can be attributed to additional processing delay with the 802.11s MAC scheme.

5.5.6 Communication Overhead

Communication overhead can be expressed as:

$(\text{Communication Overhead})\% = 100 \times [(\text{Total Bits in Network transfer}) - (\text{Actual bits of data Contained in transfer})] / (\text{Total Bits in Network transfer}).$

For a single request, it can be seen that the network load for the IC-WMN is higher than the host-centric WMN, but the overall network load is reduced as multiple requests are made.

5.6 Summary of Chapter Five

In this chapter, experiments are run on the IcnMesh simulator developed as specified in Chapter 4. The chapter begins by running validation tests on the simulator. The first test is of a host-centric network that proves basic connectivity between the mesh network nodes and between mesh nodes, client and server nodes. The second validation test is of a simple ICN network. The objective of this test was to prove basic information-centric networking in the simulated network, named-data routing and communication between consumer and producer nodes.

The next simulation experiment was to test mesh routing in IcnMesh. In this setup, the producer and consumer nodes were separated such that direct communication between was not possible and could only be done through a relay nodes. Two simulations were run: first with a standard IEEE 802.11 MAC stack installed on the nodes and secondly with the IEEE 802.11s MAC stack. In the IEEE 802.11 MAC case, it was not possible to route traffic between the producer and consumer nodes. It was possible to route packets between the producer and consumer nodes using IEEE 802.11s routing protocol.

Finally, a medium size WMN with multiple Content producers and consumers is simulated. It is shown that Information-Centric networking significantly improves Content distribution performance in WMNs by eliminating the bottleneck and work-locality limitations brought about by host-centric networking. The server load on the Content producers is reduced significantly because of in-network caching performed by mesh nodes in the WMN.

In the next chapter, the thesis is concluded and directions for future research are presented.

CHAPTER SIX

6 CONCLUSION AND FUTURE WORK

In this chapter, the thesis is concluded by highlighting the contributions in each chapter and summarizing the results. A number of possible directions for future research are also presented.

In Chapter 1 and 2, the theoretical background to this research was given. The challenges facing WMNs designed for Internet access were discussed. A background of Content distribution schemes for wireline and wireless Internet was also given.

Chapter 3 explained the research methodology and gave the design constraints for the simulator to be used in the research. Chapter 3 also gave the metrics that were used in the research and described the data collection and data analysis systems.

Chapter 4 described the implementation of IcnMesh simulator and integration with external libraries that IcnMesh simulator relies on. A visual representation of IcnMesh logic was given along with the class structure of the application.

In Chapter 5, a number of simulation experiments are carried out on IcnMesh. These begin with simple experiments and end with a medium-sized WMN with multiple Content producers and consumers. It is from this Chapter that the research objectives are fulfilled.

This research has been shown that with improvements, Information-Centric networking can significantly improve performance of WMNs by minimizing the impact of the bottleneck and workload locality limitations. By leveraging in-network caching and aggregating Content request packets, an Information-Centric WMN reduces server load and congestion along routes leading to the server. Popular Content is also fetched seamlessly in the caches of mesh nodes in the network.

The main problem noted in the simulated IC-WMNs is the inability of the chosen routing scheme to route traffic to hidden nodes in the WMN. The NDO routing scheme relies on all nodes in the WMN to be associated with each other in order for data transfer to take place. The IEEE 802.11s Layer 2 routing scheme was tested but not selected for further simulation experiments owing to significant delays experienced during transmission of packets in the network. These delays resulted in several interest packets being timed out and overall throughput of the network degraded.

It is noted that this research on WMN was done through simulation. Research on ad hoc networks, MANETs and WMN has mostly been carried out using simulation tools (Al-Arnaout, 2014, Canali, C. 2010, Das, *et al.*, 2007, Jun & Sichitiu, 2003.). The main reason many researchers prefer to use simulations rather than actual hardware is because of the cost and difficulty in deploying a large scale WMN, having actual users on the network. It would also take longer to collect results. It is time consuming to reconfigure the device software and could be downright impossible to modify the hardware. For these reasons, WMN research is traditionally carried out using simulation tools.

Despite the popularity of simulation tools, recent studies revealed the inherent limitations these have in modelling the physical layer and aspects of the MAC layer (Andel & Yasinsac, 2006). Therefore, it should be acknowledged that results from a simulation tool only gives a rough estimate of performance. It is also possible that different results can be obtained by running the same simulated network topology/protocol on different simulation tools. This lack of consistency lowers the credibility of simulation tools. For these reasons, validation of test results should be carried out on actual hardware in an actual WMN.

6.1 Future Work and Research Directions

In this section some research directions for extending the work presented in this thesis are introduced.

From this work, it is proposed that in the near future, a routing scheme for IC-WMNs that is aware of underlying ICN forwarding schemes and mesh network topology be

developed. This should extend the performance of WMNs that rely on ICN for Content distribution.

Another line of research that is proposed is of an ICN scheme that uses IP as a transport layer. IP routing is well developed even for WMNs and there are routing schemes in existence that will readily support it. Such a scheme would bridge-the-gap between host-centric and Information-Centric approaches to Content distribution and gradually Introduce ICN in the mainstream. There have been attempts by Chen, *et al.*, (2012) to integrate a publish-subscribe ICN scheme with IP. A performance evaluation of Content distribution in an IP-based WMN with ICN Content distribution is suggested.

Information-Centric Networking research is still at it's infancy and real-world applications are yet to be implemented. Directions for research in ICN have been outlined by de Brito, *et al.*(2013). The Internet Research Task Force (IRTF) has also established an Information-Centric Networking Research Group (ICNRG) which outlines the following research challenges in ICN (ICNRG, 2016):

- Naming schemes for ICN, including scalable name resolution for flat names
- Scalable routing schemes
- Congestion control, QoS approaches, and caching strategies
- Metrics that make it possible to evaluate ICN implementations in a consistent manner
- Security and privacy, including scoping of information objects and access control to them
- Application/application-protocol design and APIs
- Business, legal and regulatory frameworks

REFERENCES

- Afanasyev, A., Moiseenko, I. and Zhang, L., 2012. ndnSIM: NDN simulator for NS-3. *University of California, Los Angeles, Tech. Rep.*
- Aguayo, D., Bicket, J., Biswas, S., De Couto, D.S. and Morris, R., 2003. MIT roofnet implementation.
- Ahlgren, B., Dannewitz, C., Imbrenda, C., Kutscher, D. and Ohlman, B., 2012. A survey of information-centric networking. *IEEE Communications Magazine*, 50(7).
- Akyildiz and Wang, 2009. *Wireless mesh networks* (Vol. 3). John Wiley & Sons.
- Akyildiz, I.F., Wang, X. and Wang, W., 2005. Wireless mesh networks: a survey. *Computer networks*, 47(4), pp.445-487.
- Al Asaad, A., Gopalakrishnan, S. and Leung, V., 2009, August. Peer-to-peer file sharing over wireless mesh networks. In *Communications, Computers and Signal Processing, 2009. PacRim 2009. IEEE Pacific Rim Conference on* (pp. 697-702). IEEE.
- Al Hamra, A., Barakat, C. and Turletti, T., 2006, June. Network coding for wireless mesh networks: A case study. In *Proceedings of the 2006 International Symposium on World of Wireless, Mobile and Multimedia Networks* (pp. 103-114). IEEE Computer Society.
- Al-Arnaout, Z., 2014. Content Replication and Placement Schemes for Wireless Mesh Networks.
- Amadeo, M. and Molinaro, A., 2011, November. CHANET: A content-centric architecture for IEEE 802.11 MANETs. In *Network of the Future (NOF), 2011 International Conference on the* (pp. 122-127). IEEE.
- Amadeo, M., Campolo, C. and Molinaro, A., 2012. CRoWN: Content-centric networking in vehicular ad hoc networks. *IEEE Communications Letters*, 16(9), pp.1380-1383.

Andel, T.R. and Yasinsac, A., 2006. On the credibility of manet simulations. *Computer*, 39(7), pp.48-54.

Aoun, B., Boutaba, R. and Kenward, G., 2006, May. Analysis of capacity improvements in multi-radio wireless mesh networks. In *Vehicular Technology Conference, 2006. VTC 2006-Spring. IEEE 63rd* (Vol. 2, pp. 543-547). IEEE.

Berkoben, K.A., 2011. *Extending broadband past the urban fringe with wireless mesh: a strategic analysis with policy implications for Kenya's Universal Service Fund* (Doctoral dissertation, Massachusetts Institute of Technology).

Berlin Freifunk. <http://berlin.freifunk.net/>.

Bernardos, C.J., Soto, I. and Banchs, A., 2008. IEEE 802.11 standards. *Medium Access Control in Wireless Networks*, p.235.

Campista, M.E.M., Esposito, P.M., Moraes, I.M., Costa, L.H.M., Duarte, O.C.M., Passos, D.G., De Albuquerque, C.V.N., Saade, D.C.M. and Rubinstein, M.G., 2008. Routing metrics and protocols for wireless mesh networks. *IEEE network*, 22(1).

Canali, C., Renda, M.E., Santi, P. and Burresti, S., 2010. Enabling efficient peer-to-peer resource sharing in wireless mesh networks. *IEEE Transactions on Mobile Computing*, 9(3), pp.333-347.

Chakraborty, S. and Nandi, S., 2012, January. MAC layer fairness in IEEE 802.11 DCF based wireless mesh networks. In *Computing, Networking and Communications (ICNC), 2012 International Conference on* (pp. 1138-1143). IEEE.

Chen, J., Arumaithurai, M., Fu, X. and Ramakrishnan, K.K., 2012, October. Coexist: integrating content oriented publish/subscribe systems with IP. In *Proceedings of the eighth ACM/IEEE symposium on Architectures for networking and communications systems* (pp. 223-234). ACM.

Chokkalingam, A. and Riyaz, F., 2004. BitTorrent protocol specification v 1.0. *CS/5321*.

Clausen, T., Hansen, G., Christensen, L. and Behrmann, G., 2001, September. The

optimized link state routing protocol, evaluation through experiments and simulation. In *IEEE Symposium on" Wireless Personal Mobile Communications*.

Clausen, Thomas, and Philippe Jacquet. *Optimized link state routing protocol (OLSR)*. No. RFC 3626. 2003.

Costa, L.H.M.K., Fdida, S. and Duarte, O.C.M.B., 2006. Incremental service deployment using the hop-by-hop multicast routing protocol. *IEEE/ACM Transactions on Networking*, 14(3), pp.543-556.

Crow, B.P., Widjaja, I., Kim, L.G. and Sakai, P.T., 1997. IEEE 802.11 wireless local area networks. *IEEE Communications magazine*, 35(9), pp.116-126.

da Silva, C., Passos, D., Duarte, J., Moraes, I. and de Albuquerque, C.V.N., 2010. Dyntun: A tool for providing multihoming support in wireless mesh networks. In *International Information and Telecommunication Technologies Symposium (I2TS)*.

Dannewitz, C., Kutscher, D., Ohlman, B., Farrell, S., Ahlgren, B. and Karl, H., 2013. Network of information (netinf)—an information-centric networking architecture. *Computer Communications*, 36(7), pp.721-735.

Das, S.M., Pucha, H. and Hu, Y.C., 2007. Mitigating the gateway bottleneck via transparent cooperative caching in wireless mesh networks. *Ad Hoc Networks*, 5(6), pp.680-703.

de Brito, G.M., Velloso, P.B. and Moraes, I.M., 2013. *Information Centric Networks: A New Paradigm for the Internet*. John Wiley & Sons.

Ding, J.W., Wang, W.T. and Wang, C.F., 2008, August. An efficient data replication scheme for peer-to-peer video streaming over wireless-mesh community networks. In *Intelligent Information Hiding and Multimedia Signal Processing, 2008. IHHMSP'08 International Conference on* (pp. 767-770). IEEE.

Dogar, F.R., Phanishayee, A., Pucha, H., Ruwase, O. and Andersen, D.G., 2008, September. Ditto: a system for opportunistic caching in multi-hop wireless networks. In *Proceedings of the 14th ACM international conference on Mobile computing and networking* (pp. 279-290). ACM.

ElRakabawy, S.M. and Lindemann, C., 2007, January. Peer-to-peer file transfer in wireless mesh networks. In *Wireless on Demand Network Systems and Services, 2007. WONS'07. Fourth Annual Conference on* (pp. 114-121). IEEE.

freie Netzwerke eV, F., Freifunk. Net.

Fricker, C., Robert, P., Roberts, J. and Sbihi, N., 2012, March. Impact of traffic mix on caching performance in a content-centric network. In *Computer Communications Workshops (INFOCOM WKSHPS), 2012 IEEE Conference on* (pp. 310-315). IEEE.

Fricker, C., Robert, P., Roberts, J. and Sbihi, N., 2012, March. Impact of traffic mix on caching performance in a content-centric network. In *Computer Communications Workshops (INFOCOM WKSHPS), 2012 IEEE Conference on* (pp. 310-315). IEEE.

Gummadi, K.P., Dunn, R.J., Saroiu, S., Gribble, S.D., Levy, H.M. and Zahorjan, J., 2003. Measurement, modeling, and analysis of a peer-to-peer file-sharing workload. *ACM SIGOPS Operating Systems Review*, 37(5), pp.314-329.

ICNRG 2016, *Information Centric Networking Research Group* , 14 November 2016, Internet Research Task Force Available from: <https://irtf.org/icnrg>. [10 May 2017].

Jacobson, V., Mosko, M., Smetters, D. and Garcia-Luna-Aceves, J.J., 2009. Content centric networking. *Whitepaper 2007*.

Jiang, W., Zhang, Z. and Zhong, X., 2007, March. High throughput routing in large-scale multi-radio wireless mesh networks. In *Wireless Communications and Networking Conference, 2007. WCNC 2007. IEEE* (pp. 3598-3602). IEEE.

Jin, S. and Wang, L., 2005, October. Content and service replication strategies in multi-hop wireless mesh networks. In *Proceedings of the 8th ACM international symposium on Modeling, analysis and simulation of wireless and mobile systems* (pp. 79-86). ACM.

Johnson, D., Ntlatlapa, N. and Aichele, C., 2008. Simple pragmatic approach to mesh routing using BATMAN.

Jun, J. and Sichitiu, M.L., 2003. The nominal capacity of wireless mesh networks.

IEEE wireless communications, 10(5), pp.8-14.

Koponen, T., Chawla, M., Chun, B.G., Ermolinskiy, A., Kim, K.H., Shenker, S. and Stoica, I., 2007, August. A data-oriented (and beyond) network architecture. In *ACM SIGCOMM Computer Communication Review* (Vol. 37, No. 4, pp. 181-192). ACM.

Kroeger, T.M., Long, D.D. and Mogul, J.C., 1997, December. Exploring the Bounds of Web Latency Reduction from Caching and Prefetching. In *USENIX Symposium on Internet Technologies and Systems* (pp. 13-22).

Kumar, S., Raghavan, V.S. and Deng, J., 2006. Medium access control protocols for ad hoc wireless networks: A survey. *Ad Hoc Networks*, 4(3), pp.326-358.

Lin, X. and Ni, L.M., 1993. Multicast communication in multicomputer networks. *IEEE transactions on Parallel and Distributed Systems*, 4(10), pp.1105-1117.

Deering, S.E., 1988. Host extensions for IP multicasting.

Liu, H., Guo, Y. and Zhu, Y., 2007. *Unified peer-to-peer and cache system for content services in wireless mesh networks*. U.S. Patent Application 12/733,336.

Maggs, B.M., auf der Heide, F.M., Vocking, B. and Westermann, M., 1997, October. Exploiting locality for data management in systems of limited bandwidth. In *Foundations of Computer Science, 1997. Proceedings., 38th Annual Symposium on* (pp. 284-293). IEEE.

Montemurro, M.A., 2001. Beyond the Zipf–Mandelbrot law in quantitative linguistics. *Physica A: Statistical Mechanics and its Applications*, 300(3), pp.567-578.

Neumann, A., Aichele, C., Lindner, M. and Wunderlich, S., 2008. Better approach to mobile ad hoc networking (BATMAN). *IETF draft*, pp.1-24.

Ng, P.C., Liew, S.C., Sha, K.C. and To, W.T., 2005. Experimental study of hidden node problem in IEEE 802.11 wireless networks. *Sigcomm Poster*, 26.

Passarella, A., 2012. A survey on content-centric technologies for the current Internet: CDN and P2P solutions. *Computer Communications*, 35(1), pp.1-32.

Passos, D., Teixeira, D.V., Muchaluat-Saade, D.C., Magalhães, L.C.S. and Albuquerque, C., 2006, December. Mesh network performance measurements. In *International Information and Telecommunications Technologies Symposium (I2TS)* (pp. 48-55).

Robinson, J. and Knightly, E.W., 2007, May. A performance study of deployment factors in wireless mesh networks. In *INFOCOM 2007. 26th IEEE International Conference on Computer Communications. IEEE* (pp. 2054-2062). IEEE.

Sivasubramanian, S., Szymaniak, M., Pierre, G. and Steen, M.V., 2004. Replication for web hosting systems. *ACM Computing Surveys (CSUR)*, 36(3), pp.291-334.

Stallings, W., 1987. *Handbook of computer-communications standards; Vol. 1: the open systems interconnection (OSI) model and OSI-related standards*. Macmillan Publishing Co., Inc..

Stoffers, M. and Riley, G., 2012, August. Comparing the ns-3 propagation models. In *Modeling, Analysis & Simulation of Computer and Telecommunication Systems (MASCOTS), 2012 IEEE 20th International Symposium on* (pp. 61-67). IEEE.

Stoica, I., Morris, R., Liben-Nowell, D., Karger, D.R., Kaashoek, M.F., Dabek, F. and Balakrishnan, H., 2003. Chord: a scalable peer-to-peer lookup protocol for internet applications. *IEEE/ACM Transactions on Networking (TON)*, 11(1), pp.17-32.

Tarkoma, S., Ain, M. and Visala, K., 2009, April. The Publish/Subscribe Internet Routing Paradigm (PSIRP): Designing the Future Internet Architecture. In *Future Internet Assembly* (pp. 102-111).

Wang, L., Hoque, A.K.M.M., Yi, C., Alyyan, A. and Zhang, B., 2012. OSPFN: An OSPF based routing protocol for Named Data Networking. *University of Memphis and University of Arizona, Tech. Rep.*

Wu, X., Liu, J. and Chen, G., 2006, October. Analysis of bottleneck delay and throughput in wireless mesh networks. In *Mobile ad hoc and Sensor Systems (MASS), 2006 IEEE International Conference on* (pp. 765-770). IEEE.

APPENDIX A

This is an example Rscript file used to generate graphs from the IcnMesh trace file
This script generates eight graphs for consumers 0-3.

```
#!/usr/bin/env Rscript

# install.packages ('ggplot2')
library(ggplot2)

data = read.table("IcnMeshTrace.txt", header=T)
data$Node = factor(data$Node)
data$Kilobits <- data$Kilobytes * 8
data$Packets <- data$PacketRaw
data$Type = factor(data$Type)

data2 = subset(data, Type %in% c("OutInterests", "InSatisfiedInterests",
"OutSatisfiedInterests", "TimedOutInterests"))
data2 = subset(data2, Node %in%
c("consumer0", "consumer1", "consumer2", "consumer3"))

data = subset(data, Type %in% c("InData", "OutData"))
data = subset(data, FaceId %in% c("257", "256"))
data = subset(data, Node %in%
c("consumer0", "consumer1", "consumer2", "consumer3"))

g.all <- ggplot(data, aes(x=Time, y=Kilobits, color=Type)) +
  geom_point(size=2) +
  geom_line() +
  ylab("Data (Kb/s)") +
  facet_wrap(~ Node) +
  theme_grey()

png("graphs/consumer0.png", width=640, height=480)

print(g.all)

g.all <- ggplot(data2, aes(x=Time, y=Packets, color=Type)) +
  geom_point(size=2) +
  geom_line() +
  ylab("Raw Packets") +
  facet_wrap(~ Node) +
  theme_grey()
png("graphs/consumer0_requests.png", width=640, height=480)
print(g.all)

x = dev.off()
```

APPENDIX B

Excerpt of Trace file from ICN simulation

Time	Node	FaceId	FaceDescr	Type	Packets	Kilobytes	PacketRaw	KilobytesRaw
0.5	meshNode0	1	internal://	InInterests	0	0	0	0
0.5	meshNode0	1	internal://	OutInterests	0	0	0	0
0.5	meshNode0	1	internal://	InData	0	0	0	0
0.5	meshNode0	1	internal://	OutData	0	0	0	0
0.5	meshNode0	1	internal://	InNacks	0	0	0	0
0.5	meshNode0	1	internal://	OutNacks	0	0	0	0
0.5	meshNode0	1	internal://	InSatisfiedInterests	0	0	0	0
0.5	meshNode0	1	internal://	InTimedOutInterests	0	0	0	0
0.5	meshNode0	1	internal://	OutSatisfiedInterests	3.2	0	2	0
0.5	meshNode0	1	internal://	OutTimedOutInterests	0	0	0	0
0.5	meshNode0	256	netdev://[00:00:00:00:00:01]	InInterests	0	0	0	0
0.5	meshNode0	256	netdev://[00:00:00:00:00:01]	OutInterests	8	0.242188	8	0.242188
5	0.151367	256	netdev://[00:00:00:00:00:01]	InData	4.8	4.86094	3	3
0.5	meshNode0	256	netdev://[00:00:00:00:00:01]	OutData	0	0	0	0
0.5	meshNode0	256	netdev://[00:00:00:00:00:01]	InNacks	0	0	0	0
0.5	meshNode0	256	netdev://[00:00:00:00:00:01]	OutNacks	0	0	0	0
0.5	meshNode0	256	netdev://[00:00:00:00:00:01]	InSatisfiedInterests	0	0	0	0
0.5	meshNode0	256	netdev://[00:00:00:00:00:01]	InTimedOutInterests	0	0	0	0
0.5	meshNode0	256	netdev://[00:00:00:00:00:01]	OutSatisfiedInterests	4.8	0	0	0
0.5	meshNode0	256	netdev://[00:00:00:00:00:01]	OutTimedOutInterests	0	0	0	0
0.5	meshNode0	257	netdev://[00:00:00:00:00:0a]	InInterests	8	0.242188	5	5
0.5	meshNode0	257	netdev://[00:00:00:00:00:0a]	OutInterests	0	0	0	0
0.5	meshNode0	257	netdev://[00:00:00:00:00:0a]	InData	0	0	0	0
0.5	meshNode0	257	netdev://[00:00:00:00:00:0a]	OutData	4.8	4.86094	3	3
0.5	meshNode0	257	netdev://[00:00:00:00:00:0a]	InNacks	0	0	0	0
0.5	meshNode0	257	netdev://[00:00:00:00:00:0a]	OutNacks	0	0	0	0
0.5	meshNode0	257	netdev://[00:00:00:00:00:0a]	InSatisfiedInterests	4.8	0	0	0
0.5	meshNode0	257	netdev://[00:00:00:00:00:0a]	InTimedOutInterests	0	0	0	0
0.5	meshNode0	257	netdev://[00:00:00:00:00:0a]	OutSatisfiedInterests	0	0	0	0
0.5	meshNode0	257	netdev://[00:00:00:00:00:0a]	OutTimedOutInterests	0	0	0	0
0.5	meshNode0	258	internal://	InInterests	0	0	0	0
0.5	meshNode0	258	internal://	OutInterests	0	0	0	0
0.5	meshNode0	258	internal://	InData	0	0	0	0
0.5	meshNode0	258	internal://	OutData	0	0	0	0
0.5	meshNode0	258	internal://	InNacks	0	0	0	0
0.5	meshNode0	258	internal://	OutNacks	0	0	0	0
0.5	meshNode0	258	internal://	InSatisfiedInterests	3.2	0	2	0
0.5	meshNode0	258	internal://	InTimedOutInterests	0	0	0	0
0.5	meshNode0	258	internal://	OutSatisfiedInterests	0	0	0	0