Multi-hop Device-to-Device Routing Protocols for Software-Defined Wireless Networks

by Mahrokh Abdollahi Lorestani

Thesis submitted in fulfilment of the requirements for the degree of

Doctor of Philosophy

under the supervision of A/Professor Mehran Abolhasan and Dr. Negin Shaiati

University of Technology Sydney
Faculty of Engineering and Information Technology

February 2022

Statement of Originality

I, Mahrokh Abdollahi Lorestani declare that this thesis, is submitted in fulfilment of the requirements for the award of the degree of doctor of philosophy, in the faculty of Engineering and Information Technology at the University of Technology Sydney. This thesis is wholly my own work unless otherwise referenced or acknowledged. In addition, I certify that all information sources and literature used are indicated in the thesis.

This document has not been submitted for qualifications at any other academic institution.

This research is supported by the Australian Government Research Training Program.

Signature:

Production Note:
Signature removed prior to publication.

Mahrokh Abdollahi Lorestani
October 16, 2021

ABSTRACT

Multi-hop device-to-device (MD2D) communications are an integral part of future wireless networks. Multi-hop communications enable mobile devices in close proximity to communicate directly or through multi-hop connections instead of traversing through a network infrastructure. This provides numerous benefits for cellular networks, such as low-cost communications, enhanced cellular coverage and capacity, reduced total power consumption in devices, and improved spectral efficiency. Consequently, service providers can leverage the advantages of both D2D and cellular networks to enhance the quality of their services. However, tight coupling of control and data functions in cellular equipment and the utilization of proprietary interfaces and protocols in existing cellular infrastructure make integration difficult and rigid. Hence, there is a need for open and reprogrammable frameworks to make the network more flexible and scalable. Software-defined networking (SDN) is a promising technology for future wireless networks that provides an open and reprogrammable framework wherein the control functions are taken from network devices and are logically centralized in a control entity. The open framework of SDN provides an opportunity for service providers to manage networks more intelligently and develop services in a more agile manner.

This thesis introduces an SDN-based framework for cellular networks, referred to as virtual ad hoc routing protocol framework (VARP), capable of developing different types of multi-hop routing protocols. In the proposed framework, an SDN controller determines the mode of communication for mobile devices (i.e., cellular or multi-hop modes). Two different multi-hop routing protocols are designed for the proposed framework: source-based virtual ad hoc routing protocol (VARP-S) and SDN-based multi-hop D2D routing protocol (SMDRP). In both protocols, a source of data packet sends a route request to the controller and receives the forwarding information from the controller in response. This thesis then presents a multi-protocol framework capable of developing multiple routing protocols under a single framework. In the proposed framework, an SDN controller logically divides a cell into multiple clusters based on its knowledge of the entire cell. The controller determines which multi-hop routing protocol can provide the best performance for each cluster. The simulation results show that the proposed multi-protocol framework provides better performance than traditional single-protocol architectures. Finally, the thesis presents a novel software-defined adaptive routing algorithm for multi-hop multi-frequency communications in wireless multi-hop mesh networks. The simulation results indicate that the proposed algorithm improves the end-to-end throughput of multi-hop connections by considering the sur-

rounding WiFi traffic and adaptive selection of frequencies and routes.

ACKNOWLEDGEMENTS

This thesis would not have been fulfilled without the supervision and guidance of many individuals who contributed and extended their valuable assistance in the preparation and completion of this study.

At first, I offer my sincerest gratitude to my principal supervisor, A/Professor. Mehran Abolhasan, who has supported me throughout my thesis with his patience and knowledge. I attribute the level of my Ph.D. degree to his encouragement and effort, and without his support, this thesis would not have been completed.

I am truly thankful and grateful to my co-supervisors, Dr. Negin Shaiati and A/Professor. Justin Lipman, for their unconditional support and valuable guidance during this thesis. It is an honour for me to work with them during my studies. They were always available and willing to assist. They have not been only a great advisors, but also an encouraging and motivating friends.

My sincere thanks also goes to Dr. Wei Ni for all his support and guidance during this thesis.

Last, but not the least, I would like to thank my family and all my friends for providing superb support and friendship that I needed all these years.

To my lovely parents, Susan and Masoud

List of Publications

Journal Papers

J-1. M. Abolhasan, M. Abdollahi, W. Ni, A. Jamalipour, N. Shariati and J. Lipman, "A Routing Framework for Offloading Traffic From Cellular Networks to SDN-Based Multi-Hop Device-to-Device Networks," in IEEE Transactions on Network and Service Management, vol. 15, no. 4, pp. 1516-1531, Dec. 2018, doi: 10.1109/TNSM.2018.2875696.

J-2. "Dynamic Routing Protocol Selection in Multi-hop Device-to-Device Wireless Networks" submitted to IEEE Transactions on Vehicular Technology [under review].

J-3. M. Abdollahilorestani, W. Ni, M. Abolhasan and S. Li, "Software-defined Networking-based Adaptive Routing for Multi-hop Multi-frequency Wireless Mesh," in IEEE Transactions on Vehicular Technology, doi: 10.1109/TVT.2021.3116266.

Conference Papers

C-1. M. Abdollahi, M. Abolhasan, N. Shariati, J. Lipman, A. Jamalipour and W. Ni, "A Routing Protocol for SDN-based Multi-hop D2D Communications," 2019 16th IEEE Annual Consumer Communications & Networking Conference (CCNC), 2019, pp. 1-4, doi: 10.1109/C-CNC.2019.8651752.

Contents

1 Introduction **1**
1.1 Background . . . 1
1.2 Thesis Contributions . . . 3
1.3 Research Questions and Objectives . . . 4
1.3.1 Research Questions . . . 4
1.3.2 Research Objectives . . . 4
1.4 Thesis Organization . . . 5

2 Literature Survey **7**
2.1 Introduction . . . 7
2.2 Cellular Networks . . . 7
2.2.1 Brief Overview of Proposed Technologies and Modulation Techniques for 1G, 2G, 3G, and 4G Systems . . . 8
2.2.2 Brief Overview of Main Components of the 2G, 3G, and 4G Systems . . . 9
2.2.3 3GPP Standard . . . 12
2.2.4 Cellular Networks Limitations . . . 12
2.2.5 The Promise of 5G . . . 13
2.3 Multi-hop Routing in Wireless Networks . . . 13
2.3.1 Mobile Ad-hoc Networks (MANETs) . . . 14
2.3.2 Wireless Mesh Networks (WMN) . . . 16
2.3.3 Integration of MANETs with Cellular Networks . . . 19
2.4 Software Defined Networking . . . 25
2.4.1 SDN Architecture . . . 25
2.4.2 Brief Overview of OpenFlow Protocol . . . 27
2.4.3 SDN Platforms and Tools . . . 28
2.4.4 Integrating SDN with Wireless Networks . . . 29
2.4.5 Challenges of SDN-based Multi-hop Wireless Networks . . . 33
2.5 Summary . . . 34

3 A Routing Framework for Offloading Traffic from Cellular Networks to SDN-based Multi-Hop Device-to-Device Networks **36**

3.1 Introduction 36
3.2 Proposed Virtual Ad hoc Routing Framework 39
3.2.1 VARP Architecture 39
3.2.2 Route Quality Assessment 42
3.3 Routing Protocol for VARP Framework 43
3.3.1 Topology Discovery 43
3.3.2 Route Discovery 44
3.3.3 Route Maintenance 46
3.4 HSAW Overview 46
3.5 Scenarios and Summary of Notations 47
3.5.1 Notations 47
3.5.2 Assumptions 47
3.6 Theoretical Analysis in Static Networks 49
3.6.1 HSAW Protocol 50
3.6.2 VARP-S Protocol 50
3.6.3 Result Analysis 51
3.7 Theoretical Analysis in Mobile Networks 52
3.7.1 HSAW Protocol 53
3.7.2 VARP-S Protocol 55
3.7.3 Result Analysis 55
3.8 Conclusions 62

4 A Routing Protocol for SDN-based Multi-hop D2D Communications 63
4.1 Introduction 63
4.2 Proposed SDN-based 5G Framework for SMDRP 65
4.3 SMDRP for SDN-based 5G Networks 65
4.3.1 Route Discovery in SMDRP 66
4.3.2 Route Maintenance 67
4.4 HSAW for SDN-based 5G Networks 68
4.5 Overhead Analysis 69
4.5.1 Mobility 69
4.5.2 Hello Message Overhead 70
4.5.3 Topology-update Overhead 70
4.5.4 Route Discovery Overhead 71
4.5.5 Route Maintenance Overhead 72
4.5.6 Data Packet Delivery Overhead 72
4.5.7 Total Cellular Overhead 73
4.5.8 Total WiFi Overhead 73
4.5.9 Result Analysis 76
4.6 Memory Analysis 77
4.7 Conclusion 80

5 Dynamic Routing Protocol Selection in Multi-hop Device-to-Device Wireless Networks 81
5.1 Introduction 81
5.2 Related Work 83
5.3 Proposed multi-protocol SDN-based framework for cellular networks 86
5.4 DECISION-MAKING PROCEDURE FOR MULTI-PROTOCOL FRAMEWORK 89
5.4.1 Overview of AHP-based Decision-making Procedure 89
5.4.2 AHP-based Decision-making Model for Multi-protocol Framework . . . 90
5.5 System Model 91
5.5.1 Cellular Channel Model 91
5.5.2 Energy Consumption Model 92
5.5.3 Network Model 93
5.6 Simulation Results and Performance Analysis 94
5.6.1 Simulation Analysis of HSAW and VARP-S routing protocols 95
5.6.2 Routing Protocol Decision-making for Each Cluster 97
5.6.3 Performance Analysis of Multi-protocol Framework vs. Single-protocol Framework 97
5.7 Conclusion 101

6 Software-defined Networking-based Adaptive Routing for Multi-hop Multi-frequency Wireless Mesh 102
6.1 Introduction 102
6.2 Related Work 103
6.3 Proposed Architecture for Designed Platform 106
6.3.1 System Architecture 107
6.3.2 Proposed SDN-based Adaptive Routing Algorithm 109
6.4 MATLAB-based Simulation Study 117
6.4.1 Link Capacity 117
6.4.2 Probability of the Successful Transmission of the Nodes 117
6.5 Performance Analysis using Mininet-WiFi 119
6.6 Conclusion 122

7 Conclusions and Recommendations for Future Work 123
7.1 Future Work 125

Bibliography 126

List of Figures

2.1 GSM system. 9
2.2 3G system. 10
2.3 LTE system. 11
2.4 SDN architecture. 26
2.5 The logical structure of the designed mesh router by [1]. 31

3.1 Detailed illustration of the VARP framework for the proposed VARP-S routing protocol. 39
3.2 (a) SDN architecture in VARP framework, (b) VARP sub layers. 40
3.3 Source node responsibilities for route discovery and route maintenance in VARP-S. 42
3.4 Sub-controller responsibilities for route discovery and route maintenance in VARP-S. 42
3.5 Illustration of building three different SPTs for source node A based on the defined traffic types in Section 3.2.2. 44
3.6 Total routing overhead in the WiFi-band at time 0 vs. Number of nodes and active flows. 59
3.7 Total overhead in the cellular-band at time 0 vs. Number of nodes. 59
3.8 Total overhead in the cellular-band per time step ζ vs. Mobility. 59
3.9 Topology-update overhead in cellular-band per time step ζ vs. Mobility. 59
3.10 Misrouting overhead in cellular-band per time step ζ vs. Mobility. 59
3.11 Overhead analysis of O_{DRREP} compared to O_A. 59
3.12 Total routing overhead in the cellular-band per second vs. Number of active flows. 60
3.13 Total overhead in the cellular-band per time step ζ vs. Cell area. 60
3.14 Topology-updates overhead and misrouting overhead in the cellular-band per time step ζ vs. Cell area. 60
3.15 The consumed energy of one node vs. Number of nodes, where the network is static. 60
3.16 The consumed energy of a source node vs. Number of nodes, where the network is mobile. 60
3.17 The consumed energy of a source node vs. Mobility, where the network is mobile. 60

4.1 Proposed framework for an SDN-based 5G network. 64

4.2 Route discovery in HSAW (flow A-E). 74
4.3 Route maintenance in HSAW. 74
4.4 Route discovery in SMDRP (flow A-E). 75
4.5 Steps 1-3 of the route maintenance in SMDRP (link C-D is broken). 75
4.6 Steps 4-6 of the route maintenance in SMDRP. 75
4.7 Total routing overhead in WiFi-band at time 0 vs. Number of nodes. 78
4.8 Total routing overhead in Cellular-band at time 0 vs. Number of nodes. 78
4.9 Total routing overhead in WiFi-band per time step ζ vs. Mobility. 78
4.10 Total routing overhead in the cellular-band per time step ζ vs. Mobility. 78
4.11 Topology-updates overhead in the cellular-band per time step ζ vs. Mobility. . . . 78
4.12 Route Maintenance overhead in the cellular-band per time step ζ vs. Mobility. . . 78
4.13 Comparing O_{FERR} of both protocols. 79
4.14 Comparing O_{FREP} with O_{FERR} in SMDRP. 79
4.15 Total routing overhead in cellular-band per time step ζ vs. Number of nodes. . . . 79
4.16 Memory usage of a mobile node vs. Number of nodes. 79
4.17 Memory usage of a mobile node vs. Number of flows in which the mobile node is involved. 79

5.1 Multi-protocol framework for SDN-based cellular networks. 87
5.2 Multi-protocol framework flowchart. 88
5.3 AHP-based decision-making model for multi-protocol framework. 90
5.4 2D representation of clusters. 93
5.5 Total routing overhead in cellular-band (bits/s). 98
5.6 End-to-end delay (sec). 98
5.7 Energy consumption of nodes (%). 98
5.8 Packet Loss. 98
5.9 Packet loss over a densely populated network. 99
5.10 Performance analysis of multi-protocol framework vs. single-protocol framework in terms of packet loss. 99
5.11 AHP decision-making process to select the most suitable routing protocol for each cluster. 100

6.1 System architecture. 107
6.2 Physical view of designed hardware platform. 108
6.3 Logical view of the designed platform. 108
6.4 Functionality of the proposed SDN-based adaptive routing algorithm. 116
6.5 MATLAB simulation result of the selected route by the E-Dijkstra's algorithm in the presence of spatially varying interference within a square area of side 1,000 meters. 118
6.6 MATLAB simulation result of the selected route by the E-Dijkstra's algorithm in the presence of spatially varying interference within a square area of side 500 meters. 118

6.7 Mininet-WiFi simulation result when there are 30 interference nodes 121
6.8 TCP/UDP throughput VS. background WiFi traffic density 121

LIST OF TABLES

3.1 List of notations for network abstraction. 48
3.2 List of notations for routing overhead analysis when network is static. 48
3.3 List of notations for routing overhead analysis when the network is mobile. . . . 53
3.4 Energy consumption analysis. 62

4.1 Summary of notations. 69
4.2 Memory analysis of mobile nodes. 77

5.1 Simulation parameters. 94
5.2 Suitable protocol for energy-independent and delay-sensitive clusters based on the AHP decision-making. 101

6.1 Comparison of multi-radio multi-hop hardware platforms. 105
6.2 List of notations for network abstraction 110
6.3 List of notations for Bianchi's model 110
6.4 Forwarding instructions for flow 2-7 116

LIST OF ABBREVIATIONS

1G	First Generation
1xEVDO	1x Evolution Data Optimized
1xRTT	1x Radio Transmission Technology
2G	Second Generation
3GPP	3rd Generation Partnership Project
4G	Fourth Generation
5G	Fifth Generation
AAA	Authentication, Authorization, and Accounting
AHP	Analytic Hierarchy Process
ALARM	A Location-Aware Routing Metric
AMPS	Advanced Mobile Phone System
AODV-ST	Ad-hoc On-demand Distance Vector Spanning Tree
AP	Access Point
ARIB	Association of Radio Industries and Businesses
ARS	Ad hoc Relaying Stations
ATIS	Alliance for Telecommunications Industry Solutions
ATSSS	Access Traffic Steering, Switching and Splitting
AUC	Authentication Centre
BATMAN	Better Approach To Mobile Adhoc Networking
BS	Base Station
BSS	Base Station Subsystem
BTS	Base Transceiver Station
CCSA	China Communications Standards Association
CDMA	Code Division Multiple Access
CI	Cellular Indicator
CN	Core Network
COTS	Commercial-Off-The-Shelf
CR-Id	Controller ID
CS	Circuit-switched
CSMA/CA	Carrier Sense Multiple Access with Collision Avoidance
D2D	Device to Device

DCF	Distributed Coordination Function
DIFS	Distributed Inter Frame Space
DRERR	Data Route Error
DRR	Data Route Record
DRREP	Data Route Reply
DRREQ	Data Route Request
DRREQI	DRREQ Intervals
DRREQT	Data Route Request Table
E2E	End-to-End
EDGE	Enhanced Data rates for GSM Evolution
EIR	Equipment Information Register
eNodeB	E-UTRAN Node B
EPC	Evolved Packet Core
ePDG	evolved Packet Data Gateway
EPS	Evolved Packet System
ETSI	European Telecommunications Standards Institute
ETT	Expected Transmission Time
ETX	Expected Transmission Count
E-UTRA	Evolved Universal Terrestrial Radio Access
E-UTRAN	Evolved Universal Mobile Telecommunications System Terrestrial Radio Access Network
EWT	Error Waiting Time
ExOR	Extremely Opportunistic Routing
FDMA	Frequency Division Multiple Access
FERR	Flow Error
FERR-WI	Flow Error Waiting Interval
FI-T	Flow Invalid Time period
FREQ	Flow Request
FREQT	Flow Request Table
FT	Flow Table
G-CE	Global Control Entities
GGSN	Gateway GPRS Support Node
GMSC	Gateway Mobile Switching Centre
GPRS	General Packet Radio Service
GSM	Global System for Mobile communication
HLR	Home Location Register
HRFA	Hybrid Routing Forwarding Algorithm
HSAW	Hybrid SDN Architecture for Wireless distributed networks
HSS	Home Subscriber Server
HWMP	Hybrid Wireless Mesh Protocol
I^2C	Inter-Integrated-Circuit

I^{2}S	Inter-Integrated-Chip Sound
IB	Information Base
IBSS	Independent Basic Service Set
IEEE	Institute of Electrical and Electronic Engineers
IMEI	International Mobile Equipment Identity
IoT	Internet of Things
L-CE	Local Control Entities
LE-WARP	Linux Enriched Wireless Open Access Research Platform
LLDP	Link Layer Discovery Protocol
LOLS	Localized On-demand Link State
LQSR	Link Quality Source Routing
LSDB	Link State Database
LTE	Long Term Evolution
LTE-A	LTE Advanced
LVAP	Light Virtual Access Point
M2M	Machine-to-Machine
MAC	Media Access Control
MADF	Mobile-Assisted Data Forwarding
MANETs	Mobile Ad-hoc Networks
MC	Mission Critical
MCDM	Multiple-Criteria Decision-Making
MD2D	Multi-hop Device-to-Device
MEC	Mobile Edge Computing
MeshDV	Mesh Distance Vector
MGW	Media Gateway
MIC	Metric of Interference and Channel-Switching
MIMO	Multiple-Input Multiple-Output
ML	Minimum Loss
MME	Mobility Management Entity
MMS	Multimedia Message Services
mmWave	millimeter Wave
MS	Mobile Stations
MSC	Mobile Switching Centre
N_hop	Next Hop
NC	Non-Critical
NFV	Network Functions virtualization
NMS	Network Management Server
NPT	Node Power Threshold
NSS	Network Subsystem
NT	Neighbour Table
ODL	Open Daylight

OFDMA	Orthogonal Frequency Division Multiple Access
OLSR	Optimized Link State Routing Protocol
ONF	Open Networking Foundation
P_hop	Previous Hop
PCEF	Policy Control Enforcement Function
PCRF	Policy and Charging Rules Function
PDN	Packet Data Networks
P-GW	Packet Data Network Gateway
ProSe	Proximity Service
PS	Packet-switched
PSK	Phase Shift Keying
PSTN	Public Switched Telephone Network
QoE	Quality of Experience
QoS	Quality of service
QPSK	Quadrature Phase-Shift Keying
RAN	Radio Access Network
RA-OLSR	Radio-Aware Optimized Link State Routing
REST	Representational state Transfer
RG	Relay Gateways
RMS	Routing Method Selection
RNC	Radio Network Controller
ROMER	Resilient Opportunistic Mesh Routing
RPS	Routing Protocol Selection
RQA	Route Quality Assessment
RQAMS	Route Quality Assessment & Metric Selection
RSRP	Reference Signal Received Power
RSSI	Received Signal Strength Indication
SAE	System Architecture Evolution
SDK	Software Development Kit
SDN	Software defined networking
SDR	Software Defined Radio
SGSN	Serving GPRS Support Node
S-GW	Serving Gateway
SMDRP	SDN-based Multi-hop D2D Routing Protocol
SMS	Short Message Services
SNR	Signal to Noise Ratio
SOPRANO	Self-Organizing Packet Radio Ad hoc Network with Overlay
SPF	Shortest Path First
SPI	Serial Peripheral Interface
SPT	Shortest Path Tree
SVD	Singular Value Decomposition

TC	Topology Control
TCREQ	Topology Control Request
TDMA	Time Division Multiple Access
TSDSI	Telecommunications Standards Development Society
TTA	Telecommunications Technology Association
TTC	Telecommunication Technology Committee
UAV	Unmanned Aerial Vehicle
UCAN	Unified Cellular and Ad hoc Networks
UE	User Equipment
UMTS	Universal Mobile Telecommunications System
UnF	Unreachable Flag
URI	Uniform Resource Identifier
VANET	Vehicular Ad hoc Network
VARP	Virtual Ad hoc Routing Protocol
VARP	Virtual Ad hoc Routing Protocol
VARP-S	Virtual Adhoc Routing Protocol- Source-based
VLR	Visitor Location Register
VM	Virtual Machine
VN	Validity Number
VoIP	Voice-Over-IP
WARP	Wireless Open-Access Research Platform
WCDMA	Wideband Code Division Multiple Access
WCETT	Weighted Cumulative ETT
WiFi	Wireless Fidelity
WiMAX	Wireless interoperability for Microwave Access
WLAN	Wireless Local Area Networks
WMN	Wireless Mesh Networks
WMSC	WCDMA Mobile Switching Centre
WNS	Wireless Sensor Networks
WPR	Wireless-mesh-network Proactive Routing
WTPs	Wireless Termination Points
WWAN	Wireless Wide Area Networks

"Luck is great, but most of life is hard work."

- Iain Duncan Smith

1 Introduction

1.1 Background

The world is experiencing a significant growth of Internet users who demand anytime anywhere access to the Internet. The Cisco annual Internet report predicts that by 2023 "the cellular speeds will be more than triple, the machine-to-machine (M2M) applications will be the fastest-growing applications, and over 70 percent of the population will have mobile connectivity" [2]. There is a need to design scalable networks to enable a highly connected world and provide consumers with high-speed and high-quality access to information and multimedia-based applications. Hence, while only a few years have passed since the introduction of 4th generation LTE networks, the 5th generation mobile networks (5G) have already begun to be deployed to meet mobile networks' current and future demands. 5G integrates a number of new technologies, such as multiple-input multiple-output (MIMO), millimeter-wave (mmWave) communications, software-defined networking (SDN), network function virtualization (NFV), and device-to-device (D2D), to provide high levels of capacity and flexibility. Among the proposed technologies, D2D communications and the SDN paradigm are the main focus of this thesis.

Device-to-device communications in cellular networks enable wireless devices in close proximity to communicate directly instead of traversing through an LTE core network. This allows offloading traffic from cellular (when possible), utilizing the licensed and unlicensed spectrum more efficiently, reducing the energy consumption and the end-to-end latency. Later, the idea of multi-hop D2D (MD2D) was introduced to extend cellular coverage and be used in public safety applications and similar cases where cellular infrastructure is not available. Mobile ad-hoc networks (MANETs) are a type of MD2D that are self-configured mobile devices that form a connected network with no fixed infrastructure. Despite providing flexibility, MANETs have limited scalability owing to their distributed network management. Consequently, integrating MANETs with cellular networks allows service providers to take the best features of both networks to increase the cellular capacity and address the scalability issues in MANETs. Further, new scalable multi-hop routing protocols should be designed to make the integration between multi-hop routing and cellular networks more efficient. Regardless of the proposed solutions by different studies, multi-hop communications under cellular networks face several challenges, such as complexity in routing, difficulty in resource and interference management, and efficient switching between the communication modes (cellular or multi-hop modes). Further, in the current cellular infrastructure, hardware and interfaces are hard-coded and vendor-specific, which makes the practical development of new services and protocols slow, challenging, and complex. The recent introduction of software-defined networks address these challenges by providing an open and reprogrammable framework that allows future cellular networks such as 5G to be effectively managed and extended via software.

The control functions are taken from the hardware in the SDN framework and logically centralized in one or more control entities, referred to as the controller. Open APIs are defined to allow the controller to communicate with the forwarding network devices and applications that run on top of the controllers. Forwarding devices are inexpensive and simple devices that can receive forwarding instructions from the controller. Significant research has been conducted on SDN-based 5G cellular networks that integrate SDN with other emerging technologies for 5G networks, such as massive MIMO, millimeter waves, network function virtualization, and device-to-device communications. However, until now, little attention has been paid to the design of routing protocols integrated with SDN-based cellular networks to enable multi-hop routing under the control of future SDN-enabled wireless base stations. The focus of this study is to design scalable multi-hop routing protocols for SDN-based cellular networks. To achieve this, an SDN-based cellular framework is proposed, and two multi-hop routing protocols are designed, namely VARP-S and SMDRP, which are explained in detail in Chapters 3 and 4. Furthermore, considering the fact that the performance of multi-hop protocols is affected by various network conditions (such as network size, node mobility, traffic pattern, environmental conditions, node capabilities, and network complexity), and because there is no single routing protocol that is suitable for every possible scenario, a multi-protocol framework is proposed that enables the development of multiple multi-hop routing protocols under a single framework. In the framework, a controller clusters a cell into multiple clusters and then selects the most suitable protocol based on the cluster conditions. Thus, the total network performance increases significantly compared to the traditional single-protocol frameworks. The details of the framework are presented in Chapter 5.

In addition to the aforementioned frameworks, a software-defined networking-based adaptive routing algorithm is designed for multi-hop multi-frequency wireless mesh networks to increase the end-to-end throughput in multi-hop communications. A novel metric is defined that considers the link capacity, frequency coloring, and probability of successful transmission of the nodes. Dijkstra's algorithm is extended to support the proposed multi-hop multi-frequency platform by selecting non-overlapping frequency bands together with routing paths. The proposed algorithm improves the end-to-end throughput capacity of multi-hop connections by considering background WiFi traffic. The operation of the proposed algorithm is described in detail in Chapter 6.

1.2 Thesis Contributions

The contributions of this thesis are summarized as follows:

- A new SDN-based framework is proposed that introduces the idea of multi-hop routing as an extended service for cellular networks to offload cellular traffic efficiently.
- A source-based multi-hop routing protocol is designed, VARP-S, which integrates multi-hop routing with a cellular network using the SDN paradigm.
- A hop-by-hop routing protocol, referred to as SMDRP, is designed for SDN-based cellular base stations. This protocol is suitable for a large density of mobile pedestrian communications sharing or exchanging content.
- A theoretical analysis is performed to investigate the performance of VARP-S and SMDRP in comparison with the proposed SDN-based routing approach in [3], known as HSAW. Unlike VARP-S and SMDRP wherein mobile nodes learn routes reactively, in HSAW, routes are learned proactively in the network, and the SDN controller provides the mobile nodes with a link-state database of the entire network. The results show better performance for the proposed protocols compared to HSAW in terms of routing overhead, energy consumption, and memory usage of mobile nodes in the network.
- A multi-protocol framework is developed for mobile wireless networks, introducing the idea of clustering a cell and deploying the most efficient routing protocol for each cluster. Compared to single-framework architectures, the multi-protocol framework can employ multiple protocols under a single framework by logically dividing the cell into a number of clusters under varying conditions (such as node density and mobility speed). Through this strategy, the most suitable routing protocol can be deployed or operated to achieve the highest throughput in each cluster. The multiple-criteria decision-making (MCDM) approach based on the analytic hierarchy process (AHP) is extended to choose the most suitable routing protocol for each cluster in the multi-protocol framework. Decision-making is based on criteria such as end-to-end delay, packet loss, cellular overhead, and energy consumption of the nodes. A simulation study is conducted to investigate the performance of the proposed multi-protocol framework compared to the single-protocol framework.
- Dijkstra's algorithm is extended, referred to as E-Dijkstra, to support the new multi-hop multi-frequency hardware platform developed by CSIRO. The algorithm finds the least-cost paths for various possible channel arrangements of multi-interface nodes. Furthermore, a novel metric is designed to consider the link capacity and traffic density of the data modules in each node. Based on this metric, separate cost matrices are created for each data module. E-Dijkstra's algorithm assesses the cost matrices for each active flow and selects a path

with the maximum end-to-end capacity and alternating frequencies. A simulation study is conducted using Mininet-WiFi to investigate the performance of the proposed algorithm in multi-hop multi-radio wireless networks.

1.3 Research Questions and Objectives

This section presents the research questions and objectives of this thesis.

1.3.1 Research Questions

Research questions of this study are summarized as follows:

1. How can an SDN-based cellular framework be designed and implemented, wherein mobile users act as forwarding elements receiving instructions from the controller?
2. What are the advantages and challenges of integrating multi-hop routing and the SDN paradigm into cellular networks?
3. What type of multi-hop routing protocols can be designed to efficiently offload cellular network traffic under the control of an SDN controller?
4. How to program mobile nodes in cellular networks, and to what extent should the controller be engaged in multi-hop routing?
5. Is it possible to run multiple multi-hop routing protocols under one cellular framework, while different parts of a cell may experience various network conditions (such as node density, the existence of barriers and obstacles in some areas that lead to weak signals, and mobility rate of mobile nodes)?
6. What metrics can be designed to select multi-hop routes with minimum interference and maximum end-to-end throughput in highly dense networks?

1.3.2 Research Objectives

The main objective of this thesis is to explore the potential benefits of multi-hopping capabilities for D2D communication scenarios in SDN-enabled wireless networks and to develop scalable multi-hop routing approaches for future SDN-based cellular networks. To achieve this, the SDN paradigm and multi-hop communications need to be integrated into the cellular network. The second objective of this study is to develop a multi-protocol framework that allows the development of multiple routing protocols under a single framework. This will increase the total performance of the network as each protocol can be used in the best-suited environment.

The objectives of the research are summarized as follows:

1. To design and implement an SDN-based cellular framework wherein mobile users act as forwarding elements and receive instructions from the controller.

2. To design scalable multi-hop routing protocols for the SDN-based cellular framework and investigate their performance in terms of energy consumption, memory usage, and routing overhead.

3. To design a multi-protocol framework wherein multiple multi-hop protocols can run under a single SDN-based cellular framework.

4. To design a software-defined networking-based adaptive routing algorithm for multi-hop multi-frequency wireless networks to minimize interference and increase end-to-end throughput.

1.4 Thesis Organization

This thesis is organised as follows:

- *Chapter 2*: This chapter presents a survey of different cellular evolution and standards, 5G solutions and technologies (D2D and SDN in particular). Further, various multi-hop wireless networks, their current challenges, and proposed solutions to address these challenges are discussed. Finally, the SDN concept is introduced, and different studies on SDN-based wireless networks are presented.

- *Chapter 3*: This chapter presents an SDN-based cellular framework, named VARP, capable of developing different multi-hop routing protocols. Furthermore, a source-based routing protocol, referred to VARP-S, is proposed for the framework, and its performance is analyzed in terms of energy consumption and routing overhead, compared to the previously designed SDN-based routing approach, known as HSAW [3]. This chapter is published in "IEEE Transactions on Network and Service Management".

- *Chapter 4*: This chapter introduces a hop-by-hop routing protocol, referred to SMDRP, for SDN-based cellular networks. A theoretical study is conducted to compare the performance of SMDRP with the proposed routing approach in [3] for a large density of mobile pedestrian communications. This chapter is published in "2019 16th IEEE Annual Consumer Communications & Networking Conference (CCNC)".

- *Chapter 5*: This chapter describes a multi-protocol framework wherein multiple multi-hop routing protocols can run under a single framework to improve the total network performance. To investigate the performance of the framework, four clusters with various network conditions (such as node density, number of flows, and mobility rate) are developed. The previously designed multi-hop protocols (HSAW and VARP-S) are selected as routing protocols that can run on the framework. A simulation study is conducted to investigate the performance of the protocols for each cluster. Based on the obtained simulation results, the AHP MCDM is developed to select the most suitable routing protocol for each cluster. The total performance of the network is then evaluated when the mobile nodes of each cluster run the selected protocol. This chapter is submitted to "IEEE Transactions on Vehicular Technology" and is under review.

- *Chapter 6*: This chapter presents a new SDN-based routing algorithm on top of a lightweight hardware platform developed by CSIRO. Dijkstra's algorithm is extended to support the developed multi-hop multi-frequency platform, where non-overlapping frequency bands are selected together with the routing paths by maintaining N^2 Dijkstra processes for N frequency bands. These processes interact to recursively select the optimal upstream node and frequency for each downstream frequency of a node. Mininet-WiFi is used to evaluate the routing of the new system under dense network settings. The results indicate that the proposed system improves the end-to-end throughput by considering the background WiFi traffic and adaptively selecting routes and frequencies compared to the shortest-path-based routing strategy. This chapter is published in "IEEE Transactions on Vehicular Technology".
- *Chapter 7*: A summary of the thesis content, thesis contributions, and recommendations for future work are provided in the final chapter.

2

Literature Survey

2.1 Introduction

This chapter examines the evolution of cellular networks, the proposed technologies to enhance cellular capacity, and the limitations of current cellular networks. Multi-hop device-to-device communication is then studied as a potential technology for current and future cellular networks. Finally, the concept of SDN paradigm, its application in future wireless networks, and the role of SDN technology in the efficient integration of multi-hop communications with cellular networks are discussed.

2.2 Cellular Networks

A cellular network is a communication network of wireless mobile nodes that are distributed over cells. Each cell is a land area managed by a base station. The BS is a fixed-location transceiver that transfers data and voice between mobile users in its coverage area. The BS also connects mobile users within a cell to the Internet or any other external networks.

The cellular network underwent several technological evolutions, which resulted in the acquisition of different standards, speeds, and frequencies. The evolution of mobile cellular networks is differentiated by word generation and is known as 1G, 2G, 3G, and 4G. Currently, we are in the fourth generation of cellular networks, and the fifth generation is emerging.

2.2.1 Brief Overview of Proposed Technologies and Modulation Techniques for 1G, 2G, 3G, and 4G Systems

The first generation (1G) of mobile cellular networks is based on the advanced mobile phone system (AMPS). AMPS is a standard system for analog signal cellular telephone services in the United States. It utilizes the frequency division multiple access (FDMA) technique to allocate one or more frequency bands or channels to each mobile user. 1G is limited only to voice services, and supports a channel capacity of 30 kHz with a frequency band of 824-894 MHz and a speed up to 2.4 kbps [4].

To support additional services, such as multimedia message services (MMS), a second generation (2G) was introduced. Unlike 1G, which uses analog telecommunications standards, in 2G, digital signals are used to transmit voice at a speed of 64 kps. 2G utilizes digital modulation schemes, such as time-division multiple access (TDMA) and code division multiple access (CDMA). The TDMA divides the signal into time slots, and each user transmits in its own time slot. CDMA allows users to communicate over a multiplex physical channel by providing each user with a special code. The global system for mobile communication (GSM), based on TDMA, is the most widely used 2G technology that can support international roaming. The bandwidth supported by 2G is 30-200 kHz [4].

Continuous enhancement of the GSM led to the development of a 2.5G system. While 2G used only circuit-switched data, packet-switched data was introduced by 2.5G technology. 2.5G employs a general packet radio service (GPRS), which is a packet-switching communication protocol. GPRS supports higher data rates by allocating multiple time slots to each user using coding schemes. Further, it allows service providers to charge their subscribers based on the amount of data transmitted by their mobile phones, not the duration of their connection. The supported data rate by 2.5G is 144 kbps [4].

The increasing need to have higher data rates and to provide mobile phones with multimedia-based applications and additional services (for example, web browsing, fax, TV/videos, video conferencing, paging, e-mail, and navigational maps) resulted in the birth of third-generation mobile systems introduced by the 3rd Generation Partnership Project (3GPP) in 2000. 3G is capable of supporting high-speed data rates ranging from 144 kbps to more than 2 Mbps. Universal mobile telecommunications system (UMTS), enhanced data rates for GSM evolution (EDGE), and cdma2000 1x radio transmission technology (1xRTT) are among the first proposed 3G systems. Unlike GPRS in 2.5G, which transmits one bit per symbol, EDGE transmits 3 bits per symbol using an 8-PSK (phase shift keying) modulation scheme, which improves data rates in 3G by three times higher than GPRS. While UMTS employs the same core network as other 3G systems (i.e., GSM, GPRS, and EDGE), UMTS radio access network (RAN) is completely different. The GSM, GPRS, and EDGE systems use 200 kHz carries with eight time slots per frame, whereas UMTS utilizes the wideband code division multiple access (WCDMA) standard on a 5 MHz carries. However, it provides almost the same download rate as EDGE, up to 384kbps. From the cdmaOne family, cdma2000 1xRTT was defined by IS-2000 that provides downlink rates of up to 153.6 kbps by employing modulation techniques, such as variable-length orthogonal codes and quadrature phase shift keying (QPSK). The later evolution of the cdmaOne family, known as 1x evolution-data optimized (1xEVDO), supports a peak downlink rate of 2.4 Mbps [4, 5].

The necessity of having a general platform for all the proposed cellular technologies led to emergence of the fourth-generation (4G) mobile systems in the late 2000s. 4G was an evolution to mobile systems that proposed a full IP-based network. In 4G systems, mobile users experience higher quality, speed, capacity, and security, while incurring lower costs for access to services, such as the Internet, multimedia services, voice, and data services. The key feature of 4G networks is terminal mobility, which allows mobile users to switch between different wireless networks and access wireless services anytime and anywhere. Examples of 4G technologies are long-term evolution (LTE) and LTE-advanced (LTE-A). The supported speeds in the 4G networks were 100 Mbps and 1Gbps [4].

In addition to improvements made to 4G LTE networks, work on 5G and 6G has already begun to address the requirements of having a highly connected world wherein anytime and anywhere access to information, mobile high-speed video, and multimedia-based applications are becoming a prerequisite for consumers. 5G promises to increase the download speed to 20 Gbps and to reduce the delay of data delivery to less than 1ms, compared to 4G networks that support up to 1Gbps for download speed and delay of approximately 70 ms [6]. Further, 5G networks are going to provide services, such as Gbps data broadcasting, high mobility up to 500 km/h, software-based radios, high download and upload speeds. Notably, 5G has already started to be deployed and 6G is supposed to integrate 5G and satellite communications to provide global coverage [7, 8].

2.2.2 Brief Overview of Main Components of the 2G, 3G, and 4G Systems

This section briefly describes the main components of the 2G, 3G, and 4G cellular systems.

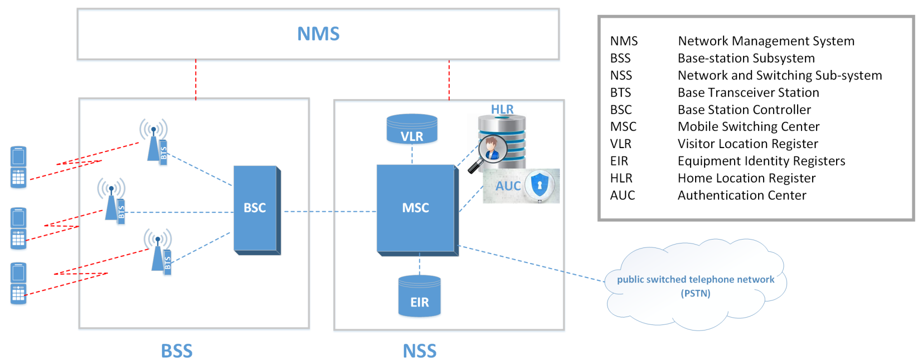


Figure 2.1: GSM system.

GSM System Architecture: GSM is the most used system among the proposed 2G mobile systems. There are three main components in GSM, as depicted in Figure 2.1 [9]:

- **Base station subsystem (BSS)**: It performs tasks like signal coding and decoding, handover, radio resource management, data encryption, and power control. Mobile stations (MSs) are connected to the base transceiver station (BTS) in the BSS via air interfaces.

- **Network subsystem (NSS)**: This lies between the GSM network and public networks. There are several elements in NSS, such as mobile switching center (MSC), home location register (HLR), visitor location register (VLR), authentication center (AUC), and equipment information register (EIR). MSCs are used to interconnect mobile users. All subscriber information (such as authorized services for the user, subscription type, and current subscriber location) is kept in the HLR. When a subscriber visits a coverage region, the VLR entity asks the HLR for the user information and keeps the information in its database. The VLR information is updated dynamically based on the region visited by the user. AUC is responsible for security policies, user authentication, and data encryption. EIR maintains the list of authorized numbers for mobile users to verify each mobile device ID, known as the international mobile equipment identity (IMEI).

- **Network management system(NMS)**: It is responsible for control and management functions, such as network monitoring, fault management, network measurement and development.

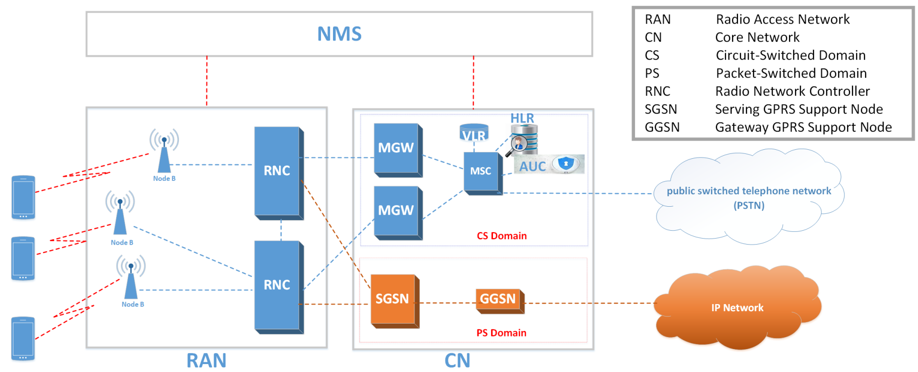


Figure 2.2: 3G system.

3G System Architecture: The system architecture in a 3G network has two main components, as illustrated in Figure 2.2: radio access network (RAN) and core network (CN) [9].

- **RAN:** This manages radio resources and telecommunications. The RAN consists of two main elements: the base station (BS), known as node B, and a radio network controller (RNC). Mobile stations are connected to the BS, and the RNC acts as an interface between the BS and the core network.

- **CN:** It has two main components: circuit-switched (CS) and packet-switched (PS) domains. CS is connected to the public switched telephone network (PSTN) and handles real-time traffic, while PS is connected to the public IP network and handles non-real-time traffic.

  - **CS domain components:** It consists of a WCDMA mobile switching center (WMSC)/VLR, HLR, media gateway (MGW), and gateway mobile switching center (GMSC).

  - **PS domain components:** This includes serving the GPRS support node (SGSN) and gateway GPRS support node (GGSN) elements. SGSN is used to connect the RAN to the PS domain and manage user mobility. GMSC and GGSN are used as interfaces between the 3G network and external CS and PS networks, respectively.

- **NMS**: NMS in 3G networks, in addition to the supported tasks in the 2G network, is also capable of managing the PS data as well as handling multi-vendor and multi-technology environments, such as 2G and 3G.

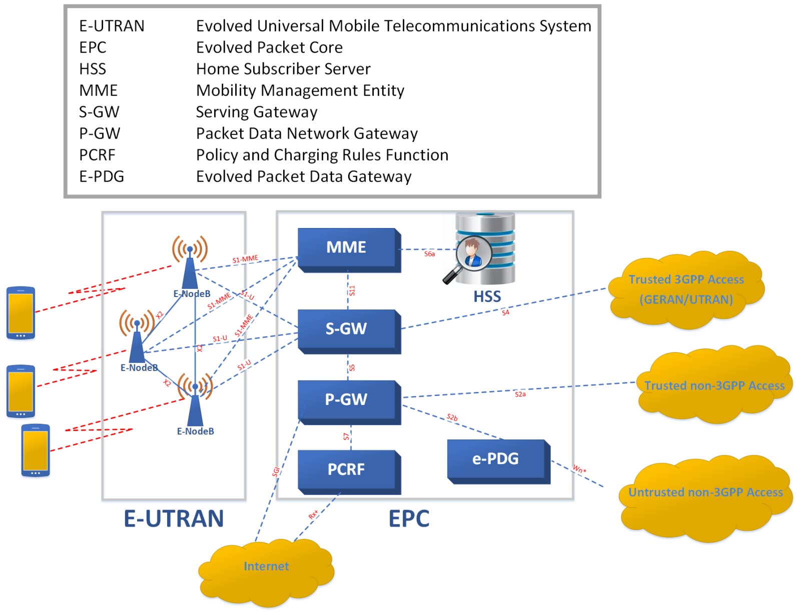


Figure 2.3: LTE system.

4G System Architecture: 3GPP 4G has a new architecture, referred to as system architecture evolution (SAE). LTE has two main components: an evolved packet core (EPC) as a core network and eNodeB as long-term evolution (LTE) RAN. The EPC includes only the packet-switched domain. Consequently, the circuit-switched voice is not supported by the new core network. However, voice over internet protocol (VoIP) technology is used to carry voice as data. eNodeB employs a new digital modulation scheme called orthogonal frequency division multiple access (OFDMA), which supports bandwidth and data rates up to 20MHz and 300 Mbps for download, respectively. LTE-advanced has been introduced to achieve a wider bandwidth. It aggregates multiple carriers and employs advanced antenna techniques for uplinks and downlinks. In LTE-advanced, five 20 MHz carriers are combined to support higher downlink data rates of up to 3000 Mbps [4, 7, 10].

LTE Architecture Components: LTE is an evolved packet system (EPS) that consists of two main entities, as indicated in Figure 2.3: E-UTRAN as a radio access network and EPC as an IP-based core network [11].

- **E-UTRAN:** It has only one component known as eNodeB that is in charge of all the radio resource control functions, without the assistance of any centralized controller.

- **EPC:** It consists of several components, as outlined below:

– **Mobility management entity (MME)**: the MME manages the UE location by tracking the UE and keeping up-to-date information of its current location.

– **Serving gateway (S-GW)**: the S-GW connects E-UTRAN to the EPC. Furthermore, it supports seamless intra-mobility within E-UTRAN.

– **Packet data network gateway (P-GW)**: the P-GW is responsible for IP allocation to UEs. Moreover, it connects UEs to external packet data networks (PDNs) and the Internet.

– **Policy and charging rules function (PCRF)**: the PCRF entity is a software component that has access to the subscriber database. PCRF controls the charging functions in PCEF.

– **Home subscriber server (HSS)**: the HSS maintains the subscriber information, such as subscriber data, a list of authorized external PDNs for the subscriber, its currently attached MME, and the subscriber QoS profile.

– **evolved packet data gateway (ePDG)**: the ePDG provides a secure connection between the EPC and non-3GPP networks.

2.2.3 3GPP Standard

In 1998, seven standardization organizations, including the European Telecommunications Standards Institute (ETSI), Association of Radio Industries and Businesses (ARIB), Telecommunications Technology Association (TTA), China Communications Standards Association (CCSA), Telecommunication Technology Committee (TTC), Telecommunications Standards Development Society (TSDSI), and Alliance for Telecommunications Industry Solutions (ATIS) formed a partnership project, known as the Third Generation Partnership Project (3GPP), to produce reports and unify the technical specifications for defining 3GPP technologies developed for mobile telecommunications. Initially the project scope was defining the global specification for the GSM core network in a 3G system, and then extended to later developed standards, such as LTE and LTE-advanced [11].

Technical documents provided by 3GPP are divided into releases. A new release includes new features that were added to the previous release. Release 17 was the latest release of 3GPP.

2.2.4 Cellular Networks Limitations

Current cellular networks have several constraints [12–14], such as 1) limited data rate and network capacity, 2) scalability issues arising from centralized architecture, as all the traffic is handled at the central gateway (P-GW entity) in the current architecture, even end-to-end traffic within the same cell, and 3) standardized and proprietary components and interfaces that make the development of new network services and functionalities time-consuming and costly, as it requires a long standardization process.

2.2.5 The Promise of 5G

Current wireless mobile networks are faced with ongoing challenges in responding to rapidly changing network requirements, such as exponential growth of mobile users and smart devices, increasing subscriber demand for higher data rates, and anytime anywhere access to multimedia applications and the Internet with high QoS and QoE. 5G promises to handle traffic with a milliseconds delay and provide 10 times longer battery life, 10-100x higher data rates, greater node density per sq km, and a peak download speed of 20 Gbps. Further, 5G promises to bind all network technologies into one network and also to improve the development of new technologies (such as IoT, virtual reality, and autonomous vehicles) [6, 15, 16]. To achieve this, intelligent resource management is required to support the increased network load and diversity of network technologies, applications, and smart devices in 5G networks.

Several emerging technologies have been proposed to be integrated with 5G networks, such as small cells, millimeter waves, massive MIMO, beam froming, full-duplex, device-to-device communications, Internet of Things (IoT), smart grids, big data, Internet of Vehicles (IoV), mobile cloud computing, software-defined networking (SDN), software-defined radio (SDR), and network function virtualization (NFV). The main focus of this thesis is to integrate multi-hop routing with the SDN paradigm in 5G and beyond 5G networks (i.e., 6G) [6].

D2D plays an important role in handling the fast growth of machine-to-machine applications and context-aware applications in 5G networks. D2D reduces the end-to-end delay by providing direct communication between peers rather than traversing the traffic through the cellular core network. Furthermore, D2D reduces the extreme overload on cellular networks by offloading traffic from the cellular to the D2D. It is also used to extend the cellular coverage of dead zones. More importantly, D2D is utilized for public safety applications where cellular infrastructure may not be available and fast network convergence is a necessity. Consequently, D2D communication is a part of 5G networks that enables developing new location-based applications and peer-to-peer services while providing benefits, such as improving throughput and spectrum reuse, extending cellular coverage, and reducing end-to-end delays and energy usage of mobile devices [17, 18].

2.3 Multi-hop Routing in Wireless Networks

In wireless networks, wireless nodes can communicate with each other in two different modes: infrastructure-based and infrastructure-less. In the former, traffic between nodes is coordinated by a fixed infrastructure, such as access point (AP) in wireless local area networks (independent basic service set (IBSS)) or BS in cellular networks. In the latter, wireless nodes communicate with each other without any infrastructure. The source of data packets forwards the packets to the destination via one or more intermediate nodes that form multi-hop device-to-device connections.

In multi-hop wireless networks, wireless nodes communicate via wireless media through one or more intermediate wireless nodes without the aid of any infrastructure. Wireless nodes can be stationary or mobile and are responsible for maintaining the network connected via self-configuring and self-optimizing methods in a distributed manner. Mobile ad hoc networks (MANETs), vehicular ad hoc networks (VANETs), wireless sensor networks (WNS), wireless mesh networks (WMN), and cellular networks are examples of multi-hop wireless networks [19].

There are two types of hop-by-hop communication: single-hop and multi-hop. In single-hop communication, the source of the data packet communicates directly with the destination node, whereas in multi-hop communication, the source node reaches the destination via relay nodes. Multi-hop communications employ multiple short single-hop links between the source and destination instead of having a direct long-distance link. Consequently, multi-hop communications provide less interference and a shorter reuse distance by reducing the transmission power [12].

Application of Device-to-Device Communications: First, the concept of device-to-device (D2D) communications was introduced for defense requirements. Then, D2D was proposed for specialized civilian applications, where a fixed infrastructure does not exist to connect devices and fast network deployment matters. Later, different studies were proposed to integrate D2D communications with different wireless technologies (e.g., cellular and WLAN) to extend their network coverage and capacity through multi-hop D2D communications.

2.3.1 Mobile Ad-hoc Networks (MANETs)

The MANET paradigm was defined as self-configured mobile devices connected to a wireless medium with no fixed infrastructure. MANETs employ multi-hop D2D communications to discover other nodes in the network and keep the network connected. In this regard, the MANET working group of the Internet Engineering Task Force (IETF) proposed several standardized routing protocols for ad hoc networks to enable mobile nodes to dynamically discover each other and maintain interconnectivity [20].

Over the past three decades, a significant amount of research has been conducted to develop scalable networking solutions for MANETs, leading to designing numerous routing protocols [21–24]. However, some fundamental challenges remain unsolved [25, 26]. One such challenge is scalability. Studies have shown that purely infrastructureless networks such as MANETs do not scale well when the overall network density is increased beyond a few hundred nodes. A wide range of routing protocols have been proposed to address scalability challenges at the network layer [27–30].

Classification of Routing Protocols in MANETS: MANET routing protocols can generally be classified into three groups: proactive, reactive, and hybrid [21, 31–34], as described in detail in the following paragraphs:

- **Proactive protocols:** These are the first protocols designed for MANETs [21, 35]. These protocols periodically maintain and update the routing structure of the network. Various proactive routing protocols have been proposed that use different methods and policies to collect and update routing information. They utilize a different number of tables to store routing-related information [23, 24]. However, proactive protocols have limited scalability as the network size increases.

- **Reactive protocols:** They were introduced to address the scalability problems of proactive protocols by allowing routes to be determined on demand [35–38]. They do not maintain the up-to-date topology of the entire network. They only keep the routing information for the active routes. When a route is desired, they flood a route query through the network and receive a route reply (from the destination node or other nodes that have a route to it) in response. While reactive protocols have shown better scalability as the number of nodes increases, they have limited scalability as the number of traffic flows increases.

- **Hybrid protocols:** They were introduced to integrate the best features of both proactive and reactive protocols to achieve high levels of scalability. The proposed hybrid routing protocols are zone-based, cluster-based, or tree-based. The zone-based protocols partition the network into multiple zones, where nodes inside the zone are connected proactively and have access to other nodes outside the zone reactively. In tree-based protocols, the network is defined as the number of trees connected through gateway nodes and form a forest. Each tree represents a zone, and each node can only be a member of a single zone. Studies have shown that hybrid protocols achieve higher scalability than purely proactive and reactive protocols [22, 39].

- **Comparing proactive, reactive, and hybrid protocols:** To sum up, the reactive protocols experience the least overhead. Proactive protocols have the highest overhead and the least scalability. Hybrid protocols are the most scalable routing protocols for MANETs. Proactive protocols produce lower delays and require more storage compared to reactive protocols [21, 23, 24]. In networks with higher mobility, proactive protocols might be more costly in terms of routing overhead compared to reactive protocols [35].

MANETs provide some benefits for wireless networks [12,19,40], such as cheap communications resulting from utilizing the unlicensed frequencies, and fast and flexible deployment because there is no need for costly and timely infrastructure development. Regardless of the benefits, MANETs face several challenges [12, 19, 19, 41–43], such as:

- Difficult interference management in multi-hop communications because of node mobility and frequent node failures.

- Distributed network management that leads to limited scalability when the network experiences dynamic topology changes.

- Limited wireless transmission range and limited power of mobile nodes.

- Dynamic network changes and lack of a structured network due to node mobility.

- High interference caused by simultaneous transmissions by devices in close proximity of each other.

- Needs for power control and power-aware media access control (MAC) protocols due to performance limitations of carrier sense multiple access with collision avoidance (CSMA/CA) access method in the IEEE 802.11 standard.

- Lack of a global network view due to distributed network management. This makes the optimal routing decision challenging.
- Limited network lifetime due to limited battery power of wireless devices for exchanging control and data traffic in the network.
- Difficulties in guaranteeing QoS due to dynamic network changes caused by mobility and unreliable wireless media.

It is noteworthy that in MANETs, the probability of route breakage increases in accordance with the length of the routing path. Consequently, “practical sizes of MANETs would range within around five hops” [35].

2.3.2 Wireless Mesh Networks (WMN)

WMN was introduced to address the scalability problems of MANETs and provide Internet access services and new applications [44–48]. IEEE 802.11s - now a part of the IEEE 802.11 standard - defines a new routing structure [49]. IEEE 802.11s provides multi-hop communications at the MAC sublayer by building a wireless distribution system [50]. Purely ad hoc networks have infrastructure-less architecture, wherein mobile nodes have arbitrary movements resulting in frequent topology changes and non-guaranteed network connectivity. Unlike Purely ad hoc networks, WMNs provide a fixed/mobile infrastructure to improve network connectivity regardless of node mobility. WMN allows each user device to connect to each other and also to a backbone network (or the Internet) via a mesh gateway. The following briefly describes different types of WMN nodes, the proposed architectures, protocols, and routing metrics for WMNs.

Types of WMN Nodes: WMN introduces three types of nodes: mesh stations (mesh clients), mesh routers, and mesh gateways [46, 51].

- **Mesh stations** are wireless end-user devices that have wired/wireless connections to mesh routers.
- **Mesh gateways** are responsible for connecting wireless mesh infrastructure to other wired/wireless networks, such as cellular and Internet.

WMN Architectures: Three types of architecture are proposed for WMNs: client mesh, infrastructure mesh, and hybrid mesh [44, 45, 47, 50, 52]

- **Client mesh:** In this architecture, mesh clients act similarly to ad hoc networks, with the difference that they create a temporary backbone to improve network connectivity. They also provide Internet and end-user applications to other nodes and perform additional functions, such as routing and self-configuration. Hence, there is no need to mesh routers in this architecture.
- **Infrastructure mesh:** Unlike the client mesh architecture, mesh clients do not participate in network connectivity and routing functions in this architecture. Mesh routers provide

end-to-end connectivity between mesh clients by forming a mesh backbone. Mesh routers also give mesh clients access to the mesh network and Internet.

- **Hybrid mesh:** This architecture is a combination of two other architectures in which mesh routers form a mesh backbone. The backbone connects devices to the Internet and other networks using gateway/bridge mesh routers. Mesh clients connect to the mesh backbone directly or through multi-hop communications. Routing flexibility in mesh clients improves network connectivity inside wireless mesh networks.

Routing Metrics in WMN: In ad hoc networks, because of node mobility and a high rate of link breaks, the number of hops is mostly considered as a routing metric to find a quick path to the destination [50, 52]. In WMNs, the link break rate is not high compared to ad hoc networks because of stationary routers. Therefore, the quality of links could be the primary concern in routing protocols. In other words, routing protocols in WMNs look for the shortest paths with the best possible quality of links. Link quality can be measured by factors such as packet length, data rate, packet loss rate, link load, medium instability, and channel diversity [50]. The metrics used by routing protocols in WMNs are as follows [50, 53, 54]:

- **Expected transmission count (ETX) [52,54]:** This metric is the expected number of transmissions for successful packet transmission to the next hop. ETX is calculated based on the amount of packet loss in each link and considers only the quality of the links.

- **Expected transmission time (ETT) [52–55]:** This metric is the expected time for successful packet transmission to the next hop. ETT is calculated based on the packet size, bandwidth for each link, and ETX (inter/intra-flow interference and medium-instability are not considered).

- **Minimum loss (ML) [50,54]:** This metric multiplies the delivery ratio of all the links in the forward and reverse directions of the paths to select a path with the lowest end-to-end loss probability. ML only considers link quality.

- **Airtime link metric [54]:** This metric is the time taken to transmit a frame to the next hop successfully. Airtime link metric is calculated based on the channel access, protocol latency, data rate, frame size, and frame bit error rate.

- **Weighted cumulative ETT (WCETT) [52, 54, 55]:** This metric is calculated based on the sum of each link ETT and the maximum ETT of a link between available channels. WCETT is the first metric that considers channel diversity to avoid intra-flow interference. This metric does not consider inter-flow interference and medium instability.

- **A location-aware routing metric (ALARM) [52, 56]:** This metric improves WCETT by considering the impact of the distance between co-channel links.

- **Metric of interference and channel-switching (MIC) [52, 54, 57]:** This metric considers the number of interfering nodes in the adjacency of a node to avoid inter-flow interference. MIC also finds and guarantees the routes with minimum cost using the ETT metric in the calculations (medium instability is not considered).

- **iAWARE [52,54,58]:** This metric calculates the cost of a route by considering quality, data transmission time, data rate, medium instability, inter-flow and intra-flow interference, and packet size.

- **Energy consumption [53]:** The energy level of nodes can be considered as a routing metric. This is because the energy constraints of the relay nodes participating in routing can lead to a path failure.

- **Path availability/reliability [53]:** This metric estimates the percentage of route availability.

Classification of Routing Protocols in WMN: Routing protocols in WMN can be classified into four categories [47, 50], namely ad-hoc, controlled-flooding, opportunistic, and traffic-aware. They differ in routing structure (flat or hierarchical), routing approaches (route discovery and maintenance), topology control, mobility management, and location awareness [50, 53].

- **Ad-hoc based protocols:** They adapt ad hoc routing protocols such as DSDV, DSR, and OLSR to the wireless mesh environment considering link quality in their metric (for example, link quality source routing (LQSR), mesh distance vector (MeshDV), and radio-aware optimized link state routing (RA-OLSR)).

- **Controlled-flooding protocols:** They use different techniques of controlled flooding to minimize the overhead of routing updates in the network (for example, localized on-demand link-state (LOLS) and wireless-mesh-network proactive routing (WPR)).

- **Opportunistic protocols:** They utilize the broadcast nature of wireless devices to refrain from retransmissions in the network (such as extremely opportunistic routing (ExOR) and resilient opportunistic mesh routing (ROMER)).

- **Traffic-aware protocols:** They perform traffic control optimization based on the assumption that most traffic is toward the gateway (for example, ad hoc on-demand distance vector spanning tree (AODV-ST), hybrid wireless mesh protocol (HWMP), and rainWala traffic-aware protocol).

WMN Challenges: While the WMN architecture allows implementing hybrid routing structures to achieve higher scalability levels than MANETs, it still does not address the fundamental scalability problems caused by multi-hop flooding under high node density. Furthermore, the existing WMN standards and routing protocols have limited capability to operate harmoniously with other types of wireless networks, such as cellular networks [44, 45, 51, 52, 59]. While this has not been a problem in the past, the growing interest in multi-hop D2D communications along with the multi-radio capability of mobile devices has created the need to develop scalable protocols to address this problem [60].

2.3.3 Integration of MANETs with Cellular Networks

In single-hop cellular networks, all communications between the mobile nodes are via the BS, even those in close proximity to each other. The drawbacks of this structure are 1) the costly infrastructure due to the need for more BSs to cover more areas, 2) the existence of dead/hot spots because of weak channel conditions and high data traffic during peak times, and 3) increased power consumption of mobile devices due to using higher transmission range as BS [12, 40, 40].

The integration of MANETs with cellular networks provides an opportunity to address some of the aforementioned difficulties. To achieve this, the integration of cellular networks and WLANs, the availability of multiple interfaces for mobile nodes, and multi-hop communications are the factors that must be considered in the design [43, 61]. The following briefly describes different aspects of the integration (such as types of relay nodes and the frequency bands used for multi-hop connections under cellular networks), the proposed solutions and architectures to make the integration possible, the benefits and limitations of the integration, and finally the role of SDN in addressing those constraints.

Types of Relay Nodes in Multi-hop Communications Under Cellular Networks: Multi-hop communications can have three types of relay nodes [12]: 1) fixed relay nodes installed by service providers with lower costs compared to BS or APs (no need for the wired backbone) [62], 2) mobile relays that are mobile stations acting as relay nodes to forward traffic [63, 64], and 3) hybrid relays that are a combination of fixed relays and mobile relays [65, 66].

Both fixed and mobile relays require the following: 1) at least two interfaces to prevent self-interference, 2) a powerful computational unit for routing operations, and 3) efficient interference management and power control to enhance the coexistence of D2D and cellular connections. In [12], the performances of the three types of relay nodes were compared in terms of various network criteria, resulting in the following findings:

- Fixed relays have lower costs than the BS. However, they have higher deployment costs than mobile relays because mobile relays do not have any deployment costs.
- Mobile relay nodes have faster power drainage when relaying.
- It is necessary to accurately determine the location of the fixed relay nodes and their numbers to optimize the network performance.
- The routing complexity of mobile relays is higher because of reasons such as mobility, their varying locations, and the necessity of periodic updates.
- Fixed relay nodes have better security and are more trustworthy because service providers deploy them. However, relay node authentication is difficult to achieve.
- The fixed nodes exhibit less scalability when the network grows, and the growing network should be replanned.
- Mobile relays require more intelligent routing strategies to provide better QoS, whereas fixed relay nodes have a better end-to-end delay.

- In mobile relays, it is necessary to provide incentives for mobile users who accept to be a relay node for forwarding other users' data traffic, such as monetary incentives and service level improvement.

Frequency Bands for Multi-hop Communications under Cellular Networks: D2D in cellular networks can be provisioned via cellular frequencies (inband) or unlicensed frequencies such as WiFi (outband). The former experiences higher computational overheads as well as more complexity in interference management, while the latter enables mobile users to have cellular and D2D communications simultaneously without being concerned about the interferences that may come from D2D communications on cellular transmissions. However, the outband communication has several limitations: 1) D2D interference is not controlled and managed by the BS, 2) devices must have at least two radio interfaces: one for cellular connection (i.e., LTE interface) and the other for D2D connection (i.e., WiFi interface), and 3) packets need to be decoded and encoded by each interface in a different way [67]. However, the inband method has attracted more attention from researchers because it provides mobile users with simultaneous D2D and cellular connections. The inband method requires peer discovery techniques and effective routing strategies to enable load balancing between two modes of communication (D2D and cellular modes) to maximize network performance.

Studies on Multi-hop Routing Under Cellular Networks: The proposed architectures and routing protocols to integrate multi-hop routing with cellular networks are summarized as follows:

- 3GPP release 12 is focused on employing D2D communications in LTE networks for public safety applications, proximity-based services, and network offloading by integrating IEEE802.11 technologies into cellular technologies. LTE-Direct was proposed by the 3GPP workgroup, referenced as proximity service (ProSe), for peer-to-peer discovery in cellular networks. The supported network scenarios by 3GPP release 12 are direct communication between two devices when both devices are under network coverage, or both UEs are outside the network coverage (only used for public safety scenarios), or one is under network coverage and the other is outside the network coverage. 3GPP release 13-15 also approved two-hop communication to further increase network capacity. However, the development of routes over more than two hops is not supported by 3GPP [68, 69].

- Haiyun et al. [63] proposed an architecture called unified cellular and ad hoc networks (UCAN). In this architecture, mobile nodes have two interfaces: a cellular CDMA/HDR interface to connect to a cellular network and an IEEE 802.11b interface for multi-hop connectivity. This is proposed for scenarios in which all nodes are in the BS coverage. This architecture improves the performance using multi-hop D2D communication when downlink signaling is poor. It uses proxy clients with higher-quality downlink signals to relay packets to their destinations. Proxy clients act as relay nodes between the cellular and ad hoc networks and decide when a vertical hand-off must be performed.

- Hung-yu et al. [70] proposed a two-hop relay architecture that integrates wireless wide area networks (WWAN : 3G or 4G) with wireless local area networks (WLAN: IEEE802.11) using dual-mode (WWAN/WLAN) mobile terminals as relay gateways to relay traffic between the BS and end-users. The relay gateway (RG) in the proposed architecture is equipped with both WLAN and WWAN interfaces. RG is not allowed to relay traffic until it is authenticated and registered by a cellular network. After the RG is registered, it is attached to the BS and broadcasts a relay advertisement message via its WLAN interface. When a mobile node receives a relay advertisement, it decides to transfer its data directly to the BS or through a relay gateway by considering factors such as channel state, power consumption, mobility pattern, and time limitations of the application. In the case of selecting a two-relay, the mobile terminal sends a relay request to the RG. The RG then checks with the cellular network to determine whether the mobile terminal is authorized for two-relay services. Afterward, the cellular network sends a relay reply to inform the RG if the request is accepted or declined. If the mobile terminal is authorized for two-relay services, the RG forwards the relay initiation message to the mobile terminal to start the application. Their results showed significant capacity improvement in a fixed-rate uplink CDMA system and a variable data rate downlink HDR-like system.

- Ruay-Shiung et al. [71] proposed a hybrid wireless network protocol integrating ad hoc and cellular networks. In the proposed protocol, two different communication modes are designed: one-hop and two-hop direct transmissions within BS orientation. Two mobile nodes can communicate directly or through a third mobile node as a relay node without assistance from a base station. If the destination is in the neighbor table of a mobile node, the communication is direct. If the destination is in the neighbor table of its neighbor, then its neighbor acts as a relay node. The proposed protocol combines the strengths of BS-oriented and ad hoc networks to address problems, such as BS failures and handoff procedures in the former, and weak route and communication connectivity in the latter. Their simulation results indicated an increased communication rate of 20% when two-hop communication occurs in the network.

- Hung-Yun et al. [72] proposed a hybrid wireless model in which mobile nodes operate in multi-hop mode. In the model, an AP (BS) serves a particular coverage area based on conventional physical cellular infrastructure. A separate control channel is utilized for communication between the AP and mobile nodes. The AP decides on the medium access mode the mobile nodes should operate and their transmission power level. Mobile nodes forward their location and the observed throughput to the offered load ratio to the AP via the control channel. The AP runs an algorithm based on the received information to determine the operating mode between two mobile nodes: cellular mode or MANET mode. Their simulation results indicated better performance for the proposed model by taking advantage of the best features of both ad hoc and cellular networks. They also discussed that the overhead resulting from switching between the two modes is negligible and does not require any changes in the MAC layer.

- Ying-Dar et al. [40] presented a new multi-hop cellular architecture, referred to as MCN, that allows multi-hop communications between mobile nodes. Unlike single-hop cellular networks in which all communications are through the BS, in MCN, mobile nodes within a cell reach each other via multi-hop connections. If the source and target are not in the same cell, the source node sends data packets to the BS through multi-hop, and the BS forwards the received packets to the BS of the target cell. Their numerical results illustrated the significantly increased throughput of the proposed architecture compared to single-hop cellular architectures. Further, their study showed that the throughput of the MCN will increase if the transmission range decreases.

- Hongyi et al. [62] designed an integrated cellular and ad hoc relaying system (iCAR) that addresses the congestion problems of a cellular network by employing modern ad hoc relaying technologies. The proposed system utilizes ad hoc relaying stations (ARSs) to dynamically relay traffic between cells to balance the traffic load efficiently. The ARS is equipped with two air interfaces: one for connecting to the BS and the other for communicating with mobile nodes and other ARSs. Mobile nodes use two separate air interfaces to communicate with the BS and ARS. The proposed model can increase the network capacity while decreasing the transmission power of mobile nodes. Their analysis and simulation results showed that in a congested cell with a limited number of ARSs, iCAR reduces the call blocking/-dropping probability and increases the overall system throughput compared with traditional cellular networks. However, these gains come at the cost of an increase in hardware complexity and signaling overhead.

- Xiaoxin et al. [65] presented a channel allocation mechanism, called a mobile-assisted data forwarding (MADF), to integrate an ad hoc overlay with the cellular infrastructure. MADF interconnects mobile users in a cell with high density to the surrounding cells with low density without going through the BS. This increases the network capacity and provides load balancing between cells. The proposed model divides the available wireless channels into fixed and forwarding channels. Forwarding channels are used to transfer data from a hot cell to the neighboring cold cells. The mobile node that desires to be a relay agent performs the following tasks: First, it measures the local traffic in its cell by sending test packets to the BS and estimating the delay from the BS reply. If the delay exceeds a specific threshold, the cell is considered a hot cell. Subsequently, it periodically broadcasts a free signal to inform other mobile nodes about its availability for relaying the packets. The mobile node receiving a free signal starts to use the agent relay to forward its traffic. The proposed model defined two traffic thresholds: one for employing the forwarding channel and the other for stopping the use of the forwarding channel when the cell is no longer a hot cell. Their results showed that the proposed model improved the network performance significantly, under a specific delay requirement.

- Ali et al. [66] Proposed a novel adaptive and scalable architecture for wireless networks, referred to as self-organizing packet radio ad hoc network with overlay (SOPRANO) that combines cellular networks with multi-hop packet radio systems. The proposed architecture aims to increase the network capacity with a reasonable degree of complexity in implementation. The proposed model employs a cell-splitting technique to enhance cellular capacity. This technique splits a cell into a number of cells, where each cell is managed by different BSs. The proposed model increases the network capacity. However, interconnecting the BS to the cellular core network is costly. To address this problem, the authors proposed a novel architecture in which wireless routers replaced the base stations in the split cells. Routers are used to connect mobile terminals to the BS via multi-hop wireless connections. The proposed architecture can significantly reduce the total transmission power by selecting an appropriate routing approach. Six steps of self-organization are recommended for the proposed architecture to optimize the network capacity: 1) neighbor discovery, 2) connection setup, 3) channel assignment, 4) mode selection, 5) mobility management and topology updating, and 6) changes in control and routing information. In the first step, the mobile terminal searches for the available BS or routers. The second step addresses node registration to the current network and location updates. In the third step, nodes receive channel information to be used to transfer data or listen to. In the mode-selection step, routers decide on their transmit or receive modes in various time slots. In the fifth step, the employed routing strategy must handle the mobility of the mobile terminals and routers to react faster to topology changes in the network. The final step should be optimized after determining the optimal transmission. This study also investigated the new challenges of the proposed system for capacity optimization. Further, the authors discussed several interference mitigation strategies to decrease the interference in packet relaying. They showed that employing different technologies and techniques in the physical, data link, and network layers can affect the network capacity in the proposed system.

Regardless of the aforementioned solutions for multi-hop routing in cellular networks, practical development was still uncertain until the recent introduction of the SDN paradigm.

Limitations, Challenges, and Requirements of D2D Under Cellular Networks: D2D under cellular networks provide several benefits that can be summarized as follows: 1) higher data rates and low-cost communications by employing IEEE802.11 wireless interfaces for D2D communications, 2) extending the cellular coverage via multi-hop connection to dead zones, 3) diminishing the total transmit power, 4) enhancing the total system capacity and data rate, 5) increasing the traffic load balancing by redirecting traffic from congested cells to less congested cells, 6) reducing the capacity bottlenecks as the number of available channels for BS increases, 7) boosting the routing reliability in multi-hop communications with the aid of BS, and 8) rapid deployment of the network when the fixed infrastructure or BS are not available (such as, emergency rescue scenarios and battlefields, military, and civil requirements) [35, 43, 73].

Regardless of the benefits, D2D under cellular networks suffers from several constraints, such as 1) latency in vertical hand-off, 2) weak security due to delivering user data by other user devices, 3) complicated resource management (frequency allocation) owing to the possibility of operating different wireless technologies in the same frequency range, 4) challenging authentication, authorization, and accounting (AAA) procedures, 5) delay due to buffered packets in relay stations or congestion that might occur in multi-hop transitions, 6) difficulty in interference management, particularly in scenarios where D2D and cellular networks share the same licensed frequency, 7) complexity in multi-hop routing and resource management, 8) QoS, and 9) challenging peer discovery and mode selection between LTE and D2D [12, 18, 74].

To address such challenges and make the integration between D2D and cellular networks more efficient, meeting the following requirements is necessary [41, 41, 43, 74, 75]:

- Providing powerful processing and power capacity for BS and APs.
- Implementing load-balancing strategies between cellular or WLAN traffic versus MANET traffic.
- Employing efficient techniques to mitigate the interference and to control the power.
- Inventing switching techniques to efficiently switch between available wireless technologies to achieve higher data rates.
- Implementing efficient vertical and horizontal handoff to minimize delay as much as possible.
- Increasing the multi-hop routing scalability under cellular networks without increasing the overhead.
- Seeking power control solutions and power-aware MAC protocols to address the performance limitations of CSMA/CA in the IEEE802.11 standard.
- Developing intelligent algorithms for efficient resource management and resource allocation for D2D and BS/APs.
- Developing the cross-layer solutions to provide the required information to the upper layers, as different mac-layer connections exist within a multi-hop path.

Benefits of SDN Technology in Multi-hop Wireless Networks: To address the aforementioned limitations and challenges, an SDN-based multi-hop architecture can be an intelligent solution. Taking advantage of the SDN concept, control functions are taken from wireless devices and are logically centralized in an SDN controller. The controller programs the wireless nodes to perform the data plane functions. Employing SDN in multi-hop wireless networks brings many benefits, such as [19]:

1. Improving network management and network optimization due to global knowledge of the network, centralized control functions, and programmable forwarding devices.
2. Enhancing node mobility management by providing real-time information on the node location. This leads to fast tracking and adaptive dynamic routing decisions in the network.
3. Optimizing energy management and reducing the energy consumption of nodes because nodes do not have to run complex routing algorithms or calculate the best route frequently

upon network topology changes. The controller provides load balancing in the network and makes routing decisions based on the full knowledge of each node position and energy.

4. Improving QoS via flow-based routing and efficient resource allocation in an SDN framework. This is because the controller periodically collects network information and consequently has up-to-date information about the entire network. This enables efficient programming of forwarding devices to meet application requirements. Furthermore, based on the requirements of each traffic type in the network, the controller applies different policies to different flows, as each flow belongs to a different traffic type.
5. Optimizing routing decisions and dynamic adaption to topology changes by centralizing routing protocols and having real-time global network knowledge in the SDN framework.

2.4 Software Defined Networking

Traditional networks suffer from several constraints, such as the difficulty of network management due to a closed network structure, the high cost of network operation and maintenance for flow control, and difficulties in network innovation and network configuration due to the distributed architecture [13]. Furthermore, in traditional networks, packet forwarding and routing decisions are performed using network devices. Hence, changing or upgrading the network requires significant changes in device configurations and network protocols, which are costly and time-consuming. The tight coupling of control and data plane functions in network devices makes the network inflexible and difficult to reconfigure or upgrade. SDN removes the control functions from the network devices and makes them simple forwarding devices that receive forwarding instructions from a logically centralized control unit, known as a controller. Network policies are determined by the application plane that resides at the top of the controller. The applications program the devices using the API provided by the controller [19].

2.4.1 SDN Architecture

Unlike traditional networks with tightly coupled control and data plane equipment, in the SDN platform, the control plane is removed from the forwarding devices, and all control functions and network intelligence are logically centralized in one or more controllers. SDN's decoupled control and data planes structure, along with its application-centric and software-based features, make the network open and programmable. Three layers are designed in the SDN framework from top to bottom, namely the application, control, and data planes (see Figure 2.4). Open APIs are also defined to enable communication between the layers, namely the northbound and southbound APIs. These APIs are used for the communication of the control plane with the application and data planes. Furthermore, Eastbound and Westbound APIs are defined for communication between the controllers to provide scalability and interoperability in SDN-based networks.

In the SDN framework, the data plane consists of simple programmable forwarding elements that receive forwarding instructions from the controller. The control plane has a global and real-time view of the network and maintains the network topology up-to-date by monitoring the network

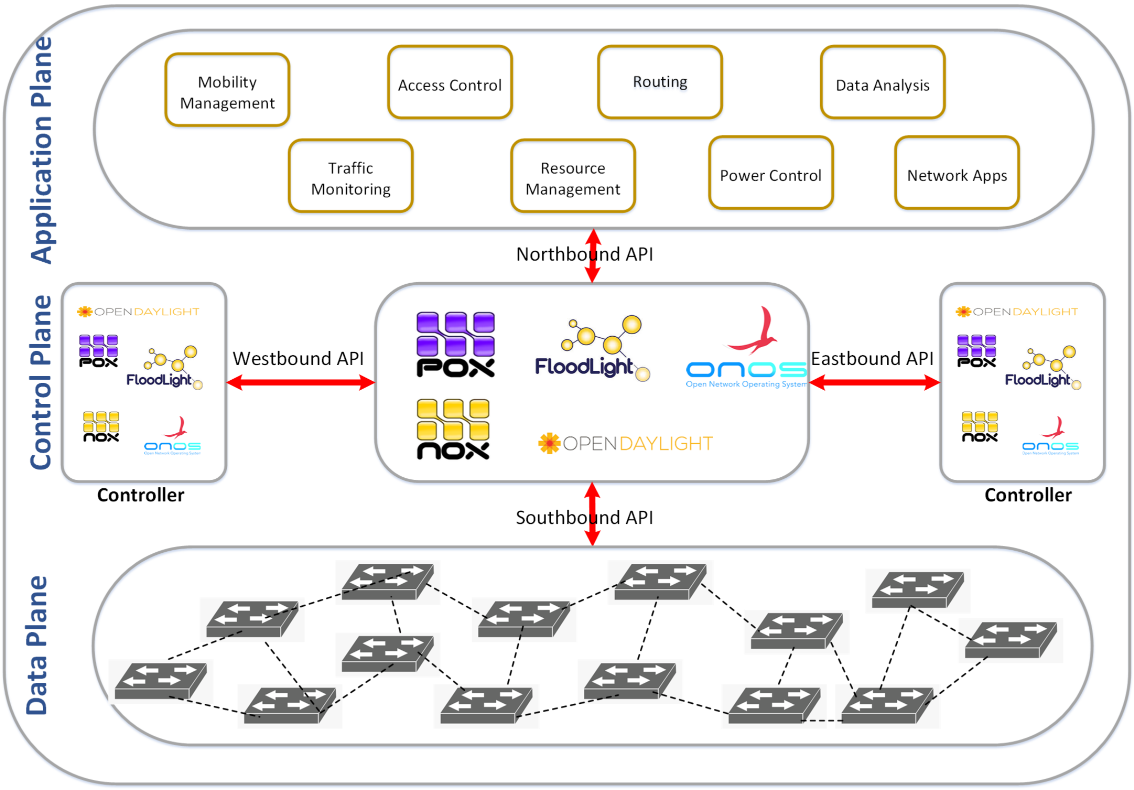


Figure 2.4: SDN architecture.

and dynamically allocating network resources based on the application requirements. The control plane installs the flow rules on the forwarding devices through the southbound interfaces. Network functions, such as routing algorithms, load balancing, and access control, are applications running on top of the controller in the application plane. The SDN programmability feature and decoupled control and data planes in the SDN architecture provide a flexible network that can be managed efficiently. Further, innovation and deployment of new services in SDN-based networks can be a simple software update. This will reduce the network cost and delay and increase the network flexibility and performance. The SDN functionality can be seen in different applications, such as WLANs, cellular networks, mobile clouds, VANETs, air-space-ground integration networks, satellite networks, and unmanned aerial vehicle (UAV) networks [13, 19, 76].

2.4.1.1 Control Plane in SDN Architecture

The controller has a global view of the network by collecting data plane information. Consequently, the controller can manage and optimize the resource allocation and routing in the network efficiently, based on real-time information of the network and considering the requirements of the application plane.

There are two communication methods between the controller and forwarding devices: in-band and out-of-band. The former uses the same channel to exchange the control and data messages. The latter employs two separate channels for data and control traffic, which provides less congestion and delay in the network [19].

The control plane in an SDN framework can be centralized or distributed. The former has limitations, such as scalability and a single point of failure. The latter method provides better scalability. However, it brings new challenges, such as decision-making on the number of required controllers, their placement, and coordination between the controllers [19].

2.4.2 Brief Overview of OpenFlow Protocol

The OpenFlow protocol is the most widely used protocol for SDN. In 2008, OpenFlow began as a project at Stanford University. The project main purpose was to provide a tool for researchers to help them implement their experimental protocols in the network. In 2011, the Open Networking Foundation (ONF) was established to coordinate the development of SDN standards and solutions. Their purpose was to stimulate the delivery of SDN applications, services, and products. As a result, different versions of the OpenFlow standard have been published by ONF [77, 78]. The current version of OpenFlow is 1.5.1, and the latest version (1.6) is only accessible to the members of the ONF.

OpenFlow includes three main components: OpenFlow protocol, controller, and OpenFlow switches. The OpenFlow protocol is used to enable communication between the controller and switches. In OpenFlow-based networks, at least one switch must be connected to the controller to make the OpenFlow operational. Open VSwitch, Indigo Virtual Switch, and Centec's Lantern are examples of OpenFlow-based switches.

An OpenFlow controller employs the OpenFlow protocol to connect and configure switches. Each OpenFlow controller supports an API written in a specific language. Further, each controller has its own set of common network applications, such as routing, load balancing, firewall, and topology discovery. The following briefly introduces well-known OpenFlow controllers: 1) NOX is the first OpenFlow-based controller developed in C++. Regardless of its complexity, it provides performance-critical applications, 2) POX is a Python version of the NOX controller that provides faster development and prototyping of the software running inside the controller. However, it does not work in a distributed manner. POX and NOX only support version 1.0 of OpenFlow, 3) Ryu is an open-source platform that is component-based and is implemented in Python. Ryu is considered an operating system for SDN that can be used in large networks. Unlike POX, it can work in a distributed manner and supports versions 1.0, 1.2, 1.3, 1.4, and 1.5 of the OpenFlow protocol. In addition to OpenFlow, it supports other protocols for southbound interfaces (for example, OF-config and Netconf). However, both Ryu and POX provide lower network performance compared to java-based and C++-based controllers, 4) Beacon is a Java-based controller that can address the portability problems of C and C++-based controllers. Furthermore, it reduces the compile time and provides better error logging for developers, 5) Floodlight is a Java-based controller that supports versions 1.0-1.5 of the OpenFlow protocol. It presents more features than Beacon, such as northbound REST APIs, supporting non-OpenFlow domains, and well-written documentation, 6) ONOS is a distributed open-source network operating system. It is java-based and supports versions 1.0 and 1.3 of the OpenFlow protocol. ONOS also supports other protocols, such as OVS-DB and OF config, and 7) Open Daylight (ODL) is a Java-based controller that allows for southbound communications with both OpenFlow-based and non-OpenFlow based protocols. It also employs powerful REST APIs. As it offers different services, it is suitable for real business environments. However, for the fast development of SDN applications that only utilize OpenFlow, Floodlight may be a better choice and is simpler, smaller, and well-documented compared to ODL. Both ODL and Floodlight support versions 1.0-1.5 of the OpenFlow protocol [77, 79–81].

The topology discovery in OpenFlow-based controllers was originally designed for wired networks, where switches are connected to the controller via a cable. Consequently, there are some limitations for node discovery and link discovery in OpenFlow-based networks, where the switches are wirelessly connected to the controller. Furthermore, wireless features, such as mobility, interference management, and techniques used for channel selection, have not yet been addressed in the OpenFlow protocol [82]. Hence, it is necessary to modify the OpenFlow protocol and extend its functionality for wireless networks.

2.4.3 SDN Platforms and Tools

Tools that can be used to test, experiment, and study different SDN-based networks can be classified into two categories [79]: tools in one category are used to simulate the networks (e.g., Mininet and Mininet-WiFi) and network events (e.g., NS3), whereas tools in the other category are utilized to monitor network performance and benchmarking [79]. OFLOPS [83], perfSONAR [84], NICE [85], and OFTest [86] are examples of the second category. The discussion of the aforementioned tools is beyond the scope of this study.

Mininet [87] and Mininet-Wifi [82] are the two most well-known platforms used for emulating SDN networks. Mininet provides a flexible and straightforward testbed to build different network topologies and develop OpenFlow applications. It is only used for emulating wired networks, including virtual switches, controllers, hosts, and links. To emulate wireless networks, the Mininet-WiFi is introduced, which is a fork of Mininet. It extends the functionality of Mininet by adding virtualized APs and WiFi stations. In both Mininet and Mininet-WiFi, Python APIs are used to easily create, test, and customize SDN networks. Users can have a complete experimental network on their laptops or PCs and build a virtual network of a large number of OpenFlow switches, hosts, and controllers.

Estinet [88] is another network simulator that supports the OpenFlow protocol. Estinet supports the features of an emulator and simulator simultaneously. Estinet simulation and emulation results are accurate and repeatable. However, it is proprietary software that runs simulation and emulation projects on company servers [77].

NS3 is a well-documented and well-established simulator that supports OpenFlow switches and the OpenFlow protocol. However, real OpenFlow controllers cannot be run without modification [77]. Further, it only supports an outdated version of the OpenFlow protocol (0.8.9), limiting the research area. Luciano et al. [89] introduced a new module to NS3 to support version 1.3 of the OpenFlow protocol.

2.4.4 Integrating SDN with Wireless Networks

Different studies have employed SDN in different wireless networks, such as MANETs, WMNs, and cellular networks. A brief overview of these studies can be summarized as follows:

Integrating SDN with MANETs: SDN-assisted routing in MANETs eliminates the flooding of broadcast messages in the network for route discovery and route maintenance. Hence, some studies have proposed SDN-based solutions to improve the performance of MANETs:

- Yu-Chee et al. [35] proposed a model to evaluate the routing path lifetime in MANETs by using the joint probability of route lifetime based on the random walk model. The proposed model improves the MANET performance by predicting the route lifetime, as it can be used for the following purposes: 1) selecting a backup route when a route is likely to be expired, 2) choosing the most reliable route in route discovery, and 3) determining the proper size of a MANET (a routing path lifetime will decrease when the route length increases). Their simulation results indicated that the network would perform reasonably well for up to five hops.
- Toan et al. [90] designed an energy-efficient vertical handover algorithm to minimize the energy consumption of intermediate nodes in multi-hop wireless communications. In the proposed approach, each device is equipped with two controllers: local and extended. The latter provides communication channels between the local controllers of the devices. Moreover, the extended controller collects the network state information and sends this information to the local controller. The local controller uses the received information to instruct the mobile nodes to perform vertical handoffs between Wi-Fi and Bluetooth based on the application bandwidth requirements. Intermediate devices employ Bluetooth if the application requires a low bandwidth. Their results revealed the effectiveness of the proposed system.
- Carlo et al. [91] proposed an SDN-based solution for dynamic spontaneous networks to enhance dynamic QoS management. In their solution, an overlay method is adopted for multi-hop packet forwarding at the operating system level without any changes in the routing table.
- Hans et al. [92] implemented an SDN MANETS framework and demonstrated the performance of the proposed framework on real devices compared to traditional ad hoc networks. In their testbed, they use wireless switches (Open vSwitch) on top of the Linux kernel that can be programmed by the controller. An ONOS controller manages MANETs and decides the routing functions. The controller asks the switches to perform neighbor discovery using the LLDP and forward the received LLDPs to the controller. In this way, the controller has updated network topology and can decide on the routes between peers. In the designed testbed, each node has two components: a host and a switch connected via an Ethernet interface. The switch inside each node manages the connection of two wireless interfaces connected to the controller and other SDN nodes. The Linux kernel of the switch is modified to enable the switch to be directly controlled by the SDN controller. They added a new

module, called the SDN bridge, between Ethernet/WiFi modules and the Ethernet stack to send the incoming data streams to the SDN bridge module for further processing.

- Ayush et al. [42] designed an SDN-based control communication protocol for multi-hop routing in infrastructure-less networks. In their proposed architecture, the SDN controller is located in one of the mobile nodes and is responsible for route selection and configuring the forwarding table of other nodes. A routing protocol is an application on an SDN controller that collaborates with other applications in the controller to select the best route. The control and data messages are exchanged using the same wireless interface. A specific port number is used to differentiate the control messages from the data messages. Forwarding devices have a local controller that runs as an application and is used to communicate with the SDN controller. Their simulation results in a network with sizes up to 50 nodes indicated a 40% reduction in routing overhead and improved throughput achieved by their proposed solution compared to traditional routing protocols, such as OLSR and DSDV.

- Klement et al. [73] proposed an SDN-based MANET architecture wherein each node has an uplink to the SDN controller to inform its local topology and channel bandwidth utilization. The SDN controller then determines the routes based on the bandwidth utilized by the MANETs. Furthermore, the controller reduces the bandwidth for flows when the MANET cannot provide the requested resources. Their results indicated that the proposed approach provides more reliable transmissions with better performance.

Integrating SDN with WMNs: Traditional WMNs suffer from some limitations [93], such as 1) complexity in network management, 2) difficulty in configuring forwarding devices based on predefined policies, 3) difficulty in redesigning, optimizing, and customizing the network in response to dynamic changes in network conditions and customer expectations, 4) tight coupling of control and data plane functions, 5) proprietary interfaces, 6) difficulty in hardware innovation to meet customer expectations, and 7) costly network maintenance. SDN can address such challenges by breaking the vertical integration between the planes, decoupling the control functions from forwarding devices, and centralizing them in control logic. Forwarding devices become programmable elements that are programmed by the control entity [93]. The programmability feature of the SDN platform facilitates faster and easier network innovation and application development in WMNs. Furthermore, this feature will help optimize resource management and load balancing in the network [13, 93]. The following highlights some of the studies conducted on the integration of SDN with WMNs:

- Yuhuai et al. [93] integrated the SDN paradigm with WMNs by employing OpenFlow-based mesh routers and a centralized controller. They designed a new traffic engineering algorithm called the hybrid routing forwarding algorithm (HRFA). The proposed scheme divides the network into legacy and SDN nodes. The legacy nodes run the OSPF to determine a route to the packet target. In contrast, SDN nodes forward the data packets based on the flows installed by the controller. In addition, SDN nodes perform some traffic measurements and forward the measurements to the controller. The controller uses the received traffic information to dynamically update the routing tables based on the network traffic changes. Their simulation results showed increased throughput and decreased delay and packet loss in the proposed scheme compared to traditional routing methods.

- Dawood et al. [1] presented a comparative analysis of channel switching latency for multi-hop multi-radio protocols, such as OLSR, BATMAN-ADV, and Open80211s via an experimentation testbed. Further, they evaluated the channel switching latency for SDN-based routing in wireless mesh networks. They implemented two topologies in their testbed: 1) 1-Hop topology with two mesh routers, and 2) 7-Hop ring topology. An out-of-band management network connects mesh routers to a network management server (NMS). The NMS performs channel switching by executing remote commands on mesh routers. For SDN analysis, NMSs perform additional tasks, such as routing and decision-making operations. The logical structure of the designed mesh router is illustrated in Figure 2.5. Their experimental results showed that regardless of the improved channel switching latency achieved by SDN, the latency is not negligible. They believe that applying software-defined radio (SDR) and split-mac on SND-enabled WMNs can mitigate latency.

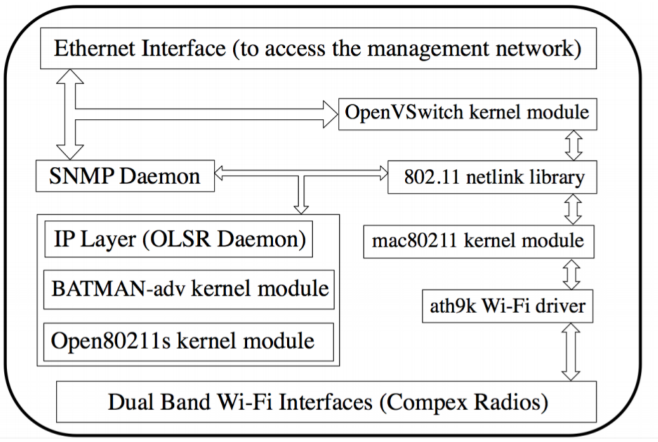


Figure 2.5: The logical structure of the designed mesh router by [1].

Integrating SDN with Cellular Networks: Regardless of the proposed solutions and techniques to enhance the performance of cellular networks, there are several limitations [94, 95], such as 1) poor support for virtualization, 2) inflexible and expensive equipment due to proprietary hardware and protocols, 3) costly and challenging network upgrades resulting from vendor-specific configuration of interfaces and devices, 4) complicated service deployments because of tight coupling of control and data planes, and 5) scalability issues caused by centralized data plane functions in the core entity of LTE networks and lack of fine-grained control over resources.

Software-defined networking aims to address the aforementioned challenges by providing a framework with open control interfaces wherein the future evolution in wireless networking could be as simple as a software update rather than a complete hardware and architecture upgrade. In the following, some of the proposed solutions for integrating SDN with cellular networks are described[1]:

[1]More literature on SDN-based cellular networks are presented in other chapters of this thesis.

- Seil et al. [96] proposed an SDN-based mobile networking approach wherein SDN is integrated with a legacy cellular control plane (e.g., 3GPP EPC) to make the cellular network more flexible and scalable. The proposed approach utilizes the previously designed control planes for IP-based and 3GPP-based mobility management to manage the location of mobile nodes and their movements. Two controller models are proposed for the designed approach, referred to as single-controller and hierarchical models. In the first model, the SDN controller only monitors the network and manages the routing between the forwarding elements. IP address allocation and mobility management are performed using legacy mobility control protocols in the cellular network. This model is not scalable and is suitable for small-scale network deployment. The second model provides more scalability by determining the local and global control entities. Local control entities (L-CE) monitor and manage the routing between forwarding elements under their mobility domains, while global control entities (G-CE) manage the mobility between the local mobility domains. Their evaluations of the proposed architecture and designed models can be summarized as follows: 1) latency in detecting the location changes of a mobile node depends on the selected mobility protocols (host-based or network-based), while latency to update the flow table of the involved forwarding elements depends on the instantiated SDN controller; 2) the controller can take fast action to establish alternative paths in the case of failure since it has a global view of its managed domains. Consequently, the proposed approach is more reliable and flexible than legacy mobility management because it provides more flexibility in handling the flows; and 3) both controller models maintain the legacy mobility control plane in traditional cellular architecture. However, defining specific protocols is required to integrate the legacy control plane with the SDN. This depends on which control entities should be bound by the controller (i.e., MME with S-GW/P-GW or only MME) and the location of those entities (i.e., front or back from the controller). The authors believe that their approaches provide insights into the realistic evolution of current mobile network architectures.

- Abbas et al. [14] proposed an SDN-based cellular network architecture called CSDN, which leverages the best features of SDN and NFV in cellular networks: SDN to abstract the network and to decouple the control plane from the data plane, and NFV to separate logical network functions from the underlying hardware. This combination will help with the dynamic management of resource allocation, fast deployment of new services, support for multi-cell collaborative signal processing, and easy network upgrades. The authors also suggested that collecting network information in mobile edge networks can be beneficial for optimizing resource usage and application performance and enhancing the QoE. In the proposed SDN-based architecture, network control and management are simplified, and the deployment of new services can be more flexible in a new programmable and open manner. In the proposed architecture, large parts of the main LTE entities are implemented as applications in a centralized cloud-based infrastructure. They also designed a new architecture with an additional knowledge plane layer that enables service providers to achieve user-aware control and network resource orchestration.

- Xueli et al. [97] explored the possibilities of virtualizing EPC entities (e.g., P-GW) in LTE networks. A novel SDN-based P-GW architecture is proposed, wherein data forwarding functions are performed using low-cost IT hardware. The authors utilized the NFV approach to implement the forwarding functions of P-GW in software and to use a number of low-cost IT servers to run the software. In P-GW, the control plane is separated from the data plane. The control plane of P-GW, referred to as PGW-C, is implanted in software that can be run on physical machines or VMs in the data center. The data plane of P-GW, referred to as PGW-U, performs data plane functions under the control of PGW-C. A software-based P-GW orchestrator is also designed to manage the tunnels (i.e., creating, removing, and modifying the tunnels) based on the instructions received from the PGW-C. The proposed architecture reduces the cost of LTE operating services. Further, it makes the cellular networks more intelligent and reliable by implementing fine-grained resource management policies on the designed virtualized gateway.

2.4.5 Challenges of SDN-based Multi-hop Wireless Networks

SDN-based multi-hop routing in wireless networks has been faced with several challenges that must be taken into account when designing the network, such as 1) determining the communication approach between the controller and the forwarding devices and how forwarding devices should be programmed; 2) selecting between one-hop wired or multi-hop wireless connection to the controller. The former is simple but not scalable. The latter requires an approach to find the least-cost path to the controller; 3) designing isolation methods between the control and data traffic and defining policies to minimize the exchange of control messages between the controller and forwarding devices for in-band SDN architectures, to improve energy usage and reduce the delay; and 4) selecting an appropriate physical structure for the control plane, centralized or distributed. Each structure has its advantages and disadvantages. The centralized architecture provides lower deployment costs, and the controller has a global view of the entire network. However, this structure is not scalable and reliable because of a single point of failure. In addition, the centralized structure increases the energy consumption and network congestion in the high-mobility networks owing to the frequent exchange of control messages between the controller and forwarding devices to obtain up-to-date network information. In contrast, the distributed structure is more reliable and reduces the congestion and delay in the network by using multiple controllers and load balancing between them. However, this structure has several challenges, such as finding the optimized number of controllers and their physical location to achieve a trade-off between cost and network performance, effective management of multiple controllers to achieve a centralized view of the network [19].

2.5 Summary

This chapter provided an overview of the evolution of cellular networks. As discussed, the fifth and sixth generations of cellular networks are undergoing development due to the limitation of the 4th generation and the necessity of global network coverage. The SDN-based multi-hop routing in the cellular network was introduced as a potential solution to enhance the cellular network performance and to address the scalability issues of the pure ad hoc networks with distributed network management. This is because in SDN-based multi-hop routing, MD2D communications are partially or fully managed by one or more controllers and the controllers have a global knowledge of the network. The following paragraphs highlight the main findings of this chapter.

As reviewed in Section 2.2.4, the current cellular networks are faced with ongoing challenges, such as the increased number of Internet users and the need for a higher number of connected devices, growing demands of subscribers for peer-to-peer applications that place significant traffic load on the cellular network, and the necessity of handling high-load network traffic with the least delay and the highest data rates. The fourth generation of the cellular network, LTE in particular, cannot handle the rapid change of network requirements and the fast-growing of Internet users. This is due to LTE limitations, such as 1) inflexible architecture owing to standardized and proprietary hardware, protocols, and interfaces. Consequently, any network changes or development of new services require a timely standardization process and costly hardware replacement, 2) limitations in scalability and traffic bottlenecks due to handling all the traffic via the cellular network or a BS, including even the communication between mobile nodes within a cell. The literature has proposed several solutions to address the above-mentioned limitations and challenges. Among those solutions, device-to-device communications (D2D) and software-defined networking (SDN) were the main interests.

Device-to-device communication involves direct wireless communication between two mobile devices in close proximity. Multi-hop D2D (MD2D) communication is an indirect communication between two mobile devices through relays. A mobile ad hoc network consists of self-configured mobile devices that use MD2D to communicate without the help of any fixed infrastructure. Despite the flexibility, purely ad hoc networks were not scalable due to their distributed management and limited power of wireless devices participating in multi-hop routing. The integration of ad hoc networks with cellular networks was introduced to address some of the limitations of both networks. The integration can address scalability issues of ad hoc networks. Furthermore, cellular networks can benefit from MD2D to extend cellular coverage for dead zones, maintain connectivity when the BS is not available, and reduce the cellular traffic load by offloading traffic to multi-hop routing when possible. It is necessary to design scalable multi-hop routing protocols with the least routing overheads to efficiently switch between the communication modes (i.e., cellular or MD2D) and maximize the cellular network performance. However, the integration of MD2D routing protocols and services into the LTE architecture requires a timely and costly standardization process due to its proprietary components. SDN and NFV can address this issue by enabling reprogrammability to network, providing open interfaces, virtualizing the network functions, removing the control functions from the hardware equipment, and logically centralizing the control functions in one or more control entities. By employing SDN in cellular networks, service providers can experience easier network management and more flexibility in the services. Further, the development of new services and protocols requires a simple software update that is rapid and inexpensive. The recent introduction of open frameworks such as O-RAN, and the support of SDN in the edge and core of 5G architecture provides a basic construct to integrate MD2D concept to a cellular network.

In summary, the integration of SDN and MD2D into cellular networks can significantly increase cellular performance by taking advantage of both MD2D and SDN technologies. To this end, new open and reprogrammable frameworks should be designed to integrate SDN into cellular networks. Further, new multi-hop routing protocols should be developed to conform to SDN-based cellular frameworks. Until now, only a few studies have been conducted on SDN-based multi-hop routing in cellular networks, which is the main focus of this thesis.

3

A Routing Framework for Offloading Traffic from Cellular Networks to SDN-based Multi-Hop Device-to-Device Networks

3.1 Introduction

We are living in a highly connected world, where access to mobile high-speed video and multimedia-based applications is becoming a prerequisite for consumers. Device manufacturers and network providers face an ongoing challenge in designing highly scalable networks capable of servicing current and future demands. Although 4G LTE networks were only recently introduced, studies on 5G networks have already begun to address the shortcomings of current cellular networks, such as high operational costs and difficulty in upgrading/modifying existing architecture due to its proprietary hardware features, complex and hard-coded network protocols and operations, slow and complicated network and service deployment because of tight coupling of control and data planes in the devices and equipment, inefficient radio resource management due to the difficulty of LTE management, and the existence of various heterogeneous access networks and technologies (such as LTE, WiFi, MTS, and GSM) [98–101]. Consequently, there is a need for programmable networks to speed up and facilitate network innovation and deployment.

Software-defined networking (SDN) enables the programmability feature for 5G networks to accelerate service deployment and network innovation and to simplify network management for mobile operators. SDN, through the separation of the control plane from the data plane, provides a framework for faster innovation, where the future evolution in wireless networking would be as simple as a software update rather than complete hardware and architecture upgrade [102]. SDN allows future cellular networks, such as 5G, to be effectively managed and extended via software. This means that such networks can be more easily integrated with other mobile networks, such as MANETs or sensor networks. Significant research has been conducted on SDN-based 5G cellular networks [14, 103–111].

D2D communication is another promising technology for 5G networks that enables wireless network devices to communicate directly instead of going through a cellular base station. This is seen as a potential benefit by the providers to offload traffic (when possible) and make more efficient use of both the licensed and unlicensed spectrum. 3GPP Release 12 has introduced the support of D2D or direct mode communications in LTE-Advanced (LTE-A), enabling peer-to-peer transmission between devices in close proximity to each other, without the involvement of cellular infrastructure for the user plane [68, 69, 112]. The presented scenarios include D2D under the control of eNodeB, D2D outside network coverage without supervision from eNodeB, and D2D in partial network coverage. In the latter, one of the UEs is inside the network coverage, while the other is outside the network coverage. 3GPP Release 13 has introduced the concept of the UE-to-Network to enable a remote node to connect to eNodeB or other nodes through a relay node via a side link interface. In addition to enabling direct D2D communication, 3GPP has also introduced three traffic functions, referred to as access traffic steering, switching, and splitting (ATSSS), to enable traffic management across multiple access technologies connected to the same core network sharing a common anchor point. ATSSS enables hybrid access and convergence of mobile and fixed networks and allows UEs to switch between WLAN and cellular links for more efficient and cost-effective data transfer. The functionalities of the ATSSS reside on both the core network and the UE side. The core network is responsible for traffic monitoring, defining different ATSSS policies, forwarding policies to UEs, and simultaneous use of available links and accesses [113].

Multi-hop device-to-device (MD2D) communications have been proposed to expand the capabilities of D2D communications to create a wireless mesh network with the assistance of cellular base stations [114–116]. MD2D provides greater coverage within cells (using the WiFi band) for offloading local traffic using multi-hop routes. While MD2D exhibits similar characteristics to mobile ad hoc networks (MANETs), their main difference is that MD2D is driven and controlled by the cellular network. The advantage of MD2D over MANETs (or wireless mesh networks) is their ability to be more tightly controlled and scaled more effectively in large networks [117–119]. The growing interest in MD2D communications along with the multi-radio capability of mobile devices has created the need to develop new routing protocols [43, 120–124]. Several studies have been conducted on MD2D communications in SDN-based cellular networks [125–127] that take advantage of both technologies. These studies show that MD2D communication under the management of a centralized controller can improve energy consumption, achieve a higher packet delivery ratio, and better QoS and network capacity. However, little attention has been paid to the design of routing protocols integrated with SDN-based cellular networks to enable MD2D communications. In our previous study, we demonstrated that SDN-assisted MD2D routing functionality can achieve significantly higher scalability by reducing network overheads when compared to existing MANET protocols, such as OLSR and AODV [128]. This chapter presents a new routing framework, called virtual ad hoc routing protocol framework (VARP), that supports developing different routing protocols. For this framework, the VARP-S routing protocol is designed that operates based on source routing. In VARP-S, the wireless SDN controller provides and maintains a complete source-to-destination path for each flow request from the source node.

In our proposed framework, we focus on LTE radio access networks (i.e., edges). The SDN controller is responsible for interference management, radio resource management, traffic engineering, load balancing, local topology management, and data offloading. The related algorithms and strategies can be deployed as an application on top of an SDN controller and can be easily upgraded owing to SDN programmability features. Further, the SDN controller provides fine-grained control over QoS considering the subscriber requirements. In the proposed framework, mobile operators can push programmable data offloading policies into the network based on real-time network conditions. Routing modules and resource traffic management modules are implemented as programmable software functions running on an SDN controller. It is noteworthy that in the future, there will be a large number of IoT devices that should be seamlessly integrated with future 5G networks. Our proposed routing framework makes this integration straightforward.

We perform a detailed analytical study to demonstrate the benefits and performance improvements of our proposed protocol. Further, we investigate the impact of node density on the protocol performance by changing the cell size for a fixed number of nodes. The contributions of this chapter can be summarized as follows:

- A wireless SDN routing framework is proposed by introducing the idea of MD2D routing as an extended service for cellular networks.
- VARP-S, a hybrid MD2D routing protocol, is proposed that operates based on source routing.
- Details of the proposed routing protocol (VARP-S) are described for the VARP framework in three sections: topology discovery, route discovery, and route maintenance.
- A detailed analytical model for VARP-S is presented to model its computational and routing overhead and compare its performance with the hybrid SDN architecture for WDNs (HSAW).

The remainder of this chapter is organized as follows. Section 3.2 provides a detailed description of the proposed virtual ad hoc routing framework. Section 3.3 describes the proposed routing protocol for the VARP framework and details of its operation. Section 3.4 provides an overview of HSAW. Section 3.5 describes the scenarios and defined notations for the theoretical analysis. Sections 3.6 and 3.7 develop a detailed theoretical analysis of the computational and routing overhead of the proposed routing protocol and compare it with the overhead analysis of hybrid SDN architecture for WDNs [3] in static and mobile networks, respectively. Finally, Section 5.7 presents concluding remarks of this chapter.

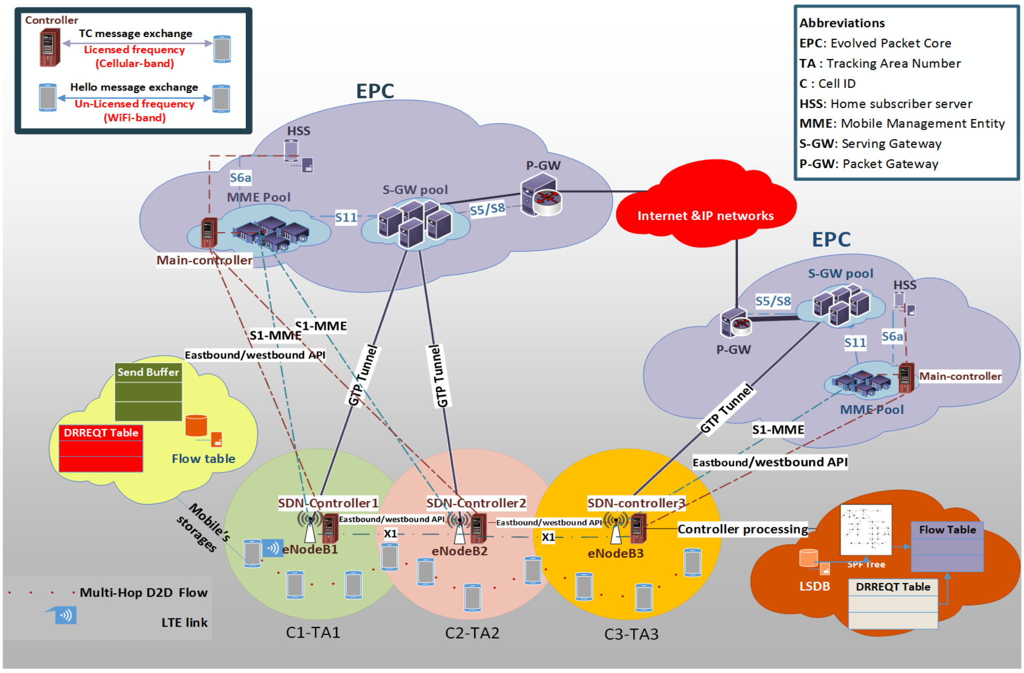


Figure 3.1: Detailed illustration of the VARP framework for the proposed VARP-S routing protocol.

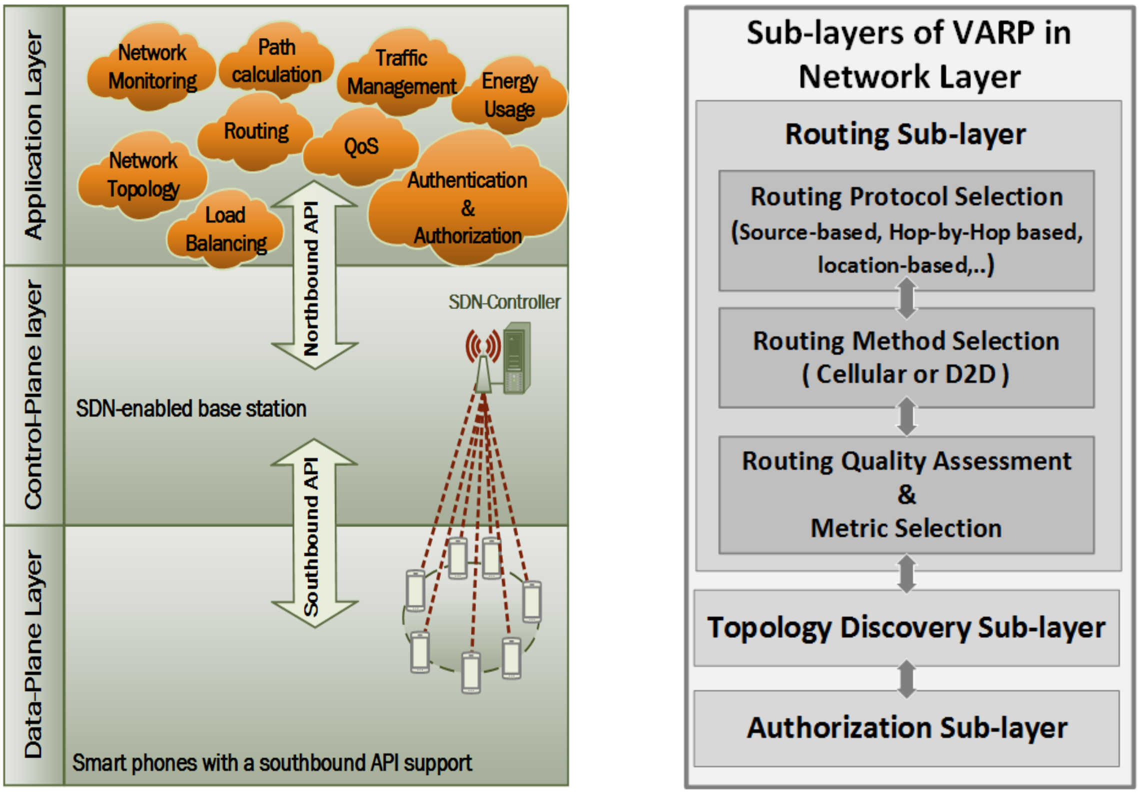
3.2 Proposed Virtual Ad hoc Routing Framework

This section presents a detailed description of the VARP framework that can support the implementation of various types of MD2D routings, such as source-based, hop-by-hop, or location-based routing. Further, the VARP-S routing protocol is proposed that extends the idea of source routing for an SDN-controlled MD2D network. It is noted that throughout this chapter, we use the terms "UE" and "node" interchangeably.

3.2.1 VARP Architecture

One of the key factors of the VARP architecture that distinguishes it from purely ad hoc routing protocols and conventional cellular architecture is the integration of SDN (see Figure 3.1).

An LTE architecture consists of two main components: an evolved universal terrestrial radio access network (E-UTRAN) and an evolved packet core (EPC). E-UTRAN includes an e-NodeB entity responsible for all radio-related functions and the connectivity of UEs to the EPC. EPC is composed of five main entities, which are described as follows: 1) the mobile management entity (MME) is responsible for tracking the idle UE positions, allocating a temporary unique ID to UEs, serving gateway selection for UE based on the network topology, and UE authentication; 2) the home subscriber server (HSS) entity maintains subscriber information, its authorized services, and its current MME address to have up-to-date information of a UE position; 3) serving gateway (S-GW) is responsible for redirecting data flows between the eNodeB entity and packet gateway (P-GW), and handover operations; 4) the P-GW entity is responsible for communication to/from the external data network, connectivity assurance, IP allocation, and billing-charging support; and 5) the policy and charging rules function (PCRF) entity is used for policy and charging functions.



(a) (b)

Figure 3.2: (a) SDN architecture in VARP framework, (b) VARP sub layers.

In our proposed framework, an SDN controller (acting as the main controller) is located in the MME entity to manage a set of sub-controllers. A sub-controller is an SDN controller that is directly connected to a base station to manage UE within its coverage area (VARP services are authorized for the subscribers who are accepted as relay nodes). The sub-controllers are identified by a unique ID assigned by the main controller. Using a sub-controller for each cell provides scalability and independency in the network infrastructure and enables intelligent local forwarding decisions within each cell. Each main controller manages only the sub-controllers located in a tracking area[1] of the MME. When a UE moves to the tracking area of other controllers, its information is delivered from the current controller to the new controller. Mobile operators install sub-controllers for high-traffic areas to offload traffic and balance the network load.

Both licensed and unlicensed frequency bands are used in this framework: the cellular frequencies for exchanging data and control information between the sub-controller and UEs under its coverage (Cellular-band), and the unlicensed frequencies for exchanging data and control information between the mobile or end user nodes in the network (WiFi frequency band). In our proposed architecture, each UE plays two different roles: an end user from the perspective of LTE and a forwarding element from the controller perspective. To that end, it must support a Southbound API compatible with this architecture and at least two wireless interfaces: LTE and WiFi. For the second role, each UE sends information about its directly connected links to the sub-controller. It

[1]Tracking areas are the areas that subscribers move freely within without informing MME.

can also request different routing information from the sub-controller. For example, in VARP-S (described in Section 3.3), the UE requests the path information to transmit data to a specific destination. The UE then uses such information to build an MD2D routing architecture over unlicensed WiFi frequency bands, such as WiFi Direct.

The SDN controller uses the received control information to build a link-state database (LSDB). It is assumed that the sub-controller has enough space to store the route information but may offload processing to a cluster (if required) to perform various operations such as route calculation and topology control. Edge servers under the control of the SDN controller can also be used in each base station for local data storage, real-time data processing, and computations. Figure 3.2a presents a logical view of an SDN architecture in the VARP framework.

Figure 3.2b indicates the sub-layers of the VARP architecture in the network layer. In the authorization sub-layer, the SDN controller via the HSS entity in the core network checks if UEs are authorized for using VARP services. The topology discovery sub-layer periodically communicates with each UE to build and maintain an LSDB (described in Section 3.3.1). The routing sub-layer manages the routing process in the network.

The routing sub-layer consists of three sub-layers: route quality assessment & metric selection (RQAMS), routing method selection (RMS), and routing protocol selection (RPS). The RQAMS verifies the type of end-to-end route quality required for each specific local data flow (described in Section 3.2.2) and selects the associated metric. After determining the traffic type of the data flow, the RMS decides which routing method can meet the quality requirements of this traffic type. If MD2D is selected, then the RPS determines which routing protocol to run on the UE. For example, a video or voice-based data flow would have different performance requirements than IoT-based or text-based data flows. The idea here is to determine whether WiFi-band MD2D routes can successfully meet the flow requirements. If they do, they are used. Otherwise, the sub-controller uses the cellular network to interconnect end-to-end nodes. Furthermore, depending on the availability of MD2D routes (which meet the minimum RQA requirements), the sub-controller may dynamically switch between MD2D and cellular modes. This ensures that traffic offloading does not jeopardize the quality of data packet flows.

Compared to purely ad hoc networks, there are a number of benefits provided by this architecture: security (authentication/registration of devices), elimination of multi-hop flooding, and UE can operate as a simple forwarding device (in the case of VARP-S) or a routing device.

Compared to conventional cellular networks, this architecture provides higher data rates for mobile operators through the efficient use of both licensed and unlicensed frequency bands. Further, this architecture allows operators to propose a new business model in which mobile users can take advantage of discounts on their monthly bills or free services offered by mobile operators based on the amount of relayed data [129].

3.2.2 Route Quality Assessment

The VARP architecture classifies each type of traffic flow into different groups with different QoS requirements. While it is envisioned that emerging wireless networks (such as 5G) will carry a diverse range of traffic types, for simplicity, we have defined three traffic types[2], namely mission-critical (MC), multimedia, and non-critical data (NC). Mission-critical traffic requires the highest level of QoS (in terms of link reliability, bandwidth, and delay). Multimedia traffic has strict bandwidth and delay requirements, with limited tolerance for link reliability. Non-critical data (such as periodic sensor data) do not have strict bandwidth requirements and can tolerate some delays in end-to-end transmission.

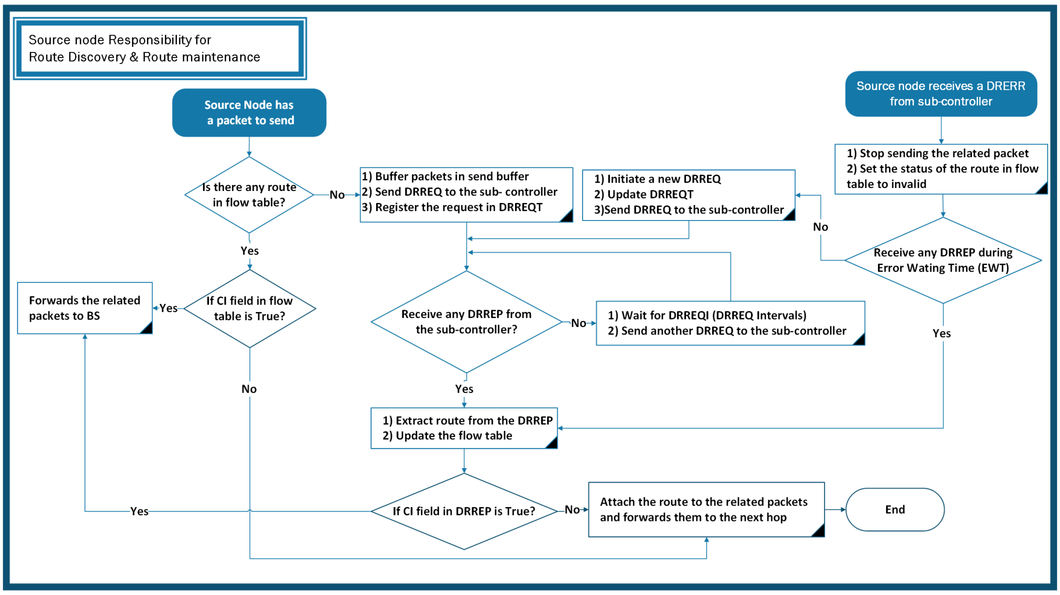


Figure 3.3: Source node responsibilities for route discovery and route maintenance in VARP-S.

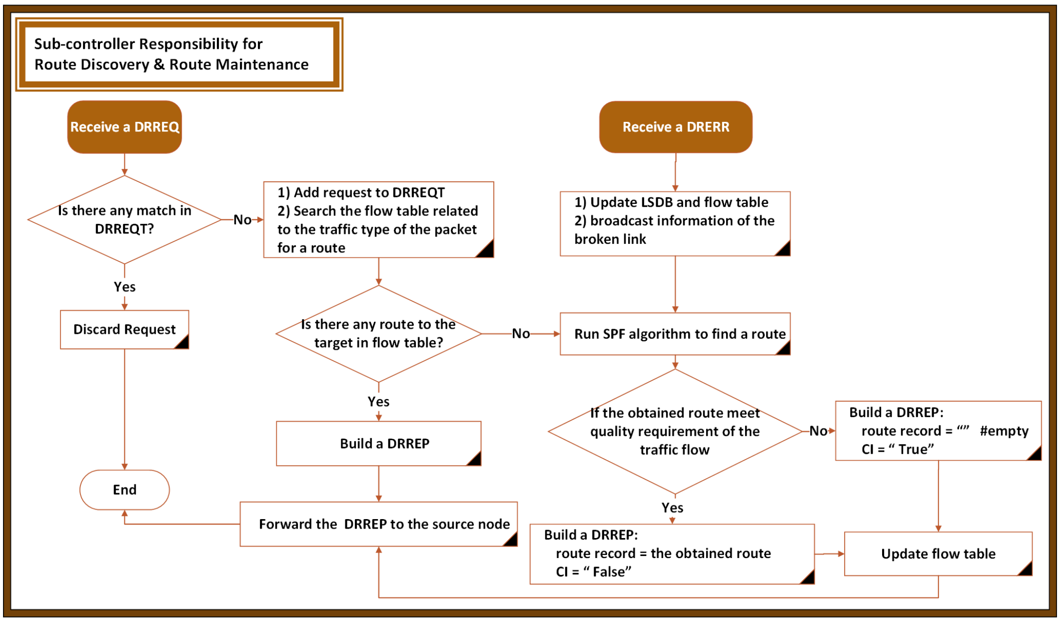


Figure 3.4: Sub-controller responsibilities for route discovery and route maintenance in VARP-S.

[2]Defining a detailed traffic classification strategy and QoS is beyond the scope of this chapter.

3.3 Routing Protocol for VARP Framework

This section describes the different components of the proposed VARP-S routing protocol in three sub-sections: topology discovery, route discovery, and route maintenance. The roles of mobile nodes and the sub-controller in route discovery and route maintenance are depicted in Figures 3.3 and 3.4.

3.3.1 Topology Discovery

We define three control messages in the VARP-S protocol:

- **The Hello message** is a broadcast packet carrying the node interface information, and only one-hop neighbors of the node process this packet. Each VARP-S interface of a node must generate Hello messages independently.
- **The topology control (TC) message** is a unicast packet carrying node information base (IB)[3] and is only received and processed by the sub-controller. The mobile node first is authenticated by its operator. The sub-controller then checks whether the mobile node is authorized to use the VARP-S services. If true, the sub-controller sends a topology control request (TCREQ) to the node and asks for the node's IB. The node then runs the VARP-S protocol and sends its IB to the sub-controller in response. Therefore, the first TC message is sent when a node receives a TCREQ from the sub-controller. Later TC messages are generated whenever a change occurs in the node's IB. In other words, a TC message is generated through a response mechanism. For example, if the remaining power of the node reaches the predetermined threshold, referred to as the node power threshold (NPT), then a TC message is sent to the sub-controller. The optimization of power control is beyond the scope of this chapter and is planned for future studies.
- **The Hello-C message** is exchanged only between the sub-controllers in adjacent cells to exchange identification (Controller-ID) and routing information. Each sub-controller maintains two different LSDBs: Inter-LSDB contains the details of the local topology (sub-controller and nodes of a given cell) and Intra-LSDB includes the information of neighboring cells. Hello-C messages allow adjacent sub-controllers to exchange Inter-LSDBs. After receiving the Inter-LSDB, the Intra-LSDB is updated. Thus, each sub-controller only has knowledge of the nodes inside the cell and the adjacent cells. In addition to the Inter-LSDB, each sub-controller also sends the diameter of its VARP-S coverage. This enables the protocol to limit the routing scope and communication between the nodes inside the cell or adjacent cells.

Considering the defined traffic types in Section 3.2.2, the LSDB maintains three different costs for each link. Each cost is calculated based on the required parameters for each traffic type. The sub-controller runs the SPF algorithm for each cost separately, which leads to the construction of three different shortest path trees (SPTs) against the LSDB. When a sub-controller receives a data route

[3] IB includes node interface information, neighbor information, link information, and power information (remaining power of the node)

request (DRREQ), it checks the traffic type field in the request. Subsequently, the sub-controller searches the associated SPT to determine the best path. Finally, the selected path is added to the related flow table. Therefore, the sub-controller maintains three distinct flow tables for each traffic type and keeps multiple routes to each destination to react much faster to routing changes.

Figure 3.5 shows the sub-controller operation with LSDB links. For each link, the LSDB maintains three different costs associated with MC, Multimedia, and NC traffic types defined in Sec. 3.2.2 (cost1, cost2, and cost3). For root node A, three different SPTs are derived from running the shortest path first (SPF) algorithm for each cost, as shown in Figure 3.5. SPT1 includes only path A_F_C_D, as this path is the best path for cost1. SPT2 and SPT3 maintain paths A_H_D and A_-B_C_D, respectively. If the sub-controller receives a DRREQ from source node A to destination D, it first checks the traffic type field in the request. If the type is "MC", it searches only the SPT1 and adds the least-cost path into the flow table 1.

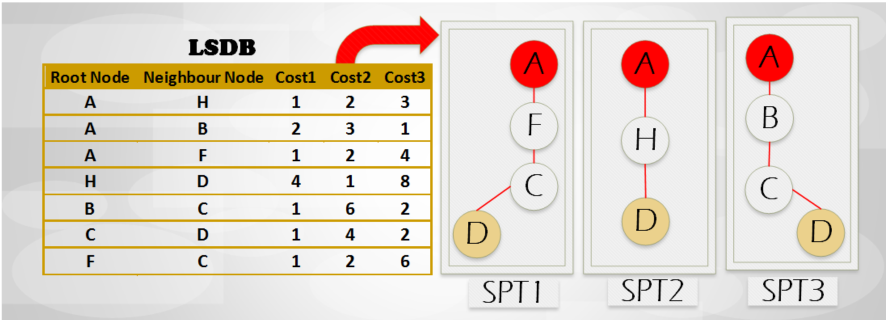


Figure 3.5: Illustration of building three different SPTs for source node A based on the defined traffic types in Section 3.2.2.

In our framework, data traffic is conveyed through the cellular or the MD2D network. Hence, the sub-controller must decide which communication mode (i.e., cellular or MD2D) can better support a traffic flow. To this end, the sub-controller first determines the maximum diameter (the maximum number of hops and the physical distance between the source and destination) of the MD2D communications for the VARP-S routing protocol. Within this diameter, an MD2D data transmission (using the VARP-S routing protocol) provides the same or better performance than the cellular transmission. When the sub-controller selects a path for a data flow from the SPF tree, it checks whether the selected path meets the determined diameter. If yes, the sub-controller sends the path information to the source node of the flow. Otherwise, the sub-controller asks the source node to transfer the data via the cellular. By examining and restricting the number of hops based on the bandwidth and QoS requirements, the sub-controller will increase the overall quality of WiFi-band MD2D communications.

3.3.2 Route Discovery

VARP-S is a source routing protocol[4] in which each data packet carries the complete source to the destination path. Hence, relay nodes can forward the packet based on the attached route information. Two important components of this protocol are route discovery and route maintenance.

[4] VARP-S extends some of the routing concepts described in [130] for multi-hop D2D communication.

Route discovery is an on-demand process that uses data route request (DRREQ) and data route reply (DRREP) messages. These messages are exchanged between the source node of a flow and the sub-controller to obtain a valid route to the flow target. The source node and sub-controller responsibilities in route discovery are described in the following subsections:

3.3.2.1 Source Node

When a node has data to send, it first checks its flow table. If there is a route to the packet destination, the source node attaches the route to the packet and forwards the packet to the next hop. Otherwise, the source node generates and sends a data route request (DRREQ) as a unicast packet to the sub-controller. The source node keeps a copy of the packet in its local storage called "send buffer". As long as there are packets in the send buffer, the source node sends DRREQ to the sub-controller to obtain a valid route to the packet destination. To prevent flooding of DRREQs for the same target address and reduce the overhead caused by DRREQs, a minimum acceptable interval is defined between DRREQs for the same destination, referred to as DRREQI (DRREQ Intervals). When the target is not reachable, the source node will receive a Data Route Error (DR-ERR) message with the type of "Destination_unreachable" from the sub-controller. Subsequently, the source node removes the associated packets to that destination from its send buffer.

The DRREQ packet carries the target address of the route discovery, a unique ID assigned to this request by the source node, and the size and traffic type of the data packet. Each node maintains a data route request table (DRREQT) to store the DRREQs originating from this node. Each node also has a flow table to keep the received routing information from the sub-controller.

After receiving a DRREP from the sub-controller, the source node extracts the Data Route Record (DRR) and updates its flow table. DRR includes the list of relay nodes for a data flow. The source node then attaches the DRR to related data packets and forwards the packets to the next hop. The source node also checks the reachability of the next hop through the route maintenance process, as described in Section 3.3.3. Each intermediate node receiving a data packet with a DRR attached examines the route record to verify whether it is the next hop for this packet. If a source node receives a DRREP with an empty route record and a Cellular Indicator (described in Section 3.3.2.2) with a value of "True", the source node then forwards the associated data packets to the base station and marks that route as a cellular route in its flow table.

3.3.2.2 Sub-controller

The sub-controller maintains a data route request table (DRREQT) for each active flow in the network. This table keeps the address of the node initiating the request, request-ID, and target address of that request. When the sub-controller receives a request from a node for a specific destination, the sub-controller adds the request to DRREQT if no match is found in the table. Otherwise, the request is discarded. After inserting the request into DRREQT, the sub-controller searches the flow table related to the request traffic type. If a route is found, the sub-controller initiates and sends a DRREP to the source node. Otherwise, the sub-controller runs the SPF algorithm for the source node to find the best path to the requested target.

A DRREP carries a list of relay nodes between the source and destination (defined as a DRR), target address, VN (validity number) indicating the freshness of the route, and cellular indicator (CI) field that accepts Boolean values. The value of this field is "True" when a data packet must be forwarded through LTE networks.

3.3.3 Route Maintenance

Route maintenance is the mechanism of error detection during the forwarding of data packets. This process consists of two parts: initiating a DRERR and processing a received DRERR. Each intermediate node in the path between the source and destination of a data packet is responsible for transmitting the packet to the next hop. Each hop confirms the reception of the packet through the link layer or passive acknowledgments.

If a node detects an error during the transmission of the data packet to the next hop, the node initiates a new DRERR containing Error Type[5] and all relevant information, and sends it to the sub-controller. Upon receiving the DRERR, the sub-controller updates the LSDB and broadcasts the information of the broken link. If a change occurs in the LSDB, the sub-controller updates the old entries of the flow tables, deletes any outdated ones, and replaces them with newly found routes. As a result, any time the allocated route to a source node is changed and replaced with a new one, the new route record is sent to the source node with the updated VN through a DRREP message.

After receiving the DRERR from the sub-controller, the source node stops sending the packets associated with that error and updates its flow table based on the DRERR information (sets the status of the route to invalid). The source node then initiates a new route discovery after the error waiting time (EWT). EWT is the time that a source node must wait to receive a DRREP from the sub-controller upon receipt of a DRERR. After this time, a new DRREQ is initiated with a new request ID.

3.4 HSAW Overview

In this section, we extend the behavior of the HSAW to fit the framework proposed in this chapter. In the HSAW protocol, like VARP-S, neighbor discovery involves the exchange of Hello messages. Each mobile node sends its link-state information (i.e., the TC message) to the sub-controller. Subsequently, the sub-controller builds an LSDB of its coverage area and broadcasts the LSDB to the network. As a result, all mobile nodes have a complete view of the network. The sub-controller also broadcasts traffic policies (such as traffic type, designed metric for each traffic type, and maximum allowed end-to-end MD2D cost for each traffic type considering data packet size) to the network. If a mobile node has a packet to send, it first checks its routing table to find a route to the packet target. If a route is found, then the mobile node forwards the packets to the next hop. Otherwise, the mobile node runs the SPF algorithm to determine the best path. If the obtained path meets the quality requirements of the packet, the route is added to the routing table,

[5] Error types: Link_Broken, Destination_unreachable, and Unknown_Error

and packets are forwarded to the next hop. Otherwise, the route to this target is marked as cellular, and packets are forwarded to the BS for cellular transmission. If any error occurs when sending data packets to the next hop, the upstream node of the broken link sends a DRERR to the sub-controller carrying information of the broken link. Subsequently, the sub-controller broadcasts the information of the broken link to the network and updates its LSDB. After receiving the error message, the mobile nodes update their LSDB and their routing tables accordingly.

3.5 Scenarios and Summary of Notations

We present two different scenarios for routing overhead analysis. The first scenario investigates the performance of VARP-S in a static network as this could be relevant to sensor networks and large-scale IoT networks. The second scenario explores VARP-S performance in mobile networks.

3.5.1 Notations

The key notations used in this chapter for the theoretical analysis are listed in Tables 3.1, 3.2, and 3.3. Table 3.1 covers the network parameters used for network abstraction. Tables 3.2 and 3.3 contain the parameters defined for overhead analysis in static and mobile networks, respectively.

$H(X)$ [133] is the entropy function of a discrete random variable X with possible values $\{x_1, ..., x_n\}$ occurring with probability $\{P(x_1), ..., P(x_n)\}$, which is defined as:

$$H(X) = -X log_2(X) - (1 - X)log_2(1 - X); \;\; 0 < X < 1$$

We refer to this entropy function for measuring the expected value of information needed to notify the next state of any link when a link change occurs in the network.

3.5.2 Assumptions

The routing overheads in this chapter are analyzed only for one sub-controller and all the UEs within its management area. It is assumed that each UE has two interfaces: one interface for WiFi-band communications and one LTE interface for cellular communication. It is also assumed that the mobile nodes have a uniform distribution in the cell — a square area of side a— and periodically send Hello packets and TC messages at the same intervals. For our analysis, we extended the defined framework in [132] to suit our scenarios. In the following sections, we analyze the routing overhead of HSAW and VARP-S for static and mobile scenarios, respectively. Considering the use of two separate frequency bands in both protocols, we analyze the routing overhead for each frequency band independently, namely, WiFi-band overhead (O_W) and cellular-band overhead (O_C).

Table 3.1: List of notations for network abstraction.

Notation	**Description**
A	Cell area (a square area of side a)
N	Number of the UEs in the cell (100)
R	Radio transmission range of each UE (250m)
ξ	Ratio of the UE coverage area to the whole network area: $\frac{\pi R^2}{A}$
N_D	Neighbor density (average UE's degree): $\xi\,(N-1)$
F	Number of active flows
UE_{ID}	UE identifier
ζ	Network topology update interval.
λ	Packet generation rate (2 pkt/s)
λ_u	Link break rate (s^{-1})
d	Average distance between the source and the destination. According to [131], the value of d is given by: $d = 0.5214a$
L	Average number of hops in a route. The lower bound of L is used for this study analyses. According to [132], the value of L is given by: $L \geq \lceil \frac{d}{R} \rceil$
S_D	Data packet size (512 Byte)

Table 3.2: List of notations for routing overhead analysis when network is static.

Notation	**Description**
LSDB	Length of a link-state data base
$L_{(stat)}$	Amount of information that describes the status of a link
$P_{(cost)}$	Path cost
M	Length of a Tc message
Req-ID	Amount of information that differentiates one request from the others (8 bits)
O_H	The overhead of broadcasting the Hello packets at time 0
O_{T_0}	The overhead of advertising the network link-states at time 0
$O_{T_{0U}}$	The overhead of distributing each node link-state to the sub-controller at time 0
$O_{T_{0C}}$	The overhead of distributing the whole network link-states by the sub-controller at time 0
O_{DRREQ_0}	The overhead of sending the route requests to the sub-controller at time 0 for a valid route to the target
O_{DRREP_0}	The overhead of sending route information by the sub-controller at time 0
O_{RD_0}	The overhead of the route discovery process at time 0
$O_{(W_0)}$	Total WiFi-band routing overhead per second at time 0
$O_{(C_0)}$	Total cellular-band routing overhead per second at time 0

3.6 Theoretical Analysis in Static Networks

The total routing overhead is based on the bandwidth consumed by the control messages. The exchange of the control messages in a pure ad hoc network is through WiFi-band frequencies. In contrast, in the VARP-S protocol and HSAW, two different frequency bands are used [6].

In both protocols, each UE broadcasts the Hello messages with a length of UE_{ID} through the WiFi-band frequency to advertise its existence to its neighbors. UE_{ID} is unique to each UE and differentiates the UE from other UEs in the network. If there are N nodes in the network, the total bandwidth consumed by the Hello messages is given by:

$$O_H = N\ UE_{ID},\ where\ UE_{ID} = Log_2 N \tag{3.1}$$

Each UE also sends a TC message to the sub-controller through cellular-band frequencies. Subsequently, the sub-controller computes the cost of each possible path in the network and builds the link-state database of the entire network using the received TC messages.

A TC message carries the information of the UE (UE_{ID}), its one-hop neighbors (N_D), and their associated link status ($L_{(stat)}$). Depending on how the UEs are identified in the network and the defined format for a TC message, the amount of information required for a TC message could be different. If the total number of UEs in the network is N, then the average number of one-hop neighbors for each UE can be estimated by "$\xi\ (N-1)$" (to study medium to large node density scenarios, we model the neighbor density of every node as "$\xi\ (N-1)$". This provides a good approximation of the neighbor density [134]). Consequently, the length of a TC message (M) generated by one node in the network at time 0 is:

$$M = UE_{ID} + N_D(UE_{ID} + L_{(stat)}) \tag{3.2}$$

In HSAW, each UE must have a complete view of the network and take all routing decisions. For this reason, the sub-controller must broadcast the entire LSDB at the beginning of the network operation. Subsequently, each UE builds its LSDB and runs the SPF to determine the routes. Unlike HSAW, in VARP-S, only the sub-controller has a complete view of the network and manages all the routing computations and routing decisions. Therefore, the LSDB is not broadcast by the sub-controller. If a node has a data packet, it sends a DRREQ to the sub-controller and asks for a path to the destination, as explained in Section 3.3.2.

The routing overhead of both protocols is analyzed for two different scenarios. In the first scenario, it is assumed that there is no data packet generation in the network. In the second scenario, it is assumed that the data packet generation rate is λ. As explained in detail in the following sections, the overhead of HSAW remains unchanged in both scenarios, but there is an additional overhead for VARP-S in the second scenario compared to the first scenario. This is due to the exchange of DRREQs and DRREPs between UEs and the sub-controller for a valid route to the destination. However, in HSAW, each UE requires more memory and consumes more energy compared to the

[6]WiFi frequency for links between the UEs (WiFi-band) and the cellular frequency for the links between the sub-controller and the UEs (cellular-band)

UEs in VARP-S as UEs have to keep the LSDB of the whole network, and they must run the SPF algorithm to find the shortest path when a route does not exist for an incoming packet.

3.6.1 HSAW Protocol

Each UE in the network sends a TC message with length M only to the sub-controller. After receiving all the TC messages, the sub-controller builds and broadcasts the LSDB to the network. The total consumed bandwidth of the cellular-band for the topology updates (O_{T_0}) at time 0 for the N number of nodes is a combination of the consumed bandwidth by both the UEs ($O_{T_{0U}}$) and the sub-controller ($O_{T_{0C}}$) and is given by:

$$\mathit{LSDB} = N\ N_D\ (2\mathit{UE}_{ID} + P_{cost}) \tag{3.3}$$

$$O_{T_{0U}} = NM \tag{3.4}$$

$$O_{T_{0C}} = \mathit{LSDB} \tag{3.5}$$

$$O_{T_0} = O_{T_{0U}} + O_{T_{0C}} \tag{3.6}$$

Because the exchange of Hello and TC messages are through the WiFi-band and the cellular-band frequencies, respectively, the total routing overhead of each frequency band at time 0 is:

$$O_{(W_0)} = \frac{1}{\zeta}(O_H) \tag{3.7}$$

$$O_{(C_0)} = \frac{1}{\zeta}(O_{T_0}) \tag{3.8}$$

3.6.2 VARP-S Protocol

Unlike HSAW, in this protocol, the sub-controller does not broadcast the LSDB. Consequently, the consumed bandwidth for the topology updates is the bandwidth consumed by the UEs and is $O_{T_0} = O_{T_{0U}}$, where $O_{T_{0U}}$ is defined by (3.4).

If there is no data packet generation in the network, the total consumed WiFi-band and cellular-band are calculated using (3.7) and (3.8), respectively.

If the UE has a packet to send, then there is an additional overhead for VARP-S in the cellular-band since the flow table of each UE is empty. Therefore, for forwarding the packets, a route discovery must be initiated to find a valid route to the destination of packets. If we define the number of flows as F, then the total number of DRREQs sent to the sub-controller is F. In response, the sub-controller returns F number of the DRREPs.

Each DRREQ carries a unique request ID (*Req-ID*) along with the target information. The bandwidth consumed by DRREQs in the network is:

$$O_{DRREQ_0} = F[\textit{Req-ID} + UE_{ID}] \tag{3.9}$$

In response to each DRREQ, the sub-controller sends a DRREP carrying the shortest path information and the target identity. If the minimum average number of hops in a route is L, then a DRREP message contains a path with L hops plus target information. As a result, the average length of a DRREP message would be “$(L+1)UE_{ID}$". The bandwidth consumed by the DRREPs at time 0 is:

$$O_{DRREP_0} = F[(L+1)UE_{ID}] \tag{3.10}$$

The total consumed bandwidth for the route discovery process at the beginning of the network (O_{RD_0}) is the bandwidth consumed by the DRREQs and DRREPs and is calculated as follows:

$$O_{RD_0} = \zeta(O_{DRREQ_0} + O_{DRREP_0}) \tag{3.11}$$

Consequently, the total overhead of VARP-S in the cellular-band at time 0 is increased as follows:

$$O_{(C_0)} = \frac{1}{\zeta}(O_{T_0} + O_{RD_0}) \tag{3.12}$$

The total WiFi-band overhead of VARP-S will also be increased because a DRR must be attached to the data packets before being forwarded to their target. If the extracted DRR from the received DRREP includes L hops, then the amount of information that must be attached to each data packet will be "$L\ UE_{ID}$". If we define the packet generation rate for each UE as λ, then the total WiFi-band overhead of VARP-S at time 0 is:

$$O_{(W_0)} = \frac{1}{\zeta}(O_H + \zeta\lambda\ F\ L\ UE_{ID}) \tag{3.13}$$

3.6.3 Result Analysis

In this section, we compare the overhead of VARP-S with HSAW for the two scenarios described in the previous section. The considered value for $L_{(stat)}$ and $P_{(cost)}$ is 1 byte. It is assumed that the λ packets generated by each node belong to the same target.

3.6.3.1 Routing Overhead in the WiFi-band at Time 0

When the network is static and there is no packet generation, the total WiFi-band overhead in both protocols is equivalent because of the constant broadcasting of the Hello messages.

Figure 3.6 presents the overhead of VARP-S compared to HSAW for a different number of nodes, where the network is static and the number of active flows is F. Figure 3.6 also shows how the number of active flows could affect the overhead of VARP-S. The considered values for the active flows are 10%, 50%, and 80% of the number of nodes, respectively. As shown in Figure 3.6, the overhead of VARP-S is more than HSAW and continues to increase when the number of active flows increases. The reason behind this is that in VARP-S, in addition to the Hello packets, the DRRs attached to the data packets consume the WiFi-band bandwidth. Figure 3.6 shows that by growing the number of mobile nodes in the network, the total overhead of both protocols increases accordingly.

3.6.3.2 Routing Overhead in the Cellular-band at Time 0

Figure 3.7 compares the routing overhead of VARP-S with HSAW for a different number of mobile nodes. As shown, VARP-S achieves higher scalability than HSAW. This is because HSAW introduces a significantly higher overhead resulting from broadcasting the entire LSDB by the sub-controller, and the overhead increases as the number of nodes grows.

In the following, we calculate the ratio of HSAW overhead to VARP-S overhead for two different scenarios to further study the performance benefits of VARP-S in static networks.

- The network is static and there is no active flow:

$$\frac{O_{C_{o(HSAW)}}}{O_{C_{0(VARP-S)}}} = \frac{NM + LSDB}{NM} \approx LSDB \tag{3.14}$$

- The network is static, and the number of active flows is F (the maximum value of F is $N(N-1)$):

$$\frac{O_{C_{o(HSAW)}}}{O_{C_{0(VARP-S)}}} = \frac{(3.8)}{(3.12)} \approx \frac{O(N^2)}{O(N^2)} \approx 1 \tag{3.15}$$

As shown in (3.14), the routing overhead of VARP-S is up to N^2 lower than that of HSAW when there are no active flows in the network. This is because in VARP-S, the LSDB is not broadcast by the sub-controller. From (3.15), it is apparent that in the worst-case scenario where all nodes communicate with each other, both protocols exhibit almost the same performance.

3.7 Theoretical Analysis in Mobile Networks

When the status of the network changes from static to dynamic, link failure is likely to occur more frequently. If a link failure occurs, additional overhead is added due to the misrouting.

For this analysis, it is assumed that nodes can move freely within the square area of side a, and the coverage area of each node is a disk-based covering area, with radius R. If the random variable $X(t) \in \{0, 1\}$ indicates the link status between nodes i and j at time t, where 1 and 0 represent the existence or non-existence of the link, respectively, when the distance between nodes is less or greater than R. As explained in [132], $q(t)$ is the probability of a link status change event within a time interval $[t, t + \zeta]$ when the value of $X(t)$ changes from 0 to 1 and vice versa, and is given by:

$$q(t) = 1 - e^{-\lambda_u t} \tag{3.16}$$

where λ_u is a constant average rate for the link change and represents the mobility in the network. A link change event can occur continuously and independently at the λ_u.

Table 3.3: List of notations for routing overhead analysis when the network is mobile.

Notation	Description
$q(t)$	The probability of link change event within a time interval $[t, t+\zeta]$
O_H	The overhead of broadcasting the Hello packets within a time step ζ
O_T	The overhead of advertising the network topology changes within a time step ζ
O_{T_U}	The overhead of distributing the changes of each node link-state to the sub-controller within a time step ζ
O_{T_C}	The overhead of distributing the network link-states changes by the sub-controller within a time step ζ
O_D	The overhead caused by unsuccessfully delivered data messages per packet generation
O_A	The overhead of distributing the broken link information of an active flow within a time step ζ
O_{A_U}	The overhead of distributing the broken link information by the upstream nodes to the sub-controller within a time step ζ
O_{A_C}	The overhead of distributing the broken link information of an active route by the sub-controller within a time step ζ
O_{DRREP}	The overhead of the sub-controller for updating the source node of data packets on new valid routes within a time step ζ
$O_{W(RI)}$	Total WiFi-band overhead incurred by the route invalidity within a time step ζ
$O_{C(RI)}$	Total cellular-band overhead caused by the route invalidity within a time step ζ
O_W	Total routing overhead in the WiFi-band per time step ζ
O_C	Total routing overhead in the cellular-band per time step ζ

In the following sections, the overhead analysis of HSAW and VARP-S are described separately for mobile networks.

3.7.1 HSAW Protocol

In this protocol, each UE broadcasts a Hello message to inform its existence to its neighbors, where the consumed bandwidth is given by (3.1).

Each UE also sends a TC message carrying information about itself and any link-state changes to the sub-controller within a time step ζ (O_{T_U}). Subsequently, the sub-controller broadcasts the changes to the entire network (O_{T_C}). If the probability of a link change event at time t is $q(t)$, according to the entropy function, the expected value of information informing the next state of the link is $H(q(t))$. If the average number of links for each node is N_D and the links have the same probability of the change, then the amount of information notifying the changes is given by:

$$O_{T_U} = N \sum_{k=0}^{N_D} \binom{N_D}{k} q(t)^k (1 - q(t))^{N_D - k} \, I \tag{3.17}$$

$$O_{T_C} = N \sum_{k=0}^{N_D} \binom{N_D}{k} q(t)^k (1 - q(t))^{N_D - k}\ I \tag{3.18}$$

where, $I = UE_{ID} + kH(q(t))$

The total cellular bandwidth consumed by the UEs and the sub-controller for updating the network topology within a time step ζ is $O_T = O_{T_U} + O_{T_C}$.

In the case of route invalidity, two more overheads will be added to the network: 1) the WiFi bandwidth consumed by data messages that have not been delivered successfully (O_D), and 2) the cellular bandwidth consumed for advertising the failed links of the active flows (O_A).

From [132], depending on which link has failed, the consumed bandwidth is:

$$O_D = FS_D[\frac{1 - (1 - q(t))^L}{q(t)} - L(1 - q(t))^L] \tag{3.19}$$

If a link failure occurs in the links of a data flow at time t, and the upstream UE of that link detects this error at the time of sending packets to the next hop, the upstream node sends the broken link-state information to the sub-controller (O_{A_U}) through a DRERR message. Subsequently, the sub-controller broadcasts this information to the entire network (O_{A_C}) through a DRERR message. As a result, the consumed bandwidth for advertising the broken link (O_A) is the bandwidth consumed by both the UE and the sub-controller. If the probability of change for each link at time t is $q(t)$, depending on the number of links that have failed in each flow (k) at time t, then the number of information exchanges between the UEs and the sub-controller, when the number of active flows in the network is F and the average number of links in each flow is L, would be:

$$O_{A_U} = F \sum_{k=1}^{L} \binom{L}{k} q(t)^k (1 - q(t))^{L-k} K[UE_{ID} + H(q(t))] \tag{3.20}$$

$$O_{A_C} = F \sum_{k=1}^{L} \binom{L}{k} q(t)^k (1 - q(t))^{L-k} K[UE_{ID} + H(q(t))] \tag{3.21}$$

$$O_A = O_{A_U} + O_{A_C} \tag{3.22}$$

The consumed WiFi bandwidth ($O_{W(RI)}$) and cellular bandwidth ($O_{C(RI)}$) due to invalid routes, within a time step ζ, are calculated as below:

$$O_{W(RI)} = \lambda\zeta(O_D) \tag{3.23}$$

$$O_{C(RI)} = \zeta(O_A) \tag{3.24}$$

Consequently, the total routing overhead in the WiFi-band (O_W) and the cellular-band (O_C) per time step (ζ) is given by:

$$O_W = \frac{1}{\zeta}(O_H + O_{W(RI)}) \tag{3.25}$$

$$O_C = \frac{1}{\zeta}(O_T + O_{C(RI)}) \tag{3.26}$$

3.7.2 VARP-S Protocol

In this protocol, as with HSAW, the exchange of Hello packets and data loss due to misrouting consume the WiFi bandwidth. The only difference is related to the path information (DRR) carried by the data packets. This information affects the amount of data loss overhead, as follows:

$$O_D = F(S_D + L(UE_{ID}))[\frac{1-(1-q(t))^L}{q(t)} - L(1-q(t))^L]$$

The total routing overhead in the WiFi-band within a time step ζ is defined by (3.25).

The consumed bandwidth for the topology updates is the bandwidth consumed by each UE for sending the TC messages to the sub-controller, and is:

$$O_T = O_{T_U} \tag{3.27}$$

where O_{T_U} is defined by (3.17).

In the case of route invalidity, in addition to O_A, there is an extra overhead for updating the source node of data packets about the new valid route (as explained in Section 3.3.3). In this case, the sub-controller sends a DRREP containing a new valid route (it is assumed that the average number of links in the new route is L) to the source node of the packet (O_{DRREP}).

The total cellular bandwidth consumed by the sub-controller and the UEs due to route invalidity within time step ζ is:

$$O_{DRREP} = F[1-(1-q(t))^L][L\ H(q(t))] \tag{3.28}$$

$$O_{C(RI)} = \zeta(O_A + O_{DRREP}) \tag{3.29}$$

The total routing overhead in the WiFi-band and the cellular-band per time step ζ are given by (3.25) and (3.26), respectively.

3.7.3 Result Analysis

When the network is mobile, the cellular-band is used for topology updates and link-failure updates. It is also used by the sub-controller (in VARP-S) to send a new valid route to the source node.

In the following equation, the ratio of the cellular-band overhead of HSAW to VARP-S is calculated to compare the performance of both protocols.

From (3.18) & (3.28), we have:

$$\frac{O_{C_{(HSAW)}}}{O_{C_{(VARP-S)}}} \approx \frac{O_{T_C}}{\zeta O_{DRREP}} \tag{3.30}$$

Equation (3.30) reveals that VARP-S achieves higher scalability and lower overheads compared to HSAW if the following inequality is met:

$$\zeta O_{DRREP} \leq O_{T_C} \tag{3.31}$$

Figure 3.8 compares the cellular-band overhead of VARP-S with HSAW for different mobility levels, where the number of nodes and the number of data flows are 100. It also outlines how the network topology update intervals (ζ) can affect the overhead of both protocols. The various parts of the overhead are analyzed separately in the following sections.

3.7.3.1 Topology-update Overhead Analysis versus Mobility

Figure 3.9 illustrates the topology-update overhead of VARP-S compared with HSAW for two different intervals (1s and 3s). As can be seen clearly from Figure 3.9, both protocols experience greater overhead for lower time intervals owing to frequent topology updates. Furthermore, HSAW introduces more overhead than VARP-S.

The ratio of the HSAW topology-update overhead to VARP-S is calculated as below. It can be seen from the ratio that the VARP-S topology-update overhead is always less than that of HSAW.

$$\frac{O_{A(HSAW)}}{O_{A(VARP-S)}} = \frac{O_{A_U} + O_{A_C}}{O_{A_U}} \approx O_{A_C} \tag{3.32}$$

3.7.3.2 Misrouting Overhead Analysis versus Mobility

To investigate the levels of overhead experienced in the network due to route invalidity, the misrouting overhead of VARP-S is compared to HSAW versus different levels of mobility, as shown in Figure 3.10. The overhead results are presented for two different intervals ($1s$ and $3s$). The plot for $1s$ indicates more overhead for VARP-S compared to HSAW, while the plot for $3s$ illustrates that the overhead of both protocols tends to stay at the same level for mobility higher than 4. This is because the overhead of O_{DRREP} is negligible for higher mobility, as depicted in Figure 3.11. Figure 3.10 also shows that the overhead of both protocols rises when the interval changes from $1s$ to $3s$.

By calculating the ratio of the misrouting overhead in VARP-S to HSAW as follows, it is obvious that the misrouting overhead of VARP-S is always greater than or almost close to the HSAW. This is due to the additional overhead introduced by the sub-controller to the network to send new valid routes to the source nodes.

$$\frac{O_{C(RI[VARP-S])}}{O_{C(RI[HSAW])}} = \frac{(3.29)}{(3.24)} = \frac{O_A + O_{DRREP}}{O_A} \approx O_{DRREP} \tag{3.33}$$

3.7.3.3 Total Overhead Analysis versus Mobility

Figure 3.8 is derived from the sum of Figures 3.9 and 3.10. HSAW experiences more overhead compared to VARP-S, especially for lower topology-update intervals because of the increased topology-update overhead. First, the overhead of both protocols increases rapidly to reach its highest level when the mobile nodes start moving in the network. Subsequently, by increasing the mobility, the overhead of the HSAW decreases slightly for a $1s$ interval, while its overhead tends to remain at the same level for a $3s$ interval. By increasing the mobility, the VARP-S overhead also tends to remain stable for both intervals. The peak value of the overhead differs depending on the selected interval and the mobility.

3.7.3.4 Total Overhead Analysis versus the Number of Active Flows

Figure 3.12 compares the cellular-band overhead of VARP-S with HSAW over a different number of active flows, where the coverage area of the nodes is 250m, the cell area is $10^6 m^2$, the topology-update intervals are 1 second, and the mobility (λ_u) is 1. The number of active flows is based on different percentages of the number of nodes in the network. Initially, it is assumed that the number of active flows is 100% of the total number of nodes in the network. Similarly, the number of active flows increases to 10 times the number of nodes. As explained previously, topology updates and route invalidity consume the cellular bandwidth. The topology update is performed periodically and its overhead is independent of the number of flows; conversely, the route-invalidity overhead is dependent on the number of flows. As can be seen in Figure 3.12, when the number of flows is zero, we only have topology-update overhead for both protocols. In this case, HSAW produces more overhead than VARP-S. This becomes more apparent as the number of nodes increases. After that, when the number of active flows gradually increases, the total overhead of both protocols increases slightly owing to route invalidity. Unlike the plot for 500 nodes, the plot for 100 nodes indicates more overhead for VARP-S compared to HSAW when the number of active flows exceeds 800.

It is noteworthy that the memory and energy consumption of the nodes in the HSAW are higher than those in the VARP-S. This is because in HSAW, each node needs to update its routing table periodically regardless of whether it is part of an active route. This will increase the energy consumption and CPU processing overhead due to the frequent SPF and route table recalculations. Unlike HSAW, nodes in VARP-S only maintain on-demand routes in their flow tables, and the sub-controller performs all routing calculations and required processing. Further, in HSAW, each node will have a higher memory overhead, as it keeps routes to all destinations. Conversely, in VARP-S, each node maintains only the active routes.

3.7.3.5 Overhead Analysis versus Node Density

For this analysis, we evaluate the node density by changing the cell area for a fixed number of nodes. Figure 3.13 depicts the overhead of VARP-S compared to HSAW for different cell areas ($10^6(m^2) - 16 \times 10^6(m^2)$), where the number of nodes is 500 and the number of flows is 500. As can be seen, VARP-S introduces lower levels of overhead compared to HSAW for different node densities. In both protocols, parameters related to the cell size (defined below), such as ξ and L can affect the level of routing overhead in the network.

$$\xi = \frac{\pi R^2}{a^2} \tag{3.34}$$

$$L = \lceil \frac{d}{R} \rceil \tag{3.35}$$

$$d = 0.5214a \tag{3.36}$$

$$(3.35)\ \&\ (3.36) \Rightarrow L = \lceil \frac{0.5214a}{R} \rceil \tag{3.37}$$

From (3.34) and (3.37), if we define a constant value for the coverage area of nodes (R), then expanding the cell size leads to a decrease in ξ and an increase in L. In other words, by expanding the cell size, the average number of hops in each flow increases, while the average number of neighbors for each node decreases.

As shown in Figure 3.14, by increasing the cell size, the topology-update overhead of both protocols decreases owing to the diminution of each node neighbors. This reduction is significant when the size of the cell side changes from 1000 m to 1500 m. Subsequently, the overhead will decrease slightly when the size of the cell increases. Figure 3.14 also shows the misrouting overhead of both protocols versus the cell area. By expanding the cell size, the average number of hops in a route increases accordingly. This increases the misrouting overhead in both protocols due to the increased average number of links in each flow.

Figure 3.13 is derived from the sum of the plots in Figure 3.14, and illustrates the total overhead of both protocols for various cell areas. As shown, the total overhead of both protocols sharply declines when the size of the cell side increases to 1500 m. Subsequently, the overhead decreases gradually when the size of the cell side increases to 2000 m. For larger cells, the overhead of VARP-S increases slightly. The HSAW overhead continues to decline slowly when the size of cell side changes from 2000 m to 2500 m. The overhead will gradually rise for areas with a cell side bigger than 2500 m.

In summary, it can be construed from (3.30) that the number of nodes, number of active flows, node density, the transmission range of mobile nodes, topology update intervals, and mobility are the factors that determine the efficiency of both protocols compared to each other.

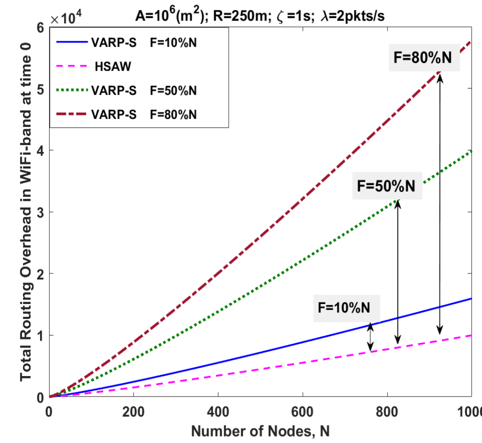


Figure 3.6: Total routing overhead in the WiFi-band at time 0 vs. Number of nodes and active flows.

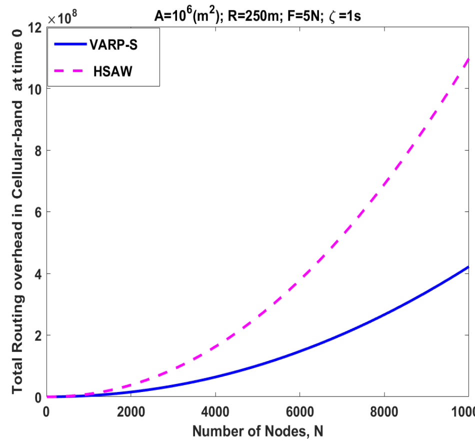


Figure 3.7: Total overhead in the cellular-band at time 0 vs. Number of nodes.

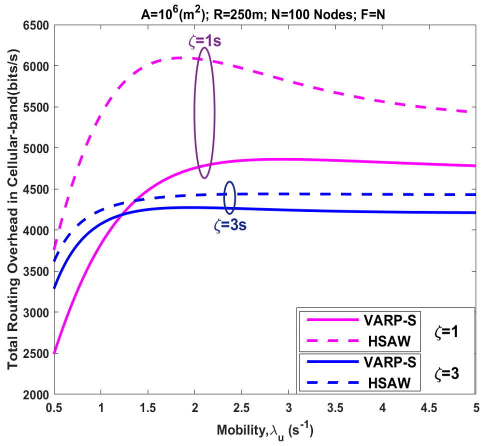


Figure 3.8: Total overhead in the cellular-band per time step ζ vs. Mobility.

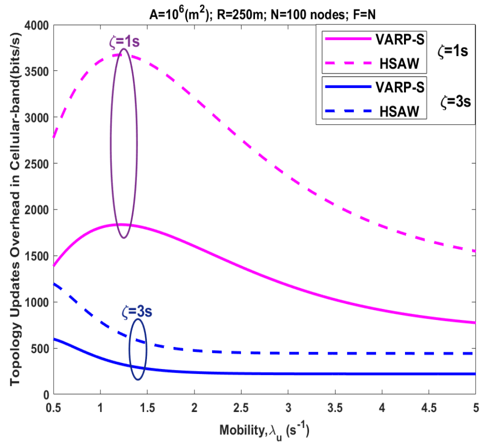


Figure 3.9: Topology-update overhead in cellular-band per time step ζ vs. Mobility.

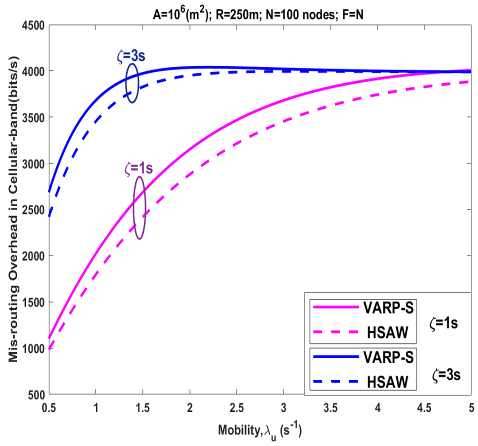


Figure 3.10: Misrouting overhead in cellular-band per time step ζ vs. Mobility.

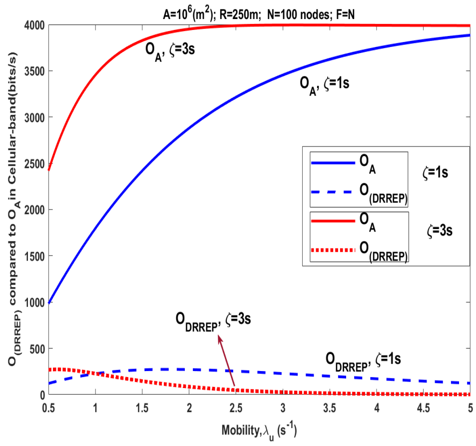


Figure 3.11: Overhead analysis of O_{DRREP} compared to O_A.

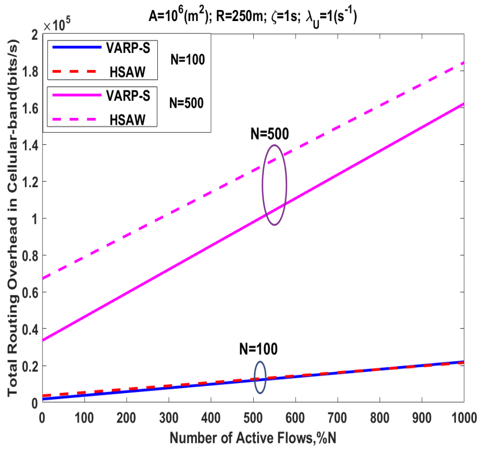


Figure 3.12: Total routing overhead in the cellular-band per second vs. Number of active flows.

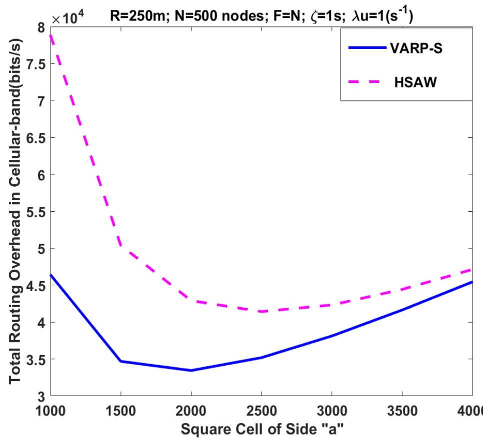


Figure 3.13: Total overhead in the cellular-band per time step ζ vs. Cell area.

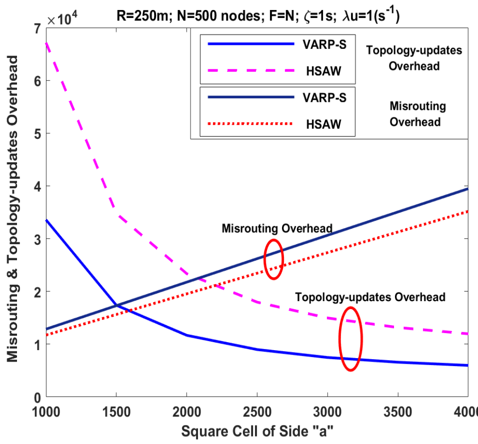


Figure 3.14: Topology-updates overhead and misrouting overhead in the cellular-band per time step ζ vs. Cell area.

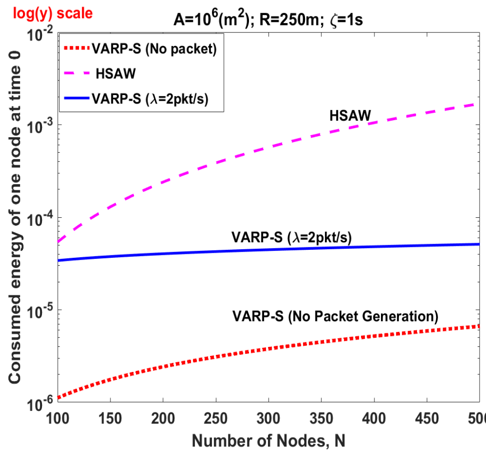


Figure 3.15: The consumed energy of one node vs. Number of nodes, where the network is static.

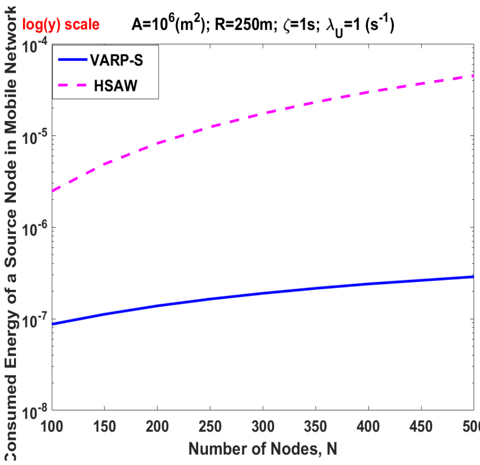


Figure 3.16: The consumed energy of a source node vs. Number of nodes, where the network is mobile.

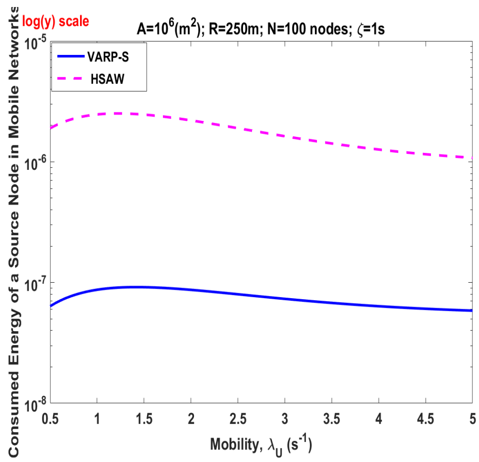


Figure 3.17: The consumed energy of a source node vs. Mobility, where the network is mobile.

3.7.3.6 Energy Analysis of the Nodes for the Routing Operations

In this section, we calculate the amount of energy consumed by each mobile node for the added routing functionalities. We employed the energy model used in [135–137] for our analysis. The energy consumed for transmitting or receiving packets of a given size is calculated as follows:

$$E_{Tx} = (Transmitting\ power(T_x) \times packet\ size)/BR \tag{3.38}$$

$$E_{Rx} = (Receiving\ power(R_x) \times packet\ size)/BR \tag{3.39}$$

where transmission/reception power is the consumed voltage and current for turning on the T_X/R_X circuits, and BR is the bit rate of the network interface.

It is assumed that all the UEs support LTE UE category 4, where the up-link and down-link rates for 20 MHz bandwidth are 150 Mbps and 100 Mbps, respectively [138]. If the maximum allowed transmission power for an LTE device is 0.2 W (23 dBm), then the maximum received power cannot exceed the transmitted power. For this reason, we consider an upper bound for the receiving power with a value of 0.2 W. It is also assumed that each UE is equipped with IEEE 802.11 network interface cards for WiFi connectivity. According to the 2.4 GHz Wave LAN implementation of IEEE 802.11, the values of T_x, R_x, and BR are 1.65 W, 1.15 W, and 2 Mbps, respectively [135–137].

In HSAW, each UE sends a TC message with a length of M to the sub-controller and receives an LSDB with a length of *LSDB* from the sub-controller. In VARP-S, each UE sends only a TC message with a length of M to the sub-controller. If a UE has a packet to send, then it sends a DRREQ to the sub-controller and receives a DRREP from the sub-controller with a length of $(Req\text{-}ID + UE_{ID})$ and $((L+1)UE_{ID})$, respectively. The UE also attaches a DRR with a length of $(L\ UE_{ID})$ to each data packet, whereas in HSAW, the UE must run the SPF algorithm to find the shortest path to the target. In addition, each relay node must calculate the route to the target.

The consumed energy for the Hello message is the same for both protocols and is ignored in the energy analysis. The amount of energy that the mobile nodes consume when they are in idle mode or hearing mode is also ignored in the analyses.

When the network is mobile, in HSAW, each UE sends a TC message with a length of $(\frac{1}{N}O_{T_U})$ to the sub-controller to inform about the changes in the status of its links. Each UE also receives a message with a length of $((N-1)\frac{1}{N}O_{T_C})$ from the sub-controller for the updated status of the links in the network. If an error occurs, the UE sends the broken link information to the sub-controller with a length of $(\frac{1}{F}O_{A_U})$ and receives the information of the broken links from the sub-controller with a length of $((F-1)\frac{1}{F}O_{A_C})$. In VARP-S, the UEs do not receive the topology updates from the sub-controller. Furthermore, each UE, after a failed route event, receives a DRREP with a length of $(\frac{1}{F}O_{DRREP})$ from the sub-controller if it is the source of data packets.

Table 3.4 analyzes the consumed energy of one node for static and mobile networks distinctly. Figures 3.15, 3.16, and 3.17 show that the consumed energy of a node for routing purposes is

Table 3.4: Energy consumption analysis.

The consumed energy of one node when the network is static (see Figure 3.15)		
Routing protocol	**HSAW** (energy for route calculation is ignored)	**VARP-S**
Energy for topology update	$E_{top} = \frac{0.2M}{50\times10^6} + \frac{0.2\ LSDB}{150\times10^6}$	$E_{top} = \frac{0.2M}{50\times10^6}$
Total consumed energy	$E_{Total} = E_{top}$	$E_{Total} = E_{top}$
The consumed energy of a source node in VARP-S, where the packet generation rate is λ (see Figure 3.15)		
Energy for route discovery	$E_{Rd} = \frac{0.2(Req\text{-}ID+UE_{ID})}{50\times10^6} + \frac{0.2((L+1)UE_{ID})}{150\times10^6}$	
Energy for forwarding data packets	$E_{FW} = \frac{1.65\lambda LUE_{ID}}{2\times10^6}$	
Total consumed energy	$E_{Total} = E_{top} + E_{Rd} + E_{FW}$	
The consumed energy of a source node when network is mobile (see Figs. 3.16 and 3.17)		
Routing protocol	**HSAW** (energy for route calculation is ignored)	**VARP-S**
Energy for topology update	$E_{top} = [\frac{0.2\frac{1}{N}O_{T_U}}{50\times10^6}] + [\frac{0.2(N-1)\frac{1}{N}O_{T_C}}{150\times10^6}]$	$E_{top} = [\frac{0.2\frac{1}{N}O_{T_U}}{50\times10^6}]$
Energy for receiving information of a broken link in the path of data packets	$E_{failed} = [\frac{0.2\frac{1}{F}O_{A_C}}{150\times10^6}]$	$E_{failed} = [\frac{0.2\frac{1}{F}O_{A_C}}{150\times10^6}]$
Energy for receiving a DRREP from the sub-controller	—	$E_{DRREP} = \frac{0.2\frac{1}{F}O_{DRREP}}{150\times10^6}$
Total consumed energy	$E_{Total} = E_{top} + E_{failed}$	$E_{Total} = E_{top} + E_{failed} + E_{DRREP}$

significantly higher for HSAW compared to VARP-S, especially for a higher number of nodes. Figure 3.15 shows that the energy usage of a node in the static networks increases when the number of nodes grows. It also shows that if a node has data to send, it consumes more energy in VARP-S as it has to perform the route discovery. For mobile networks, by increasing the number of nodes, the energy consumed by a node will increase accordingly, as depicted in Figure 3.16. Figure 3.17 indicates that the energy consumed by a node increases when the mobility increases to 1.5. Subsequently, by increasing the mobility, the energy usage of each node decreases accordingly.

3.8 Conclusions

This chapter presented a new routing framework for multi-hop device-to-device (MD2D) communications in SDN-based wireless networks. The proposed routing framework takes advantage of routing virtualization and software-defined networking to create highly scalable MD2D networks within and between each SDN cell. We proposed a source-based routing protocol, referred to as VARP-S, on top of this framework. We conducted detailed analytical and numerical studies comparing VARP-S with our previously proposed hybrid SDN architecture for WDNs (HSAW) [128]. Our results show that the proposed routing protocol achieves higher network scalability and lower power consumption for mobile nodes when compared to HSAW.

4

A Routing Protocol for SDN-based Multi-hop D2D Communications

4.1 Introduction

Current 4G networks suffer from a several limitations, such as the complexity of control-plane protocols, lack of fine-grained control for load balancing, poor support for virtualization, scalability issues resulting from centralizing data-plane functions in the packet gateway entity, inflexibility and high cost of equipment, and the vendor-specific configuration of interfaces [94] [95]. SDN aims to address such challenges by providing features such as real-time monitoring, a common control interface for various cellular technologies, distributed enforcement of QoS and firewall policies, efficient resource allocation, and support for network virtualization in cellular networks [95]. Consequently, several SDN-based architectures have been proposed for future 5G networks in which one or more components of wireless networks are virtualized. The existing literature has shown how SDN and network function virtualization (NFV) can improve network programmability, flexibility in deploying new services and applications, scalability, resource management, energy efficiency, capacity, quality of services (QoS), end-users quality of experience (QoE), and performance of 5G networks. This is achieved by centralizing the control functions and separating the control and data planes. SDN enables mobile operators to control and manage networks simply and intelligently.

There are also several studies on D2D communications under the control of an SDN controller [139] [140] in which forwarding was defined in the software. These studies showed that SDN-based routing services can achieve better scalability and packet delivery ratio compared to traditional ad hoc routing protocols due to the faster response of the controller to network topology changes. Further, the studies indicated that SDN-based multi-hop routing improves data usage and energy consumption by offloading the bandwidth from cellular networks.

In our previous study [3], we proposed an SDN-based hybrid architecture for wireless distributed networks (HSAW), where network control signaling and data forwarding were separated using two different frequency bands, namely in-band for p2p data forwarding (unlicensed frequencies) and out-of-band for point-to-multipoint control signaling between the controller and mobile nodes (licensed frequencies). Our study indicated that the proposed architecture increases the scalability and reliability of WDNs by eliminating the need for multi-hop flooding in route discovery. This was achieved by splitting the complexity of the routing functionalities between an SDN controller and mobile nodes. However, HSAW still requires the SDN controller to flood a complete LSDB to all end-user nodes, limiting scalability as the number of nodes increases in the network. This chapter presents a new MD2D routing protocol named SMDRP (SDN-based multi-hop D2D routing protocol) for SDN-based cellular networks, eliminating the need for broadcasting LSDB by the controller. In SMDRP, only the next-hop information is required for routing the data packets. Our proposed protocol is suitable for a large density of mobile pedestrian communications, where mobile nodes want to share or exchange content (for example, inside large sports stadiums, shopping centers, exhibitions, or events).

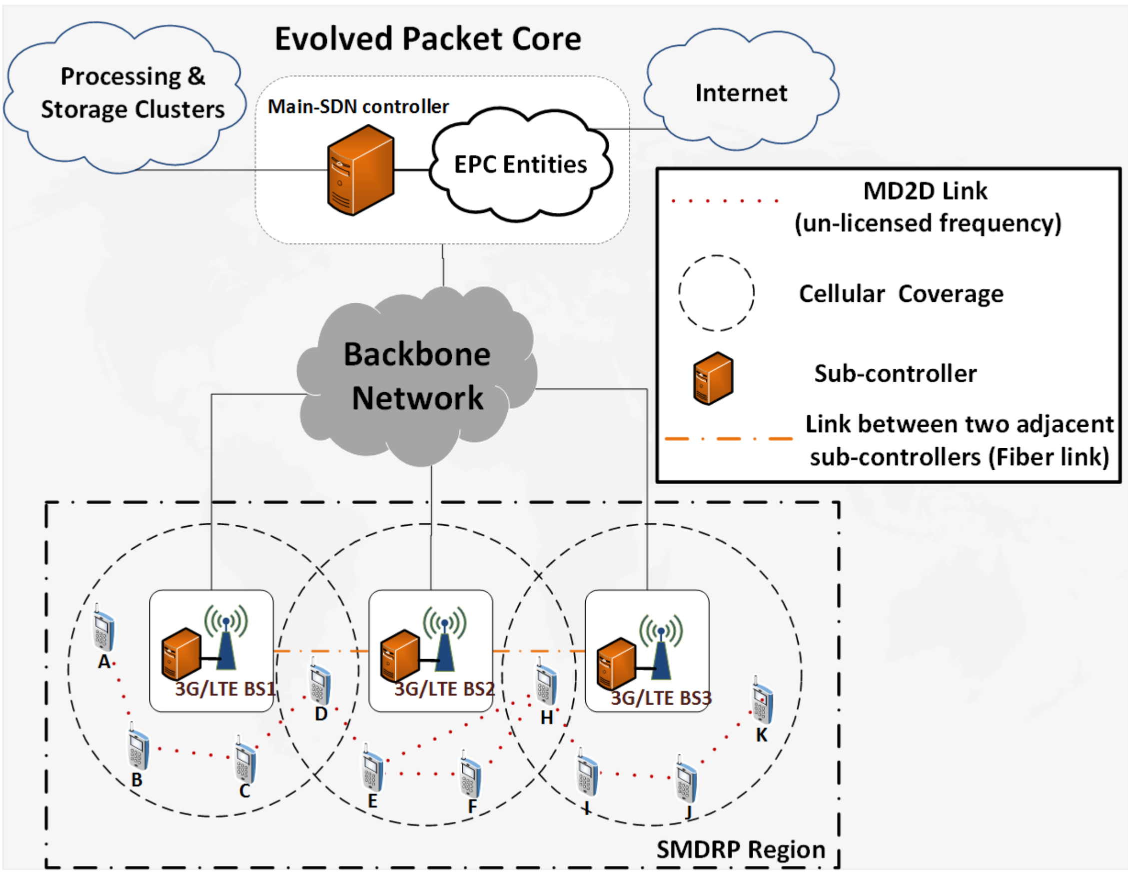


Figure 4.1: Proposed framework for an SDN-based 5G network.

4.2 Proposed SDN-based 5G Framework for SMDRP

Figure 4.1 indicates a framework that introduces the idea of MD2D routing under the control and management of an SDN controller. An SDN controller, referred to as a sub-controller, is connected to each base station (BS) or access point (AP) in this framework. Each sub-controller manages the data forwarding of the mobile nodes under its coverage area. The sub-controllers are connected to the cellular core network (Evolved Packet Core (EPC)) to connect to the Internet or other wireless networks. There is a main SDN controller in the EPC that manages the sub-controllers under EPC coverage. The main controller assigns a unique Id to each sub-controller defined as CR-Id. The main controller manages the access of sub-controllers to the processing and storage clusters (if needed) for advanced computations, real-time data processing, content caching, and data storage. Each mobile node is equipped with at least two interfaces: LTE (cellular frequency) for connecting to the sub-controller and WiFi (unlicensed frequency) for MD2D communications. Each mobile node sends its link-state information to the sub-controller via cellular frequencies. The sub-controller makes forwarding decisions based on the information received from the mobile nodes. When a mobile node initiates a data flow, it receives instructions from the sub-controller for forwarding data flows. Based on the sub-controller traffic policies, the forwarding of the data packets can be through a BS or MD2D. Different network applications can run on the sub-controller for topology discovery, load balancing, customized routing, traffic classification, and resource allocation.

4.3 SMDRP for SDN-based 5G Networks

SMDRP is a hop-by-hop routing protocol in which data packets carry only a flow Id, referred to as F_{Id}, and each intermediate mobile node between the source and destination of data packets acts as a forwarding device. The relay nodes forward data packets based on their flow table information. The sub-controller makes forwarding decisions based on the traffic type attached to the route request by the source of the flow. Because each mobile application can generate different flows with different QoS requirements and security/resource policies [141], the sub-controller allocates different metrics to each traffic type based on their quality and security requirements. When a mobile node associates with a BS, after authentication by the sub-controller, it runs the MD2D routing protocol if authorized to use advanced routing services. Subsequently, the mobile node informs its existence to its neighbors by broadcasting a Hello message from each of its interfaces except the LTE one. The mobile node also sends a topology control (TC) message from its LTE interface to the sub-controller. The TC message includes the information that helps the sub-controller manage MD2D communications, such as link-state information (list of one-hop neighbors and associated link status), position, and battery level. The next TC messages will be sent to the sub-controller if any changes occur in the link-sate information or the status of the mobile node (for example, when the battery level of the mobile node falls below a threshold, it must inform the sub-controller about the battery status). Based on the received TC messages, the sub-controller can make its Inter-LSDB (local topology). The adjacent sub-controllers exchange their Inter-LSDB to make the Intra-LSDB (neighbors topology). Consequently, each sub-controller has a global view of its cell and its adjacent cells. The sub-controller also categorizes traffic flows and assigns a specific metric for each traffic type. It calculates the cost of each link based on the defined metrics and maintains different costs for each link in its LSDB. The following pseudocode (see Algorithm 1) highlights the way in which the controller select the forwarding path based on the traffic type. It is assumed that the controller classifies the traffic into three categories, referred to as TF_1, TF_2, and TF_3. It is also assumed that the M_1, M_2, and M_3 metrics are defined for each traffic type, respectively. Based on the defined metrics for each traffic type, cost matrices are built, referred to as $\mathbf{C_{TF_1}}$, $\mathbf{C_{TF_2}}$ and $\mathbf{C_{TF_3}}$.

Algorithm 1 Find the least-cost path based on the traffic type of a packet

Input

S: Index of source node, $1 \leq s \leq n$, where n is the total number of nodes
D: Index of the destination node, $1 \leq t \leq n$
$TF_i, i = 1, ..., N_{tf}$, where TF_i is the traffic type of a packet, i is the index of the traffic type, and N_{tf} is the number of defined traffic types (3 in our example)
$\mathbf{C_{TF_i}}$: Cost graph for the TF_i

Output

$BestPath$: The least-cost path from a source node to a destination node based on the traffic type of a packet.

```
procedure FIND_PATH(TF)
    if TF ∈ {TF_1, TF_2, ..., TF_{N_tf}} then
        BestPath = Dijkstra(S, D, C_TF);
        return BestPath;
    else
        return Undefined Traffic Type!;
    end if
end procedure
```

4.3.1 Route Discovery in SMDRP

The route discovery process includes two sub-processes: sending a flow request to the sub-controller for a valid route to the target of data flow and receiving a flow reply from the sub-controller informing about a valid route to the requested target. Each mobile node running SMDRP maintains two tables: a flow table (FT) to keep the received forwarding rules from the sub-controller and a neighbor table (NT) to maintain the information of its one-hop neighbors. The sub-controller assigns a unique Id to each active flow, referred to as F_{Id}. When a mobile node has a data packet to send, it initially checks its flow table to find a valid route. If a route is found, the mobile node attaches the assigned F_{Id} to the data packets and forwards the packets to the next hop. Otherwise, the mobile node keeps a copy of packets in its send buffer and forwards a flow request message (FREQ) to the sub-controller for a valid route. A FREQ message carries the requested target address or Id, a unique request number (Req_Num) that has been allocated to this request by the source node, and the traffic type of the flow. The source node also registers its request in a flow request table (FREQT).

The sub-controller keeps two tables, namely the forwarding information base (FIB) and flow request table (FREQT). The former keeps information of active flows, while the latter maintains the information of flow requests sent by the source of data flows. After receiving a FREQ from a mobile node, the sub-controller first checks its FREQT to find a match. If no match is found, then the sub-controller adds the request to the FREQT. The sub-controller discards requests for the same target with the same Req_Num and traffic type. After updating FREQT, the sub-controller adds an entry for this F_{Id} into its FIB. The sub-controller runs a least-cost-path algorithm (for example, Dijkstra's algorithm) to find a least-cost route from the source node to the target. In this stage, the sub-controller checks whether the selected path can provide equal or better quality requirements for this data flow compared to the cellular data transmission. If it does, the value of the cellular indicator (CI) in the FREP is set to False, and the selected path is placed in the FREP. Otherwise, the sub-controller sends a unicast FREP message to the source node with the CI value of True. A FREP message is a message from the sub-controller to the network, informing the involved nodes of an active flow about the forwarding path. The FREP carries F_{Id}, CI, unreachable flag (UnF), V.No, src, target, and an ordered list of intermediate nodes. V.No indicates the freshness of the route. Each mobile node in the network updates its flow table if it receives a FREP with a higher validity number. Whenever the sub-controller finds a better path, it broadcasts a FREP message carrying a new path with updated V.No. The value of the CI field indicates the method of forwarding data packets, "True" for cellular data transmission or "False" for MD2D data transmission.

When a mobile node receives a FREP message, it first checks the full path to determine whether it is involved in the flow. If not, the mobile node drops the message. Otherwise, the mobile node searches the flow table to find the matched F_{Id}. If the F_{Id} is found, the related fields are updated. Otherwise, an entry for this F_{Id} is added to the flow table.

The flow table in mobile nodes keeps the following fields: F_{Id}, CI, V.No, P_hop (previous hop), N_hop (next hop), and the status of the route. The status field of a flow can be set to one of the following values: "Valid", "Invalid", or "Unreachable". The value of CI field can be set to "True" or "False". The default value for this field is "False". P_hop and N_hop are the hops before and after a node in the path of the data flow. The reason for keeping the previous hop in the flow table is for TCP connections that require target acknowledgments for receiving data packets.

4.3.2 Route Maintenance

Each hop in the path of a data flow is responsible for forwarding packets to the next hop. Each hop confirms the receipt of data packets through layer 2 acknowledgments. A mobile node initiates a flow error message (FERR) to the sub-controller if an error occurs at the time of data transmission. The FERR carries the F_{Id} of data packets and error type, referred to as ERRT (error types: Link_broken, Low_battery, and unknown_Error). After receiving the error, the sub-controller broadcasts a FERR carrying the F_{Id}. Subsequently, the sub-controller checks its FIB to find the matched entry with the reported F_{Id}, updates its LSDB, and runs a least-cost-path algorithm to find a new path for the flow. In this stage, the three possible outcomes are as follows:

- **Destination is not reachable:** In this case, the sub-controller sends a unicast FREP to the source of data flow carrying F_{Id}, UnF value of True, updated V.No, and empty path.
- **Destination is not reachable through MD2D, or the new multi-hop path cannot provide the quality requirements of this data flow compared to cellular data transmission:** In this case, the sub-controller sends a unicast FREP to the source node of data flow carrying F_{Id}, CI value of True, updated V.No, and empty path.
- **The new multi-hop path can provide the quality requirements of this data flow:** In this case the sub-controller broadcasts a FREP consisting of F_{Id}, CI value of False, updated V.No, and an ordered list of relay nodes in the newly selected multi-hop path.

When a mobile node receives a FERR, it first stops sending data flows carrying the reported F_{Id}. The mobile node then searches the flow table to find the matched entry with this F_{Id} and to change the status of that flow to "invalid". If the flow status remains invalid for a pre-determined time, called flow error waiting interval (FERR-WI), and no FREP is received from the sub-controller during this time, the mobile node then drops the related data packets. Entries with invalid status are deleted from the flow table after a flow invalid time period (FI-T).

4.4 HSAW for SDN-based 5G Networks

In HSAW routing protocol, like SMDRP, neighbor discovery is through Hello messages, and the sub-controller has a complete view of its coverage area and its adjacent cells by collecting topology control messages from mobile nodes and other sub-controllers in neighboring cells. The sub-controller broadcasts Inter-LSDB and Intra-LSDB to the network. After receiving the LSDB, each mobile node can build its LSDB and have a global view of other mobile nodes in its cell and its adjacent cells. The sub-controller also broadcasts the traffic policies to the network, including the list of traffic types, specified metrics for each traffic type, and maximum allowed end-to-end cost for each traffic type in an MD2D transmission considering the size of data packets. For example, suppose the traffic type metric is the number of hops. In that case, the sub-controller determines the maximum allowed number of hops between the source and target of this traffic type that could provide the demanded quality requirements for this traffic. If a mobile node has a data packet to send, it first runs the SPF algorithm or other designed routing algorithms (such as Dijkstra's algorithm and the Bellman-Ford algorithm) to find the shortest or least costly route to the target. If the target is not reachable through MD2D or the achieved path cannot provide the quality requirements for this traffic type considering the traffic policies, the source node then adds this flow to its flow table with the CI value of True and forwards data packets to the BS. Otherwise, if the achieved path agrees with the sub-controller traffic policies, the source node adds the obtained path to its flow table with the CI value of False and forwards data packets to the next hop. The flow table in mobile nodes keeps the target address or Id, CI, traffic type, next hop that has a route to the target, and status field.

If any error occurs during data transmission to the next hop, the upstream node of the broken link sends a FERR to the sub-controller carrying the address of the failed node and ERRT. Subsequently, the sub-controller updates the LSDB and broadcasts a FERR carrying information of the broken link (upstream node address, failed node address, and ERRT). Each mobile node updates its LSDB and its flow table after receiving the FERR. The upstream node of the broken link finds a new route to the packet target and forwards the packets through the new route.

4.5 Overhead Analysis

In this section, we extend the proposed algorithm model in [142] to fit our framework to investigate the routing overhead of SMDRP compared to HSAW. The routing overhead of both protocols is analyzed distinctly for cellular and WiFi frequency bands, based on the amount of exchanged information (control and routing messages) in each band. Table 4.1 indicates the network parameters used for network abstraction.

Table 4.1: Summary of notations.

Notation	Description
A	Cell area (an square area with sides of $a \times a$)
N	Total number of mobile nodes in the cell
R	Radio transmission range of each mobile node
ξ	Ratio of mobile node coverage area to network area: $\frac{\pi R^2}{A}$
ND	Neighbor density (average mobile degree): $\xi N - 1$
F	Total number of active flows at time t
F_{Id}	Amount of information differentiates one flow from the other flows: $\text{Log}_2 F$
M_{Id}	Mobile identifier
ζ	Network topology update interval
λ	Packet generation rate (pkt/s)
λ_u	Link break rate
d	Average distance between source and destination of a flow. From [143], $d = 0.5214a$
L	Average number of hops in a flow. From [142], $L \geq \lceil \frac{d}{R} \rceil$
S_D	Data packet size (512 Byte)

4.5.1 Mobility

It is assumed that the WiFi transmission range of each mobile node is $R = 250m$, the topology update interval is $1s$, and mobile nodes can move freely within the square area of side $1000m$. It is also assumed that the probability of a link status change event within a time interval $[t, t + \zeta]$, referred to as q(t), is the same for all links in the network and is calculated as follows:

$$q(t) = 1 - e^{-\lambda_u t} \tag{4.1}$$

where λ_u is link break rate and represents the mobility in the network.

The link status between two mobile nodes at time t is up (down) if the distance between these nodes at time t is less (more) than R.

4.5.2 Hello Message Overhead

Each mobile node is identified by M_{Id} in the network. If the total number of mobile nodes is N, then the amount of information required to identify a mobile node in the network is Log_2N [142]. In each update interval (ζ), a mobile node broadcasts a Hello message carrying M_{Id} from its WiFi interfaces. If we assume that each mobile node has only one WiFi interface, then the overhead caused by Hello messages, referred to as O_H, is given by:

$$O_H = N\ M_{Id} = N\ Log_2N \tag{4.2}$$

4.5.3 Topology-update Overhead

Each mobile node sends a TC message to the sub-controller from its LTE interface in each update interval (ζ). It is assumed that mobile nodes send only link-state information to the sub-controller and the average number of neighbors for each mobile node is *ND*. At the beginning (Time 0)[1], when mobile nodes forward their first TC message to the sub-controller (i.e., list of one-hop neighbors and associated link status[2] (LS)), the total overhead resulting from the TC messages, referred to as O_{TC0}, is as follows:

$$O_{TC0} = N\ ND\ (M_{Id} + LS) \tag{4.3}$$

In HSAW, the sub-controller builds an Inter-LSDB based on the received TC messages and broadcasts Inter-LSDB to the network at time 0. The overhead of broadcasting the Inter-LSDB, referred to as O_{LSDB}, is as below:

$$O_{LSDB} = N\ ND\ (2\ M_{Id} + LS) \tag{4.4}$$

The total topology-update overhead at time 0, defined as $O_{Topology0}$, for the HSAW and SMDRP protocols are as follows:

$$O_{Topology0} = O_{TC0} + O_{LSDB} \qquad (HSAW) \tag{4.5}$$

$$O_{Topology0} = O_{TC0} \qquad\qquad (SMDRP) \tag{4.6}$$

When nodes start moving in the network, link failure is likely to occur more frequently. The amount of information sent by mobile nodes to the sub-controller at time t to update the sub-controller on the topology changes, referred to as O_{TCM}, is calculated as follows (it is assumed that the average number of neighbors for a mobile node in each time step ζ is *ND* and all the

[1]The term of Time 0 is used for indicating a static network in which mobile nodes are not moving.

[2]Link status types are Heard, Symmetric, and Lost. The number of bits describing the link status is 1 octet [144].

connected links have changed in each time step):

$$O_{TCM} = N\ q(t)\ ND\ (M_{Id} + LS) \tag{4.7}$$

In HSAW, the amount of information that is broadcast by the sub-controller to update the mobile nodes about the topology changes, referred to as O_{TCC}, is given by:

$$O_{TCC} = N\ q(t)\ ND\ (2\ M_{Id} + LS) \tag{4.8}$$

The total overhead resulting from topology updates at time t, defined as $O_{Topology}$, is as follows:

$$O_{Topology} = O_{TCM} + O_{TCC} \qquad (HSAW) \tag{4.9}$$

$$O_{Topology} = O_{TCM} \qquad (SMDRP) \tag{4.10}$$

4.5.4 Route Discovery Overhead

In both protocols, the flow table of the mobile nodes at the beginning of the network is empty. If a mobile node has a data packet to send, in HSAW, the mobile node runs SPF to find the shortest path, adds the obtained path to its flow table, and forwards data packets to the next hop (see Figure 4.2). Unlike HSAW, in SMDRP, the source of the data flow sends a FREQ to the sub-controller carrying Req_Num (8 bits) and target information (see Figure 4.4). If the number of active flows in the network at time 0 is F, the overhead caused by the source of data packets for a valid route, referred to as O_{FREQ0}, is as below:

$$O_{FREQ0} = F(M_{Id} + Req\_Num)$$

In response to FREQs, the sub-controller assigns a unique F_{Id} to each flow, finds a route for each request, and broadcasts a FREP per request (see Figure 4.4). A FREP[3] carries F_{Id}, CI, UnF, $V.No$, Src, Des, and a list of relay nodes[4]. If the number of active flows in the network at time 0 is F, the overhead resulting from broadcasting FREPs by the sub-controller, referred to as O_{FREP0}, is as follows (number of bits indicating CI, UnF, $V.No$ fields are 1, 1, and 8 bits, respectively):

$$O_{FREP0} = F(F_{Id} + 10 + (L+2)\ M_{Id}) \tag{4.11}$$

The total overhead caused by route discovery (in SMDRP) at time 0, defined as O_{RD0}, is given by:

$$O_{RD0} = \zeta(O_{FREQ0} + O_{FREP0}) \tag{4.12}$$

[3] It is assumed that the selected method by the sub-controller to reach to the requested targets is through MD2D connections.

[4] If average number of hops in each flow is L, the amount of information describing the relay nodes of a flow is $L\ M_{Id}$.

4.5.5 Route Maintenance Overhead

If a link failure occurs in one of the links of a flow at time t, then the upstream node of the broken link sends a FERR to the sub-controller. The information carried by a FERR is different for each protocol. In HSAW, a FERR carries failed node information and ERRT (see Figure 4.3), while in SMDRP, the F_{Id} of the flow along with ERRT is sent to the sub-controller (see Figure 4.5). The amount of overhead caused by sending FERRs to the sub-controller at time t, referred to as O_{FERRM}, is as follows (the number of bits describing the error is 2 bits. It is assumed that the probability of link change ($q(t)$) is the same for L links):

$$O_{FERRM} = F\ q(t)\ (M_{Id} + ERRT) \qquad \textbf{(HSAW)} \tag{4.13}$$

$$O_{FERRM} = F\ q(t)\ (F_{Id} + ERRT) \qquad \textbf{(SMDRP)} \tag{4.14}$$

Following the receipt of FERRs, the sub-controller broadcasts a FERR per received error to the network to inform mobile nodes about the failed links. Information on the FERR is different for each protocol. In HSAW, a FERR carries the upstream node of the broken link, failed node, and ERRT. While in the SMDRP, a FERR carries only the F_{Id} of the broken flow (see Figure 4.5). The overhead resulting from broadcasting FERRs by the sub-controller at time t, referred to as O_{FERRC}, is given by:

$$O_{FERRC} = F\ q(t)\ (2M_{Id} + ERRT) \tag{4.15}$$

$$O_{FERRC} = F\ q(t)\ F_{Id} \tag{4.16}$$

In SMDRP, in addition to FERR, the sub-controller also broadcasts a FREP to update the involved nodes on a new route (see Figure 4.6). The overhead caused by broadcasting FREPs in response to the broken flows at time t, referred to as O_{FREP}, is as follows:

$$O_{FREP} = F\ q(t)\ (F_{Id} + 10 + (L+2)\ M_{Id}) \tag{4.17}$$

The total overhead resulting from route maintenance at time t, defined as O_{RM}, is given by:

$$O_{RM} = \zeta(O_{FERRM} + O_{FERRC}) \qquad \textbf{(HSAW)} \tag{4.18}$$

$$O_{RM} = \zeta(O_{FERRM} + O_{FERRC} + O_{FREP}) \quad \textbf{(SMDRP)} \tag{4.19}$$

4.5.6 Data Packet Delivery Overhead

Data packet transmission through MD2D connections consumes WiFi-bandwidth. In SMDRP, a F_{Id} is attached to each data packet before being sent to the next hop. If the packet generation rate of each mobile node is λ and the size of data packets is S_D, the overhead of data packet delivery through the WiFi-band at time 0, defined as O_{PD}, is given by:

$$O_{PD} = \zeta\, F\;\; L\, \lambda\, S_D \qquad \textbf{(HSAW)} \tag{4.20}$$

$$O_{PD} = \zeta\, F\;\; L\, \lambda\, (S_D + F_{Id}) \qquad \textbf{(SMDRP)} \tag{4.21}$$

When a link failure occurs in the links of a data flow, there is an overhead in the WiFi-band resulting from the bandwidth consumed by data packets that have not been delivered successfully. Depending on which link (K^{th} link of the flow) has failed first in a flow with L number of links, the overhead of data packets that have delivered unsuccessfully, referred to as O_{FD}, when the packet generation rate of mobile nodes is λ, is as follows:

$$O_{FD} = \zeta\, F\, q(t)\;\; \lambda\, S_D \sum_{k=1}^{L} k\, (1 - q(t))^{k-1} \qquad \textbf{(HSAW)} \tag{4.22}$$

$$O_{FD} = \zeta\, F\, q(t)\;\; \lambda\, (S_D + F_{Id}) \sum_{k=1}^{L} k\, (1 - q(t))^{k-1} \qquad \textbf{(SMDRP)} \tag{4.23}$$

4.5.7 Total Cellular Overhead

The total overhead of the cellular-band at time 0, defined as $O_{Cellular0}$, is given by:

$$O_{Cellular0} = \frac{1}{\zeta}(O_{Topology0}) \qquad \textbf{(HSAW)} \tag{4.24}$$

$$O_{Cellular0} = \frac{1}{\zeta}(O_{Topology0} + O_{RD0}) \quad \textbf{(SMDRP)} \tag{4.25}$$

The total overhead of the cellular-band at time t, defined as $O_{Cellular}$, is given by:

$$O_{Cellular} = \frac{1}{\zeta}(O_{Topology} + O_{RM}) \tag{4.26}$$

4.5.8 Total WiFi Overhead

The total overhead of the WiFi-band at time 0, referred to as O_{WiFi0}, is as follows:

$$O_{WiFi0} = \frac{1}{\zeta}(O_H + O_{PD}) \tag{4.27}$$

The total overhead of the WiFi-band at time t, referred to as O_{WiFi}, is as follows:

$$O_{WiFi} = \frac{1}{\zeta}(O_H + O_{FD}) \tag{4.28}$$

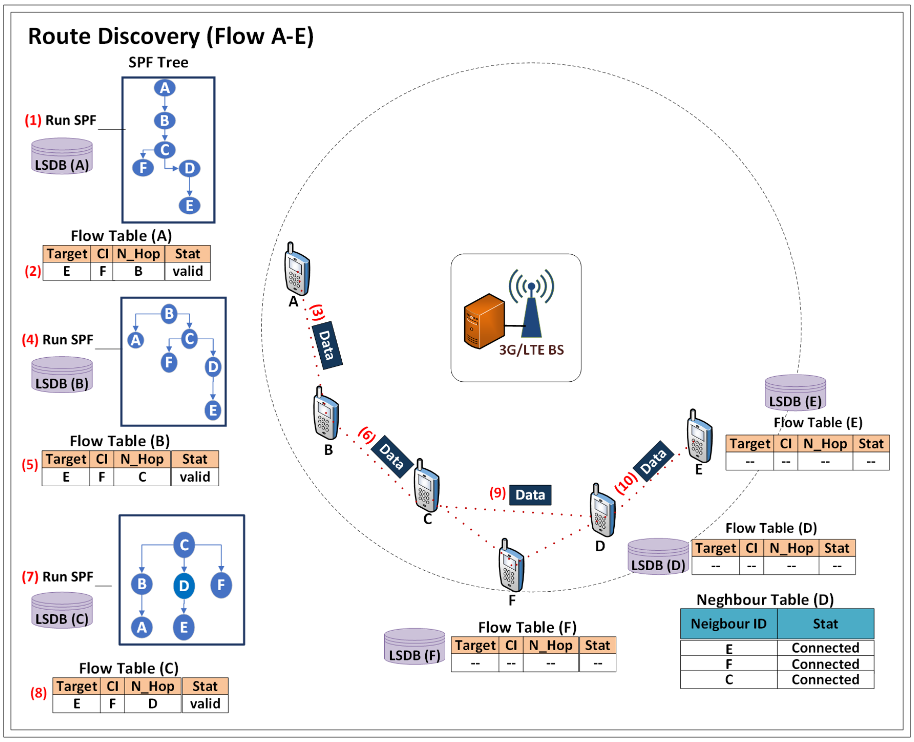


Figure 4.2: Route discovery in HSAW (flow A-E).

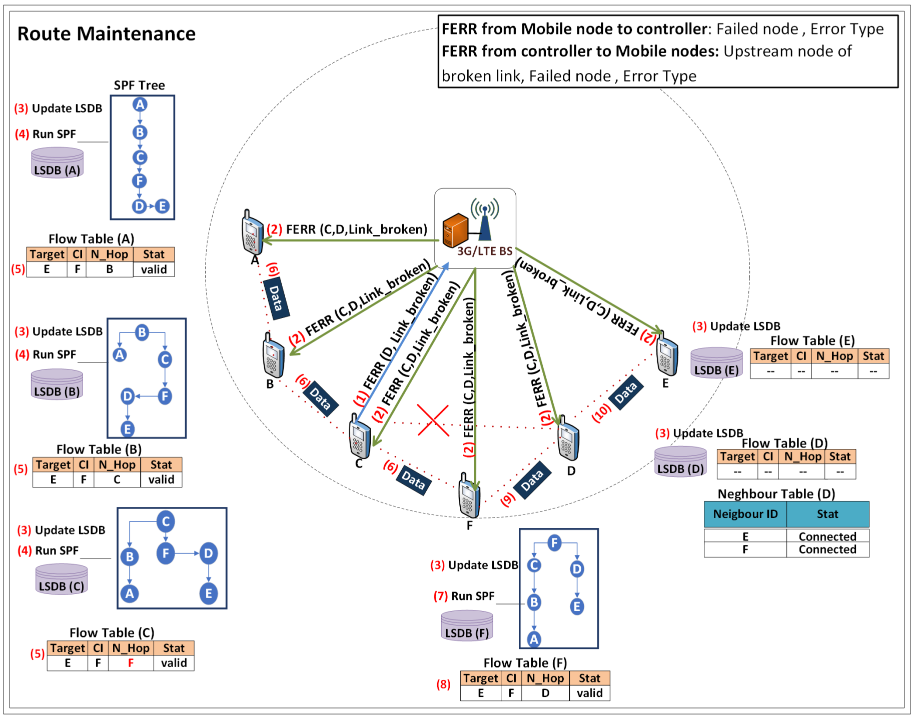


Figure 4.3: Route maintenance in HSAW.

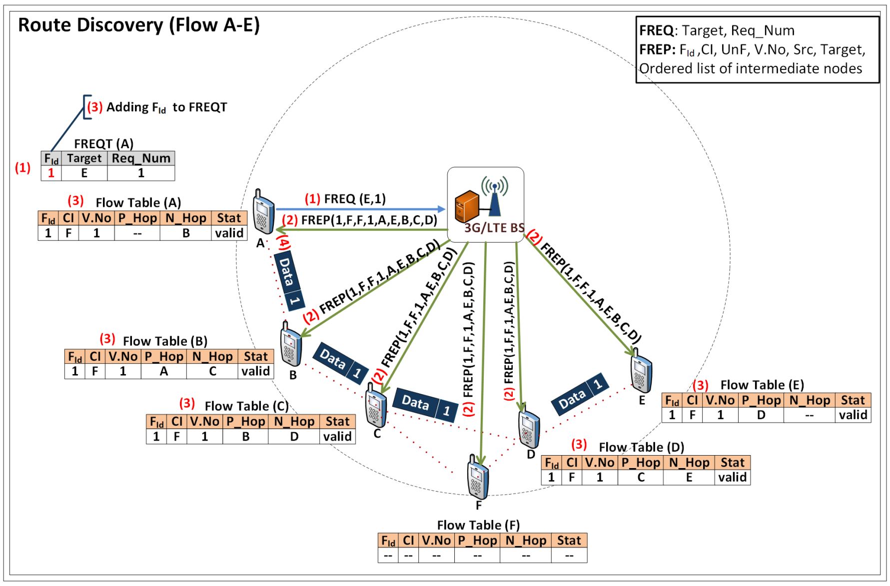


Figure 4.4: Route discovery in SMDRP (flow A-E).

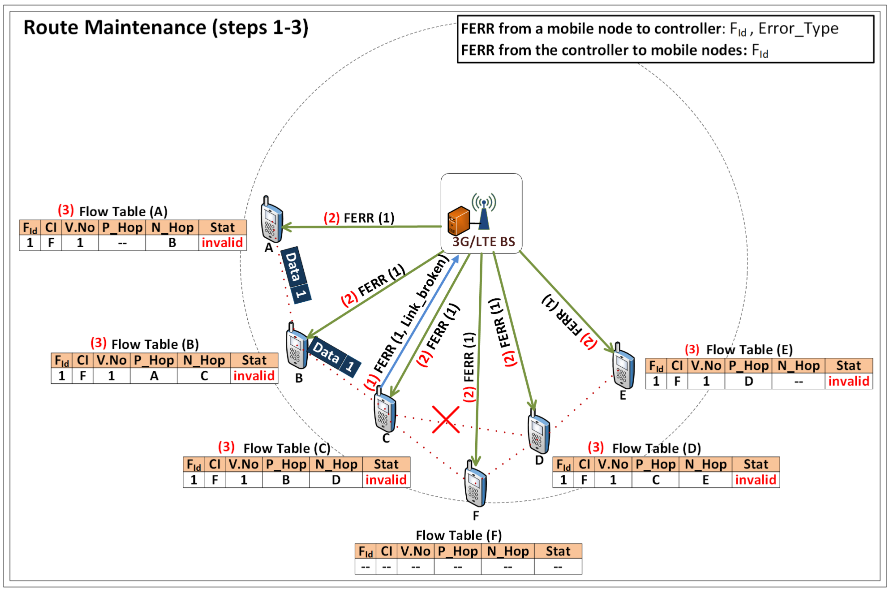


Figure 4.5: Steps 1-3 of the route maintenance in SMDRP (link C-D is broken).

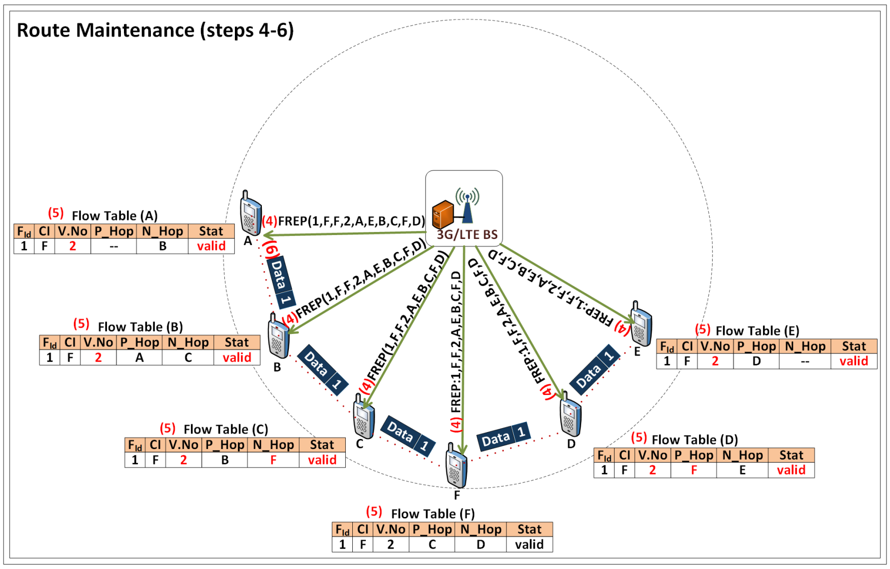


Figure 4.6: Steps 4-6 of the route maintenance in SMDRP.

4.5.9 Result Analysis

This section compares the overhead of both protocols for static and mobile networks distinctly, where $A = 10^6(m^2)$, $R = 250(m)$, the number of nodes is 500, and the topology-update intervals are $1s$.

4.5.9.1 Overhead Analysis in Static Networks (Time0)

Figures 4.7 and 4.8 respectively present the overhead of both protocols in WiFi-band and cellular-band versus the number of mobile nodes. As shown in Figure 4.7, SMDRP produces more overhead on the WiFi-band compared to HSAW because of the attached F_{Id} to the data packets. However, this additional overhead is negligible. In the cellular-band, HSAW experiences more overhead than SMDRP, especially for a higher number of nodes (see Figure 4.8) because of the consumed cellular bandwidth in HSAW for broadcasting link-state information from the sub-controller to the network.

4.5.9.2 Overhead Analysis in Mobile Networks

Figures 4.9 and 4.10 present respectively the overhead of both protocols in WiFi-band and cellular-band versus mobility, where the number of active flows is 2000. In the WiFi-band, SMDRP exhibits a higher overhead than HSAW owing to the F_{Id} attached to the data packets. However, this extra overhead is negligible, as shown in Figure 4.9.

In the cellular-band, the overhead of both protocols increases in accordance with mobility. The overhead resulting from topology updates in HSAW is significantly higher than SDMRP because of the extra bandwidth consumed by the sub-controller in HSAW for broadcasting the topology changes to the network. The SMDRP route maintenance overhead is greater than HSAW, especially for higher mobility, owing to the FREPs sent by the sub-controller to update mobile nodes on the new routes (see Figure 4.12). In HSAW, the overhead caused by FERR messages is higher than the SMDRP, as depicted in Figure 4.13. In SMDRP, the bandwidth consumed by FREP messages is more than the bandwidth consumed by FERR messages, as can be seen in Figure 4.14. However, the total overhead resulting from topology updates and route maintenance in HSAW is greater than SMDRP, as illustrated in Figure 4.10.

Figure 4.15 compares the overhead of both protocols versus the number of mobile nodes, where λ_U is 5. As shown in Figure 4.15, by increasing the number of nodes, the total overhead of both protocols increases accordingly. However, SMDR introduces better performance compared to HSAW, especially for a higher number of nodes. It is noteworthy that the memory and energy consumption of mobile nodes in the SMDRP are less than those of the HSAW. This is because in SMDRP, each mobile node keeps only on-demand routes in its flow table, and the sub-controller performs all the routing calculations and required processing. Unlike SMDRP, mobile nodes in HSAW need to update their flow tables periodically regardless of whether they are part of an active flow. This will increase the energy consumption and CPU processing overhead due to the frequent SPF and flow table recalculations. Moreover, HSAW has more memory overhead (see Figure 4.16) as each mobile node maintains the LSDB of the entire network. Section 4.6 presents a numerical analysis of a mobile node memory usage.

4.6 Memory Analysis

In this section, the memory usage of a mobile node for both protocols is analyzed in Table 4.2. It is assumed that the average number of neighbors for each mobile node is *ND*, and each mobile node is involved in the F number of active flows.

Figure 4.16 presents a mobile node memory usage versus the number of nodes, where the number of active flows is five times the number of nodes. It can be seen from the figure that by increasing the number of nodes, the total memory usage of both protocols increases accordingly, but this growth is significant in HSAW. Figure 4.16 also illustrates the memory usage of a mobile node in HSAW is greater than SMDRP. Since in the HSAW, the memory usage for maintaining the LSDB of the entire network increases when the number of nodes in the network grows.

Figure 4.17 depicts the memory usage of a mobile node versus the number of flows in which the mobile node is involved. As can be seen, when the number of flows is zero, a mobile node in HSAW consumes more memory as it keeps the LSDB and neighbor table, whereas in SMDRP, it retains only the neighbor table. By increasing the flows, the memory usage of the mobile node will rise accordingly, as they have to keep flow information. Figure 4.17 reveals that SMDRP performs better in terms of memory usage than HSAW.

Table 4.2: Memory analysis of mobile nodes.

Amount of memory that each mobile node requires for routing purposes in HSAW	
Memory for keeping the LSDB	$M_{LSDB} = (N-1)\,ND\,(2\,M_{Id} + LS)$
Memory for keeping the neighbour table (NT)	$M_{NT} = ND\,(M_{Id} + LS)$
Memory for keeping the flow table (FT) FT keeps target, CI, next hop, and status of route (2 bits)	$M_{FT} = F\,(2\,M_{Id} + 3)$
Total memory	$M_{Total} = M_{LSDB} + M_{NT} + M_{FT}$
Amount of memory that each mobile node requires for routing purposes in SMDRP	
Memory for keeping the FREQT FREQT keeps flow Id, requested target, and request number	$M_{FREQT} = F\,(F_{Id} + M_{Id} + Req\_Num)$
Memory for keeping the neighbour table (NT)	$M_{NT} = ND\,(M_{Id} + LS)$
Memory for keeping the flow table (FT) FT keeps flow Id, CI, validity number, previous hop, next hop, and status of route (2 bits)	$M_{FT} = F\,(F_{Id} + 11 + 2\,M_{Id})$
Total memory	$M_{Total} = M_{FREQT} + M_{NT} + M_{FT}$

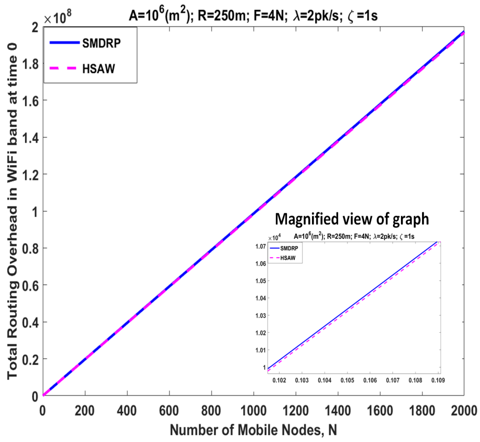


Figure 4.7: Total routing overhead in WiFi-band at time 0 vs. Number of nodes.

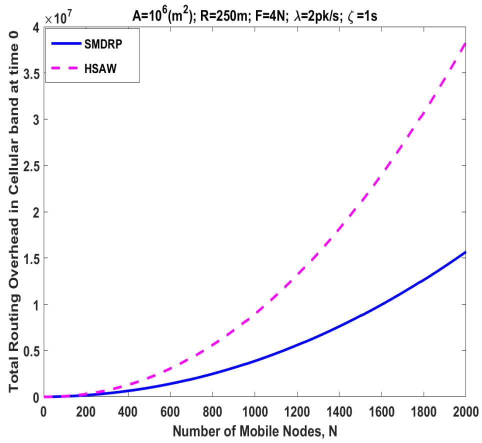


Figure 4.8: Total routing overhead in Cellular-band at time 0 vs. Number of nodes.

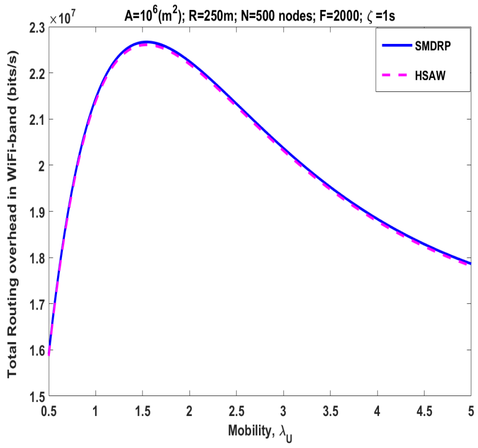


Figure 4.9: Total routing overhead in WiFi-band per time step ζ vs. Mobility.

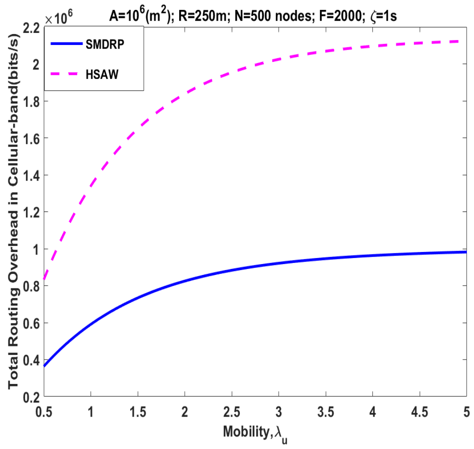


Figure 4.10: Total routing overhead in the cellular-band per time step ζ vs. Mobility.

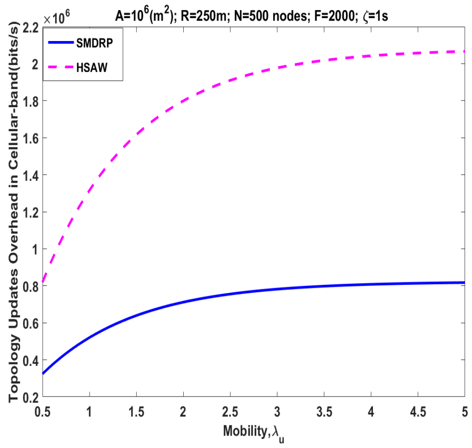


Figure 4.11: Topology-updates overhead in the cellular-band per time step ζ vs. Mobility.

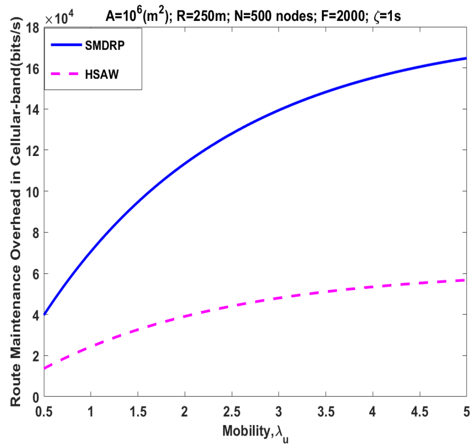


Figure 4.12: Route Maintenance overhead in the cellular-band per time step ζ vs. Mobility.

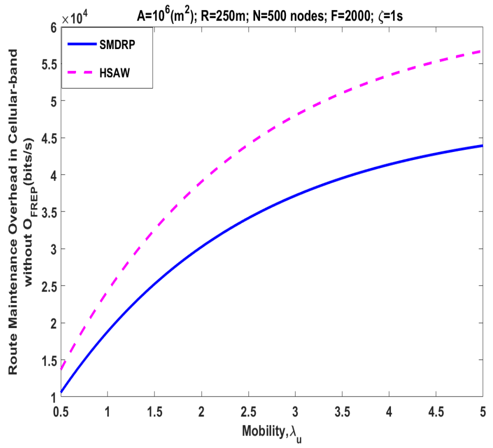


Figure 4.13: Comparing O_{FERR} of both protocols.

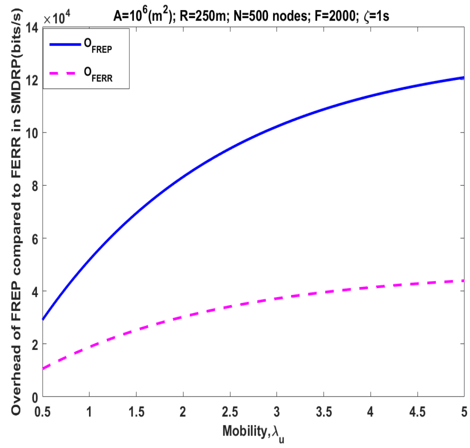


Figure 4.14: Comparing O_{FREP} with O_{FERR} in SMDRP.

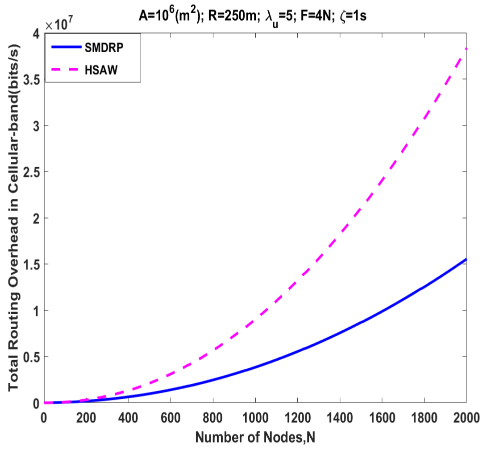


Figure 4.15: Total routing overhead in cellular-band per time step ζ vs. Number of nodes.

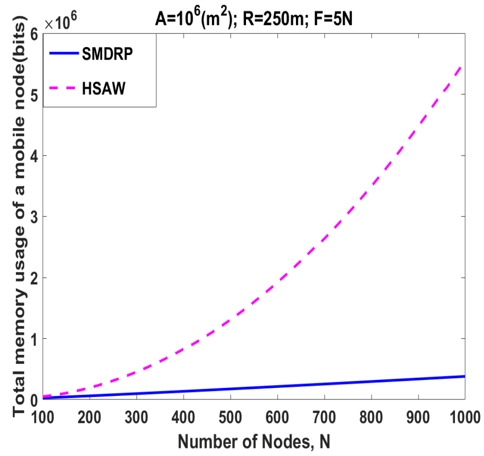


Figure 4.16: Memory usage of a mobile node vs. Number of nodes.

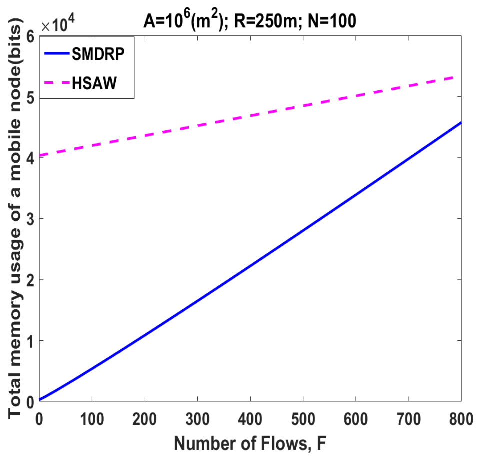


Figure 4.17: Memory usage of a mobile node vs. Number of flows in which the mobile node is involved.

4.7 Conclusion

This chapter presented a new SDN-based multi-hop device-to-device routing protocol, referred to as SMDRP, which takes advantage of the concept of software-defined networking to efficiently manage MD2D routing within and between each SDN-enabled cell under the supervision of an SDN-controller. The proposed protocol can be applied to different SDN-based wireless networks. Our proposed protocol can be considered as a semi-distributed routing protocol, where an SDN controller manages and controls part of the overall MD2D routing functionality to increase scalability while enabling network operators to control and maintain the out-of-band packet forwarding network. This chapter also extended prior work on the hybrid SDN architecture for wireless distributed networks (HSAW) [3] and adapted that to the framework presented in this chapter. In HSAW, because the controller floods all link-state information to the nodes, the network experiences a scalability problem. This problem is overcome by passing only the next hop for each active route to the mobile nodes in our approach. To investigate this, we performed theoretical and simulation studies comparing HSAW with SMDRP. From our results, it can be seen that for larger density populated networks, SMDRP shows better scalability than HSAW. In addition, mobile nodes require less memory and energy for communication.

5

Dynamic Routing Protocol Selection in Multi-hop Device-to-Device Wireless Networks

5.1 Introduction

Today, wireless cellular networks have been facing several challenges: 1) increasing demand from mobile users for a better quality of service (QoS) and quality of experience (QoE), 2) the growing interest in machine-to-machine (M2M) applications and the internet of things (IoT) to connect a large number of devices, 3) costly and timely network upgrade due to proprietary hardware and protocols, and 4) compatibility issues between various existing heterogeneous networks and difficulties in their management. Several emerging technologies are integrated into 5G and beyond networks to address the aforementioned challenges and enhance the capacity, flexibility, and scalability of the network, including software-defined networking (SDN), network function virtualization (NFV), multi-hop device-to-device (MD2D) communications, massive multiple-input multiple-output (MIMO), mobile edge computing (MEC), millimeter wave (mmWave) communications, and deploying ultra-dense small cells [145–150]. The main focus of this study is on SDN-based multi-hop routing in cellular networks.

SDN provides logically centralized control over a network via separating the control plane from the data plane. It employs open APIs between different layers to facilitate network programmability. In the SDN architecture, vendor-specific configurations, control decisions, complexity, and computational overheads are taken from the data plane. Data plane devices are programmable forwarding elements receiving instructions from the control plane to relay data traffic. All network functionalities, such as performance monitoring, traffic management, network discovery, routing, load balancing, and security appliances, are applications running on top of the controller [13, 151, 152]. Hence, SDN can provide service providers with simplified network management, smooth service and application deployments, dynamic resource configuration and management, fast integration of different heterogeneous radio technologies, and enhanced QoS and QoE through fine-grained control over resources. Mobile devices in an SDN-based cellular network can be configured through open APIs. SDN enables intelligent network traffic management by directing the multi-interface mobile users to employ the most suitable wireless technology for communications [153–155].

Significant research has been conducted on SDN-based cellular networks [156–159]. However, to the best of our knowledge, no publications can be found on the dynamic selection of routing protocols under varying network conditions. The motivation of this study is to design inclusive multi-protocol architecture which can fully utilize the complementing strengths of individual routing protocols in different network conditions. This is important for supporting the dynamic selection of routing protocols, adapting to network conditions.

In this chapter, we design a multi-protocol framework to maximize the overall network performance. In the proposed framework, an SDN controller provides oversight of the entire cell and initially clusters the cell based on the various traffic densities (i.e., sparse, semi-dense, and dense) observed in different parts of a cell. The SDN controller identifies which routing protocol can provide better end-to-end connectivity within each cluster. To this end, the controller analyzes each cluster based on parameters that impact the protocols' functionalities, such as the average mobility rate of the devices, number of flows, and traffic density. Subsequently, the controller selects a protocol with better end-to-end connectivity for each cluster.

To investigate which protocol suits which network condition, we first analyze the performance of the protocols in each cluster in terms of different network performance criteria, such as E2E delay, energy usage, cellular overhead, and packet loss. Furthermore, we evaluate other factors, such as the high-demand services and energy-dependency of the devices, when selecting a routing protocol. We employ the AHP method to choose the best protocol for each cluster based on the level of importance of each performance criterion. For example, suppose the high-demand services in a cluster are delay-sensitive services. In this case, a routing protocol that better supports these services and introduces the least E2E delay is selected.

The contributions of this chapter are summarized as follows:

- A novel framework is developed to integrate MD2D and cellular systems under a unified framework. The framework introduces a joint clustering and dynamic selection strategy for MD2D routing in cellular networks and supports diverse traffic conditions and requirements.
- The proposed framework streamlines the decision-making process and facilitates using computationally efficient decision-making approaches. An AHP-based method is developed to assist in the protocol selection. A set of important network performance criteria are identified with the importance level of each criterion varying based on the cluster requirements, hence making the AHP model fit for purpose.

- An extensive simulation study is conducted to evaluate the proposed framework in comparison with the state-of-the-art routing frameworks. The proposed framework has the potential to integrate and orchestrate current and emerging MD2D routing protocols.

The rest of this chapter is organized as follows. Section 5.2 gives a thorough review of previous studies on the integration of multi-hop communications and the SDN paradigm with cellular networks. Section 5.3 provides a detailed description of the proposed multi-protocol framework. Section 5.5 presents the channel, energy and network model used to implement the multi-protocol framework. Section 5.6 demonstrates the simulation results and emphasizes the potential benefits of the framework. Finally, Section 5.7 presents the concluding remarks of this chapter and the future work.

5.2 Related Work

In cellular networks, users experience lower data rates and QoS levels when their distance from the base station (BS) increases. This is mainly due to signal attenuation and the existence of different obstacles and interferers. Increasing the number of BSs could help extend the cellular coverage, but it imposes additional operational and deployment costs. Integrating MD2D or ad hoc networks with the cellular network can be an advantageous solution with no need for extra hardware expenditure. MD2D communications in cellular networks enable wireless devices in close proximity to communicate directly instead of traversing through the cellular network. This will enhance the cellular network scalability, system capacity, resource allocation, energy efficiency, spectrum utilization, transmission delay, and interference management. The reason is that devices in MD2D communications consume less energy and experience high-speed data transfer over short-distance hop-by-hop connections than long-distance one-hop connections with the BS. Further, the traffic load on the cellular network is reduced, and consequently, the cellular capacity is increased by offloading the cellular traffic to MD2D communications. The devices in the areas with poor cellular coverage can use MD2D connections to reach the BS. In addition, the usage of the unlicensed frequencies for multi-hop connections enables a mobile user to have simultaneous cellular and MD2D connections. The applications of MD2D in the cellular networks are content distribution, local advertisements, proximity-based social networking applications, machine-to-machine (M2M) applications, and emergency scenarios in which cellular infrastructure is damaged or not available [160–164]. 3G Partnership Project (3GPP) Release 12 also employed D2D communications in LTE networks for applications, such as public safety, proximity-based services, and network offloading by integrating IEEE 802.11 technologies into cellular networks. 3GPP Release 13-15 approved two-hop communication to further increase the network capacity. However, the development of routes over more than two hops has not yet been supported by 3GPP [69,121,165].

Significant research has been conducted on developing techniques to enhance interface management, power efficiency, cellular coverage expansion, resource allocation management, spectrum utilization, and QoS for MD2D communications in cellular networks [67,166–169]. The proposed strategies were based on fixed or mobile relay nodes for uploading and downloading to and from a BS to increase the network throughput and capacity. Regardless of the proposed solutions and techniques, a variety of constraints still exist. First, there is inflexibility and a high cost of equipment due to proprietary hardware and protocols. Second, the vendor-specific configuration of interfaces and devices makes the network upgrade costly and complex. Third, the tight coupling of control and data planes in the equipment complicates the network and service deployments. Fourth, there is a lack of fine-grained control over resources and centralized data plane functions in the core entity of long-term evolution (LTE) networks resulting in scalability issues. Finally, there is poor support for virtualization [94, 170, 171]. 5G wireless and beyond has addressed these issues, by integrating SDN and NFV at the core and the edge. Integrating SDN and NFV into cellular networks provides an open and reprogrammable framework to facilitate rapid and cost-effective service deployment.

Several studies have been conducted to integrate SDN into the existing cellular architecture. The proposed solutions focus on replacing the main forwarding entities of the LTE architecture with inexpensive and simple programmable components managed and controlled by the SDN controller. This introduces higher network throughput, enhanced handover management, less energy consumption, lower latency, and a significant reduction in the deployment and operational cost of infrastructure used by the cellular network [155–158, 172–175]. Further, a few studies focused on SDN-based D2D[1] and MD2D routing in the cellular networks [176]. In [177], a three-tiered SDN architecture was proposed, namely management, controller, and physical. In this architecture, BSs perform basic packet-processing functions and receive instructions from applications running on a centralized controller. Two different frequencies were used for offloading the traffic: one for the macro BS and the other for the femtocells. Moreover, data-plane rules defined by operators were distributed over multiple low-cost network switches to make the network more scalable and minimize the overhead of the core entity. In [101], a QoE enhancing algorithm was presented based on one-hop D2D communication in software-defined multi-tier LTE-advanced (LTE-A) networks to facilitate Internet access and enhance the QoE of users for uploading and downloading transmissions. In the proposed algorithm, the defined application modules on the SDN controller collaborate to establish a one-hop D2D link. The load balancer and resource allocator modules determine which eNodeB can provide the best QoE for each UE. If no idle eNodeB is found, mobility management, load balancer, resource allocator, and routing setup modules collaborate to establish a one-hop D2D link using a relay node between mobile devices and the BS for uploading and downloading transmissions. The authors in [98] proposed a hierarchical D2D communication architecture with a centralized SDN controller that communicates with the cloud head (local controller) to minimize the number of LTE requests. The authors believe that the proposed cloud-based architecture improves energy consumption and scalability and helps with public safety applications through multi-hop routing under the management of the SDN controller. In [178], an SDN-based routing scheme was designed, called low-overhead D2D routing (LODR), for multi-hop D2D communications of mobile nodes in wireless networks using SDN techniques. In the proposed architecture, each user equipment (UE) is equipped with OpenFlow switch capabilities.

[1] In D2D routing, there is only one relay node between BS and mobile device. The relay node could be a mobile user or an AP.

An OpenFlow controller controls the multi-hop routing behavior of UEs. Their simulation results showed a better performance for the proposed scheme compared to traditional ad-hoc networks in terms of control overhead and network convergence. In [140], a software-defined communication layer within mobile devices was designed, referred to as WASP (WiFi, ad-hoc, software-defined networking, and personal-mobile), for hybrid wireless networks. Mobile nodes in the proposed architecture forward their neighboring information to an SDN controller. The acquired information is used by the SDN controller to instruct mobile devices to relay traffic. Their experimental results demonstrated better performance for WASP compared to traditional ad hoc networks in terms of scalability with a minimum trade-off of energy. They also presented a content distribution scheme using WASP to minimize the load on cellular networks. The authors in [179] proposed a multi-layer SDN-based architecture to efficiently interconnect multi-interface mobile users (LTE/WiFi). In the proposed architecture, a global controller manages the entire LTE/WiFi network with the assistance of multiple local controllers. Each local controller monitors and manages a specific area. The global controller manages the radio resource allocation by collecting information from the local controllers and makes the offloading decision when traffic load on a local controller is high. Authors in [180] designed a software-defined cooperative offloading model (SDCOM) in which an SDN controller is deployed at the packet data network (PDN) gateway of the LTE-A architecture and performs a centralized task scheduling to reduce access links traffic and energy usage of the mobile devices. In the proposed model, mobile users execute tasks cooperatively and share the results. In [181], a hierarchical D2D communication architecture was proposed that improves power consumption by reducing the number of LTE communication channels. In the proposed architecture, mobile clouds are created using D2D to facilitate various services. A global SDN controller registers the formed mobile clouds in the network and has a global view of the services offered by the served clouds. Hence, the controller can set up the clouds against users' requests. The controller interacts with the cloud heads and determines the routing paths between the cloud heads by considering the link quality between the cloud heads and their residual batteries.

In our previous studies [3, 182, 183], an SDN-based framework was proposed to offload traffic from the cellular network to multi-hop D2D routing. In the proposed architecture, the SDN controller has a global view of the entire network using the received neighbor information from the mobile nodes. The controller determines how mobile users direct their data traffic. The proposed framework supports various multi-hop routing protocols. Taking advantage of this feature, two types of routing strategies were designed in the proposed framework: proactive and reactive. For the proactive approach, the HSAW protocol was proposed wherein the SDN controller broadcasts traffic policies and the ink-state database (LSDB) of the entire network [3]. Thus, mobile nodes can make routing tables to forward traffic. In the reactive approach, the controller provides forwarding information to the mobile nodes for any active route in the network. For this strategy, two different routing protocols were proposed: VARP-S [182] and SDN-based multi-hop D2D routing protocol (SMDRP) [183]. In the VARP-S, the controller provides the source node with forwarding information. Subsequently, the source node attaches the full path to the data packets and forwards the packets to the next hop. In the SMDRP, the controller assigns a unique flow ID to each active flow and broadcasts the forwarding information. Only the nodes involved in the flow store the flow information in their flow tables. Unlike VARP-S, in SMDRP, the source node only attaches the flow ID to each data packet. Intermediate nodes forward data packets to the next hop based on the flow ID attached to the packets. Our studies indicated that the proposed framework significantly reduces the network overhead compared to traditional ad hoc networks. Further, for densely populated networks, the reactive approach performs better than the proactive approach in terms of cellular overhead and energy consumption.

In contrast to existing work which primarily focus on single-protocol structure, we propose a novel unified multi-protocol architecture enabling the integration of MD2D and cellular networks. Furthermore, the framework empowers the BS controller to seamlessly switch or select the best protocol based on network conditions and requirements. The use of multiple protocols in one framework can maximize the overall performance when different network conditions exist.

5.3 Proposed multi-protocol SDN-based framework for cellular networks

Figure 5.1 presents the proposed multi-protocol framework. Initially, an SDN-based BS[2] logically splits the cell into multiple clusters based on the traffic density. The BS has global knowledge of the network and can dynamically determine which routing protocol is suitable for each cluster. Depending on the cluster condition, the SDN controller dynamically updates the routing protocol used by mobile nodes and the routing parameters. For example, the HSAW routing protocol may be used in one cluster, while another may use the VARP-S protocol. Routing parameters, such as Hello packet intervals, may differ between clusters to support various mobility rates. If the condition of a cluster changes, then the controller instructs nodes to switch to a new selected protocol. However, this switching should occur in a non-real-time manner to prevent unnecessary route flapping. The controller uses the collected historical data to decide whether switching is required for a cluster and when switching should occur.

The motivation behind partitioning a cell into different clusters and using a specific routing protocol for each cluster is because each routing protocol differs in terms of complexity and routing overhead, and consequently, is suitable for networks with specific features. The use of a single routing protocol may not be suitable for networks experiencing various network conditions in different areas. Hence, the controller investigates each area in terms of factors affecting the functionality and performance of the multi-hop routing protocols, such as average mobility rate, node density, and the number of flows. The SDN controller then selects the suitable routing protocol for each cluster by having sufficient knowledge of each protocol and its routing parameters and capabilities. The following example is provided to better clarify the purpose of the clustering in our framework: in urban environments, assuming a stadium full of users and comparing it with the surrounding areas. While the stadium network is fully congested, the surrounding areas might experience a lower traffic density. If we assume two separated clusters for the stadium and sur-

[2] In this study, the terms BS and controller are used interchangeably.

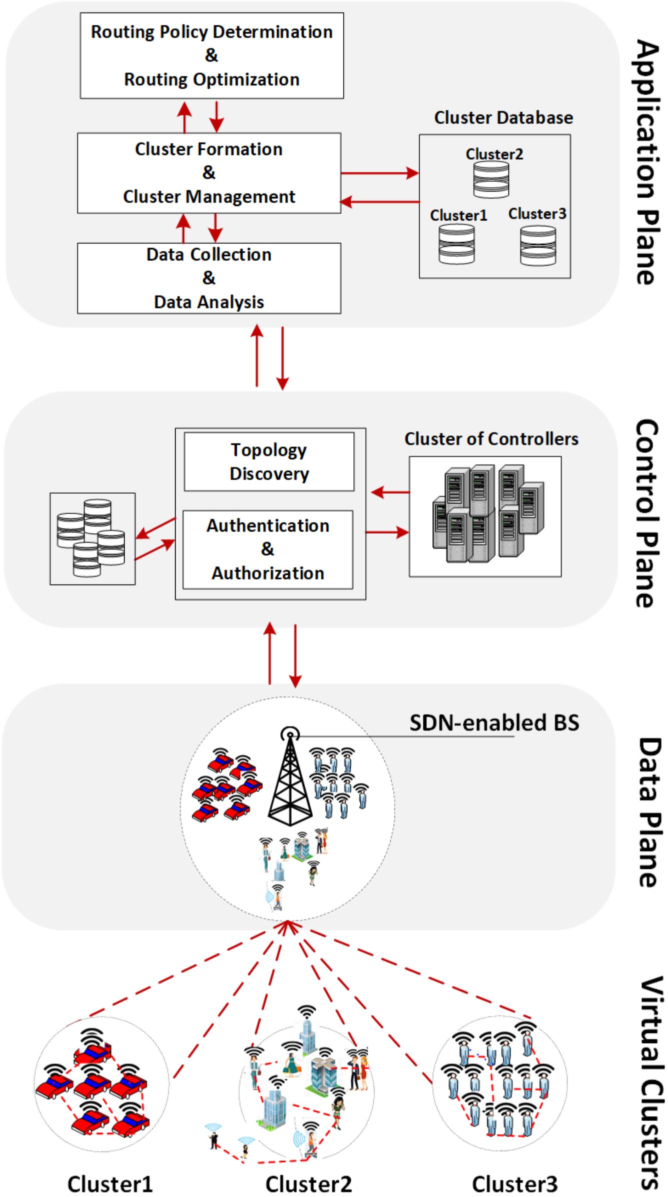


Figure 5.1: Multi-protocol framework for SDN-based cellular networks.

rounding areas, then the network requirements of one cluster are entirely different from the other one. Therefore, the two clusters can be virtually separated: one fully congested with almost static behavior and the other with low density and relatively high mobility rates. Accordingly, a suitable routing protocol must be identified for each cluster based on the cluster requirements. The SDN controller identifies each cluster by a unique ID, referred to as cluster-ID, and communicates with each cluster using an individual multicast address. When a node attaches to the BS, the controller sends the cluster ID and multicast address to that node while confirming the node association. The controller maintains a table called ClusterTable, which keeps cluster information. The 5G and WiFi frequencies are utilized for cellular and MD2D communications, respectively. The cellular channel is used to exchange control traffic between mobile nodes and the controller, download and upload from and to the BS to connect to the Internet or other networks that are not reachable

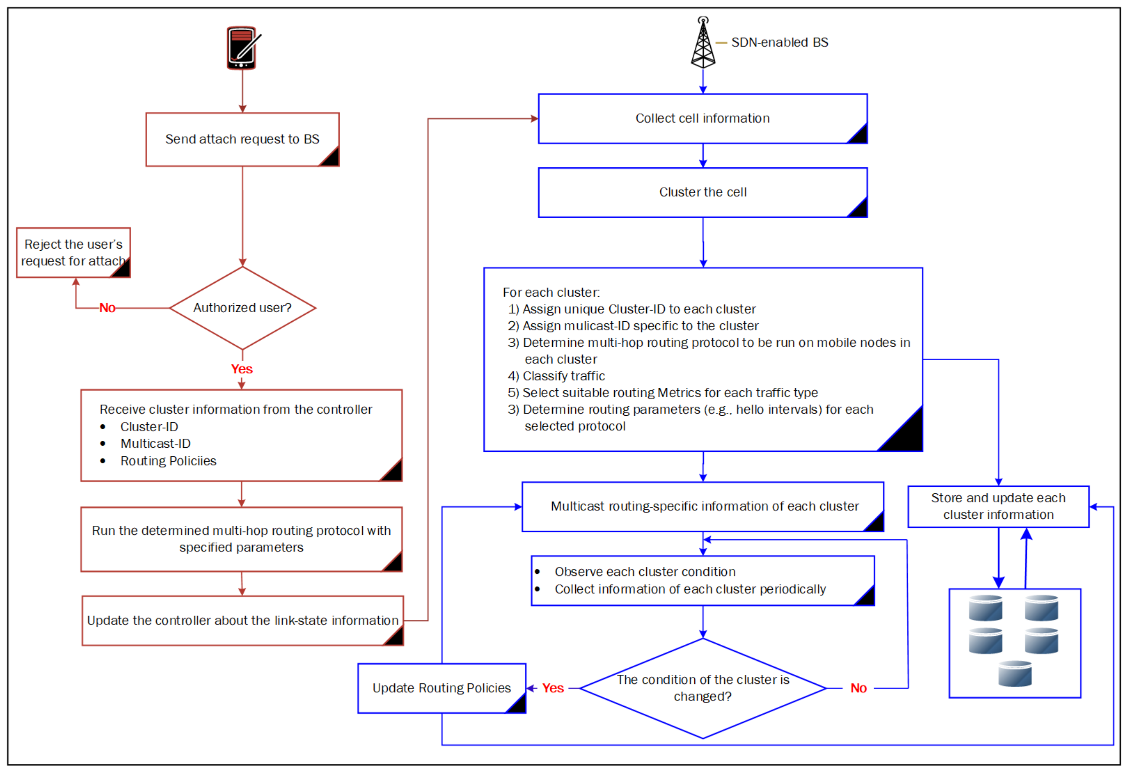


Figure 5.2: Multi-protocol framework flowchart.

through MD2D routing or MD2D cannot provide the required quality for that data flow compared to the cellular data transmission. The WiFi channel is used to exchange data traffic via multi-hop connections. The overall operation of the our proposed framework is summarized in Figure 5.2.

Mobile devices in our proposed framework can be programmed using two approaches: active and passive. In active programming, the SDN controller dictates the forwarding information to the cluster, whereas in passive programming, the functionality of the existing ad hoc routing protocols remains unchanged. The controller provides the information required for the protocols self-optimization, such as the maximum number of hops in an MD2D connection, routing metrics and parameters (e.g., hello intervals and topology control intervals). This significantly improves the overall functionality of the network. The focus of this chapter is active programming, and comparing the functionality of the proposed approaches is beyond the scope of this study and is deferred to our future work.

In active programming, the controller is engaged in multi-hop routing and provides forwarding information to mobile nodes. The controller decides on the multi-hop routing strategy to be used in a cluster, considering the cluster requirements and features. The forwarding information sent by the controller can vary depending on the selected routing method, reactive or proactive. In both reactive and proactive routing approaches, the controller builds an LSDB of the entire cluster. In our previous studies, we designed different routing strategies for active programming in cellular networks, namely HSAW [3], VARP-S [182], and SMDRP [183]. If a proactive routing approach is selected, such as HSAW, the controller multicasts the LSDB to the cluster. The controller also multicasts traffic policies carrying a list of traffic types, selected metrics for each type, and maximum allowed end-to-end cost for multi-hop routing. Each mobile node then builds the LSDB of the cluster. The source node of a data packet runs Dijkstra´s algorithm to find the least-cost path to the target and forward the data to the next hop. The controller periodically updates the mobile nodes with lost or newly added links to the cluster. In reactive approaches, such as VARP-S or SMDRP, mobile nodes are not required to maintain the LSDB of the cluster. If a node has a data packet, it sends a flow request message (FREQ) to the controller. The controller then provides forwarding information for each active flow. Further, the controller updates the involved nodes of a flow if any link failure occurs.

In this chapter, it is assumed that all nodes are under the coverage of the BS. The routing procedure for the nodes that are out of cellular coverage and the communication between mobile nodes in adjacent clusters are planned for future study.

5.4 DECISION-MAKING PROCEDURE FOR MULTI-PROTOCOL FRAMEWORK

We use the MCDM approach based on AHP [184] to select the best routing protocol for each cluster in terms of different network performance criteria and the high-demand services in a cluster. The following sections summarize the AHP-based decision-making procedure in the multi-protocol framework.

5.4.1 Overview of AHP-based Decision-making Procedure

In a decision-making process, there are several criteria and alternatives to be chosen from. MCDM evaluates explicitly multiple conflicting criteria in the discrete decision spaces, examines the alternative options based on the preference, and selects the best option. We employ the AHP method as it can be applied to various types of decision-making problems.

The AHP model is based on a hierarchical structure and was developed by Saaty [184] as a potential tool to manage qualitative and quantitative multi-criterion elements involved in decision-making behaviors. In this method, a fundamental scale in [184] is used to determine the importance level of each criterion compared to the others; and the preference level of its alternatives compared to one another for each criterion. The following steps are taken to conduct the AHP: 1) creating a hierarchy of the problem, 2) giving a nominal value to each level of the hierarchy, and 3) creating a pairwise comparison judgment matrix. The steps are detailed in the following paragraphs. To make a decision based on the AHP method, first, we determine the goal of decision-making (level 1 – Goal), i.e., selecting the most suitable protocol for a cluster, decision indicators (level 2 – Criteria), i.e., network performance criteria, and decision choices (level 3 – Alternatives), i.e., MD2D routing protocols. Next, we assign weights to the criteria based on their relative importance. A pairwise comparison matrix is generated, and a normalized eigenvector of the matrix is calculated to achieve the required relative weights of the criteria. To this end, a score/weight from the fundamental scale is given to each criterion to determine the importance

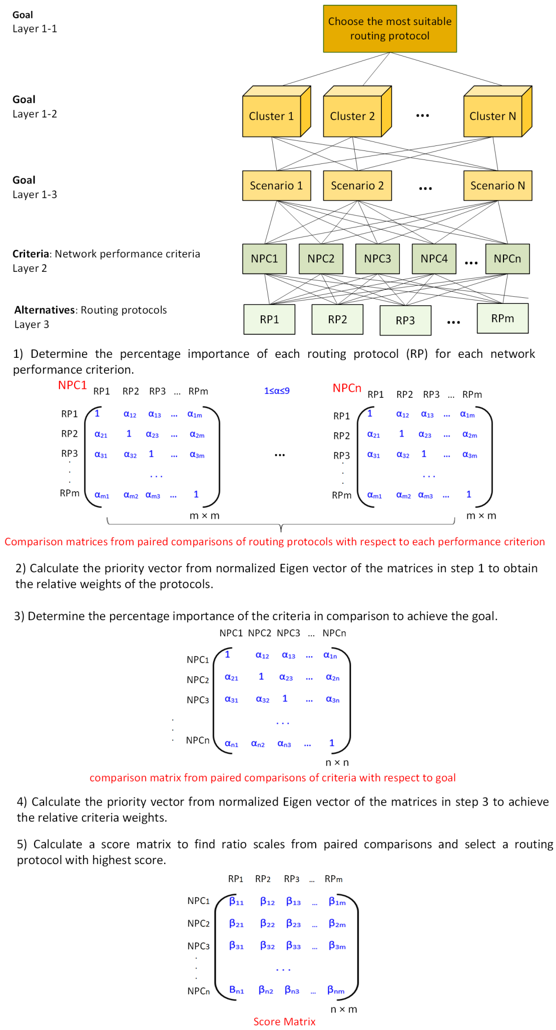


Figure 5.3: AHP-based decision-making model for multi-protocol framework.

level of each criterion compared to the others to achieve the goal. The score/weight ranges from 1 to 9, and higher values are preferable to smaller values. The size of the comparison matrix is $n \times n$, where n is the number of criteria. Then, the alternatives (i.e., routing protocols) are compared for each of the criteria to determine the percentage importance distribution of the alternatives. Consequently, n comparison matrices with size $m \times m$ are generated if the number of alternatives is m and there are n criteria for each alternative. Subsequently, the priority vectors of the created matrices are computed. After that, a scoring matrix with size of $n \times m$ is calculated to find the ratio scales from paired comparisons. This matrix is the overall composite weight of each alternative derived from the level-2 and level-3 weights. The overall weight is achieved by normalizing the linear combination of multiplication between the priority vector and weight. Finally, each alternative is ranked based on the score matrix [185, 186].

5.4.2 AHP-based Decision-making Model for Multi-protocol Framework

Figure 5.3 summarizes the proposed AHP-based decision-making process for the multi-protocol framework. The goal is to select the most suitable routing protocol for each cluster based on the cluster condition and requirements (e.g., node density, traffic density, average mobility rate, and high-demand services). The criteria are network performance parameters that determine the cluster performance, including E2E delay, packet loss, and energy usage. The alternatives are the available MD2D routing protocols operating differently in various network conditions. The importance of each performance criterion for a cluster, and the performance level of each routing protocol for each performance criterion, are evaluated using AHP. A score matrix is obtained from the AHP evaluation that indicates the most suitable protocol (i.e., the protocol with the highest score).

5.5 System Model

Three major parts of our implementation model are explained in this section, namely, the channel estimation, energy model, and network model. First, the cellular channel capacity is estimated using MIMO technology. Second, the link data rate is calculated based on the estimated channel. Third, the total energy consumed by each node to send or receive packets is computed. Finally, the specifications of the network model are detailed.

5.5.1 Cellular Channel Model

Our model integrates MIMO technology into the BS to increase the overall network performance and address challenges, such as signal propagation, increased number of users, and growing demand for low-latency communications. Further, we estimate the channel and the maximum data rate required for each user to establish a link to the BS. The following notations are used in our channel estimation and link capacity: $\mathbf{A}$, $\mathbf{a}$, and α represent matrix, vector, and scalar, respectively, and $\mathbf{A}^{-1}$ and $\mathbf{A}^{H}$ are the inverse and Hermitian (conjugate transpose) of matrix $\mathbf{A}$.

In MIMO, once the transmitted signal reaches the BS, each antenna receives multiple copies, which may be affected by scattering, shadowing, or pathloss. However, signals with the highest amplitude/power are selected for our channel estimation as proposed by [187, 188]. There are precoding matrices at both the transmitter and receiver to mitigate the channel effect and improve the QoS. To obtain the precoding matrices, we first model the communication channel between a mobile station (MS) and the BS as a collection of signals that are affected by multipath propagation, which is calculated as follows [189]:

$$\mathbf{H}(\tau) = \sum_{l=1}^{L} \mathbf{C}_l \delta(\tau - \tau_l) \tag{5.1}$$

$$\mathbf{C}_l = \begin{bmatrix} c^l_{1,1} & c^l_{1,2} & \cdots & c^l_{1,N_{BS}} \\ c^l_{2,1} & c^l_{2,2} & \cdots & c^l_{2,N_{BS}} \\ \vdots & \vdots & \ddots & \vdots \\ c^l_{N_{MS},1} & c^l_{N_{MS},2} & \cdots & c^l_{N_{MS},N_{BS}} \end{bmatrix} \tag{5.2}$$

where δ is the Dirac delta function, $\mathbf{C}_l$ is the complex channel gain matrix consisting of the l^{th} path which are delayed in time with τ_l, and $c^l_{N_{MS},N_{BS}}$ is the channel coefficient between the antennas of both sides (i.e., the k^{th} BS antenna and the m^{th} MS antenna for the corresponding l^{th} path). It is assumed that $c^l_{N_{MS},N_{BS}}$ is Rayleigh distributed.

The MIMO channel singular value decomposition (SVD) is calculated to estimate the precoding matrices, as below:

$$\mathbf{H} = \mathbf{U}\mathbf{S}\mathbf{V}^H \tag{5.3}$$

where $\mathbf{S}$ is a non-negative singular diagonal matrix with values of $\mathbf{H}$. $\mathbf{U}$ and $\mathbf{V}$ are the unitary matrices with sizes $N_{RX} \times N_{RX}$ and $N_{TX} \times N_{TX}$, respectively. In the downlink scenario, N_{RX} is referred to as the number of MS antennas (N_{MS}), and N_{TX} is the number of BS antennas (N_{BS}). The resulting matrices from SVD, $\mathbf{U}$, and $\mathbf{V}$ are the precoding matrices used at the transmitter ($\mathbf{F}_T = \mathbf{U}$) and receiver ($\mathbf{W}_R = \mathbf{V}$) in the MIMO system, respectively. The channel capacity is estimated based on the obtained precoding matrices as follows:

$$R = \log_2 |\mathbf{I}_{N_p} + \frac{P}{N_p}\mathbf{R}_n^{-1}\mathbf{W}_R^H\mathbf{H}\mathbf{F}_T\mathbf{F}_T^H\mathbf{H}^H\mathbf{W}_R| \tag{5.4}$$

$\mathbf{I}_{N_p}$ is the unitary matrix, P/N_p is the normalized total transmitted power, N_p is the number of transmitted packet symbols, P is the average total transmission power, and $\mathbf{R}_n$ is the noise covariance matrix.

Given the channel rate in $b/s/Hz$, the MIMO channel data rate (DR) between the transmitter and receiver is:

$$DR = N_{TX}BWR \tag{5.5}$$

where BW is the channel bandwidth in Hz, and N_{TX} is the number of antennas at the transmitter.

Based on the achieved data rate, the transmission delay (TD) of the link is estimated as below:

$$TD = \frac{S}{DR} \tag{5.6}$$

where S is the packet size in bits.

5.5.2 Energy Consumption Model

In our simulation, a classical first order radio model is used to calculate the energy consumed by the nodes for transmitting and receiving data [190, 191]. It is assumed that the power loss is based on multipath fading and free-space power loss. Based on the proposed energy model, the energy consumed by node$_i$ for transmitting or receiving a packet is calculated as below:

$$E^{TX}_{i\in N} = \begin{cases} e_{TX}S_i + e_{mp}S_i d^4, & if\ d > a; \\ e_{TX}S_i + e_{mp}S_i d^2, & if\ d < a; \end{cases} \tag{5.7}$$

$$E^{RX}_{i\in N} = S_i(e_{RX} + e_{da}) \tag{5.8}$$

where e_{TX} and e_{RX} are the consumed power by the electronic devices, S_i is the packet size that node$_i$ tries to transmit or receive ($i \in 1, 2, ..., N$), e_{da} is the data aggregation energy, d is the distance between the transmitter and receiver, and a is a threshold distance for the transmit power that determines the selection of a multipath or free-space model in the calculation. The value of a is defined as below:

$$a = \sqrt{\frac{e_{fs}}{e_{fm}}} \tag{5.9}$$

where e_{fs} and e_{fm} are the amplifier constants for the free-space model and multipath model, respectively. If the value of a is less than the distance between two nodes, the multipath model is selected. Otherwise, the free-space model will be employed.

Based on (5.7) and (5.8), let E_T denotes the total power consumed by N nodes in the network and is calculated as follows:

$$E_T = \sum_{i=1}^{N} E_i^{TX} + \sum_{i=1}^{N} E_i^{RX} \tag{5.10}$$

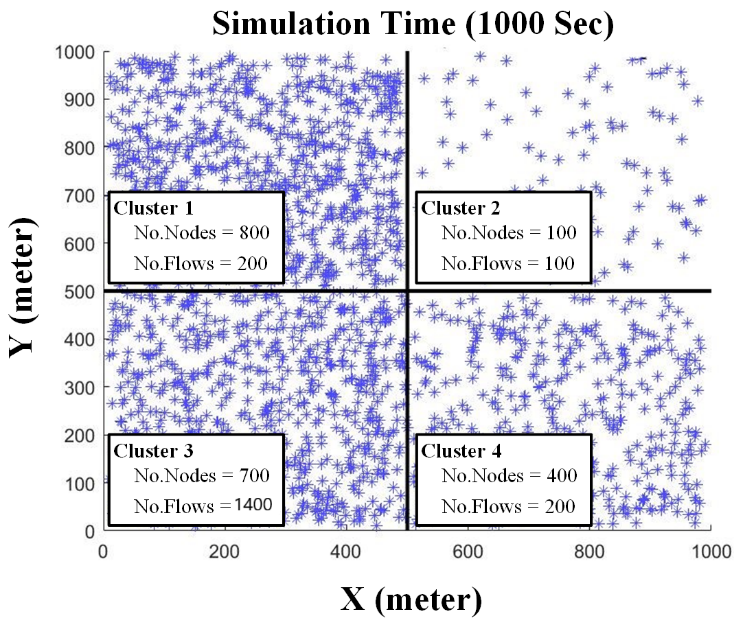


Figure 5.4: 2D representation of clusters.

5.5.3 Network Model

Two different frequency bands are used in our model: one with a licensed frequency for cellular communication (5G standards) and the other with unlicensed frequency for MD2D communications (IEEE 802.11g [3]). Further, four clusters are initially developed with varying node densities and mobility rates. The cluster density is categorized as sparse, semi-dense, and dense. Figure 5.4 shows the 2-dimensional view of our framework, and the specifications of the clusters are presented for each area. Clusters 1 and 3 are considered dense networks, and clusters 2 and 4 are considered sparse and semi-dense networks, respectively. Different mobility speeds are specified

[3]The proposed architecture can use other IEEE 802.11-based radio standards and is not limited to IEEE 802.11g.

Table 5.1: Simulation parameters.

Parameters	**Value**
Simulation tool	MATLAB
Simulation area	1 Km $\times$ 1 Km
Simulation time	1000 sec
Routing protocols	HSAW, VARP-S
Node mobility model	Random waypoint mobility
Wireless standard	IEEE 802.11g
Max BS antennas	64
Max MS antennas	4
Pathloss constant	3
Carrier frequency	5 GHz
Max bandwidth	80 MHz
Packet size	50 Mbits
Number of nodes	100, 400, 700, and 800
Node transmission range	75 m
Node speed	3, 10, and 20 (m/s)

to model pedestrians and vehicles of various types, i.e., $3m/s$ for the pedestrian network, and mobility of $10m/s$ and $20m/s$ for vehicles such as bicycles, scooters, and cars. The mobility pattern of mobile nodes is the random waypoint mobility model. Two different network scenarios, energy-independent and delay-sensitive, are assumed for the clusters to take into account the type of devices and the high-demand services when selecting a protocol. The performance of HSAW and VARP-S is compared for each cluster separately to identify which protocol is the potential candidate for each cluster. The AHP method is used to determine the most efficient protocol based on the simulation results. Subsequently, the selected protocol for each cluster is run on the mobile nodes of that cluster to evaluate the total throughput of the network.

5.6 Simulation Results and Performance Analysis

This section presents the performance analysis of the HSAW and VARP-S protocols for each cluster in terms of various network performance criteria. Then, the AHP decision-making procedure to select the most appropriate protocol is described, and the performance of multi-protocol is compared with a single-protocol framework. The simulation parameters are listed in Table 5.1.

For simplicity of analysis, the following assumptions have been made:

- The controller has oversight of the entire cell and initially has clustered the cell (i.e., 4 clusters) based on the traffic density.
- All nodes are under the coverage of the BS.
- The energy specifications of all nodes are the same, and the nodes in the network have the same initial energy. Furthermore, for the HSAW protocol, the energy consumed by the

mobile nodes to run Dijkstra´s algorithm is ignored.

- Three average mobility rates are considered for mobile nodes: 3 m/s, 10 m/s, and 20 m/s.

5.6.1 Simulation Analysis of HSAW and VARP-S routing protocols

At the initial stage of our simulation, a random number of flows with different sources and destinations are generated. The controller then provides different forwarding information to nodes according to the selected routing protocol for the cluster (i.e., HSAW or VARP-S). In the case of VARP-S, the source node sends a flow request for a specific target to the controller, and the controller replies with forwarding information (i.e., a list of intermediate hops to the target). Then, the source node attaches the full route to the data packets and forwards the packet to the next hop. In HSAW, the controller multicasts the LSDB of the entire cluster. Subsequently, if a node has a data packet to send, it runs Dijkstra's algorithm to find the next hop to the target of the data packet. Each protocol acts differently in response to the link failure. In HSAW and VARP-S, the upstream node of the failed link forwards the broken link information to the controller, and subsequently, the controller multicasts that information to the cluster. In HSAW, the upstream node determines a new route to the target by running Dijkstra's algorithm on the cluster LSDB. In VARP-S, the source node waits to receive the updated forwarding information from the controller. However, HSAW is a proactive protocol in nature. Hence, all the nodes continually update their LSDB and calculate routes whenever any changes occur in the cluster.

Figures 5.5, 5.6, 5.7, and 5.8 demonstrate the functionality of the routing protocols for 4 clusters based on four network performance criteria: cellular routing overhead, E2E delay, energy consumption, and packet loss. Each figure includes three sub-figures presenting the performance of the protocols for three mobility rates. Each cluster experiences various traffic densities and number of flows as detailed in Figure 5.4.

5.6.1.1 Overhead

Figure 5.5 presents the cellular routing overhead of the protocols. As highlighted in the figure, HSAW produces a significant overhead on the cellular channel in all the clusters compared to VARP-S. Since in HSAW, the LSDB of the whole cluster will be multicast by the controller. Therefore, for a large number of nodes, the size of LSDB increases, leading to higher overhead. Whereas in VARP-S, only route request and route reply control messages are exchanged between the source node and the controller for the demanded targets. For HSAW, cluster 1 experiences higher overhead than clusters 2 and 3 due to more node density. However, cluster 3 shows significant routing overhead compared to the other clusters because of high network density and data traffic. It can be concluded that for highly dense networks, VARP-S operate better as fewer control messages are required for route discovery and route maintenance. Moreover, overhead increases exponentially as the mobility rate increases due to frequent link failure and the need for more updates.

5.6.1.2 End-to-end delay

The HSAW provides better E2E delay compared to VARP-S, as shown in Figure 5.6. In the HSAW for route discovery and maintenance, nodes run Dijkstra's algorithm to determine the route to the target and do not wait for forwarding information from the controller. In VARP-S, the source node has no knowledge of the cluster and must wait for the controller instructions. Cluster 3 experiences the highest E2E delay compared to other clusters as node density and the number of flows are high. The least E2E delay can be seen in cluster 2 because of low traffic density. The figure also shows that by increasing mobility, the E2E rises accordingly.

5.6.1.3 Energy

The HSAW provides better E2E delay compared to VARP-S, as shown in Figure 5.6. In the HSAW for route discovery and maintenance, nodes run Dijkstra's algorithm to determine the route to the target and do not wait for forwarding information from the controller. In VARP-S, the source node has no knowledge of the cluster and must wait for the controller instructions. Cluster 3 experiences the highest E2E delay compared to other clusters as node density and the number of flows are high. The least E2E delay can be seen in cluster 2 because of low traffic density. The figure also shows that by increasing mobility, the E2E rises accordingly.

It is notable that in HSAW, nodes must run Dijkstra´s algorithm more frequently in response to the topology changes, which leads to more processing overhead and further energy consumption. However, this is not taken into account in this analysis.

5.6.1.4 Packet Loss

Figure 5.8 demonstrates the functionality of the protocols in terms of packet loss. Except cluster 3, HSAW provides more reliable data delivery than VARP-S because it has a faster response to the link failure, as explained in Sec. 5.6.1.2. In cluster 3, the network experiences a higher number of flows leading to nodes consume more energy in HSAW compared to VARP-S, as depicted in Figure 5.7. Consequently, nodes run out of the battery, which leads to more packet loss and link failure in HSAW. Figure 5.8 also shows that by increasing the mobility rate, the packet loss increases due to frequent link failure.

To further investigate the impact of the number of flows on the packet loss in dense networks, in Figure 5.9, we compare the performance of the protocols for cluster 3 with 700 nodes, when the number of flows increased from 0 to 1400. As shown, HSAW provides better performance when the number of flows is less than 450 (see Figure 5.9b). However, when the number of flows exceeds 450, nodes in HSAW deplete their batteries, resulting in a high number of dead nodes in the cluster (see Figure 5.9a). This significantly increases the packet loss in the HSAW. Consequently, VARP-S is a better choice for cluster 3 when the number of flows is greater than 450 (see Figure 5.9c).

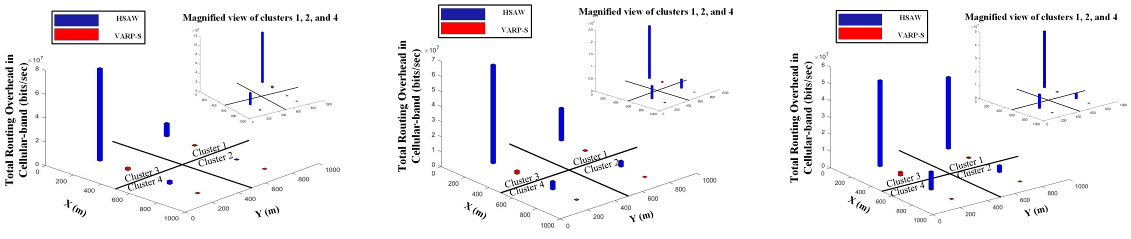
5.6.2 Routing Protocol Decision-making for Each Cluster

We use the MCDM approach based on AHP [184] to select the best routing protocol for each cluster in terms of the defined criteria (i.e., E2E delay, packet loss, energy consumption, and cellular overhead). Further, the high-demand services and energy-dependency of the devices in the cluster are taken into account when selecting the routing protocol. The selection of protocols is investigated for two different scenarios to demonstrate the advantages of having a multi-protocol framework when different protocols are selected for each cluster based on the cluster requirements. The first scenario is for the clusters where nodes are energy-independent, such as vehicular networks. The second scenario is for clusters wherein the high-demand services are delay-sensitive applications that time matters most, such as hospitals or public safety scenarios. In such applications, the level of importance of cellular overhead is very low and can be ignored in AHP decision-making.

As shown in Figure 5.11, the AHP hierarchy architecture is adapted to our model. The goal is to select the best routing protocol for each cluster based on the defined scenarios in Sec. 5.6.2. The criteria are E2E delay, packet loss, energy consumption, and cellular overhead. Two potential routing protocols, VARP-S and HSAW, are shortlisted alternatives for each cluster. Due to space constraints, we only demonstrate the AHP process for cluster 1 for the first scenario that nodes are energy-independent. For such applications, we can ignore the energy criteria because the weight of this criterion is too small and its impact on the overall decision is negligible. The AHP online calculator [192] is used to calculate the priorities of the criteria and alternatives. Based on the decision criteria, HSAW with a priority of 0.732 is by far the most suitable routing protocol for cluster 1. Similarly, the same process can be applied to the other clusters. The results of decision-making are summarized in Table 5.2 for the previously defined scenarios: energy-independent and delay-sensitive. The most suitable protocol for each cluster is concluded in the "Decision Strategy" section of Table 5.2 based on the AHP results.

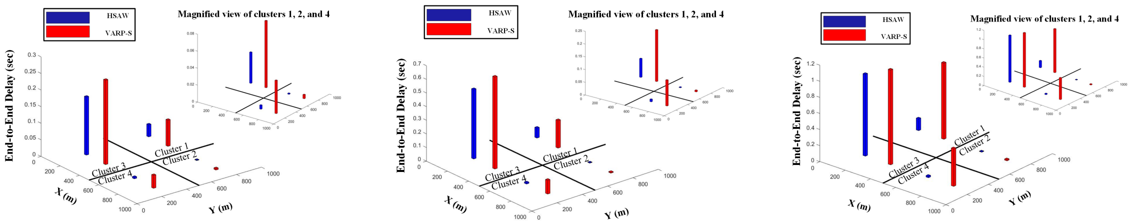
5.6.3 Performance Analysis of Multi-protocol Framework vs. Single-protocol Framework

From the simulation results, it is concluded that for a sparsely distributed cellular network, both protocols perform well as neither are pushed to their operational limits. However, there are clear distinctions between the protocols when either traffic flow or node density is high. To clarify and show how a specific network performance parameter can be improved using our proposed multi-protocol framework, we investigate the framework performance in terms of packet loss. At first, one type of protocol is run in all the clusters to determine the performance of each protocol (see bars VARP-S and HSAW in Figure 5.10). As shown in Figure 5.8, except cluster 3, HSAW provides better performance than VARP-S, owing to its faster response to network failures, as discussed in Sec. 5.6.1.4. The "Multi" bar in Figure 5.10 represents the advantage of leveraging the best features of both protocols to provide reliable data delivery. This is achieved by running the most efficient protocol in each cluster, i.e., VARP-S for cluster 3 and HSAW for other clusters, to minimize the total packet loss in the entire network.



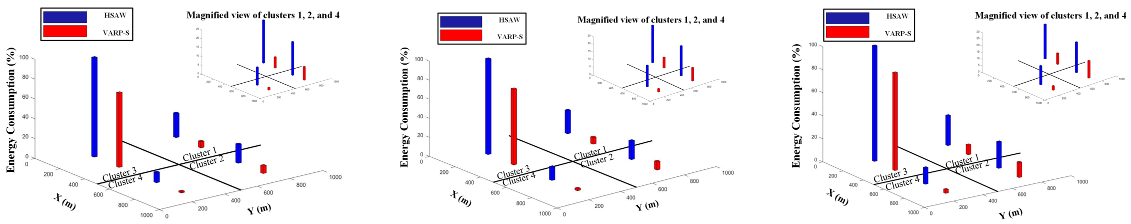
(a) Node´s velocity = 3 m/s (b) Node´s velocity = 10 m/s (c) Node´s velocity = 20 m/s

Figure 5.5: Total routing overhead in cellular-band (bits/s).



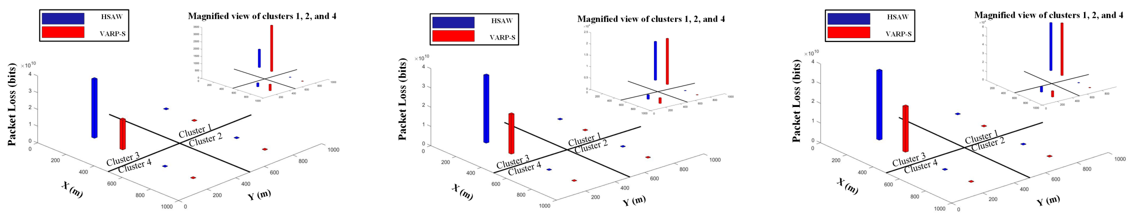
(a) Node´s velocity = 3 m/s (b) Node´s velocity = 10 m/s (c) Node´s velocity = 20 m/s

Figure 5.6: End-to-end delay (sec).



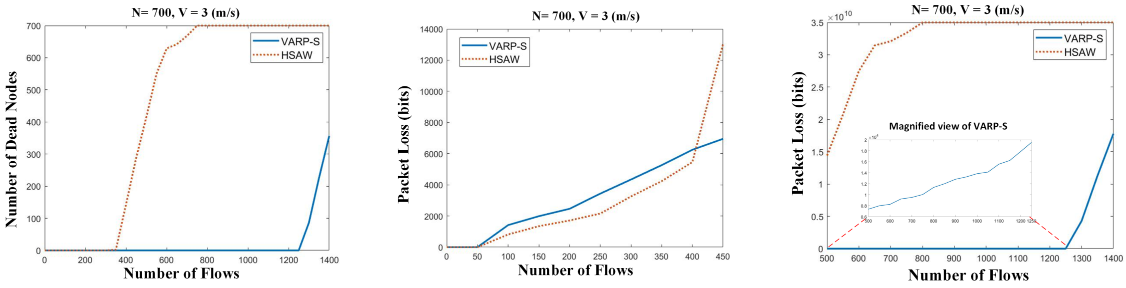
(a) Node´s velocity = 3 m/s (b) Node´s velocity = 10 m/s (c) Node´s velocity = 20 m/s

Figure 5.7: Energy consumption of nodes (%).



(a) Node´s velocity = 3 m/s (b) Node´s velocity = 10 m/s (c) Node´s velocity = 20 m/s

Figure 5.8: Packet Loss.



(a) No.Dead nodes vs. No.Flows

(b) Packet loss vs. No.Flows (0-450 flows)

(c) Packet loss vs. No.Flows (500-1400 flows)

Figure 5.9: Packet loss over a densely populated network.

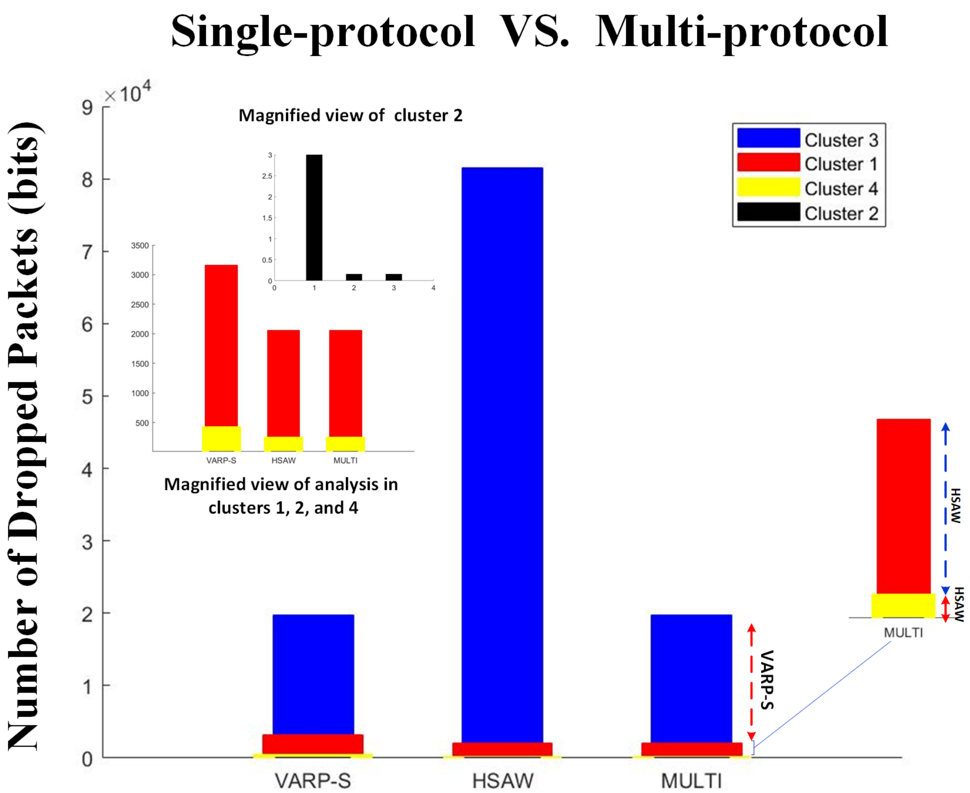


Figure 5.10: Performance analysis of multi-protocol framework vs. single-protocol framework in terms of packet loss.

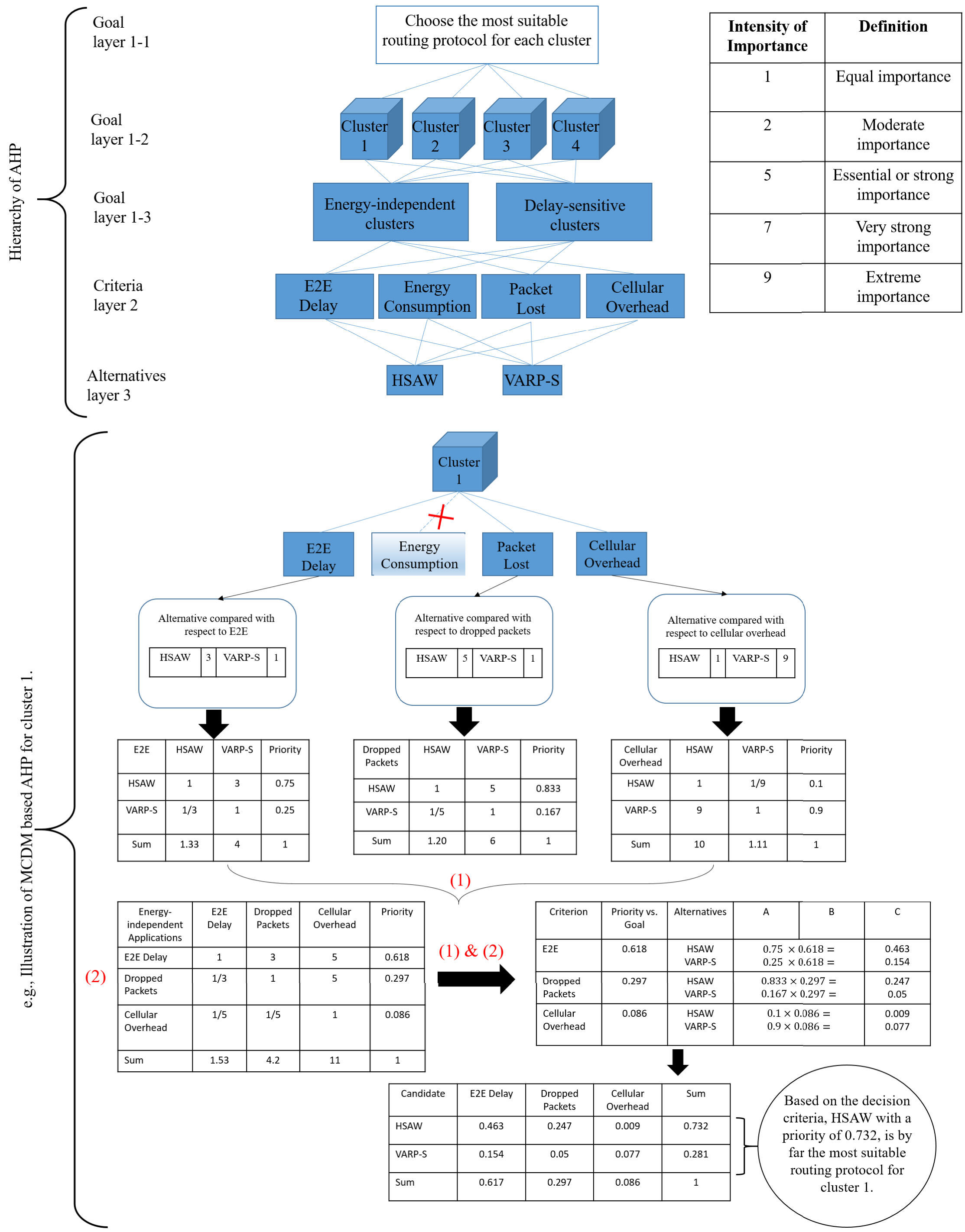


Figure 5.11: AHP decision-making process to select the most suitable routing protocol for each cluster.

Table 5.2: Suitable protocol for energy-independent and delay-sensitive clusters based on the AHP decision-making.

Decision / Cluster	Decision Strategy	
	Energy-independent clusters	**Delay-sensitive clusters**
Cluster 1	HSAW	HSAW
Cluster 2	VARP-S	VARP-S
Cluster 3	VARP-S	VARP-S
Cluster 4	HSAW	HSAW

5.7 Conclusion

Until now, multi-hop device-to-device (MD2D) routing in mobile wireless networks has been designed to operate over a single-protocol routing framework. Consequently, to the best of our knowledge, no framework has been developed to enable the dynamic deployment and switching of multiple MD2D routing protocols under one framework. This chapter presented a novel SDN-based multi-protocol routing architecture to enable the use of multiple MD2D routing protocols under one framework. In this chapter, an SDN controller split the network into different clusters based on the observed network geometry. Moreover, the performance of the selected protocols, HSAW and VARP-S, was analyzed in each cluster in terms of energy, E2E delay, packets dropped and overhead. Building on our analysis and results, the AHP MCDM method was used to choose the most suitable routing protocol for each cluster. Our results showed that our proposed framework provides a better performance than traditional single-protocol network scenarios.

This study has the potential to be a steppingstone for exploring the integration of different multi-hop routing protocols into a multi-protocol framework, wherein each protocol is used in its suitable network environment. In the future, we will analyze the proposed passive programming and compare its performance with active programming approaches. We will also extend our simulation study and perform a detailed comparison with other existing routing frameworks. We also intend to explore the potential integration of our proposed multi-protocol framework over O-RAN. O-RAN specification defines an open architecture for 5G and beyond wireless networks, which has opened the pathway towards developing and integrating novel network applications for cellular networks [193–195]. O-RAN provides open-source software for modeling and developing protocols for the RAN. This enables us to develop novel network applications and protocols, in the form of NFVs (also referred to as rApps and xApps in the O-RAN architecture) over the RAN.

6

Software-defined Networking-based Adaptive Routing for Multi-hop Multi-frequency Wireless Mesh

6.1 Introduction

Multi-hop networking with high end-to-end throughput is critical to many safety-related applications, such as underground mining, disaster rescue and recovery [55, 196–198]. Multi-hop networks play an important role in sending back images and videos captured by sensors or robots to command posts and control centers. Existing commercial hardware platforms supporting multi-hop routing under wireless mesh configurations have several limitations, including the tight coupling of the control plane and data plane that makes the network inflexible and reduces the network performance, as well as proprietary hardware and hard-coded network protocols leading to high operational costs [19, 199].

Despite many studies on multi-hop networking, commercially available systems are scarce. Several commercially available bespoke systems are typically bulky, energy-hungry, and are designed for static deployment. Moreover, proprietary networking protocols are often used with limited programmability, as revealed in our literature review, as discussed in Section 6.2.

This chapter presents a new development of lightweight, low-cost, portable mesh nodes (by our Commonwealth Scientific and Industrial Research Organisation (CSIRO) team), which can be deployed quickly and support large numbers of hops (with slow or no diminishing of end-to-end throughput), as shown in Figs. 6.1 and 6.2. Designed for underground robotic exploration missions, the nodes are expected to be carried by robots and dropped on the go. Once dropped, a node is activated, joins the earlier nodes, and extends the multi-hop mesh network, which sends the robot's observation back to a command post. The node is developed based on a commercial-off-the-shelf ESP32 WiFi module due to its small footprint and excellent programmability. The ESP32 module operates in the 2.4GHz frequency band, where there are three non-overlapping WiFi channels. Our new node is equipped with three ESP32 modules, allowing it to operate in any of the three non-overlapping channels at a time. The use of different channels for the upstream and downstream of every node, subject to channel availability, mitigates co-channel and adjacent frequency interference.

A key aspect of this work is the development of a new software-defined networking (SDN) architecture and routing algorithms, which can operate the new ESP32-based network, utilize the three non-overlapping WiFi channels (namely, channel coloring), and produce routes and channel selections adaptively. We extend the Dijkstra's algorithm, which we refer to as E-Dijkstra's algorithm, to discover the route with the least cost while selecting the WiFi channels that each node uses for its upstream and downstream (as opposed to finding the route only, as done in the classical Dijkstra's algorithm). With N available frequency channels ($N = 3$ in our ESP32-based hardware platform), N^2 parallel Dijkstra's routing processes are maintained in the E-Dijkstra's algorithm to evaluate the N^2 possible routing costs between a node and a possible upstream node. The N^2 parallel routing processes interact to evaluate the routing costs under different possible channel arrangements of the nodes.

The proposed E-Dijkstra's algorithm also considers the WiFi traffic density (and subsequently the co-channel interference) when evaluating the throughput of each link. Our new routing metric allows the cost of each of the multiple channels to be separately recorded and jointly updated in the E-Dijkstra's algorithm, and allows the background WiFi traffic of each channel around a node to be factored in the cost.

An extensive simulation study is conducted using Mininet-WiFi to investigate the performance of the proposed E-Dijkstra's algorithm. The simulation results indicate that the algorithm can achieve substantially higher end-to-end throughput by adaptively selecting routing paths and frequency channels, especially in the presence of background WiFi back, as compared to the shortest-path-based routing strategy.

The rest of the chapter is organized as follows: Section 6.2 reviews existing commercial and academic solutions for wireless multi-radio multi-hop communications. Section 6.3 describes the developed hardware platform and SDN-based adaptive routing for the platform. Sections 6.4 and 6.5 provide MATLAB and Mininet-WiFi based simulation studies. Section 6.6 concludes the chapter.

6.2 Related Work

Multi-hop wireless networking is a cost-effective solution to the fast deployment of communication infrastructure in the case of emergencies [196, 200–202]. However, the use of a single frequency in a multi-hop network would suffer from a rapid decline in the end-to-end capacity with the growing number of hops due to the half-duplex nature of the relay. Multi-frequency/channel multi-hop networks can potentially support consistent end-to-end throughput, even when the number of hops increases [203–206]. This is due to the fact that the use of multiple frequency interfaces at a node allows the node to operate full-duplex. IEEE 802.11 is one of the most widespread standards for connecting wireless nodes in multi-hop networks. However, existing commercially available and typically bespoke WiFi systems do not support multiple frequency channels when operating in a multi-hop setting. Moreover, the number of non-overlapping channels in IEEE 802.11 standard is limited (i.e., 3 and 12 non-overlapping channels in IEEE 802.11b/g and IEEE 802.11a, respectively) [207].

In multi-radio multi-channel multi-hop wireless connections, channel assignments and interference management are challenging, and designing efficient channel assignment algorithms, routing metrics, and policies is necessary to effectively utilize and allocate channels and maximize end-to-end throughput capacity by minimizing the interference. Prior research on interference management of multi-radio wireless networks [208–214] was predominantly based on a distributed routing framework and did not comprehensively investigate how integrating an SDN-based architecture could be leveraged to mitigate interference when creating large multi-hop routes. To this end, SDN could play an important role by providing a programmable framework wherein the control functions are logically centralized in one or more control entities, referred to as the controller. The controller has a global view of the network by collecting real-time network information from forwarding devices via open APIs. Various network applications and control algorithms run on top of the controller to dynamically manage the network and to optimize resource allocation based on an updated global view of the network [19, 215].

Several studies have been conducted on SDN-based multi-radio multi-channel multi-hop wireless networks wherein routing and radio resource allocation are dynamically controlled and optimized by a centralized controller [1, 93, 173, 216–219]. Regardless of the proposed SDN-based frameworks, studies on practical implementation are still lacking. The author of [220] deployed an SDN-based wireless mesh testbed called SoftMesh. In the proposed testbed, a centralized controller programs multi-radio wireless routers and manages the network. A channel assignment algorithm is designed for SoftMesh to increase network capacity by minimizing the interference and number of channels used in wireless mesh networks. The algorithm considers the available channels, number of radios in each router, number of adjacent nodes, and number of wireless interfaces at gateway nodes to find the possible number of interference-free channels and links from gateway nodes to adjacent nodes. Their results showed that by increasing the number of channels used in parallel, the transmission delay is diminished by 75%, and the network throughput increases linearly. However, the proposed architecture cannot be applied to inexpensive commercial-off-the-shelf (COTS) WiFi chipsets because of the adjacent frequency interference between channels in a band in the chipsets.

A number of commercial and prototype hardware platforms have been proposed for multi-radio multi-hop networks (see Table 6.1). [223] performed an experiment in a multi-interface WMN

Table 6.1: Comparison of multi-radio multi-hop hardware platforms.

Reference	Support of Multi-channel	Commercial Availability	Software Programmability
RAJANT Kinetic Mesh [221]	No, end-to-end throughput declines fast with growing number of hops	Yes	Proprietary software and protocol with limited programmability
Ubiquiti radio from UniFi [222]	No, end-to-end throughput declines fast with growing number of hops	Yes	Proprietary software and protocol with limited programmability
M. A. Filho et al. [223]	Yes, two-radio devices (IEEE 802.11n/2.4&5.8 GHz)	No, it is a testbed based on router products	Yes, used the OpenFlow protocol for routing and static frequency arrangement
M. Tanha et al. [224]	Yes, multi-radio mesh routers (IEEE 802.11n/5GHz)	No, this is a testbed that required wired connection for control signalling	Yes, used the OpenFlow protocol for routing and static frequency arrangement
D. Sajjadi et al. [1]	Yes, two-radio mesh routers (IEEE 802.11n/5GHz)	No, it is a testbed based on router products	Yes, used NET-SNMP v5.7.2 on mesh routers to apply channel frequency configuration on radios
Proposed Hardware Platform	Yes, three non-overlapping channels in 2.4GHz	Yes, built on commercially available ESP32 WiFi modules	Yes, highly reprogrammable in support of custom-designed protocols

testbed with homogeneous multi-channel routers (configured with the IEEE 802.11n in the 2.4 GHz and 5 GHz ISM band) to compare the performance of OpenFlow with traditional mesh protocols, i.e., better approach to mobile ad-hoc networking (BATMAN), ad hoc on-demand distance vector (AODV), and hybrid wireless mesh protocol (HWMP), in terms of throughput, jitter, delivery rate, and the number of lost packets. They developed two mesh networks, one for the control plane and the other for the data plane. They experimented the impact of different channel and interface assignments in their testbed. Their results showed that the OpenFlow-based network provides more throughput compared to the traditional mesh protocols since all route calculations are performed in the controller. [224] developed a prototype for reactive failure recovery in multi-radio multi-channel SDN-based wireless mesh networks and compared the performance of the proposed prototype with traditional mesh protocols (i.e., optimized link state routing protocol (OLSR), BATMAN-adv and Open80211s) in terms of recovery time. In their experiment, two multi-interface routers were connected to two hosts via their Ethernet interface to maintain the end-to-end connectivity between the hosts. Except for these two routers, the remaining mesh routers in the network were connected to the controller via cable. The multi-interface routers were connected to the backbone mesh routers via their wireless interfaces while using non-overlapping channels for their connections. In this way, three disjoint paths were created over the backhaul mesh network to maintain connectivity. Further, an out-of-band management network was set up to enable the controller to manage the mesh routers. Their experimental results showed that the proposed prototype provides better performance than traditional mesh protocols in terms of recovery time. [1] developed an experimental testbed to compare the performance of SDN-based wireless mesh networks with traditional protocols (i.e., OLSR, BATMAN-adv, and Open80211s) in terms of channel switching latency in multi-hop multi-radio network scenarios. They created a ring topology of seven mesh routers as a multi-radio multi-channel mesh backhaul. Each router was equipped with two wireless radios operating in the IEEE 802.11n non-overlapping channels. All mesh routers were connected to a network management server (NMS) via an out-of-band Ethernet-based connection. NMS performed synchronization and channel switching processes in the network. In addition to the aforementioned tasks, NMS handled routing in the SDN scenarios. On each mesh router, NET-SNMP v5.7.2 was installed to apply the new channel frequency configuration of the radios and to monitor the mesh routers. Their results indicated that SDN-based routing reduces the channel switching latency compared to traditional mesh protocols. However, the focus of the above work has been on developing a proof of concept for integrating SDN-based concepts and protocols such as OpenFlow into multi-hop mesh routing. Very little research has been conducted on the use of SDN concepts to address or improve some of the fundamental challenges of multi-hop routing. Furthermore, the experimental work has been based on testbeds that have been put together from commodity hardware. In this research, light-weight SDN-based multi-radio hardware is developed. We also develop a new routing metric and interference mitigation method that enables us to significantly increase the multi-hopping capacity of the network.

RAJANT corporation proposed a high-bandwidth wireless mesh network [221], called "Kinetic Mesh", that provides a reliable network for remote management and control of a large volume of assets in safety-related applications such as underground mining. In the Kinetic Mesh network, each multi-radio wireless mesh node runs the InstaMesh network protocol to route data traffic through the best available path and frequency with extremely low overhead. In the Kinetic Mesh network, each node acts independently and can switch between the available radio frequencies by analyzing the best path at the node level. Each node performs a real-time evaluation to determine the best link to the destination of the packet. This reduces network overhead and accelerates the response to network changes. However, the platform does not support software-defined networking and still relies on tightly coupled specialized hardware and software (i.e., control and data planes).

UniFi Corporation built a new mesh AP [222], named "UniFi AC Mesh". The AP supports alternative antenna options. UniFi introduced Wireless Uplink to extend the wireless connectivity between APs more flexibly and conveniently. It allows each AP with a wired data connection to act as a base station/Uplink AP for up to four other access points on a single band. However, the proposed platform is suited for static applications, and the bandwidth is reduced in large-range multi-hop connections. Further, bulky, heavy, and expensive devices are used in the proposed architecture, which cannot suppress adjacent frequency interference. To address this, we developed our own device for faster deployment and faster recovery.

This chapter proposes a lightweight SDN-based multi-radio hardware platform that supports frequency coloring and simultaneous transmissions between wireless nodes. A novel metric is proposed that takes into account the frequency coloring, channel capacity, and probability of successful transmission of the nodes. The Dijkstra's algorithm is also extended to support frequency coloring in our proposed platform. The algorithm takes into account all possible configurations of alternating frequency arrangements for an end-to-end connection request and selects a path with the maximum throughput. In the proposed multi-radio platform, one radio is used to connect to the SDN controller and multiple non-overlapping frequencies are used for data traffic.

6.3 Proposed Architecture for Designed Platform

Figure 6.1 shows the architecture of our designed platform, which supports multi-hop multi-frequency routing with the assistance of a controller. The detailed design of the platform and

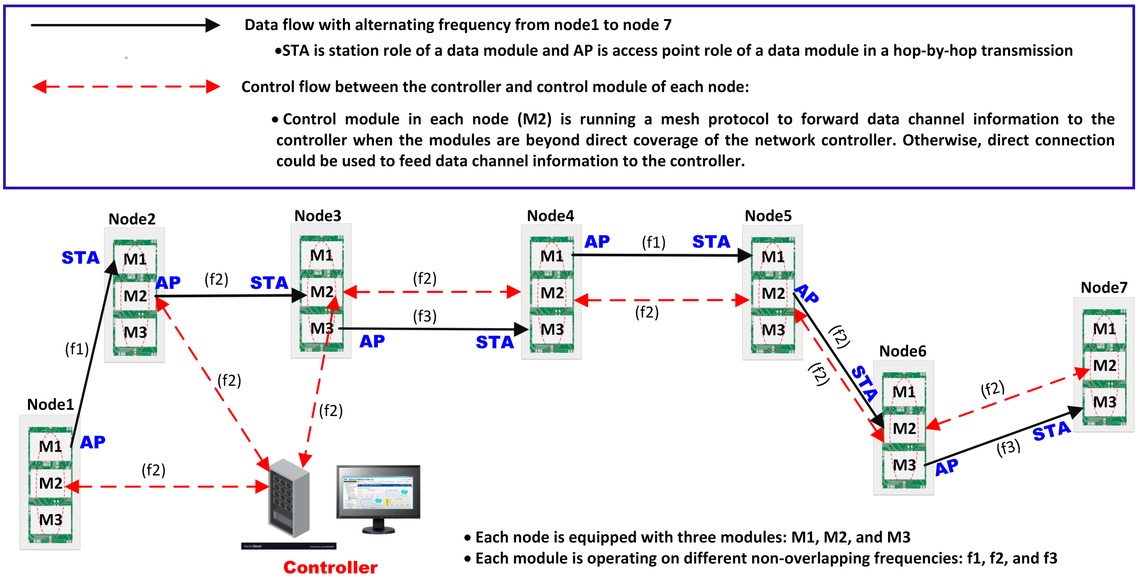


Figure 6.1: System architecture.

its functionality are described in the following subsections.

6.3.1 System Architecture

The proposed architecture integrates multiple commercial-off-the-shelf (COTS) WiFi modules to play the role of a data switch. Each WiFi module operates in non-overlapping frequency bands using channel-filtering techniques. A node can transmit and receive in different link directions (including upstream and downstream) simultaneously without self-interference. To develop a low-cost device, a token ring message bus is implemented between the radio modules using the Ethernet medium access interface provided on the radio modules. The other alternatives to connect WiFi modules are inter-integrated-chip sound (I^2S), inter-integrated-circuit (I^2C), and serial peripheral interface (SPI). However, Ethernet provides higher bandwidth and greater simplicity as the main functions, such as packetization, error correction, and flow control, are supported by the IP stack.

While the modules in a device need to talk with each other, yet each module has only a single Ethernet interface. One option is to connect the Ethernet interfaces of the radio modules in a device using a hub. However, this option is costly and increases the power consumption of the device. The other option used in our platform is an Ethernet ring, which is a better solution in terms of cost and power consumption. The Ethernet driver of each module is modified to facilitate communication between modules and eliminate the loops in the Ethernet ring.

To separate the control plane from the data plane, one module, referred to as control module, is used to form the control plane. The other modules, referred to as data modules, are used to constitute the data plane. The control module is also used for data plane functions. However, its priority is given to the control functions.

Figures 6.2 and 6.3 show the physical and logical views of the hardware platform. It is notable

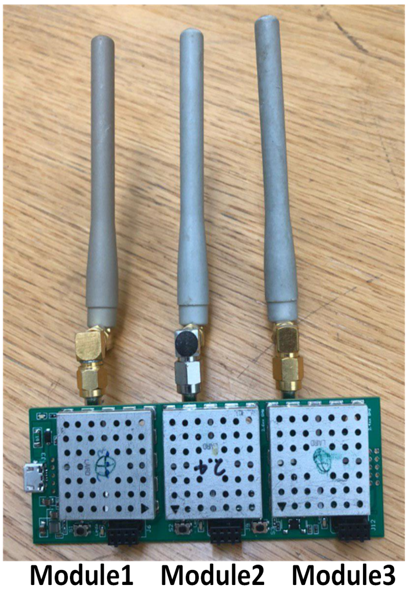


Figure 6.2: Physical view of designed hardware platform.

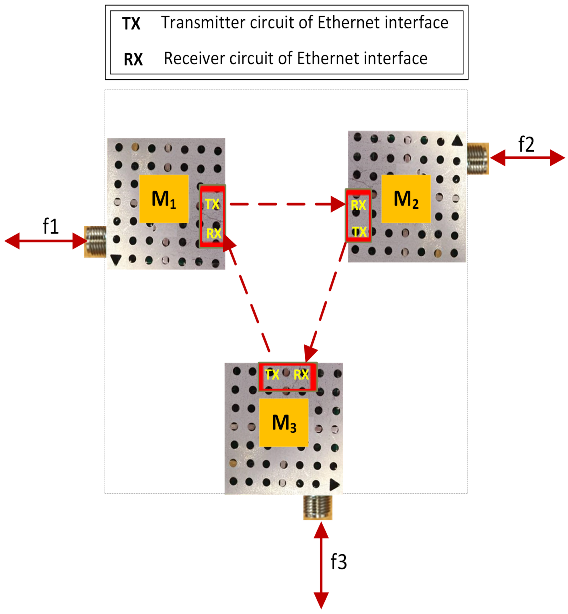


Figure 6.3: Logical view of the designed platform.

that in our proposed platform, we can have more modules on the data plane. Further, the selection of the frequency band used by each module is important and needs to be taken into account. Since the modules are placed close to each other, the frequency bands for two adjacent modules should be selected to minimize the adjacent and co-channel interference.

The main purpose of using three WiFi interfaces operating on non-overlapping frequency bands for the data plane functions is to minimize the interference between wireless links in multi-hop connections. The other purpose is to improve the network throughput by enabling the nodes to transmit and receive data traffic simultaneously since a single WiFi interface cannot transmit and receive on the same frequency concurrently. Consequently, the use of non-overlapping frequency bands enables simultaneous transmission and reception in each node.

Each node forwards its link-state information (i.e., the list of one-hop neighbors and the received signal strength indication (RSSI) from the neighbor) to the controller, using its control module. The controller can build a link-state database (LSDB) of the entire network using the received information. The controller can instruct the nodes to forward their data traffic on the data plane. The control module of each node receives the control information from the controller and forwards the information to the other modules on the data plane. On the data plane, each module can play two roles, namely, the AP role when sending data traffic and the STA role when receiving data traffic. The controller determines the role of each data module in the end-to-end connectivity.

As indicated in Figure 6.3, there are three WiFi modules operating in non-overlapping frequency bands: f_1, f_2, and f_3. The modules are connected to one another through Ethernet interfaces creating a LAN broadcast domain and forming a high-speed Ethernet ring. Figure 6.3 illustrates the way in which the transmitter pins of a module (T_X) are connected to the receiver pins (R_X) of the other modules. In the designed platform, we modify the Ethernet driver of each module

Algorithm 2 Operations of SDN-based adaptive routing for multi-hop multi-frequency wireless networks

Initialization.
Step 1 - Control modules (M2) of all the devices run a mesh protocol (e.g., WiFi ad-hoc mode for the control channel).
Step 2 - Periodically or driven by events, data modules (M1 and M3) measure link-state information (including the number of neighboring nodes and the RSSI) and forward it to the control module (M2) at each node.
Step 3 - Control module (M2) of all devices reports the received link-state information to the controller.
Step 4 - The controller evaluates the transmission probability of each node by using (6.6) and the cost matrices of data links by using (6.7) – (6.9), and runs the proposed E-Dijkstra's algorithm to generate routes and frequency arrangements.
Step 5 - The controller notifies the involved nodes of the routes and frequency arrangements.
Step 6 - The involved nodes implement the routes and frequency arrangements.

Ethernet interface to facilitate the Ethernet connection between the modules in the ring. The following four rules are written in the driver to instruct the devices on how to process a received packet:

- If the source address of the received frame is the same as the device address, then the device must drop the frame. This case occurs when the device sends a broadcast message, and the broadcast message is received by the device through the ring. In other words, the device receives its broadcast message, which results in a loop.
- If the frame's destination does not match the device address or the broadcast address (FF:FF:FF:FF:FF:FF), then the module forwards the frame to the next module.
- If the destination of the received frame is a broadcast address (FF:FF:FF:FF:FF:FF), then the device must process and forward the frame.
- If the destination address of the received frame is the same as the device address, then the device processes the frame but not forward it.

6.3.2 Proposed SDN-based Adaptive Routing Algorithm

In our proposed architecture, there is a main controller that manages multi-hop routing in the network. The devices are equipped with the designed hardware platform. Each device employs one module as a control module to exchange control information with the controller and uses other existing modules as its data modules to communicate with other devices in the network. The controller uses the same frequency as the control module of the nodes. The controller collects the link-state information of each node from their control module. Algorithm. 2 summarizes the operation of the proposed system.

The control module of each node runs a multi-hop (low-data-rate) mesh protocol to reach the controller. We consider the ESP-MESH protocol, which is the default mesh protocol of the ESP32 modules [225], [226]. The controller does not have to be in the direct transmission range of

Table 6.2: List of notations for network abstraction

Notation	Description
n	Total number of nodes in the network
C	Link capacity (b/s)
B	Bandwidth (Hz)
$RSSI$	Received Signal Strength Indicator (Watt)
N_0	Noise
$\mathbf{C^k_{n\times n}}$	n-by-n link capacity matrix for module$_k$
$P_{i,j}$	Probability of successful transmission of node$_i$ with frequency f_j
d_{ij}	Distance between $node_i$ and $node_j$ (m)
L_{ij}	Link between $node_i$ and $node_j$
$\mathbf{CM^i_{n\times n}}$	n-by-n cost matrix for module$_i$
$E_{Th(i,j)}$	End-to-end throughput between $node_i$ and $node_j$
R	Radio transmission range of a node (m)

Table 6.3: List of notations for Bianchi's model

Notation	Description
t	Slot time
τ	Probability of collision (more than one transmission at the same time)
p	Probability of transmission
CW	Backoff contention window
N	Number of links in the one single-collision domain
P	Probability of successful transmission during a random time slot
m	Maximum backoff stage, $CWmax = 2^m CWmin$

every single node. By running the multi-hop mesh protocol, the controller can deliver routing and frequency arrangement decisions to every node and collect the traffic density information of the nodes (see dashed control links in Figure 6.1). The control module collects the link state information of the data modules via the Ethernet ring and forwards the information to the controller in each device. Consequently, the controller can have a global view of the network and instruct the nodes on their data forwarding functions. Because each data module has its neighbor table (i.e., the list of one-hop neighbors and RSSI from the neighbors), these tables are forwarded to the controller. The controller builds three LSDBs (in the case of using three data modules) based on the information received from each data module. Depending on the metric used by the routing protocol, the controller calculates the cost of each link in the network— for each data module separately— and builds three cost matrices. The controller then finds the least-cost path with alternating frequencies for active flows in the network. To this end, we extend the Dijkstra's algorithm, referred to as E-Dijkstra, to support frequency coloring in our proposed platform.

For each active flow in the network, the controller finds the least-cost path with alternating frequencies and forwards the forwarding information (data module to be used to forward or receive data traffic, the role of the data module (AP or station), and SSID to connect to the AP data module) to the involved nodes in the flow. Next, the control module of the involved nodes forwards the received information to the associated data modules through the ring. Finally, the nodes involved in an end-to-end connection forward the data packets based on the controller instructions.

The routing metric used by the controller and the operation of the E-Dijkstra's algorithm are detailed in Sections 6.3.2.1 and 6.3.2.2, respectively.

6.3.2.1 Proposed Metric for Multi-hop Multi-frequency Routing

This section describes the metric designed for multi-hop multi-frequency routing in the proposed architecture. The designed metric considers the link capacity, frequency coloring, and probability of successful transmission for each node. Any routing protocol can run as an application at the controller and can use the proposed metric to define the paths with the maximum end-to-end capacity. The details of the factors used by the proposed metric and the manner in which the controller determines the least-cost paths in the network are explained in the following paragraphs. Table 6.2 outlines the list of key notations for network abstractions.

As described in Section 6.3, the controller collects the link-state information of each data module (f_1, f_2, and f_3) from each node (i.e., the list of one-hop neighbors and RSSI from the neighbors) through the control channel and builds three link-state matrices for each data module. In our analysis, there is a link between node$_i$ and node$_j$, if the distance between the two nodes (d_{ij}) is shorter than $\min(R_i, R_j)$, where R_i and R_j are the radio transmission ranges of node$_i$ and node$_j$, respectively.

$$L_{ij} = \begin{cases} 0, & if \;\; d_{ij} \geq \min(R_i, R_j) \;\; or \;\; i = j; \\ 1, & if \;\; d_{ij} < \min(R_i, R_j); \end{cases}$$

The measured power level (i.e., RSSI in watts) that node$_i$ receives from node$_j$ is calculated as follows [227, 228]:

$$RSSI_{ij} = \begin{cases} 0, & if \;\; L_{ij} = 0; \\ 10^{-3}(10^{\frac{A - 10\,\eta\, log(d_{ij})}{10}}), & if \;\; L_{ij} = 1; \end{cases}$$

where A is the received signal strength at 1 meter, η is the path-loss exponent, and d_{ij} is the Euclidean distance between node$_i$ and node$_j$ in meters.

After building the link-state matrices, the controller calculates the channel capacity for the links made by each data module (see (6.1)) and builds three capacity matrices, denoted by $\mathbf{C^1}$, $\mathbf{C^2}$, and $\mathbf{C^3}$, for the data modules, f_1, f_2, and f_3, respectively. We use the Shannon–Hartley theorem [229] to calculate the capacity of each link. According to the Shannon–Hartley theorem, the maximum channel capacity (C) of a wireless communication link with a specified bandwidth is calculated as follows:

$$C = B\log_2(1 + \frac{RSSI}{N_0}) \tag{6.1}$$

where B is the channel's bandwidth in Hertz and N_0 is the average noise power.

From (6.1), the channel capacity of a link between node$_i$ and node$_j$, denoted by $C_{(ij)}$, is defined as below:

$$C_{(ij)} = \begin{cases} 0, & if\ \ RSSI_{ij} = 0; \\ B_j \log_2(1 + \frac{RSSI_{ij}}{N_0}), & if\ \ RSSI_{ij} \neq 0; \end{cases}$$

Based on $C_{(ij)}$ and the RSSI matrices, the controller builds the following link capacity matrix for each data module:

$$\mathbf{C^k_{n \times n}} = \begin{bmatrix} 0 & C^k_{12} & \cdots & C^k_{1n} \\ C^k_{21} & 0 & \cdots & C^k_{2n} \\ C^k_{31} & C^k_{32} & \cdots & C^k_{3n} \\ \vdots & \vdots & \ddots & \vdots \\ C^k_{n1} & C^k_{n2} & \cdots & 0 \end{bmatrix} \tag{6.2}$$

where k is the index to the data modules.

In addition to the link capacity, the probability of successful transmission for each node, denoted by P, is another factor that must be considered when calculating the cost of the link. The value of P can be different between the data modules of a node, depending on the traffic density of each data module. To find the value of P, we employ the Bianchi's model [230] proposed for wireless networks that use distributed coordination function (DCF) techniques for contention-based media access control (MAC). In the DCF technique, binary exponential backoff rules are implemented to minimize the probability of transmission collision in wireless communications. When a station has a packet to send, it starts to monitor the channel. If the channel is idle for a Distributed Inter-Frame Space (DIFS) time, the station will transmit. Otherwise, it monitors the channel until it is measured as idle for a DIFS time. Then, the station generates a random backoff interval and waits for this interval before transmitting the packets. The value of the backoff time depends on the number of retransmissions that failed for the packet and is randomly and uniformly chosen in the range $(w - 1)$, where w is the size of the contention window, and its value depends on the physical layer of the devices. We do not delve into the details of the DCF techniques due to space constraints.

In the Bianchi's model, it is assumed that there is a fixed number of nodes in the network, the channel condition is ideal (hidden terminals are ignored), and each node always has a packet available for transmission. Table 6.3 indicates the notations used in the Bianchi's model.

Based on the two-dimensional Markov model presented by the Bianchi's model, the probability of transmission (p), probability of collision (τ), and probability of successful transmission (P) during a random time slot are calculated as below [230]:

$$p = 1 - (1 - \tau)^{N-1} \tag{6.3}$$

$$\tau = \frac{2}{1 + CW_{min} + p\, CW_{min} \sum_{k=0}^{m-1} (2p)^k} \tag{6.4}$$

$$P = \frac{N\tau(1-\tau)^{N-1}}{1-(1-\tau)^N} \tag{6.5}$$

We extend (6.5) to fit our proposed platform, as follows:

$$P_{i,j} = \frac{N_{i,j}\tau(1-\tau)^{N_{i,j}-1}}{1-(1-\tau)^{N_{i,j}}}, \quad i \in \{1, \cdots, n\} \;\&\; j \in \{1, 2, 3\} \tag{6.6}$$

where $P_{i,j}$ is the probability of the successful transmission of node$_i$ with frequency f$_j$, and $N_{i,j}$ indicates the frequency density of data module f$_j$ in node$_i$.

The controller calculates the probability of successful transmission of each data module based on (6.6). The controller then builds three cost matrices based on the values of P and C, referred to as $\mathbf{CM^1}$, $\mathbf{CM^2}$, and $\mathbf{CM^3}$, for f_1, f_2, and f_3 data modules, respectively.

$$\mathbf{CM^1_{n\times n}} = \begin{bmatrix} \infty & \frac{1}{P_{1,1}C^1_{12}} & \cdots & \frac{1}{P_{1,1}C^1_{1n}} \\ \frac{1}{P_{2,1}C^1_{21}} & \infty & \cdots & \frac{1}{P_{2,1}C^1_{2n}} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{1}{P_{n,1}C^1_{n1}} & \frac{1}{P_{n,1}C^1_{n2}} & \cdots & \infty \end{bmatrix} \tag{6.7}$$

$$\mathbf{CM^2_{n\times n}} = \begin{bmatrix} \infty & \frac{1}{P_{1,2}C^2_{12}} & \cdots & \frac{1}{P_{1,2}C^2_{1n}} \\ \frac{1}{P_{2,2}C^2_{21}} & \infty & \cdots & \frac{1}{P_{2,2}C^2_{2n}} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{1}{P_{n,2}C^2_{n1}} & \frac{1}{P_{n,2}C^2_{n2}} & \cdots & \infty \end{bmatrix} \tag{6.8}$$

$$\mathbf{CM^3_{n\times n}} = \begin{bmatrix} \infty & \frac{1}{P_{1,3}C^3_{12}} & \cdots & \frac{1}{P_{1,3}C^3_{1n}} \\ \frac{1}{P_{2,3}C^3_{21}} & \infty & \cdots & \frac{1}{P_{2,3}C^3_{2n}} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{1}{P_{n,3}C^3_{n1}} & \frac{1}{P_{n,3}C^3_{n2}} & \cdots & \infty \end{bmatrix} \tag{6.9}$$

In the $\mathbf{CM^1}$, $\mathbf{CM^2}$, and $\mathbf{CM^3}$ matrices, for link$_{ij}$, $P_{i,1}$, $P_{i,2}$, and $P_{i,3}$ are the probabilities of successful transmission of node$_i$ for the f$_1$, f$_2$, and f$_3$ data modules, respectively. C^1_{ij}, C^2_{ij}, and C^3_{ij} indicates the channel capacities of the link between node$_i$ and node$_j$ for the f$_1$, f$_2$, and f$_3$ frequencies, respectively.

A path is selected to maximize the end-to-end throughput between node$_i$ and node$_j$, denoted by $E_{Th(ij)}$, as given by

$$E_{Th(ij)} = \frac{1}{\sum_{k=1}^{L} \frac{1}{P_k C_k}}, \tag{6.10}$$

where C_k is the capacity of the k^{th} link in the path to the destination, P_k is the probability of successful transmission of the upstream node of the k^{th} link along the path, and L is the number of links between the source and destination nodes.

To find the least-cost path with alternating frequencies between any source and destination, we extend the Dijkstra's algorithm, referred to as the E-Dijkstra's algorithm. In the extended algorithm, three cost metrics are searched in an alternating manner, and for each search, the least-cost link is selected. The details of the E-Dijkstra's algorithm are described in the following section.

6.3.2.2 E-Dijkstra's Algorithm

The Dijkstra's algorithm and the Bellman-Ford algorithm are the known least-cost-path algorithms that find the least-cost paths between all the nodes in a graph. The Dijkstra's algorithm only supports edges with non-negative weights and provides less complexity compared to the Bellman-Ford algorithm. The time complexity of the Dijkstra's algorithm and the Bellman-Ford algorithm is $\mathcal{O}(V + E\log(V))$ and $\mathcal{O}(V.E)$, respectively, where V is the number of nodes, and E is the number of edges in the graph [231]. In this study, we consider the Dijkstra's algorithm since the weights are non-negative for the edges of our cost graph.

The Dijkstra's algorithm finds the least-cost paths between all the nodes in a graph. For a given source node in the graph, the algorithm finds the least-cost paths between that node and every other node. The algorithm takes two values, that is, a cost graph representing the cost of each link in the network and the source node ID. It calculates the least-cost path from the source node to every other node in the network.

We extend the Dijkstra's algorithm to support frequency coloring along with path selection, since we have three cost graphs (one for each data module) in the proposed ESP32-based hardware platform. Algorithm 3 summarizes the proposed E-Dijkstra's algorithm, which runs the Dijkstra's algorithm to recursively evaluate the routing cost of a node and its potential upstream node for each of the frequency channels potentially used between the nodes. Let N denote the number of available channels ($N = 3$ in the ESP32 platform). The key difference between the E-Dijkstra's algorithm and the standard Dijkstra's algorithm is that N costs are evaluated for the pair of nodes for a frequency channel potentially used between the nodes, accounting for all the N possible selections of the incoming frequency channel at the upstream node. Given the N possible selections of the channel from the upstream node to the designated node, N^2 possible incoming routing costs of the designated node need to be recursively updated in parallel in the E-Dijkstra's algorithm. Upon convergence, the frequency channel with the lowest routing cost is selected at the destination. The route and its channel arrangement can be accordingly obtained by backward induction, as done in the Dijkstra's algorithm.

Figure 6.4 depicts an example network consisting of seven nodes. Every node supports three frequencies for the data plane: f_1, f_2, and f_3. The controller collects the control information of all nodes and builds three cost matrices for the data modules. Suppose that the source and destination nodes of a flow are node_2 and node_7. The controller then runs the E-Dijkstra algorithm, which takes four input values, that is, the cost matrix of each data module and source node ID. The algo-

Algorithm 3 The proposed E-Dijkstra's algorithm

Input

s: index of source node, $1 \leq s \leq n$, where n is the total number of nodes
t: index of the destination node, $1 \leq t \leq n$
$C^i \in R^{n \times n}, i = 1, ..., N_{ch}$, where $[C^i]_{j,k}$ is the cost between node j and node k in channel i, N_{ch} is number of available channels (3 in our study)

Output

The least-cost path tree from the source node to every other node in the network, and the corresponding channel assignments.

procedure E-DIJKSTA:
  Let $Q = \{\{(i, v)\}_{i=1}^{N_{ch}}\}_{v=1}^{n}$
  for $v = 1, ..., n$ **do**
    for $i = 1, ..., N_{ch}$ **do**
      if $v == s$ **then**
        $d[i, v] = 0;$
      else
        $d[i, v] = \infty;$
      end if
      $p[i, v] =$ Undefined
    end for
  end for

  while Q is not empty **do**
    Let $(i, u) = \arg\min_{(j,v) \in Q} d[j, v]$
    $Q = Q \backslash (i, u)$
    for each neighbour v of u **do**
      for $j = \{1, ..., N_{ch}\} \backslash i$ **do**
        if $d[i, u] + [C^j]_{u,v} < d[j, v]$ **then**
          $d[j, v] = d[i, u] + [C^j]_{u,v};$
          $p[j, v] = (i, u);$
        end if
      end for
    end for
  end while

  Let $i = \arg\min_{j=1,...,N_{ch}} d[j, t], u = t, path = [\,]$
  while $u \neq s$ **do**
    $path.append((i, u));$
    $(i, u) = p[i, u];$
  end while

  return $path$
end procedure

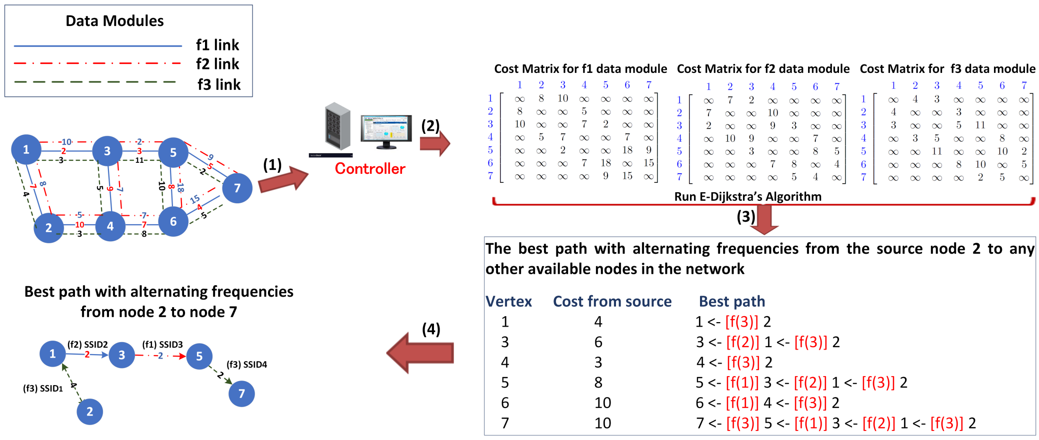


Figure 6.4: Functionality of the proposed SDN-based adaptive routing algorithm.

Table 6.4: Forwarding instructions for flow 2-7

Node ID	Transmitter module	Module operation mode	Downstream SSID	Receiver module	Module operation mode	Upstream SSID
$Node_2$	f_3	AP	SSID1	-	-	-
$Node_1$	f_2	AP	SSID2	f_3	STA	SSID1
$Node_3$	f_1	AP	SSID3	f_2	STA	SSID2
$Node_5$	f_3	AP	SSID4	f_1	STA	SSID3
$Node_7$	-	-	-	f_3	STA	SSID4

rithm considers the possible frequency arrangements. For each possible order of frequencies, the algorithm searches the three matrices until it finds the least-cost path to any other available node in the network. Finally, the lowest-cost path (i.e., with the maximum end-to-end throughput) is selected among all the obtained least-cost paths for each possible receiving frequency of the destination. It is worth mentioning that our proposed platform can support different routing protocols and strategies. The following section describes the proposed routing strategy for this study.

6.3.2.3 Routing Strategy

In our platform, the control module of each node forwards the link state information of the data modules to the controller. Based on the received information, the controller builds the LSDB of the entire network, calculates the capacity of the available links, and creates three cost matrices for each data module. If the controller receives a path request to a specific target, it runs the E-Dijkstra's algorithm on the cost matrices to find a path with alternating frequencies and the maximum end-to-end throughput. Based on the obtained path, the controller instructs the involved nodes of the flow on the frequency that must be used for sending or receiving data traffic, the operation mode of the selected data module (AP or Station), and the SSID is used to connect to the AP data module.

Step 4 in Figure 6.4 indicates the selected data path by the E-Dijkstra's algorithm for flow 2-7. The controller sends the forwarding instructions to the control module of the involved nodes (2, 1, 3, 5 and 7). For nodes that are far from the controller, the control information is sent through multi-hop connections between the control modules of the nodes. The forwarding instructions sent by the controller for flow 2-7 are illustrated in Table 6.4. The information includes the frequency that must be used by the relay nodes to forward and receive data packets, the operation mode of transmitter and receiver data modules, and SSIDs that are defined for upstream and downstream connections.

After receiving the instructions from the controller, the control module forwards the instructions to the data plane modules via the aforementioned Ethernet ring. Subsequently, the module with the AP role starts to send the data. For each hop-to-hop connection in the path to the destination, there are identified pairs with unique SSID. Therefore, for each AP, there is only one station. In this example, the following connections exist:

- $2 -> 1$ SSID = SSID1
- $1 -> 3$ SSID = SSID2
- $3 -> 5$ SSID = SSID3
- $5 -> 7$ SSID = SSID4

When node$_2$ is sending data, the only station listening to f$_3$ frequency with SSID1 is node$_1$. Consequently, the only node that can receive data is node$_1$. Then, node$_1$ forwards the received data through its f$_2$ frequency. The only station listening to f$_2$ frequency with SSID2 is node$_3$, So node$_3$ is the only node that can receive data from node$_1$ and so on. Using this strategy, we can create a multi-hop data flow with minimum overhead.

6.4 MATLAB-based Simulation Study

This section presents an analytical study of the performance of our proposed SDN-based adaptive routing for multi-hop, multi-frequency platforms. To this end, we calculate the cost of each link in the network for each data module based on the metric proposed in Section 6.3.2.1. For this purpose, we first calculate the RSSI for each node to build the capacity matrices. Then, we calculate the probability of successful transmission of each data module. Finally, the cost of each link is calculated based on the obtained C and P values, as explained in the following sections.

6.4.1 Link Capacity

For each node in the network, we calculate the RSSI from the neighbors based on the equation defined in Section 6.3.2.1. Then, based on the obtained RSSIs, the capacity of each link is calculated using (6.1), wherein N_0 is 10^{-6} Watt/Hz and B is 40 MHz.

6.4.2 Probability of the Successful Transmission of the Nodes

To evaluate the probability of the successful transmission of a data module based on (6.6), the value of τ is calculated from (6.4). To calculate τ, the values of the following parameters are given in (6.4):

- The value of p (the probability of transmission) is calculated using the dimensional search in (6.3) and (6.4). The obtained value of p is 0.29.

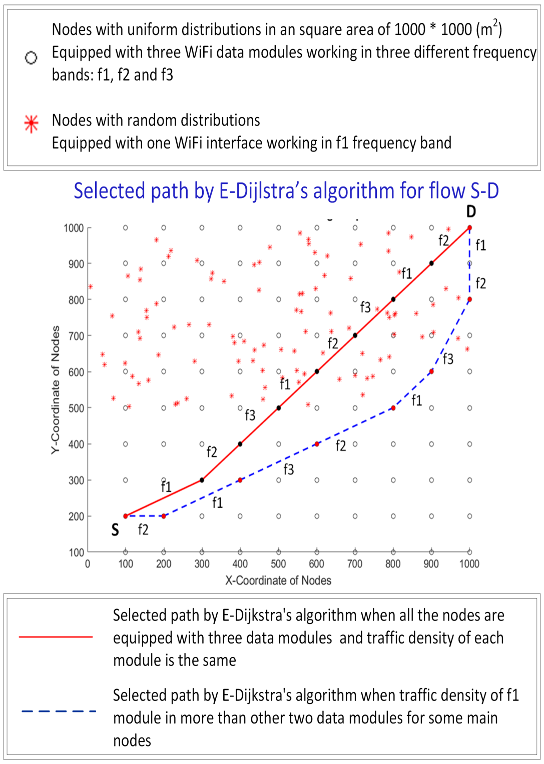


Figure 6.5: MATLAB simulation result of the selected route by the E-Dijkstra's algorithm in the presence of spatially varying interference within a square area of side 1,000 meters.

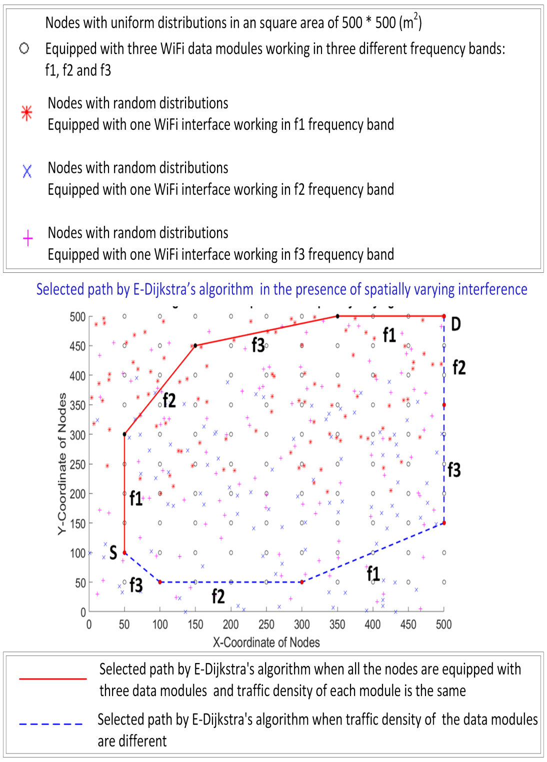


Figure 6.6: MATLAB simulation result of the selected route by the E-Dijkstra's algorithm in the presence of spatially varying interference within a square area of side 500 meters.

- $CWmin$ and $CWmax$ are taken form the IEEE 802.11 standard [232,233], where $CWmin$ and $CWmax$ are set to 32 and 1024, respectively.
- The maximum backoff stage in (6.4), denoted by m, is defined as follows [230]:

$$m = \log_2 \frac{CWmax}{CWmin} \tag{6.11}$$

After calculating τ, we place its value in (6.6) to evaluate the probability of successful transmission of each data module. The number of neighbors for each data module is calculated based on the adjacency matrix of each data module. For our analysis, we make the following assumptions:

- It is assumed that there are two types of nodes in the network. One type, referred to as the main node, is equipped with three interfaces operating at frequencies f_1, f_2, and f_3. The main nodes are uniformly distributed in a square area of side $1,000$ meters and participate in multi-hop routing by receiving the forwarding information from the controller. The second type of nodes, referred to as interference nodes, interfere with the data modules of the main nodes. The interference nodes are randomly distributed and equipped with only one WiFi interface.
- All nodes in the network have the same radio transmission range (250 meters by default).

To demonstrate the performance of the proposed routing algorithm and metric, we define two different scenarios: in the first scenario, there is no interference node in the network. For each main node, the calculated cost matrix for each data module is the same. Consequently, when the controller runs the E-Dijkstra's algorithm and searches for different possible alternating frequencies between the source and destination, the achieved paths for six possible alternating frequencies have the same throughput. This is because each data module (or, in other words, each channel) of a relay node undergoes the same traffic density around the node. The cost matrix is identical among the modules. In the second scenario, we add randomly located interference nodes. By this means, the cost matrix of each data module is different. This is because the inserted interference nodes only work in one frequency and cause interference for the data module operating at the same frequency. Therefore, the frequency density of the affected data module is increased. The controller, based on our designed metric, selects a path with less affected interference. The main nodes and interference nodes in the plot are indicated by the $\circ$ and $*$ markers, respectively. The paths selected by the controller for the first and second scenarios are indicated by solid red and dashed blue lines, respectively. The intermediate nodes between the source and destination in the selected paths are filled in black and red colors for the first and second scenarios, respectively.

Figure 6.5 depicts the selected path by the controller for both defined scenarios when the number of interference nodes in the second scenario is 100 nodes, and the interference nodes are equipped with one WiFi interface working on f_1 frequency. As depicted, in the second scenario, because of the increased traffic density of the f_1 data module in some main nodes, the selected path in the first scenario (solid red line) no longer provides the maximum throughput, and the controller selects another path (dashed blue line) as depicted in Figure 6.5. If the selected path by the first scenario is used for the second scenario with interference nodes, then the cost of the path based on the updated cost matrices ((6.7), (6.8), and (6.9)) would be $3.16 \times 10^{-}7$, while the cost of the selected path by the controller is $1.34 \times 10^{-}7$ (almost three times lower).

Figure 6.6 presents the paths selected by the controller when each data module experiences different traffic densities in the second scenario. In this analysis, the main nodes are distributed uniformly in an area of $500 \times 500\ m^2$, and 300 interference nodes are distributed randomly in the network. Each interference node has only one WiFi interface. One hundred interference nodes operate in f_1 frequency, 100 operate in f_2 frequency, and the rest work in f_3 frequency. As illustrated, the controller selects links with the least interference to direct data traffic between the nodes.

6.5 Performance Analysis using Mininet-WiFi

We implement the platform in Mininet-WiFi emulator to demonstrate the application of the proposed platform in multi-hop multi-frequency wireless mesh networks. This is because performing a real-world experiment with a large number of nodes is not possible due to the lack of sufficient prototype devices.

The algorithms simulated here include:

- The original Dijkstra's algorithm, which is known to be optimum for single frequency/channel routing in the absence of interference and can be readily deployed in the first class of the platforms compared in Table 6.1.

- The extended Dijkstra's algorithm, which is extended from Dijkstra's algorithm and optimum for multi-channel multi-frequency routing in the absence of interference, and can be readily deployed in the second class of the platforms compared in Table 6.1.
- The full proposed E-Dijkstra's algorithm with background interference considered (when evaluating the capacity of each link). The algorithm can be readily applied to the second class of the platforms compared in Table 6.1.

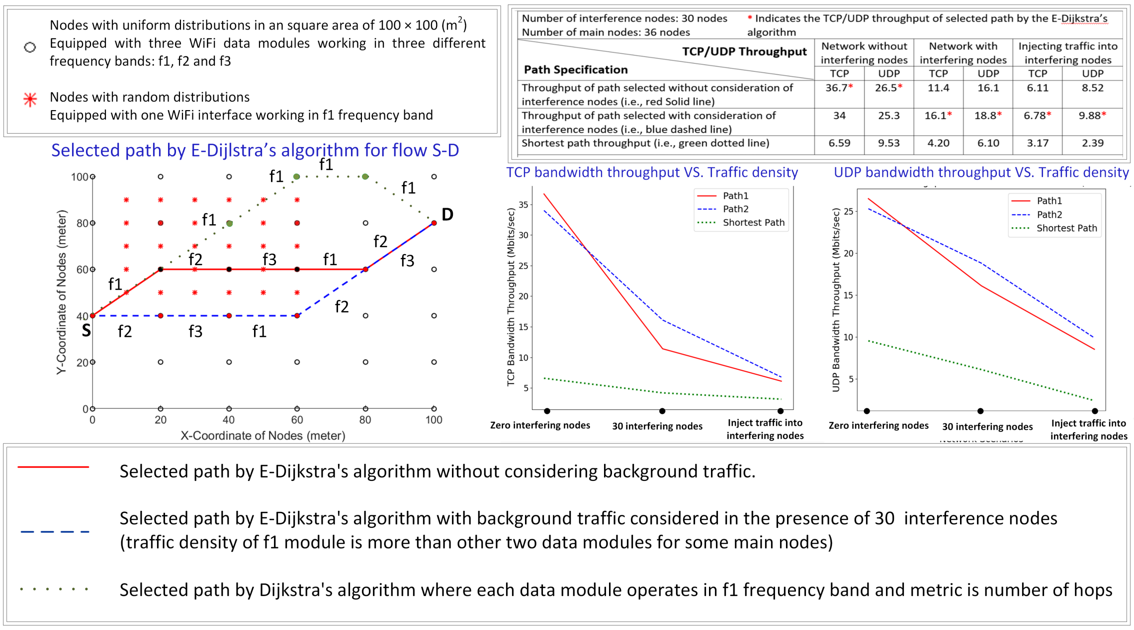
All these algorithms can be deployed in the state-of-the-art platforms summarized in Table 6.1. The Dijkstra's algorithm and its extension without consideration of background interference are expected to provide the best possible routes for single-channel and multi-channel platforms, respectively. By comparing the algorithms, we see that by taking background traffic into account, the proposed E-Dijkstra's algorithm can considerably outperform the Dijkstra's algorithm and its extension without considering background traffic.

Mininet-WiFi [82] is a network emulator that enables the creation of different numbers of virtual wireless stations and access points based on the most common Linux wireless device driver. We implement the designed platform in Mininet-WiFi, and on top of that, we develop an adaptive routing algorithm and investigate its performance in large-scale networks. To verify the results achieved in the previous section, we implement similar scenarios in Mininet-WiFi to demonstrate the achieved throughput.

The network built in Mininet-WiFi is characterized as follows:

- All main nodes ($\circ$) are uniformly distributed and equipped with three wireless interfaces with IEEE 802.11g standard working at different non-overlapping channels ($'1'$, $'6'$ and $'11'$).
- Interference nodes ($*$) are equipped with one IEEE 802.11g standard wireless interface working in channel $'1'$. They run a mesh routing protocol to communicate.
- The transmission range of all the wireless interfaces is 40 meters
- The Log-Distance Propagation Loss Model is selected ($exp = 3$)
- The main nodes employ channels '1', '6', and '11' in an alternating manner in a multi-hop connection.
- A unique SSID is assigned to each hop-to-hop link. For each link, the upstream interface plays an AP role, and the downstream interface plays a station role.
- All the nodes are stationary. The investigation of the proposed protocol under mobility will be conducted in the future.

We run the E-Dijkstra's algorithm to select the best path for a specific source and destination. Then, the wireless interface of the involved main nodes is configured accordingly. TCP and UDP iperf tests are run between the source and destination of a flow to measure the end-to-end throughput. The measurements lasts 240 seconds. A network topology is implemented with 36 main nodes



Number of interference nodes: 30 nodes
Number of main nodes: 36 nodes
* Indicates the TCP/UDP throughput of selected path by the E-Dijkstra's algorithm

Path Specification \ TCP/UDP Throughput	Network without interfering nodes		Network with interfering nodes		Injecting traffic into interfering nodes	
	TCP	UDP	TCP	UDP	TCP	UDP
Throughput of path selected without consideration of interference nodes (i.e., red Solid line)	36.7*	26.5*	11.4	16.1	6.11	8.52
Throughput of path selected with consideration of interference nodes (i.e., blue dashed line)	34	25.3	16.1*	18.8*	6.78*	9.88*
Shortest path throughput (i.e., green dotted line)	6.59	9.53	4.20	6.10	3.17	2.39

Figure 6.7: Mininet-WiFi simulation result when there are 30 interference nodes

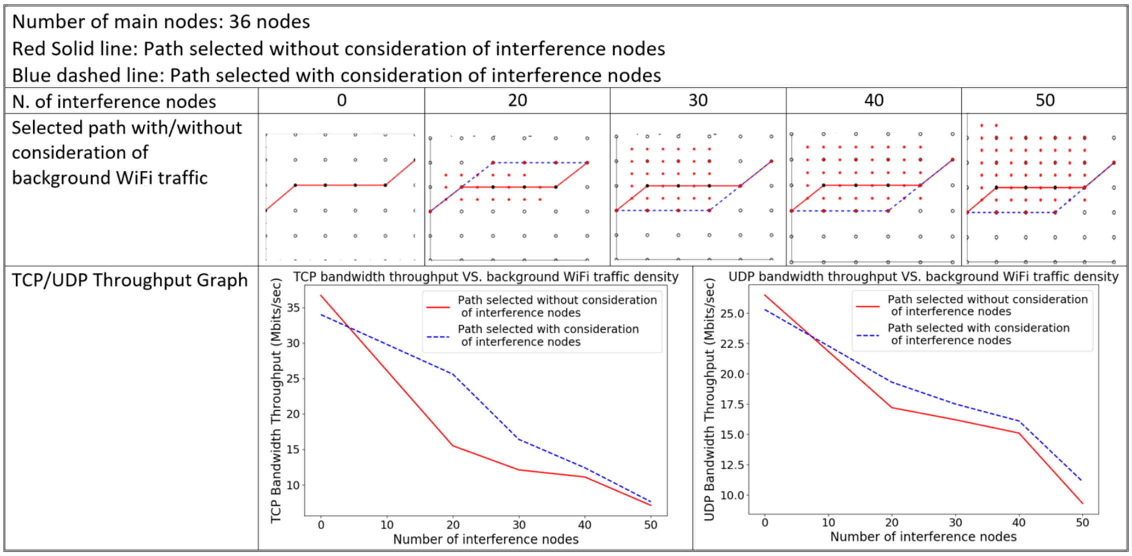


Figure 6.8: TCP/UDP throughput VS. background WiFi traffic density

in the presence and absence of 30 interference nodes, as depicted in Figure 6.7. In the absence of 30 interference nodes, the E-Dijkstra's algorithm selects the solid red line with a throughput of 36.7 Mbps for TCP traffic and 26.5 Mbps for UDP traffic. In the presence of 30 interference nodes, the algorithm chooses a different path (i.e., dashed blue line) with a throughput of 16.1 Mbps for TCP traffic and 18.8 Mbps for UDP traffic. For comparison purposes, we also show the throughput of the path selected without considering the interference nodes (i.e., the above-mentioned red solid path) in the presence of the interference nodes. The throughput is 11.1 Mbps, and 16.1 Mbps for TCP and UDP traffic, respectively, both of which are substantially lower than the throughput of the path selected with consideration of interference nodes (i.e., the above-mentioned blue dashed line).

We also inject more traffic into interference nodes and show that the path selected with the consideration of interference nodes increasingly outperforms the path selected without consideration. We also run the Dijkstra's algorithm to compare its performance with our developed algorithm. The Dijkstra's algorithm selects paths with the minimum number of hops and does not take into account the interference. As it is shown in Figure 6.7, the throughput of the shortest path (dotted green line) is significantly lower, as compared to the selected paths by the E-Dijkstra's algorithm (i.e., throughput of 6.59 Mbps and 9.53 Mbps for TCP and UDP traffic in the absence of 30 interference nodes and throughput of 4.20 Mbps and 6.10 Mbps in the presence of 30 interference nodes).

We further vary the number of interference nodes from 20 to 50, as shown in Figure 6.8. This figure shows that the consideration of background WiFi traffic density is important for the selection of effective paths and frequencies to maximize the end-to-end throughput of the link.

6.6 Conclusion

While multi-hop multi-frequency mesh has been extensively studied in the past decades, only several deployable and relatively bulky systems have been developed to support small numbers of hops under stationary settings. This chapter proposed a new SDN-based adaptive routing framework for multi-hop multi-frequency wireless mesh networks. Specifically, a lightweight hardware platform was developed by CSIRO, which supports multiple non-overlapping WiFi channels. We developed the new E-Dijkstra's algorithm that can adaptively select routing paths and frequency arrangements to maximize the end-to-end throughput. By incorporating background WiFi traffic into the routing cost metrics, the algorithm can find the least-cost path and mitigate the impact of interference. We used Mininet-WiFi to evaluate the performance of the proposed adaptive routing algorithm under various network scenarios. Our results showed significant improvements in terms of end-to-end throughput and hence achieved higher levels of scalability when compared with the shortest-path-based routing strategy.

7

Conclusions and Recommendations for Future Work

A comprehensive literature review was undertaken to identify the challenges and limitations of existing wireless networks (cellular networks in particular) and different solutions to improve the cellular network capacity. Further, 5G networks and their promising technologies, such as device-to-device communications and SDN were reviewed. From the literature, it was concluded that regardless of various SDN-based solutions for cellular networks, little attention has been paid to designing scalable multi-hop protocols for future SDN-based base stations wherein multi-hop routing can be tightly and intelligently controlled and managed by an SDN-based BS.

An SDN-based routing framework, referred to as VARP, was proposed for cellular networks to offload cellular networks. One of the key distinguishing factors between VARP architecture and purely ad hoc routing protocols is the integration of an SDN backbone architecture. In this architecture, the SDN controller is directly connected to the base-stations. Each base-station then transmits and receives control information to the mobile nodes via one or more cellular channels. On the SDN controller side, the control information received from the mobile nodes is used to build a link-state database (LSDB). The SDN controller performs various operations, such as route calculation and topology control. On the UE side, the UE can request different routing information from the SDN controller.

Two routing protocols were designed for multi-hop routing in the proposed framework: VARP-S and SMDRP. The former is a source-based routing protocol, while the latter is a hop-by-hop routing protocols. In VARP-S, the UE requests the path information to transmit data to a specific destination. In the SMDRP, the UE will ask only for the next-hop route information (i.e., each node along the path will make a routing decision). The UE then uses this information to build an MD2D routing architecture over unlicensed WiFi frequency bands. For both protocols, the controller determines whether the out-of-band MD2D routes can successfully meet the requirements of the flows; if they do, then they are used. Otherwise, the controller uses the cellular network to inter-connect the end-to-end nodes. Furthermore, depending on the availability of MD2D routes, the controller may dynamically switch between MD2D and cellular modes, which ensures that traffic offloading does not jeopardize the quality of data packet flows. The benefits of the proposed architecture are numerous: security (authenticated/registered devices), elimination of multi-hop flooding, and the end-user nodes can operate as a simple forwarding device in a more controlled manner than the traditional ad hoc protocols. The performance of the proposed routing protocols was compared with the previously designed SDN-based protocol in [3], known as HSAW. Our results showed that VARP-S and SMDRP provide better performance in terms of the routing overhead, energy, and memory usage of mobile nodes.

A multi-protocol framework was designed to enable the development of multiple routing protocols under a single framework. An SDN-based BS logically divides a cell into a number of clusters based on the collected information from the cell (such as, node density, mobility rate, and traffic density). Then, the SDN-based BS selects the most suitable multi-hop routing protocol for each cluster. Active and passive programming approaches have been proposed to program mobile devices. In the former, the controller determines the forwarding paths to mobile nodes, whereas in the latter, the conventional MANET routing protocols run on the mobile nodes, and the controller provides the information that enables those protocols to self-optimize. In this study, we employed an active approach to program mobile devices. A simulation study was conducted to investigate the performance of the framework. To achieve this, four clusters were developed with various network conditions (such as, node density, mobility rate, and number of flows). Then, the performance of previously designed protocols (HSAW and VARP-S) was investigated for each cluster to determine which protocol operates better in which cluster. Based on the achieved results, a protocol selection methodology based on the AHP MCDM was developed to select the most efficient routing protocol for each cluster. Then, the mobile nodes in each cluster ran the selected protocol to investigate the achieved performance. The simulation results illustrated that the multi-protocol framework increased the total network throughput compared to the single-protocol frameworks.

An SDN-based adaptive routing algorithm was designed for multi-hop multi-frequency wireless networks. To achieve this, an SDN-based hardware platform was developed by CSIRO, where low-power, inexpensive, and lightweight chipsets were used to address the power constraints of the devices participating in multi-hop communications. Furthermore, the control plane functions were separated from the data plane functions using different interconnected WiFi radio modules for each plane. The multiple general-purpose commercial-off-the-shelf (COTS) WiFi modules were modified and repurposed to operate in non-overlapping frequency bands to enable a node to simultaneously receive and forward data traffic from and to other nodes in the network. High-speed inter-module connections were designed on the new platform, where Ethernet interfaces were utilized to connect the WiFi modules in a high-speed ring, up to 40 Mbps (rather than using the typical I^2C or SPI bus which is up to 10 Mbps). New rules were specified to address the connection and deliver data between the modules. In the proposed platform, the control module of each node collects the link-state information of other data modules and forwards that information to the controller. Hence, the controller has a global view of the network and makes forwarding decisions in the network. A novel metric was designed for routing on the platform. The metric considers the link capacity and traffic density of each data module. Dijkstra's algorithm, referred to as E-Dijkstra, was extended to support the proposed platform and to select paths with maximum end-to-end throughput by evaluating the end-to-end routing costs under different possible channel arrangements of the nodes. The performance of the proposed adaptive routing algorithm under dense network settings was investigated using Mininet-WiFi. The simulation results showed that the proposed routing algorithm improves the end-to-end throughput capacity by considering the surrounding WiFi traffic and adaptively selecting routes and channels.

7.1 Future Work

Future research should further develop and confirm the initial findings of this thesis for the proposed frameworks and protocols. To achieve this, a large-scale study for different types of applications (such as IOT) should be conducted to show the benefits of the proposed frameworks and MD2D protocols beyond our study using real-world hardware. Some of the specific future research directions that could be explored includes: 1) the functionality of the proposed protocols should be extended to support out-of-cell, downlink, and uplink communications, 2) parametric optimization using machine-learning techniques to enhance the performance of the protocols is another interesting area to explore, and 3) the performance of the proposed SDN-based adaptive routing algorithm should be investigated for mobility scenarios. Furthermore, the impact of using different types of multi-hop routing protocols on the developed hardware platform can be evaluated.

Bibliography

[1] D. Sajjadi, M. Tanha, and J. Pan, “A comparative study of channel switching latency for conventional and sdn-based routing in multi-hop multi-radio wireless mesh networks,” in *2016 13th IEEE Annual Consumer Communications & Networking Conference (CCNC)*. IEEE, 2016, pp. 330–334. 11, 31, 104, 105

[2] Cisco, “Cisco annual internet report (2018–2023) white paper,” https://www.cisco.com/c/en/us/solutions/collateral/executive-perspectives/annual-internet-report/white-paper-c11-741490.html, 2020. 1

[3] M. Abolhasan, J. Lipman, W. Ni, and B. Hagelstein, “Software-defined wireless networking: centralized, distributed, or hybrid?” *IEEE Network*, vol. 29, no. 4, pp. 32–38, 2015. 3, 5, 38, 64, 80, 85, 88, 124

[4] A. Gawas, “An overview on evolution of mobile wireless communication networks: 1g-6g,” *International Journal on Recent and Innovation Trends in Computing and Communication*, vol. 3, no. 5, pp. 3130–3133, 2015. 8, 9, 11

[5] R. S. Campos, “Evolution of positioning techniques in cellular networks, from 2g to 4g,” *Wireless Communications and Mobile Computing*, vol. 2017, 2017. 9

[6] A. Nordrum, K. Clark *et al.*, “Everything you need to know about 5g,” *IEEE Spectrum*, vol. 27, 2017. 9, 13

[7] P. M. Mariappan, D. R. Raghavan, S. H. A. Aleem, and A. F. Zobaa, “Effects of electromagnetic interference on the functional usage of medical equipment by 2g/3g/4g cellular phones: A review,” *Journal of Advanced Research*, vol. 7, no. 5, pp. 727–738, 2016. 9, 11

[8] K. Guan, B. Ai, B. Peng, D. He, G. Li, J. Yang, Z. Zhong, and T. Kürner, “Towards realistic high-speed train channels at 5g millimeter-wave band—part i: paradigm, significance analysis, and scenario reconstruction,” *IEEE Transactions on Vehicular Technology*, vol. 67, no. 10, pp. 9112–9128, 2018. 9

[9] A. R. Mishra, *Fundamentals of cellular network planning and optimisation: 2G/2.5 G/3G... evolution to 4G*. John Wiley & Sons, 2004. 9, 10

[10] E. Dahlman, S. Parkvall, and J. Skold, *4G: LTE/LTE-advanced for mobile broadband*. Academic press, 2013. 11

[11] A. A. Atayero, M. K. Luka, M. K. Orya, and J. O. Iruemi, "3gpp long term evolution: Architecture, protocols and interfaces," *International Journal of Information and Communication Technology Research*, vol. 1, no. 7, pp. 306–310, 2011. 11, 12

[12] X. J. Li, B.-C. Seet, and P. H. J. Chong, "Multihop cellular networks: Technology and economics," *Computer Networks*, vol. 52, no. 9, pp. 1825–1837, 2008. 12, 14, 15, 19, 24

[13] T. Li, J. Chen, and H. Fu, "Application scenarios based on sdn: An overview," in *Journal of Physics: Conference Series*, vol. 1187, no. 5. IOP Publishing, 2019, p. 052067. 12, 25, 26, 30, 82

[14] A. Bradai, K. Singh, T. Ahmed, and T. Rasheed, "Cellular software defined networking: a framework," *IEEE communications magazine*, vol. 53, no. 6, pp. 36–43, 2015. 12, 32, 37

[15] S. Singh, N. Saxena, A. Roy, and H. Kim, "A survey on 5g network technologies from social perspective," *IETE Technical Review*, vol. 34, no. 1, pp. 30–39, 2017. 13

[16] X. Shen, "Device-to-device communication in 5g cellular networks," *IEEE Network*, vol. 29, no. 2, pp. 2–3, 2015. 13

[17] L. Ferdouse, W. Ejaz, K. Raahemifar, A. Anpalagan, and M. Markandaier, "Interference and throughput aware resource allocation for multi-class d2d in 5g networks," *Iet Communications*, vol. 11, no. 8, pp. 1241–1250, 2017. 13

[18] M. Noura and R. Nordin, "A survey on interference management for device-to-device (d2d) communication and its challenges in 5g networks," *Journal of Network and Computer Applications*, vol. 71, pp. 130–150, 2016. 13, 24

[19] A. Zahmatkesh and T. Kunz, "Software defined multihop wireless networks: Promises and challenges," *Journal of Communications and Networks*, vol. 19, no. 6, pp. 546–554, 2017. 14, 15, 24, 25, 26, 27, 33, 102, 104

[20] R. Ramanathan, J. Redi *et al.*, "A brief overview of ad hoc networks: challenges and directions," *IEEE communications Magazine*, vol. 40, no. 5, pp. 20–22, 2002. 14

[21] M. Abolhasan, T. Wysocki, and E. Dutkiewicz, "A review of routing protocols for mobile ad hoc networks," *Ad hoc networks*, vol. 2, no. 1, pp. 1–22, 2004. 14, 15

[22] M. Abolhasan and T. A. Wysocki, "Dynamic zone topology routing protocol for manets," *European transactions on telecommunications*, vol. 18, no. 4, pp. 351–368, 2007. 14, 15

[23] R. Kaur and M. K. Rai, "A novel review on routing protocols in manets," *Undergraduate Academic Research Journal (UARJ)*, vol. 1, no. 1, pp. 103–108, 2012. 14, 15

[24] A. K. Gupta, H. Sadawarti, and A. K. Verma, "Review of various routing protocols for manets," *International Journal of Information and Electronics Engineering*, vol. 1, no. 3, p. 251, 2011. 14, 15

[25] P. Goyal, V. Parmar, and R. Rishi, "Manet: vulnerabilities, challenges, attacks, application," *IJCEM International Journal of Computational Engineering & Management*, vol. 11, no. 2011, pp. 32–37, 2011. 14

[26] I. Chlamtac, M. Conti, and J. J.-N. Liu, "Mobile ad hoc networking: imperatives and challenges," *Ad hoc networks*, vol. 1, no. 1, pp. 13–64, 2003. 14

[27] E. Alotaibi and B. Mukherjee, "A survey on routing algorithms for wireless ad-hoc and mesh networks," *Computer networks*, vol. 56, no. 2, pp. 940–965, 2012. 14

[28] J. Eriksson, M. Faloutsos, and S. V. Krishnamurthy, "Dart: Dynamic address routing for scalable ad hoc and mesh networks," *IEEE/ACM transactions on Networking*, vol. 15, no. 1, pp. 119–132, 2007. 14

[29] X. Hong, K. Xu, and M. Gerla, "Scalable routing protocols for mobile ad hoc networks," *IEEE network*, vol. 16, no. 4, pp. 11–21, 2002. 14

[30] E. Baccelli and J. Schiller, "Towards scalable manets," in *2008 8th International Conference on ITS Telecommunications*. IEEE, 2008, pp. 133–138. 14

[31] K. Pandey and A. Swaroop, "A comprehensive performance analysis of proactive, reactive and hybrid manets routing protocols," *arXiv preprint arXiv:1112.5703*, 2011. 14

[32] M. A. Rahman, F. Anwar, J. Naeem, and M. S. M. Abedin, "A simulation based performance comparison of routing protocol on mobile ad-hoc network (proactive, reactive and hybrid)," in *International Conference on Computer and Communication Engineering (ICCE'10)*. IEEE, 2010, pp. 1–5. 14

[33] C. E. Perkins, *Ad hoc networking*. Pearson Education India, 2008. 14

[34] M. Keshtgary and V. Babaiyan, "Performance evaluation of reactive, proactive and hybrid routing protocols in manet," *International Journal on Computer Science and Engineering*, vol. 4, no. 2, p. 248, 2012. 14

[35] Y.-C. Tseng, Y.-F. Li, and Y.-C. Chang, "On route lifetime in multihop mobile ad hoc networks," *IEEE Transactions on Mobile computing*, vol. 2, no. 4, pp. 366–376, 2003. 14, 15, 16, 23, 29

[36] C. Perkins, E. Belding-Royer, and S. Das, "Rfc3561: Ad hoc on-demand distance vector (aodv) routing," 2003. 15

[37] D. Johnson, Y.-c. Hu, D. Maltz *et al.*, "The dynamic source routing protocol (dsr) for mobile ad hoc networks for ipv4," RFC 4728, Tech. Rep., 2007. 15

[38] Y.-B. Ko and N. H. Vaidya, "Location-aided routing (lar) in mobile ad hoc networks," *Wireless networks*, vol. 6, no. 4, pp. 307–321, 2000. 15

[39] J. Wang, E. Osagie, P. Thulasiraman, and R. K. Thulasiram, "Hopnet: A hybrid ant colony optimization routing algorithm for mobile ad hoc network," *Ad Hoc Networks*, vol. 7, no. 4, pp. 690–705, 2009. 15

[40] Y.-D. Lin and Y.-C. Hsu, "Multihop cellular: A new architecture for wireless communications," in *Proceedings IEEE INFOCOM 2000. Conference on Computer Communications. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies (Cat. No. 00CH37064)*, vol. 3. IEEE, 2000, pp. 1273–1282. 15, 19, 22

[41] H. ElSawy, E. Hossain, and M.-S. Alouini, "Analytical modeling of mode selection and power control for underlay d2d communication in cellular networks," *IEEE Transactions on Communications*, vol. 62, no. 11, pp. 4147–4161, 2014. 15, 24

[42] A. Dusia, V. K. Mishra, and A. S. Sethi, "Control communication in sdn-based dynamic multi-hop wireless infrastructure-less networks," in *2018 IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS)*. IEEE, 2018, pp. 1–6. 15, 30

[43] D. Cavalcanti, D. Agrawal, C. Cordeiro, B. Xie, and A. Kumar, "Issues in integrating cellular networks wlans, and manets: a futuristic heterogeneous wireless network," *IEEE Wireless Communications*, vol. 12, no. 3, pp. 30–41, 2005. 15, 19, 23, 24, 37

[44] I. F. Akyildiz and X. Wang, "A survey on wireless mesh networks," *IEEE Communications magazine*, vol. 43, no. 9, pp. S23–S30, 2005. 16, 18

[45] I. F. Akyildiz, X. Wang, and W. Wang, "Wireless mesh networks: a survey," *Computer networks*, vol. 47, no. 4, pp. 445–487, 2005. 16, 18

[46] D. Benyamina, A. Hafid, and M. Gendreau, "Wireless mesh networks design—a survey," *IEEE Communications surveys & tutorials*, vol. 14, no. 2, pp. 299–310, 2011. 16

[47] K. Karthika, "Wireless mesh network: A survey," in *2016 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET)*. IEEE, 2016, pp. 1966–1970. 16, 18

[48] Y. Liu, C. Li, R. Venkatesan *et al.*, "Wireless mesh networks: a survey," 2007. 16

[49] G. R. Hiertz, D. Denteneer, S. Max, R. Taori, J. Cardona, L. Berlemann, and B. Walke, "Ieee 802.11 s: the wlan mesh standard," *IEEE Wireless Communications*, vol. 17, no. 1, pp. 104–111, 2010. 16

[50] M. E. M. Campista and M. G. Rubinstein, *Advanced routing protocols for wireless networks*. Wiley Online Library, 2014. 16, 17, 18

[51] M. Eslami, O. Karimi, and T. Khodadadi, "A survey on wireless mesh networks: Architecture, specifications and challenges," in *2014 IEEE 5th Control and System Graduate Research Colloquium*. IEEE, 2014, pp. 219–222. 16, 18

[52] H. A. Mogaibel and M. Othman, "Review of routing protocols and it's metrics for wireless mesh networks," in *2009 International Association of Computer Science and Information Technology-Spring Conference*. IEEE, 2009, pp. 62–70. 16, 17, 18

[53] S. Waharte, R. Boutaba, Y. Iraqi, and B. Ishibashi, "Routing protocols in wireless mesh networks: challenges and design considerations," *Multimedia tools and Applications*, vol. 29, no. 3, pp. 285–303, 2006. 17, 18

[54] M. E. M. Campista, P. M. Esposito, I. M. Moraes, L. H. M. Costa, O. C. M. Duarte, D. G. Passos, C. V. N. De Albuquerque, D. C. M. Saade, and M. G. Rubinstein, "Routing metrics

and protocols for wireless mesh networks," *IEEE network*, vol. 22, no. 1, pp. 6–12, 2008. 17, 18

[55] R. Draves, J. Padhye, and B. Zill, "Routing in multi-radio, multi-hop wireless mesh networks," in *Proceedings of the 10th annual international conference on Mobile computing and networking*, 2004, pp. 114–128. 17, 102

[56] E. Alotaibi and S. Roy, "A location-aware routing metric (alarm) for multi-hop, multichannel wireless mesh networks," pp. 2081–2086, 03 2008. 17

[57] Y. Yang, J. Wang, and R. Kravets, "Designing routing metrics for mesh networks," in *IEEE workshop on wireless mesh networks (WiMesh)*. Santa Clara, CA, 2005, pp. 1–9. 17

[58] A. P. Subramanian, M. M. Buddhikot *et al.*, "Interference aware routing in multi-radio wireless mesh networks," in *2006 2nd IEEE Workshop on Wireless Mesh Networks*. IEEE, 2006, pp. 55–63. 18

[59] N. Nandiraju, D. Nandiraju, L. Santhanam, B. He, J. Wang, and D. P. Agrawal, "Wireless mesh networks: Current challenges and future directions of web-in-the-sky," *IEEE Wireless Communications*, vol. 14, no. 4, pp. 79–89, 2007. 18

[60] X. Wang and A. O. Lim, "Ieee 802.11 s wireless mesh networks: Framework and challenges," *Ad Hoc Networks*, vol. 6, no. 6, pp. 970–984, 2008. 18

[61] G. N. Aggelou and R. Tafazolli, "On the relaying capability of next-generation gsm cellular networks," *IEEE Personal Communications*, vol. 8, no. 1, pp. 40–47, 2001. 19

[62] H. Wu, C. Qiao, S. De, and O. Tonguz, "Integrated cellular and ad hoc relaying systems: icar," *IEEE Journal on Selected areas in Communications*, vol. 19, no. 10, pp. 2105–2115, 2001. 19, 22

[63] H. Luo, R. Ramjee, P. Sinha, L. Li, and S. Lu, "Ucan: a unified cellular and ad-hoc network architecture," in *Proceedings of the 9th annual international conference on Mobile computing and networking*, 2003, pp. 353–367. 19, 20

[64] J. Zhou and Y. Yang, "Parcels: Pervasive ad-hoc relaying for cellular systems," *Proceedings of Med-Hoc-Net, Sardegna, Italy*, 2002. 19

[65] X. Wu, S.-H. Chan, and B. Mukherjee, "Madf: A novel approach to add an ad-hoc overlay on a fixed cellular infrastructure," in *2000 IEEE Wireless Communications and Networking Conference. Conference Record (Cat. No. 00TH8540)*, vol. 2. IEEE, 2000, pp. 549–554. 19, 22

[66] A. N. Zadeh, B. Jabbari, R. Pickholtz, and B. Vojcic, "Self-organizing packet radio ad hoc networks with overlay (soprano)," *IEEE communications magazine*, vol. 40, no. 6, pp. 149–157, 2002. 19, 22

[67] A. Asadi, Q. Wang, and V. Mancuso, "A survey on device-to-device communication in cellular networks," *IEEE Communications Surveys & Tutorials*, vol. 16, no. 4, pp. 1801–1819, 2014. 20, 83

[68] A. Pyattaev, J. Hosek, K. Johnsson, R. Krkos, M. Gerasimenko, P. Masek, A. Ometov, S. Andreev, J. Sedy, V. Novotny *et al.*, "3gpp lte-assisted wi-fi-direct: Trial implementation of live d2d technology," *Etri Journal*, vol. 37, no. 5, pp. 877–887, 2015. 20, 37

[69] X. Lin, J. G. Andrews, A. Ghosh, and R. Ratasuk, "An overview of 3gpp device-to-device proximity services," *IEEE Communications Magazine*, vol. 52, no. 4, pp. 40–48, 2014. 20, 37, 83

[70] H.-y. Wei and R. D. Gitlin, "Two-hop-relay architecture for next-generation wwan/wlan integration," *IEEE Wireless Communications Magazine*, vol. 11, no. 2, pp. 24–30, 2004. 21

[71] R.-S. Chang, W.-Y. Chen, and Y.-F. Wen, "Hybrid wireless network protocols," *IEEE Transactions on Vehicular Technology*, vol. 52, no. 4, pp. 1099–1109, 2003. 21

[72] H.-Y. Hsieh and R. Sivakumar, "Performance comparison of cellular and multi-hop wireless networks: A quantitative study," *ACM Sigmetrics Performance Evaluation Review*, vol. 29, no. 1, pp. 113–122, 2001. 21

[73] K. Streit, N. Rodday, F. Steuber, C. Schmitt, and G. D. Rodosek, "Wireless sdn for highly utilized manets," in *2019 International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*. IEEE, 2019, pp. 226–234. 23, 30

[74] N. Lee, X. Lin, J. G. Andrews, and R. W. Heath, "Power control for d2d underlaid cellular networks: Modeling, algorithms, and analysis," *IEEE Journal on Selected Areas in Communications*, vol. 33, no. 1, pp. 1–13, 2014. 24

[75] G. Zhang, K. Yang, P. Liu, and J. Wei, "Power allocation for full-duplex relaying-based d2d communication underlaying cellular networks," *IEEE Transactions on Vehicular Technology*, vol. 64, no. 10, pp. 4911–4916, 2014. 24

[76] X. Chen, W. Ni, T. Chen, I. B. Collings, X. Wang, R. P. Liu, and G. B. Giannakis, "Multi-timescale online optimization of network function virtualization for service chaining," *IEEE Transactions on Mobile Computing*, vol. 18, no. 12, pp. 2899–2912, 2018. 26

[77] M. B. Al-Somaidai and E. B. Yahya, "Survey of software components to emulate openflow protocol as an sdn implementation," *American Journal of Software Engineering and Applications*, vol. 3, no. 6, pp. 74–82, 2014. 27, 28

[78] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner, "Openflow: enabling innovation in campus networks," *ACM SIGCOMM Computer Communication Review*, vol. 38, no. 2, pp. 69–74, 2008. 27

[79] L. Suomalainen, E. Nikkhouy, A. Y. Ding, and S. Tarkoma, "Open source platforms, applications and tools for software-defined networking and 5g research," University of Helsinki, Department of Computer Science, Tech. Rep., 2014. 28

[80] A. L. Stancu, S. Halunga, A. Vulpe, G. Suciu, O. Fratu, and E. C. Popovici, "A comparison between several software defined networking controllers," in *2015 12th international con-*

ference on telecommunication in modern satellite, cable and broadcasting services (TELSIKS). IEEE, 2015, pp. 223–226. 28

[81] A. M. D. Tello and M. Abolhasan, "Sdn controllers scalability and performance study," in *2019 13th International Conference on Signal Processing and Communication Systems (ICSPCS)*. IEEE, 2019, pp. 1–10. 28

[82] R. R. Fontes, S. Afzal, S. H. Brito, M. A. Santos, and C. E. Rothenberg, "Mininet-wifi: Emulating software-defined wireless networks," in *2015 11th International Conference on Network and Service Management (CNSM)*. IEEE, 2015, pp. 384–389. 28, 120

[83] C. Rotsos, G. Antichi, M. Bruyere, P. Owezarski, and A. W. Moore, "Oflops-turbo: Testing the next-generation openflow switch," in *2015 IEEE International Conference on Communications (ICC)*. IEEE, 2015, pp. 5571–5576. 28

[84] B. Tierney, J. Metzger, J. Boote, E. Boyd, A. Brown, R. Carlson, M. Zekauskas, J. Zurawski, M. Swany, and M. Grigoriev, "perfsonar: Instantiating a global network measurement framework," *SOSP Wksp. Real Overlays and Distrib. Sys*, 2009. 28

[85] M. Canini, D. Venzano, P. Perešíni, D. Kostić, and J. Rexford, "A {NICE} way to test openflow applications," in *Presented as part of the 9th {USENIX} Symposium on Networked Systems Design and Implementation ({NSDI} 12)*, 2012, pp. 127–140. 28

[86] D. Talayconja and R. Lane, Available online: https://floodlight.atlassian.net/wiki/spaces/OFTest/overview, 2015. 28

[87] K. Kaur, J. Singh, and N. S. Ghumman, "Mininet as software defined networking testing platform," in *International Conference on Communication, Computing & Systems (ICCCS)*, 2014, pp. 139–42. 28

[88] S.-Y. Wang, C.-L. Chou, and C.-M. Yang, "Estinet openflow network simulator and emulator," *IEEE Communications Magazine*, vol. 51, no. 9, pp. 110–117, 2013. 28

[89] L. J. Chaves, I. C. Garcia, and E. R. M. Madeira, "Ofswitch13: Enhancing ns-3 with openflow 1.3 support," in *Proceedings of the Workshop on ns-3*, 2016, pp. 33–40. 28

[90] T. Nguyen-Duc and E. Kamioka, "A self-adaptive and energy-efficient wireless multi-hop network," in *2016 22nd Asia-Pacific Conference on Communications (APCC)*. IEEE, 2016, pp. 296–301. 29

[91] C. Giannelli, P. Bellavista, and D. Scotece, "Software defined networking for quality-aware management of multi-hop spontaneous networks," in *2018 International Conference on Computing, Networking and Communications (ICNC)*. IEEE, 2018, pp. 561–566. 29

[92] C. Y. Hans, G. Quer, and R. R. Rao, "Wireless sdn mobile ad hoc network: From theory to practice," in *2017 IEEE International Conference on Communications (ICC)*. IEEE, 2017, pp. 1–7. 29

[93] Y. Peng, L. Guo, Q. Deng, Z. Ning, and L. Zhang, "A novel hybrid routing forwarding algorithm in sdn enabled wireless mesh networks," in *2015 IEEE 17th International Conference*

on High Performance Computing and Communications, 2015 IEEE 7th International Symposium on Cyberspace Safety and Security, and 2015 IEEE 12th International Conference on Embedded Software and Systems. IEEE, 2015, pp. 1806–1811. 30, 104

[94] K. Pentikousis, Y. Wang, and W. Hu, "Mobileflow: Toward software-defined mobile networks," *IEEE Communications magazine*, vol. 51, no. 7, pp. 44–53, 2013. 31, 63, 84

[95] L. E. Li, Z. M. Mao, and J. Rexford, "Toward software-defined cellular networks," in *Software Defined Networking (EWSDN), 2012 European Workshop on*. IEEE, 2012, pp. 7–12. 31, 63

[96] S. Jeon, C. Guimarães, and R. L. Aguiar, "Sdn-based mobile networking for cellular operators," in *Proceedings of the 9th ACM workshop on Mobility in the evolving internet architecture*, 2014, pp. 13–18. 32

[97] X. An, W. Kiess, and D. Perez-Caparros, "Virtualization of cellular network epc gateways based on a scalable sdn architecture," in *2014 IEEE Global Communications Conference*. IEEE, 2014, pp. 2295–2301. 33

[98] M. Usman, A. A. Gebremariam, F. Granelli, and D. Kliazovich, "Software-defined architecture for mobile cloud in device-to-device communication," in *Computer Aided Modelling and Design of Communication Links and Networks (CAMAD), 2015 IEEE 20th Int. Workshop on*. IEEE, 2015, pp. 75–79. 36, 84

[99] V. Yazıcı, U. C. Kozat, and M. O. Sunay, "A new control plane for 5g network architecture with a case study on unified handoff, mobility, and routing management," *IEEE Commun. Mag.*, vol. 52, no. 11, pp. 76–85, 2014. 36

[100] S. Rajalakshmi, N. Deepika, C. Srivardhini, A. Vignesh, and D. Vignesh, "Sdn controller for lte networks," *Int. Journal of Computer Applications*, vol. 133, no. 3, pp. 31–36, 2016. 36

[101] J. Liu, S. Zhang, N. Kato, H. Ujikawa, and K. Suzuki, "Device-to-device communications for enhancing quality of experience in software defined multi-tier lte-a networks," *IEEE Network*, vol. 29, no. 4, pp. 46–52, 2015. 36, 84

[102] D. Kreutz, F. M. Ramos, P. E. Verissimo, C. E. Rothenberg, S. Azodolmolky, and S. Uhlig, "Software-defined networking: A comprehensive survey," *Proc. of the IEEE*, vol. 103, no. 1, pp. 14–76, 2015. 37

[103] W. H. Chin, Z. Fan, and R. Haines, "Emerging technologies and research challenges for 5g wireless networks," *IEEE Wireless Communications*, vol. 21, no. 2, pp. 106–112, 2014. 37

[104] V. Yazıcı, U. C. Kozat, and M. O. Sunay, "A new control plane for 5g network architecture with a case study on unified handoff, mobility, and routing management," *IEEE Commun. Mag.*, vol. 52, 2014. 37

[105] S. Sun, M. Kadoch, L. Gong, and B. Rong, "Integrating network function virtualization with sdr and sdn for 4g/5g networks," *IEEE Network*, vol. 29, no. 3, pp. 54–59, 2015. 37

[106] I. F. Akyildiz, P. Wang, and S.-C. Lin, "Softair: A software defined networking architecture for 5g wireless systems," *Computer Networks*, vol. 85, pp. 1–18, 2015. 37

[107] A. Basta, A. Blenk, K. Hoffmann, H. J. Morper, M. Hoffmann, and W. Kellerer, "Towards a cost optimal design for a 5g mobile core network based on sdn and nfv," *IEEE Transactions on Network and Service Management*, vol. 14, no. 4, pp. 1061–1075, 2017. 37

[108] M. Yang, Y. Li, D. Jin, L. Su, S. Ma, and L. Zeng, "Openran: a software-defined ran architecture via virtualization," in *ACM SIGCOMM computer communication review*, vol. 43, no. 4. ACM, 2013, pp. 549–550. 37

[109] J. Zhang, W. Xie, and F. Yang, "An architecture for 5g mobile network based on sdn and nfv," 2015. 37

[110] K. Nakauchi and Y. Shoji, "Wifi network virtualization to control the connectivity of a target service," *Network and Service Management, IEEE Transactions on*, vol. 12, no. 2, pp. 308–319, 2015. 37

[111] R. Riggio, M. Marina, J. Schulz-Zander, S. Kuklinski, and T. Rasheed, "Programming abstractions for software-defined wireless networks," *Network and Service Management, IEEE Transactions on*, vol. 12, no. 2, pp. 146–162, 2015. 37

[112] K. Doppler, M. Rinne, C. Wijting, C. B. Ribeiro, and K. Hugl, "Device-to-device communication as an underlay to lte-advanced networks," *IEEE Communications Magazine*, vol. 47, no. 12, 2009. 37

[113] M. Condoluci, S. H. Johnson, V. Ayadurai, M. A. Lema, M. A. Cuevas, M. Dohler, and T. Mahmoodi, "Fixed-mobile convergence in the 5g era: From hybrid access to converged core." 37

[114] S. Othmen, F. Zarai, A. Belghith, and L. Kamoun, "Anonymous and secure on-demand routing protocol for multi-hop cellular networks," in *Networks, Computers and Communications (ISNCC), 2016 Int. Symposium on*. IEEE, 2016, pp. 1–6. 37

[115] L. Wei, A. Papathanassiou, Q. C. Li, and G. Wu, "System-level simulations for multi-hop d2d communications overlay lte networks," in *Computing, Networking and Communications (ICNC), 2016 Int. Conference on*. IEEE, 2016, pp. 1–5. 37

[116] L. Militano, A. Orsino, G. Araniti, A. Molinaro, and A. Iera, "A constrained coalition formation game for multihop d2d content uploading," *IEEE Trans. Wireless Commun.*, vol. 15, no. 3, pp. 2012–2024, 2016. 37

[117] A. Asadi, Q. Wang, and V. Mancuso, "A survey on device-to-device communication in cellular networks," *IEEE Communications Surveys & Tutorials*, vol. 16, no. 4, pp. 1801–1819, 2014. 37

[118] X. Lin, J. Andrews, A. Ghosh, and R. Ratasuk, "An overview of 3gpp device-to-device proximity services," *IEEE Communications Magazine*, vol. 52, no. 4, pp. 40–48, 2014. 37

[119] J. Liu, N. Kato, J. Ma, and N. Kadowaki, "Device-to-device communication in lte-advanced networks: A survey," *IEEE Communications Surveys & Tutorials*, vol. 17, no. 4, pp. 1923–1940, 2015. 37

[120] H. Yuan, W. Guo, and S. Wang, "D2d multi-hop routing: Collision probability and routing strategy with limited location information," in *Communication Workshop (ICCW), 2015 IEEE Int. Conference on*. IEEE, 2015, pp. 670–674. 37

[121] F. S. Shaikh and R. Wismüller, "Routing in multi-hop cellular device-to-device networks: A survey," *IEEE Commun. Surveys & Tuts.*, 2018. 37, 83

[122] L. Babun, A. İ. Yürekli, and I. Güvenç, "Multi-hop and d2d communications for extending coverage in public safety scenarios," in *Local Computer Networks Conference Workshops (LCN Workshops), 2015 IEEE 40th*. IEEE, 2015, pp. 912–919. 37

[123] L. Militano, M. Condoluci, G. Araniti, A. Molinaro, A. Iera, and G.-M. Muntean, "Single frequency-based device-to-device-enhanced video delivery for evolved multimedia broadcast and multicast services," *IEEE Transactions on Broadcasting*, vol. 61, no. 2, pp. 263–278, 2015. 37

[124] P. Bellavista, A. Corradi, and C. Giannelli, "Differentiated management strategies for multi-hop multi-path heterogeneous connectivity in mobile environments," *Network and Service Management, IEEE Transactions on*, vol. 8, no. 3, pp. 190 –204, 2011. 37

[125] M. Usman, A. A. Gebremariam, U. Raza, and F. Granelli, "A software-defined device-to-device communication architecture for public safety applications in 5g networks," *IEEE Access*, vol. 3, pp. 1649–1654, 2015. 37

[126] M. Tanha, D. Sajjadi, F. Tong, and J. Pan, "Disaster management and response for modern cellular networks using flow-based multi-hop device-to-device communications," in *Vehicular Technology Conference (VTC-Fall), 2016 IEEE 84th*. IEEE, 2016, pp. 1–7. 37

[127] T. Chen, M. Matinmikko, X. Chen, X. Zhou, and P. Ahokangas, "Software defined mobile networks: concept, survey, and research directions," *IEEE Communications Magazine*, vol. 53, no. 11, pp. 126–133, 2015. 37

[128] M. Abolhasan, J. Lipman, W. Ni, and B. Hagelstein, "Software-defined wireless networking: centralized, distributed, or hybrid?" *IEEE Network*, vol. 29, no. 4, pp. 32–38, 2015. 38, 62

[129] M. N. Tehrani, M. Uysal, and H. Yanikomeroglu, "Device-to-device communication in 5g cellular networks: challenges, solutions, and future directions," *IEEE Commun. Mag.*, vol. 52, no. 5, pp. 86–92, 2014. 41

[130] D. Johnson, Y.-c. Hu, and D. Maltz, "The dynamic source routing protocol (dsr) for mobile ad hoc networks for ipv4," Tech. Rep., 2007. 44

[131] S. R. Dunbar, "The average distance between points in geometric figures," *The College Mathematics Journal*, vol. 28, no. 3, p. 187, 1997. 48

[132] Q.-M. Tran and A. Dadej, “Proactive routing overhead in mobile ad-hoc networks,” in *World of Wireless, Mobile and Multimedia Networks (WoWMoM), 2014 IEEE 15th Int. Symposium on a*, 2014, pp. 1–6. 47, 48, 52, 54

[133] N. Zhou and A. A. Abouzeid, “Information theoretic analysis of proactive routing overhead in mobile ad hoc networks,” *IEEE Transactions on Inform. Theory*, vol. 55, no. 10, pp. 4608–4625, 2009. 47

[134] B. Ishibashi and R. Boutaba, “Topology and mobility considerations in mobile ad hoc networks,” *Ad hoc networks*, pp. 762–776, 2005. 49

[135] J.-C. Cano and P. Manzoni, “A performance comparison of energy consumption for mobile ad hoc network routing protocols,” in *Modeling, Analysis and Simulation of Computer and Telecommunication Systems, 2000. Proc. 8th Int. Symposium on*. IEEE, 2000, pp. 57–64. 61

[136] M. Fotino, A. Gozzi, J.-C. Cano, C. Calafate, F. De Rango, P. Manzoni, and S. Marano, “Evaluating energy consumption of proactive and reactive routing protocols in a manet,” in *Wireless Sensor and Actor Networks*. Springer, 2007, pp. 119–130. 61

[137] A. Kumar, M. Rafiq, and K. Bansal, “Performance evaluation of energy consumption in manet,” *Int. Journal of Computer Apps.*, 2012. 61

[138] I. Poole. (2010) Lte ue category & class definitions with performance specifications for the lte ue categories. [Online]. Available: http://www.radio-electronics.com/info/cellulartelecomms/lte-long-term-evolution/ue-category-categories-classes.php 61

[139] I. Ku, Y. Lu, and M. Gerla, “Software-defined mobile cloud: Architecture, services and use cases,” in *Wireless Communications and Mobile Computing Conference (IWCMC), 2014 Int.* IEEE, 2014, pp. 1–6. 63

[140] M. Kaplan, C. Zheng, M. Monaco, E. Keller, and D. Sicker, “Wasp: a software-defined communication layer for hybrid wireless networks,” in *Proceedings of the tenth ACM/IEEE symposium on Architectures for networking and communications systems*. ACM, 2014, pp. 5–16. 63, 85

[141] M. Uddin, “Toward open and programmable wireless network edge,” Ph.D. dissertation, Old Dominion University, 2016. 65

[142] Q.-M. Tran and A. Dadej, “Proactive routing overhead in mobile ad-hoc networks,” in *World of Wireless, Mobile and Multimedia Networks (WoWMoM), 2014 IEEE 15th International Symposium on a*. IEEE, 2014, pp. 1–6. 69, 70

[143] S. R. Dunbar, “The average distance between points in geometric figures,” *The College Mathematics Journal*, vol. 28, no. 3, p. 187, 1997. 69

[144] T. H. Clausen, J. W. Dean, and C. Dearlove, “Mobile ad hoc network (manet) neighborhood discovery protocol (nhdp),” 2011. 70

[145] A. Ahad, M. Tahir, M. Aman Sheikh, K. I. Ahmed, A. Mughees, and A. Numani, "Technologies trend towards 5g network for smart health-care using iot: A review," *Sensors*, vol. 20, no. 14, p. 4047, 2020. 81

[146] K. Samdanis and T. Taleb, "The road beyond 5g: A vision and insight of the key technologies," *IEEE Network*, vol. 34, no. 2, pp. 135–141, 2020. 81

[147] M. Kumhar and J. Bhatia, "Emerging communication technologies for 5g-enabled internet of things applications," in *Blockchain for 5G-Enabled IoT*. Springer, 2021, pp. 133–158. 81

[148] A. Dogra, R. K. Jha, and S. Jain, "A survey on beyond 5g network with the advent of 6g: Architecture and emerging technologies," *IEEE Access*, vol. 9, pp. 67 512–67 547, 2020. 81

[149] X. You, C.-X. Wang, J. Huang, X. Gao, Z. Zhang, M. Wang, Y. Huang, C. Zhang, Y. Jiang, J. Wang *et al.*, "Towards 6g wireless communication networks: Vision, enabling technologies, and new paradigm shifts," *Science China Information Sciences*, vol. 64, no. 1, pp. 1–74, 2021. 81

[150] S. Manap, K. Dimyati, M. N. Hindia, M. S. A. Talip, and R. Tafazolli, "Survey of radio resource management in 5g heterogeneous networks," *IEEE Access*, vol. 8, pp. 131 202–131 223, 2020. 81

[151] K. Nisar, I. Welch, R. Hassan, A. H. Sodhro, and S. Pirbhulal, "A survey on the architecture, application, and security of software defined networking," *Internet of Things*, p. 100289, 2020. 82

[152] S. H. Haji, S. R. Zeebaree, R. H. Saeed, S. Y. Ameen, H. M. Shukur, N. Omar, M. A. Sadeeq, Z. S. Ageed, I. M. Ibrahim, and H. M. Yasin, "Comparison of software defined networking with traditional networking," *Asian Journal of Research in Computer Science*, pp. 1–18, 2021. 82

[153] S. Faid and M. Moughit, "The benefits of sdn integration on 5g mobile network," in *International Conference on Artificial Intelligence and Symbolic Computation*. Springer, 2019, pp. 266–276. 82

[154] S. K. Tayyaba and M. A. Shah, "5g cellular network integration with sdn: Challenges, issues and beyond," in *2017 International Conference on Communication, Computing and Digital Systems (C-CODE)*. IEEE, 2017, pp. 48–53. 82

[155] M. Hicham, N. Abghour, and M. Ouzzif, "5g mobile networks based on sdn concepts," *International Journal of Engineering and Technology (UAE)*, vol. 7, no. 4, pp. 2231–2235, 2018. 82, 84

[156] A. Abdulghaffar, A. Mahmoud, M. Abu-Amara, and T. Sheltami, "Modeling and evaluation of software defined networking based 5g core network architecture," *IEEE Access*, vol. 9, pp. 10 179–10 198, 2021. 82, 84

[157] H. Fall, O. Zytoune, and M. Yahyai, "to radio resource distribution approach in wireless cellular software defined networking," *International Journal of Advanced Intelligence Paradigms*, vol. 13, no. 3-4, pp. 449–461, 2019. 82, 84

[158] C. N. Tadros, M. R. Rizk, and B. M. Mokhtar, "Software defined network-based management for enhanced 5g network services," *IEEE Access*, vol. 8, pp. 53 997–54 008, 2020. 82, 84

[159] Z. Zhou, J. Gong, Y. He, and Y. Zhang, "Software defined machine-to-machine communication for smart energy management," *IEEE Communications Magazine*, vol. 55, no. 10, pp. 52–60, 2017. 82

[160] H. H. Hussein, H. A. Elsayed, A. El-kader, and M. Sherine, "Intensive benchmarking of d2d communication over 5g cellular networks: prototype, integrated features, challenges, and main applications," *Wireless Networks*, vol. 26, no. 5, pp. 3183–3202, 2020. 83

[161] L. Nadeem, M. A. Azam, Y. Amin, M. A. Al-Ghamdi, K. K. Chai, M. F. N. Khan, and M. A. Khan, "Integration of d2d, network slicing, and mec in 5g cellular networks: Survey and challenges," *IEEE Access*, vol. 9, pp. 37 590–37 612, 2021. 83

[162] B. Manoj, R. Ananthapadmanabha, and C. S. R. Murthy, "Multi-hop cellular networks: Architecture and protocols for best-effort and real-time communication," *Journal of Parallel and Distributed Computing*, vol. 65, no. 6, pp. 767–791, 2005. 83

[163] U. N. Kar and D. K. Sanyal, "An overview of device-to-device communication in cellular networks," *ICT express*, vol. 4, no. 4, pp. 203–208, 2018. 83

[164] S. V. A. Amanuel and S. Y. A. Ameen, "Device-to-device communication for 5g security: A review," *Journal of Information Technology and Informatics*, vol. 1, no. 1, pp. 26–31, 2021. 83

[165] U. N. Kar and D. K. Sanyal, "A critical review of 3gpp standardization of device-to-device communication in cellular networks," *SN Computer Science*, vol. 1, no. 1, pp. 1–18, 2020. 83

[166] P. K. Malik, D. S. Wadhwa, and J. S. Khinda, "A survey of device to device and cooperative communication for the future cellular networks," *International Journal of Wireless Information Networks*, pp. 1–22, 2020. 83

[167] A. H. Alquhali, M. Roslee, M. Y. Alias, and K. S. Mohamed, "D2d communication for spectral efficiency improvement and interference reduction: A survey," *Bulletin of Electrical Engineering and Informatics*, vol. 9, no. 3, pp. 1085–1094, 2020. 83

[168] G. Zhao, S. Chen, L. Qi, L. Zhao, and L. Hanzo, "Mobile-traffic-aware offloading for energy-and spectral-efficient large-scale d2d-enabled cellular networks," *IEEE Transactions on Wireless Communications*, vol. 18, no. 6, pp. 3251–3264, 2019. 83

[169] S. Jayakumar and S. Nandakumar, "A review on resource allocation techniques in d2d communication for 5g and b5g technology," *Peer-to-Peer Networking and Applications*, vol. 14, no. 1, pp. 243–269, 2021. 83

[170] O. Narmanlioglu and E. Zeydan, "Software-defined networking based network virtualization for mobile operators," *Computers & Electrical Engineering*, vol. 57, pp. 134–146, 2017. 84

[171] S. K. Tayyaba and M. A. Shah, "Resource allocation in sdn based 5g cellular networks," *Peer-to-Peer Networking and Applications*, vol. 12, no. 2, pp. 514–538, 2019. 84

[172] G. Yu, R. Liu, Q. Chen, and Z. Tang, "A hierarchical sdn architecture for ultra-dense millimeter-wave cellular networks," *IEEE Communications Magazine*, vol. 56, no. 6, pp. 79–85, 2018. 84

[173] K. Wang, Y. Wang, D. Zeng, and S. Guo, "An sdn-based architecture for next-generation wireless networks," *IEEE Wireless Communications*, vol. 24, no. 1, pp. 25–31, 2017. 84, 104

[174] K. Abbas, M. Afaq, T. Ahmed Khan, A. Rafiq, J. Iqbal, I. Ul Islam, and W.-C. Song, "An efficient sdn-based lte-wifi spectrum aggregation system for heterogeneous 5g networks," *Transactions on Emerging Telecommunications Technologies*, p. e3943, 2020. 84

[175] K. Ouali, M. Kassar, T. M. T. Nguyen, K. Sethom, and B. Kervella, "An efficient d2d handover management scheme for sdn-based 5g networks," in *2020 IEEE 17th Annual Consumer Communications & Networking Conference (CCNC)*. IEEE, 2020, pp. 1–6. 84

[176] H. B. Valiveti and C. Duggineni, "Software defined device to device communication handover-latest advancements," in *2021 6th International Conference on Inventive Computation Technologies (ICICT)*. IEEE, 2021, pp. 1079–1083. 84

[177] I. Elgendi, K. S. Munasinghe, and A. Jamalipour, "A three-tier sdn architecture for densenets," in *2015 9th International Conference on Signal Processing and Communication Systems (ICSPCS)*. IEEE, 2015, pp. 1–7. 84

[178] R. Jayadi and Y.-C. Lai, "Low-overhead multihop device-to-device communications in software defined wireless networks," in *2017 International Conference on Soft Computing, Intelligent System and Information Technology (ICSIIT)*. IEEE, 2017, pp. 144–149. 84

[179] J. Park and W. Yoon, "An architecture of multi-layered sdn based lte/wifi network for multi-interface d2d users," in *2018 International Conference on Information and Communication Technology Convergence (ICTC)*. IEEE, 2018, pp. 1161–1163. 85

[180] Y. Cui, J. Song, K. Ren, M. Li, Z. Li, Q. Ren, and Y. Zhang, "Software defined cooperative offloading for mobile cloudlets," *IEEE/ACM Transactions on Networking*, vol. 25, no. 3, pp. 1746–1760, 2017. 85

[181] A. Muthanna, A. A. Ateya, M. Al Balushi, and R. Kirichek, "D2d enabled communication system structure based on software defined networking for 5g network," in *2018 International Symposium on Consumer Technologies (ISCT)*. IEEE, 2018, pp. 41–44. 85

[182] M. Abolhasan, M. Abdollahi, W. Ni, A. Jamalipour, N. Shariati, and J. Lipman, "A routing framework for offloading traffic from cellular networks to sdn-based multi-hop device-to-device networks," *IEEE Transactions on Network and Service Management*, vol. 15, no. 4, pp. 1516–1531, 2018. 85, 88

[183] M. Abdollahi, M. Abolhasan, N. Shariati, J. Lipman, A. Jamalipour, and W. Ni, "A routing protocol for sdn-based multi-hop d2d communications," in *2019 16th IEEE Annual Consumer Communications & Networking Conference (CCNC)*. IEEE, 2019, pp. 1–4. 85, 88

[184] T. L. Saaty, "Decision making—the analytic hierarchy and network processes (ahp/anp)," *Journal of systems science and systems engineering*, vol. 13, no. 1, pp. 1–35, 2004. 89, 97

[185] H. Taherdoost, "Decision making using the analytic hierarchy process (ahp); a step by step approach," *International Journal of Economics and Management Systems*, vol. 2, 2017. 90

[186] T. Kardi, "Analytic hierarchy process (ahp) tutorial," *Revoledu. com. Available online: https://tinyurl.com/jpu7pms6 (accessed on 21 September 2021)*, 2006. 90

[187] A. Alkhateeb, O. El Ayach, G. Leus, and R. W. Heath, "Channel estimation and hybrid precoding for millimeter wave cellular systems," *IEEE Journal of Selected Topics in Signal Processing*, vol. 8, no. 5, pp. 831–846, 2014. 91

[188] S. Ashtari, F. Tofigh, M. Abolhasan, J. Lipman, and W. Ni, "Efficient cellular base stations sleep mode control using image matching," in *2019 IEEE 89th Vehicular Technology Conference (VTC2019-Spring)*. IEEE, 2019, pp. 1–7. 91

[189] Y. Kanemiyo, Y. Tsukamoto, H. Nakabayashi, and S. Kozono, "Mimo channel model with propagation mechanism and the properties of correlation and eigenvalue in mobile environments," *International Journal of Antennas and Propagation*, vol. 2012, 2012. 91

[190] P. Mishra, C. Gandhi, and B. Singh, "Link quality and energy aware geographical routing in manets using fuzzy logics," *Journal of Telecommunications and Information Technology*, 2016. 92

[191] M. U. Younus, S. W. Kim *et al.*, "Proposition and real-time implementation of an energy-aware routing protocol for a software defined wireless sensor network," *Sensors*, vol. 19, no. 12, p. 2739, 2019. 92

[192] D. Goepel, Klaus, "AHP Priority Calculator," 2019, https://bpmsg.com/ahp/ahp-calc.php, Last accessed on 2021-01-20. 97

[193] S. K. Singh, R. Singh, and B. Kumbhani, "The evolution of radio access network towards open-ran: challenges and opportunities," in *2020 IEEE Wireless Communications and Networking Conference Workshops (WCNCW)*. IEEE, 2020, pp. 1–6. 101

[194] P. H. Masur and J. H. Reed, "Artificial intelligence in open radio access network," *arXiv preprint arXiv:2104.09445*, 2021. 101

[195] L. Bonati, S. D'Oro, M. Polese, S. Basagni, and T. Melodia, "Intelligence and learning in o-ran for data-driven nextg cellular networks," *arXiv preprint arXiv:2012.01263*, 2020. 101

[196] J. Shibalabala and T. G. Swart, "Performance analysis of wireless mesh networks for underground mines," in *2020 International Conference on Artificial Intelligence, Big Data, Computing and Data Communication Systems (icABCD)*. IEEE, 2020, pp. 1–6. 102, 103

[197] D. Sicignano, D. Tardioli, S. Cabrero, and J. L. Villarroel, "Real-time wireless multi-hop protocol in underground voice communication," *Ad Hoc Networks*, vol. 11, no. 4, pp. 1484–1496, 2013. 102

[198] Y. S. Dohare, T. Maity, P. Das, and P. Paul, "Wireless communication and environment monitoring in underground coal mines–review," *IETE technical Review*, vol. 32, no. 2, pp. 140–150, 2015. 102

[199] A. Issa, "Software defined wireless network (sdwn) for industrial environment: case of underground mine," Ph.D. dissertation, Université du Québec en Abitibi-Témiscamingue, 2020. 102

[200] G. A. Kennedy and P. J. Foster, "High resilience networks and microwave propagation in underground mines," in *2006 European Conference on Wireless Technology*. IEEE, 2006, pp. 193–196. 103

[201] M. Moutairou, H. Aniss, and G. Delisle, "Wireless mesh access point routing for efficient communication in underground mine," in *2006 IEEE Antennas and Propagation Society International Symposium*. IEEE, 2006, pp. 577–580. 103

[202] A. Ranjan, H. B. Sahu, and P. Misra, "Wireless sensor networks: An emerging solution for underground mines," *International Journal of Applied Evolutionary Computation (IJAEC)*, vol. 7, no. 4, pp. 1–27, 2016. 103

[203] A. Raniwala and T.-c. Chiueh, "Architecture and algorithms for an ieee 802.11-based multi-channel wireless mesh network," in *Proceedings IEEE 24th Annual Joint Conference of the IEEE Computer and Communications Societies.*, vol. 3. IEEE, 2005, pp. 2223–2234.

[204] J. Nachtigall, A. Zubow, and J.-P. Redlich, "The impact of adjacent channel interference in multi-radio systems using ieee 802.11," in *2008 International Wireless Communications and Mobile Computing Conference*. IEEE, 2008, pp. 874–881.

[205] A. Adya, P. Bahl, J. Padhye, A. Wolman, and L. Zhou, "A multi-radio unification protocol for ieee 802.11 wireless networks," in *First international conference on broadband networks*. IEEE, 2004, pp. 344–354. 104

[206] J. Zhang, Z. Han, B. Liang, and Z. Sun, "Study on the transmission performance of mine wireless mesh network based on ns2," in *2012 International Conference on Industrial Control and Electronics Engineering*. IEEE, 2012, pp. 1602–1604. 104

[207] P. Miklavcic, "On the number of non-overlapping channels in the ieee 802.11 wlans operating in the 2.4 ghz band," *Elektrotehniški vestnik*, vol. 81, pp. 148–152, 2014. 104

[208] R. D. Vallam, A. A. Kanagasabapathy, and C. S. R. Murthy, "A non-cooperative game-theoretic approach to channel assignment in multi-channel multi-radio wireless networks," *Wireless Networks*, vol. 17, no. 2, pp. 411–435, 2011. 104

[209] Y. Peng, Y. Yu, L. Guo, D. Jiang, and Q. Gai, "An efficient joint channel assignment and qos routing protocol for ieee 802.11 multi-radio multi-channel wireless mesh networks," *Journal of Network and Computer Applications*, vol. 36, no. 2, pp. 843–857, 2013. 104

[210] D. Yang, X. Fang, and G. Xue, "Channel allocation in non-cooperative multi-radio multi-channel wireless networks," in *2012 Proceedings IEEE INFOCOM*. IEEE, 2012, pp. 882–890. 104

[211] M. Doraghinejad, H. Nezamabadi-Pour, and A. Mahani, "Channel assignment in multi-radio wireless mesh networks using an improved gravitational search algorithm," *Journal of Network and Computer Applications*, vol. 38, pp. 163–171, 2014. 104

[212] K. N. Ramachandran, E. M. Belding-Royer, K. C. Almeroth, and M. M. Buddhikot, "Interference-aware channel assignment in multi-radio wireless mesh networks." in *Infocom*, vol. 6, 2006, pp. 1–12. 104

[213] P. G. Namboothiri and M. S. Krishna, "Capacity analysis of multi-hop wireless sensor networks using multiple transmission channels: A case study using ieee 802.15. 4 based networks," in *IEEE Local Computer Network Conference*. IEEE, 2010, pp. 168–171. 104

[214] P. Clifford and D. J. Leith, "Channel dependent interference and decentralized colouring," in *International Conference on Network Control and Optimization*. Springer, 2007, pp. 95–104. 104

[215] N. A. Jagadeesan and B. Krishnamachari, "Software-defined networking paradigms in wireless networks: A survey," *ACM Computing Surveys (CSUR)*, vol. 47, no. 2, pp. 1–11, 2014. 104

[216] L. Chen, S. Abdellatif, A. F. S. Tegueu, and T. Gayraud, "Embedding and re-embedding of virtual links in software-defined multi-radio multi-channel multi-hop wireless networks," *Computer Communications*, vol. 145, pp. 161–175, 2019. 104

[217] D. Sajjadi, Z. Zheng, R. Ruby, and J. Pan, "Randomized single-path flow routing on sdn-aware wi-fi mesh networks," in *2018 IEEE 15th International Conference on Mobile Ad Hoc and Sensor Systems (MASS)*. IEEE, 2018, pp. 184–192. 104

[218] M. Seyedebrahimi, F. Bouhafs, A. Raschellà, M. Mackay, and Q. Shi, "Sdn-based channel assignment algorithm for interference management in dense wi-fi networks," in *2016 European conference on networks and communications (EuCNC)*. IEEE, 2016, pp. 128–132. 104

[219] V. A. Siris, G. Stamatakis, and E. Tragos, "A simple end-to-end throughput model for 802.11 multi-radio multi-rate wireless mesh networks," *IEEE Communications Letters*, vol. 15, no. 6, pp. 635–637, 2011. 104

[220] P. Pinyoanuntapong, "Software defined wireless mesh networks: from theory to practice," Ph.D. dissertation, Wichita State University, 2017. 104

[221] K. Venter. (2019) Kinetic mesh networks for underground mining. [Online]. Available: https://rajant.com/kinetic-mesh-networks-for-underground-mining 105, 106

[222] A. PieperPosted. (2017) Moving beyond the conventional wireless network with unifi mesh. [Online]. Available: https://blog.ui.com/2017/01/29/moving-beyond-conventional-wireless-network-unifi-mesh/ 105, 106

[223] M. A. Filho, M. Ribeiro, C. Celes, A. Santos, C. Oliveira, and R. Braga, "Performance analysis of openflow in a multi-interface wmn testbed," in *Proceedings of the Euro American Conference on Telematics and Information Systems*, 2018, pp. 1–8. 104, 105

[224] M. Tanha, D. Sajjadi, and J. Pan, "Demystifying failure recovery for software-defined wireless mesh networks," in *2018 4th IEEE Conference on Network Softwarization and Workshops (NetSoft)*. IEEE, 2018, pp. 488–493. 105

[225] M. LLC. (1999) ESP-MESH. [Online]. Available: https://docs.espressif.com/projects/esp-idf/en/stable/api-guides/mesh.html 109

[226] W. Marwan and T. Wiklom, "Prototype development and performance evaluation of esp-wifi-mesh sensor networks for iot," in *Proceedings of Asia Pacific Conference on Robot IoT System Development and Platform*, no. 2020, 2021, pp. 63–64. 109

[227] F. Shang, W. Su, Q. Wang, H. Gao, and Q. Fu, "A location estimation algorithm based on rssi vector similarity degree," *International Journal of Distributed Sensor Networks*, vol. 10, no. 8, p. 371350, 2014. 111

[228] "dBm to mW Conversion," https://www.rapidtables.com/convert/power/dBm_to_mW.html, Last accessed on 2022-01-03. 111

[229] C. E. Shannon, "A mathematical theory of communication," *Bell system technical journal*, vol. 27, no. 3, pp. 379–423, 1948. 111

[230] G. Bianchi, "Performance analysis of the ieee 802.11 distributed coordination function," *IEEE Journal on selected areas in communications*, vol. 18, no. 3, pp. 535–547, 2000. 112, 118

[231] S. W. AbuSalim, R. Ibrahim, M. Z. Saringat, S. Jamel, and J. A. Wahab, "Comparative analysis between dijkstra and bellman-ford algorithms in shortest path optimization," in *IOP Conference Series: Materials Science and Engineering*, vol. 917, no. 1. IOP Publishing, 2020, p. 012077. 114

[232] A. Symington and P. Kritzinger, "A hardware test bed for measuring ieee 802.11 g distribution coordination function performance," in *2009 IEEE International Symposium on*

Modeling, Analysis & Simulation of Computer and Telecommunication Systems. IEEE, 2009, pp. 1–7. 118

[233] J. Sartthong and S. Sittichivapak, "Backoff algorithm optimization for ieee802. 11 wireless local area networks," in *2012 9th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology*. IEEE, 2012, pp. 1–4. 118