

STABLE ROUTE SELECTION AND NETWORK LOAD  
REDUCTION IN WIRELESS AD HOC NETWORKS

KOK GIN XIAN

THESIS SUBMITTED IN FULFILMENT  
OF THE REQUIREMENTS  
FOR THE DEGREE OF DOCTOR OF PHILOSOPHY

FACULTY OF ENGINEERING  
UNIVERSITY OF MALAYA  
KUALA LUMPUR

2015

**UNIVERSITY OF MALAYA**

**ORIGINAL LITERARY WORK DECLARATION**

Name of Candidate: KOK GIN XIAN (I/C No: 850121-14-5689)  
Registration/Matric No.: KHA100003  
Name of Degree: Doctor of Philosophy (Ph.D.)  
Title of Thesis: Stable Route Selection and Network Load Reduction in  
Wireless Ad Hoc Networks  
Field of Study: Telecommunications

I do solemnly and sincerely declare that:

- (1) I am the sole author/writer of this Work;
- (2) This work is original;
- (3) Any use of any work in which copyright exists was done by way of fair dealing and for permitted purposes and any excerpt or extract front, or reference to or reproduction of any copyright work has been disclosed expressly and sufficiently and the title of the Work and its authorship have been acknowledged in this Work;
- (4) I do not have any actual knowledge nor do I ought reasonably to know that the making of this work constitutes an infringement of any copyright work;
- (5) I hereby assign all and every rights in the copyright to this Work to the University of Malaya ("UM") who henceforth shall be owner of the copyright in this Work and that any reproduction or use in any form or by any means whatsoever is prohibited without the written consent of UM having been first had and obtained;
- (6) I am fully aware that if in the course of making this Work I have infringed any copyright whether intentionally or otherwise I may be subject to legal action or any other action as may be determined by UM.

Candidates Signature

Date:

Subscribed and-solemnly declared before,

Witness's Signature

Date:

Name:

Designation:

## ABSTRACT

A wireless ad hoc network is a dynamic communication network formed by decentralized wireless devices called nodes located, arranged, and moving arbitrarily without the support of a pre-installed infrastructure. The properties of wireless ad hoc networks such as node mobility, multi-hop communication, and self-configuration make them rapidly deployable and flexible. However, the same properties that provide these benefits also result in some issues. For example, node mobility results in frequent network topology changes. Routes that are built over a highly dynamic network are vulnerable to breakage, which leads to interruptions in data transmission and packet loss. To reduce the detrimental effects of node mobility on the network performance, two routing metrics to guide nodes in discovering and establishing stable routes are proposed. The central idea is to select routes consisting of shorter links and/or links formed by low mobility nodes over other routes. A drawback of using link length and node mobility information is that hardware sensors such as global positioning system (GPS) sensors are required. We overcome this inconvenience by proposing a method to estimate the link length between two nodes, and also a method to estimate the relative node mobility of a node with respect to its neighboring nodes. Through our investigation, we found that the proposed routing metrics to be effective in guiding nodes discover and establish routes that are more stable. As a result of using routes that are more stable, higher network performance is achieved. The same property that allows a wireless ad hoc network to span a large area, i.e., multi-hop communication also results in worse network throughput. This is because multi-hop communication results in higher network load as the delivery of a packet from the source to the destination may require several packet transmissions. Network coding is an efficient method that can reduce the network load. We propose an ad hoc routing protocol called Network Coding Routing (NCRT). In NCRT, a new set of coding conditions to find more coding

opportunities was established, and a coding-aware and load-aware routing metric is available to guide source nodes in selecting better paths for sending their packets. Due to reduced network load and improved path selection, NCRT outperforms existing network coding and non-network coding routing protocols. In addition to increasing the network load, the multi-hop structure in wireless ad hoc networks and the decentralized nature of the nodes make broadcasting a complex and inefficient operation. On the one hand, the transmission from a source node may not reach every other node in the network; hence, other nodes need to decide whether to forward the packet when they receive it. On the other hand, if every node forwards the packet that it receives, many redundant transmissions may be resulted in the network. To resolve this problem, we propose an enhanced broadcast protocol for wireless ad hoc networks that reduces the number of redundant transmissions while maintaining packet reachability to all nodes in a network.

## ***ABSTRAK***

Sebuah rangkaian wayarles ad hoc adalah sebuah rangkaian komunikasi dinamik yang dibentuk oleh peranti wayarles yang dikenali sebagai nod yang terletak, tersusun, dan bergerak secara sembarangan tanpa sebarang sokongan daripada infrastruktur yang tersedia ada. Sifat-sifat rangkaian wayarles ad hoc seperti mobiliti nod, komunikasi multi-hop, dan konfigurasi diri menyebabkan mereka cepat untuk dibentuk dan fleksibel. Walau bagaimanapun, sifat-sifat yang sama ini juga mewujudkan beberapa isu. Sebagai contoh, mobiliti nod menyebabkan topologi rangkaian kerap berubah. Laluan yang dibina pada atas rangkaian yang sangat dinamik mudah putus, dan ini menyebabkan gangguan pada penghantaran data dan kehilangan paket. Untuk mengurangkan kesan-kesan buruk mobiliti nod ke atas prestasi rangkaian, dua metrik routing untuk membimbing nod dalam mencari dan membina laluan stabil telah dicadangkan. Idea utama ialah untuk memilih laluan yang terdiri daripada pautan yang lebih pendek dan/atau pautan yang dibentuk oleh nod yang mempunyai mobiliti yang rendah. Satu kelemahan menggunakan maklumat panjang pautan dan mobiliti nod ialah perkakasan seperti sensor sistem kedudukan global (GPS) diperlukan. Kami mengatasi kesulitan ini dengan mencadangkan kaedah untuk menganggar panjang pautan antara dua nod, dan juga kaedah untuk menganggar mobiliti sebuah nod relatif kepada nod berjiranan. Melalui siasatan kami, kami dapati bahawa metrik routing yang dicadangkan berkesan dalam membimbing nod mencari dan membina laluan-laluan yang lebih stabil. Hasil daripada itu, prestasi rangkaian yang lebih baik dapat dicapai. Sifat yang sama yang membolehkan rangkaian wayarles ad hoc untuk meliputi kawasan yang luas, iaitu, komunikasi multi-hop juga menghasilkan kapasiti rangkaian menurun. Ini kerana terdapat beban yang lebih tinggi memandangkan penghantaran paket dari sumber ke destinasi mungkin memerlukan beberapa penghantaran. Pengekoden rangkaian adalah satu kaedah yang berkesan yang boleh mengurangkan beban rangkaian. Kami

mencadangkan protokol routing ad hoc dipanggil Network Coding Routing (NCRT). Dalam NCRT, satu set syarat pengekodan yang baru untuk mencari lebih banyak peluang pengekodan, dan satu metrik routing kod-sedar dan beban-sedar untuk membimbing nod sumber dalam memilih laluan yang lebih baik untuk penghantaran paket mereka telah disediakan. Disebabkan pengurangan beban rangkaian dan pemilihan laluan yang lebih baik, NCRT menghasilkan prestasi rangkaian yang lebih bagus berbanding prestasi protokol routing pengekodan dan bukan pengekodan yang lain. Di samping meningkatkan beban rangkaian, struktur multi-hop dalam rangkaian wayarles ad hoc dan sifat tidak bertumpu nod membuat operasi penyiaran kompleks dan tidak cekap. Pada satu masa, penghantaran dari nod sumber mungkin tidak tercapai ke setiap nod lain dalam rangkaian; oleh itu, nod yang lain perlu membuat keputusan sama ada untuk mengemukakan paket apabila mereka terima. Pada masa yang lain, ramai penghantaran yang tidak diperlukan mungkin terjadi jika setiap nod mengemukakan paket yang mereka terima,. Untuk menyelesaikan masalah ini, kami mencadangkan satu protokol siaran untuk rangkaian wayarles ad hoc yang dapat mengurangkan bilangan penghantaran berlebihan disamping mengekalkan pencapaian paket kepada semua nod dalam sebuah rangkaian.

## ACKNOWLEDGEMENTS

First and foremost, I would like to thank my thesis supervisor, Associate Professor Dr. Chow Chee Onn from the Department of Electrical Engineering, University of Malaya, who has supported me throughout this study with his vast technical knowledge. I started this study from a different field and he offered me to join his lectures to obtain the basics needed for this study. The journey for this study has been a tough one and there were times when I felt demotivated. In those times, Dr. Chow encouraged and motivated me to keep on going. Without him, this study would never have been possible. For this, I am forever indebted to him.

Next, I would like to thank my mother Chong Lai Fong for encouraging me to undertake this study in the first place and also for her financial support. Without her financial support, I cannot imagine how much more difficult this study will be.

My gratitude also goes to the internal and external reviewers of this thesis for their time and effort spent on examining this thesis.

I would also like to thank some close friends that I made during my stay in the university who have brought me lots of joy in these stressful times – Goh Chun Keat, Chang Tong Seng, Ng Kah Mei, Dr. Francios Hoo Chyi Shiang, and Yeo Hock Chai.

Last but not least, I would like to thank two friends from my first degree – Chong Lit Jia and Tang Tzyy Tjinn. I have little friends in the university and they have kept me accompanied by keeping in touch with me and going to lunch together.

# TABLE OF CONTENTS

Abstract .....	iii
<i>Abstrak</i> .....	v
Acknowledgements .....	vii
Table of Contents .....	viii
List of Figures .....	xii
List of Tables.....	xvi
List of Symbols and Abbreviations.....	xvii
1. Chapter 1: Introduction.....	1
1.1 Overview .....	1
1.2 Issues and Challenges in Wireless Ad Hoc Networks .....	8
1.3 Research Motivation.....	13
1.4 Research Aims and Objectives.....	19
1.5 Research Methods .....	20
1.6 Research Contributions .....	22
1.7 Thesis Outline.....	23
2. Chapter 2: Route Stability Metrics for Mobile Ad Hoc Networks.....	25
2.1 Introduction .....	25
2.2 Related Work.....	27
2.3 Route Stability Metrics.....	33
2.3.1 Estimating Link Remaining Lifetime.....	33
2.3.2 Effect of Link Length on the Link Remaining Lifetime .....	35



2.3.3	Route Stability 1 (RS1) Metric .....	37
2.3.4	Route Stability 2 (RS2) Metric .....	39
2.3.5	Estimating Link Lengths .....	40
2.3.6	Estimating Node Mobility .....	45
2.3.7	Packet Header Modification.....	47
2.4	Simulation Studies.....	49
2.4.1	Shorter Links or Links with Higher LRLs? .....	50
2.4.2	Comparing RS1 and RS2 .....	55
2.4.3	Comparing the Sensor-free Methods .....	58
2.4.4	True vs. Estimated Information .....	62
2.4.5	Effect of Threshold Link Length.....	66
2.5	Conclusions .....	70
3.	Chapter 3: Network Coding Routing Protocol for Wireless Ad Hoc Networks.....	71
3.1	Introduction .....	71
3.2	Related Work.....	74
3.2.1	Network Coding and Coding Conditions.....	74
3.2.2	Routing Metrics.....	80
3.3	Network Coding Routing (NCRT) .....	83
3.3.1	Base Routing Protocol .....	83
3.3.2	Coding Conditions .....	84
3.3.3	Route Discovery.....	89
3.3.4	Routing Metric .....	92

3.4	Simulation Studies.....	98
3.4.1	The Illustrative Network .....	99
3.4.2	Chain Networks.....	102
3.4.3	Grid Network .....	108
3.4.4	Random Networks.....	112
3.5	Conclusions .....	118
4.	Chapter 4: Improved Partial Dominant Pruning Broadcast Protocol for Wireless Ad Hoc Networks.....	119
4.1	Introduction .....	119
4.2	Related Work.....	121
4.3	Preliminaries.....	129
4.3.1	Network Model and Assumptions.....	129
4.3.2	Definitions.....	129
4.4	Improved Partial Dominant Pruning (IPDP) .....	130
4.4.1	An Illustrative Example .....	141
4.4.2	Relaxed Assumptions.....	142
4.5	Other Enhancements for IPDP .....	145
4.6	Simulation Studies.....	148
4.6.1	Static Scenario.....	149
4.6.2	Mobile Scenario .....	155
4.7	Conclusions .....	162
5.	Chapter 5: Summary and Future Work.....	163
5.1	Summary .....	163

5.2	Future Work .....	166
	References .....	167
	List of Publications and Papers Presented .....	176

## LIST OF FIGURES

Figure 1.1: A WSN for forest fire monitoring .....	5
Figure 1.2: A mobile ad hoc network.....	5
Figure 1.3: A vehicular ad hoc network.....	7
Figure 1.4: The border effect .....	13
Figure 1.5: Analysis of border node probability using Bernoulli trials .....	14
Figure 1.6: Probability of two nodes having common border nodes .....	15
Figure 1.7: (a) multi-hop route, (b) conflict graph.....	17
Figure 1.8: Two nodes may be located outside the transmission range of each other ....	18
Figure 1.9: Overview of this thesis .....	24
Figure 2.1: Stable zone and caution zone.....	29
Figure 2.2: An illustration to show how various routing metrics are computed.....	30
Figure 2.3: Estimating the remaining lifetime of a link using node location information and node mobility information.....	33
Figure 2.4: Average link lifetime vs. link length .....	36
Figure 2.5: An example illustrating how the RS1 routing metric is used .....	39
Figure 2.6: Estimating the length of the link between two neighboring nodes.....	41
Figure 2.7: Finding the area of the overlapping region of two equal circles .....	42
Figure 2.8: The plot of the relationship between link length and overlap ratio .....	44
Figure 2.9: (a) estimating the length of a link, (b) estimating the distance a node has moved from its previous location.....	46
Figure 2.10: Storing neighbor set information of a node compactly in RREQs and RREPs .....	48
Figure 2.11: Comparing RS1-true, SZ/LRL, and LRL: (a) packet delivery ratio, (b) normalized routing load, (c) average packet latency, (d) average hop count, (e) number of route discoveries .....	55

Figure 2.12: Comparing RS1-true and RS2-true: (a) packet delivery ratio, (b) normalized routing load, (c) average packet latency, (d) average hop count, (e) number of route discoveries .....	58
Figure 2.13: Comparing the sensor-free methods: (a) packet delivery ratio, (b) normalized routing load, (c) average packet latency, (d) average hop count, (e) number of route discoveries .....	62
Figure 2.14: Comparing our proposed sensor-free methods with their corresponding sensor-using methods: (a) packet delivery ratio, (b) normalized routing load, (c) average packet latency, (d) average hop count, (e) number of route discoveries.....	65
Figure 2.15: Investigating the effect of threshold link length: (a) packet delivery ratio, (b) normalized routing load, (c) average packet latency, (d) average hop count, (e) number of route discoveries .....	69
Figure 3.1: The Alice and Bob network.....	72
Figure 3.2: Opportunistic listening allows for packet encoding in the cross topology network.....	73
Figure 3.3: Path selection: (a) coding-oblivious routing, (b) coding-aware routing.....	75
Figure 3.4: Relaxed coding conditions allow for packet encoding beyond two hops.....	76
Figure 3.5: The false-coding effect .....	77
Figure 3.6: The onion-peeling-like approach.....	79
Figure 3.7: The illustrative network.....	87
Figure 3.8: Contents of RREP: (a) node 6, (b) node 2.....	91
Figure 3.9: A simple function to map the number of packets code-able with a packet of the considered path to a value in $[0, 1]$ .....	93
Figure 3.10: A simple function to map the network interface queue utilization of a node to a value in $(0, 1]$ .....	95

Figure 3.11: The illustrative network: (a) number of encodings, (b) total network throughput, (c) average packet delay, (d) routing overhead .....	102
Figure 3.12: Two-hop chain: (a) packet delivery ratio, (b) total network throughput, (c) normalized total network throughput .....	104
Figure 3.13: Three-hop chain: (a) packet delivery ratio, (b) total network throughput, (c) normalized total network throughput .....	106
Figure 3.14: Four-hop chain: (a) packet delivery ratio, (b) total network throughput, (c) normalized total network throughput .....	108
Figure 3.15: Grid network: (a) packet delivery ratio, (b) total network throughput, (c) normalized total network throughput, (d) average packet delay, (e) number of coding structures, (f) number of encodings .....	112
Figure 3.16: Random networks: (a) packet delivery ratio, (b) total network throughput, (c) normalized total network throughput, (d) average packet delay, (e) number of coding structures, (f) number of encodings, (g) average number of hops .....	117
Figure 4.1: Redundant transmission in Self-Pruning .....	124
Figure 4.2: EFNLA proposed using REQ packets to discover neighbors that are not discovered due to link asymmetry.....	126
Figure 4.3: Construction of $U(v)$ in DP .....	132
Figure 4.4: Ideally, $v$ should only include in $U(v)$ nodes within its two-hop region that are not in $N(N(u))$ .....	133
Figure 4.5: Construction of $U(v)$ in PDP .....	134
Figure 4.6: Forwarding node list.....	135
Figure 4.7: (a) an example network, (b) the forwarding node list of the packet transmitted by node 0 .....	136
Figure 4.8: Reducing redundancy in IPDP .....	138
Figure 4.9: An illustrative example.....	142

Figure 4.10: (a) an example network where nodes have irregular transmission range, (b) connectivity graph.....	144
Figure 4.11: Network connected probability .....	150
Figure 4.12: Static scenario: (a) number of transmissions, (b) packet reachability, (c) transmissions to nodes ratio, (d) average number of packets received per node .....	154
Figure 4.13: Sixty nodes: (a) average number of transmissions per broadcast, (b) average packet reachability per broadcast.....	157
Figure 4.14: Ninety nodes: (a) average number of transmissions per broadcast, (b) average packet reachability per broadcast.....	158
Figure 4.15: One hundred and twenty nodes: (a) average number of transmissions per broadcast, (b) average packet reachability per broadcast.....	159
Figure 4.16: One hundred and fifty nodes: (a) average number of transmissions per broadcast, (b) average packet reachability per broadcast.....	160
Figure 4.17: Maximum difference in packet reachability between PDP and the worst performing variant of IPDP.....	161

## LIST OF TABLES

Table 1.1: Types of wireless ad hoc networks and their characteristics .....	7
Table 3.1: Terms and their definitions .....	85
Table 3.2: ns-2 settings .....	98
Table 4.1: Comparison of some two-hop neighbor knowledge broadcast protocols....	128
Table 4.2: Terms and their definitions .....	129
Table 4.3: Comparison between PDP and IPDP .....	142
Table 4.4: Models used in ns-2 .....	148
Table 4.5: Settings used in static and mobile scenarios .....	149
Table 4.6: Settings used in the static scenario.....	150
Table 4.7: Settings used in the mobile scenario.....	155



## LIST OF SYMBOLS AND ABBREVIATIONS

ABR	Associativity-Based Routing
AER	Average Encounter Rate
AHBP	Ad Hoc Broadcast Protocol
AODV	Ad Hoc On-demand Distance Vector
AODV-RRS	AODV-Reliable Route Selection
AOMDV	Ad Hoc On-demand Multipath Distance Vector
CAMP	Coding-Aware Multi-Path
CAOR	Coding Aware Opportunistic Routing
CBR	Constant Bit Rate
CIAR	Coding and Interference Aware Routing
CLARM	Coding- and Load-Aware Routing Metric
CRM	Coding-aware Routing Metric
D2D	Device-to-device
DCAR	Distributed Coding Aware Routing
DCF	Distributed Coordination Function
DP	Dominant Pruning
DSR	Dynamic Source Routing
DSR-wBND	DSR-weighted Bridge Node Density
DV	Distance Vector
ECX	Expected Number of Coded Transmission for an Exchange
EFNLA	Efficient Forward Node List Selection Algorithm
EPDP	Enhanced Partial Dominant Pruning
ETT	Expected Transmission Time

ETX	Expected Transmission Count
FORM	Free-ride Oriented Routing Metric
FORP	Flow Oriented Routing Protocol
GCC	Generalized Coding Conditions
GPS	Global Positioning System
GSC	Greedy Set Cover
HARP	Heading-direction Angles Routing Protocol
HC	Hop Count
ID	Identity
IEEE	Institute of Electrical and Electronics Engineers
IGCC	Improved Generalized Coding Conditions
IoT	Internet of Things
IP	Internet Protocol
IPDP	Improved Partial Dominant Pruning
IR	Interference Ratio
LAR	Location Aided Routing
LDCF	Load-aware Dynamic Counter-based Flooding
LDPF	Load-aware Dynamic Probabilistic Flooding
LET	Link Expiration Time
LRL	Link Remaining Lifetime
LSB-AODV	Link Stability Based-AODV
LSF	Link Stability Factor
MAC	Media Access Control
MANET	Mobile Ad Hoc Network
MCDS	Minimum Connected Dominating Set
MDC	Multiple Description Coding

MF	Mobility Factor
MIC	Metric of Interference and Channel Switching
MIQ	Modified Interface Queue Length
MPR	Multi-Point Relaying
NCAC-MAC	Network Coding Aware Cooperative Medium Access Control
NCR	Neighbor Change Ratio
NCRT	Network Coding Routing
NIC	Network Interface Card
NP	Nondeterministic Polynomial Time
ns-2	network simulator 2
OFFP	Optimized Flooding Protocol
OLSR	Optimized Link State Routing
PC	Personal Computer
PDP	Partial Dominant Pruning
PER	Path Encounter Rate
PMAR	Power and Mobility Aware Routing
PMLAR	Predictive Mobility and Location-Aware Routing
PSF	Path Stability Factor
QLAODV	Q-Learning AODV
QoS	Quality of Service
RDAB	Relative Degree Adaptive Broadcast
RERR	Route Error
RET	Route Expiration Time
RREP	Route Reply
RREQ	Route Request

RS1	Route Stability 1
RS2	Route Stability 2
RSQR	Route Stability based QoS Routing
RSS	Received Signal Strength
RSU	Road Side Unit
RT	Regular Tiling
RVRTUPD	Reverse Route Update
SP	Self-Pruning
ST_OLSR	STable OLSR
SZ	Safety Zone
SZ/LRL	Safety Zone based Route Discovery/Link Remaining Lifetime
TDP	Total Dominant Pruning
UDP	User Datagram Protocol
UMA	Unlicensed Mobile Access
USD	United States Dollar
VANET	Vehicle Ad Hoc Network
WCETT	Weighted Cumulative Expected Transmission Time
WLAN	Wireless Local Area Network
WMN	Wireless Mesh Network
WSN	Wireless Sensor Network

# CHAPTER 1: INTRODUCTION

## 1.1 Overview

The proliferation of mobile devices and services in the recent years spurred an explosive growth in the global mobile data traffic, which is projected to increase nearly ten folds between 2014 and 2019 (Cisco, 2015). In 2014, global mobile data traffic grew 69%. In the same year, 30 exabytes of traffic were communicated, an amount that is nearly 30 times the amount of traffic of the entire Internet in 2000 (Cisco, 2015).

One way to cope with the increasing mobile traffic demand is to offload some traffic to local networks. Major network operators, and network and mobile-terminal manufacturers have proposed Unlicensed Mobile Access (UMA). It was proposed that the unlicensed bands be used to convey cellular services using Wi-Fi and Bluetooth between the mobile station and the core network, for the ultimate goal of offloading the crowded cellular networks and extending the coverage of existing networks (Fitzek, Popovski, & Zorzi, 2005). Device-to-device (D2D) communication among mobile devices that are within transmission range from each other is also being investigated (Doppler, Rinne, Wijting, Ribeiro, & Hug, 2009). It was shown that by allowing D2D communication to underlay a cellular network, the overall throughput in the network may increase significantly compared to when the D2D traffic is relayed by the cellular network. It is predicted that there will be more offload traffic than mobile traffic by 2016 (Cisco, 2015).

Wireless ad hoc networks, which are the natural extension of Wi-Fi networks to multi-hop scenarios, can serve as an excellent platform for mobile traffic offloading. Several efforts in this direction were noted. In (Al-Kanj & Dawy, 2011), it was proposed that mobile terminals interested in obtaining a common content to operate in ad hoc mode. It was reported that this method significantly reduces the number of

cellular channels required to serve a set of mobile terminals. In (Fitzek et al., 2005), an IEEE 802.11-based multi-channel MAC protocol was proposed to allow multi-hop WLANs to better interwork with cellular networks to provide broadband and/or real-time services. In (Do, Hsu, Singh, & Venkatasubramanian, 2011), it was proposed that video be distributed using hybrid cellular and ad hoc networks. Compared to using only the cellular network for video transmission, significant video quality, transmission latency, delivery ratio, and missed frame ratio improvements were reported.

Meanwhile, we are moving away from the Personal Computer (PC)-cum-Internet era to the Internet of Things (IoT) era. Nowadays, more and more commonly used objects are becoming equipped with sensing, processing, and communication capabilities. It is envisioned that objects will become smarter and be connected to the Internet at all times. These objects will bring an evolutionary change in the way we live our lives by providing us with the required information at anytime and anywhere. One of the system-level characteristics of IoT is that smart objects will have the ability to communicate wirelessly among themselves, and form ad hoc networks of interconnected objects (Miorandi, Sicari, De Pellegrini, & Chlamtac, 2012).

Wireless ad hoc networks will become a vital part of future communication systems due to their roles in supporting mobile networks and realizing the IoT vision, which is the next-big-thing in communications. A wireless ad hoc network is a dynamic communication network formed by mobile devices called nodes arranged arbitrarily without an existing communication infrastructure. Nodes are interconnected in a decentralized/peer-to-peer fashion (Bettstetter & Bettstetter, 2004). Every node in a network works as both a host and a router. In addition to transmitting their own packets, they also forward packets for other nodes. Two nodes can communicate with each other without having to be located close to each other as communication between two distant nodes proceeds in a multi-hop fashion.

Offloading traffic from mobile networks and the realization of the IoT vision are data intensive applications. In order to make wireless ad hoc networks better support such applications, the throughput of wireless ad hoc networks must be improved. The throughput of a wireless ad hoc network and the throughput of a flow are known to scale poorly with the number of nodes in the network and hop count, respectively. One factor contributing to the poor performance is the higher network load due to multi-hop communication. In a wireless ad hoc network, the delivery of a packet from the source to the destination often involves relaying from intermediate nodes. In conventional single channel wireless ad hoc networks, packet transmissions are performed on a single channel. Due to multi-hop communication, the single channel has to support more packet transmissions. However, as a channel has limited capacity, higher network load results in a lower effective network throughput. Another cause of poor network performance is route instability. Node mobility is a characteristic of wireless ad hoc networks. In wireless ad hoc networks, when the network topology changes frequently, nodes spend most of their time updating and computing routes in sympathy of nodes movement (Toh, 1997). Much bandwidth is consumed as the routes last for only a short time before they need to be computed or updated again.

Wireless ad hoc networks can be categorized more specifically depending on their applications: wireless sensor networks (WSNs), mobile ad hoc networks (MANETs), wireless mesh networks (WMNs), and vehicular ad hoc networks (VANETs). A WSN is a network formed by sensor nodes. WSNs are usually deployed for sensing and monitoring of physical quantities such as temperature, intensity of light, atmospheric pressure, etc. A sensor node can perform the role of a sensing node, relay node, or base station. Sensing nodes perform the actual sensing in a WSN. They are usually placed at specific locations in a sensing field. They must be placed close to the data sources so that the sensed data is accurate. For example, when sensing the temperature from a heat

source, the sensing node should be placed close to the source; otherwise, the sensed data might be inaccurate. A base station is a node that serves as a gateway for interacting with a WSN. Having a base station in a WSN provides the following benefits: (1) providing a central location for the retrieval of data from the sensing nodes, and (2) makes controlling the sensor nodes and network maintenance easier by providing a centralized control and management interface for the network. Relay nodes relay sensed data from the sensing nodes to the base station. Figure 1.1 shows an example WSN deployed for monitoring forest fire depicting the various types of sensor nodes. Sensor nodes are generally assumed to be static as they are assigned specific tasks. A WSN could contain anywhere from a few hundred to thousands of sensor nodes. Due to the large network size, it is not feasible to recharge or replace the batteries in the individual sensor nodes; hence, the methods developed for WSNs should be energy-efficient. Network maintenance could also prove to be difficult due to the large network size; hence, methods developed for WSNs should also be scalable and fault tolerant. Due to the unique characteristics of WSNs, existing protocols and algorithms designed for wireless ad hoc networks cannot be used directly in WSNs (Aziz, Sekercioglu, Fitzpatrick, & Ivanovich, 2012). A survey on WSNs can be found in (Ian F. Akyildiz, Su, Sankarasubramaniam, & Cayirci, 2002) and (I.F. Akyildiz, Su, Sankarasubramaniam, & Cayirci, 2002).



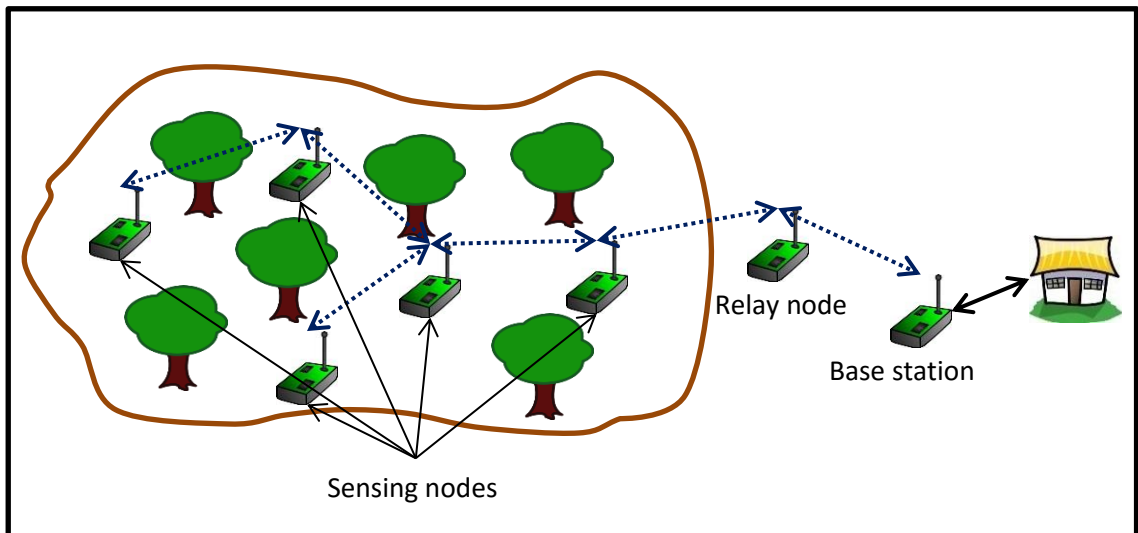


Figure 1.1: A WSN for forest fire monitoring

A mobile ad hoc network (MANET) is a network formed by mobile nodes. Unlike the sensor nodes in a WSN, mobile nodes are not assigned with specific tasks (for example, sensing); therefore, mobile nodes have fewer restrictions in their movement. Node mobility causes the topology of a network to change dynamically. This can cause interruptions to the data transmission in the network and packet loss. Hence, much of the focus in the study of MANETs is in dealing with problems that are associated with node mobility. Figure 1.2 shows an example MANET. A survey on MANETs is found in (Sesay, Yang, & He, 2004).

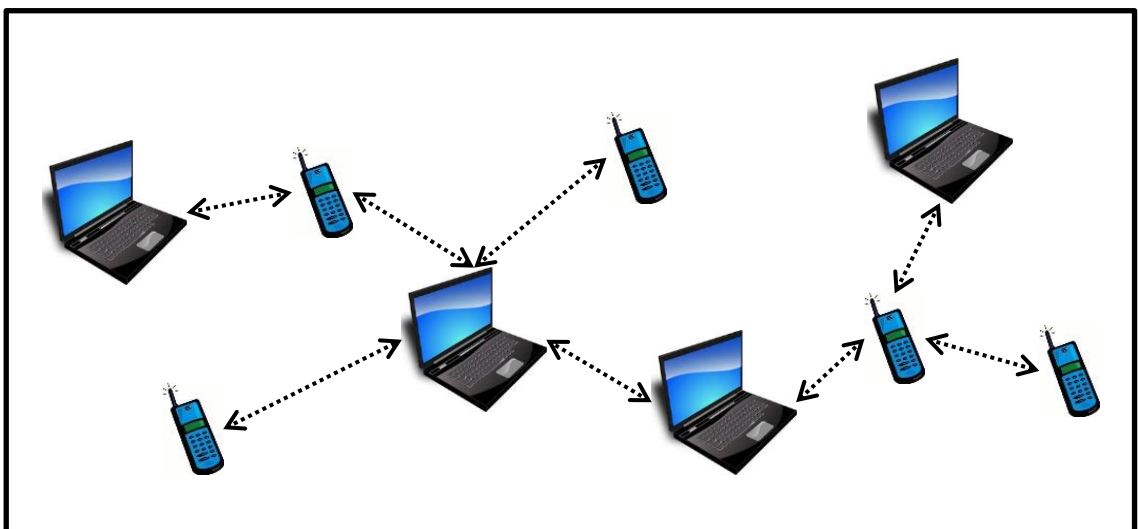


Figure 1.2: A mobile ad hoc network

A wireless mesh network (WMN) consists of mesh routers and mesh clients. Mesh routers are interconnected through wireless links and form the wireless backbone/infrastructure of a WMN. Mesh clients are connected to the mesh routers, or gateways/bridges connected to the mesh routers. Unlike the mobile nodes in a MANET, mesh routers generally have low mobility and no energy constraint as they are usually powered by electrical mains. As mesh routers have no or few constraints, the focus of research in WMNs is usually to improve network performance. A survey on WMNs can be found in (Ian F. Akyildiz & Wang, 2005). Surveys of routing protocols for wireless ad hoc networks and WMNs are available at (Boukerche et al., 2011) and (Alotaibi & Mukherjee, 2012).

Vehicular Ad Hoc Networks (VANETs) is a special type of MANETs that is formed by vehicles and road side units (RSUs) such as lamp posts, traffic lights, and billboards. In VANETs, nodes tend to follow a certain mobility pattern; for example, cars move on roads, stop at traffic lights, and avoid collisions as opposed to moving randomly. Hence, special attention and methods are required for dealing with the problems in VANETs as the methods developed for MANETs may not work very well. In Figure 1.3, the observation that cars queue at a long duration traffic light at a busy road could be used to interconnect two buildings that are located on different sides of the road. A survey of routing protocols for VANETs is available in (Sharef, Alsaqour, & Ismail, 2014).

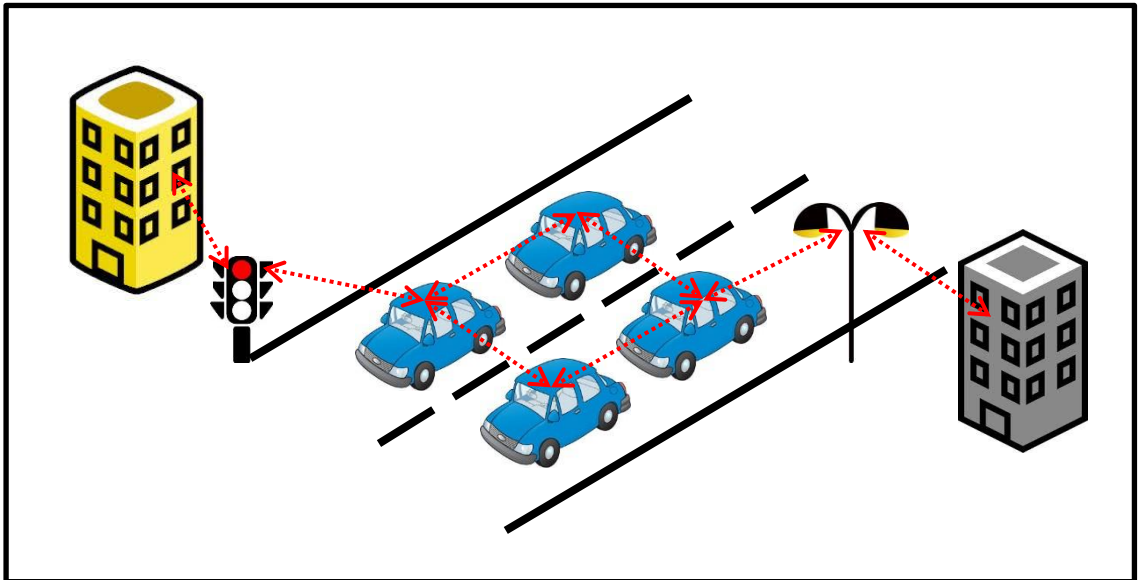


Figure 1.3: A vehicular ad hoc network

A summary of the various types of wireless ad hoc networks is given in Table 1.1.

Table 1.1: Types of wireless ad hoc networks and their characteristics

	Type	Node mobility	Energy constraint	Network size	Mobility pattern	Bandwidth requirement
1	WSN	Static	Yes	Large	N/A	Low
2	WMN	Static (mesh routers)	No	Small to medium	N/A	High
3	MANET	Low to high	Yes or no	Small to medium	Random	Low to high
4	VANET	Low to high	No	Small to medium	Specific	Low

## 1.2 Issues and Challenges in Wireless Ad Hoc Networks

Wireless ad hoc networks have unique characteristics that make them more desirable over other networks for many applications, but these characteristics also raise new issues and challenges:

- *Routing:* Routing is important in any communication network to ensure that packets are delivered to the destinations successfully and efficiently. In wireless ad hoc networks, routing protocols must not only guide nodes discover and route packets through high quality paths, but also deal with the frequent topology changes which makes maintaining the routes difficult and expensive. Hence, we observed an evolution in ad hoc routing protocols from proactive routing protocols (Ibm & Perkins, 1994)(Jacquet et al., 2001) to reactive routing protocols (Perkins & Royer, 1999)(Johnson, Maltz, & Broch, 2001), which are more efficient and produce less routing overhead. Furthermore, it is also extremely challenging if not impossible to optimize routing in wireless ad hoc networks due to their decentralized nature, lack of central administration, and frequent topology changes.
- *Security:* The lack of central servers, specialized hardware, and fixed routers in wireless ad hoc networks precludes the deployment of centralized host relationships. Therefore, any security solution should rely on a distributed scheme instead of a centralized one. The use of wireless links also makes wireless ad hoc networks susceptible to attacks. Unlike in wired networks where an attacker must gain physical access to the network wires, attacks on a wireless ad hoc network can come from all directions and target any node. Also, due to the lack of central administration and infrastructure, packets travel on multi-hop routes and may pass through adversary nodes before arriving at their destinations (Djenouri, Khelladi, & Badache, 2005). The following are some

excellent surveys on security issues in wireless ad hoc networks: (H. Yang, Luo, Ye, Lu, & Zhang, 2004), (Yih-Chun & Perrig, 2004), (Djenouri et al., 2005), and (Kannhavong, Nakayama, Nemoto, Kato, & Jamalipour, 2007).

- *Localization:* Location information is important in wireless ad hoc networks. It is used in a variety of applications such as routing (Ko & Vaidya, 2000)(Karp & Kung, 2000)(Chowdhury & Akyildiz, 2011), broadcasting/flooding (Hur, Le, Jo, & Choo, 2012), environmental monitoring, health care, target tracking, search and rescue, and military surveillance. One way a node can obtain its location is by using a Global Positioning System (GPS) sensor. However, GPS sensors are expensive, consume a lot of energy, and work less effectively in indoor environments. Hence, it is desirable to design localization schemes that can determine node location with high accuracy that are simple to implement and have low node and system requirements and costs.
- *Energy Conservation:* Nodes have a limited supply of energy. A node fails when its energy storage is depleted. When many nodes in a network fail, the network could become partitioned or lose functionality. Furthermore, in the Internet-of-Things (IoT) vision, it is envisioned that smart objects that can self-organize into ad hoc networks will be creating and consuming content through the global Internet infrastructure (Miorandi et al., 2012). In such a large system, it is inconvenient and a great challenge to recharge or replace the batteries in the individual objects. Hence, it is vital to ensure that energy is used wisely.
- *Topology Control:* A WSN could comprise of several thousands of sensor nodes (Santi, 2005). In such a dense or large network, many links exist in the network. Topology control is about keeping only a reduced set of links to improve network performance (reduce interference, energy consumption, etc.). Topology control can be achieved via several methods such as power adjustment, power

mode selection, clustering, or a hybrid combination of these methods (Aziz et al., 2012). Clustering is the partitioning of nodes into smaller groups called clusters. In each cluster, there is a leader and coordinator called the cluster head, and a number of member nodes. Clustering can improve network lifetime and results in a two-tier hierarchy network organization which supports data aggregation (Younis, Krunz, & Ramasubramanian, 2006). However, there are several issues faced in clustering such as ensuring connectivity, determining the optimal frequency of cluster head rotation and cluster size, MAC layer design, node synchronization, and accounting for node duty cycle (Younis et al., 2006).

- *Quality of Service (QoS)*: Providing QoS means to provide some kind of guarantee or assurance by the network regarding the level or grade of service provided to applications (Abbas & Kure, 2010). First, the wireless medium is a shared medium; hence, it is difficult for a node to determine accurately the channel condition as it is determined not only by the activity of the node itself but also the activity of the other nodes sharing the channel. Second, the wireless medium is considerably much less reliable than the wired medium due to various effects such as fading, shadowing, and interference. Third, the topology of a network changes dynamically due to node mobility. Under such circumstances, it is difficult to design a scheme or protocol to provide hard guarantees desired by an application (Abbas & Kure, 2010). Nodes have limited resources; hence, any scheme or protocol designed must also be lightweight (Abbas & Kure, 2010). Examples of QoS parameters/constraints are bandwidth, end-to-end delay, jitter, and packet loss probability (Abbas & Kure, 2010). A routing protocol based on the AODV routing protocol for providing soft QoS is proposed in (Chen & Heinzelman, 2005). The Route Stability based QoS Routing (RSQR) protocol, which is also based on AODV, quantifies link

stability using a rule-based method with the received signal strengths (RSSs) of packets. In RSQR, the destination node selects the most stable path that passes admission control. A routing protocol for multi-interface multi-channel ad hoc networks based on the OLSRv2 routing protocol is proposed in (Kajioka et al., 2011).

- *Dynamic Network Topology:* Due to node mobility, the topology of a network changes frequently. Routes that are constructed over a highly dynamic network are vulnerable to breakage. When a route is broken, data transmission for the source-destination node pair is suspended (Y. Wang, Zhou, Yu, Wang, & Du, 2012), and the network reconfigures itself according to the routing protocol by performing self-healing operations such as route repair or new route discovery process (Perkins & Royer, 1999)(Johnson et al., 2001). These procedures involve the communication of control packets such as Route Error (RERR), Route Request (RREQ), and Route Reply (RREP) among the nodes which further robs the network of its precious bandwidth. As a result, when routes break often, routing overhead increases and the network performance deteriorates.
- *Limited Bandwidth:* Wireless multi-hop networks are typically characterized by a limited bandwidth available to the nodes (Santi, 2005). Studies have shown that the network throughput and end-to-end throughput of individual flows decrease rapidly as node density or number of hops increases (Gupta & Kumar, 2000)(Xu & Saadawi, 2001)(Jain, Padhye, Padmanabhan, & Qiu, 2005)(Ng & Liew, 2007).
- *Transmission Redundancy in Broadcast:* Broadcasting refers to sending a packet from a source node to every other node in the network. It is a vital fundamental operation in communication networks. In wireless ad hoc networks, it is used in

on-demand ad hoc routing protocols for route discovery (Perkins & Royer, 1999)(Johnson et al., 2001)(Marina & Das, 2006), for providing reliable multicast communication in highly dynamic ad hoc networks, disseminating warning messages (for example, in vehicular ad hoc networks (Zhao, Zhang, & Cao, 2007)(Yu & Heijenk, 2008)), and advertising and requesting services by service providers and requestors, respectively (Ververidis & Polyzos, 2008). It is desirable to use a minimum number of packet transmissions to complete a broadcast operation. To minimize the number of transmissions required for a broadcast while ensuring packet reachability to all nodes in a network is similar to finding the optimal solution for the minimum connected dominating set (MCDS) problem, which is an NP-complete problem (H Lim & Kim, 2001). Hence, it is difficult to devise a transmission plan that minimizes the number of transmission required while still ensuring that a packet reaches every node in a network, especially when there is no central administration in wireless ad hoc networks (Chlamtac, Conti, & Liu, 2003) and nodes have a limited view of the network topology.



### 1.3 Research Motivation

One of the characteristics of wireless ad hoc networks is node mobility. Due to node mobility, the topology of a network changes rapidly. In wireless ad hoc networks, packets are sent on multi-hop routes. Routes that are constructed over a highly dynamic network are prone to breakage. This is especially true when the hop count routing metric is used as it is known to cause the border (Jianzhen Sun, Yuan'an Liu, Hefei Hu, & Dongming Yuan, 2010)/edge (Yoon, 2002) effect, as illustrated in Figure 1.4. As node density increases, the probability of finding a node that is located close to the border of the transmission coverage areas of two nodes increases, as illustrated in Figure 1.4b. When the hop count routing metric is used, links are formed over these border nodes. These links are long and highly vulnerable to breakage. A slight movement from the nodes could cause the links to break.

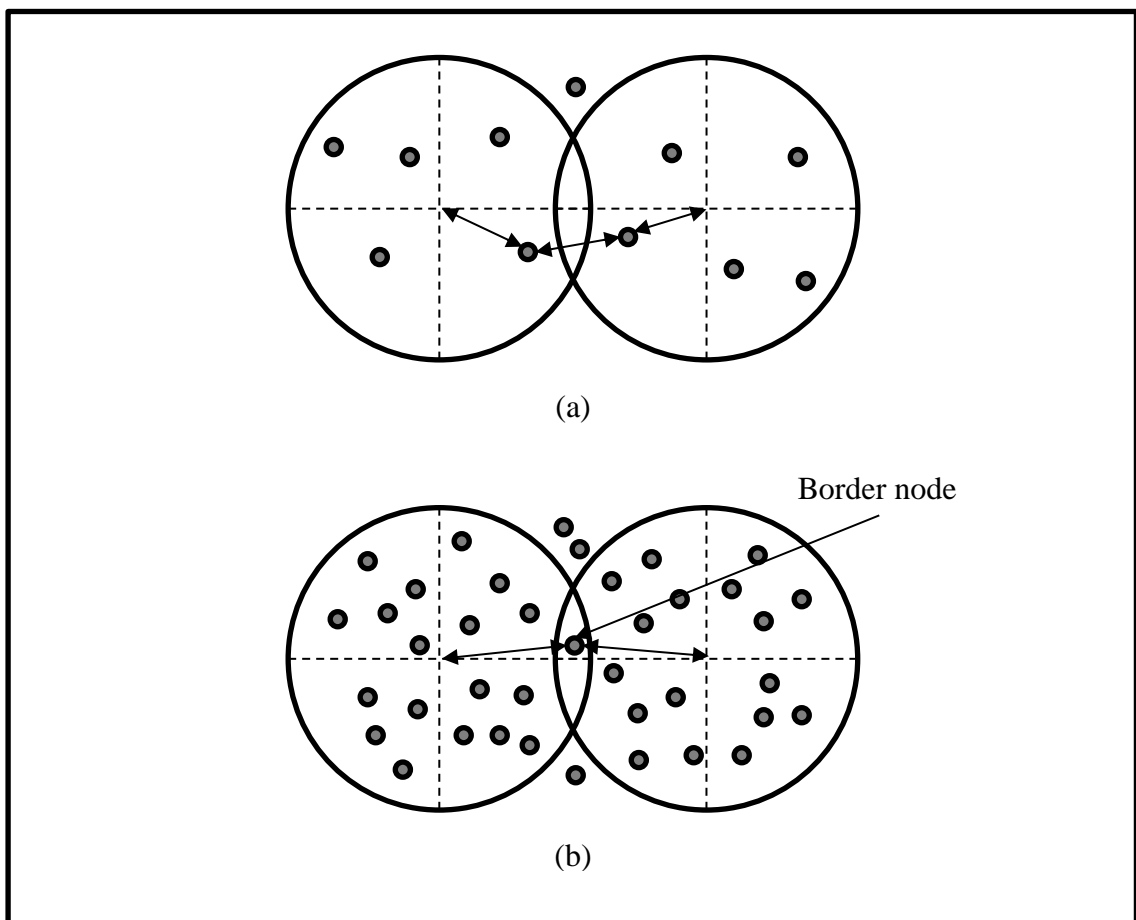


Figure 1.4: The border effect

We analyze the probability of two nodes having at least one common border node. For two nodes A and B that are separated by a distance  $d \leq 2R$ , where  $R$  is the node transmission range, which we assume to be common for all nodes in a network for ease of analysis, the area of the intersection of the transmission coverage areas of nodes A and B,  $A_{overlap}$ , is given as follows:

$$A_{overlap} = 2R^2 \cos^{-1} \frac{d}{2R} - \frac{d}{2} \sqrt{4R^2 - d^2} \quad \text{Eq. 1.1}$$

We assume a rectangular region encapsulating the intersection of the transmission coverage areas of the two nodes, with width  $d$  and height  $2R$ , as shown in Fig. 1.5. We can obtain the probability of having a node in the overlap region using Bernoulli trials. If we randomly deploy (“throw”) a node in the rectangular region, the probability of the node landing in the overlap region of the transmission coverage areas of nodes A and B,  $p$ , can be computed as follows:

$$p = \frac{2R^2 \cos^{-1} \frac{d}{2R} - \frac{d}{2} \sqrt{4R^2 - d^2}}{2Rd} \quad \text{Eq. 1.2}$$

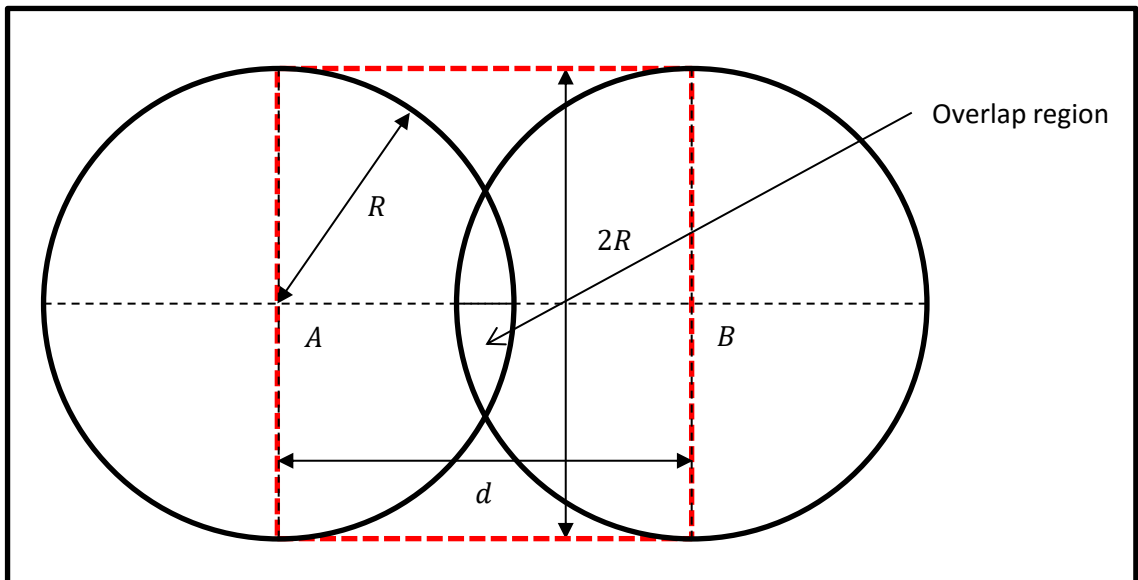


Figure 1.5: Analysis of border node probability using Bernoulli trials

Let  $X$  be the number of nodes in the overlap region when we randomly deploy  $n$  nodes in the rectangular region. The probability of having  $X = k$  nodes in the overlap region

when we randomly deploy  $n$  nodes in the rectangular region can be computed using the Binomial probability function, as follows:

$$P_r(X = k) = \binom{n}{k} p^k (1 - p)^{n-k} \quad \text{Eq. 1.3}$$

The probability of having no node in the overlap region can be computed as follows:

$$P_r(X = 0) = \binom{n}{0} p^0 (1 - p)^n = 1 \times 1 \times (1 - p)^n = (1 - p)^n \quad \text{Eq. 1.4}$$

Consequently, the probability of having at least one node in the overlap region can be computed as follows:

$$P_r(X > 0) = 1 - P_r(X = 0) = 1 - (1 - p)^n \quad \text{Eq. 1.5}$$

The graph for  $P_r(X > 0)$  for  $d = 450$  meters and  $R = 250$  meters is plotted in Fig. 1.6.

It can be observed that as node density increases, there is a higher probability of finding a node in the intersection of the transmission coverage areas of two nodes.

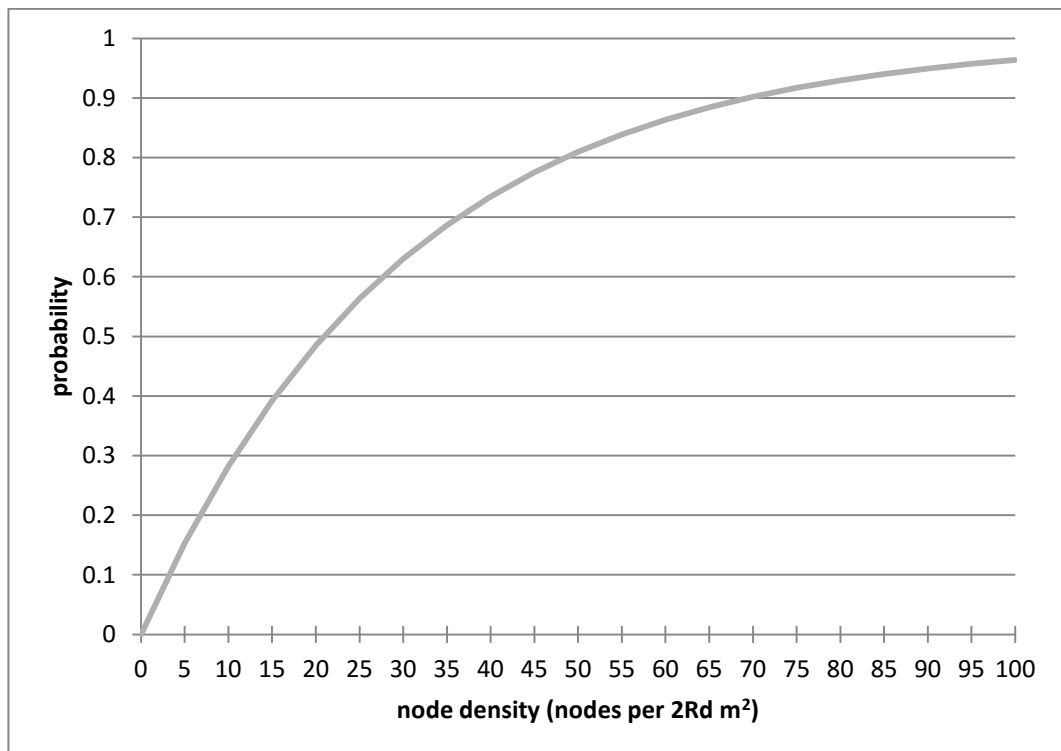


Figure 1.6: Probability of two nodes having common border nodes

When a route is broken, data transmission is halted and the routing protocol reconfigures the network by performing route repair or initiates a new route discovery

process. When routes break frequently, the network performance suffers because nodes waste much channel capacity communicating control packets among themselves to compute the routes or keep the routes updated. Motivated to improve the performance of mobile ad hoc networks, in this thesis we seek to improve route stability.

Besides fast adaptation to rapid topology changes, multi-hop communication also increases the network load. The network in Fig. 1.7 shows a multi-hop route with three links (A, B), (B, C), and (C, D), and the conflict graph for the network. From Fig. 1.7a, it can be observed that the delivery of a packet from node A to node D requires three transmissions if we disregard packet loss and retransmissions. In other words, there is an increased network load when nodes communicate using multi-hop communication. In conventional single-channel wireless ad hoc networks, nearby nodes compete with each other for access to the channel to send their packets. We use the Protocol Model of Interference (Gupta & Kumar, 2000) for determining link interference, and assume that the IEEE 802.11 MAC is used and two-hop interference range. According to the interference model, a transmission is interfered if another transmitter within interference range from the receiver transmits at the same time. Furthermore, in the IEEE 802.11 MAC, a transmitter has to sense the medium to be free before it is allowed to transmit. In Fig. 1.7a, for each link in the network, either the receiver or the transmitter or both are located within two hops from each of the other transmitters in the network. Hence, all the links interfere with each other and cannot be scheduled simultaneously. Therefore, the maximum end-to-end throughput possible with this three-hop route is at most one-third of the maximum channel throughput.

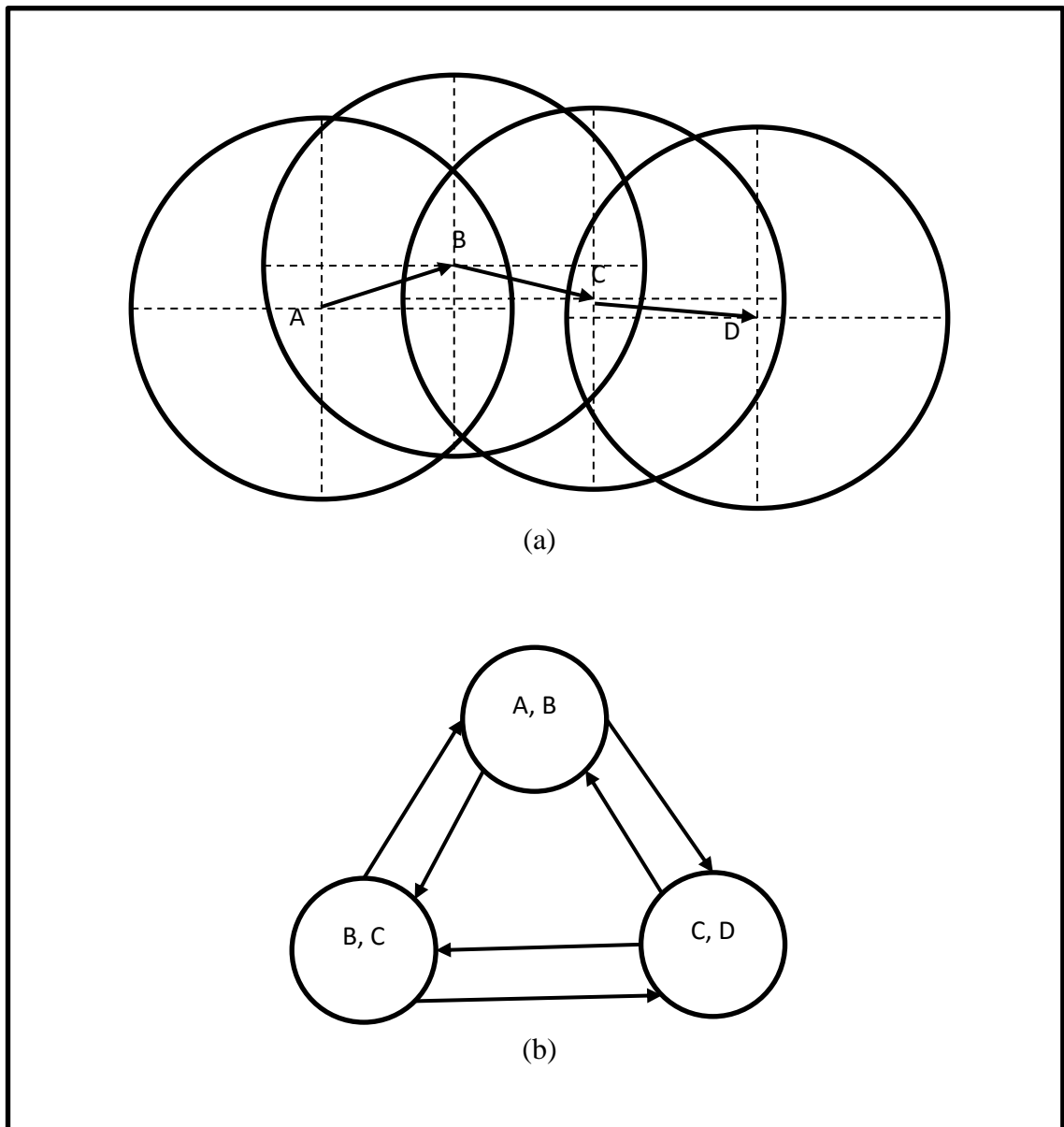


Figure 1.7: (a) multi-hop route, (b) conflict graph

Network coding can be used to counter the increase in network load. With network coding, a node can combine multiple packets together and send only the resultant encoded packet. One of the benefits of using network coding is that network performance can be improved without incurring additional cost. Motivated by this, we seek to improve network throughput of wireless ad hoc networks using network coding.

Due to the distributed nature of wireless ad hoc networks, a source node and a destination node may not be located within transmission range from each other, as shown in Figure 1.8. For a packet from the source node to reach the destination node, intermediate nodes are required to relay the packet. In broadcasting, every node in the

network other than the source node can be viewed as destination nodes for the packet from the source node. It is extremely challenging to devise a transmission plan so that a packet can reach all the nodes in the network while at the same time using a minimum number of transmissions to do so. Motivated by this challenging issue, we seek to design an efficient distributed protocol for broadcasting in wireless ad hoc networks that can effectively reduce redundant transmissions while at the same time maintain perfect reachability to all nodes in a network.

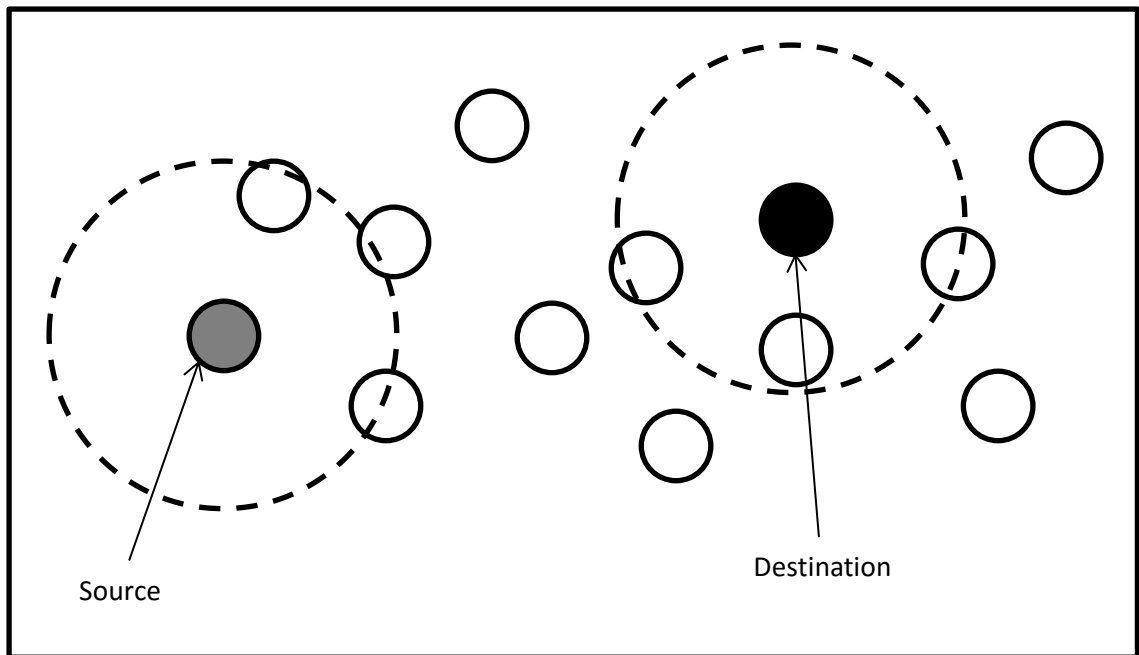


Figure 1.8: Two nodes may be located outside the transmission range of each other

## 1.4 Research Aims and Objectives

The main aim of this thesis is to address some performance issues in wireless ad hoc networks. More specifically, we aim to: (1) improve the performance of mobile ad hoc networks in the presence of node mobility, (2) improve network throughput in wireless ad hoc networks, and (3) reduce redundant transmissions during broadcasting in wireless ad hoc networks. The objectives for achieving these aims are as follows:

1. Investigate the factors affecting network performance in mobile ad hoc networks, design a method for improving route stability in mobile ad hoc networks, and compare the proposed method with other methods.
2. Investigate the reasons wireless ad hoc networks suffer from low network throughput, design a routing protocol to improve the throughput of wireless ad hoc networks, and compare the proposed routing protocol with others to check if it provides better network performance.
3. Design an enhanced broadcast protocol for wireless ad hoc networks that reduces the number of redundant transmissions in a broadcast while ensuring packet reachability to all nodes in a network, and evaluate the performance of the proposed broadcast protocol in terms of its effectiveness in reducing redundant transmissions and its ability to obtain perfect packet reachability.

## 1.5 Research Methods

We seek to improve route stability in mobile ad hoc networks in order to improve network performance in Chapter 2. First, we will investigate the factors affecting route stability in mobile ad hoc networks. Next, we will develop a model for analyzing the remaining lifetime of the link between two neighboring nodes based on their motion. Then, using random velocities for the two nodes, we will obtain the relationship between the expected link remaining lifetime and the initial distance between the two nodes. From there, we will develop a method to increase route stability by finding routes consisting of good links. Then, we will implement the method in network simulator 2 (ns-2) (“The Network Simulator - ns-2,” n.d.) as it is the most popular network simulator and is also free. We will also implement other methods in ns-2 so that we can compare our method against them. Then, we will develop appropriate tests to compare the methods. Finally, we will analyze the results and provide conclusions.

We use the idea of network load reduction by combining several packets for transmission (inter-flow network coding) in Section 3. First, we will study the existing coding conditions to investigate their correctness and effectiveness. Then, we will try to improve upon those coding conditions. After that, we will modify the route discovery process in the Dynamic Source Routing (DSR) (Johnson et al., 2001) protocol to allow nodes to gather the necessary information during route discoveries for using our proposed coding conditions. We will also develop a new routing metric to guide nodes in path selection to fully exploit the network coding benefit. After that, we will modify ns-2 to support simulation of inter-flow network coding, and implement our routing protocol into ns-2. Next, we will develop a set of appropriate tests for comparing our protocol with other routing protocols. Finally, we will analyze the results and provide conclusions.



We seek to improve the efficiency of the existing broadcast protocols in reducing redundant transmissions during broadcasting in wireless ad hoc networks in Chapter 4. First, we will investigate the existing broadcast protocols. Next, we will look for ways to improve these protocols. After that, we will implement our protocol and other protocols in ns-2. Then, we will compare the protocols using appropriate tests. Finally, we will analyze the results and provide conclusions.

## **1.6 Research Contributions**

The contributions of this thesis are:

1. The proposal of novel routing metrics that are able to discover and establish routes that are more stable than other routing metrics.
2. The proposal of a novel method that enables the proposed routing metrics to be used without requiring additional hardware (sensors) such as Global Positioning System (GPS) sensor and compass.
3. The proposal of a novel set of coding conditions that increases coding opportunities compared to existing coding conditions.
4. The proposal of a novel routing metric that causes the source nodes to select ideal paths for sending their packets on to improve network performance considering coding opportunities and network load.
5. The proposal of a broadcast protocol that enhances the efficiency of an existing broadcast protocol while maintaining packet reachability to all nodes in a network.

## 1.7 Thesis Outline

In this chapter, we provide a brief overview of wireless ad hoc networks and discuss some of the issues and challenges in wireless ad hoc networks. We also outline the research motivations, aims and objectives, methods, and contributions. For your convenience, the overview of this thesis is provided in Figure 1.9.

In Chapter 2, we improve route stability in mobile ad hoc networks. Two routing metrics are proposed. The first routing metric called Route Stability 1 (RS1), which considers link length, penalizes links that exceed a certain link length threshold, and favors paths with short links. The second routing metric called Route Stability 2 (RS2), extends RS1 to also consider node mobility in addition to link length. The use of link length information (and node mobility information for RS2) necessitates that additional hardware (sensors) be used. To resolve this problem, we further propose a method to estimate link length, and a method to estimate node mobility. We then evaluate the proposed routing metrics against other routing metrics and the hop count routing metric using an extensive set of tests.

In Chapter 3, we propose a coding- and load-aware ad hoc routing protocol called Network Coding Routing (NCRT). A new set of coding conditions called Improved Generalized Coding Conditions (IGCC), which overcomes the deficiencies of the existing coding conditions and promotes more packet encodings, is first proposed. Then, we formulate a route discovery method for nodes to gather the information used in IGCC during route discoveries to allow them to determine if they can encode packets together. A routing metric is also proposed to guide a source node selects the most suitable path for sending its packets considering coding opportunities and network load. Finally, we evaluate the performance of NCRT with a simulation study.

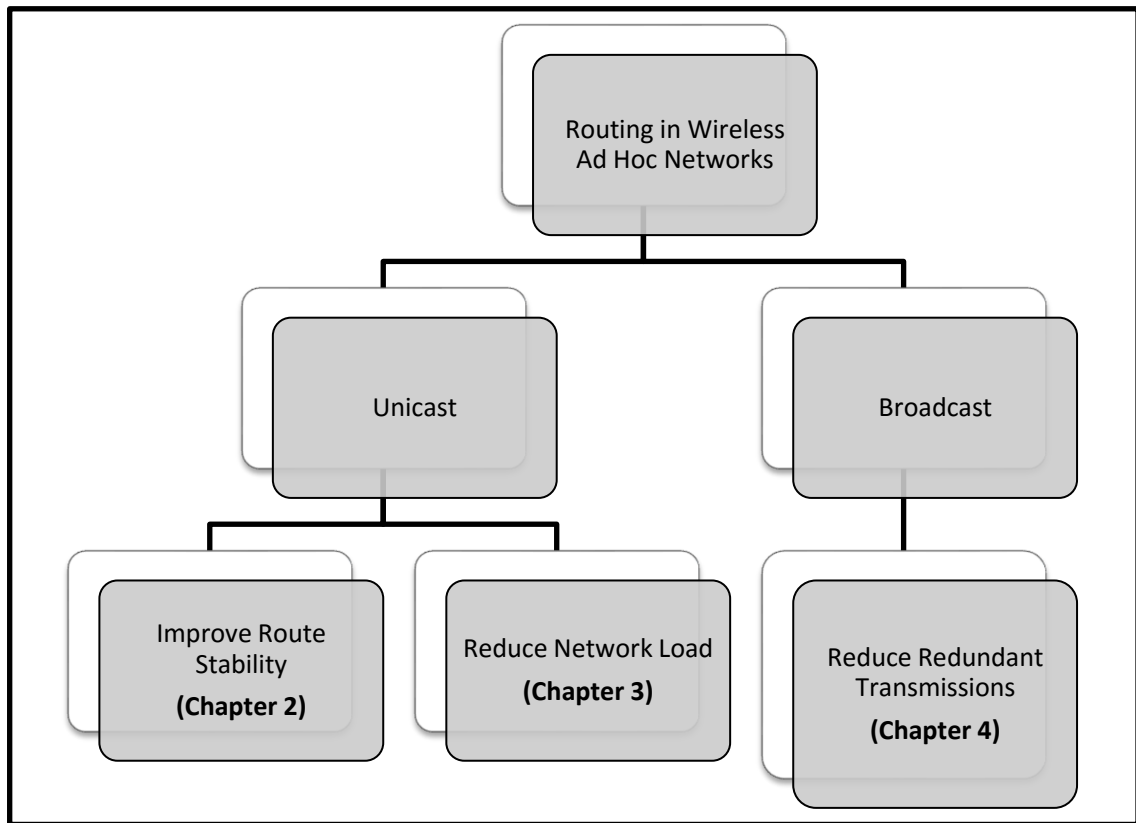


Figure 1.9: Overview of this thesis

In Chapter 4, we present an efficient broadcast protocol called Improved Partial Dominant Pruning (IPDP), which reduces the number of redundant transmissions while ensuring packet reachability to all nodes in a network. We begin by reviewing related work and then examine an efficient broadcast protocol called Partial Dominant Pruning (PDP) in more detail. After that, we extend PDP to further improve its effectiveness. Two enhancements for IPDP are subsequently proposed to further reduce redundant transmissions. Finally, we verify the performance of IPDP with a simulation study.

In Chapter 5, we summarize our findings and propose potential future work.

# CHAPTER 2: ROUTE STABILITY METRICS FOR MOBILE AD HOC NETWORKS

## 2.1 Introduction

The hop count routing metric, which minimizes the number of transmissions required to send a packet from the source to the destination, is one of the most widely used routing metrics in ad hoc routing protocols due to its simplicity and effectiveness. It is vital to reduce redundant transmissions in wireless ad hoc networks because the channel is shared and a packet may require multiple transmissions to reach its destination from the source due to multi-hop communication. When redundant transmissions are reduced, channel contention and interference are also reduced. This can improve the network throughput as less channel contention and interference allow for more concurrent transmissions. Besides, with a lower hop count path, the end-to-end delay is also reduced.

It is well known that the hop count routing metric does not perform well in high node mobility scenarios, where nodes move at high speeds and the network topology changes rapidly. Besides, the hop count routing metric is also known to cause the border (Jianzhen Sun et al., 2010)/edge (Yoon, 2002) effect in high node density scenarios. In high node density scenarios, links are formed through border nodes and have short lifetimes. A small movement from either of the two nodes of such a link could cause the link to break. In a wireless ad hoc network, when a link is broken, the network self-reconfigures by having nodes update their route information using control packets. The communication of control packets results in more channel contention and interference. As a result, the network throughput might be reduced. Therefore, it is vital to prevent the self-reconfiguration process to happen too frequently. A way to do this is to improve route stability. In this regard, stability-based routing metrics have been proposed.

However, they are either only marginally effective or incur additional cost by requiring the use of information from sensors such as GPS sensor and compass. In this chapter, we propose two routing metrics for discovering and establishing stable routes that can be used without using sensors. We show through analysis and simulations that the proposed routing metrics are effective and they outperform the hop count and other routing metrics.

The remainder of this chapter is organized as follows. Related work is reviewed in Section 2.2. In Section 2.3, we provide the details of our route stability metrics. Results and discussion are provided in Section 2.4. Finally, we conclude in Section 2.5.

## 2.2 Related Work

Routing metrics are used for ranking paths according to some criteria and is used in making routing decisions. It is well known that the hop count routing metric is not suitable for high node mobility scenarios. Several routing metrics for wireless mesh networks (WMNs) were reviewed in (Campista et al., 2008), such as Expected Transmission Count (ETX) (Couto, Aguayo, Bicket, & Morris, 2005), Expected Transmission Time (ETT) (Draves, Padhye, & Zill, 2004), Weighted Cumulative Expected Transmission Time (WCETT) (Draves et al., 2004), and Metric of Interference and Channel Switching (MIC) (Yaling Yang, Wang, & Kravets, 2005). However, these routing metrics have a different objective, for example, to improve network throughput or reduce interference. Hence, they are not suitable for the problem that we seek to solve.

Various information can be used to improve network performance in mobile scenarios. Some information can only be obtained by using sensors. Node location information and node mobility information (speed and direction) are used to estimate the remaining time before a link breaks called Link Expiration Time (LET) in the Flow Oriented Routing Protocol (FORP) (Gerla, 1999). In FORP, the path with the highest Route Expiration Time (RET), which is the minimum of the LETs of the links in a path, is preferred over other paths. In addition to LET, the Power and Mobility Aware Routing (PMAR) (Wesley Chee-Wah Tan, Bose, & Cheng, 2011)(W. Chee-Wah Tan, Bose, & Cheng, 2012) protocol also employs RREQ propagation control using node location information (method first used in location-aided routing protocols such as LAR (Ko & Vaidya, 2000) and PMLAR (Lu & Feng, 2005)). In the AODV-Reliable Route Selection (AODV-RRS) (Kim, 2001) routing protocol, during a route discovery, only nodes that are located inside stable zones forward RREQs. A stable zone is defined as a circular region centered at the stable zone center with radius  $r = R - \beta \times S_{MN}$ , where  $R$

is the node transmission range,  $\beta$  is a tunable constant, and  $S_{MN}$  is the movement speed of a considered node  $MN$ . Suppose a node  $x$ , initially at location  $A$ , has a RREQ to forward and is moving towards  $A^*$ , as shown in Figure 2.1. It defines a circular region called stable zone with radius  $r \leq R$  centered at the stable zone center  $A^*$ . Node  $x$  records the location of the stable zone center and the radius of the stable zone in the RREQ before forwarding the RREQ. When the neighbors of node  $x$  receive the RREQ, only those that are located inside the stable zone forward the RREQ. Obviously, AODV-RRS requires the use of node location information and node mobility information. Link length is mapped onto a value called Link Availability and Link Stability Factor (LSF) in a nameless DSR-based routing protocol (H. Peng & Shao, 2010) and Link Stability Based AODV (LSB-AODV) (Jianzhen Sun et al., 2010), respectively. The destination selects the path with the highest Path Availability/Path Stability Factor (PSF), where the Path Availability/PSF of a path is the minimum link availability/Link Stability Factor (LSF) of all the link availabilities/LSFs in the path. In LSB-AODV, link lengths are estimated using the signal strengths of received packets, which can be highly unreliable. Node heading direction information, which is obtained using compass, is used in the Heading-direction Angles Routing Protocol (HARP) (Al-Akaidi & Alchaita, 2007). The main idea is to propagate the RREQ from the source to the destination along a single direction during a route discovery. As the nodes in a route established in such a manner move as a group along the same direction, the links in the route are less prone to breakage.



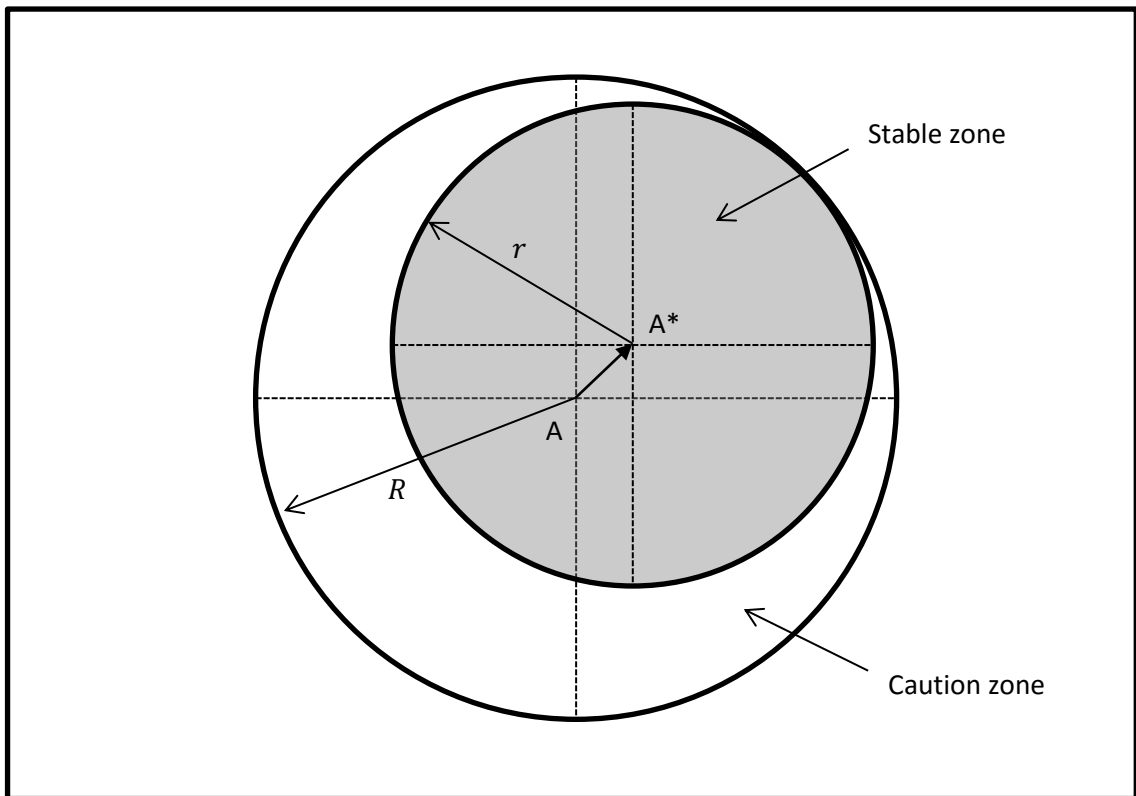


Figure 2.1: Stable zone and cautious zone

Some methods use readily available information to improve network performance in mobile scenarios. In the Associativity-Based Routing (ABR) (Toh, 1997) protocol, link stability is measured using “associativity ticks”, which is the measure of time the two nodes of a link are connected to each other. A node can measure the associativity of a neighbor by counting the number of beacon packets it received from the neighbor. The author claimed that links that are stable for at least a threshold amount of time are more likely to continue being stable. To support the claim, the author performed a mobility trace of 52 badge wearers from the Active Badge System for five consecutive days from 8 am to 6.30 pm at the Cambridge University Computer Laboratory (Toh, 1997). The number of bridge nodes, which are common nodes of the two nodes of a link, is used as the measure of the stability of the link in an extension of the DSR routing protocol (Penz, 2007). The Q-Learning AODV (QLAODV) (Wu, Kumekawa, & Kato, 2009) routing protocol uses a metric called Mobility Factor (MF) to measure link stability. The MF of a node can be computed using only local connectivity (neighborhood)

information, as is evident in Eq. 2.1.

$$MF = \begin{cases} \sqrt{1 - \frac{|(N_x \cap \overline{N_x^p}) \cup (\overline{N_x} \cap N_x^p)|}{|N_x \cup N_x^p|}}, & \text{if } N_x \cup N_x^p \neq \emptyset \\ 0 & \text{otherwise} \end{cases} \quad \text{Eq. 2.1}$$

where  $N_x$  is the current neighbor set of a considered node  $x$ , and  $N_x^p$  is the previous neighbor set of node  $x$ . We illustrate how MF is computed using Figure 2.2.  $N_x \cap \overline{N_x^p}$  is the set of nodes in the blue-colored region, i.e., {7, 8, 9, 10}.  $\overline{N_x} \cap N_x^p$  is the set of nodes in the yellow-colored region, i.e., {1, 2, 4, 5}. Hence, MF of node  $x$  is given by

$$\sqrt{1 - \frac{|{7,8,9,10} \cup {1,2,4,5}|}{|{3,6,7,8,9,10} \cup {1,2,3,4,5,6}|}} = 0.4472. \text{ MF is also used in the MQ-Routing (Macone, Oddi, \& Pietrabissa, 2013) protocol, which is mobility-, GPS-, and energy-aware. A metric quite similar to MF called Neighbor Change Ratio (NCR) was proposed in (Dutkiewicz, 2006). NCR is related to MF mathematically as } NCR = MF^2.$$

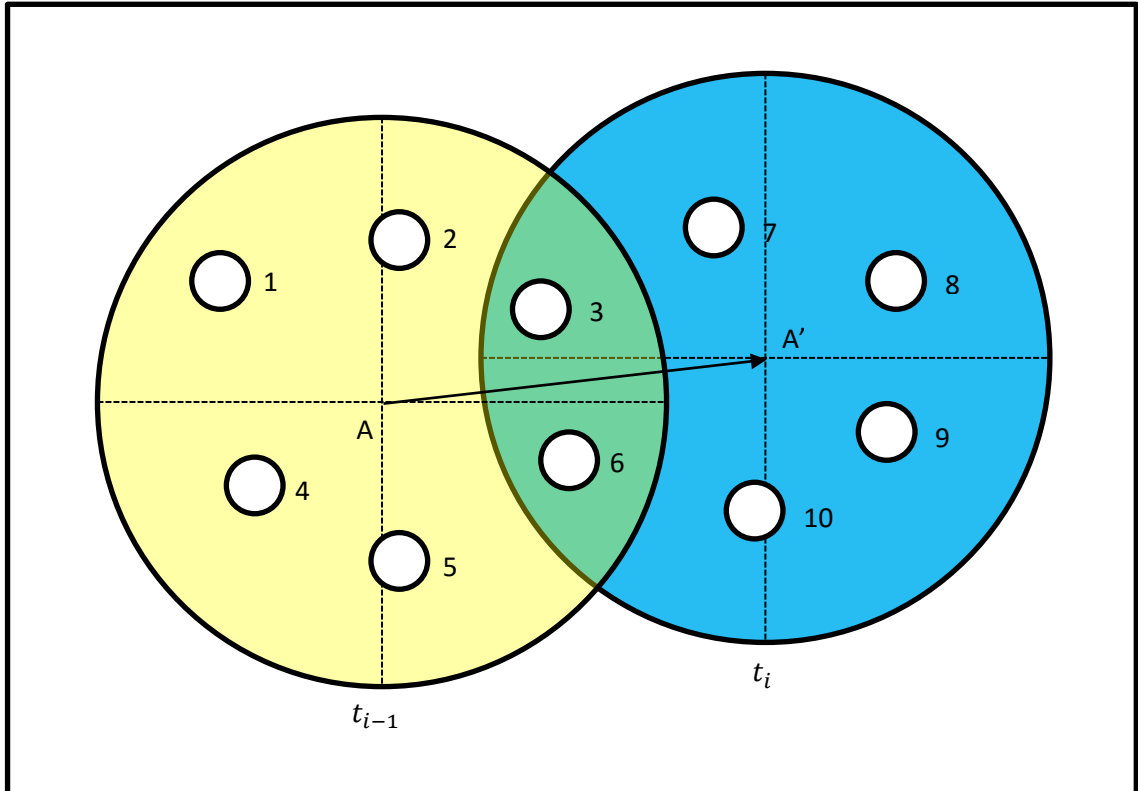


Figure 2.2: An illustration to show how various routing metrics are computed

The AD-AODV (Y. Wang et al., 2012) routing protocol uses a routing metric based on neighbor set change and hop count. The equation of the routing metric is given in Eq.

2.2, where  $M$  is the metric value of a considered path  $p$ ,  $\lambda$  is a tunable constant,  $hops$  is the number of hops in path  $p$ ,  $D$  is the number of intermediate nodes in path  $p$ ,  $J_i$  is the number of nodes that joined node  $i$ 's coverage area in the last HELLO interval,  $L_i$  is the number of nodes that left node  $i$ 's coverage area in the last HELLO interval, and  $N_i$  is the number of nodes in node  $i$ 's neighbor list.

$$M = avr(p) + \lambda * hops \quad \text{Eq. 2.2}$$

where:

$$avr(p) = \frac{\sum_{i=1}^D Q_i}{D} \text{ and } Q_i = \frac{J_i + L_i}{N_i}$$

Using the example in Figure 2.2,  $Q_x$  is given by  $(|\{7, 8, 9, 10\}| + |\{1, 2, 4, 5\}|) / |\{3, 6, 7, 8, 9, 10\}| = 1.33$ . The Path Encounter Rate (PER) (Son, Minh, Sexton, & Aslam, 2014) routing metric guides nodes discover and establish stable routes by preferring paths consisting of nodes with low Average Encounter Rates (AERs). The PER of a considered path  $p$  can be computed using the following equation, where  $n$  is an intermediate node in path  $p$ .

$$PER = \sum_{n \in p} (AER_n^2) \quad \text{Eq. 2.3}$$

where:

$$AER_x = \frac{|N_x - N_x^p|}{T}$$

We use Figure 2.2 to illustrate how the AER of a node is computed. Suppose node  $x$  moves from location A at time  $t_{i-1}$  to a new location A' at time  $t_i$ . The neighbor set of node  $x$  at  $t_i$  is the set of nodes in the blue-colored circle, i.e.,  $\{3, 6, 7, 8, 9, 10\}$ . The neighbor set of node  $x$  at  $t_{i-1}$  is the set of nodes in the yellow-colored circle, i.e.,  $\{1, 2, 3, 4, 5, 6\}$ . The new encounters of node  $x$  are the nodes in the blue-colored region, i.e.,  $\{7, 8, 9, 10\}$ . If the time interval between two successive HELLO messages given by  $T = t_i - t_{i-1}$  is set to 1 second, then AER of node  $x$  is given by  $|\{7, 8, 9, 10\}| / 1 = 4$ . It

was claimed that PER outperforms the hop count routing metric because PER leads to the formation of routes that are formed by low mobility nodes or nodes in low node density areas. In the STable OLSR (ST\_OLSR) (Moussaoui, Semchedine, & Boukerram, 2014) protocol, the variance of the received packet powers is used as the measure of the stability of a link.

Various information is used to measure and imply route stability. Some information is readily available while others can only be obtained using sensors. On the one hand, it is desirable to improve route stability without using sensors to reduce cost. On the other hand, methods that do not use sensors usually provide only negligible performance gains.

## 2.3 Route Stability Metrics

### 2.3.1 Estimating Link Remaining Lifetime

In mobile ad hoc networks (MANETs), it is desirable to discover, establish, and use stable routes, i.e., routes consisting of links with long link remaining lifetime (LRL). If nodes are equipped with Global Positioning System (GPS) sensors, the remaining lifetime of a link can be estimated based on node location information and node mobility information. Figure 2.3 shows two nodes A and B initially separated by distance  $d$ .

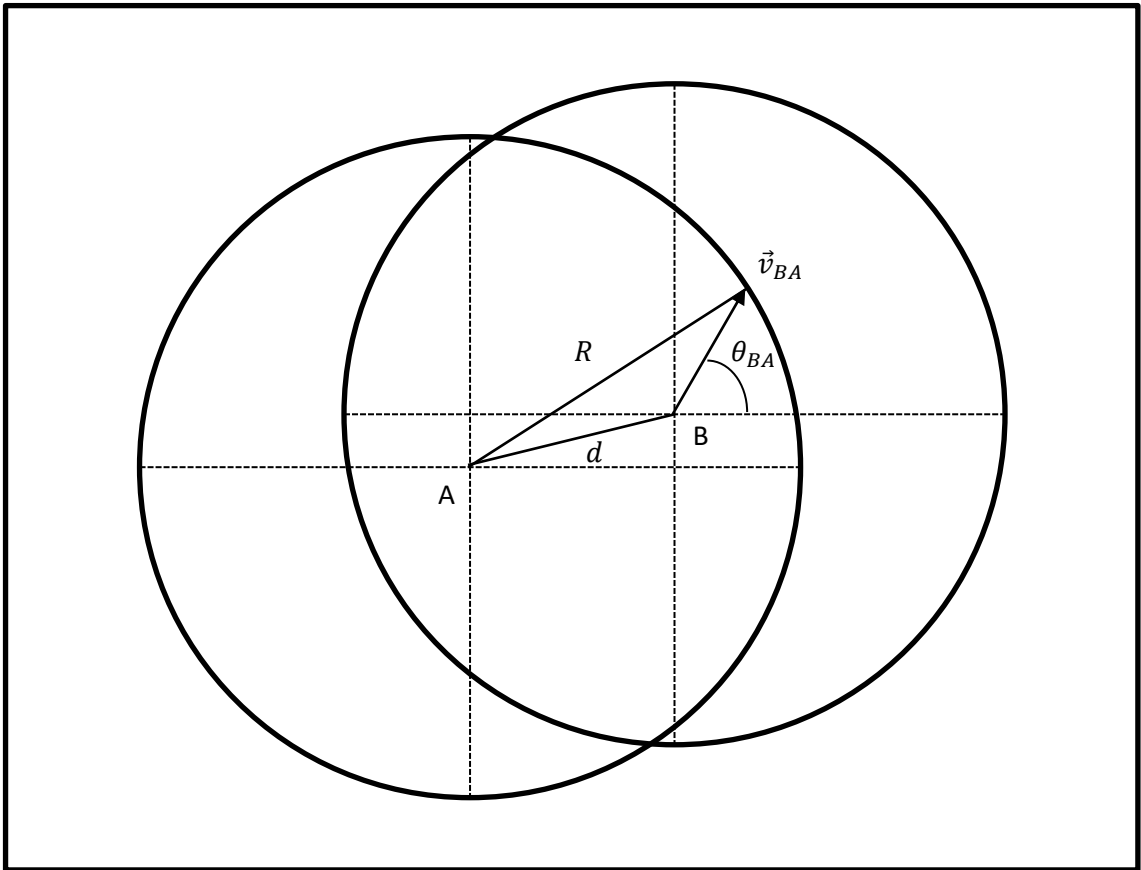


Figure 2.3: Estimating the remaining lifetime of a link using node location information and node mobility information

To derive the time link A-B remains up, we compute the amount of time node B remains within transmission range  $R$  from node A. The position of node B with respect to node A is given by  $(x_{BA}, y_{BA}) = (x_B - x_A, y_B - y_A)$ . The distance of node B from node A is initially given by  $d = \sqrt{x_{BA}^2 + y_{BA}^2}$ . Subsequently, due to the motion of the

two nodes, the position of node B with respect to node A after  $t$  seconds is given by  $(x_{BA}', y_{BA}')$ , where:

$$x_{BA}' = x_{BA} + |\vec{v}_{BA}|t \cos \theta_{BA} \quad \text{Eq. 2.4}$$

and

$$y_{BA}' = y_{BA} + |\vec{v}_{BA}|t \sin \theta_{BA}$$

We would like to know the amount of time  $t$  before node B goes out of range from node A, and vice versa. Assuming that the velocity of node B with respect to node A is constant, the critical time at which node B goes out of range from node A can be determined by solving the following equation, where  $d'$  is the critical distance between nodes A and B before they are out of range from each other.

$$(d')^2 = (x_{BA}')^2 + (y_{BA}')^2 = R^2 \quad \text{Eq. 2.5}$$

Substituting Eq. 2.4 into Eq. 2.5 and rearranging, we have the following equation:

$$|\vec{v}_{BA}|^2 t^2 + 2|\vec{v}_{BA}|(x_{BA} \cos \theta_{BA} + y_{BA} \sin \theta_{BA})t + d^2 - R^2 = 0 \quad \text{Eq. 2.6}$$

where:

$$\vec{v}_A = (|\vec{v}_A| \cos \theta_A)\vec{i} + (|\vec{v}_A| \sin \theta_A)\vec{j}$$

$$\vec{v}_B = (|\vec{v}_B| \cos \theta_B)\vec{i} + (|\vec{v}_B| \sin \theta_B)\vec{j}$$

$$\vec{v}_{BA} = \vec{v}_B - \vec{v}_A$$

$$= (|\vec{v}_B| \cos \theta_B - |\vec{v}_A| \cos \theta_A)\vec{i} + (|\vec{v}_B| \sin \theta_B - |\vec{v}_A| \sin \theta_A)\vec{j}$$

$$|\vec{v}_{BA}| = \sqrt{(|\vec{v}_B| \cos \theta_B - |\vec{v}_A| \cos \theta_A)^2 + (|\vec{v}_B| \sin \theta_B - |\vec{v}_A| \sin \theta_A)^2}$$

$$\theta_{BA} = \tan^{-1}\left(\frac{(|\vec{v}_B| \sin \theta_B - |\vec{v}_A| \sin \theta_A)}{(|\vec{v}_B| \cos \theta_B - |\vec{v}_A| \cos \theta_A)}\right)$$

Note that Eq. 2.6 is a quadratic equation of the form  $ax^2 + bx + c = 0$ , where  $x = t$ ,  $a = |\vec{v}_{BA}|^2$ ,  $b = 2|\vec{v}_{BA}|(x_{BA} \cos \theta_{BA} + y_{BA} \sin \theta_{BA})$ , and  $c = d^2 - R^2$ . The root of a quadratic equation can be computed by using the method of completing the squares with the following equation:

$$x = \frac{-b + \sqrt{b^2 - 4ac}}{2a} \quad \text{Eq. 2.7}$$

Since  $d \leq R$  (the two nodes are initially connected), we have  $c = d^2 - R^2 \leq 0$ . Since  $a = |\vec{v}_{BA}|^2 \geq 0$  and  $c = d^2 - R^2 \leq 0$ , we have  $b^2 - 4ac \geq b^2$ . Hence, we have  $\sqrt{b^2 - 4ac} \geq b$ . In other words, one of the roots is positive while the other is negative. Since we cannot have negative duration, the positive root is the correct answer.

### 2.3.2 Effect of Link Length on the Link Remaining Lifetime

The link remaining lifetime (LRL) of a link can be used directly as a routing metric. For example, it is called Link Expiration Time (LET) and is used in the Flow Oriented Routing Protocol (FORP) (Gerla, 1999). However, to calculate LRLs, nodes are required to obtain not only their locations, but also their movement speeds and directions. In addition, from Eq. 2.6, it can be observed that node velocities that are measured at one time are used to estimate the LRL. In other words, current information is used to estimate a future outcome. However, nodes do not necessarily move at constant velocities. A node could change its velocity abruptly after its velocity is sampled. Considering that using the LRL metric not only increases cost by requiring information from sensors but could also lead to inaccurate LRL values, an alternate way of quantifying link stability is required.

Intuitively, shorter links have longer LRLs than longer links. To verify this, we performed the following experiment. We put node A at the origin and node B to the right of node A, separating the two nodes with a certain distance less than or equal to the transmission range  $R$ . Then we assign random node velocities to the two nodes, and determine the time required for the two nodes to move out of range from each other by solving Eq. 2.6. For a particular value of the initial link length, we repeat the experiment many times. Varying the link length and using different maximum node speeds, we obtained the graphs in Figure 2.4.

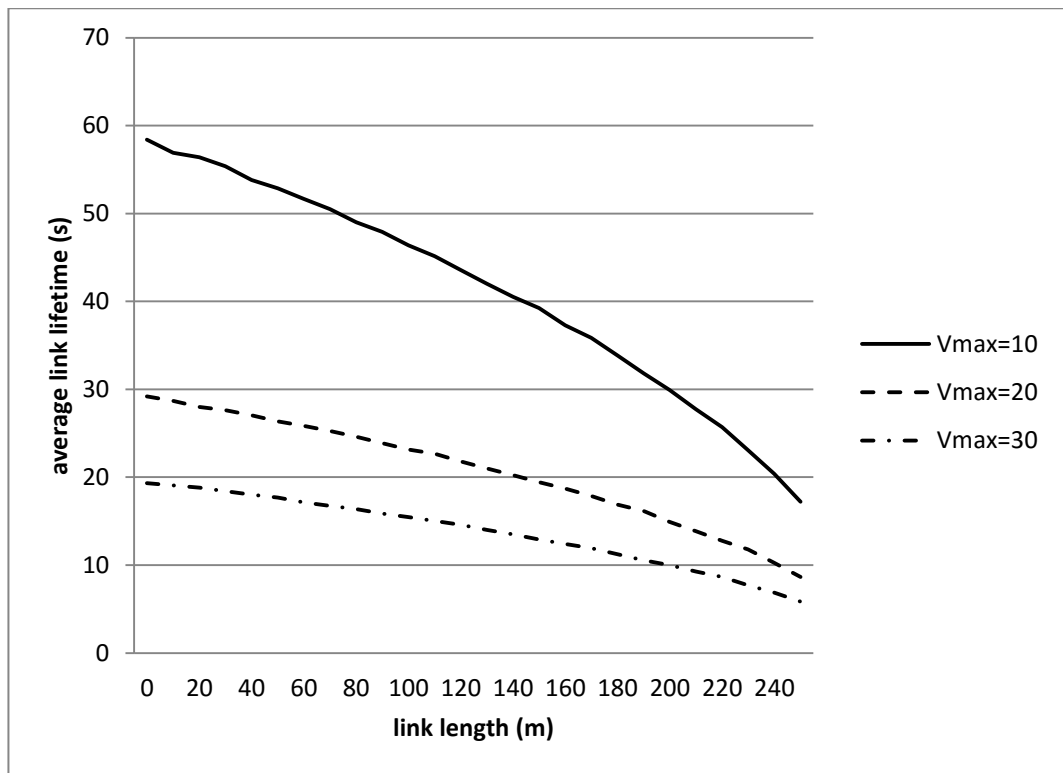


Figure 2.4: Average link lifetime vs. link length

From the figure, we observed that shorter links are indeed more stable than longer links. Furthermore, as the maximum node speed increases, the average link lifetime decreases. However, since the maximum node speed is not something that we can enforce (because the nodes in a network often belong to different owners), we focus on reducing link length. One question arises – how do we modify a routing protocol such that it discovers and establishes routes with short links? In reactive routing protocols, one way to do so is by modifying the route discovery process such that paths consisting of long links are not discovered. This is the idea proposed in the AODV-RRS (Kim, 2001) routing protocol with the safety zone (SZ)-based route discovery method. To avoid the formation of routes consisting of long links, in a route discovery, nodes that reside outside a threshold distance from their previous hops drop the RREQs that they receive. Using the safety zone based route discovery method with the LRL metric for route selection, we have the safety zone based route discovery with LRL metric method (SZ/LRL). This method selects the path with the highest estimated remaining lifetime



among paths consisting of links that are shorter than a threshold length. It guarantees a certain degree of route stability by using link length information and it performs well as can be seen in Section 2.4.

### 2.3.3 Route Stability 1 (RS1) Metric

We discovered several issues with the SZ/LRL method. First, the use of the safety zone based route discovery method could result in a scenario where a path from a source to a destination could not be found even if one or more paths between the node pair exist in the network. This happens because some RREQs are dropped during route discoveries. Second, LRLs are computed using node location information and node velocity information. Using this information has the drawback that sensors such as GPS sensor must be used to provide this information.

To avoid these problems, an alternate method of discovering and establishing routes with short links is needed. A routing metric could be used. First, a routing metric does not drop RREQs; hence, the problem of not finding a path between two nodes in a network is resolved. To resolve the second problem, we could design the routing metric to not use information that can only be obtained using sensors. We now propose the *Route Stability 1 (RS1)* metric. Its equation is given as follows:

$$RS1_p = \sum_{l \in p} (length_l - \min(length_l, THRESHOLD\_LENGTH))^2 \quad \text{Eq. 2.8}$$

where  $l$  is a link in a considered path  $p$ ,  $length_l$  is the length of link  $l$ , and  $THRESHOLD\_LENGTH$  is the threshold link length. The selected path  $p^*$  is defined as follows:

$$p^* = \arg \min_{p \in \mathbf{P}} (RS1_p) \quad \text{Eq. 2.9}$$

where  $\mathbf{P}$  is the set of discovered or available paths from a source to a destination.

The central idea of the RS1 routing metric is to penalize links that exceed a threshold length. With the use of a threshold link length, we avoid the problem of nodes

selecting very short links leading to the formation of routes with very high hop count (Lal, Laxmi, & Gaur, 2011)(Moussaoui et al., 2014). The minimum sum of squares is also inherently hop count minimizing. This is because when a path consists of more links, there is a higher probability that some links in the path will exceed the threshold link length therefore increasing the metric value of the path, assuming all other things equal.

The RS1 routing metric requires the use of a parameter, i.e., the threshold link length, *THRESHOLD\_LENGTH*. Many heuristics could be used to set the value of this parameter. One heuristic is to set it according to a desired minimum LRL. From Figure 2.3, it can be observed that the LRL of the link between nodes A and B is minimum when the velocity of node B relative to node A is in the direction of from node A to node B. In other words, if nodes A and B are moving directly away from each other, the LRL of the link between them is shortest. Assuming that nodes have a maximum movement speed of  $v_{max}$ , the desired link length threshold can be set to the value of  $length_l^{max}$  using Eq. 2.10, where  $t_l^{desired\_min}$  is the desired minimum LRL of link  $l$ , and  $R$  is the node transmission range.

$$t_l^{desired\_min} = \frac{R - length_l^{max}}{2v_{max}} \Rightarrow length_l^{max} = R - 2v_{max}t_l^{desired\_min} \quad \text{Eq. 2.10}$$

We now illustrate how the RS1 routing metric is computed and used with the example shown in Figure 2.5. The link lengths are given next to the links. Suppose node 1 is the source, node 4 is the destination, and *THRESHOLD\_LENGTH* is set to 200 meters. Three paths exist in the network: 1-2-3-4, 1-5-4, and 1-6-7-8-4. According to the RS1 routing metric, path 1-2-3-4 has the lowest RS1 value and so is selected.

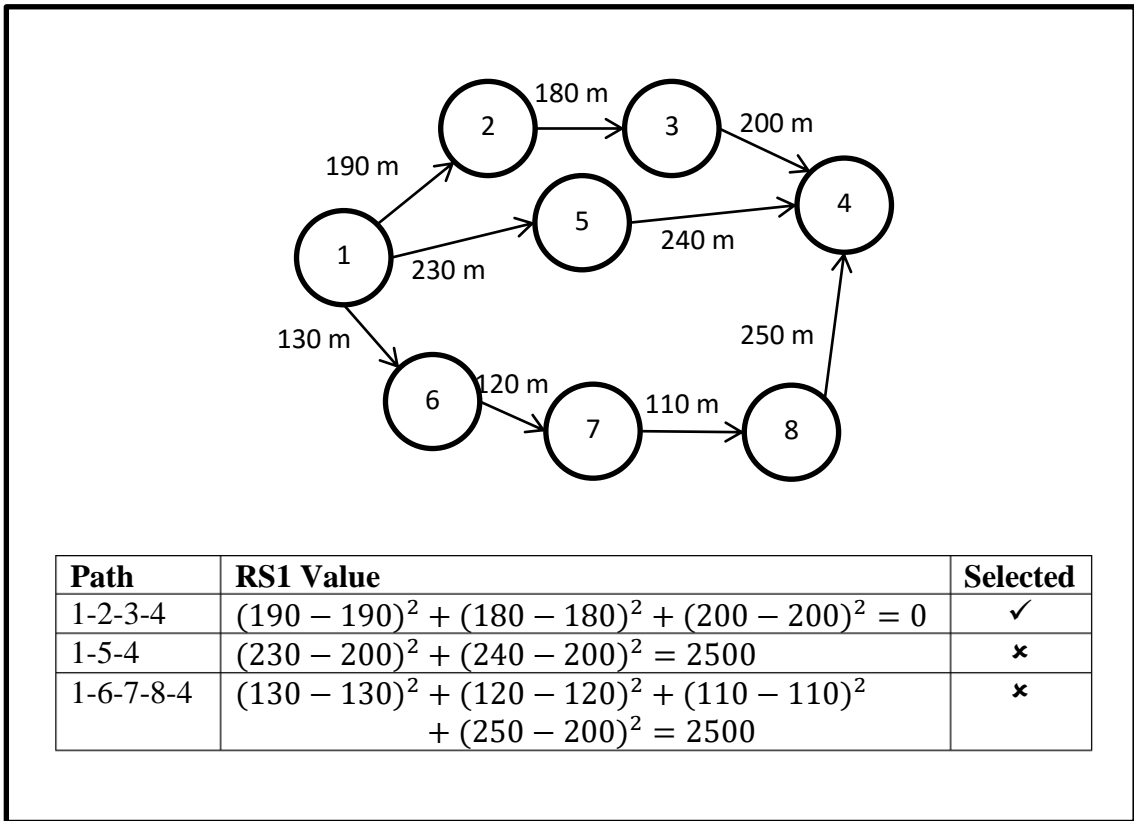


Figure 2.5: An example illustrating how the RS1 routing metric is used

By having a route consisting of short links, the RS1 routing metric can handle the unpredictability of node mobility. For example, suppose the two nodes of a link changes their velocities abruptly. The shortness of the link provides a “buffer” for preventing breakage. As the link is short, it requires some time before the link can become broken. Hence, the RS1 routing metric is effective regardless of the node mobility model used.

### 2.3.4 Route Stability 2 (RS2) Metric

In Section 2.3.3, we proposed the RS1 routing metric, which does not consider node mobility. In this section, we extend it to also consider node mobility. We call the extended routing metric *Route Stability 2 (RS2)*; its equation is given as follows:

$$RS2_p = \sum_{l \in p, n \in p, n \neq src(p)} ((length_l - \min(length_l, THRESHOLD\_LENGTH)) + mobility_n)^2 \quad \text{Eq. 2.11}$$

where  $l$  is a link in a considered path  $p$ ,  $n$  is an intermediate node or the destination node in path  $p$ ,  $mobility_n$  is the measure of the mobility of node  $n$ , and  $src(p)$  is the

source of path  $p$ . When computing the metric value of path  $p$ , for  $mobility_n$ , we plug in the amount of distance in meters node  $n$  moves in one second. Generally, it measures the speed at which node  $n$  moves. Both  $(length_l - \min(length_l, THRESHOLD\_LENGTH))$  and  $mobility_n$  are given the same unit of measurement, i.e., meters, so that we do not add and compare incomparable units akin to comparing apples with oranges. The selected path  $p^*$  is defined as follows:

$$p^* = \arg \min_{p \in \mathcal{P}} (RS2_p) \quad \text{Eq. 2.12}$$

Compared to the RS1 routing metric, the RS2 routing metric penalizes not only paths with long links but also paths that are formed through fast moving nodes.

### 2.3.5 Estimating Link Lengths

In the RS1 and RS2 routing metrics, link length information and node movement speed information are required. A method to obtain them is through the use of sensors such as GPS sensor. However, it is undesirable to use sensors as doing so would incur additional cost. For example, we found the price of the MTS420 GPS sensor board (circa 450 USD) to be about thrice the price of the IRIS XM2110 mote (circa 150 USD). Besides, there are also scenarios where sensors are inapplicable, for example, GPS sensors perform poorly in indoor environments and are unsuitable for low power nodes. In some ad hoc routing protocols, nodes periodically broadcast HELLO messages to allow other nodes to sense their presence. This feature could be exploited to estimate the length of the link between two neighboring nodes.

Figure 2.6 shows two scenarios of the length of the link between two neighboring nodes A and B.

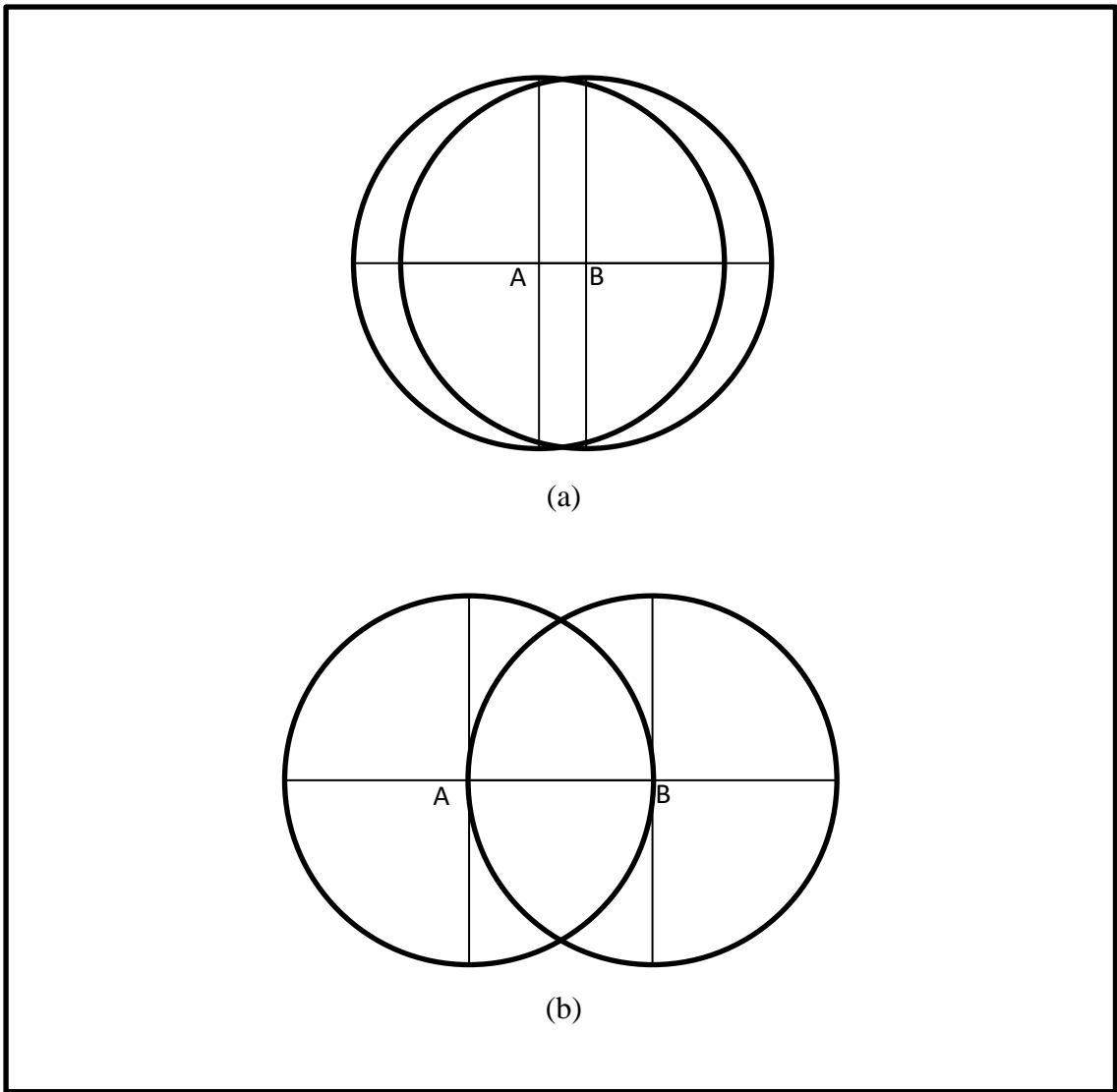


Figure 2.6: Estimating the length of the link between two neighboring nodes

In Figure 2.6a, nodes A and B are located close to each other while in Figure 2.6b the nodes are at a maximum distance from each other, i.e., subject to link A-B not broken. There seems to be a relation between the length of the link between the two nodes and the area of the intersection of their transmission coverage areas. However, a node is unable to measure the area of the intersection of its transmission coverage area and the transmission coverage area of a neighboring node. The length of the link between nodes A and B can be estimated by evaluating the ratio of the number of nodes in the intersection of sets  $\mathbf{U}$  and  $\mathbf{V}$  to the number of nodes in the union of sets  $\mathbf{U}$  and  $\mathbf{V}$ , where  $\mathbf{U} = \{A\} \cup N_A$ ,  $\mathbf{V} = \{B\} \cup N_B$ , and  $N_x$  is the neighbor set of node  $x$ . For simplicity, we refer to the ratio of the number of nodes in the intersection of sets  $\mathbf{U}$  and  $\mathbf{V}$  to the

number of nodes in the union of sets  $U$  and  $V$  as  $overlap\_ratio$ .

$$overlap\_ratio = \frac{|U \cap V|}{|U \cup V|} \quad \text{Eq. 2.13}$$

Assuming that nodes are uniformly distributed, as node density approaches infinity,  $overlap\_ratio$  is approximately equal to the ratio of the area of the overlapping region of two equal circles to the area jointly covered by the two circles. We first derive the area of the overlapping region of two equal circles as shown in Figure 2.7.

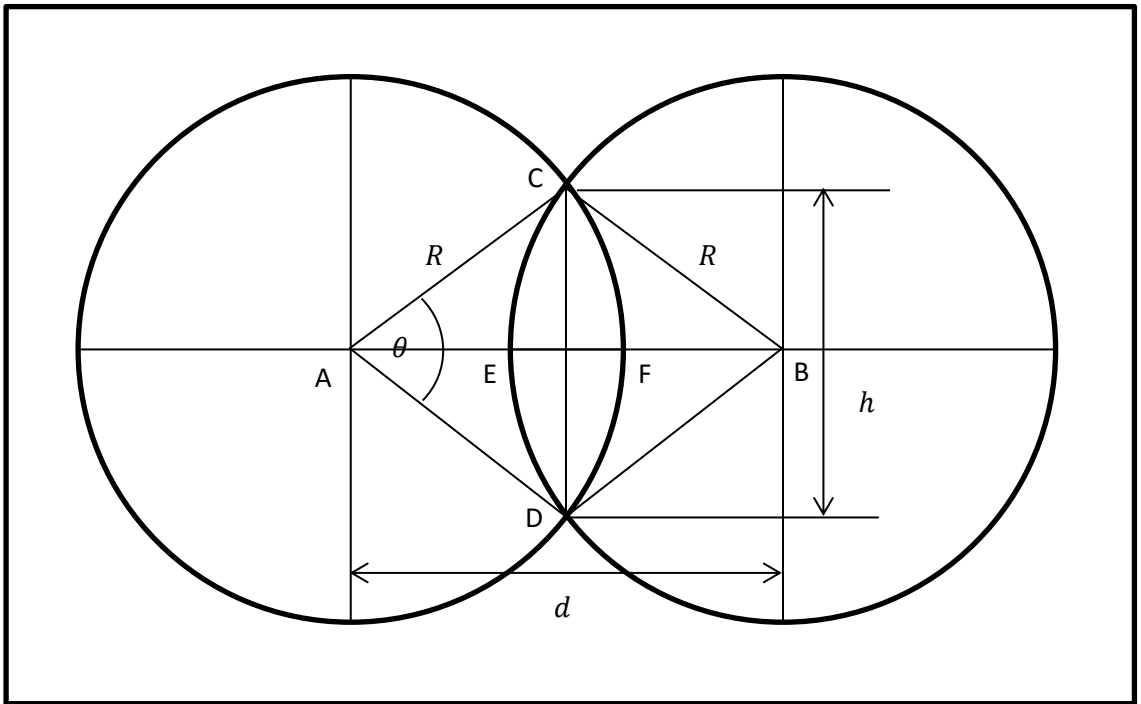


Figure 2.7: Finding the area of the overlapping region of two equal circles

From trigonometry, we have:

$$\cos \frac{\theta}{2} = \frac{d/2}{R} = \frac{d}{2R} \Rightarrow \theta = 2 \cos^{-1} \frac{d}{2R} \quad \text{Eq. 2.14}$$

The area of the sector with angle  $\theta$  (ACFD) can be computed by using the concept of proportionality:

$$area_{ACFD} = \frac{\theta}{2\pi} \times \pi R^2 = \frac{\theta R^2}{2} \quad \text{Eq. 2.15}$$

Substituting Eq. 2.14 into Eq. 2.15, we have:

$$area_{ACFD} = \frac{(2 \cos^{-1} \frac{d}{2R}) R^2}{2} = R^2 \cos^{-1} \frac{d}{2R} \quad \text{Eq. 2.16}$$

Using Pythagoras' Theorem, we have:

$$(d/2)^2 + (h/2)^2 = R^2 \Rightarrow h = \sqrt{4R^2 - d^2} \quad \text{Eq. 2.17}$$

The area of the triangle ACD can be computed as follows:

$$area_{ACD} = \frac{1}{2} \times \frac{d}{2} \times h = \frac{d}{4} \sqrt{4R^2 - d^2} \quad \text{Eq. 2.18}$$

Finally, the area of the overlapping region of the two circles is given by two times the area of the “D-shaped region” CDE or CFD, each of which is given by the area of the sector ACFD minus the area of the triangle ACD.

$$\begin{aligned} area_{overlapping\_of\_two\_equal\_circles} &= 2 \times (area_{ACFD} - area_{ACD}) \quad \text{Eq. 2.19} \\ &= 2 \left( R^2 \cos^{-1} \frac{d}{2R} - \frac{d}{4} \sqrt{4R^2 - d^2} \right) \\ &= 2R^2 \cos^{-1} \frac{d}{2R} - \frac{d}{2} \sqrt{4R^2 - d^2} \end{aligned}$$

From Eq. 2.19, the ratio of the area of the overlapping region of the two circles to the area jointly covered by the two circles is given as follows:

$$\begin{aligned} &\frac{area_{overlapping\_of\_two\_equal\_circles}}{area_{union\_of\_two\_equal\_circles}} \quad \text{Eq. 2.20} \\ &= \frac{(2R^2 \cos^{-1} \frac{d}{2R} - \frac{d}{2} \sqrt{4R^2 - d^2})}{(\pi R^2 + (\pi R^2 - (2R^2 \cos^{-1} \frac{d}{2R} - \frac{d}{2} \sqrt{4R^2 - d^2})))} \\ &\approx overlap\_ratio \end{aligned}$$

In Eq. 2.20, we let the *overlap\_ratio* to be approximately equal to the ratio of the area of the overlapping region of the two circles to the area of the union of the two circles, which is determined only by the distance between the centers of the two circles  $d$ , and the node transmission range  $R$ . Hence, to determine the relationship between *overlap\_ratio* and  $d$ , we need to express  $d$  in terms of *overlap\_ratio*. We first plot

the relationship between  $d$  and  $overlap\_ratio$  by solving Eq. 2.21 using the bisection root finding numerical method by setting a constant value for  $R$  (250 meters is assumed in our work) and varying the value of  $k$ .

$$\frac{(2R^2 \cos^{-1} \frac{d}{2R} - \frac{d}{2} \sqrt{4R^2 - d^2})}{(\pi R^2 + (\pi R^2 - (2R^2 \cos^{-1} \frac{d}{2R} - \frac{d}{2} \sqrt{4R^2 - d^2})))} - k = 0 \quad \text{Eq. 2.21}$$

The graph of the relationship between  $d$  and  $overlap\_ratio$  is plotted in Figure 2.8. Using the curve fitting method with a polynomial of degree two, the relationship between link length and overlap ratio is as shown by the equation in Figure 2.8. This equation can be used to estimate the length of a link given the overlap ratio as computed using Eq. 2.13. The value of the overlap ratio is generally in the range of approximately 0.25 to 1.00. For values smaller than that, we simply set the estimated length of the considered link to be 250 meters.

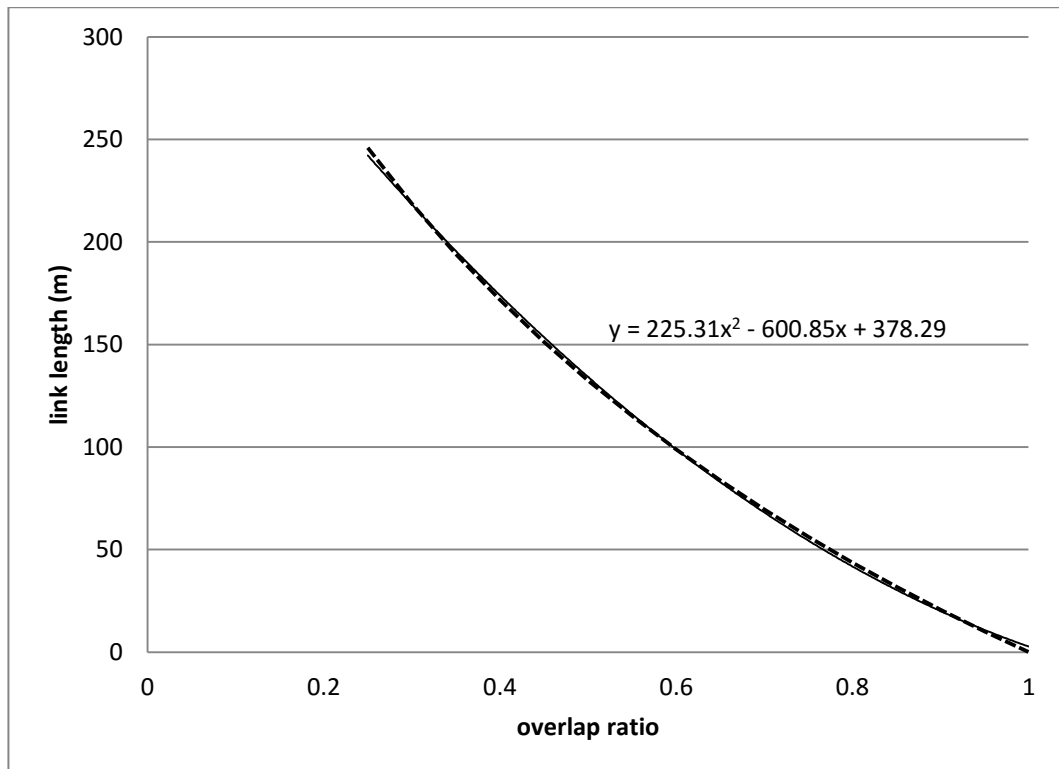


Figure 2.8: The plot of the relationship between link length and overlap ratio

The accuracy of the estimated link lengths is governed by how closely the considered regions in a real scenario agree on the assumptions made when deriving the



relationship between the overlap ratio and the link length: (1) the uniformity of the node distribution in the considered regions, and (2) the node densities in the considered regions. When node distribution is more uniform or node density is higher, the accuracy of the estimations will be higher. We also identified stale neighborhood information as another source of estimation inaccuracy. For example, suppose node  $y$  was previously a neighbor of node  $x$  but has moved out of transmission range from node  $x$  but the neighbor list entry of node  $y$  in node  $x$ 's neighbor list has not expired yet. Hence, node  $x$  still regards node  $y$  as its neighbor. A similar scenario is encountered when a new node  $z$  has moved within transmission range from node  $x$  but has not broadcast a new HELLO message. In this case, node  $x$  does not yet recognize node  $z$  as its neighbor.

When estimated link length values are used in place of actual link length values when computing the metric values of paths using the RS1 routing metric, we actually transformed the routing metric from one that considers link length to another that considers the ratio of the number of nodes in the intersection of the transmission coverage areas of two nodes forming a link to the number of nodes in the union of the transmission coverage areas of the two nodes. As described above, a source of link length estimation inaccuracy is in how closely the assumptions made are followed in actual scenarios. However, our routing metrics continue to work well even if the assumptions are only followed loosely, albeit with lower performance.

### **2.3.6 Estimating Node Mobility**

In Section 2.3.4, we proposed the RS2 routing metric, which considers link length and node mobility. A method to estimate the length of a link was given in Section 2.3.5. However, without relying on sensors, it is difficult to measure node mobility. Fortunately, it turns out that the method proposed in Section 2.3.5 to estimate the length of the link between two neighboring nodes can also be used to estimate the amount of distance a node has moved from its previous location with some modifications. Instead

of computing *overlap\_ratio* using the neighbor sets of two neighboring nodes using Eq. 2.13, we compute *overlap\_ratio* using the current and previous neighbor sets of a node, as shown in Eq. 2.22.

$$overlap\_ratio = \frac{|N_x \cap N_x^p|}{|N_x \cup N_x^p|} \quad \text{Eq. 2.22}$$

Figure 2.9 shows the difference between the two different operations: (1) estimating the length of the link between two neighboring nodes, and (2) estimating the amount of distance a node has moved from its previous location.

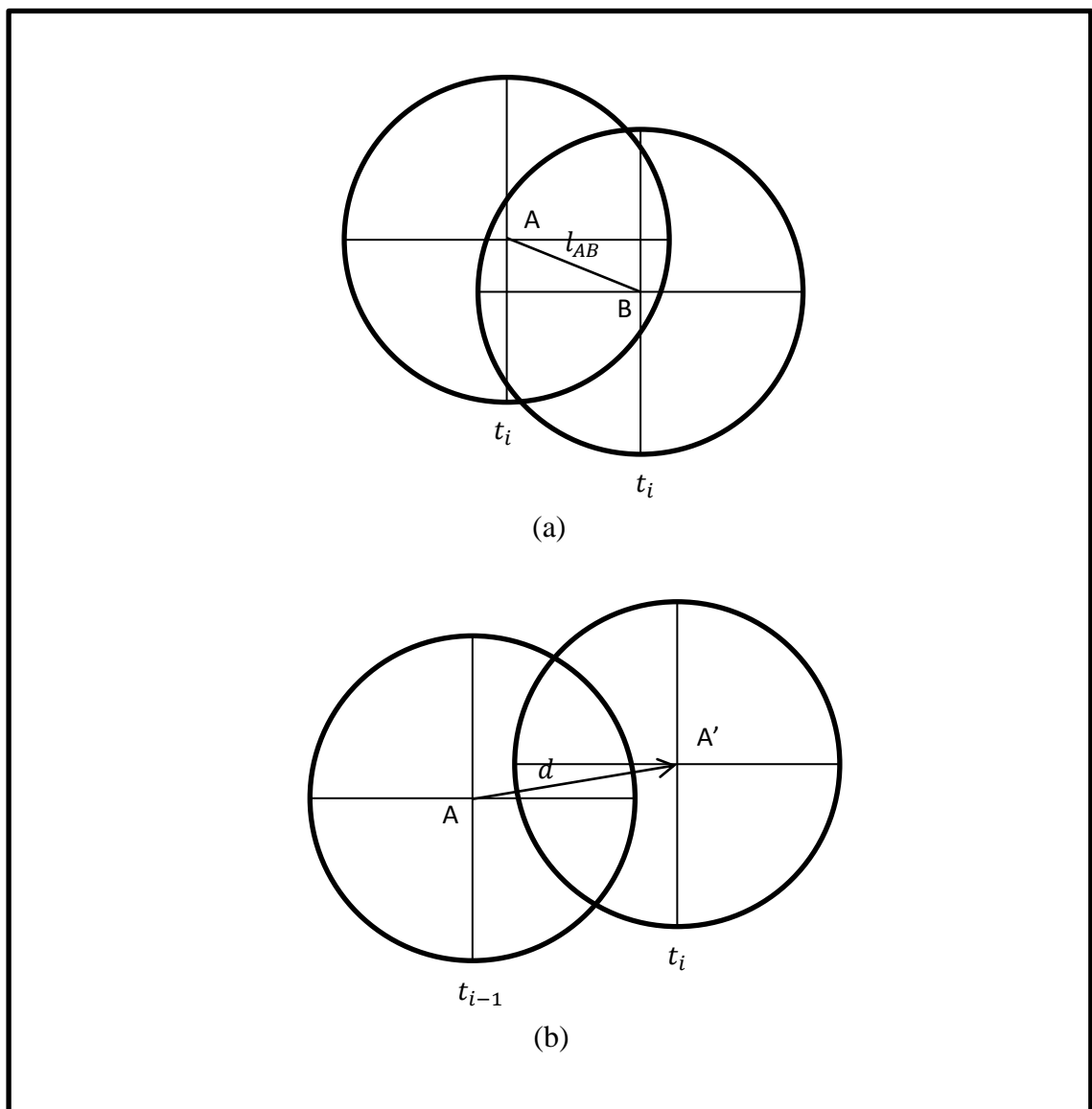


Figure 2.9: (a) estimating the length of a link, (b) estimating the distance a node has moved from its previous location

Generally, the overlap ratio in Eq. 2.22 measures the degree of change in the neighbor set of a node between two successive sampling times. The change in the neighbor set of a node can be viewed as one of the following scenarios: (1) the node is moving but its neighbors are static, (2) the node is static but its neighbors are moving, or (3) both the node and its neighbors are moving. In each and every scenario, a higher change in the neighbor set of a node signifies higher relative velocities between the node and its neighbors. The change in neighbor set of a node is also used in many other routing metrics to measure node mobility, for example, Mobility Factor (MF) (Wu et al., 2009) and Neighbor Change Ratio (NCR) (Dutkiewicz, 2006). The amount of distance a node has moved from its previous location is then used as the measure of its mobility ( $mobility_n$ ) in Eq. 2.11.

### 2.3.7 Packet Header Modification

In the proposed routing metrics, if true link length information is used, the location information of a transmitting node is required. Hence, when true link length information is used, a RREQ and a RREP each is extended by three fields of type float to store the x- and y-coordinates of a transmitting node, and the aggregate metric value of a path. The aggregate metric value of a path is initialized to 0 at the RREQ source for a RREQ, and initialized to 0 at the RREQ destination for a RREP.

If estimated link length values are used in place of actual link length values as proposed in Section 2.3.5, a RREQ and a RREP each is extended by one field of type float, and  $m$  fields each 1 byte sized. Like before, the field of type float is used for storing the aggregate metric value of a path and is initialized to 0 at the RREQ source for a RREQ, and initialized to 0 at the RREQ destination for a RREP. To estimate the length of the link between a node and its previous hop, the node requires the neighbor set of its previous hop. To store all of the addresses of the neighbors of a node using conventional methods incurs high packet overhead. To resolve this problem, we propose

a method to store node addresses in a compact manner. We line up  $m$  bytes contiguously and encode whether a node is a neighbor of a considered node using a binary value, as shown in Figure 2.10. The number of bytes required with this method depends on the number of nodes in the network. For example, for a network consisting of 200 nodes,  $m = 200/8 = 25$ , which is arguably quite manageable compared to 200 bytes if node addresses are stored using 1-byte fields.

1 <sup>st</sup> byte	1	2	3	4	5	6	7	8
2 <sup>nd</sup> byte	9	10	11	12	13	14	15	16
.								
.								
m <sup>th</sup> byte	$1+(m-1)8$	$2+(m-1)8$	$3+(m-1)8$	$4+(m-1)8$	$5+(m-1)8$	$6+(m-1)8$	$7+(m-1)8$	$8+(m-1)8$

Figure 2.10: Storing neighbor set information of a node compactly in RREQs and RREPs

As opposed to estimating the length of the link between two neighboring nodes, estimating the amount of distance a node has moved from its previous location requires no additional packet overhead. This is because the overlap ratio is computed using only local information, i.e., the neighbor set and the previous neighbor set of a node.

## 2.4 Simulation Studies

We evaluated the proposed routing metrics using network simulator 2 (ns-2) (“The Network Simulator - ns-2,” n.d.). We placed 75 nodes in a rectangular region of dimension 1500 meters by 300 meters (same setup as used in (Son et al., 2014)). The network traffic consists of five pairs of CBR traffic flows, each flowing at the rate of 40 Kibps (512 bytes packets at the rate of 10 packets/s;  $1\text{ Ki} = 2^{10} = 1024$ ,  $1\text{ K} = 10^3 = 1000$ ) and starting at a random time in  $[0, 20]$  seconds simulation time. The physical and MAC related parameters were set to emulate the IEEE 802.11 ERP-DSSS (Vassis, Kormentzas, Rouskas, & Maglogiannis, 2005)(Villaseñor-González, 2007) physical layer. Nodes were given a transmission range  $R$  of 250 meters and a carrier sensing range of 550 meters. The packet overhead needed as described in Section 2.3.7 was taken into account in the simulations for a fair and accurate comparison. The node mobility model used is the Random Waypoint Model. We varied the degree of node mobility by adjusting the maximum node speed from 5 m/s to 25 m/s in increments of 5 m/s while the pause time was set to 0 seconds so that nodes were constantly moving. For a particular maximum node speed value, we performed 20 simulation runs, each run using a different seed number in the interval  $[1, 20]$  when generating the node mobility and network traffic patterns. The metrics used for comparing the performance of the various methods are as follows:

1. *Packet delivery ratio (%)*: the number of data packets that were successfully delivered divided by the number of data packets sent by all sources
2. *Normalized routing load*: the number of transmissions of all routing control packets (RREQ, RREP, RERR, and HELLO) divided by the number of data packets that were successfully delivered. It measures the average number of transmissions required for routing control packets for every data packet successfully delivered.

3. *Average packet latency (milliseconds)*: the average of the end-to-end delays of data packets that were successfully delivered
4. *Average hop count*: the average hop count of data packets that were successfully delivered
5. *Number of route discoveries*: The number of route discoveries is a measure of the stability of the discovered routes as more route discoveries are needed when they are more route breakages. In AODV, a route discovery is uniquely identified by a <source, broadcast ID> pair.

#### **2.4.1 Shorter Links or Links with Higher LRLs?**

We seek to answer the following two questions. First, is the path with the highest remaining lifetime the most stable path? Second, can paths with shorter links perform better than paths with higher estimated remaining lifetimes? To answer these questions, we compared the LRL method with the SZ/LRL method. For the LRL method, we modified the AODV (Perkins & Royer, 1999) routing protocol to use the LRL routing metric. For the SZ/LRL method, we modified the AODV routing protocol to use the Safety Zone (SZ)-based route discovery method (Kim, 2001) and also the LRL routing metric. From the viewpoint of path stability alone, the LRL method should provide paths that are more optimal because in the SZ/LRL method, some RREQs are dropped during route discoveries causing paths with links that are long but have high LRLs not being discovered. The threshold distance for dropping RREQs was set to 200 meters for the SZ/LRL method.

We found that the SZ/LRL method outperforms the LRL method. The SZ/LRL method obtained higher packet delivery ratio (Figure 2.11a) and lower normalized routing load (Figure 2.11b). This is because the routes established by the SZ/LRL method were considerably more stable as can be verified by the number of route discoveries produced (Figure 2.11e). Compared to the LRL method, the SZ/LRL

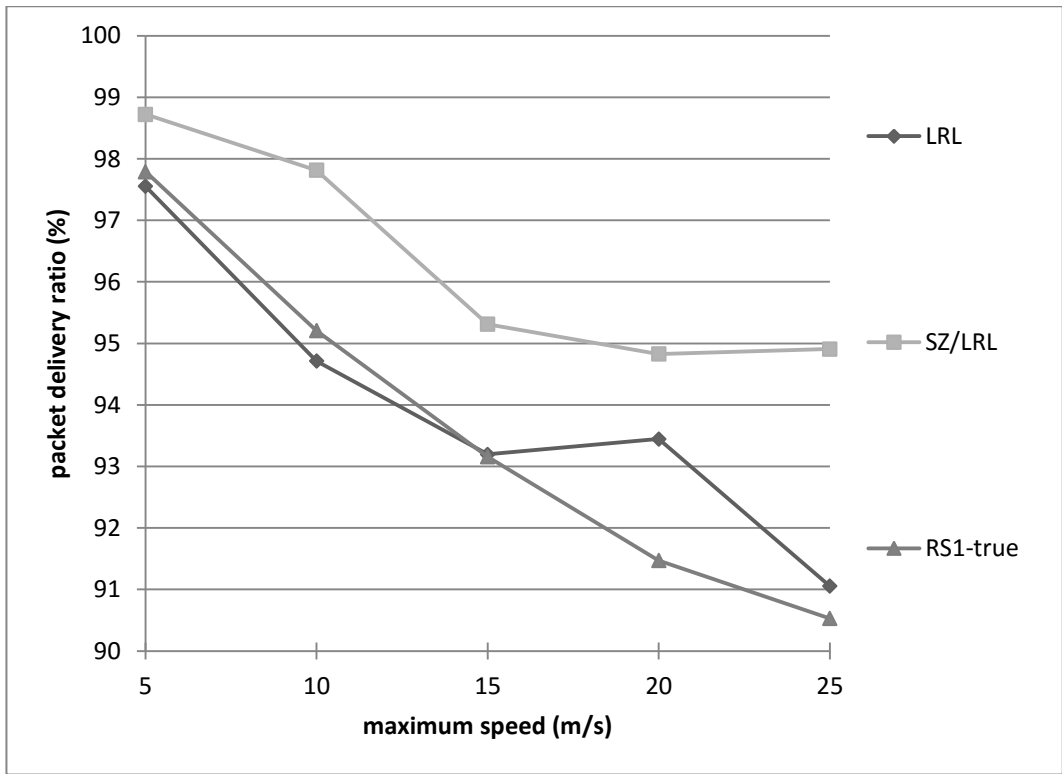
method produced approximately 14.5%, 9%, 12.5%, 16.4%, and 13.2% fewer route discoveries at 5, 10, 15, 20, and 25 m/s maximum node speed, respectively. Fewer route discoveries mean that the established routes were more stable. The average hop count of delivered packets is higher with the SZ/LRL method (Figure 2.11d) because in order to discover paths consisting of stable links, in a route discovery, nodes that exceed a certain threshold distance from their previous hops drop their received RREQs. In terms of the average packet latency (Figure 2.11c), there is no clear winner as the SZ/LRL method outperformed the LRL method at 5, 10, and 25 m/s maximum node speeds while the reverse is true at 15 and 20 m/s maximum node speeds.

There is no doubt that the LRL routing metric can discover and establish stable routes. However, it requires the use of certain information that can be obtained only from sensors. We then ask the following question. Is it possible to obtain good network performance without using node velocity information? If we could refrain from using node velocity information, then route stability could be improved without using sensors. To answer this question, we compared the RS1-true method against the LRL method. For the RS1-true method, we modified the AODV routing protocol to use the RS1 routing metric. In this experiment, true link length information was used in computing the RS1 values, hence the “-true” suffix at the end of the name of the method (Note: Previously we mentioned about not using sensors. However, in this test, using true link length values for calculating RS1 values indicate using sensors. We do this only to investigate the feasibility of the method). The threshold link length value was set to  $0.7R=175$  meters in order to balance the tradeoff between path remaining lifetime and hop count. From Figure 2.4, it can be observed that links that are 175 meters long achieved an average remaining lifetime of approximately 60% of the average remaining lifetime of links that are 0 meters long. Due to diminishing returns with lower link length threshold values, we avoided setting the link length threshold to a value that is

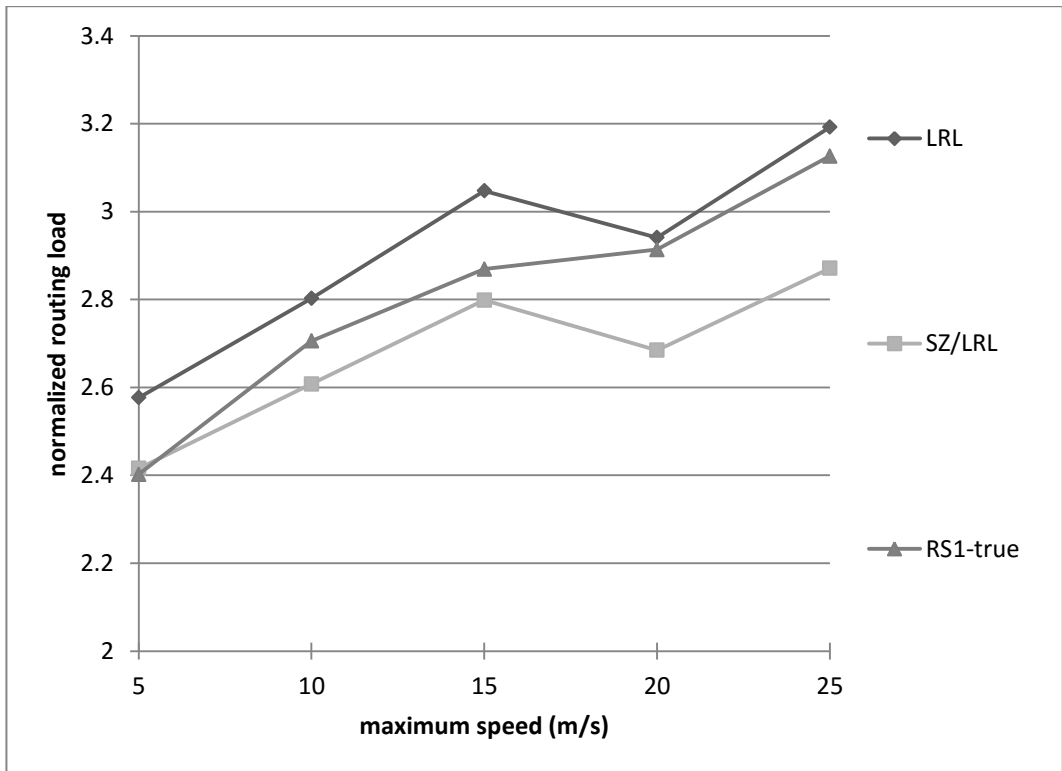
too small as this could result in paths with very high hop count and very short links.

We found that the RS1-true method performed quite closely to the LRL method even when the former uses less information, i.e., not using node velocity information. However, the RS1-true method was outperformed by the SZ/LRL method in terms of packet delivery ratio (Figure 2.11a). When comparing the normalized routing load (Figure 2.11b), the RS1-true method is in between the LRL and SZ/LRL methods. The SZ/LRL method produced the lowest normalized routing load because some RREQs were dropped during route discoveries. The RS1-true method produced the lowest average packet latency as can be seen in Figure 2.11c. This is attributed to the lower average hop count of packets when the RS1 routing metric was used (Figure 2.11d). The stability of the routes discovered by the RS1-true method is the best at lower degrees of node mobility (5-15 m/s). However, at higher degrees of node mobility, the SZ/LRL method outperformed the RS1-true method, as shown in Figure 2.11e. From this comparison, we found that the RS1 routing metric does a good job at approximating and sometimes even outperforms the path remaining lifetime metric (the LRL method) at lower degrees of node mobility (5-15 m/s) despite using less information. From this experiment, it also seems that the RS1 routing metric may be further improved by also considering node mobility (speed).

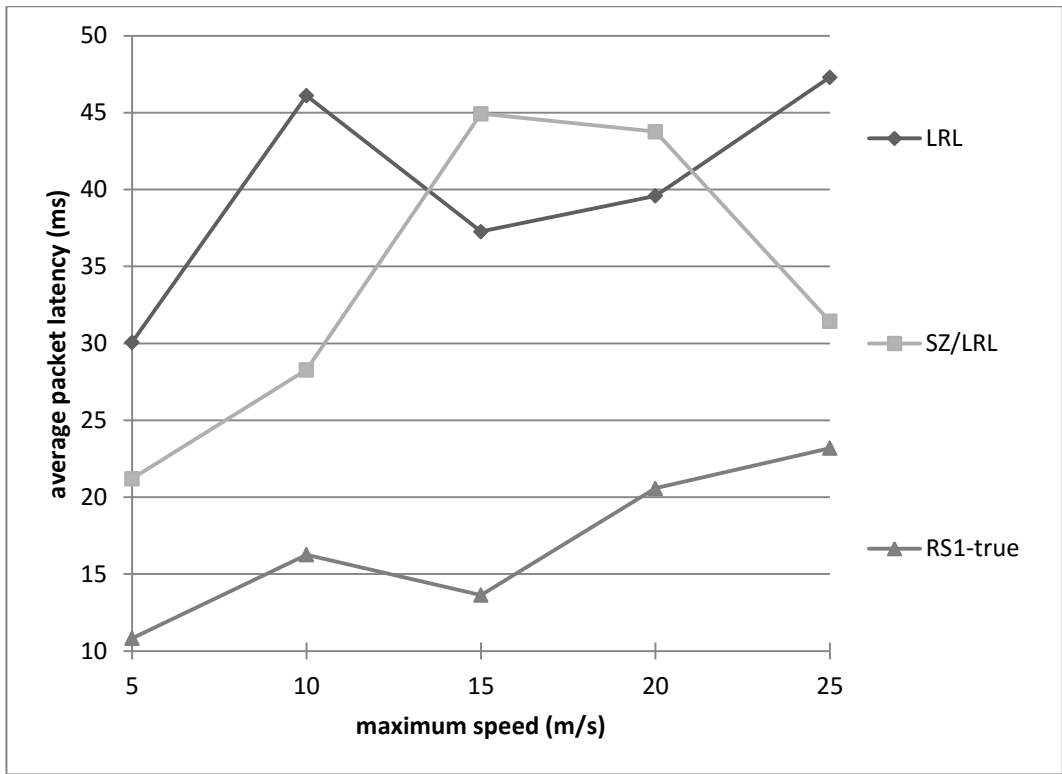




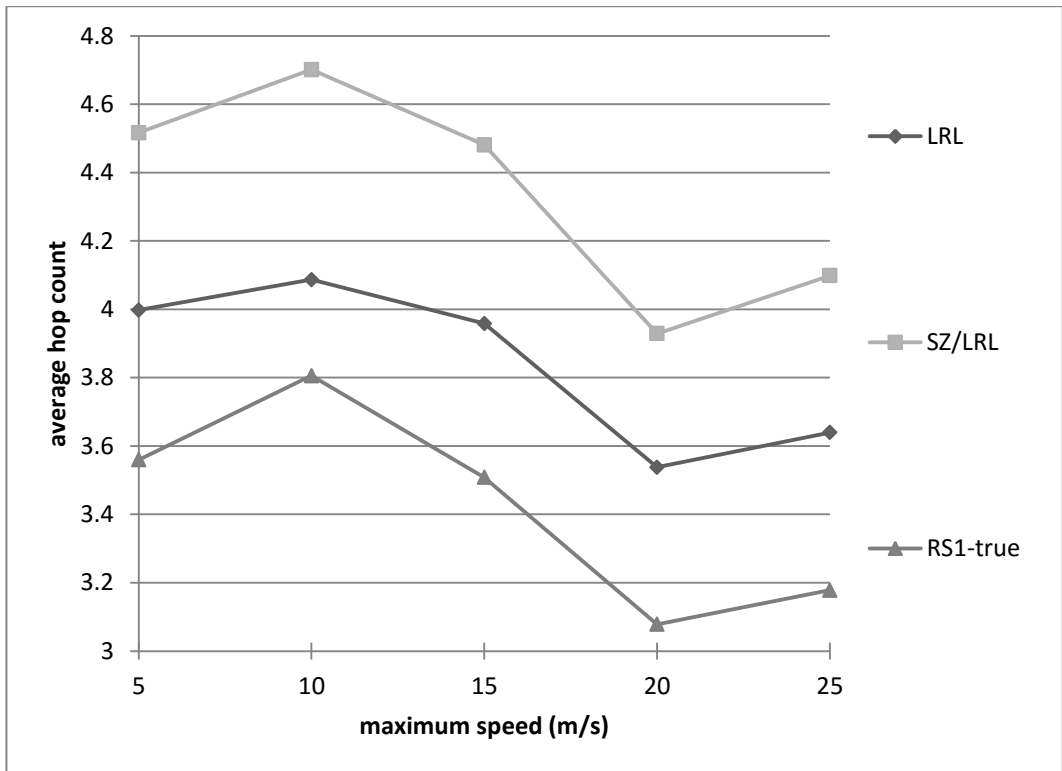
(a)



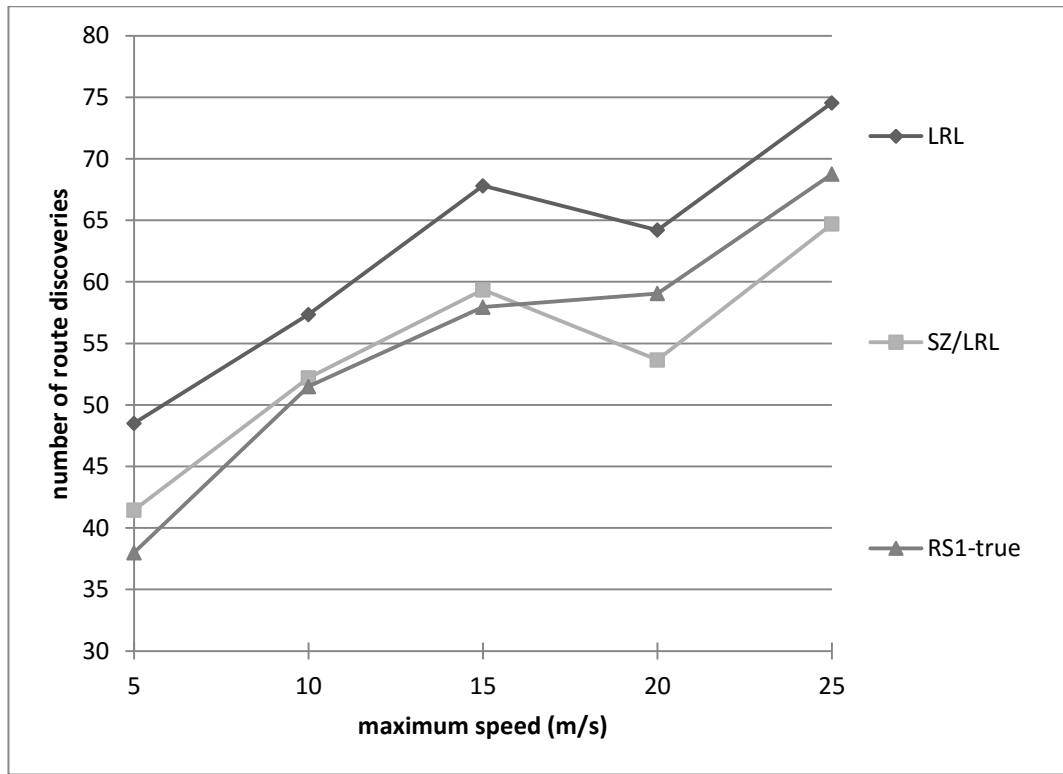
(b)



(c)



(d)



(e)

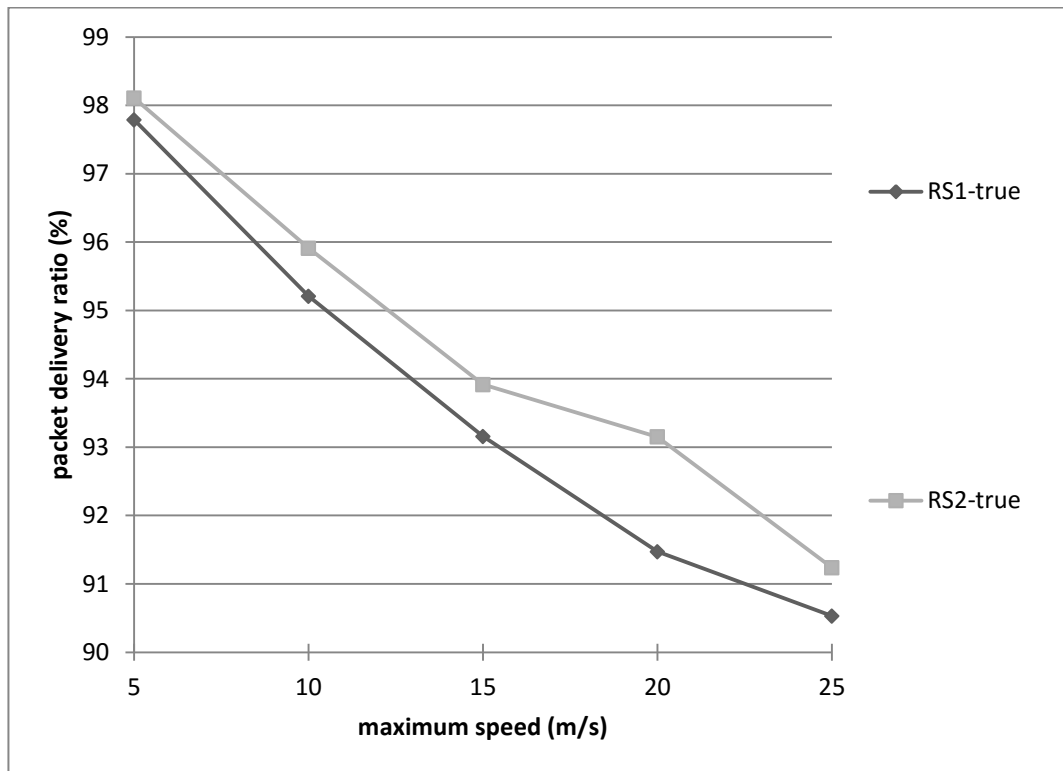
Figure 2.11: Comparing RS1-true, SZ/LRL, and LRL: (a) packet delivery ratio, (b) normalized routing load, (c) average packet latency, (d) average hop count, (e) number of route discoveries

#### 2.4.2 Comparing RS1 and RS2

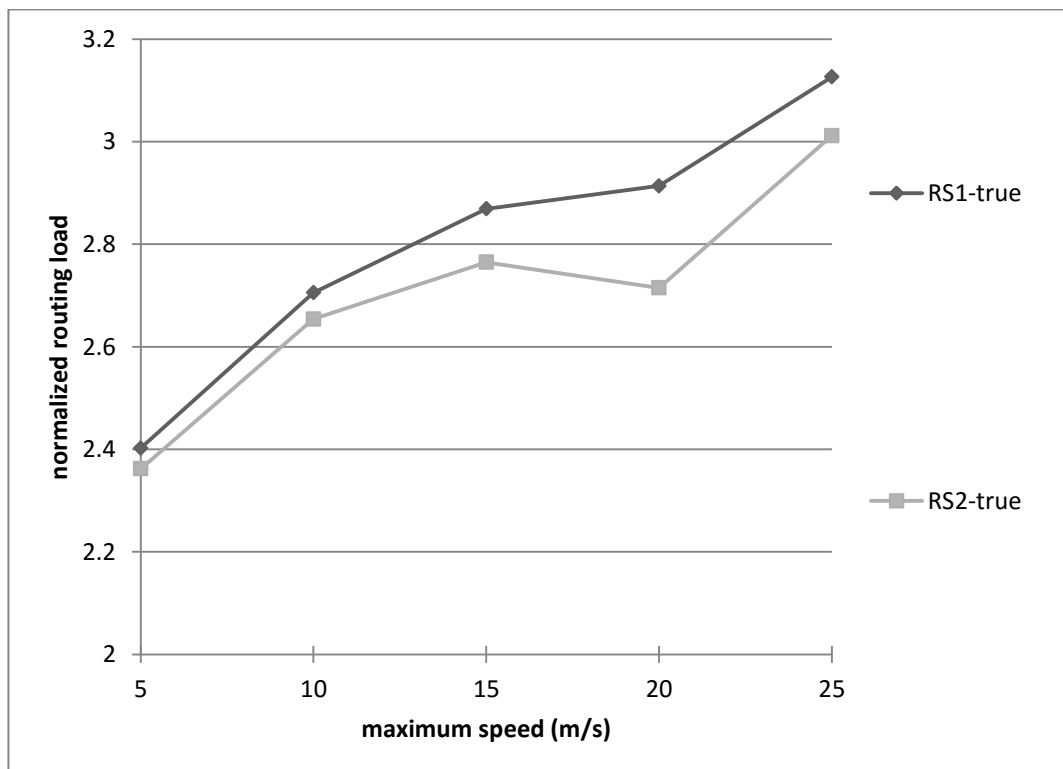
The RS1 routing metric takes into account only link length while the RS2 routing metric takes into account both link length and node movement speed. In this section, we compare these two routing metrics. True link length values were used in both methods and the threshold link length value used in both methods is  $0.7R=175$  meters. The purpose of this experiment is to investigate if higher route stability can be obtained by also considering node movement speed in addition to link length.

We found that the RS2-true method outperforms the RS1-true method due to the use of extra information, i.e., node movement speed. The RS2-true method obtained higher packet delivery ratio (Figure 2.12a), lower normalized routing load (Figure 2.12b), lower average packet latency (Figure 2.12c), and lower number of route

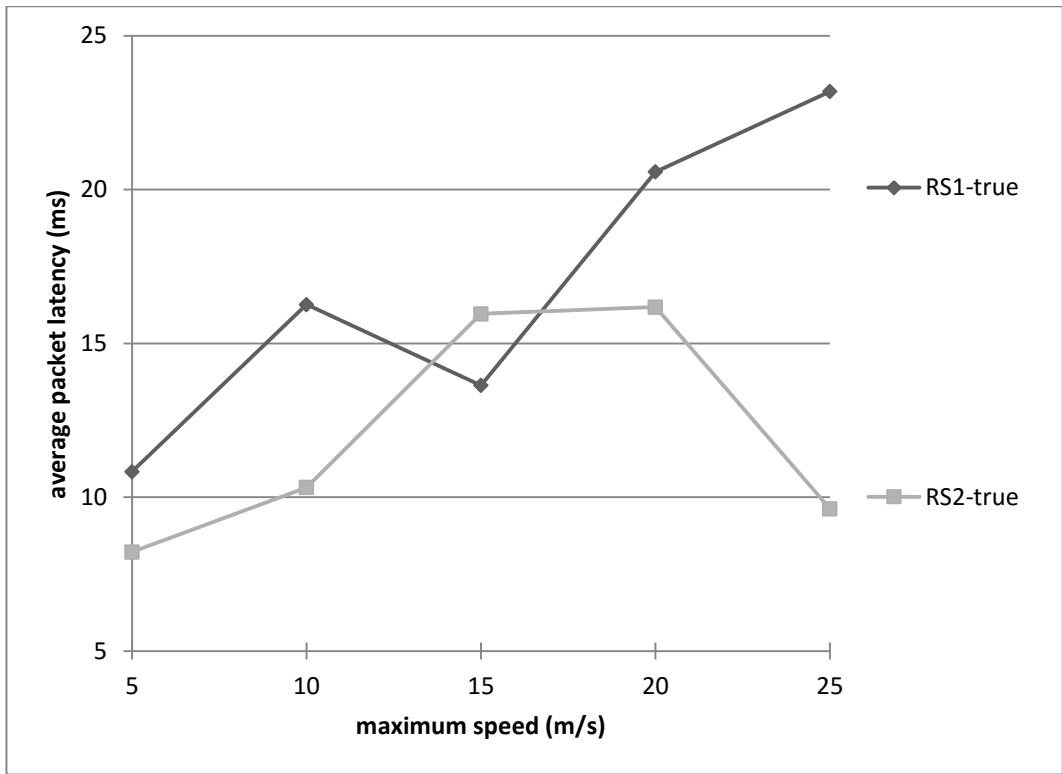
discoveries (Figure 2.12e). However, there is little difference between the average hop counts of delivered packets obtained by the two methods, as shown in Figure 2.12d.



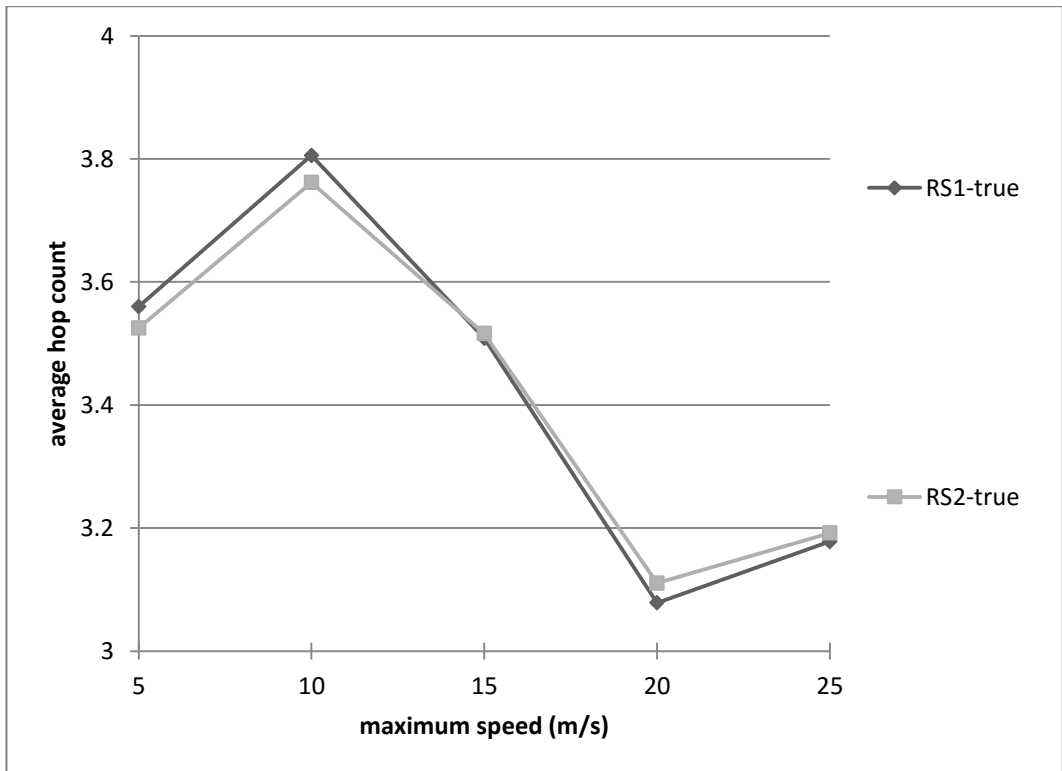
(a)



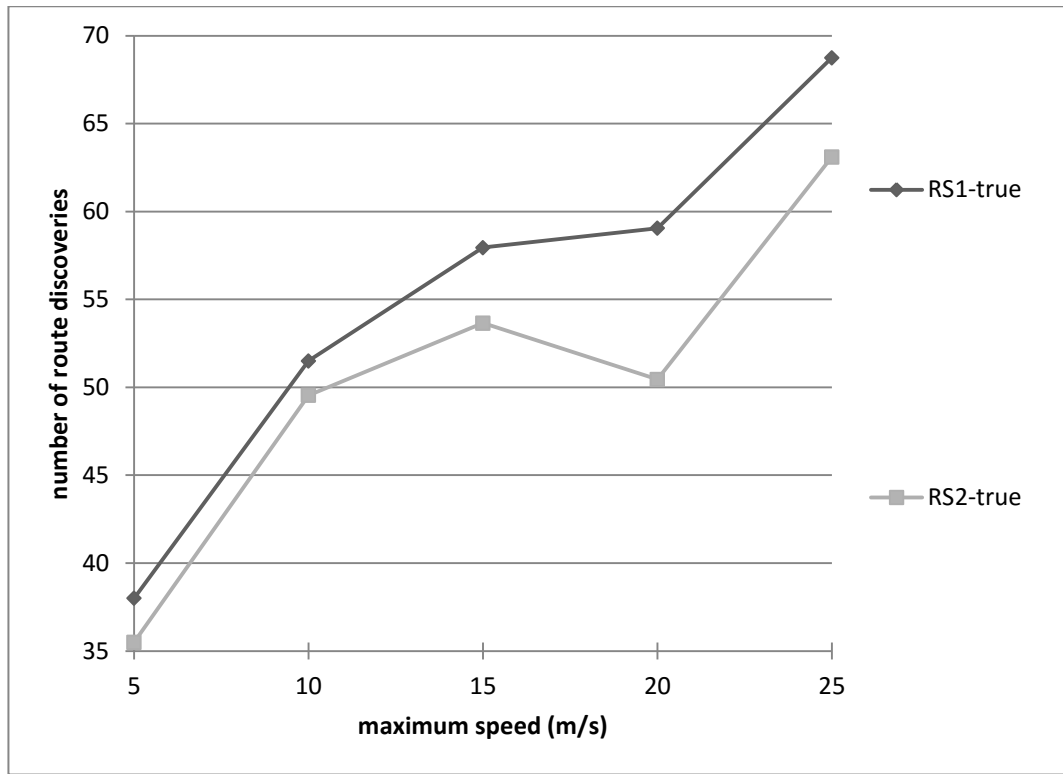
(b)



(c)



(d)



(e)

Figure 2.12: Comparing RS1-true and RS2-true: (a) packet delivery ratio, (b) normalized routing load, (c) average packet latency, (d) average hop count, (e) number of route discoveries

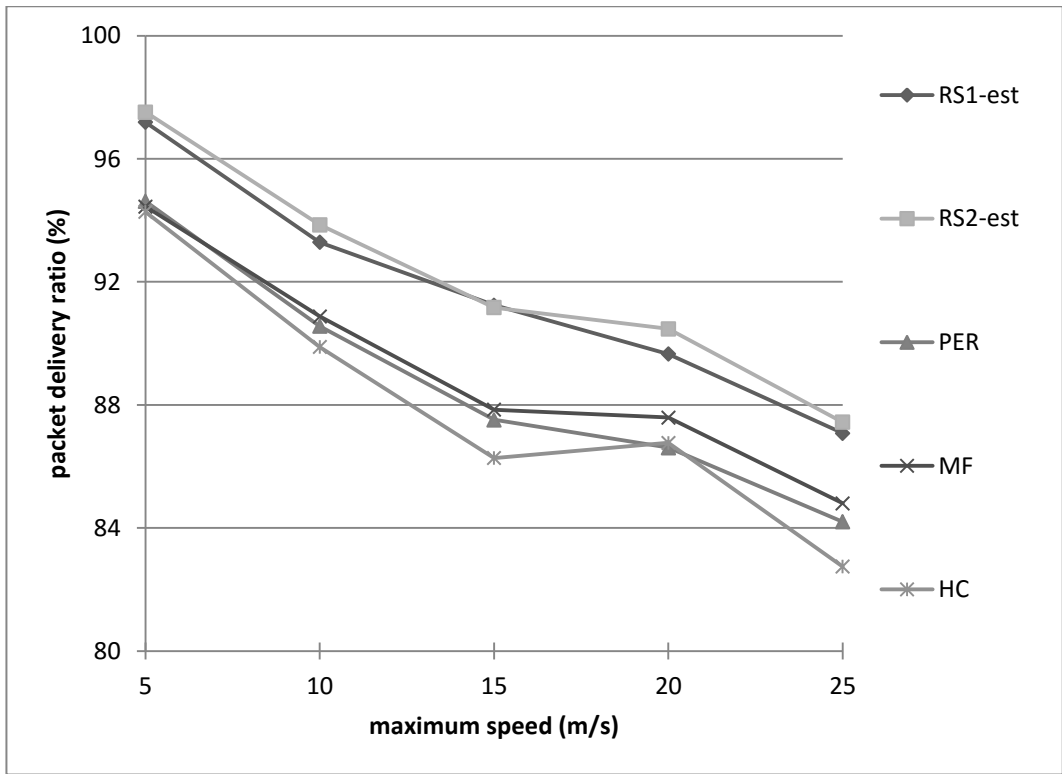
### 2.4.3 Comparing the Sensor-free Methods

From Sections 2.4.1 and 2.4.2, we found that the RS1 and RS2 routing metrics to be feasible and represent a promising solution for improving route stability without using node velocity information. In this section, we compare the two routing metrics against other routing metrics, i.e., Path Encounter Rate (PER) (Son et al., 2014), Mobility Factor (MF) (Wu et al., 2009), and Hop Count (HC). The link length and/or node velocity values used to compute the RS1 and RS2 routing metrics were estimated, hence the “-est” suffix at the end of the names of the methods using these routing metrics. The PER, MF, and HC routing metrics do not require information from sensors; hence, all the routing metrics compared in this test are sensor-free methods. The threshold link length value used for the RS1-est and RS2-est methods is  $0.5R=125$

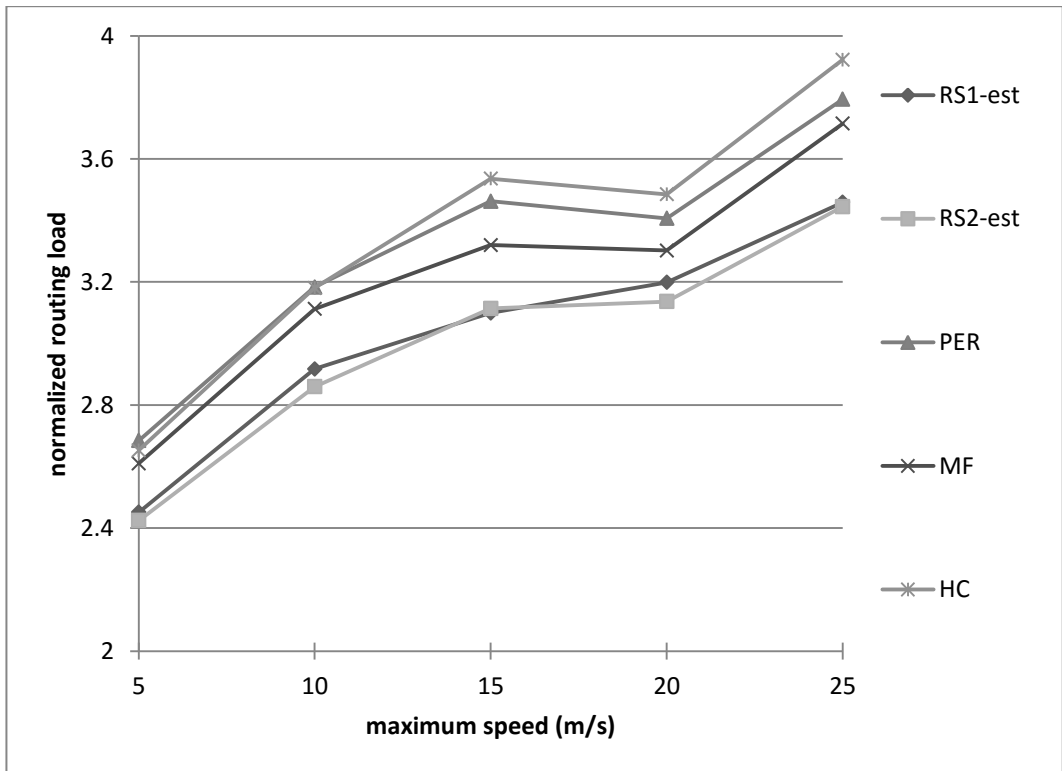
meters.

Similar to when true information was used (Section 2.4.2), we found that the RS2-est method outperforms the RS1-est method slightly. Furthermore, both outperformed the PER, MF, and HC methods in terms of packet delivery ratio (Figure 2.13a), normalized routing load (Figure 2.13b), and average packet latency (Figure 2.13c). The RS1-est method produced approximately 3.1%, 3.8%, 5.8%, 3.3%, and 5.2% higher packet delivery ratio than the HC method at 5, 10, 15, 20, and 25 m/s maximum node speed, respectively. In contrast, the PER method produced approximately 0.4%, 0.7%, 1.4%, -0.2%, and 1.8% higher packer delivery ratio than the HC method at 5, 10, 15, 20, and 25 m/s maximum node speed, respectively.

The routes discovered by the RS1-est and RS2-est methods were more stable than the routes discovered by the PER, MF, and HC methods. The RS1-est method produced approximately 25.1%, 19.9%, 24.0%, 19.2%, and 22.5% fewer route discoveries than the HC method at 5, 10, 15, 20, and 25 m/s maximum node speed, respectively. On the contrary, the PER method produced approximately -1.3%, 0%, 4.8%, 7.5%, and 4.7% fewer route discoveries than the HC method at 5, 10, 15, 20, and 25 m/s maximum node speed, respectively. The RS1-est and RS2-est methods obtained better average packet latency (Figure 2.13c), even when the average hop counts of delivered packets obtained by these methods are higher than those obtained by the PER, MF, and HC methods (Figure 2.13d). This is because the routes established by the RS1-est and RS2-est methods were more stable (Figure 2.13e). One of the factors contributing to packet latency is route stability. When a route breaks, data transmission for that route is interrupted and a new route discovery is performed. Data transmission for that route can only be resumed when the broken path is replaced. When routes are unstable, data transmission is frequently interrupted and this causes data packets to have high latencies.

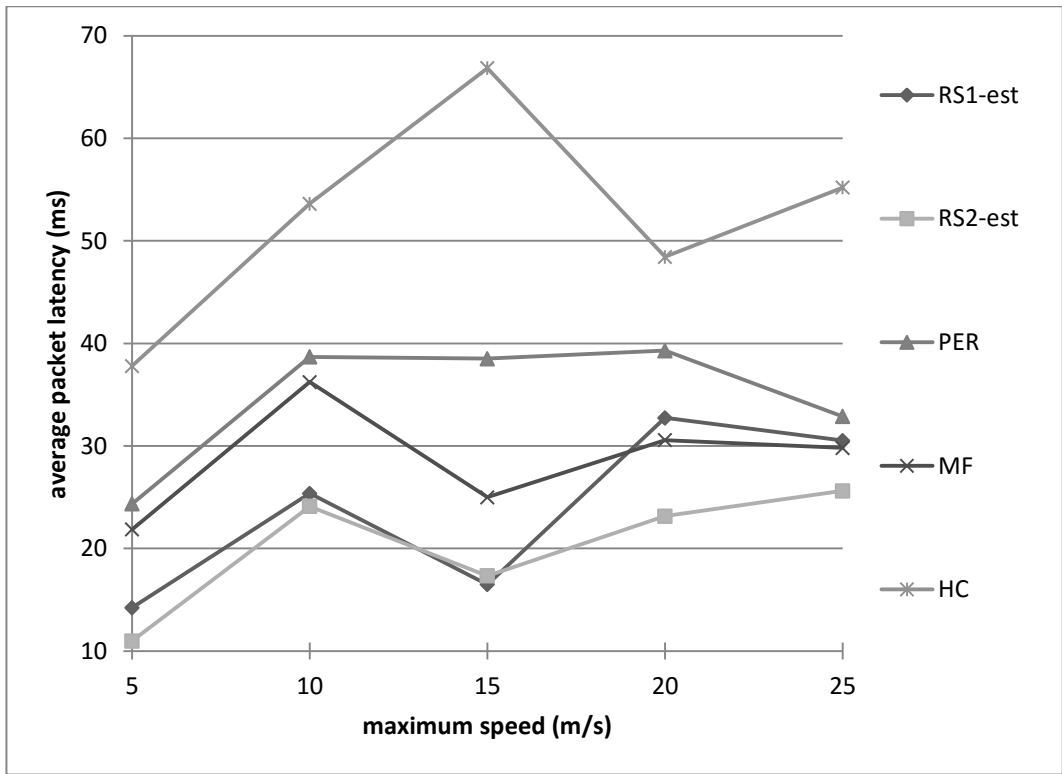


(a)

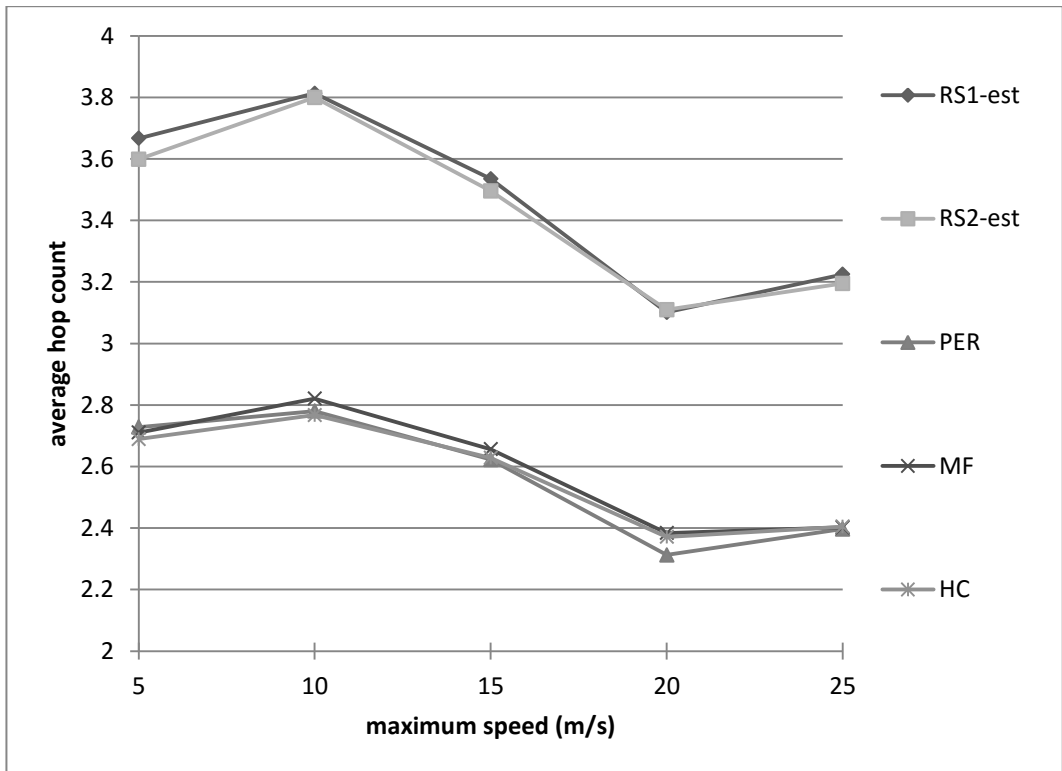


(b)

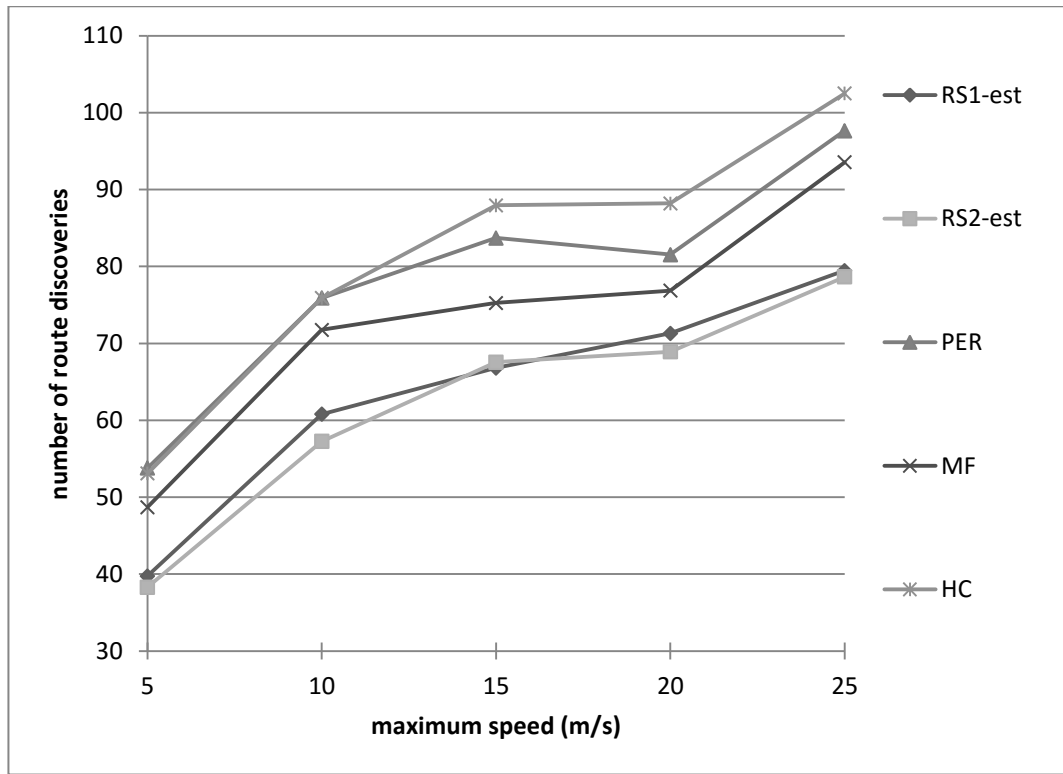




(c)



(d)



(e)

Figure 2.13: Comparing the sensor-free methods: (a) packet delivery ratio, (b) normalized routing load, (c) average packet latency, (d) average hop count, (e) number of route discoveries

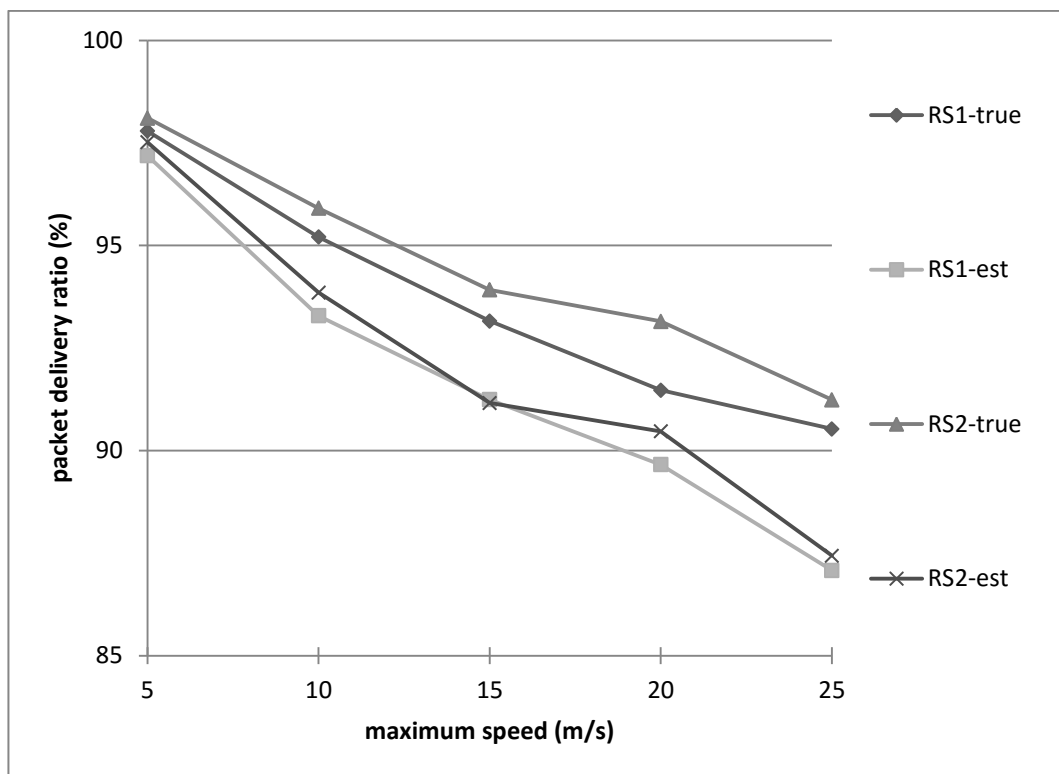
#### 2.4.4 True vs. Estimated Information

RS1-est and RS2-est are the sensor-free version of RS1-true and RS2-true, respectively. The link length threshold values used are: (1)  $0.7R=175$  meters for RS1-true and RS2-true, and (2)  $0.5R=125$  meters for RS1-est and RS2-est. A lower threshold link length value is used for the sensor-free methods as they generally perform worse than their sensor-using counterparts due to the use of less accurate information.

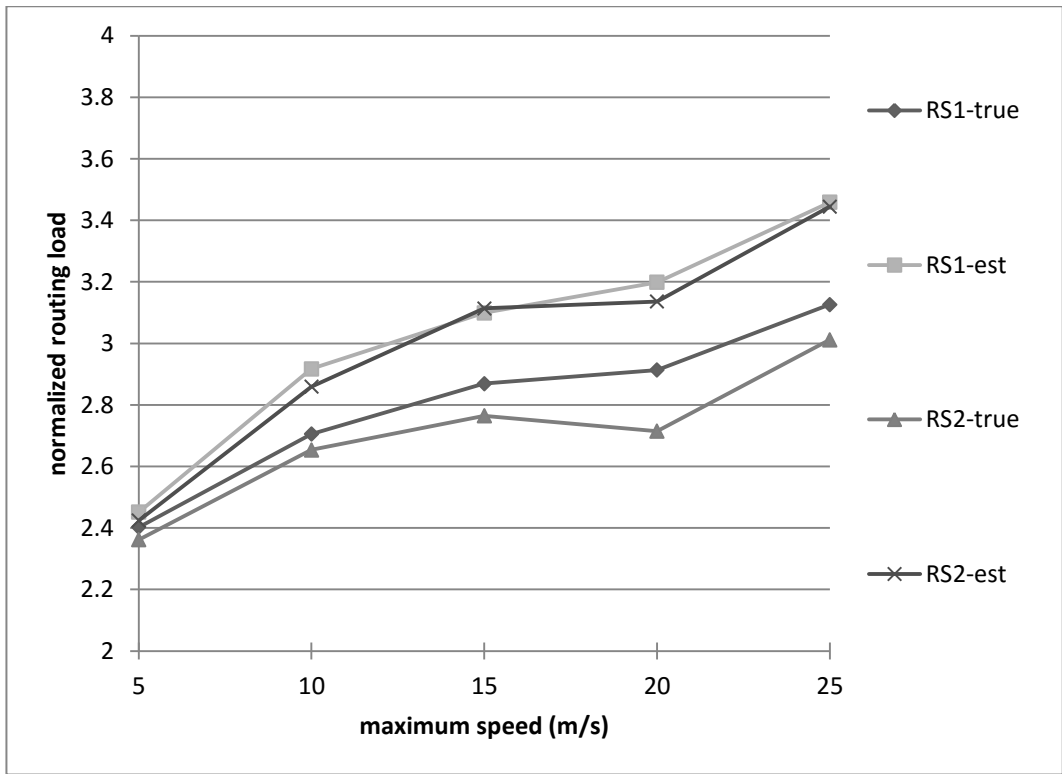
The sensor-free methods performed worse than their sensor-using counterparts in almost all aspects, i.e., packet delivery ratio (Figure 2.14a), normalized routing load (Figure 2.14b), and average packet latency (Figure 2.14c). This is due to the generally more stable routes found by the sensor-using methods (Figure 2.14e). The average hop counts of delivered packets (Figure 2.14d) obtained by all methods however are quite

similar to each other.

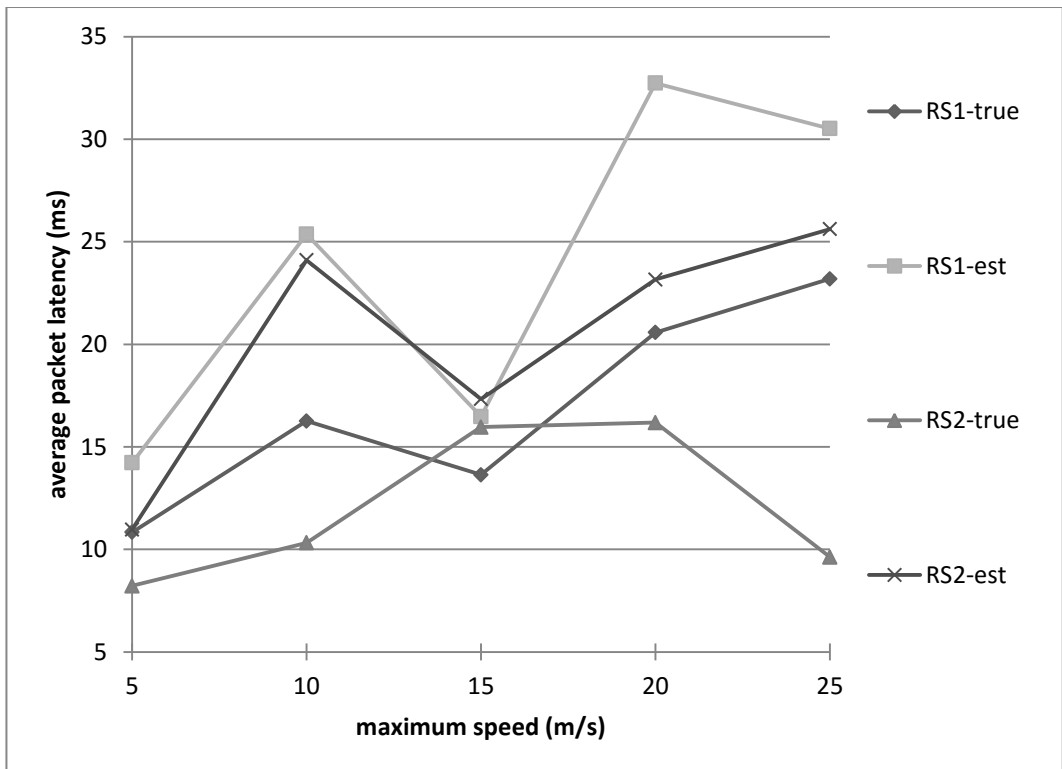
It turns out that the sensor-free methods generally perform worse and cannot replace their sensor-using counterparts. This is to be expected considering that the sensor-free methods use less accurate estimated link length information and node velocity information. However, we argue that this is acceptable considering that cost is reduced when sensors are not used. Besides, from Section 2.4.3, we also noted that the RS1-est and RS2-est methods outperformed the MF, PER, and HC methods. The gains of our sensor-free methods over the HC method are respectable. In contrast, the PER method provides only negligible gains over the HC method.



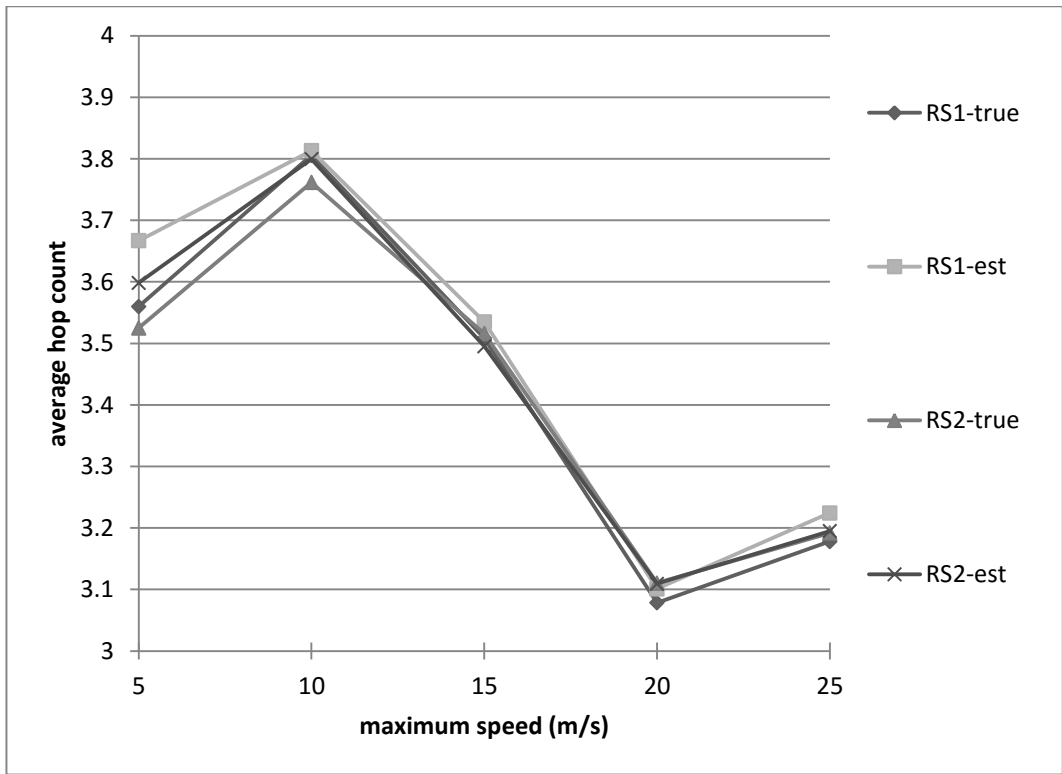
(a)



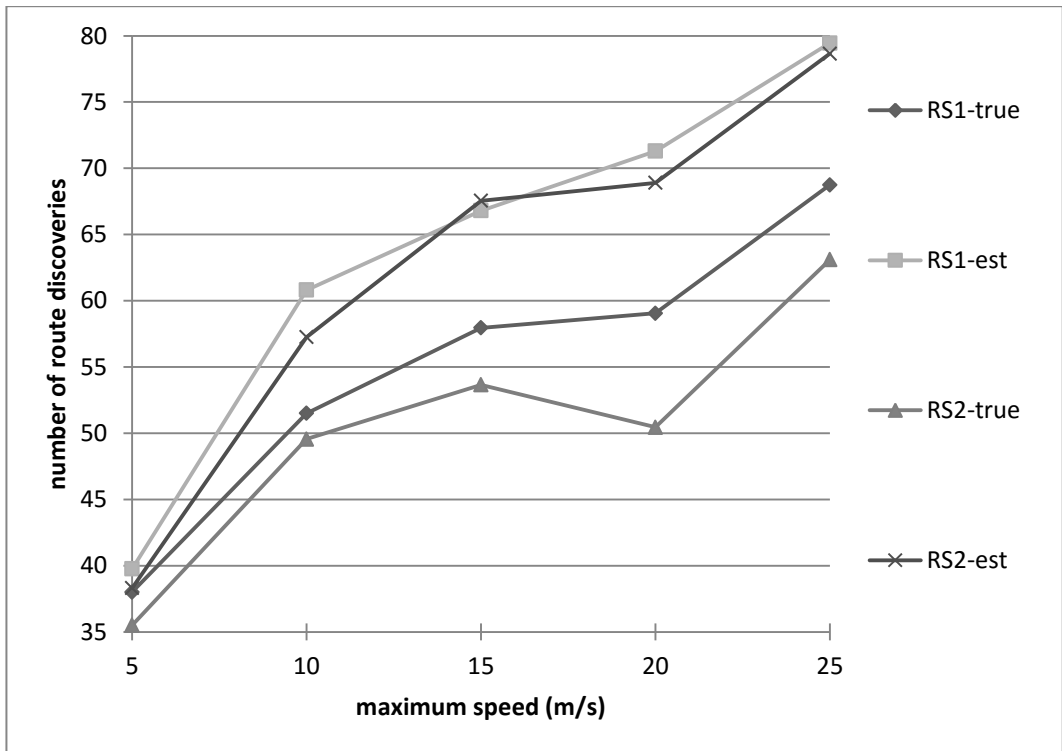
(b)



(c)



(d)



(e)

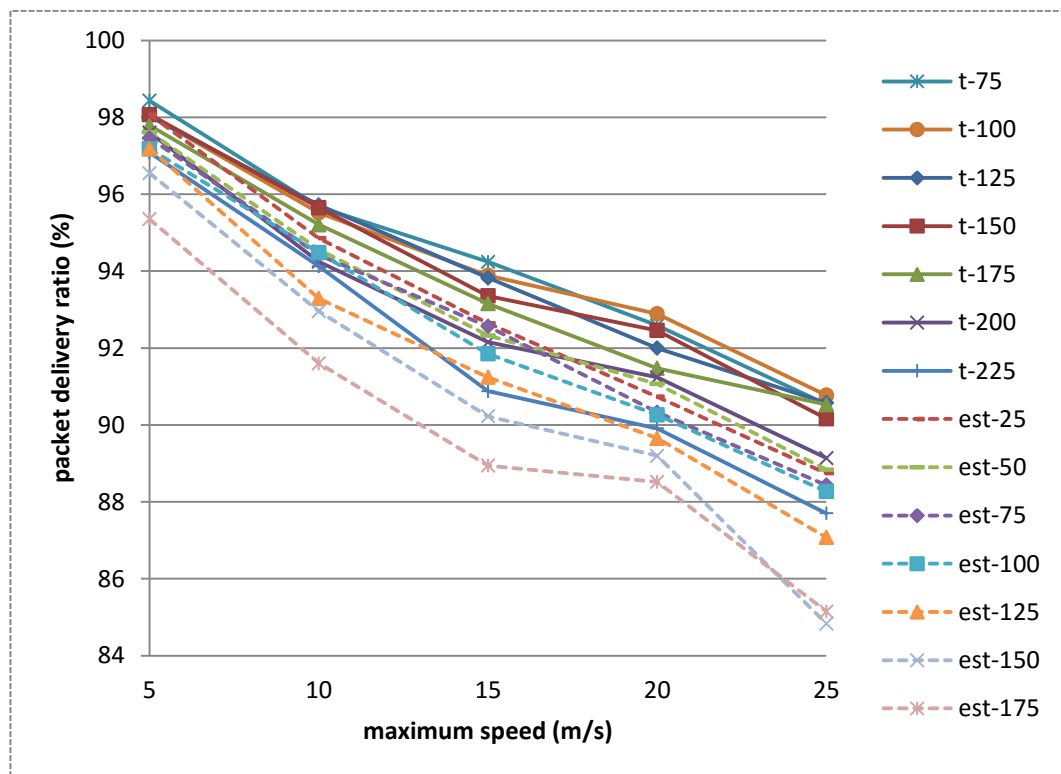
Figure 2.14: Comparing our proposed sensor-free methods with their corresponding sensor-using methods: (a) packet delivery ratio, (b) normalized routing load, (c) average packet latency, (d) average hop count, (e) number of route discoveries

### 2.4.5 Effect of Threshold Link Length

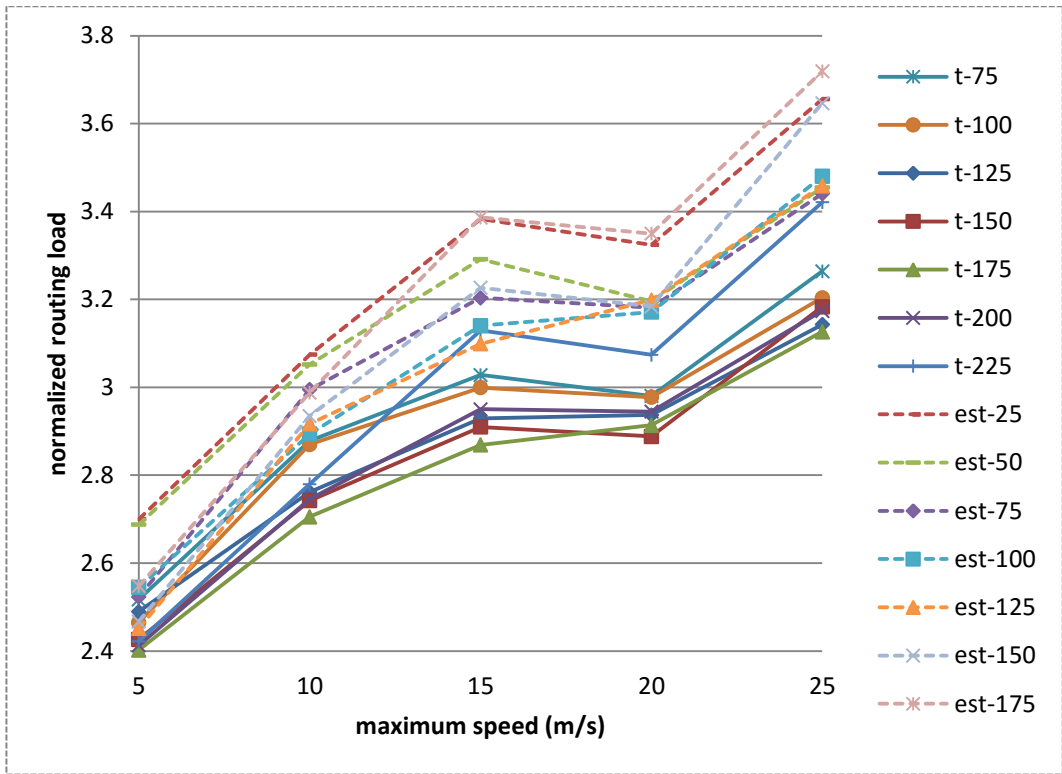
In this section, we investigate the effect of the threshold link length using the RS1 routing metric. We conducted the experiment using both true and estimated link length values. We varied the threshold link length value as follows: (1) 75-225 meters in increments of 25 meters when true link lengths were used, and (2) 25-175 meters in increments of 25 meters when estimated link lengths were used. Figure 2.15 shows the results obtained. The solid lines correspond to the results obtained when true link length values were used while the dotted lines correspond to the results obtained when estimated link length values were used.

Figure 2.15a shows the packet delivery ratio. It can be observed that packet delivery ratio increases as the threshold link length value decreases regardless of whether true or estimated link length information was used. It can also be observed that the packet delivery ratio lines are spread further apart at higher threshold link length values. This shows that there are diminishing returns at smaller threshold link length values. For this reason, we should not set the threshold link length to a value that is too small. That is why we have considered the values of  $0.7R=175$  meters and  $0.5R=125$  meters for the threshold link length when true or estimated link length values were used in our simulations in Sections 2.4.3 and 2.4.4. Figure 2.15b shows the normalized routing load. From this figure, we observed that when true link length values were used, setting the threshold link length value to 175 meters gave the overall best result. When estimated link length values were used, setting the threshold link length value to 125 meters seems to give good result. Figure 2.15c shows the average packet latency. We observed that in general, average packet latency decreases as the threshold link length value decreases although it becomes difficult to identify the overall best performing scheme at low threshold link length values as lines become converged in the 10-25 milliseconds region. Figure 2.15d shows the average hop count of delivered packets.

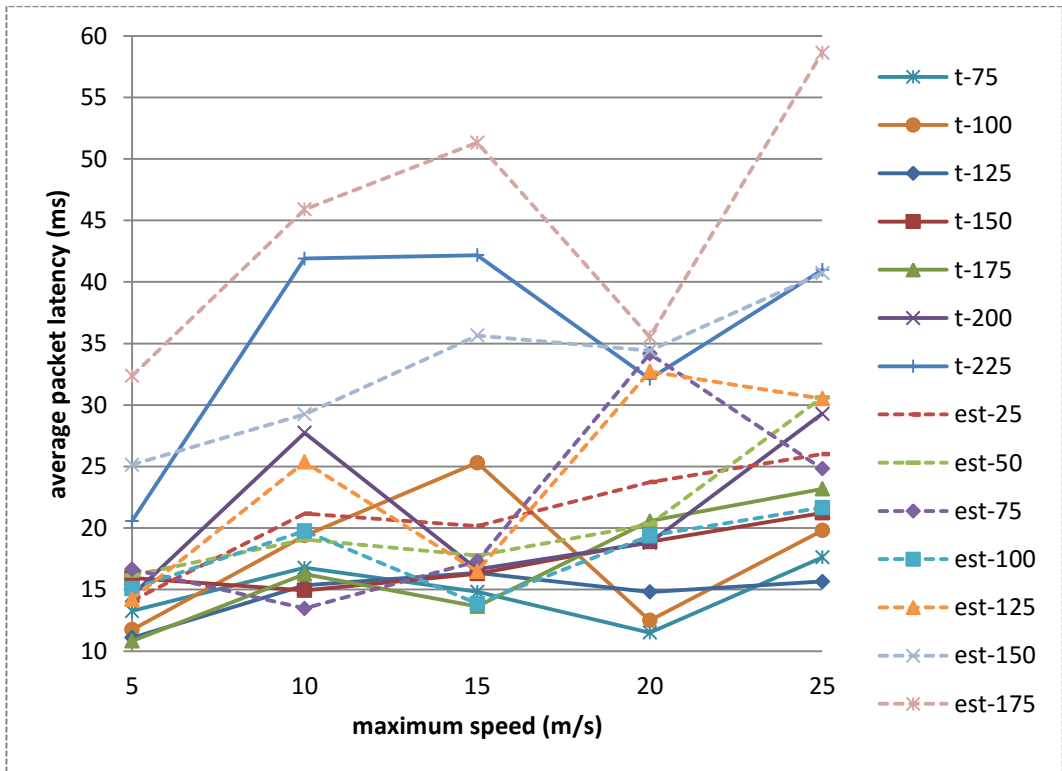
From this figure, we observed an inverse relation between average hop count and the threshold link length value. This can be expected because a lower threshold link length value gives higher penalty to longer links and favors higher hop count paths consisting of shorter links. We also observed that a threshold link length value difference of 25 meters corresponds to about 0.25 point in average hop count difference. Figure 2.15e shows the number of route discoveries produced. We observed that the overall best performance was obtained when the threshold link length value is about 175 meters when true link length values were used, and about 125 meters when estimated link length values were used.



(a)

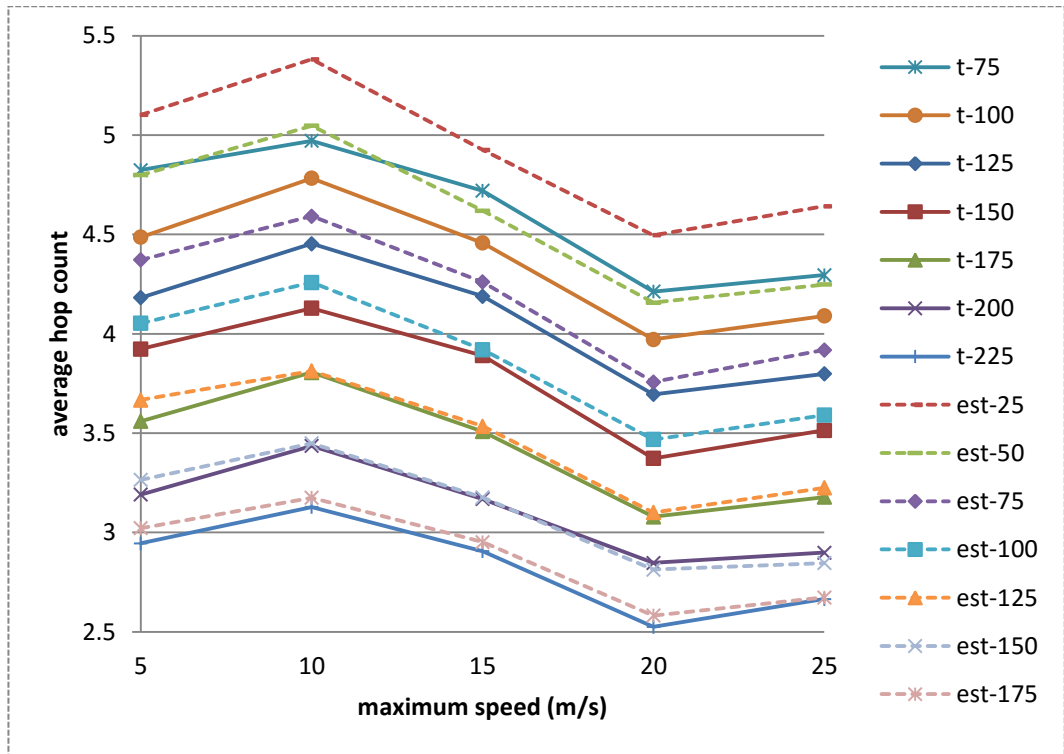


(b)

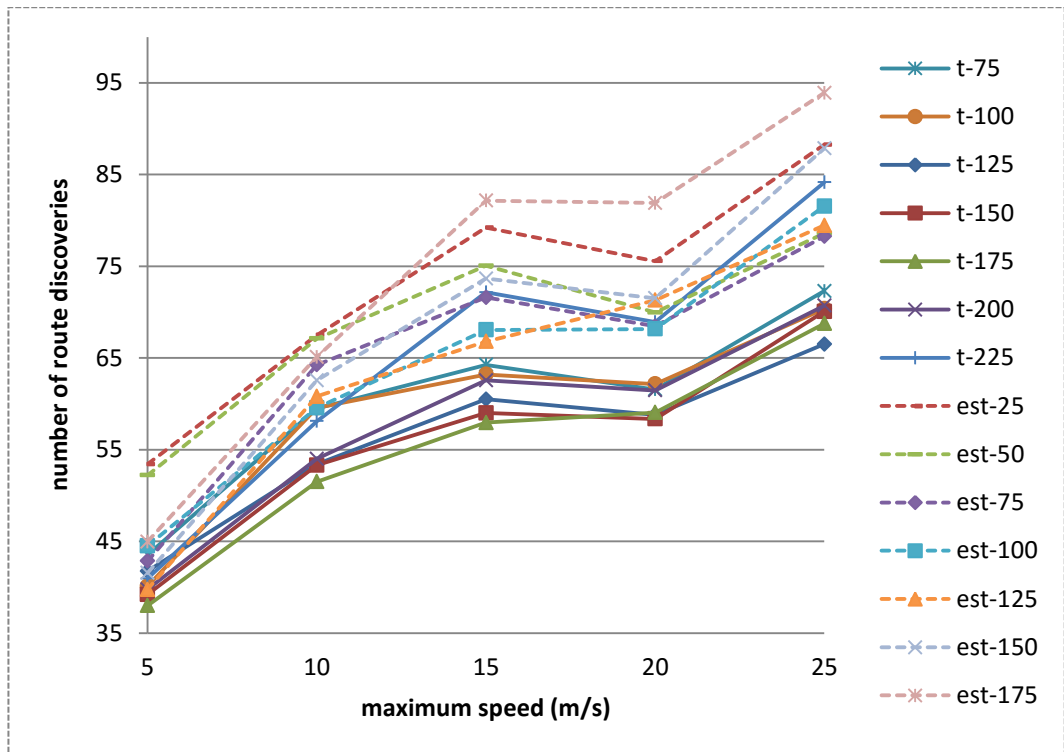


(c)





(d)



(e)

Figure 2.15: Investigating the effect of threshold link length: (a) packet delivery ratio, (b) normalized routing load, (c) average packet latency, (d) average hop count, (e) number of route discoveries

## **2.5 Conclusions**

In this chapter, we proposed two routing metrics for improving route stability in MANETs. The RS1 routing metric uses link length information and assigns a penalty to links exceeding a threshold length. The RS2 routing metric extends the RS1 routing metric to also consider node mobility. When accurate information is available, for example, by using GPS sensor, these routing metrics perform well and approach or exceed the level of performance possible with the link remaining lifetime (LRL) metric. However, the true beauty of these routing metrics is that they can be used even without sensors. From our investigation, we found that the proposed routing metrics to be highly effective and they outperformed existing routing metrics and the hop count metric even when used with less accurate information.

# CHAPTER 3: NETWORK CODING ROUTING PROTOCOL FOR WIRELESS AD HOC NETWORKS

## 3.1 Introduction

In Chapter 2, we found that route instability negatively impacts the performance of wireless ad hoc networks and we proposed routing metrics that can be used without sensors to guide source nodes select stable paths. Besides route instability, higher number of packet transmissions required for multi-hop communication in wireless ad hoc networks also has a detrimental effect on the network performance.

Wireless ad hoc networks usually suffer from low network throughput due to the higher number of packet transmissions required for multi-hop communication. In wireless ad hoc networks, the delivery of a packet from the source to the destination requires multiple packet transmissions because the source and destination are usually not within transmission range from each other and so intermediate nodes are required to relay the packet. In conventional single channel wireless ad hoc networks, these packet transmissions are performed on the same channel. However, as a channel has limited capacity, more transmissions required per packet translates to lower effective network throughput.

Many methods have been proposed to improve the throughput of wireless ad hoc networks such as multi-path transmission, multi-channel transmission using multiple network interface cards (NICs), and network coding. In this chapter, we employ the method of network load reduction using network coding. The benefit of network coding is best described with the “Alice and Bob network” (Katti et al., 2008) example, where Alice has a packet P1 for Bob and Bob has a packet P2 for Alice, as shown in Figure 3.1.

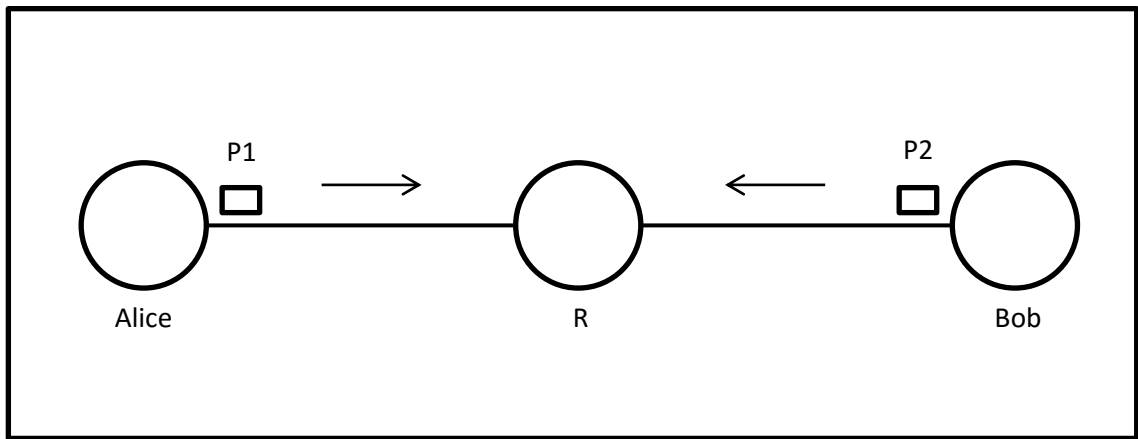


Figure 3.1: The Alice and Bob network

With the conventional store-and-forward scheme, four transmissions are required. This can be reduced to only three transmissions by using network coding as shown in the following process: (1) Alice transmits  $P1$  to node R, (2) Bob transmits  $P2$  to node R, and (3) node R encodes  $P1$  and  $P2$  together using the exclusive-or operation and broadcasts the resultant encoded packet  $P1 \oplus P2$ . Assuming that Alice has  $P1$  buffered after sending it, Alice can recover  $P2$  by decoding  $P1 \oplus P2$  using  $P1$  ( $(P1 \oplus P2) \oplus P1 = P2$ ). In a similar manner, Bob will also be able to recover  $P1$ .

If opportunistic listening is allowed, network coding can also occur in the network shown in Figure 3.2, where nodes 2 and 4 can overhear nodes 3 and 0, respectively. When node 4 receives  $P1 \oplus P2$  from node 1, node 4 can recover  $P2$  by decoding  $P1 \oplus P2$  using  $P1$  which it overheard from node 0. Similarly, node 2 can also recover  $P1$ .

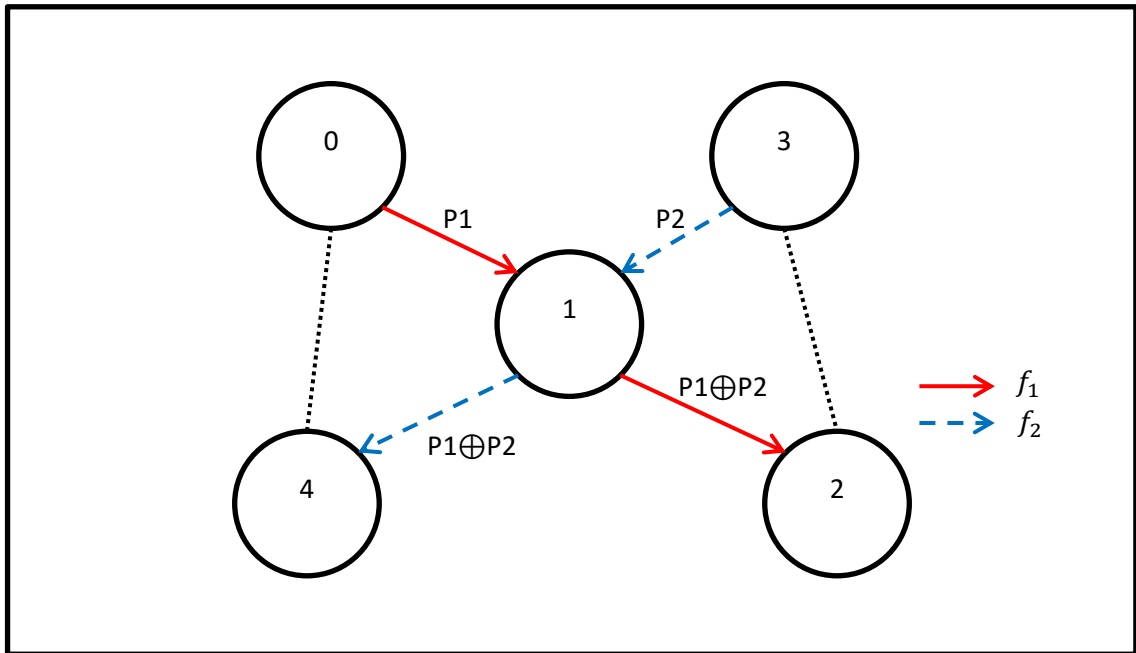


Figure 3.2: Opportunistic listening allows for packet encoding in the cross topology network

Coding conditions play a vital role in determining the network performance when network coding is used because they are referred to before packets are encoded together at a node. If the coding conditions inaccurately allow for packet encodings, packet loss due to decoding failures may occur. In addition, only the necessary coding conditions should be used; otherwise, coding opportunities may be missed. We have investigated the existing coding conditions and found some limitations in them. To overcome these limitations, we propose a new set of coding conditions called Improved Generalized Coding Conditions (IGCC). IGCC is then used to guide the design of a new ad hoc routing protocol called Network Coding Routing (NCRT). In addition, we also introduce a routing metric called Coding- and Load-Aware Routing Metric (CLARM), which takes into consideration both coding opportunities and network workload.

The remainder of this chapter is organized as follows. We review related work in Section 3.2 and detail our proposal in Section 3.3. In Section 3.4, we describe the experimental setup and present the results. Finally, we conclude in Section 3.5.

## 3.2 Related Work

### 3.2.1 Network Coding and Coding Conditions

Network coding was introduced to conserve bandwidth in wired networks for multicast flows as it was found to be non-optimal to treat multicast flows simply as fluid (Ahlswede & Cai, 2000). Network coding was later proposed for use in wireless mesh networks (WMNs) with the introduction of COPE (Katti et al., 2008). COPE enhances the forwarding layer by inserting a coding shim between the IP and MAC layers. This allows for the increase in network throughput while ensuring backward compatibility with conventional routing and higher layer protocols. A major limitation in COPE is that network coding is limited to only two hops. It is stated in COPE that to transmit  $n$  different packets  $p_1, \dots, p_n$  to  $n$  different next-hops  $r_1, \dots, r_n$ , a node has to ensure that each next-hop  $r_i$  has all  $n - 1$  packets  $p_j$  for all  $j \neq i$ . This coding condition ensures that every next-hop of an encoded packet can fully decode the packet, but it also restricts coding structures to only two hops, where a coding structure is a unique combination of packets/flows encoded together in a single packet and the coding node at which the encoding happened.

Coding-oblivious routing is often resulted when network coding is used with conventional routing protocols. In coding-oblivious routing, coding opportunities are not taken into account during route discovery and path selection; hence, nodes wait passively for coding opportunities to arise. Obviously, the network coding benefit cannot be fully utilized in such a manner and this led to the concept of coding-aware routing. In Figure 3.3, if the hop count routing metric is used, the two flows in the network, i.e., one from node 0 to node 2 and the other from node 2 to node 0, may use non-overlapping paths  $0 \rightarrow 1 \rightarrow 2$  and  $2 \rightarrow 3 \rightarrow 0$  as shown in Figure 3.3a because paths  $2 \rightarrow 3 \rightarrow 0$  and  $2 \rightarrow 1 \rightarrow 0$  have the same hop count. However, it is better to use the pair of overlapping paths  $0 \rightarrow 1 \rightarrow 2$  and  $2 \rightarrow 1 \rightarrow 0$  as shown in Figure 3.3b

because then packets can be encoded at node 1. The benefit of coding-aware routing was also confirmed in (Sengupta, Rayanchu, & Banerjee, 2010), where routing with COPE-type network coding was formulated as a linear program to determine the optimum network throughput. A similar formulation where links could have different maximum rates was established in (Zhang & Zhang, 2009).

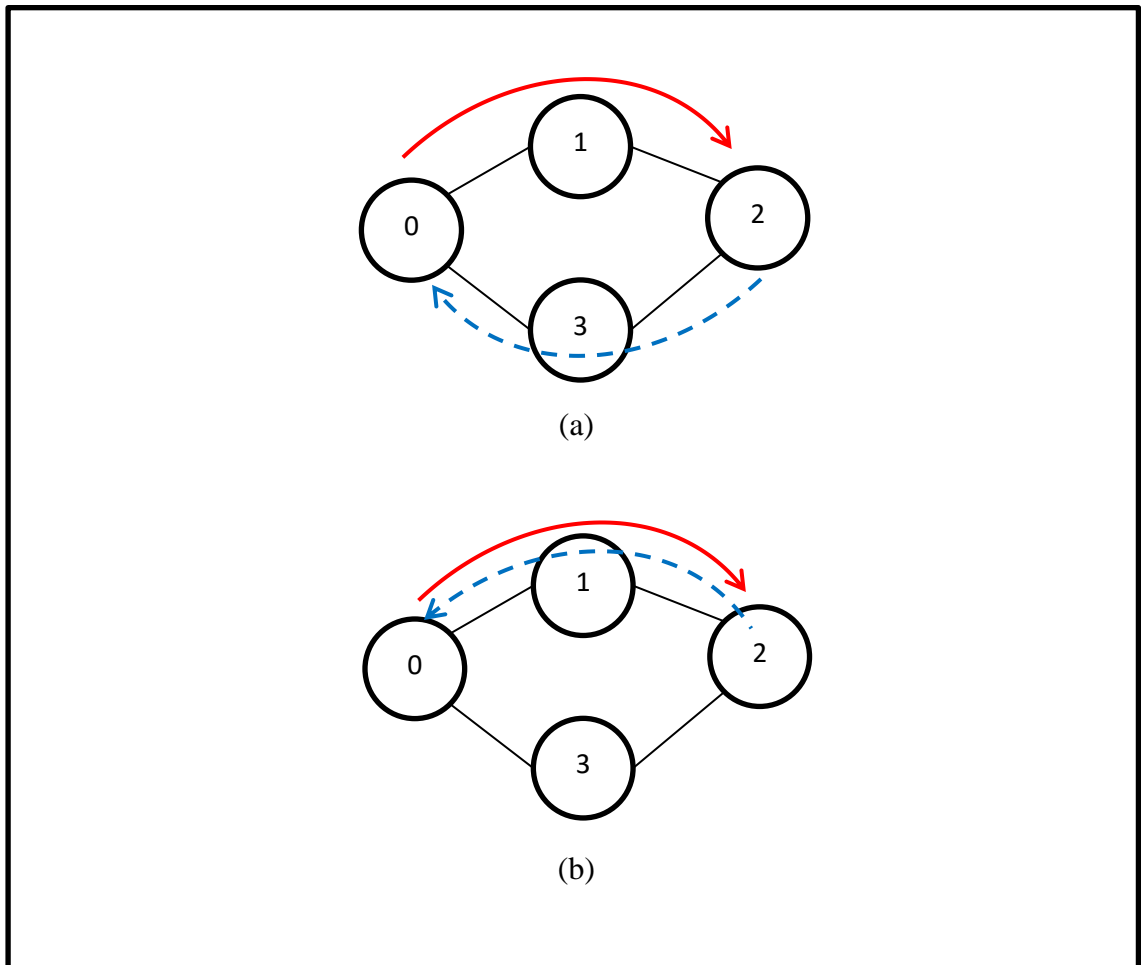


Figure 3.3: Path selection: (a) coding-oblivious routing, (b) coding-aware routing

The Distributed Coding Aware Routing (DCAR) (Le, Lui, & Chiu, 2010) protocol was proposed to overcome the two limitations in COPE (Katti et al., 2008): (1) coding structures restricted to within a region of two hops, and (2) coding-oblivious routing. The first limitation was broken with the proposal of a new set of coding conditions which states that to encode packets from two flows  $F_1$  and  $F_2$  intersecting at a node  $c$  the following conditions must be satisfied:

1. There exists  $d_1 \in D(c, F_1)$ , such that  $d_1 \in N(s_2)$ ,  $s_2 \in U(c, F_2)$ , or  $d_1 \in U(c, F_2)$ .
2. There exists  $d_2 \in D(c, F_2)$ , such that  $d_2 \in N(s_1)$ ,  $s_1 \in U(c, F_1)$ , or  $d_2 \in U(c, F_1)$ .

where  $N(\cdot)$ ,  $U(\cdot)$ , and  $D(\cdot)$  are the neighbor set, upstream nodes set, and downstream nodes set operator, respectively. Indeed, when the next-hops of an encoded packet are no longer required to fully decode the packet, network coding can happen beyond two hops. In Figure 3.4, nodes 2 and 7 are the next-hops of  $P1 \oplus P2$ . Although node 2 is unable to decode  $P1 \oplus P2$ , it can forward  $P1 \oplus P2$  to node 3, which can recover  $P1$  by decoding  $P1 \oplus P2$  using  $P2$  that node 3 overheard from node 5.

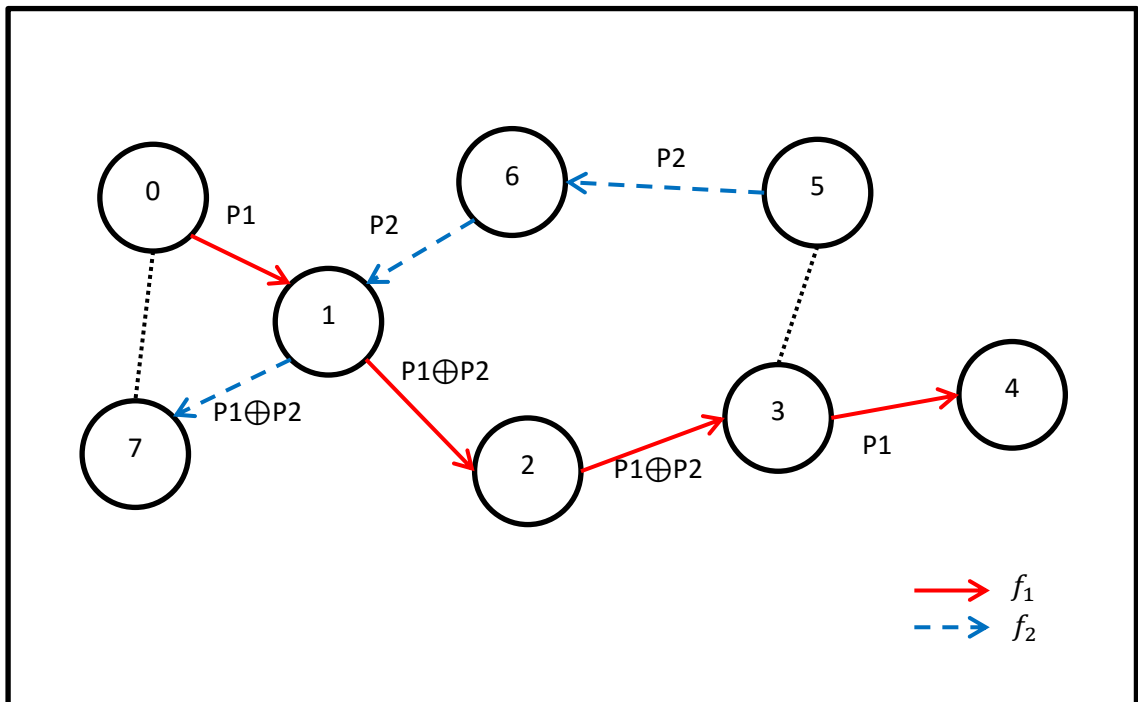


Figure 3.4: Relaxed coding conditions allow for packet encoding beyond two hops



While DCAR (Le et al., 2010) was successful in breaking the two-hop coding limitation in COPE (Katti et al., 2008), it was reported that the coding conditions in DCAR could lead to a phenomenon called the “false-coding effect” (Guo, Li, Zhou, & Cheng, 2011), which is illustrated using the network in Figure 3.5. Node 5 encodes  $P1 \oplus P3$  with  $P2$ . When node 8 receives  $P1 \oplus P3 \oplus P2$ , it cannot decode the packet and forwards the packet unchanged to node 9. However, when node 9 receives the packet, it can only recover  $P1 \oplus P2$  but not  $P2$ , which is the native packet intended for node 9. To avoid the false coding effect, more constraints were added to the coding conditions in DCAR resulting in a new set of coding conditions called Generalized Coding Conditions (GCC) (Guo et al., 2011). We argue that the false-coding effect occurs only because the coding conditions in DCAR were misinterpreted in (Guo et al., 2011) as it is stated in DCAR that to encode two flows at a node, the flows must intersect at the node. However, we observed that  $f_1$ ,  $f_2$ , and  $f_3$  do not intersect at node 5; hence,  $P1 \oplus P3$  and  $P2$  should not be encoded together at node 5.

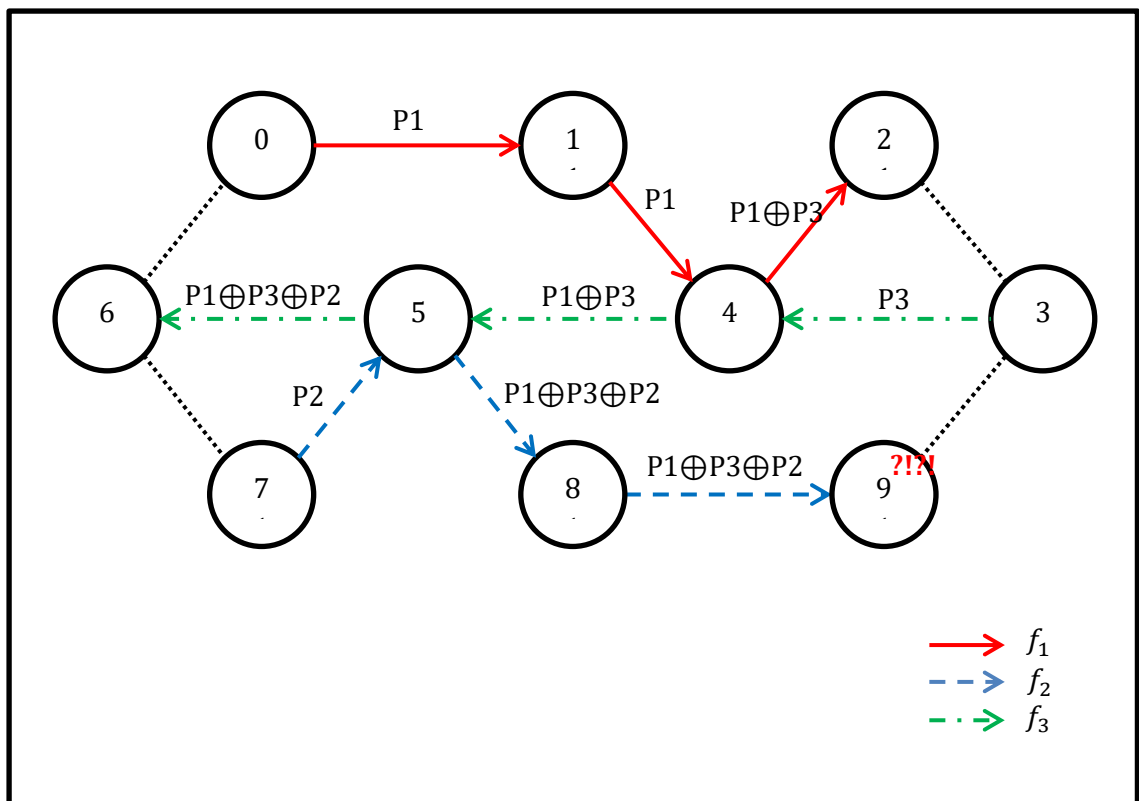


Figure 3.5: The false-coding effect

While the coding conditions in DCAR (Le et al., 2010) do not actually lead to decoding failures when applied correctly, we found there to be room for improvement. We refer to Figure 3.5 again but now suppose that node 8 can overhear node 1.  $P_2$  is a native packet and  $P_1 \oplus P_3$  is an encoded packet. The coding conditions in DCAR are unable to handle such a case as they require that the flows to be encoded together at a node also pass through the node but it can be seen that  $f_1$  does not pass through node 5. However, this constitutes to a missed coding opportunity because if the packets are encoded together, node 6 can recover  $P_3$  when it receives  $P_1 \oplus P_3 \oplus P_2$  from node 5, node 8 can recover  $P_3 \oplus P_2$  when it receives  $P_1 \oplus P_3 \oplus P_2$  from node 5, and node 9 can recover  $P_2$  when it receives  $P_3 \oplus P_2$  from node 8.

We also discovered some weaknesses in GCC (Guo et al., 2011). First, it can only determine if packets can be encoded together in either of the following two cases. The first case is that all the packets in consideration for being encoded together are native packets. The second case is that one of the packets is an encoded packet while the others are native packets. The case for evaluating if two or more encoded packets can be encoded together is not covered and this may cause some coding opportunities to be missed. The second problem lies with the “onion-peeling-like approach” used to evaluate if an encoded packet can be encoded with a native packet. The method works as follows: going down the considered native flow, the potential coding node determines if the destination of the native flow can successfully recover its native packet by enumerating all possible decoded versions of the potential encoded packet and checks if there is a version that can be fully decoded. If there exists a version of the potential encoded packet that can be fully decoded, the flows in consideration are allowed to be encoded together. We found a problem with this method. In Figure 3.6, the potential coding node  $c$  finds that the potential encoded packet can be fully decoded if the following is true: (1) node 1 decodes the encoded packet using overheard coded

flow  $f_2 \oplus f_3$  and forwards the partially decoded packet to node 2, (2) node 2 cannot decode the received packet and forwards the packet unchanged to node 3, and (3) node 3 decodes the received packet using overheard native flows  $f_1$  and  $f_4$ . In other words, if the top decoding sequence in Figure 3.6 is taken, the potential coding node finds that the potential encoded packet can be fully decoded at node 3 and so the potential coding node encodes the packets together. However, when node 1 actually receives the encoded packet later, it could decode the packet using another overheard flow instead of  $f_2 \oplus f_3$ , for example, using  $f_3 \oplus f_4$ . When this happens, the packet cannot be completely decoded by the time it reaches node 3. From this, we observed that a specific decoding sequence is required for decoding an encoded packet and the onion-peeling-like approach allows the packets in consideration to be encoded together as long as there is a decoding sequence that allows the resultant encoded packet to be fully decoded by the time or before the packet reaches the destination of the considered native flow. However, the decoding sequence that allows for the encoding might not be adhered by the downstream nodes of the coding node thus leading to decoding failure.

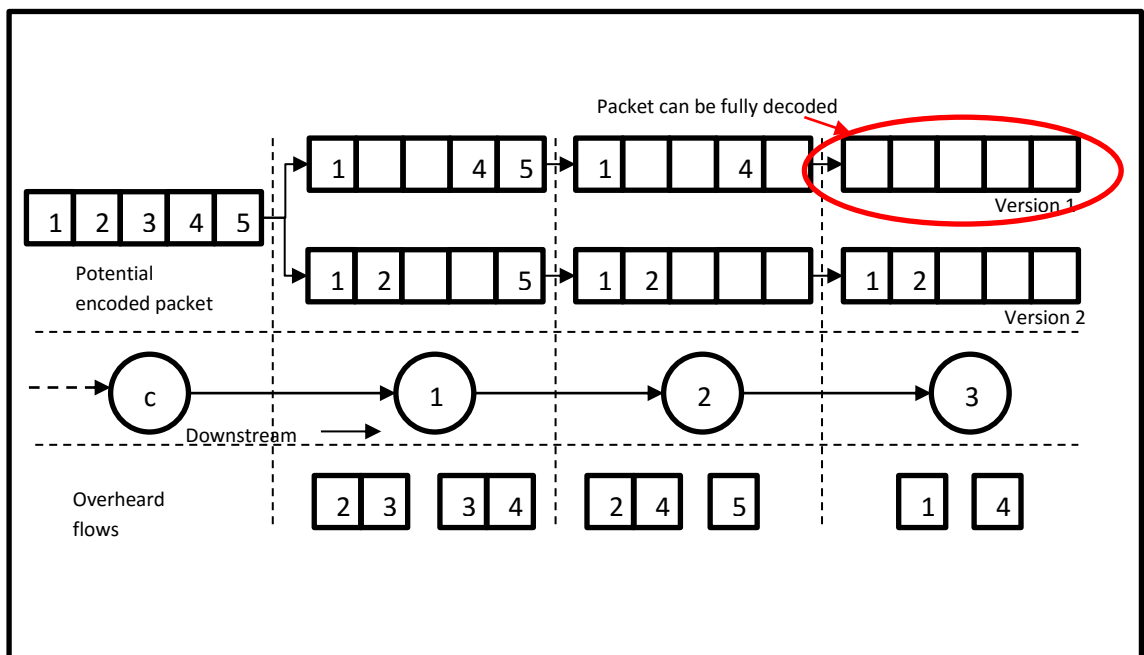


Figure 3.6: The onion-peeling-like approach

Network coding has also been proposed for use with opportunistic routing. In opportunistic routing, a node selects a subset of its neighbors as forwarding nodes as opposed to only one forwarding node in conventional routing. These forwarding nodes cooperate with each other to forward packets towards the destinations. It was said to be an effective forwarding technique for achieving high throughput in lossy wireless networks. Examples of related works are CAOR (Yan Yan, Zhang, Mouftah, & Ma, 2008), CORE (Y Yan, Zhang, Zheng, & Ma, 2010), and NCAC-MAC (X. Wang & Li, 2012). In COPE (Katti et al., 2008), a node periodically advertises packets that it heard using Reception Report packets. However, in CAOR, a node not only advertises packets that it heard but also packets heard by its neighbors. In this manner, a node would be able to obtain a two-hop local view of packets available to its neighbors and their neighbors. With this information, nodes of the same forwarding node set can cooperate with each other to route packets towards their destinations taking into account the coding opportunities.

Other than to improve network throughput, network coding has also been proposed for other purposes such as to improve energy efficiency (Abedi & Hariri, 2010) and security. A comprehensive survey of coding-aware routing protocols can be found in (Iqbal, Dai, Huang, Hassan, & Yu, 2011). Thus far, network coding that we discussed belongs to the class of inter-flow network coding. There is also another class of network coding called intra-flow network coding. Related works that belongs to intra-flow network coding are MORE (Chachulski, Jennings, Katti, & Katabi, 2007) and NC-RMR (Yuwang Yang, Zhong, Sun, & Yang, 2010). However, intra-flow network coding is beyond the scope of this thesis.

### **3.2.2 Routing Metrics**

Routing metric is a measure used to rank paths according to some criteria and is used in making routing decisions. Conventional ad hoc routing protocols mainly use

hop count as the routing metric but coding opportunities also has to be taken into consideration in coding-aware routing. The Expected Transmission Count (ETX) (De Couto, Aguayo, Bicket, & Morris, 2003) routing metric is used in the Coding-Aware Multi-Path (CAMP) (Han et al., 2008) routing protocol. In CAMP, a source node switches dynamically from one path to another if it finds that the coding gain from the switch exceeds the increase in ETX from the switch. However, the availability or abundance of packets that can be encoded with a packet of the path in consideration is not considered.

The Expected Number of Coded Transmissions for an Exchange (ECX) (Ni, Santhapuri, Zhong, & Nelakuditi, 2006) routing metric computes the expected number of transmissions required for a successful exchange of packets between two nodes via an intermediate node when network coding is used and retransmissions are allowed to recover from lost packets.

The Coding-Aware Routing Metric (CRM) is used in the DCAR routing protocol (Le et al., 2010). It computes the expected number of transmissions required to successfully transmit existing packets as well as one incoming packet of the considered path. The problem with CRM is that it considers the existing packets in the network interface queues of the nodes along the considered path but the routing metric value of the path is only computed during route discovery and is not updated afterwards. Hence, it might not work as intended. The Coding and Interference Aware Routing (CIAR) (Y. Peng et al., 2013) protocol uses a routing metric that considers interference. The routing metric was extended from CRM as follows. The Modified Interface Queue Length (MIQ) of a link in CRM was modified by multiplying it with  $\frac{1}{IR}$ , where  $IR$  is the interference ratio and was derived based on the Physical Model of Interference (Gupta & Kumar, 2000). However, we deem this to be redundant because interference is already considered in MIQ (Le et al., 2010). A coding-aware routing metric similar to

that proposed in CIAR was proposed in the On-demand Coding Aware Routing (OCAR) (Sun, Liu, Hu, & Yuan, 2010) protocol. The routing metric in OCAR is based on the Estimated Transmission Time (ETT) (Draves et al., 2004) instead of the MIQ in CIAR.

The Free-ride Oriented Routing Metric (FORM) (Guo et al., 2011) consists of two parts: (1) modified benefit,  $B_m$ , and (2) degree of free ride,  $DegFR$ . The modified benefit computes the difference between the gain (decrease) and the loss (increase) in hop count by sending a packet on a considered path instead of sending the packet on the shortest path, while the degree of free ride computes the abundance of coding opportunities in the considered path. The path with the highest FORM value is used for packet transmission. However, we think that the path with the highest FORM value may not necessary be the best path because its FORM value could be bloated by its degree of free ride. In addition, the FORM values of paths are determined only during route discoveries and are not updated afterwards. As long as a discovered path is not broken, its FORM value, which is computed during route discovery, is used for the entire duration of its lifespan. While such a strategy is appropriate for certain routing metrics such as the hop count routing metric where the hop count of a path does not change over time in the lifespan of the path, it is inappropriate for routing metrics that are based on dynamic values.

### 3.3 Network Coding Routing (NCRT)

#### 3.3.1 Base Routing Protocol

In this section, we describe our routing framework with network coding in wireless ad hoc networks called Network Coding Routing (NCRT), which was modified from the Dynamic Source Routing (DSR) (Johnson et al., 2001) protocol. DSR was chosen because source routing allows the complete path to be known before a packet is sent out by its source node. This property allows for coding-aware routing and network coding beyond two hops.

In coding-aware routing, a source node selects the best path to send a packet according to a coding-aware routing metric. When destination-vector routing is used, such as when the Ad-hoc On-demand Distance Vector (AODV) (Perkins & Royer, 1999) routing protocol is used, packets may not traverse on the optimal end-to-end path because the source can only determine the next-hop but not the entire path a packet should take to reach its destination.

In source routing, the complete path information recorded in packet headers also allows potential coding nodes to determine if packets can be encoded beyond two hops. This would not be possible if AODV is used because then a potential coding node can only identify the next-hops of a potential encoded packet and the destinations of the native packets in the potential encoded packet. As an example, consider the network shown in Figure 3.4, where packets from  $f_1$  and  $f_2$  can be encoded together at node 1. If AODV is used, node 1 does not know that node 3 is one of its downstream nodes in  $f_1$  because node 1 can only identify in  $f_1$  its next-hop which is node 2, and the destination which is node 4. Hence, node 1 does not encode P1 and P2 together and a coding opportunity is missed.

### 3.3.2 Coding Conditions

To overcome the limitations in the existing coding conditions, a new set of coding conditions called Improved Generalized Coding Conditions (IGCC) is proposed. Before we proceed, we look at some of the desirable properties of an ideal set of coding conditions:

1. *Compatibility*: An ideal set of coding conditions should ensure that when packets are encoded together, the destinations of the native packets in the encoded packets can recover their respective native packets. There are three scenarios the destination of a native packet can recover its native packet. First, it has enough packets to fully decode the encoded packet all by itself. Second, it receives a packet that is partially decoded by its upstream nodes and it has the remaining packets necessary to fully decode the packet. Third, it receives a packet that is fully decoded by its upstream nodes.
2. *Appropriateness*: Coding conditions should not be stricter than necessary as coding opportunities may be missed. For example, COPE (Katti et al., 2008) limits coding structures to only two hops.

Similar to COPE (Katti et al., 2008), DCAR (Le et al., 2010), and FORM (Guo et al., 2011), in NCRT, an encoded packet is pseudo-broadcasted to several intended receivers by the coding node. In pseudo-broadcast, a packet is unicasted to the next-hop with the addresses of the other intended receivers besides the next-hop recorded on the packet. We refer to such a receiver as a “pseudo-broadcast receiver”. The other key terms used in defining IGCC are summarized in Table 3.1.



Table 3.1: Terms and their definitions

<b>Terms</b>	<b>Definitions</b>
$f_r^*$	the primary flow of the received packet
$f_q^*$	the primary flow of the enqueued packet
$f_r$	the flow of a native packet in the received packet
$f_q$	the flow of a native packet in the enqueued packet
$F_r$	the set of flows of all native packets in the received packet
$F_q$	the set of flows of all native packets in the enqueued packet
$nexthop(x, f)$	the next-hop of node $x$ in flow $f$
$D(x, f)$	the set of downstream nodes of node $x$ in flow $f$
$U(x, f)$	the set of upstream nodes of node $x$ in flow $f$
$U_{native}(x, f)$	the set of upstream nodes of node $x$ in flow $f$ at which a native packet of $f$ was transmitted natively
$N(x)$	the set of neighbors of node $x$
$pr$	a pseudo-broadcast receiver of the enqueued packet
$PR$	the set of all pseudo-broadcast receivers of the enqueued packet
$f'_{pr}$	The primary flow of the pseudo-broadcast receiver $pr$ . It is the flow to be set as the primary flow when an encoded packet reaches $pr$ .

IGCC is formally defined as follows. At a potential coding node  $c$ , the following conditions must be satisfied before two packets can be encoded together, where each of these packets can be either native or encoded.

1.  $nexthop(c, f_r^*) \neq nexthop(c, f_q^*)$
2.  $nexthop(c, f_r^*) \neq pr, \forall pr \in PR$
3.  $\exists n_1 \in D(c, f_q^*) \cup (N(D(c, f_q^*)) \setminus \{c\})$  such that  $n_1 \in U_{native}(c, f_r), \forall f_r \in F_r$
4.  $\exists n_2 \in D(c, f_{pr}') \cup (N(D(c, f_{pr}')) \setminus \{c\})$  such that  $n_2 \in U_{native}(c, f_r), \forall f_r \in F_r, \forall pr \in PR$
5.  $\exists n_3 \in D(c, f_r^*) \cup (N(D(c, f_r^*)) \setminus \{c\})$  such that  $n_3 \in U_{native}(c, f_q), \forall f_q \in F_q$

Conditions 1 and 2 state that the next-hop of  $c$  in the primary flow of the received packet should not be equal to either the next-hop of  $c$  in the primary flow of the enqueued packet or any of the pseudo-broadcast receivers of the enqueued packet. Together, these two conditions ensure that all the intended receivers of the resultant encoded packet are unique. Condition 3 states that for each native flow in the received packet, there must exist a node in the set of downstream nodes of  $c$  in the primary flow of the enqueued packet and their neighbors that is also an upstream node of  $c$  in the native flow of the received packet. This is to ensure that the resultant encoded packet can be fully decoded by the time or before it reaches the destination of the primary flow of the enqueued packet. Condition 4 states that for each possible combination of a native flow in the received packet and a pseudo-broadcast receiver of the enqueued packet, there must exist a node in the set of downstream nodes of  $c$  in the primary flow of the pseudo-broadcast receiver and their neighbors that is also an upstream node of  $c$  in the native flow of the received packet. This is to ensure that the resultant encoded packet can be fully decoded by the time or before it reaches the destinations of the primary flows of the pseudo-broadcast receivers. Condition 5 states that for each native flow in the enqueued packet, there must exist a node in the set of downstream nodes of  $c$  in the primary flow of the received packet and their neighbors that is also an upstream

node of  $c$  in the native flow of the enqueued packet. This is to ensure that the resultant encoded packet can be fully decoded by the time or before it reaches the destination of the primary flow of the received packet. Conditions 3 and 4 determine if the enqueued packet can accommodate the received packet, while condition 5 determines if the received packet can accommodate the enqueued packet. Collectively, conditions 3-5 ensure that the native packets in the resultant encoded packet can be recovered by their respective destinations. Encoding of two or more packets together can be done by considering two packets at a time.

In the network in Figure 3.7, three coding structures exist: (1) a native packet of  $f_1$ , i.e., P1 with a native packet of  $f_2$ , i.e., P2 at node 1, (2) a native packet of  $f_1$ , i.e., P1 with a native packet of  $f_3$ , i.e., P3 at node 2, and (3) an encoded packet of  $f_1 \oplus f_2$ , i.e.,  $P1 \oplus P2$  encoded at node 1 with a native packet of  $f_3$ , i.e., P3 at node 2 (note: the packets shown in Figure 3.7 is for the third coding structure only). The first two cases are trivial so we illustrate how IGCC allows for packet encoding of the third case.

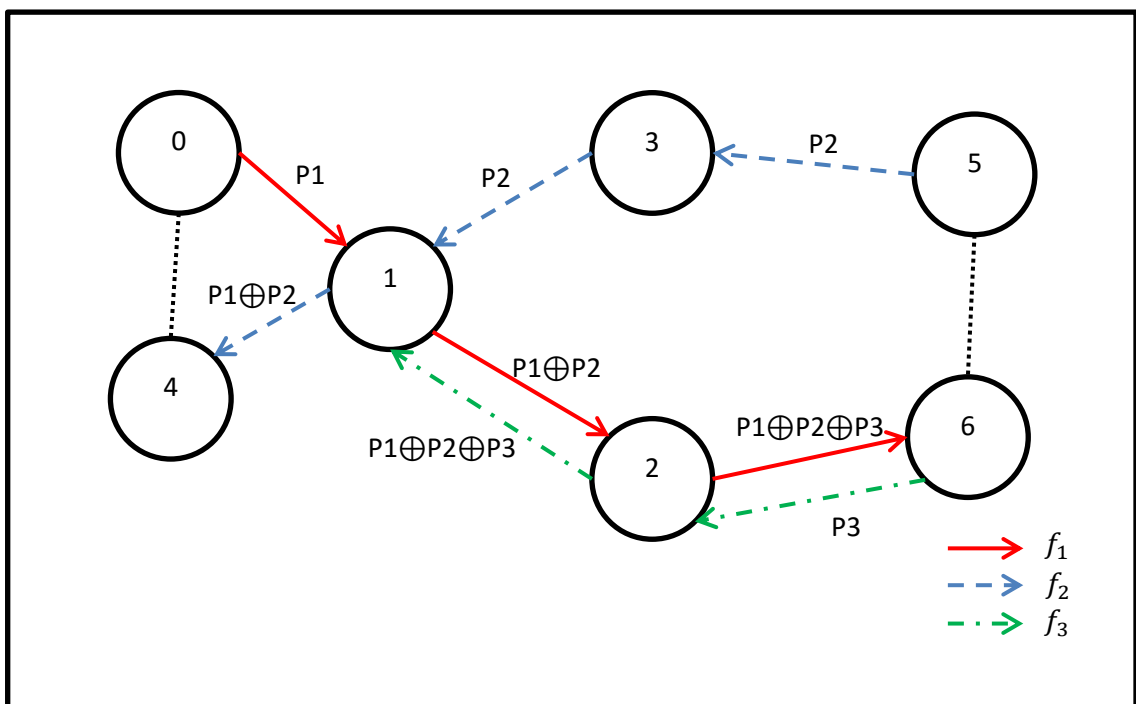


Figure 3.7: The illustrative network

We assume that P3 is the enqueued packet and  $P1 \oplus P2$  is the received packet. We have the following:

- $c = 2$
- $f_r^* = f_{r1} = f_1 = 0 \rightarrow 1 \rightarrow 2 \rightarrow 6$
- $f_{r2} = f_2 = 5 \rightarrow 3 \rightarrow 1 \rightarrow 4$
- $E_r = \{f_{r1}, f_{r2}\}$
- $f_q^* = f_{q1} = f_3 = 6 \rightarrow 2 \rightarrow 1$
- $E_q = \{f_{q1}\}$
- $PR = \emptyset$
- $nexthop(c = 2, f_r^*) = 6$
- $nexthop(c = 2, f_q^*) = 1$
- $D(c = 2, f_q^*) = \{1\}$
- $N(D(c = 2, f_q^*) \setminus \{2\}) = N(1) \setminus \{2\} = \{0, 2, 3, 4\} \setminus \{2\} = \{0, 3, 4\}$
- $D(c = 2, f_q^*) \cup (N(D(c = 2, f_q^*) \setminus \{2\})) = \{1\} \cup \{0, 3, 4\} = \{0, 1, 3, 4\}$
- $U(c = 2, f_{r1}) = \{0, 1\}$
- $U_{native}(c = 2, f_{r1}) = \{0\}$
- $U_{native}(c = 2, f_{r2}) = \{3, 5\}$
- $D(c = 2, f_r^*) = \{6\}$
- $N(D(c = 2, f_r^*) \setminus \{2\}) = N(6) \setminus \{2\} = \{2, 5\} \setminus \{2\} = \{5\}$
- $D(c = 2, f_r^*) \cup (N(D(c = 2, f_r^*) \setminus \{2\})) = \{6\} \cup \{5\} = \{5, 6\}$
- $U(c = 2, f_{q1}) = \{6\}$
- $U_{native}(c = 2, f_{q1}) = \{6\}$

Condition 1 is satisfied since  $nexthop(c = 2, f_r^*) = 6 \neq nexthop(c = 2, f_q^*) = 1$ .

1. Since  $PR = \emptyset$ , we do not have to check condition 2. Conditions 3-5 require the use of  $U_{native}(c, f)$ , which can be obtained in practice by recording in packets at which nodes the native packets in them were transmitted natively. For example, P1 is transmitted natively, i.e., not encoded with any other packets at node 0, and P2 is transmitted natively at nodes 5 and 3 before the two packets are encoded together at node 1; hence, we have  $U_{native}(c = 2, f_{r1}) = \{0\}$  and  $U_{native}(c = 2, f_{r2}) = \{3, 5\}$ . Since  $\{0\} \in D(c = 2, f_q^*) \cup (N(D(c = 2, f_q^*) \setminus \{2\}))$  and  $\{0\} \in U_{native}(c = 2, f_{r1})$ , and  $\{3\} \in D(c = 2, f_q^*) \cup (N(D(c = 2, f_q^*) \setminus \{2\}))$  and  $\{3\} \in U_{native}(c = 2, f_{r2})$ , condition 3 is satisfied. Since  $PR = \emptyset$ , we do not have to check condition 4. Since  $\{6\} \in D(c = 2, f_r^*) \cup (N(D(c = 2, f_r^*) \setminus \{2\}))$  and  $\{6\} \in U_{native}(c = 2, f_{q1})$ , condition 5 is

satisfied. Since all the conditions in IGCC are satisfied, the packets are allowed to be encoded together. When the two packets are encoded together, we then have the following for the resultant encoded packet: next-hop = 1,  $PR = \{6\}$ , and  $f'_{pr=6} = f_r^* = f_{r1} = f_1 = 0 \rightarrow 1 \rightarrow 2 \rightarrow 6$ . Collectively, node 1 (the next-hop of the resultant encoded packet) and node 6 (the single pseudo-broadcast receiver of the resultant encoded packet) form the set of intended receivers of the resultant encoded packet.

In the example above,  $P3$  and  $P1 \oplus P2$  are encoded at node 2 for the first time. Since  $PR = \emptyset$  for the enqueued packet  $P3$ , conditions 2 and 4 do not have to be checked. The two conditions are used when packets are encoded together multiple times at a node. For example, suppose packets  $P4$ ,  $P5$ , and  $P6$  are to be encoded together at a node. After encoding the first two packets together, for example,  $P4$  and  $P5$ ,  $PR$  will then be non-empty for  $P4 \oplus P5$  and will be used in evaluating if  $P4 \oplus P5$  can be encoded with  $P6$ .

### 3.3.3 Route Discovery

Some information is used in IGCC to determine if two packets can be encoded together at a node. In order for nodes to gather that information, we propose the following changes to the route discovery process in DSR (Johnson et al., 2001):

1. *Initiating a new route discovery process:* When a node has a packet to send but does not have a valid route to the packet's destination, the node initiates a new route discovery process by sending a Route Request (RREQ).
2. *Handling received RREQs:* To avoid routing loop, when a node receives a RREQ, it examines the route record of the RREQ and drops the RREQ if the RREQ has traversed through itself. The node also drops the RREQ if it is a duplicate copy (by maintaining a broadcast ID cache of RREQs). If the RREQ is not dropped and the node is the RREQ destination, the node sends a Route Reply (RREP) back to the RREQ source. If the RREQ is not dropped and the

node is not the RREQ destination, the node appends its address to the route record of the RREQ and forwards the RREQ.

3. *Replying RREQs*: Upon receiving a RREQ, the RREQ destination creates a new RREP packet and initializes the metric field in the RREP to 0. It also records the addresses of its neighbors on the RREP. After that, it sends the RREP to the RREQ source along the reverse of the path traversed by the RREQ.
4. *Handling received RREPs*: When a node receives a RREP, it records its downstream nodes and their neighbors. After that, if the node is an intermediate node, it records the addresses of its neighbors onto the RREP and updates the metric field in the RREP before finally forwarding the RREP to the next-hop. Otherwise, if the node is the RREQ source, it updates the metric field in the RREP and stores the updated value of the metric field in the routing table together with the path associated with the RREP.

To allow for a better understanding of the route discovery process, we show an example using the network shown in Figure 3.7. Initially, node 0 does not have a route to node 6; hence, node 0 initiates a new route discovery by sending a RREQ. Suppose a copy of the RREQ reaches node 6 traversing the path  $f_1$ . When node 6 receives the RREQ, it creates a new RREP and initializes the value of the metric field in the RREP to 0. Node 6 then records the addresses of its neighbors, which are nodes 2 and 5, onto the RREP. At that moment, the RREP looks like that as shown in Figure 3.8a. After that, node 6 sends this RREP to the RREQ source along the reverse of the path traversed by the RREQ. When node 2 receives the RREP from node 6, node 2 knows who the neighbors of node 6 are. Since node 2 is not the RREP destination, node 2 further modifies the RREP by recording the addresses of its neighbors, which are nodes 1 and 6, onto the packet and updates the metric field in the packet. The modified packet is as shown in Figure 3.8b if we assume that the value of the metric field in the RREP is

increased by 1 (note: we cannot determine accurately the increase in the metric value for  $f_1$  in this example because it is dependent on the network interface queue utilization of node 2 and also the number of packets in node 2's network interface queue that can be encoded with a packet of  $f_1$ ). After that, node 2 forwards the packet. This process is repeated until the RREP reaches its destination, which is node 0. In this manner, upstream nodes know who the neighbors of the downstream nodes are. In other words, upstream nodes can identify who their downstream nodes can overhear.

We already know that node 1 is a potential coding node for  $f_1$  and  $f_2$ . With the route discovery process above, node 1 has the following information: (1) its downstream nodes in  $f_1$  and their neighbors, (2) its upstream nodes in  $f_1$ , (3) its downstream nodes in  $f_2$  and their neighbors, and (4) its upstream nodes in  $f_2$ . Using IGCC, node 1 determines that packets from the two flows can be encoded together.

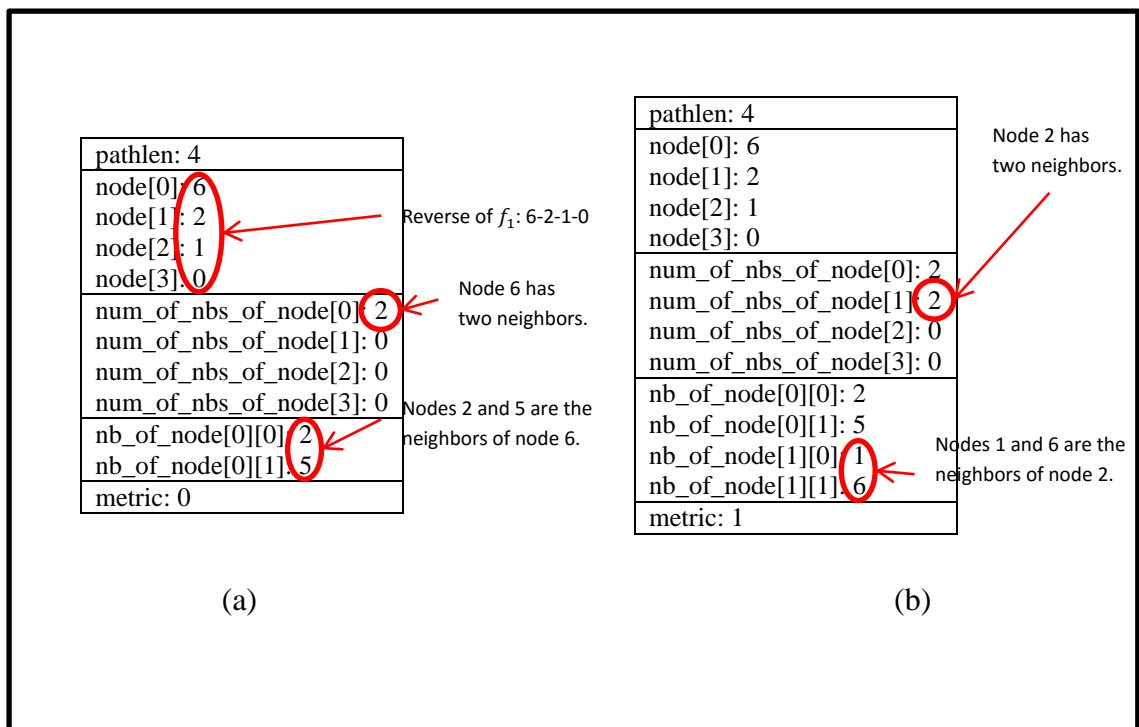


Figure 3.8: Contents of RREP: (a) node 6, (b) node 2

### 3.3.4 Routing Metric

In this section, we propose the Coding- and Load-Aware Routing Metric (CLARM) to maximize the network coding benefit to increase network throughput. Before we proceed, we look at some traits of an ideal path for packet transmission when network coding is integrated into routing.

1. *Low transmission count*: Paths with lower transmission count cause lower channel contention and interference. They also save bandwidth by reducing unnecessary transmissions.
2. *Many coding opportunities*: Paths with many coding opportunities allow packets to be encoded together to save bandwidth.
3. *Avoid congested nodes*: A congested node may have a full network interface queue and is therefore unable to accommodate more packets. When new packets are routed through such nodes, the nodes have no choice but to drop the packets thus causing packet loss.

To guide source nodes select ideal paths with the above traits, CLARM was designed with the following principle: *try to use as few transmissions as possible to send a packet taking into consideration coding opportunities*. CLARM consists of two parts: the coding-aware part and the load-aware part. The coding-aware part is given by  $1 - Penc_n^{p_i}$ , where  $p_i$  is the path in consideration, and  $Penc_n^{p_i}$  is a function that maps the sum of native packets from other paths  $p_j, \forall j \neq i$  that can be encoded with a packet of  $p_i$  at node  $n, n \in p_i, n \neq dst(p_i)$  to a value in the interval  $[0, 1]$ . We have chosen a simple function to represent  $Penc_n^{p_i}$ .  $Penc_n^{p_i}$  is shown in Figure 3.9 and its equation is given in Eq. 3.1.



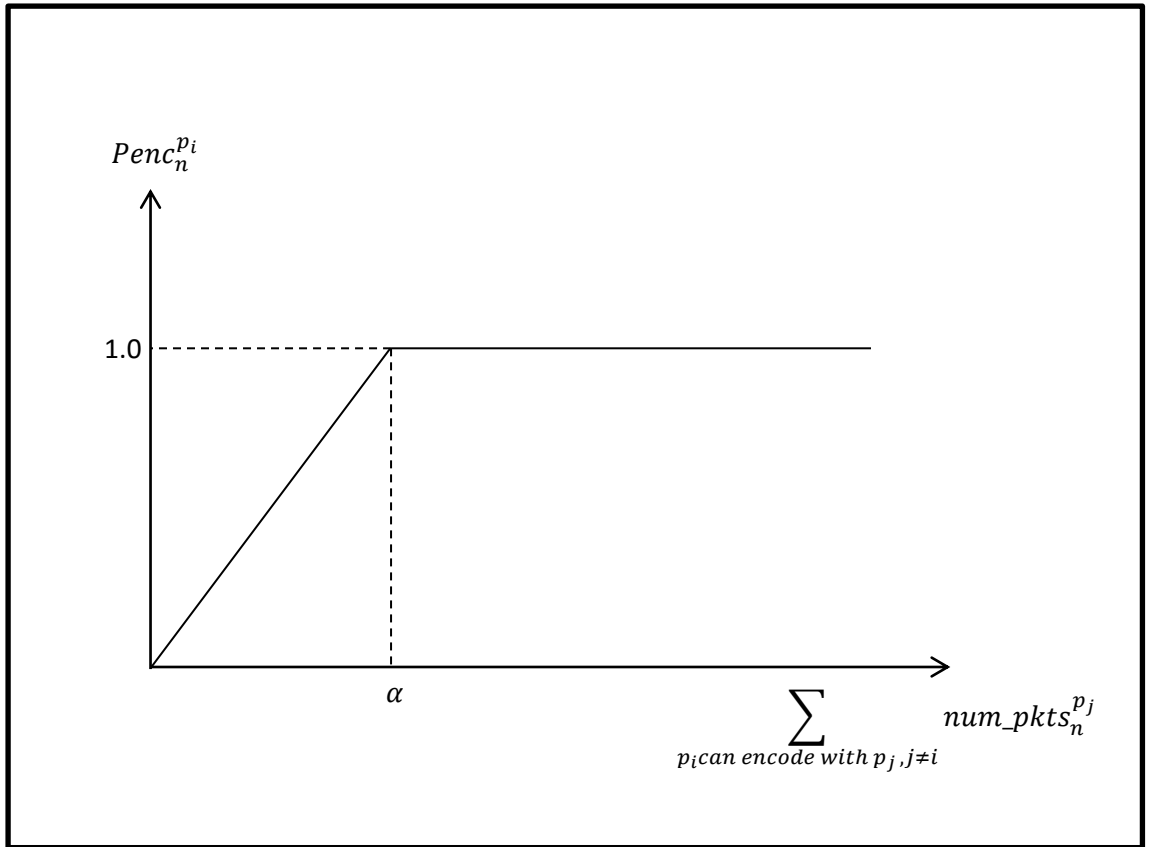


Figure 3.9: A simple function to map the number of packets code-able with a packet of the considered path to a value in  $[0, 1]$

$$Penc_n^{p_i} = \min \left( 1.0, \frac{1}{\alpha} \times \sum_{p_i \text{ can encode with } p_j, j \neq i} num\_pkts_n^{p_j} \right) \quad \text{Eq. 3.1}$$

For ease of exposition, we refer to node  $n$  and “a packet of path  $p_i$ ” as *the considered node* and *the considered packet* hereafter, respectively. We can view  $Penc_n^{p_i}$  as the probability of the considered node encoding the considered packet. When  $Penc_n^{p_i}$  is 0, it is implied that the considered node is unable to encode the considered packet. On the other hand, when  $Penc_n^{p_i}$  is 1, it is implied that the considered node can encode the considered packet. Hence,  $1 - Penc_n^{p_i}$  can be viewed as the number of transmissions required by the considered node to forward the considered packet to the next-hop. When  $Penc_n^{p_i}$  is 0,  $1 - Penc_n^{p_i}$  is 1 and this implies that the considered node would require one transmission to forward the considered packet to the next-hop. When  $Penc_n^{p_i}$  is 1,  $1 - Penc_n^{p_i}$  is 0 and this implies that the considered node would not require a

transmission to forward the considered packet to the next-hop.  $Penc_n^{p_i}$  was intentionally shaped in such a way because it is desirable for the considered node to have many packets in its network interface queue that can be encoded with the considered packet. When the considered node has only few packets in its network interface queue that can be encoded with the considered packet, for example, one or two, the considered packet may not be able to be encoded with one of those packets by the time it actually reaches the considered node because the packets may no longer be available.

The choice of the value of  $\alpha$  affects how easily or difficult it is for  $Penc_n^{p_i}$  to attain its maximum value of 1. If  $\alpha$  is set to a low value, it is sufficient for the considered node to have few packets in its network interface queue that can be encoded with the considered packet for  $Penc_n^{p_i}$  to attain its maximum value. On the contrary, if  $\alpha$  is set to a high value, it is required that the considered node has many packets in its network interface queue that can be encoded with the considered packet for  $Penc_n^{p_i}$  to attain its maximum value. In other words, it is easier to obtain  $Penc_n^{p_i}$  equals 1 when  $\alpha$  is set to a lower value. Since the value of  $Penc_n^{p_i}$  depends on the number of packets that the considered node has in its network interface queue that can be encoded with the considered packet, and this number is at most the maximum network interface queue length, setting  $\alpha$  to a value greater than the maximum network interface queue length is not sensible because it implies that it can never be certain of encoding the considered packet at the considered node because  $Penc_n^{p_i}$  would never attain the value of 1. On the other hand, setting  $\alpha$  to 0 results in  $Penc_n^{p_i}$  always equal to 1, and consequently  $1 - Penc_n^{p_i}$  always equal to 0, which implies that the considered node requires no transmission to forward the considered packet to the next-hop. Moreover, setting  $\alpha$  to 0 also makes every path to appear equal with a CLARM value of 0. Obviously, the value 0 is also not a valid choice to set for  $\alpha$ .

The load-aware part of CLARM is given by  $1/LI_n^{pi}$ , where  $LI_n^{pi}$  is a function that maps the network interface queue utilization, i.e., the fill of the network interface queue of node  $n$  in percentage,  $ifqueue\_uti_n$ , to a value in the interval  $(0, 1]$ . We can view  $1/LI_n^{pi}$  as the load indication at the considered node. We have chosen to use the function shown in Figure 3.10 for  $LI_n^{pi}$ .

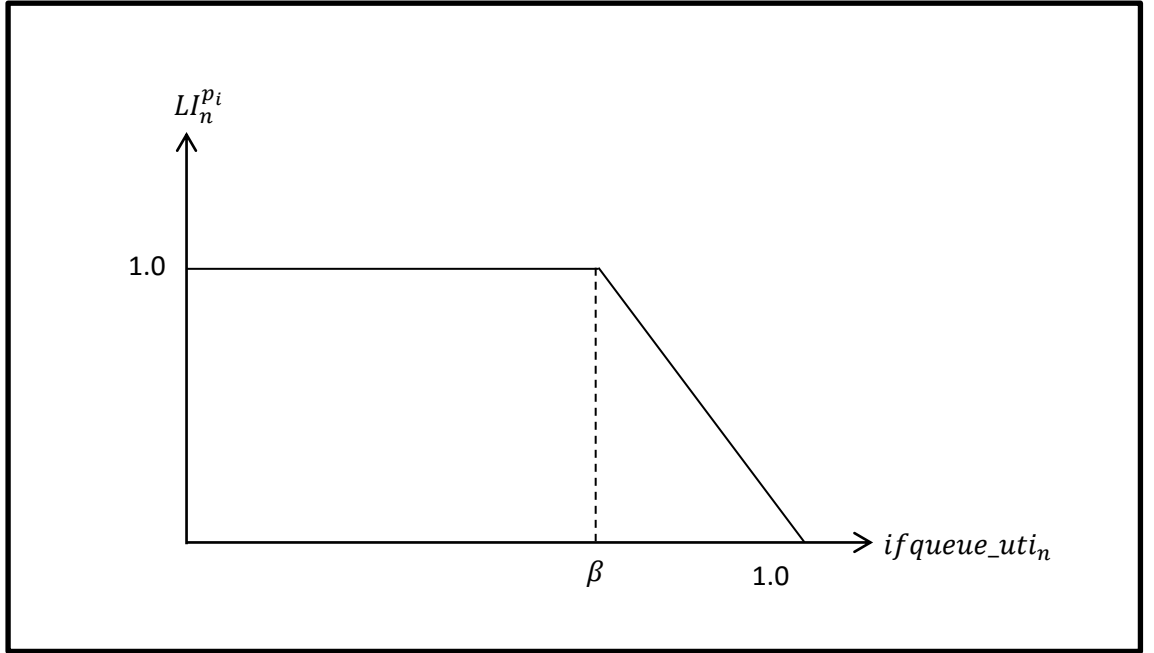


Figure 3.10: A simple function to map the network interface queue utilization of a node to a value in  $(0, 1]$

The equation of  $LI_n^{pi}$  is given in Eq. 3.2.

$$LI_n^{pi} = \min \left( 1.0, \frac{1}{\beta - 1.0} \times ifqueue\_uti_n + \frac{1}{1.0 - \beta} \right) \quad \text{Eq. 3.2}$$

As mentioned, we can view  $1 - Penc_n^{pi}$  as the number of transmissions required by the considered node to forward the considered packet to the next-hop. When we multiply  $1 - Penc_n^{pi}$  with  $1/LI_n^{pi}$ , we modify the number of transmissions required by the considered node to forward the considered packet to the next-hop. When the network interface queue utilization of the considered node is low, i.e., it takes a value lesser than the threshold value  $\beta$ ,  $LI_n^{pi}$  is equal to 1.  $1 - Penc_n^{pi}$  is unmodified when

$LI_n^{p_i}$  is equal to 1 (i.e.,  $(1 - Penc_n^{p_i})/LI_n^{p_i} = 1 - Penc_n^{p_i}$ ). This implies that the number of transmissions required by the considered node to forward the considered packet to the next-hop depends only on whether or not the considered node can encode the considered packet. However, when the network interface queue utilization of the considered node is high, i.e., it takes a value greater than  $\beta$ ,  $LI_n^{p_i}$  takes a value in the interval  $(0, 1)$ . When this happens, the number of transmissions required by the considered node to forward the considered packet to the next-hop is raised to a higher value to discourage the use of paths that contain the considered node. We deliberately shaped  $LI_n^{p_i}$  in such a way because it is difficult to differentiate the workload levels of nodes simply by comparing their network interface queue utilizations. As an example, suppose one node has a network interface queue utilization of 40% while another node has a network interface queue utilization of 60%. Is it really better to route through the first node instead of the second one? If yes, how much is it better to route through the first node instead of the second one? However, what we are certain of is to avoid routing through nodes with very high network interface queue utilizations, for example, routing through a node with a network interface queue utilization of 95%. By having a threshold, we do not differentiate nodes based on their network interface queue utilizations as long as their network interface queue utilizations are below the threshold.

The value of  $\beta$  affects the discouraging effect of routing through paths that contain nodes with very high network interface queue utilization. If  $\beta$  is set to a lower value,  $LI_n^{p_i}$  begins to drop to a value lesser than 1 at a lower network interface queue utilization. On the other hand, if  $\beta$  is set to a higher value,  $LI_n^{p_i}$  begins to drop to a value lesser than 1 at a higher network interface queue utilization. In other words, the discouraging effect of routing through the considered node is more aggressive when a lower value of  $\beta$  is used. If desired,  $\beta$  can even be set to 1. In that case, CLARM degenerates altogether to a routing metric that does not take into account the workload

of the nodes.

The metric value of the considered path  $p_i$  is given by the  $CLARM_{p_i}$  value, which is defined in Eq. 3.3.

$$CLARM_{p_i} = \sum_{n \in p_i, n \neq dst(p_i)} \frac{(1 - Penc_n^{p_i})}{LI_n^{p_i}} \quad \text{Eq. 3.3}$$

The path with the lowest CLARM value can be viewed as the path that requires the least number of transmissions to transmit a packet from the source to the destination. In NCRT, a source node selects the path with the lowest CLARM value to send its packets. From Eq. 3.3, it can be seen that  $(1 - Penc_n^{p_i})/LI_n^{p_i}$  decreases with more coding opportunities and increases with higher network interface queue utilization. With CLARM, we hope to guide source nodes select ideal paths to send their packets on, i.e., paths that are short, have many coding opportunities, and do not go through congested areas in the network.

In a route discovery process, the RREQ source discovers several paths to the RREQ destination. The CLARM value of a path is computed as its corresponding RREP traverses from the RREQ destination to the RREQ source. It is also computed and updated periodically using a new control packet called Reverse Route Update (RVRTUPD). In NCRT, nodes monitor incoming paths and periodically send RVRTUPDs to the source nodes to update the routing metric values of the paths at the sources nodes. The RVRTUPD is quite similar to the RREP in structure (Figure 3.8). RVRTUPDs are also used to update nodes with the latest information regarding their downstream nodes and downstream nodes' neighbors. With the RVRTUPD mechanism, in NCRT, a source node gets up-to-date information about the network and selects the best path according to the CLARM routing metric for sending its packets.

### 3.4 Simulation Studies

We evaluated NCRT using network simulator 2 (ns-2) (“The Network Simulator - ns-2,” n.d.). The settings used are summarized in Table 3.2. The MAC-related parameters were set in such a way to simulate the IEEE 802.11g ERP-DSSS physical layer (Villaseñor-González, 2007)(Vassiss et al., 2005). Four types of networks were used: (1) the illustrative network, (2) chain networks, (3) grid network, and (4) random networks.

Table 3.2: ns-2 settings

<b>Parameter</b>	<b>Value</b>
Radio propagation model	Two-ray ground
MAC	IEEE 802.11
Antenna model	Omni-antenna
<b>MAC Related</b>	
Minimum congestion window size	15 slots
Maximum congestion window size	1023 slots
Slot time	20 us
SIFS duration	10 us
Preamble length	72 bits
PLCP header length	48 bits
RTS threshold	2346 B
Short retry limit	7
Long retry limit	4
Data rate	11 Mbps
Basic rate	1 Mbps
PLCP data rate	1 Mbps
<b>Network Interface Related</b>	
Frequency	2.412 MHz
Transmission power	16 dBm
Receiving threshold power	5.184e-11 (250 m)
Carrier sense threshold power	2.21296e-12 (550 m)
<b>NCRT Related</b>	
$\alpha$	10
$\beta$	0.7
<b>Others</b>	
Transport protocol	UDP
Maximum network interface queue length	100 packets

### 3.4.1 The Illustrative Network

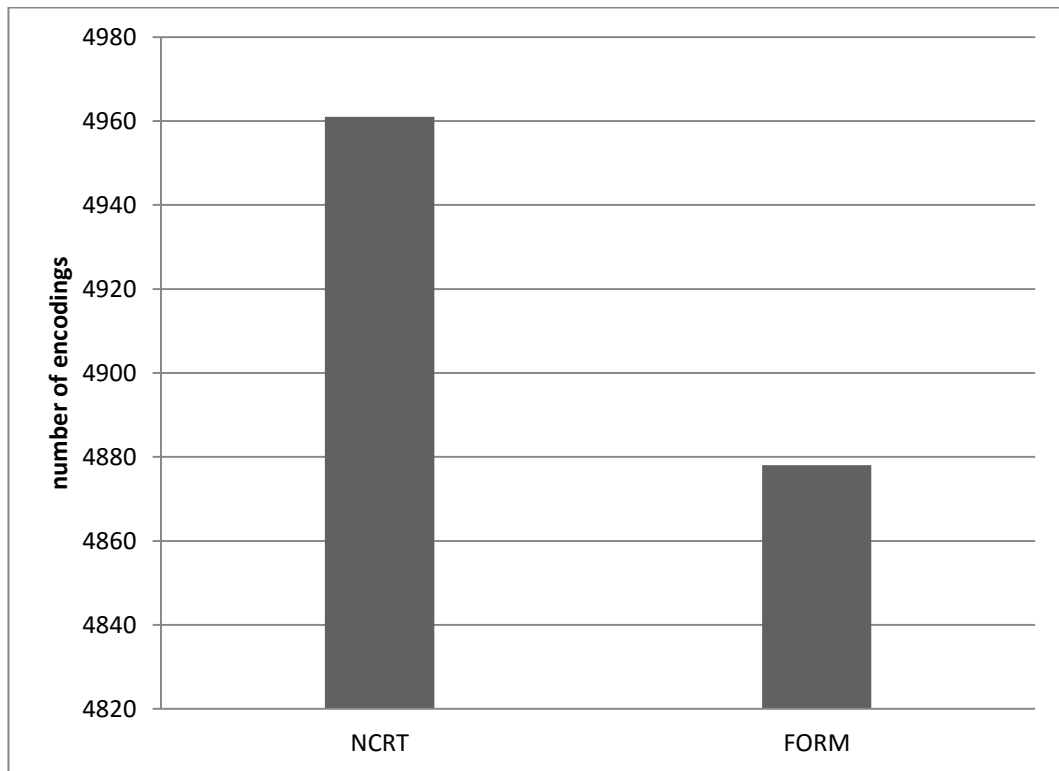
In this test, we used “the illustrative network” shown in Figure 3.7 to compare the protocols. The purpose of this test is to show that under identical scenarios, IGCC can produce more packet encodings than GCC. For the purpose of creating a level playing field, the route discovery procedures in NCRT and FORM were slightly modified so that the paths discovered by these protocols are similar to the paths shown in Figure 3.7. This modification was necessary to avoid other paths from being discovered and used. For example, for the flow from node 0 to node 6, the paths  $0 \rightarrow 4 \rightarrow 1 \rightarrow 2 \rightarrow 6$ ,  $0 \rightarrow 1 \rightarrow 3 \rightarrow 5 \rightarrow 6$ , and  $0 \rightarrow 4 \rightarrow 1 \rightarrow 3 \rightarrow 5 \rightarrow 6$  may be discovered and used if the modification was not applied. The sources of the flows were set to generate and send data packets at a rate of 800 Kibps (1 Kib = 1024 b) and the duration of the simulation is 60 seconds.

By discovering more coding structures, we hope that more packet encodings are produced so that network throughput is improved. It is widely agreed that a path that can lead to more packet encodings does not always mean better network performance because such a path may be long and winding. Long and winding paths increase the number of transmissions required sending a packet to its destination, and causes more channel contention and interference. However, in this test, since the same paths were discovered and used in both protocols, more packet encodings is certainly beneficial to network performance.

From Figure 3.11a, we observed that NCRT produced more packet encodings. This is because NCRT discovered three coding structures in the network: (1) a native packet of  $f_1$  with a native packet of  $f_2$  at node 1, (2) a native packet of  $f_1$  with a native packet of  $f_3$  at node 2, and (3) an encoded packet of  $f_1 \oplus f_2$  encoded at node 1 with a native packet of  $f_3$  at node 2. In contrast, FORM discovered only the first two coding structures. Due to more packet encodings, NCRT obtained slightly better total network

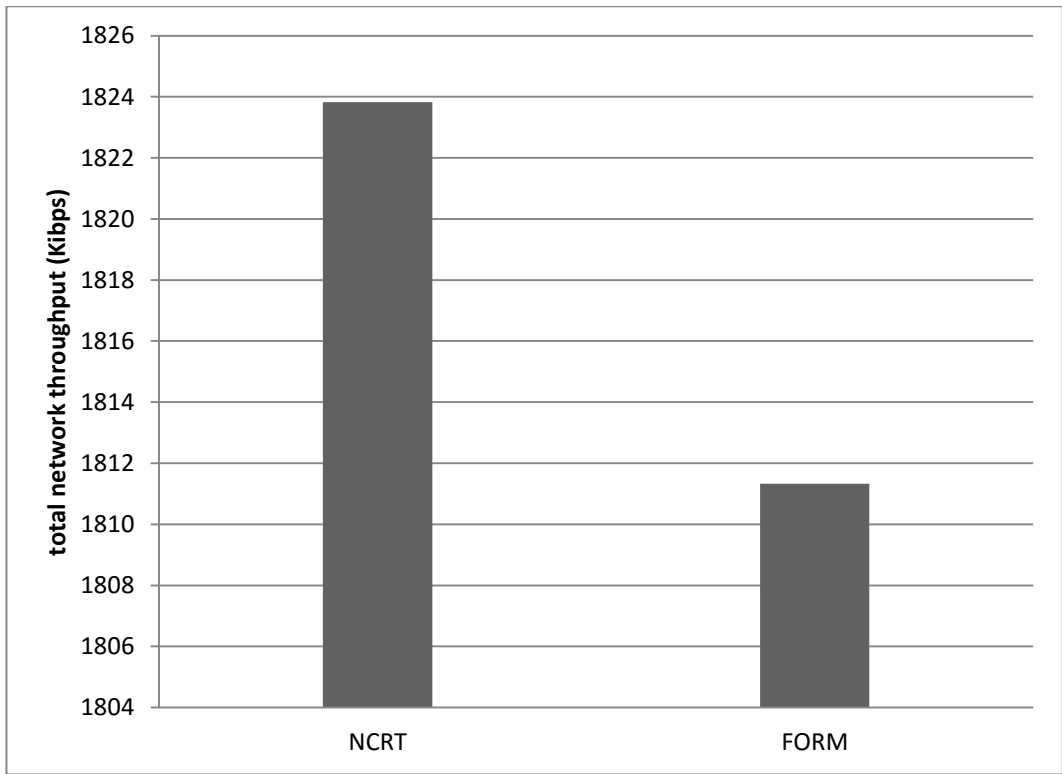
throughput and average packet delay, as shown in Figures 3.11b and 3.11c, respectively. While the gains are negligible, it should be noted that they were obtained with NCRT produced more routing overhead than FORM, as shown in Figure 3.11d.

In the illustrative network, there are three flows and each of the flows contains only one path. In NCRT, RVRTUPD packets are sent periodically from a destination to the source through different reverse paths to update the routing metric values of the forward paths. However, since there is only one path for each flow in this scenario, sending of RVRTUPDs is unnecessary. Regardless, we did not disable the RVRTUPD mechanism which explains why there was more routing overhead in NCRT than FORM as shown in Figure 3.11d.

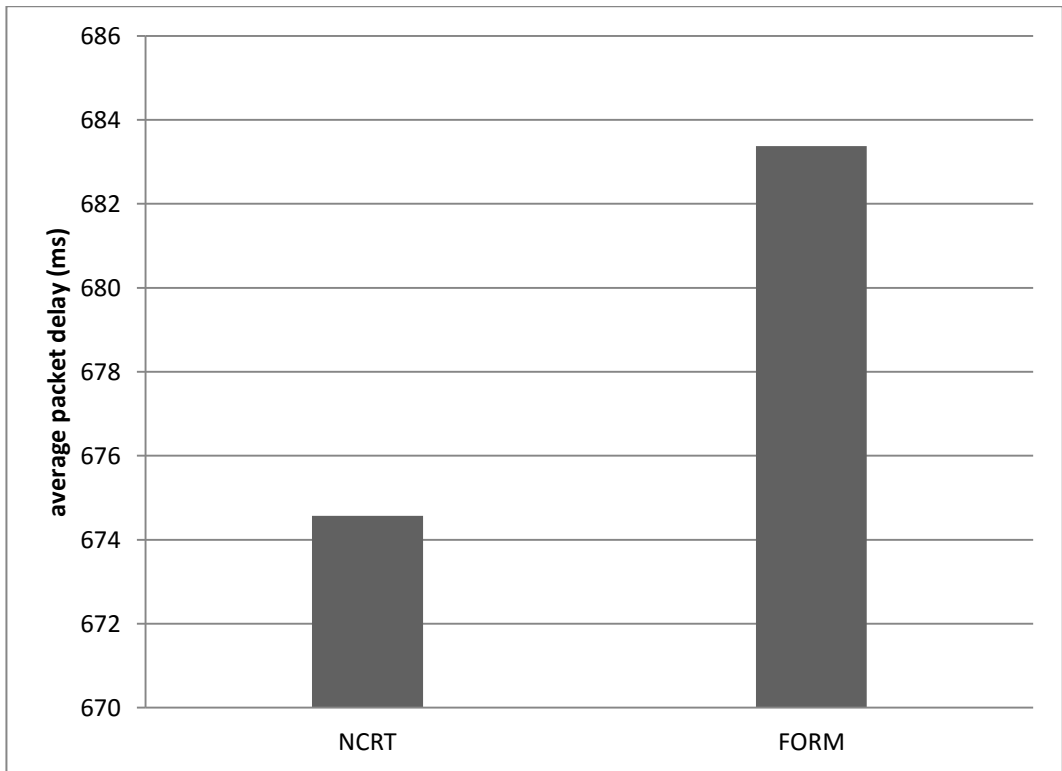


(a)

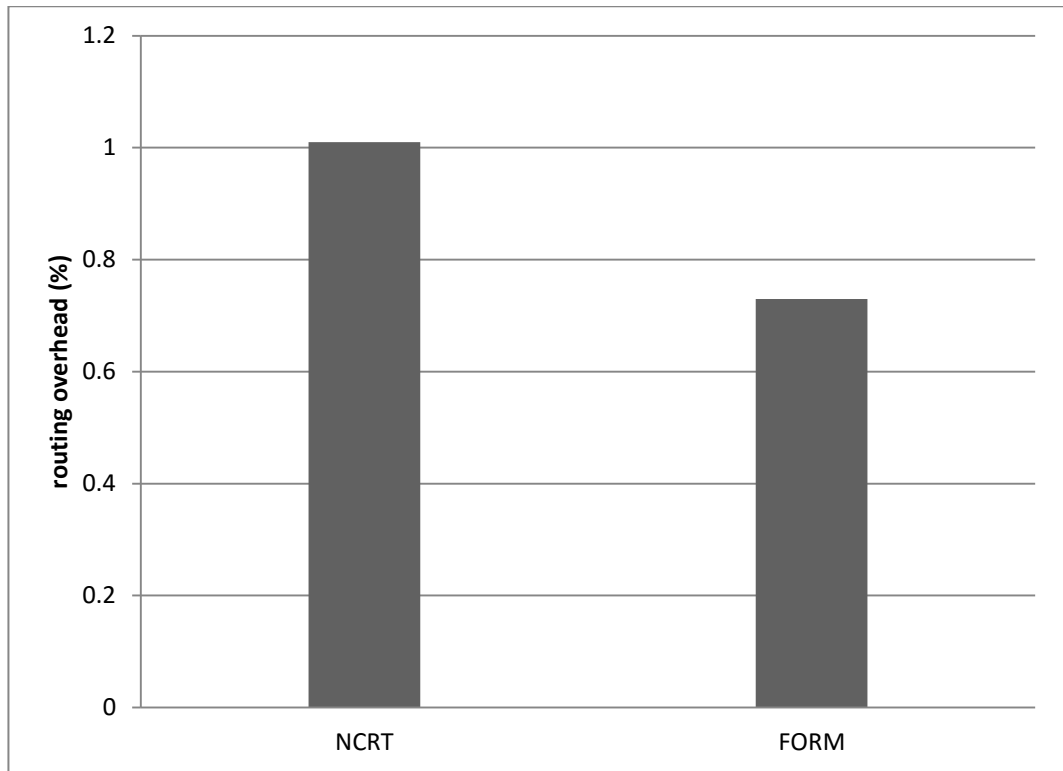




(b)



(c)



(d)

Figure 3.11: The illustrative network: (a) number of encodings, (b) total network throughput, (c) average packet delay, (d) routing overhead

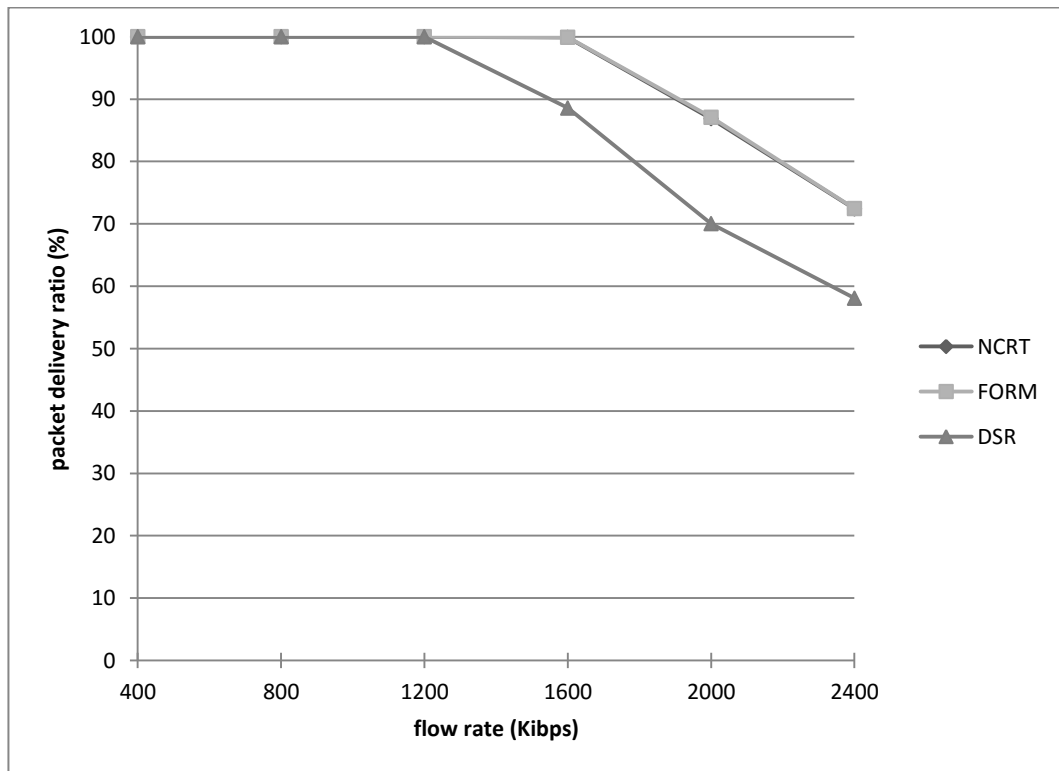
### 3.4.2 Chain Networks

Next, we evaluated NCRT using chain networks that are between two and four hops in length. In these networks, nodes were placed in a straight line with inter-node spacing of 200 meters. One hop was not evaluated as no packet encoding would occur and all protocols would perform equally. There were two flows in each network. The first node in a chain was designated as the source of the first flow and the destination of the second flow, while the last node in the chain was designated as the source of the second flow and the destination of the first flow. The purpose of this test is to provide an insight on the amount of performance gain possible with network coding in favorable conditions where many coding opportunities exist.

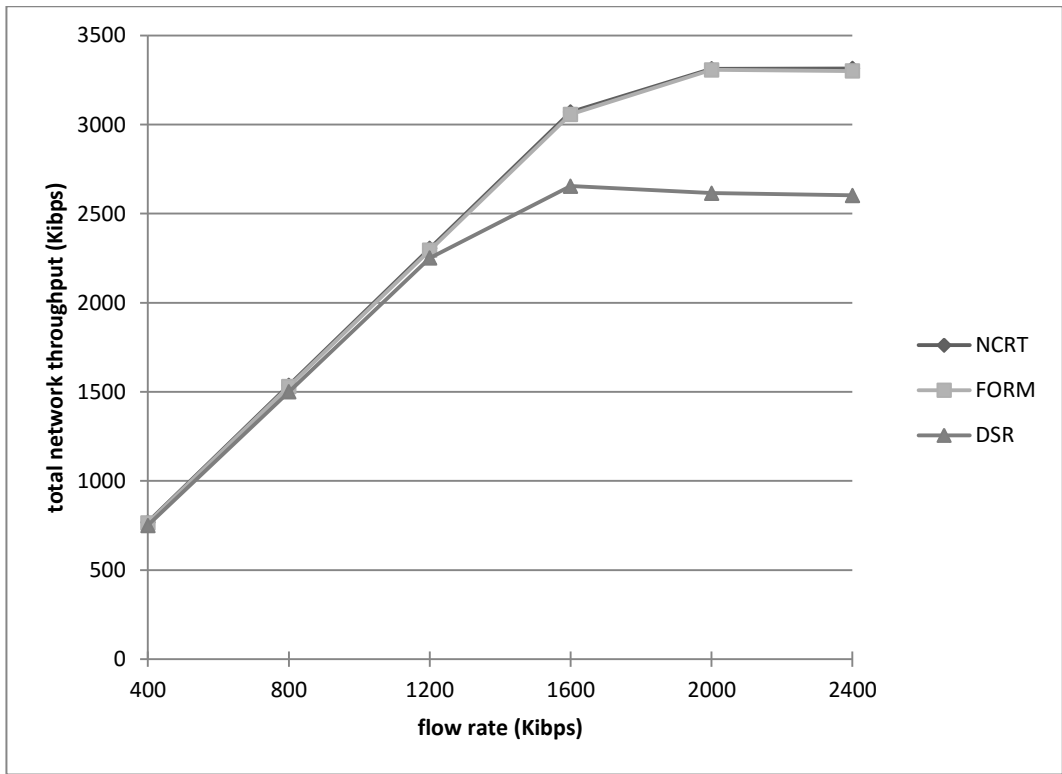
Figures 3.12, 3.13, and 3.14 show the results obtained for the two-, three-, and four-hop chain network, respectively. From the results, it can be observed that NCRT

and FORM performed similarly in packet delivery ratio and total network throughput, and both outperformed DSR, which does not employ network coding. NCRT and FORM performed similarly because chain networks are simple networks. In each of the two flows in a network, only one path is available. Due to the simplicity of these networks, NCRT and FORM also discovered the same coding structures.

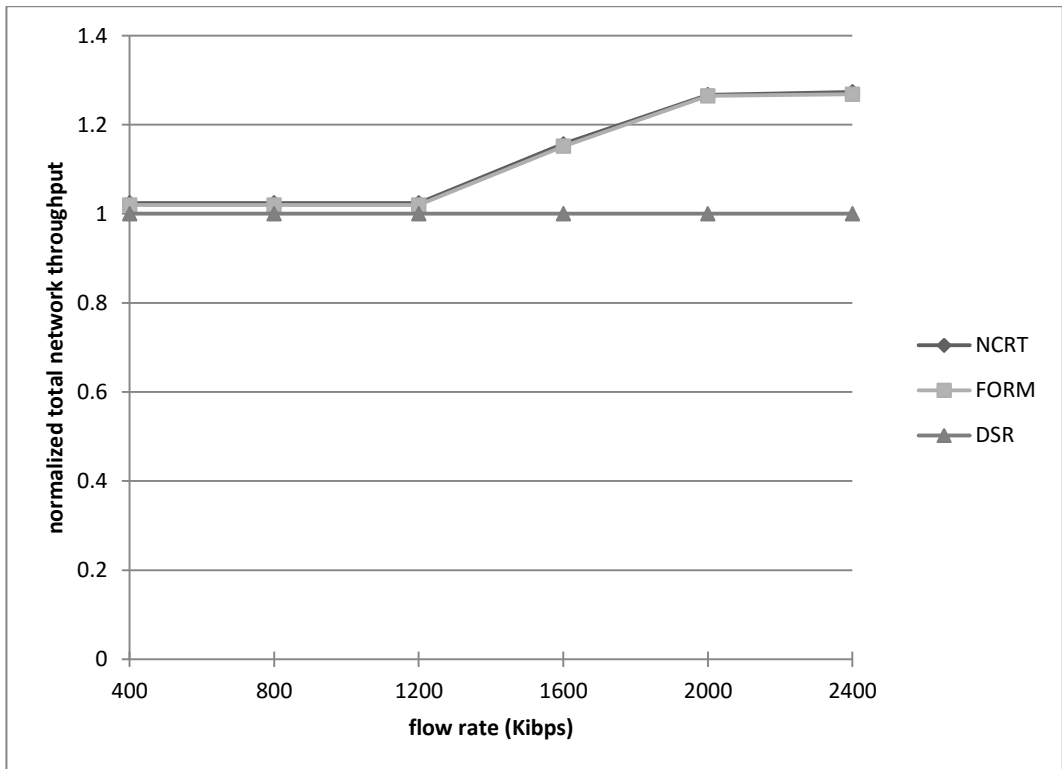
By comparing Figures 3.12b, 3.13b, and 3.14b, it can be observed that the total network throughput decreases as chain length increases. Multi-hop communication requires a packet to be transmitted multiple times from one node to another starting from the source to the destination and this causes increased channel contention and interference.



(a)

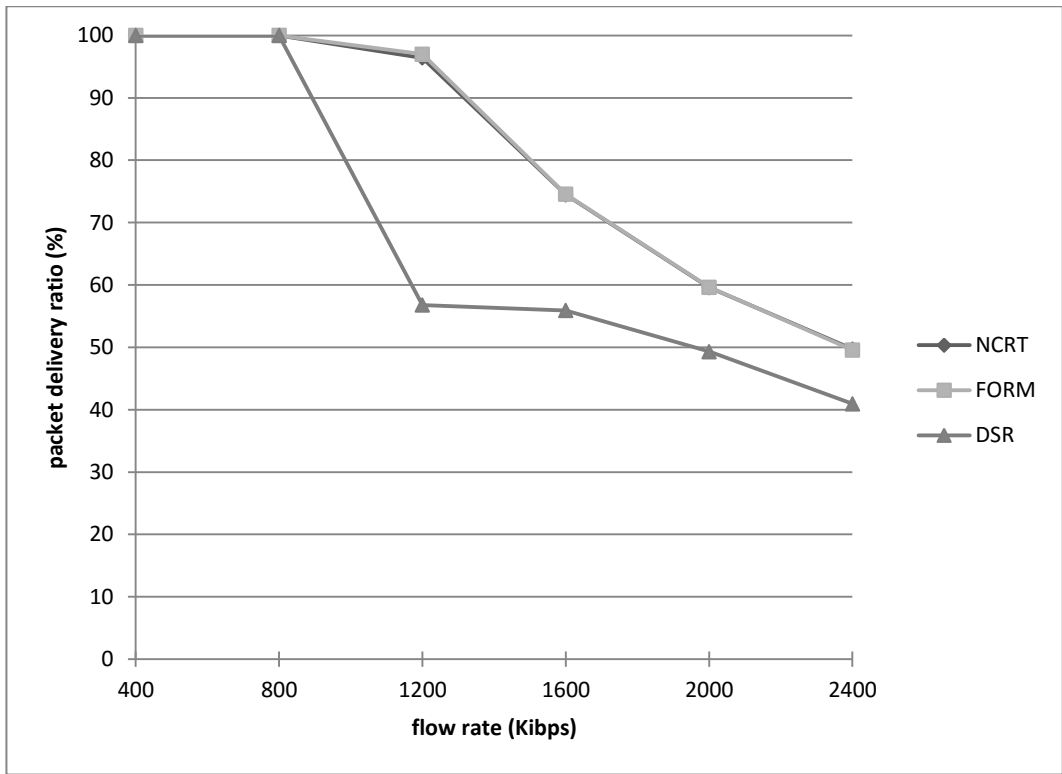


(b)

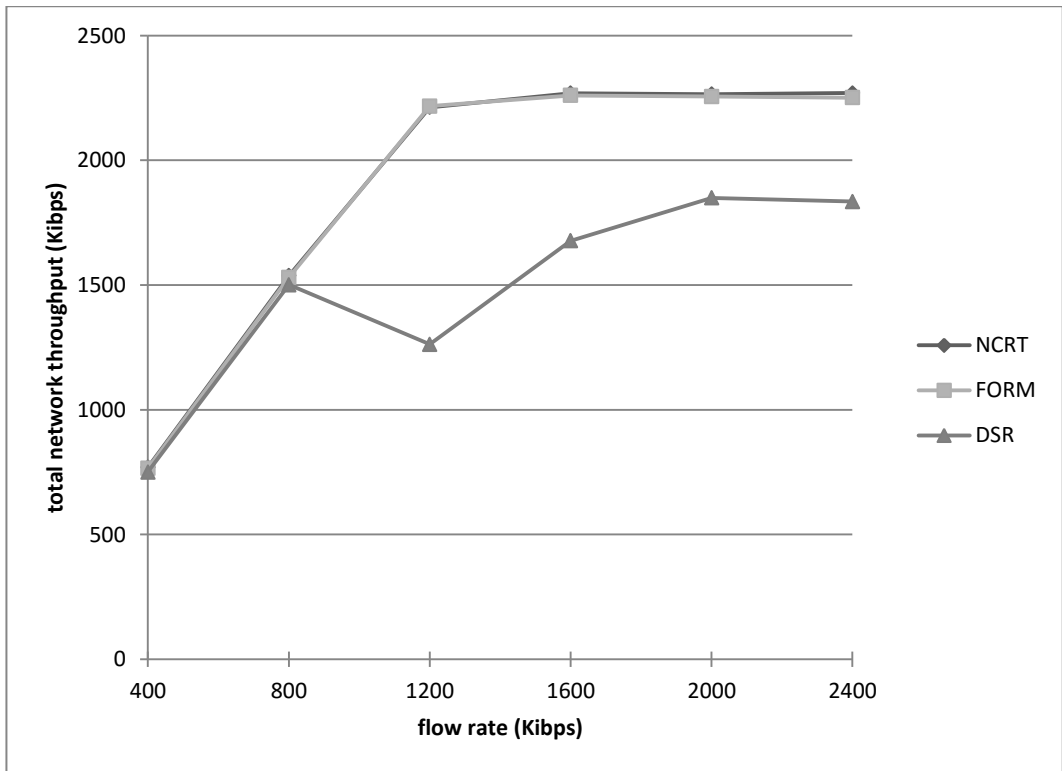


(c)

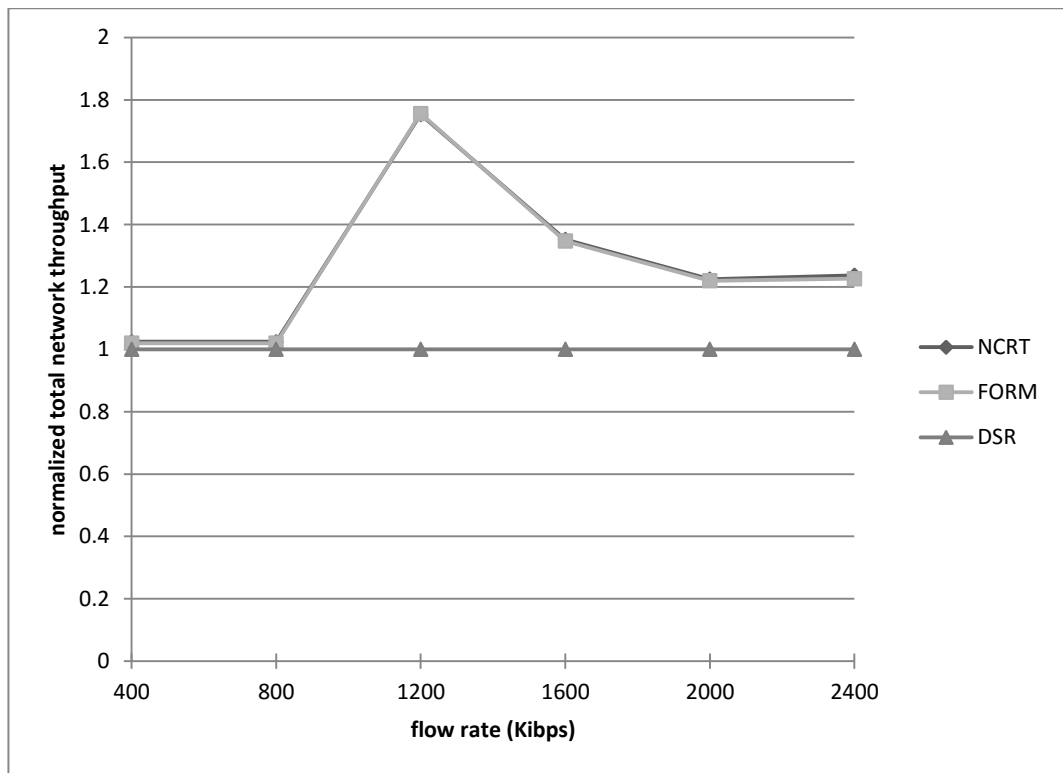
Figure 3.12: Two-hop chain: (a) packet delivery ratio, (b) total network throughput, (c) normalized total network throughput



(a)

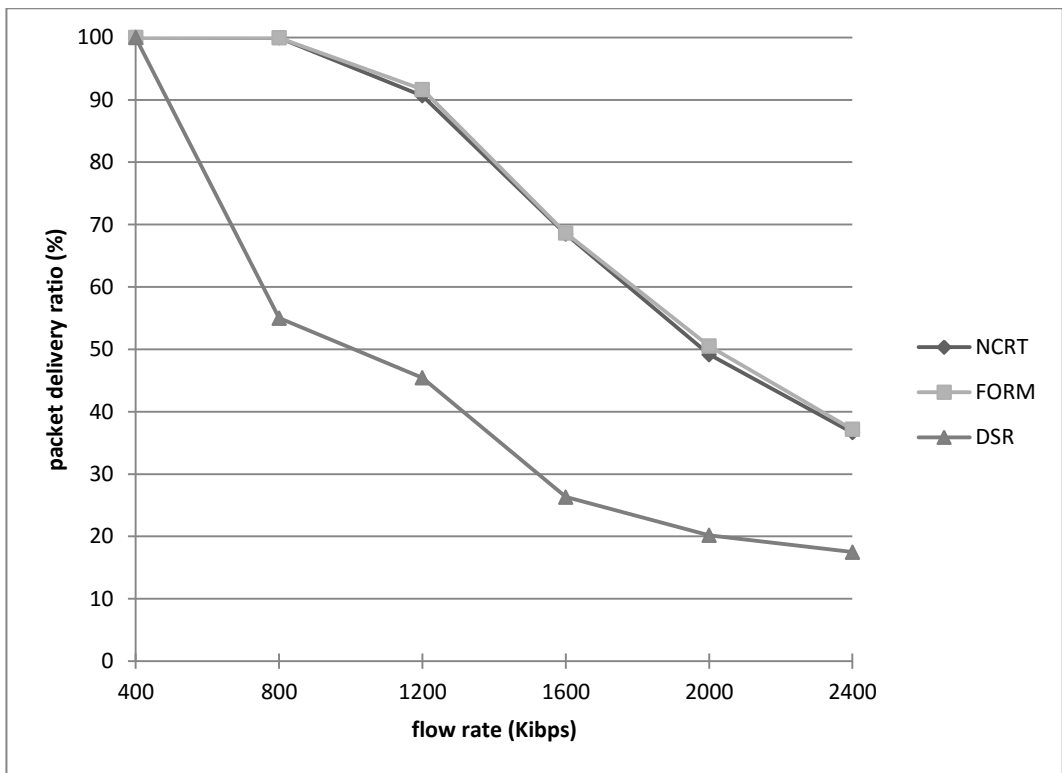


(b)

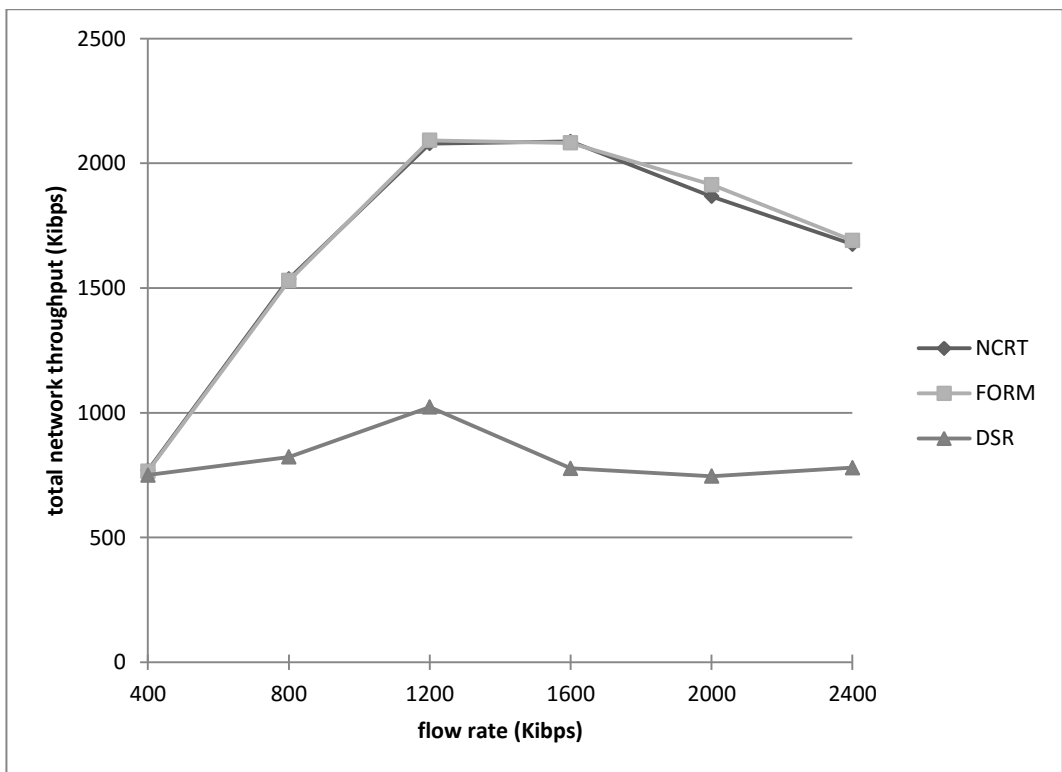


(c)

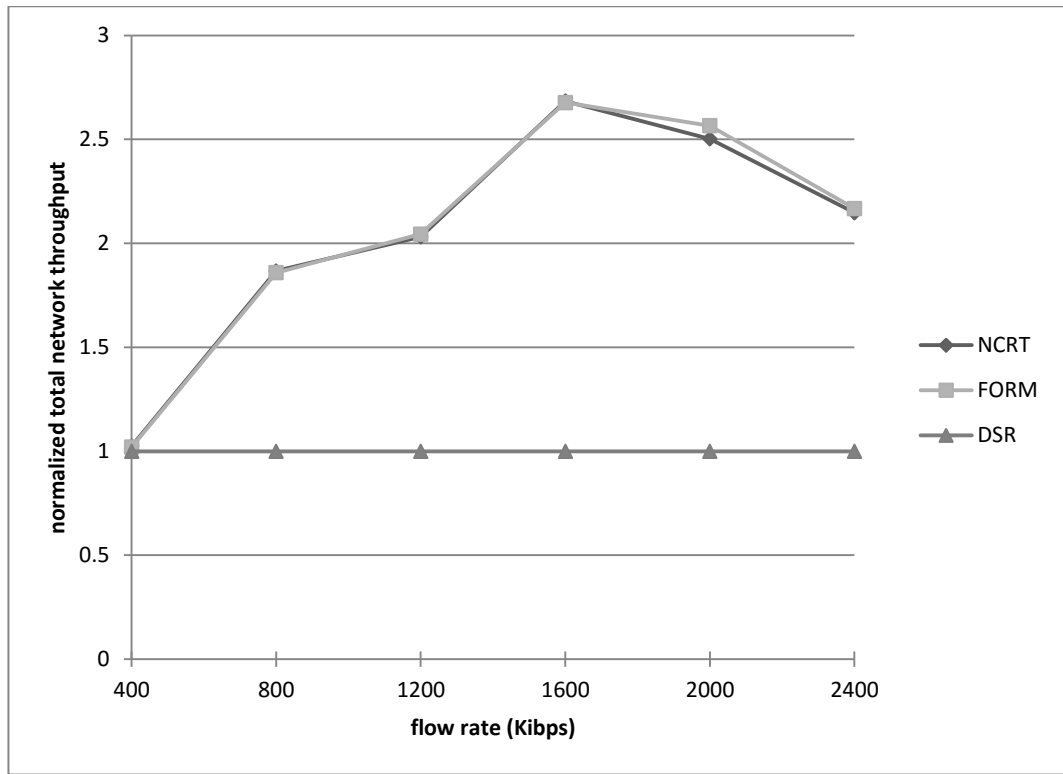
Figure 3.13: Three-hop chain: (a) packet delivery ratio, (b) total network throughput, (c) normalized total network throughput



(a)



(b)



(c)

Figure 3.14: Four-hop chain: (a) packet delivery ratio, (b) total network throughput, (c) normalized total network throughput

### 3.4.3 Grid Network

The grid network had a node spacing of 200 meters in both x- and y-directions and there were five nodes along each of the axes resulting in a topology of 25 nodes in a plane of 800 by 800 square meters. Ten pairs of nodes were randomly selected as the sources and destinations of the flows. Each source was made to start sending 1024 B (1024 B = 1 KiB) packets at a constant rate of 10-50 packet/s in increments of 10 packets/s (or equivalently 80-400 Kibps flow rate) at exactly 10 seconds simulation time and stop sending at exactly 100 seconds simulation time. For each flow rate, we performed 20 simulation runs, and the averages of the results are used in the comparisons.

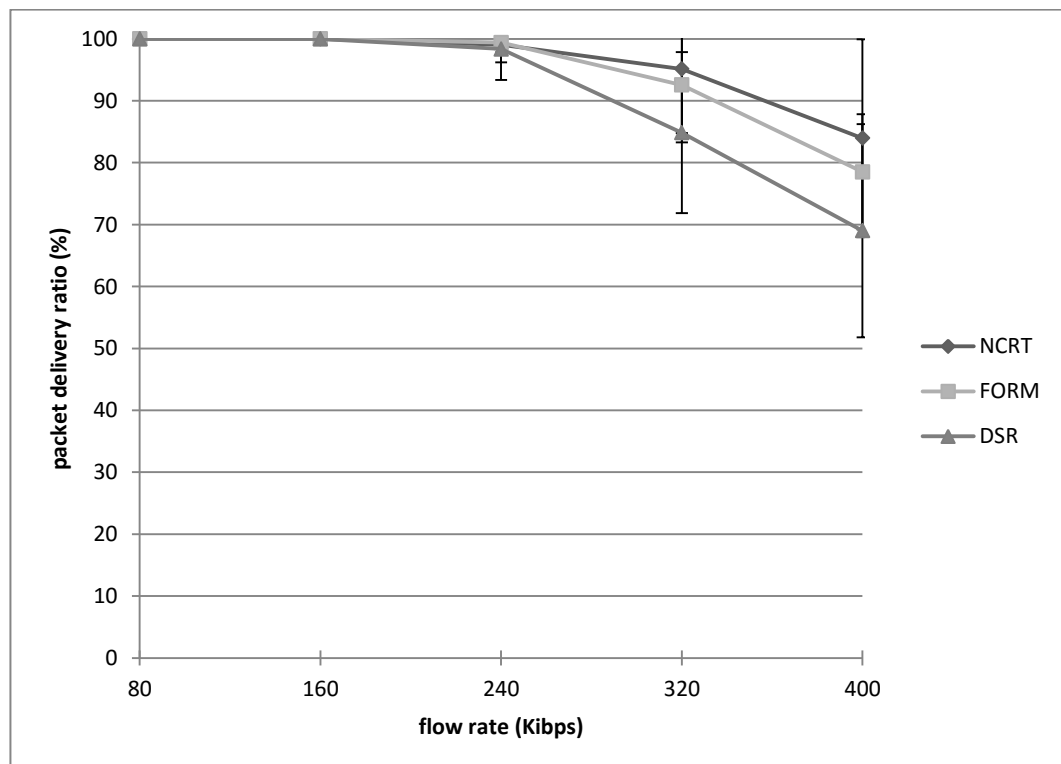
Figure 3.15a shows the packet delivery ratio while Figure 3.15b shows the total network throughput. It can be observed that NCRT outperformed FORM and DSR.



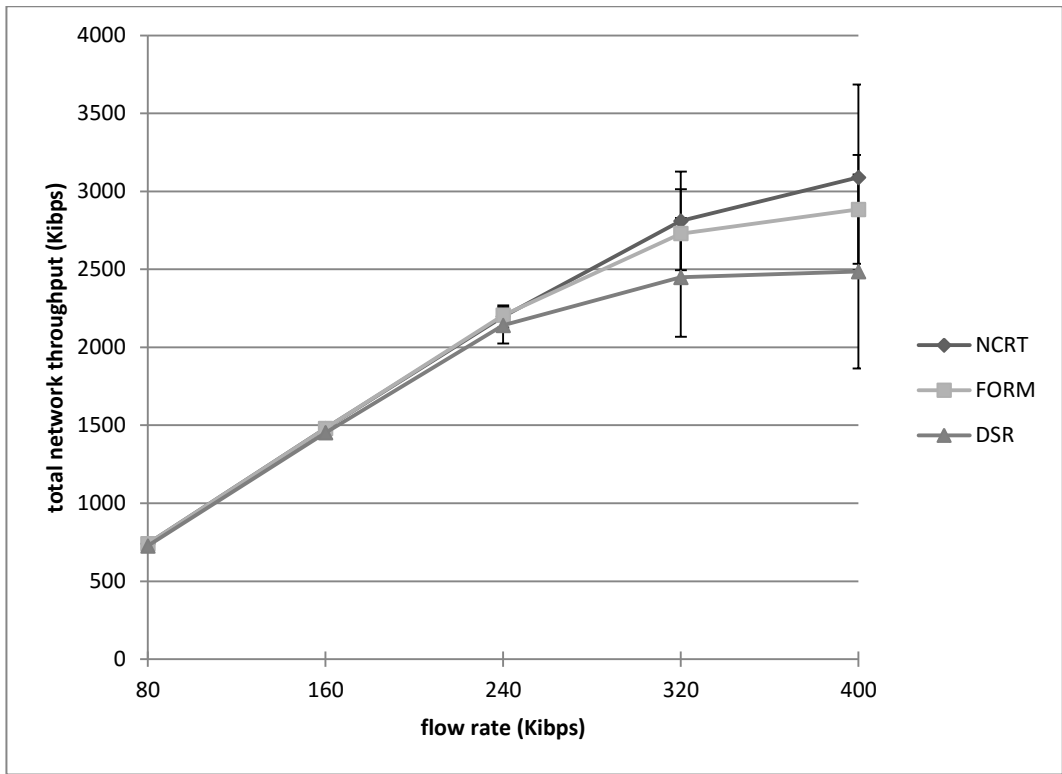
Figure 3.15c shows the normalized total network throughput with respect to DSR. From the figure, we observed that NCRT obtained a maximum gain of about 24% in total network throughput when compared to DSR. On the other hand, FORM managed to obtain only a lower gain of about 16%.

Figure 3.15d shows the average packet delay. From the figure, we observed that NCRT outperformed FORM. We also observed that DSR produced lower packet delays than NCRT and FORM. However, this does not imply that DSR outperformed NCRT and FORM because DSR also obtained lower packet delivery ratio than NCRT and FORM, as shown in Figure 3.15a. In the calculation of average packet delay, only packets that reached their destinations were taken into account as packets that never reach their destinations theoretically have infinite delay.

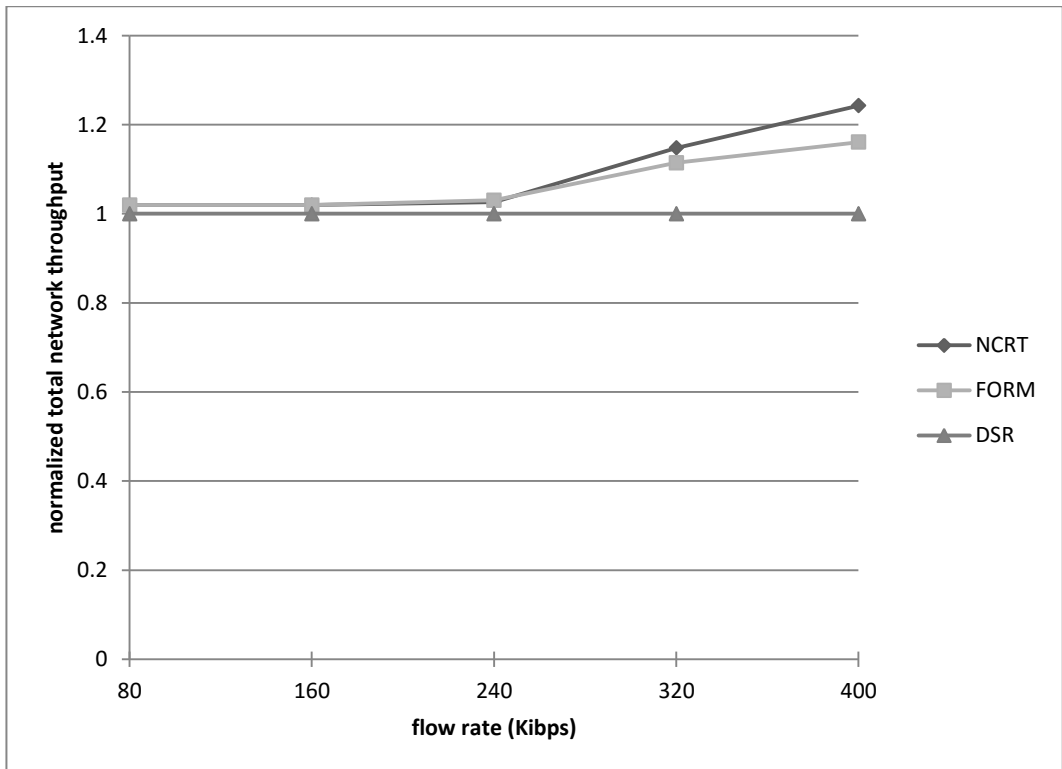
Figure 3.15e shows the number of coding structures while Figure 3.15f shows the number of packet encodings. From these figures, we observed that NCRT not only found more unique ways packets were encoded together (more coding structures) but also produced more packet encodings from the coding structures.



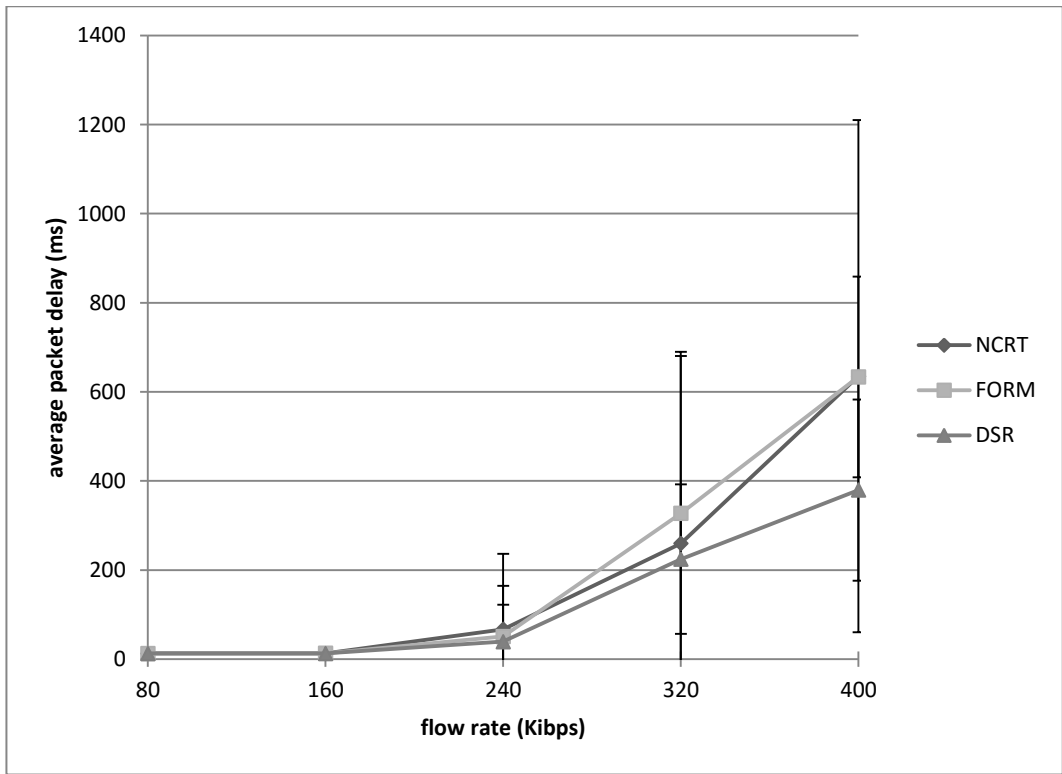
(a)



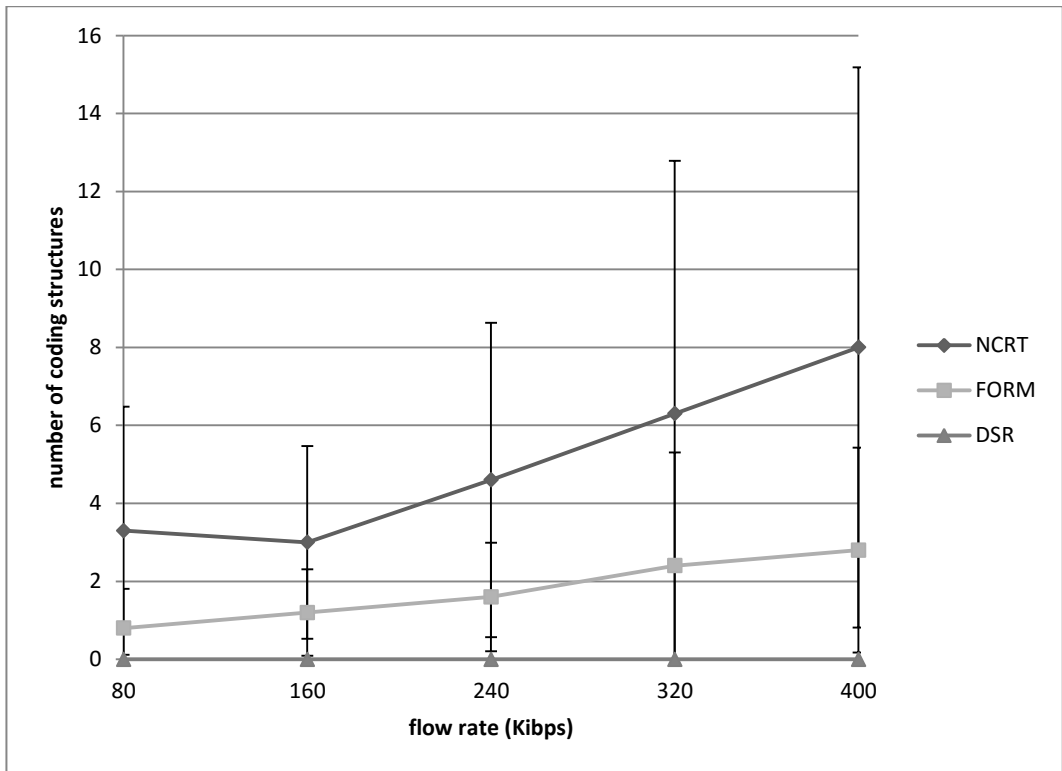
(b)



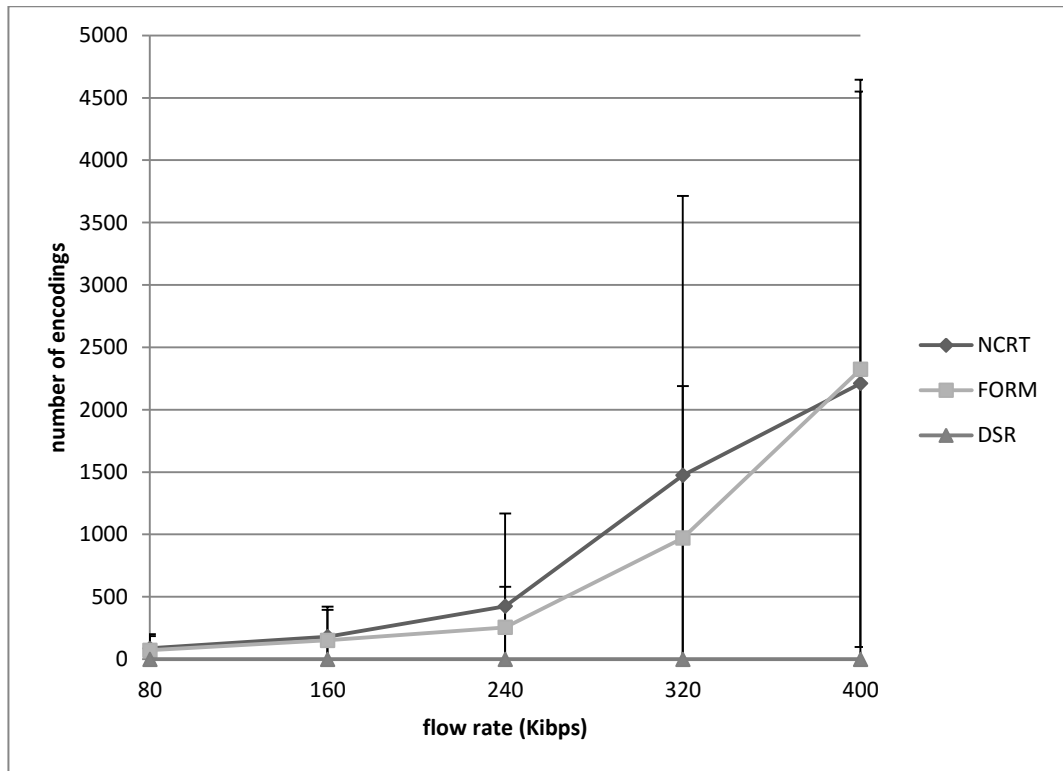
(c)



(d)



(e)



(f)

Figure 3.15: Grid network: (a) packet delivery ratio, (b) total network throughput, (c) normalized total network throughput, (d) average packet delay, (e) number of coding structures, (f) number of encodings

### 3.4.4 Random Networks

In this test, we evaluated NCRT in random networks. Fifty nodes were randomly placed in a plane of 800 by 800 square meters. This aims to create a network that is unlikely to get partitioned as partitioned networks could make analyzing of the results difficult. Ten pairs of nodes were selected randomly as the sources and destinations of the flows. The flow rates used are 80-400 Kibps in increments of 80 Kibps. For each flow rate, the simulation results are averaged over 20 simulation runs.

Figure 3.16a shows the packet delivery ratio while Figure 3.16b shows the total network throughput. We again observed that NCRT outperformed FORM and DSR. Figure 3.16c shows the normalized total network throughput with respect to DSR. From the figure, we observed that NCRT and FORM obtained a maximum gain of about 40%

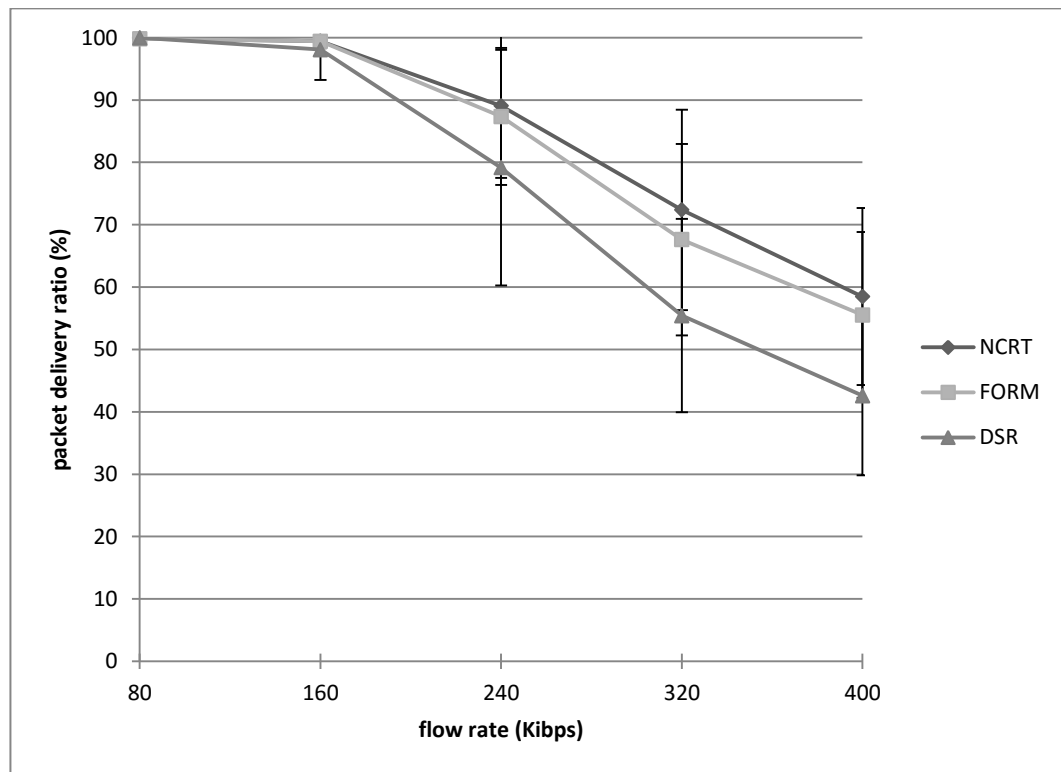
and 33% when compared to DSR, respectively. When comparing Figure 3.15a with Figure 3.16a, and Figure 3.15b with Figure 3.16b, we observed that at higher levels of offered load, the packet delivery ratios and total network throughputs obtained in this test are lower than those obtained in the grid test. This is because the node density in this test is higher than that in the grid test. Besides, randomly placed nodes could also cause a situation where nodes become more concentrated in certain areas in a network. This should not be a problem in the grid network as nodes are distributed regularly. From this test and the grid test, we can also loosely claim that NCRT outperformed FORM not only in low node density networks (the grid test) but also in higher node density networks (this test).

Figure 3.16d shows the average packet delay. From the figure, we observed that NCRT performed comparably to FORM. While NCRT produced slightly higher packet delays than FORM, it should be noted that NCRT also managed to send more packets successfully to their destinations (higher packet delivery ratio). Similar to the grid test, DSR produced the lowest packet delays. We believe that in DSR, more packets were dropped at the network interface queues of bottleneck nodes. Since more packets were dropped in DSR, there are fewer packets in the network and the remaining packets in the network experienced less channel contention and shorter queuing delay.

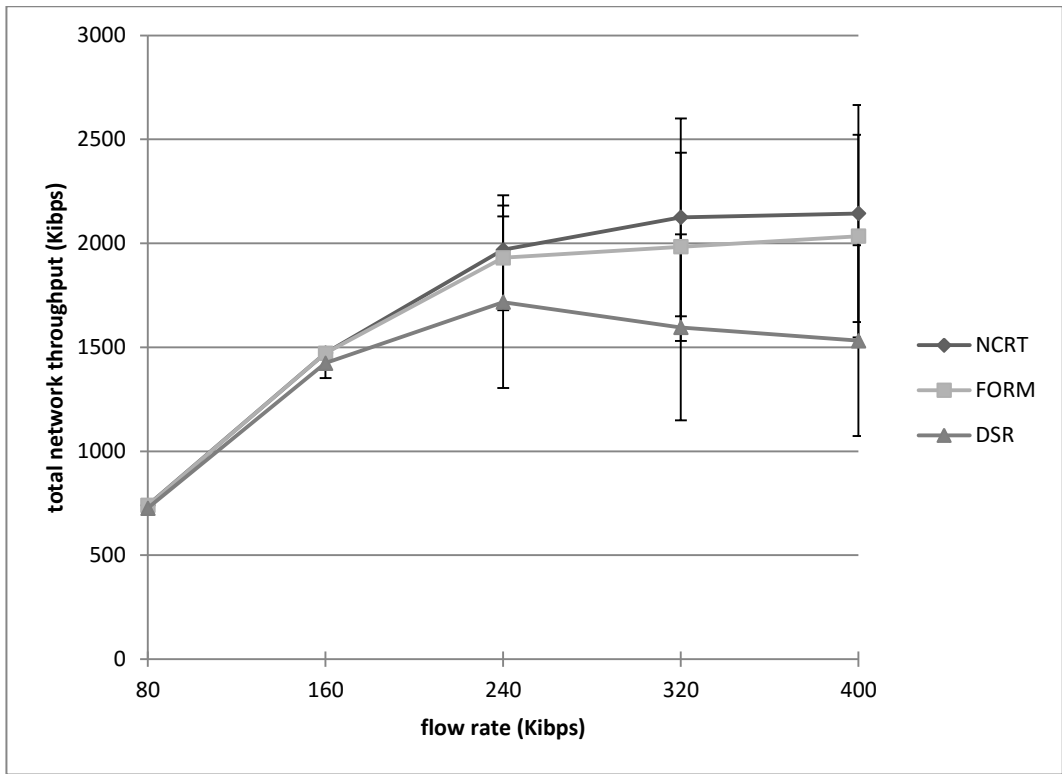
Figure 3.16e shows the number of coding structures. The figure shows that NCRT found more coding structures than FORM. Figure 3.16f shows the number of encodings produced. The figure shows that NCRT produced more packet encodings than FORM.

Figure 3.16g shows the average number of hops of delivered packets. From the figure, we observed that packets traversed shorter paths with NCRT than with FORM. However, NCRT only managed to outperform FORM slightly. This could be due to the largely similar RREQ propagation method used, i.e., nodes drop duplicate RREQs of the same route discovery. Together with Figures 3.16e (number of coding structures),

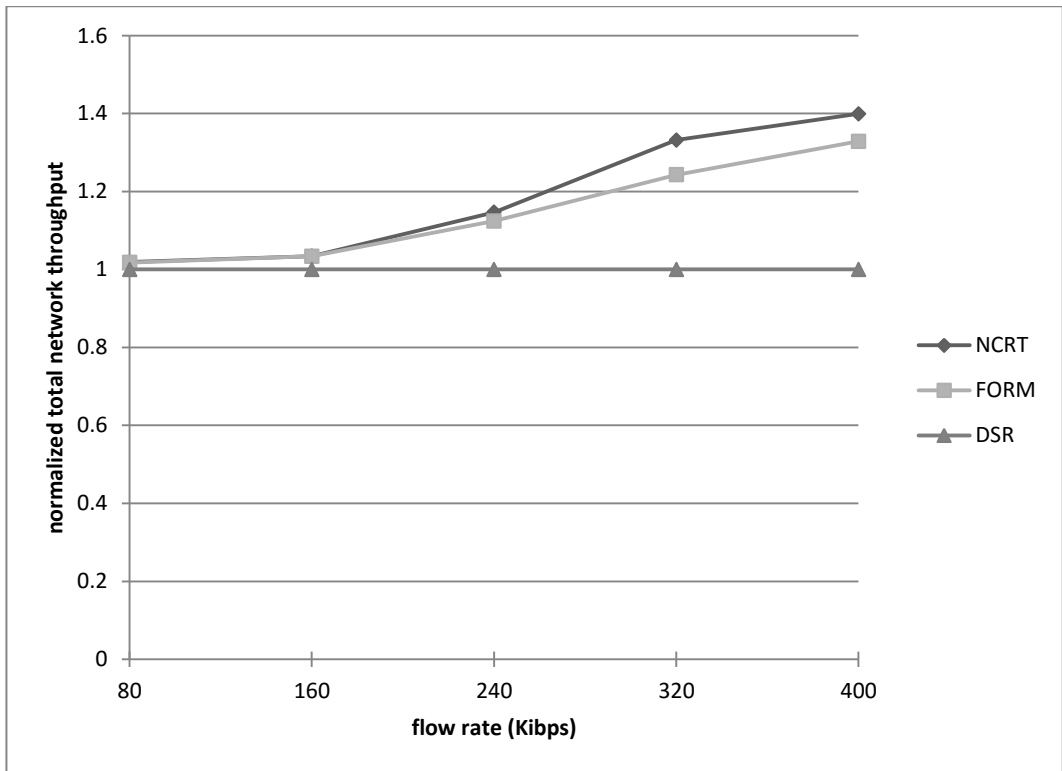
3.16f (number of encodings), and 3.16a (packet delivery ratio), Figure 3.16g shows that the routing metric employed in NCRT, i.e., CLARM, achieved the objective of making source nodes select paths that are short, have many coding opportunities (more coding structures and packet encodings), and do not go through congested areas in the network (higher packet delivery ratio).



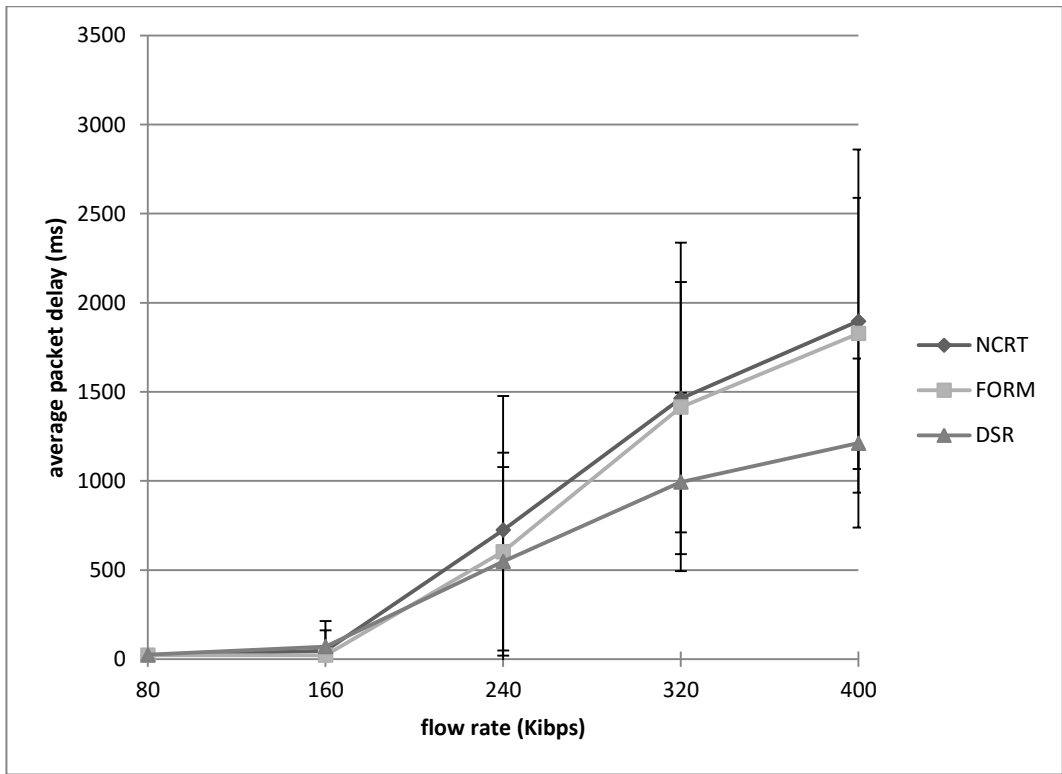
(a)



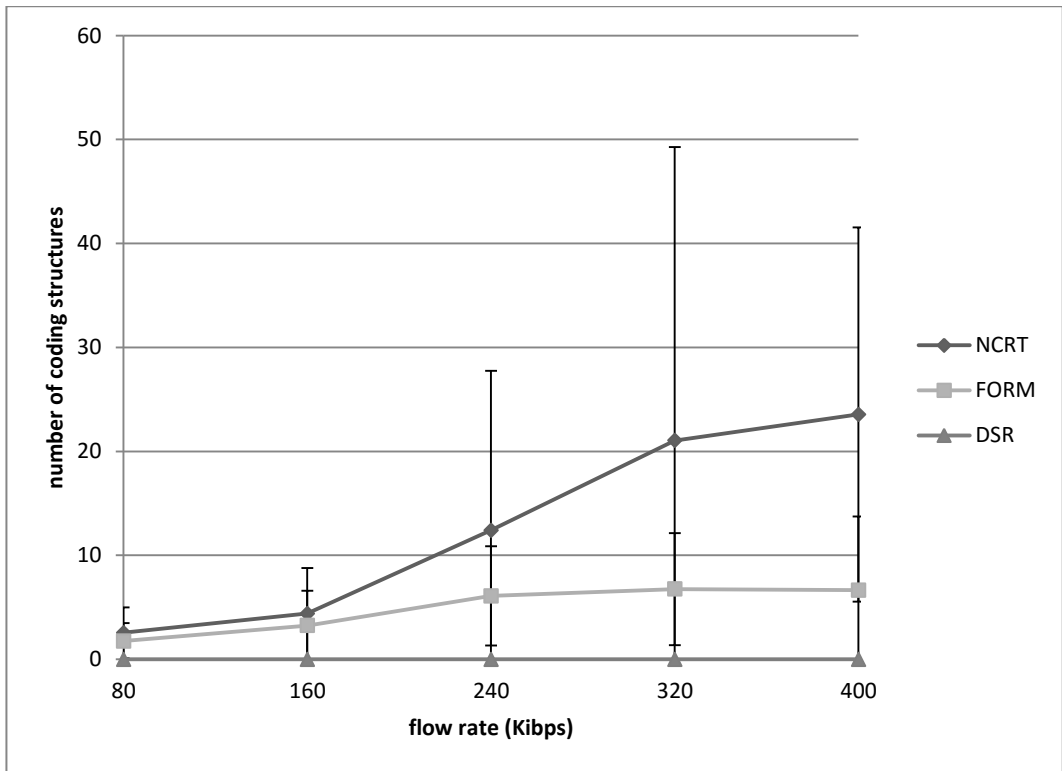
(b)



(c)

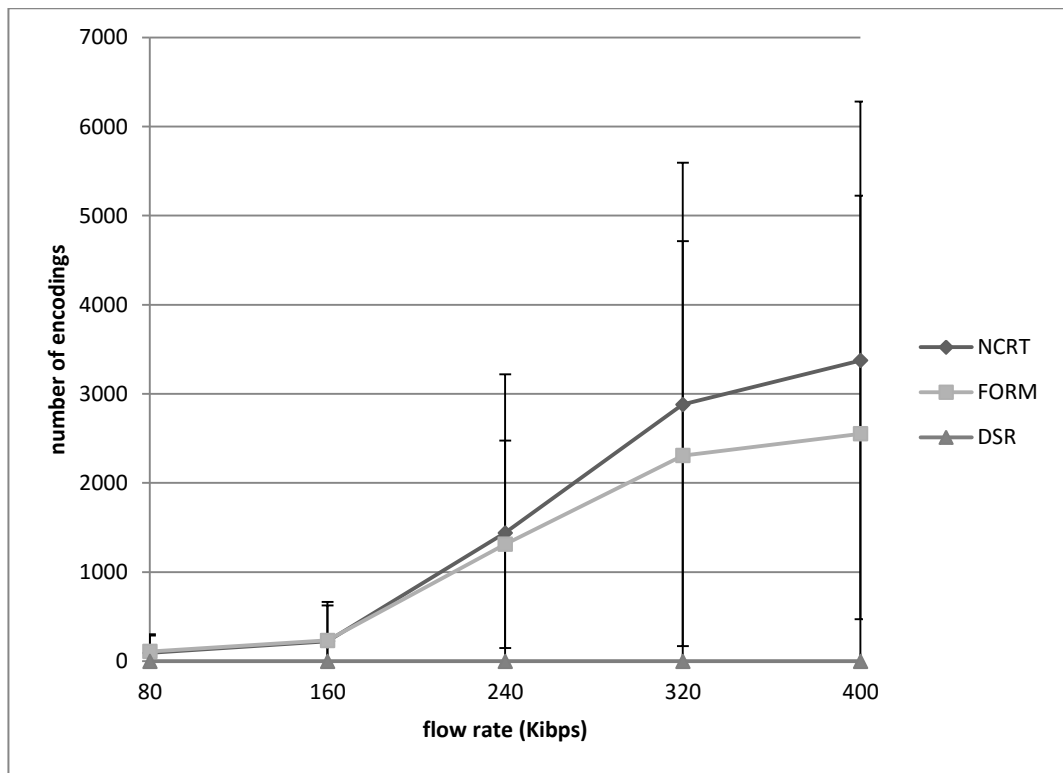


(d)

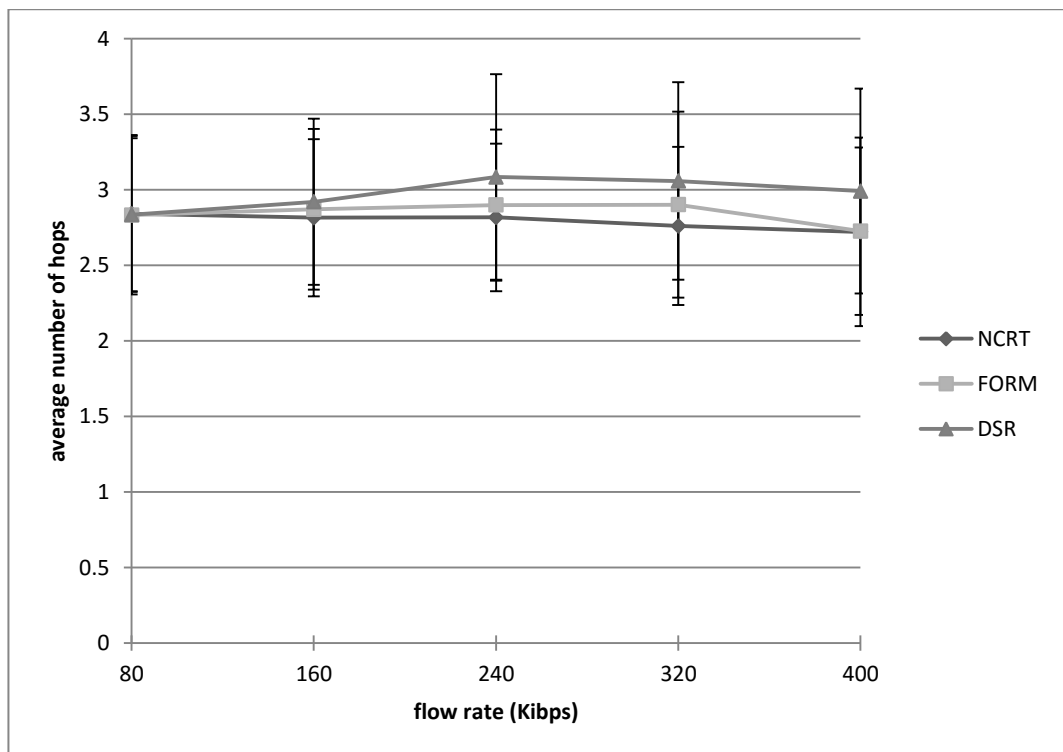


(e)





(f)



(g)

Figure 3.16: Random networks: (a) packet delivery ratio, (b) total network throughput, (c) normalized total network throughput, (d) average packet delay, (e) number of coding structures, (f) number of encodings, (g) average number of hops

### **3.5 Conclusions**

A major issue in wireless ad hoc networks is the amplification of network workload with path length which significantly decreases the maximum supported network throughput as path length increases. Network coding has been proven to be an effective means to reduce the number of transmissions required to deliver a certain amount of packets and can be used to increase the maximum supported network throughput. There are two challenges in deploying network coding in wireless ad hoc networks: (1) to identify which packets can be encoded together, and (2) to integrate the coding conditions into routing protocols. In this chapter, we proposed an ad hoc routing protocol called Network Coding Routing (NCRT), which uses a routing metric based on coding opportunities and network workload to guide source nodes select paths that are short, have many coding opportunities, and does not go through congested nodes. Besides, an enhanced set of coding conditions called Improved Generalized Coding Conditions (IGCC) was also proposed to improve upon existing coding conditions. Through an extensive simulation study, we showed that NCRT improved the network throughput of a wide variety of network configurations when compared to other protocols.

# CHAPTER 4: IMPROVED PARTIAL DOMINANT PRUNING

## BROADCAST PROTOCOL FOR WIRELESS AD HOC

### NETWORKS

#### 4.1 Introduction

Nodes are generally resource-constrained devices with limited processing power, memory, and stored energy. Inefficient use of these resources could lead to devastating consequences. For example, if the energy stored in the nodes is depleted, the network could become partitioned or lose functionality. In this chapter, we seek to reduce redundant transmissions during broadcasting in wireless ad hoc networks.

Broadcasting refers to the process of sending a packet from a source node to all other nodes in a network. It is an important primitive in any communication network as it is often necessary that a piece of information is disseminated throughout the whole network. For instance, in a network deployed for disaster monitoring or weather forecasting, early warning messages are broadcasted to warn others of an impending danger or bad weather. Broadcasting is also commonly used in ad hoc routing protocols such as Ad-hoc On-demand Distance Vector (AODV) (Perkins & Royer, 1999) and Dynamic Source Routing (DSR) (Johnson et al., 2001) for route discovery.

In wireless infrastructure networks, broadcasting is simple and efficient due to the existence of a centralized entity. For instance, in a Wi-Fi network, a node with a packet to send to the rest of the network only needs to send the packet to the router, which then forwards the packet to all other nodes in the network. Assuming no packet loss and packet retransmission, this operation requires only two transmissions because every node in the network is within one hop from the router. In wireless ad hoc networks, broadcasting becomes more complicated. Nodes have limited transmission range and a pair of nodes that wishes to communicate is often located far apart from each other.

Therefore, some intermediate nodes are required to relay the packet. Furthermore, there is no centralized administration and a node has a limited view of the network topology. Under such circumstances, it is difficult to decide which nodes in the network should act as a relay node. On the one hand, it is unnecessary for every node in the network to act as a relay node. On the other hand, some nodes in the network may not receive the packet if a node does not forward the packet when it receives it. One simple solution is to have every node forwards a broadcast packet when it receives the packet for the first time. This is the method used in the blind flooding protocol. While it is simple in operation, it often results in many redundant transmissions. On the contrary, the Partial Dominant Pruning (PDP) (Lou & Wu, 2002) broadcast protocol is one of the most efficient broadcast protocols available. It uses two-hop neighborhood information to reduce redundant transmissions while maintaining packet reachability to all nodes in a network. In this chapter, a broadcast protocol for wireless ad hoc networks is proposed. More specifically, we extend PDP to further improve its efficiency without introducing additional overhead.

The remainder of this chapter is organized as follows. Related work is reviewed in Section 4.2 and preliminaries are given in Section 4.3. The Improved Partial Dominant Pruning (IPDP) broadcast protocol is detailed in Section 4.4. Enhancements for IPDP are detailed in Section 4.5. Section 4.6 presents the evaluation work done and provides a discussion on the results obtained. Finally, we conclude in Section 4.7.

## 4.2 Related Work

Blind flooding (Yi, Gerla, & Kwon, 2003)/simple flooding (Utsu & Ishii, 2010)/pure flooding (Qayyum, Viennot, & Laouiti, 2002)(Hur et al., 2012) is the simplest and most popular broadcast protocol for wireless ad hoc networks. In blind flooding, a receiver forwards a broadcast packet when it receives the packet for the first time (Sasson, Cavin, & Schiper, 2003)(Hur et al., 2012). Subsequent packets from the same broadcast are discarded as they are duplicate packets. Blind flooding is inefficient as it results in many redundant transmissions. Many broadcast protocols for wireless ad hoc networks have been proposed. These protocols can be categorized as probabilistic forwarding protocols (Utsu & Ishii, 2010), counter-based protocols (Utsu & Ishii, 2010), distance and/or location aided protocols (Paruchuri, Durresti, & Jain, 2003)(Arango, Degermark, Efrat, & Pink, 2004)(Liu, Wan, Jia, Liu, & Yao, 2006)(Hur et al., 2012), cluster-based protocols (Kwon & Gerla, 2002)(Yi et al., 2003), and neighbor knowledge protocols (Hyojun Lim & Kim, 2000)(W. Peng & Lu, 2001)(Lou & Wu, 2002)(Qayyum et al., 2002)(Sheng, Li, & Shi, 2005)(Rahman, Endadul Hoque, Rahman, Kundu, & Gburzynski, 2009). A comparison study of some of these protocols can be found in (Williams & Camp, 2002).

In probabilistic forwarding protocols, a receiver of a broadcast packet forwards the packet with some pre-defined probability. However, as packets are forwarded probabilistically, some nodes in the network may not be able to receive the packet. The forwarding probability plays a crucial role in the performance of these protocols. In percolation theory, fluid flow in random media was studied and it was observed that there is a phase transition between having a finite number of clusters and having an infinite cluster, i.e., a large spanning cluster. This transition happens at a probability known as the critical probability. Based on this concept, the critical probability for probabilistic flooding was studied in (Sasson et al., 2003).

In counter-based protocols, redundant transmissions are reduced by counting. A timer is started when a receiver receives a broadcast packet for the first time. The receiver then counts the number of copies of the packet that it receives afterwards. If the count exceeds a pre-defined threshold when the timer fires, the packet is discarded; otherwise, the receiver forwards the packet. When a packet is heard many times by a receiver, it is likely that the receiver's one-hop neighbors have forwarded enough copies of the packet to cover all of the receiver's one-hop neighbors; hence, it is unnecessary for the receiver to forward the packet. Like probabilistic forwarding protocols, there is no guarantee that every node in the network will receive a copy of a particular broadcast packet; therefore, the threshold value used must be chosen carefully. In (Utsu & Ishii, 2010), two load-aware broadcast protocols for wireless ad hoc networks, i.e., Load-aware Dynamic Probabilistic Flooding (LDPF) and Load-aware Dynamic Counter-based Flooding (LDCF), were proposed. The authors showed that LDCF outperforms LDPF and both outperform blind flooding in transmission/reception volume and packet reachability.

Location-aided protocols reduce redundant transmissions using node location information, which can be obtained from Global Positioning System (GPS) sensor. The coverage efficiencies of three regular tiling shapes were studied in (Hur et al., 2012): (1) Model-3 hexagonal tiling, (2) Model-4 square tiling, and (3) Model-5 triangle tiling. Through a mathematical analysis, the Model-3 hexagonal tiling scheme was found to provide the best coverage efficiency out of the three schemes. Based on this result, in the Regular Tiling (RT)-based flooding (Hur et al., 2012) protocol, the nodes closest to the vertices of a virtual overlay plane of hexagonal tiles (the plane of hexagonal tiles resembles a honeycomb) are designated as forwarding nodes. The Optimized Flooding Protocol (OFP) (Paruchuri et al., 2003) is similar to the RT-based flooding protocol in concept. The Geoflood protocol (Arango et al., 2004) uses a method similar to counter-

based protocols to reduce redundant transmissions. A node receiving a broadcast packet waits for an amount of time before deciding whether to forward or discard the packet. The amount of wait time is determined from an equation based on the distance between the node and its previous hop. When the wait time is depleted, taking itself as the origin, the receiver discards the packet if it has received at least one copy of the packet from each of the four quadrants of a circle centered on itself; otherwise, it forwards the packet. An interesting point of this protocol is that it does not require neighborhood information, which is required in many other broadcast protocols.

Local connectivity information is used to reduce redundant transmissions in neighbor knowledge protocols. In the Self-Pruning (SP) (Hyojun Lim & Kim, 2000)(H Lim & Kim, 2001) broadcast protocol, a forwarding node  $u$  records its list of one-hop neighbors on broadcast packets. A receiver  $v$  of a broadcast packet checks if there are new nodes that it can cover if it forwards the packet by comparing its own set of one-hop neighbors,  $N(v)$ , with the set of one-hop neighbors of the previous hop  $u$  recorded on the packet,  $N(u)$ . If there are new nodes that the receiver can cover, i.e., if  $N(v) - N(u) - \{u\} \neq \emptyset$ , the receiver forwards the packet; otherwise, the packet is discarded. However, since there could be several receivers receiving the same packet that could cover a particular node  $w$ , and the receivers make forwarding decisions individually without coordination among themselves, SP may result in many redundant transmissions. Figure 4.1 shows a network with four nodes  $u$ ,  $v_1$ ,  $v_2$ , and  $w$ , where node  $u$  is the source. Ideally, either node  $v_1$  or  $v_2$  should forward the packet, but when SP is used, both nodes forward resulting in one redundant transmission.

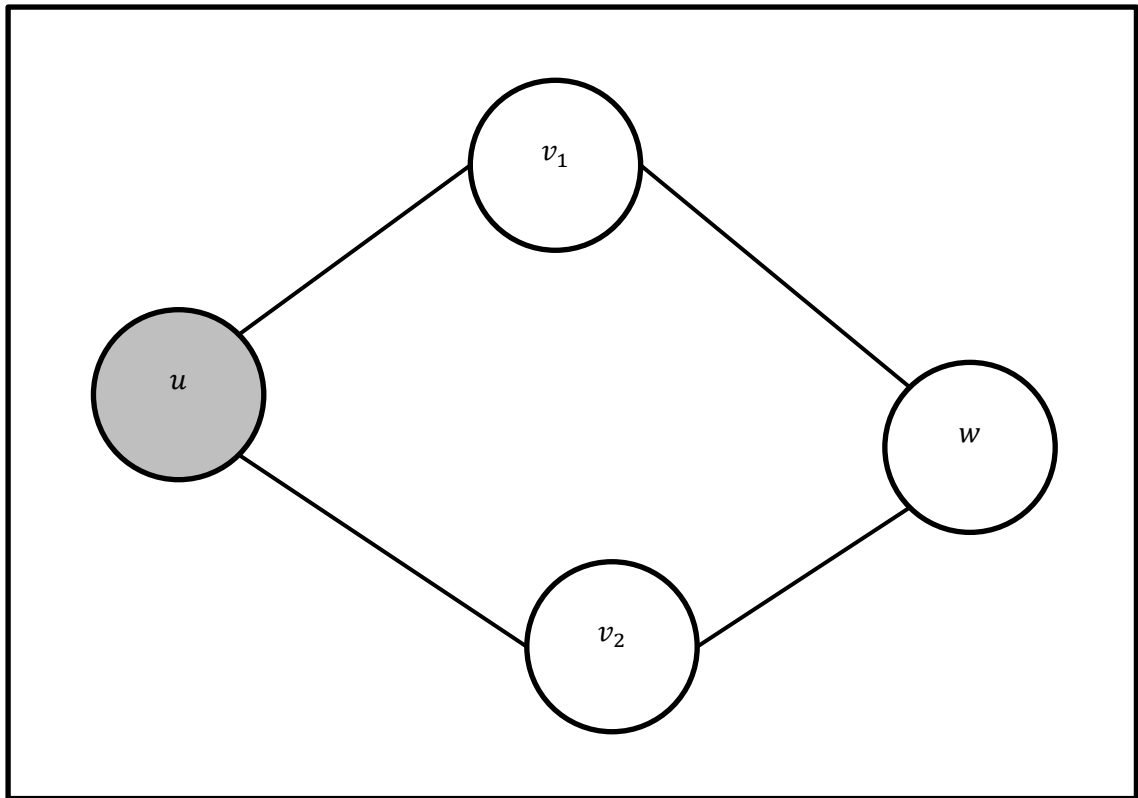


Figure 4.1: Redundant transmission in Self-Pruning

Unlike SP, in the Dominant Pruning (DP) (Hyojun Lim & Kim, 2000)(H Lim & Kim, 2001) broadcast protocol, a node does not decide whether it should forward or discard a broadcast packet. Instead, this decision is made by the previous hop of the packet. A forwarding node  $v$  selects its forwarding nodes by selecting from its one-hop neighbors a minimum set of nodes to cover all nodes in the set of nodes its neighbors must cover, i.e.,  $U(v) = N(N(v)) - N(v) - N(u)$ . Selecting the minimum set of nodes from the one-hop neighbors of  $v$  to cover all nodes in  $U(v)$  is NP-complete (Even, 1979); hence, a forwarding node uses the greedy set cover algorithm (Lovász, 1975) to select its forwarding nodes. Simulation studies showed that DP performed close to the Berman's approximate algorithm (Guha & Khuller, 1998) for the Minimum Connected Dominating Set (MCDS) problem.

Multi-Point Relaying (MPR) (Qayyum et al., 2002) is quite similar to DP (Hyojun Lim & Kim, 2000), but there are several differences between them. First, the set of nodes the neighbors of a considered forwarding node  $v$  must cover, i.e.,  $U(v)$ , is



potentially larger in MPR. Second, the forwarding node selection algorithm employed in MPR is slightly different from that in DP. In (Kadi & Agha, 2008), MPR is combined with network coding.

In addition to two-hop neighborhood information, the Ad Hoc Broadcast Protocol (AHBP) (W. Peng & Lu, 2001) also uses path information to further reduce the size of the set of nodes the neighbors of a considered forwarding node  $v$  must cover, i.e.,  $U(v)$ . However, this comes at the expense of additional overhead as the addresses of the nodes in the path a broadcast packet traversed are recorded on the packet.

The notion of the relative degree of a node, which is the number of one-hop neighbors of a node that are not already covered by any previously selected forwarding node from earlier iterations in a forwarding node selection process, is used in the Relative Degree Adaptive Broadcast (RDAB) (Sheng et al., 2005) protocol. In a forwarding node selection process, the node with the highest relative degree is selected at every iteration until all nodes in the set of nodes the neighbors of a considered forwarding node  $v$  must cover, i.e.,  $U(v)$ , are covered. We found RDAB to be quite similar to DP (Hyojun Lim & Kim, 2000).

It was argued that two-hop neighbor knowledge protocols consider equal transmission range of all nodes and do not work correctly in a network with asymmetric links (Murugesan & Krishnan, 2010). In the Efficient Forward Node List Selection Algorithm (EFNLA) (Murugesan & Krishnan, 2010), nodes discover asymmetric links using REQ packets. A REQ packet would traverse through symmetric links to detect the asymmetric links. In Figure 4.2, due to link asymmetry, node  $u$  does not know that node  $v$  is a neighbor because node  $u$  does not receive HELLO packet from node  $v$ . In EFNLA, node  $u$  could send a REQ packet to discover node  $v$  through links  $(u, w)$  and  $(w, v)$ . However, we found this to be redundant because when node  $w$  broadcasts its HELLO packet, its one-hop neighbors are recorded on the packet. When node  $u$

receives this HELLO packet, it will detect the presence of node  $v$ .

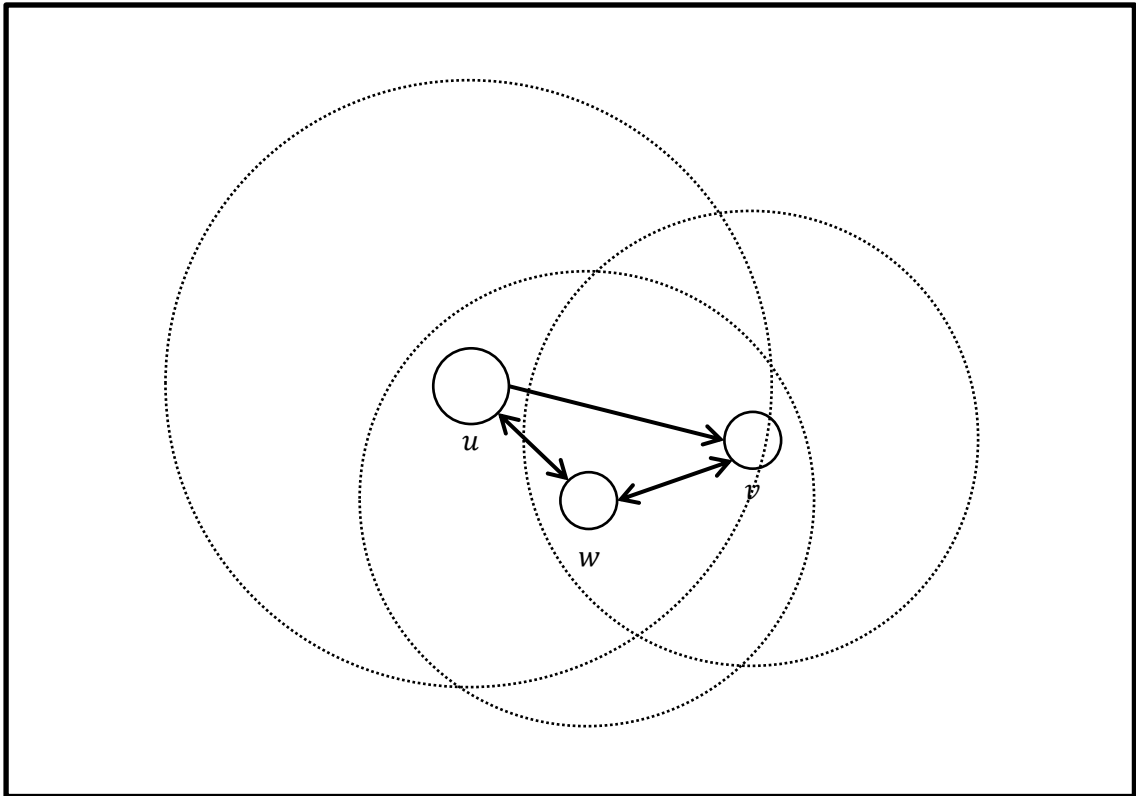


Figure 4.2: EFNLA proposed using REQ packets to discover neighbors that are not discovered due to link asymmetry

In the Total Dominant Pruning (TDP) (Lou & Wu, 2002) broadcast protocol, a considered forwarding node  $v$  prunes the set of nodes its neighboring nodes must cover, i.e.,  $U(v)$ , by nodes that are also in the set of two-hop neighbors of the previous hop  $u$ ,  $N(N(u))$ , resulting in a smaller  $U(v)$  set when compared to DP (Hyojun Lim & Kim, 2000). However, TDP requires that the addresses of the nodes in the set of two-hop neighbors of a forwarding node to be recorded on broadcast packets. This means that three-hop neighborhood information is used in TDP which may not be practical in dense or large networks such as wireless sensor networks (WSNs) as the overhead to record the addresses of the two-hop neighbors of a forwarding node can be quite high. Unlike TDP, the Partial Dominant Pruning (PDP) (Lou & Wu, 2002) broadcast protocol does not require that the addresses of the nodes within two hops from a forwarding node to be recorded on broadcast packets. Based only on two-hop neighborhood information,

PDP can make a considered forwarding node  $v$  identify the nodes in  $N(N(v)) - N(v) - N(u)$  are also in  $N(N(u))$ . The Enhanced Partial Dominant Pruning (EPDP) (Rahman et al., 2009) broadcast protocol, which was extended from PDP, further reduces redundant transmissions by delaying packet forwarding. A forwarding node defers forwarding a received broadcast packet for some time according to its position in the forwarding node list of the packet. Nodes higher in the list offers more coverage and are given lower delay while forwarding nodes lower in the list have higher delay.

Table 4.1 provides a summary of some two-hop neighbor knowledge broadcast protocols.

Table 4.1: Comparison of some two-hop neighbor knowledge broadcast protocols

No	Protocol	The set of nodes the neighbors of a forwarding node $v$ must cover, i.e., $U(v)$	Neighbor knowledge of n-hop	Comments
1	Dominant Pruning (DP)	$N(N(v)) - N(v) - N(u)$	2	-
2	Multi-point Relaying (MPR)	$N(N(v))$	2	-
3	Ad Hoc Broadcast Protocol (AHBP)	$N(N(v)) - \bigcup_{p \in P} (\{p\} \cup N(p))$	2	The addresses of the nodes in $P$ to be recorded on broadcast packets, where $P$ is the path traversed by a broadcast packet.
4	Relative Degree Adaptive Broadcast (RDAB)	$N(N(v))$	2	-
5	Efficient Forward Node List Selection Algorithm (EFNLA)	$N(N(v)) - N(v)$	2	The use of REQ control packets to discover asymmetric links.
6	Total Dominant Pruning (TDP)	$N(N(v)) - N(N(u))$	3	The addresses of nodes within two hops from a forwarding node to be recorded on broadcast packets.
7	Partial Dominant Pruning (PDP)	$N(N(v)) - N(v) - N(u) - N(N(u) \cap N(v))$	2	-
8	Enhanced Partial Dominant Pruning (EPDP)	$N(N(v)) - N(v) - N(u) - N(N(u) \cap N(v)) - \sum_{i=1}^n N(N(w_i) \cap N(v))$	2	Introduces delay in forwarding broadcast packets.

### 4.3 Preliminaries

#### 4.3.1 Network Model and Assumptions

We model a network as an undirected graph  $G(V, E)$ , where  $V$  is the set of nodes in a network, and  $E$  is the set of links in the network. An edge  $e = (i, j)$  is in the graph, i.e.,  $e \in E$ , if  $d(e) = d(i, j) = d(j, i) \leq R$ ,  $i, j \in V$ , where  $d(i, j)$  is the Euclidean distance between nodes  $i$  and  $j$ , and  $R$  is the node transmission range. This model assumes that nodes are equipped with omni-directional antennas and have a common transmission range.

#### 4.3.2 Definitions

The key terms used in this chapter are summarized in Table 4.2.

Table 4.2: Terms and their definitions

Terms	Definitions
$v$	The considered forwarding node.
$u$	The previous hop of $v$ .
$N(v)$	The neighbor set of $v$ .
$N(u)$	The neighbor set of $u$ .
$N(N(v))$	The neighbor set of the neighbor set of $v$ .
$U(v)$	The set of nodes that the neighbors of $v$ must cover.
$F(v)$	The forwarding node set of $v$ . $v$ selects its forwarding nodes from its neighbor set; hence, $F(v) \subseteq N(v)$ .

#### 4.4 Improved Partial Dominant Pruning (IPDP)

In the wireless ad hoc network broadcast problem, the optimal broadcast tree is asked for. Finding the optimal broadcast tree is essentially finding the Minimum Connected Dominating Set (MCDS) of nodes in a network rooted at the source. The MCDS of a network  $G(V, E)$  is the smallest connected set  $V'$  such that a node in  $V - V'$  is connected directly a node in  $V'$ . The MCDS problem is NP-hard; therefore, it is non-trivial to obtain the optimal solution. Besides, finding the optimal solution requires a centralized entity and the use of global information. In a wireless ad hoc network, nodes often belong to different owners and many message exchanges are required for the central entity to obtain global information. Yet, all of this is still considered acceptable if the network is static as message exchange is then required only in the initial phase. However, in a general wireless ad hoc network, nodes are also subject to mobility, which can make gathered information quickly become stale. Thus, centralized solutions are not practical in wireless ad hoc networks. Instead, we resort to approximate solutions that can be implemented easily and in a distributed manner.

Redundant transmissions can be reduced more effectively if nodes have more information. On one end of the scale, if no information is available, nodes can only discard duplicate packets of the same broadcast to reduce redundant transmissions. On the other end of the scale, if all the nodes in a network have complete network topology information, then it is possible for every node to assume the role of the central entity and determine an optimal broadcast tree. We strive to develop a scheme that is practical yet still exhibit high effectiveness in tackling the broadcast redundancy problem. In short, we aim to minimize redundant transmissions in a broadcast subject to the following constraints.

1. Nodes are not allowed to record additional information on a broadcast packet.

2. Nodes are not equipped with GPS sensors and are unaware of their own locations and the locations of the other nodes in the network.

The justification for constraint #1 is to reduce the size of broadcast packets. If packets are smaller, less time is needed to transmit them. Channel contention and interference are also reduced when packets are smaller. By adhering to constraint #1, our proposed protocol improves performance without introducing additional overhead. The justification for constraint #2 is that GPS sensors are costly devices and it is impractical to equip them on some or every node in a network.

Like PDP, Improved Partial Dominant Pruning (IPDP) belongs to the class of neighbor knowledge broadcast protocols and uses two-hop neighborhood information to reduce redundant transmissions. Compared to neighborhood information of higher hop count, two-hop neighborhood information can be easily obtained. In some ad hoc routing protocols, a neighbor discovery scheme is used where nodes periodically broadcast HELLO packets to notify other nodes within their transmission range of their presence. With slight modification, i.e., if nodes record the addresses of their one-hop neighbors in HELLO packets, then a node could obtain neighborhood information of up to two hops. There is some delay involved in obtaining this information as two rounds of HELLO packets are required. In high node mobility scenarios, this information may become stale by the time it reaches two hops. However, the problem of dealing with stale information is not the focus of this work.

We now describe how DP works. Suppose a source node  $x$  has a packet to send to all other nodes in the network. It begins by computing the set  $U(v = x)$ , which is initially populated with all nodes within two hops from itself. Then,  $v = x$  prunes one-hop neighbors of the previous hop  $u$ , i.e., nodes in  $N(u)$ , from  $U(v)$ . This is because when  $v = x$  received the packet from  $u$ , all other nodes within transmission range from  $u$ , i.e., one-hop neighbors of  $u$ , would have received the same packet. However, in this

example, this step is skipped since a source has no previous hop. Next,  $v = x$  prunes its one-hop neighbors from  $U(v = x)$ . This is because when  $v = x$  forwards the packet, all of its one-hop neighbors should receive the packet. The constructed set  $U(v)$  is then used by  $v$  to construct its forwarding node set in the forwarding node selection process.

To ensure that every node in the network can receive a copy of the packet,  $v$  must select forwarding nodes from its neighbors to cover all nodes in  $U(v)$ . Hence, the size of the constructed  $U(v)$  determines indirectly how many forwardings are required to cover all nodes in  $U(v)$ .  $v$  requires fewer one-hop neighbors to forward if  $U(v)$  is smaller, and vice versa. For example, if  $U(v = x)$  is empty,  $v = x$  requires none of its one-hop neighbors to forward the packet when they receive it. Conversely, if  $U(v = x)$  is a large set,  $v = x$  may require some or all one-hop neighbors to forward the packet when they receive it. After its forwarding nodes are determined,  $v = x$  records the addresses of the forwarding nodes in the packet header in a list known as the forwarding node list. Nodes receiving this packet determine if they should forward the packet based on this list. If the address of a receiver is in the list, then the receiver is a forwarding node and should forward the packet; otherwise, the node is not a forwarding node and it discards the packet after usage. The same process is repeated by other forwarding nodes when they receive the packet. Figure 4.3 show how  $U(v)$  is constructed in DP.

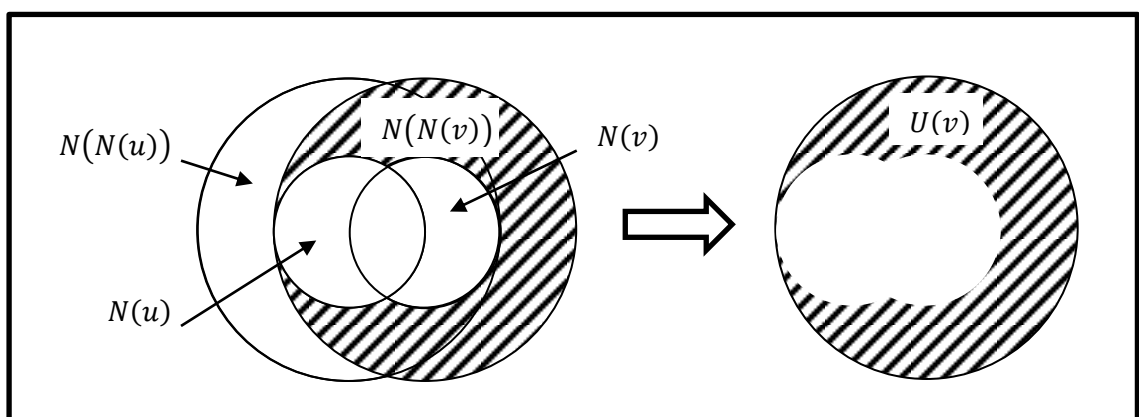


Figure 4.3: Construction of  $U(v)$  in DP



Compared to DP,  $U(v)$  is made even smaller in PDP. When  $v$  receives a broadcast packet, some of the nodes in  $U_{DP}(v) = N(N(v)) - N(v) - \{u\}$  (note: different protocols differ in the way  $U(v)$  is constructed) has already been considered by  $u$ . In other words, if  $v$  and all other forwarding nodes belonging to the same packet forward the packet, all nodes in  $N(N(u))$  will be covered. Hence, ideally,  $v$  only needs to include in  $U(v)$  nodes within two hops from itself that are not in  $N(N(u))$ , i.e.,  $U(v) = N(N(v)) - N(N(u))$ , as illustrated in Figure 4.4.

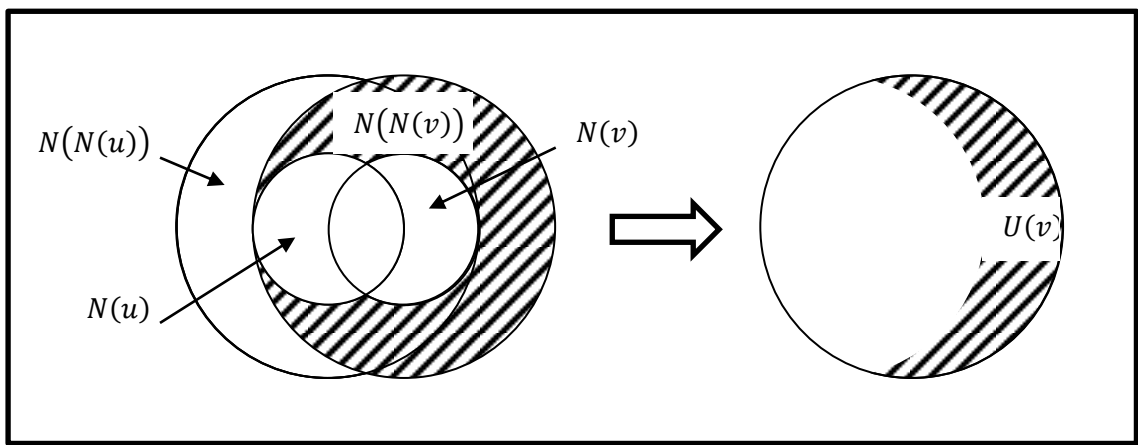


Figure 4.4: Ideally,  $v$  should only include in  $U(v)$  nodes within its two-hop region that are not in  $N(N(u))$

However, if forwarding nodes are not allowed to record the addresses of their two-hop neighbors in broadcast packets,  $v$  cannot identify the nodes in  $U(v) \cap N(N(u))$ . PDP solves this problem by deducing these nodes using only two-hop neighborhood information. This is done by making  $v$  prune from  $U(v)$  nodes that are one-hop neighbors of nodes that are both one-hop neighbor of  $v$  and its previous hop  $u$ ,  $N(N(u) \cap N(v))$ , as shown in Figure 4.5.  $v$  can prune the nodes in  $N(N(u) \cap N(v))$  from  $U(v)$  because they reside in  $U(v) \cap N(N(u))$ . Compared to DP,  $U(v)$  is potentially smaller in PDP as some nodes can be pruned from  $U(v) = N(N(v)) - N(u) - N(v)$  if  $N(N(u) \cap N(v)) \neq \emptyset$ . By comparing Figures 4.3 and 4.5, it can be

seen that  $U(v)$  in PDP is potentially smaller than  $U(v)$  in DP.

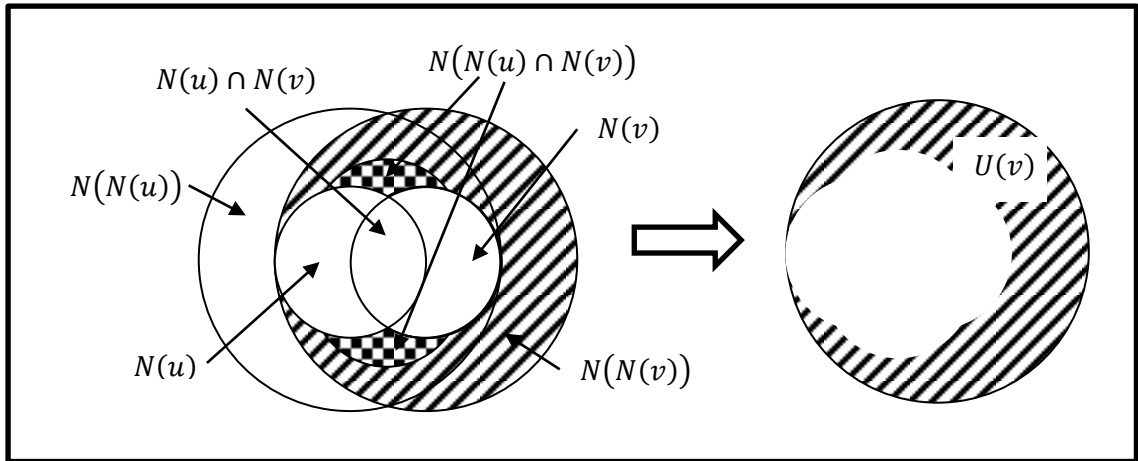


Figure 4.5: Construction of  $U(v)$  in PDP

In DP and PDP, a forwarding node selects its forwarding nodes individually without being aware of other forwarding nodes of the same packet. Suppose nodes A and B are two forwarding nodes selected by node S. When node A receives the packet from S, node A selects its forwarding nodes without considering that node B is another forwarding node selected by node S. Similarly, when node B receives the packet from node S, node B selects its forwarding nodes without considering that node A is another forwarding node selected by node S. As a result, there may be overlap of coverage in  $U(A)$  and  $U(B)$ . In other words, the intersection of the  $U$  sets of two forwarding nodes  $v$  and  $w$  of the same packet could be non-empty, i.e.,  $U(v) \cap U(w) \neq \emptyset$ ,  $v, w \in F(u)$ . We propose the Improved Partial Dominant Pruning (IPDP) broadcast protocol to tackle this problem to further improve the performance of PDP.

In IPDP, instead of making decision individually, the forwarding nodes of a packet make decision “collectively” using only existing information, i.e., information available to a forwarding node in PDP. Therefore, IPDP does not introduce additional overhead over PDP. In PDP and other similar protocols, a forwarding node records the addresses of its selected forwarding nodes in the forwarding node list on a broadcast packet. Figure 4.6 shows the forwarding node list.

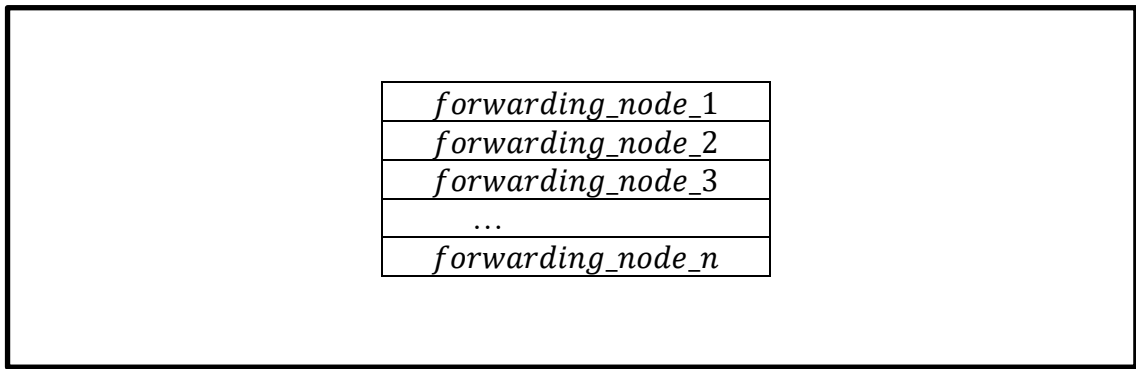


Figure 4.6: Forwarding node list

By making use of the forwarding node list, every forwarding node of a packet and in fact every node receiving the packet would be able to determine the forwarding nodes of the packet. Consider the network in Figure 4.7a and the forwarding node list in the packet transmitted by the source (node 0) shown in Figure 4.7b. When node 1 receives the packet from node 0, node 1 can be aware that node 2 is another forwarding node selected by node 0 because node 1 can find the address of node 2 in the forwarding node list of the packet. Similarly, when node 2 receives the packet, it can be aware that node 1 is another forwarding node selected by node 0. Using the forwarding node list, coordination among forwarding nodes of a packet can be enforced without explicit communication among themselves.

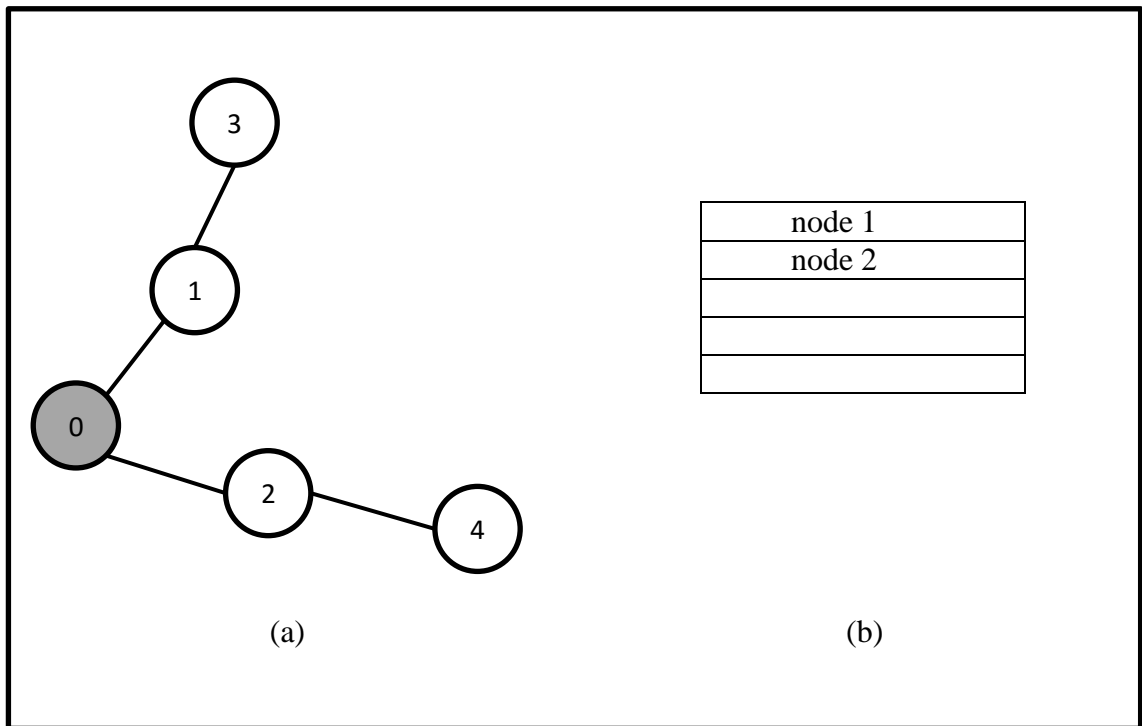


Figure 4.7: (a) an example network, (b) the forwarding node list of the packet transmitted by node 0

Since all the forwarding nodes of a packet are selected by the same previous hop  $u$ ,  $N(v) \cap N(w) \neq \emptyset$ . At the very least,  $N(v) \cap N(w) = \{u\}$ . However, if  $|N(v) \cap N(w)| > 1$ , then the nodes in  $N(N(v) \cap N(w))$  can be pruned from either  $U(v)$  or  $U(w)$ . Figure 4.8a shows the overlap of the  $U$  sets of two forwarding nodes  $v$  and  $w$  of the same packet. In Figure 4.8b, the center of the red circle is located in the area where  $N(v) \cap N(w)$ . It can be observed that this red circle covers some of the area in  $U(v) \cap U(w)$ . Hence, redundancy can be reduced by having the nodes in  $N(N(v) \cap N(w)) \cap (U(v) \cap U(w))$  to be in either  $U(v)$  or  $U(w)$ .

From the above, we saw that nodes in  $N(N(v) \cap N(w))$  can be pruned from  $U(v)$  or  $U(w)$ . In IPDP, we use the addresses of the nodes as their priority and prune the  $U$  set of the forwarding node with the higher priority. Other priority tie-breaking schemes can also be used, for example, using the position of a forwarding node in the forwarding node list as the priority. With this method, a node higher up in the list has higher

priority than a node lower down in the list. In IPDP,  $U(v)$  can finally be written as

$$U(v) =$$

$$N(N(v)) - N(v) - N(u) - N(N(u) \cap N(v)) -$$

$$\cup_{addr(v) < addr(w), w \neq v, w \in F(u)} N(N(v) \cap N(w)).$$

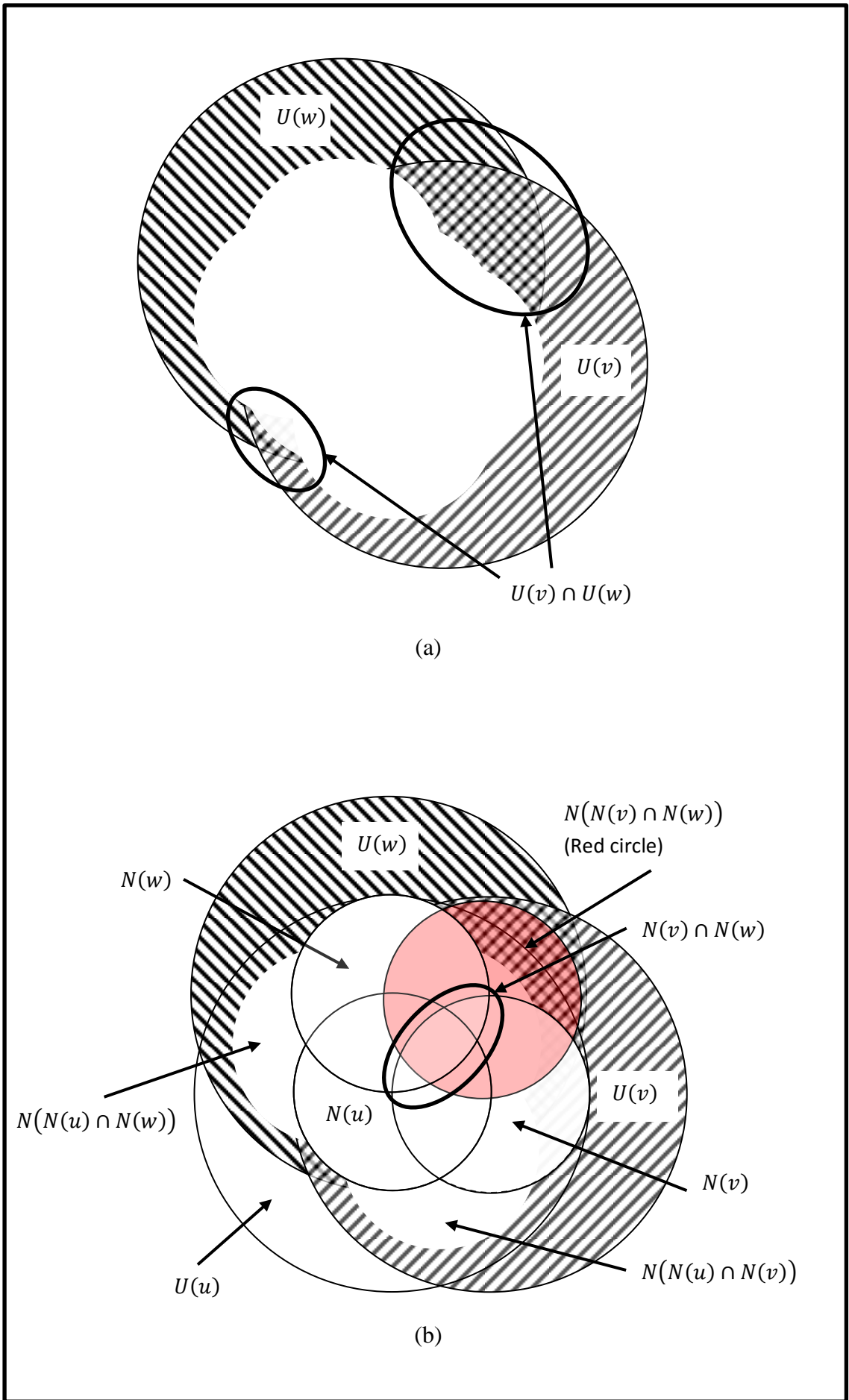


Figure 4.8: Reducing redundancy in IPDP

In IPDP, the steps for determining  $U(v)$  at  $v$  are summarized as follows:

1. Populate  $U(v)$  with nodes within two hops from  $v$ .  $U(v) \leftarrow N(N(v))$ .
2. From  $U(v)$ , prune one-hop neighbors of  $u$ .  $U(v) \leftarrow U(v) - N(u)$ . This is because one-hop neighbors of  $u$  were covered by  $u$ .
3. From  $U(v)$ , prune one-hop neighbors of  $v$ .  $U(v) \leftarrow U(v) - N(v)$ . This is because the one-hop neighbors of  $v$  will be covered when it forwards regardless whether they are in  $U(v)$  or not.
4. From  $U(v)$ , prune one-hop neighbors of nodes that are one-hop neighbor of both  $u$  and  $v$ .  $U(v) \leftarrow U(v) - N(N(u) \cap N(v))$ . This is because nodes in  $N(N(u) \cap N(v)) \subseteq N(N(u))$ . Recall that nodes in  $N(N(u))$  will be covered when all forwarding nodes in  $F(u)$  receive the packet and forward; therefore, we do not want these nodes to be in  $U(v)$ .
5. For all  $w \in F(u)$ ,  $w \neq v$ , if  $addr(v) < addr(w)$ , from  $U(v)$ , prune one-hop neighbors of one-hop neighbors of  $v$  that has  $w$  as their one-hop neighbor.  $U(v) \leftarrow U(v) - N(N(w) \cap N(v))$ . This assumes that the addresses of the nodes are used for priority differentiation.

**Theorem 1**

Let  $U(f) = N(N(f)) - N(u) - N(f) - N(N(u) \cap N(f))$ , where  $f \in F(u)$ , and  $F(u)$  is the set of forwarding nodes selected by the common previous hop  $u$  of two forwarding nodes  $v, w \in F(u)$ , then  $N(N(v) \cap N(w))$  can be pruned from either  $U(v)$  or  $U(w)$  and collectively  $U(v)$  and  $U(w)$  would still contain all the nodes in  $U(v) \cup U(w)$ .

**Proof**

$$U(v) \cup U(w) = U(v) + U(w) - (U(v) \cap U(w))$$

Rearranging, we have:

$$\begin{aligned} U(v) \cup U(w) &= U(v) + (U(w) - (U(v) \cap U(w))) \\ &= (U(v) - (U(v) \cap U(w))) + U(w) \end{aligned}$$

Observe that:

$$\begin{aligned} N(N(v) \cap N(w)) \cap U(v) &= N(N(v) \cap N(w)) \cap U(w) \\ &= N(N(v) \cap N(w)) \cap (U(v) \cap U(w)) \subseteq U(v) \cap U(w). \end{aligned}$$

Then,

$$\begin{aligned} &U(v) + (U(w) - (N(N(v) \cap N(w)))) \\ &= U(v) + (U(w) - (N(N(v) \cap N(w)) \cap U(w))) \\ &= U(v) + (U(w) - (N(N(v) \cap N(w)) \cap (U(v) \cap U(w)))) \\ &\supseteq U(v) + (U(w) - (U(v) \cap U(w))) = U(v) \cup U(w) \end{aligned}$$

Similarly,

$$(U(v) - (N(N(v) \cap N(w)))) + U(w) \supseteq U(v) \cup U(w)$$

Hence, nodes in  $N(N(v) \cap N(w))$  can be pruned from either  $U(v)$  or  $U(w)$  and collectively  $U(v)$  and  $U(w)$  would still contain all the nodes in  $U(v) \cup U(w)$ .



After  $U(v)$  has been determined,  $v$  uses it to determine its forwarding node set,  $F(v)$ . It is non-trivial for a forwarding node to determine its optimal set of forwarding nodes. For example, for a forwarding node with five one-hop neighbors, there are  ${}^5C_0 + {}^5C_1 + {}^5C_2 + {}^5C_3 + {}^5C_4 + {}^5C_5 = 32$  possible combinations. For a forwarding node with 10 one-hop neighbors, there are  ${}^{10}C_0 + {}^{10}C_1 + \dots + {}^{10}C_9 + {}^{10}C_{10} = 1,024$  possible combinations. Similarly, for a node with 20 one-hop neighbors, there are 1,048,576 possible combinations! Complexity is  $2^n$ , where  $n$  is the number of one-hop neighbors the considered forwarding node has. Hence, a greedy algorithm is used in the selection process. The greedy set cover algorithm used is given in Algorithm 4.1.

---

Input:  $U(v), N(v)$   
Output:  $F(v)$

---

1. Initialization.  $F(v) = \emptyset$
  - 2.
  3. While  $(U(v) \neq \emptyset)$ 
    - a. For (*all*  $i \in N(v)$ )
      - i. Determine  $|N(i) \cap U(v)|$  // determine how many nodes in  $U(v)$  a neighbour  $i$  can cover
    - b. End for
    - c.
    - d. /\* select the neighbor with the highest cover count \*/
    - e. Select the neighbor  $i^*$  such that  $|N(i^*) \cap U(v)| \geq |N(i) \cap U(v)|$ ,  $\forall i \neq i^*, i, i^* \in N(v)$
    - f.
    - g.  $F(v) \leftarrow F(v) + \{i^*\}$  // add  $i^*$  to  $F(v)$
    - h.  $U(v) \leftarrow U(v) - N(i^*)$  // remove nodes covered by  $i^*$  from  $U(v)$
  4. End while
- 

Algorithm 4.1: The greedy set cover algorithm

#### 4.4.1 An Illustrative Example

Figure 4.9 shows an example network with 12 nodes labelled  $a-l$ . Table 4.3 shows the comparison between PDP and IPDP. If PDP is used, seven transmissions are required for a packet from source node  $a$  to reach all the other nodes in the network. In contrast, with IPDP, only six transmissions are required.

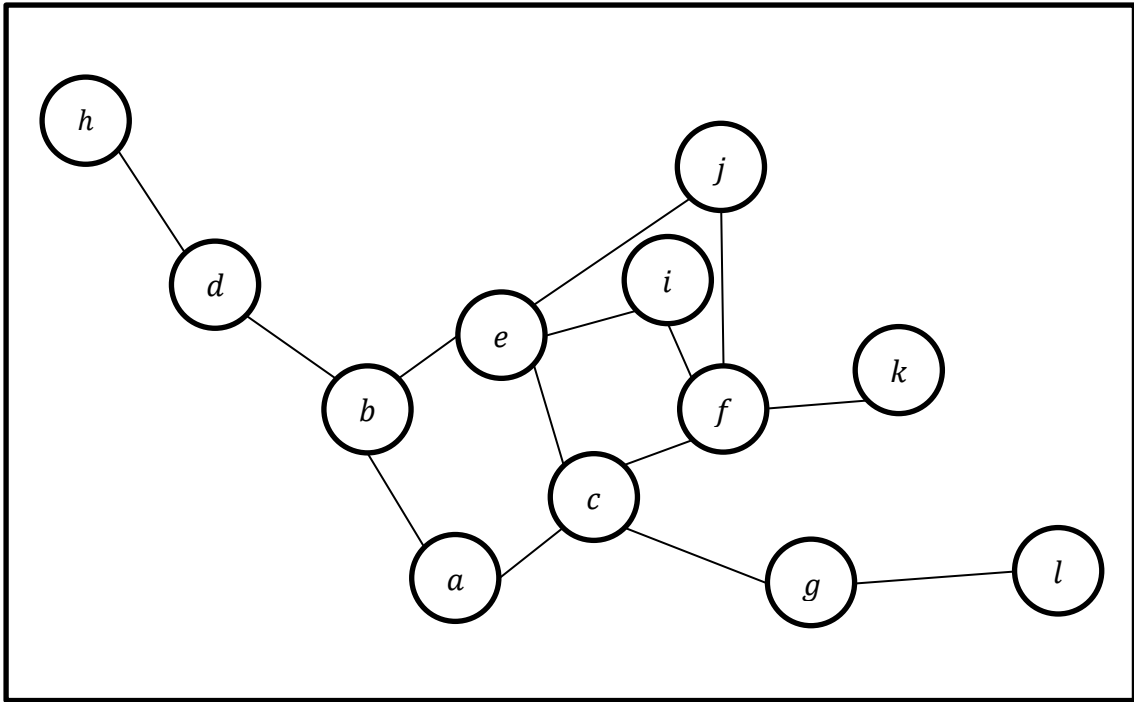


Figure 4.9: An illustrative example

Table 4.3: Comparison between PDP and IPDP

	<b>Partial Dominant Pruning (PDP)</b>	<b>Improved Partial Dominant Pruning (IPDP)</b>
$U(a)$	$\{d, e, f, g\}$	$\{d, e, f, g\}$
$F(a)$	$\{b, c\}$	$\{b, c\}$
$U(b)$	$\{h, i, j\}$	$\{h, i, j\} - \{i, j, c, b\} = \{h\}$
$F(b)$	$\{d, e\}$	$\{d\}$
$U(c)$	$\{i, j, k, l\}$	$\{i, j, k, l\}$
$F(c)$	$\{f, g\}$	$\{f, g\}$
Number of transmissions required ( $ \{a\} \cup F(a) \cup F(b) \cup F(c) $ )	$ \{a, b, c, d, e, f, g\}  = 7$	$ \{a, b, c, d, f, g\}  = 6$

**4.4.2 Relaxed Assumptions**

In Section 4.3.1, two assumptions were made, which may not hold in a real scenario. Even without these assumptions, IPDP can still work, albeit with a minor modification. To analyze this, we analyze a network using the general directed graph model instead of the undirected graph model.

The crucial part requiring modification is in how neighbor discovery is performed. In IPDP and other similar protocols, neighbor discovery are performed using HELLO

packets. However, with the HELLO packets method, the links discovered are directional. For IPDP and other similar broadcast protocols to work, the neighbor discovery method needs to be changed. One obvious method is to make a node reports only nodes that it is bi-directionally connected to as neighbors. This can be done by making a node detect its one-hop neighbors by sending a special control packet and requiring nodes who can listen to the packet to reply. The node then knows which nodes are bi-directionally connected to it through the replies that it receives and only regards such nodes as neighbors. Figure 4.10 shows an example network with irregular node transmission pattern and its corresponding connectivity graph. In this example, using the modified neighbor discovery method,  $N(a) = \{b, c\}$ ,  $N(b) = \{a, d\}$ ,  $N(c) = \{a\}$ , and  $N(d) = \{b\}$ . Suppose node  $a$  has a packet to send to all other nodes in the network, it can select node  $b$  as its forwarding node. The only restriction is that all nodes in the network should have at least one bi-directional link to at least one other node. If a node has only outgoing links but no incoming links, no other nodes can reach it. In that case, no other broadcast protocol can perform better. If a node has only incoming links but no outgoing links, other nodes cannot sense its presence. In that case, no neighbor knowledge broadcast protocols can guarantee perfect packet reachability. If a node has uni-directional links in both directions but no bi-directional links, a forwarding node has no way to know which of its one-hop neighbors can reach that node.

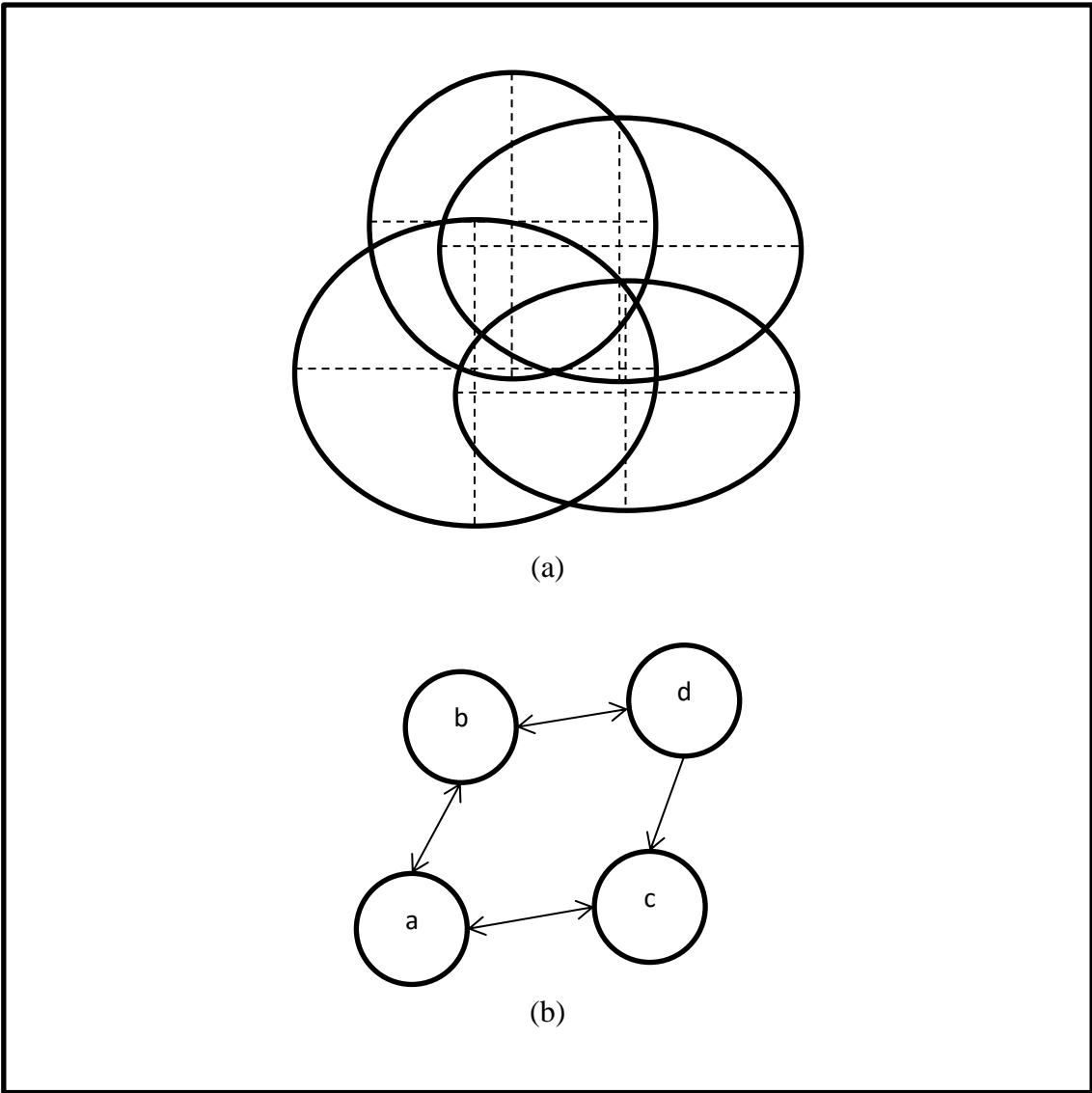


Figure 4.10: (a) an example network where nodes have irregular transmission range, (b) connectivity graph.

## 4.5 Other Enhancements for IPDP

We further propose two enhancements for IPDP to improve its effectiveness. The first enhancement is an enhanced forwarding node selection algorithm called the Improved Selection (impSel) algorithm. The greedy set cover (GSC) algorithm used in the forwarding node selection process in PDP (Lou & Wu, 2002) is inferior compared to that in MPR (Qayyum et al., 2002). In a forwarding node selection process, with the GSC algorithm, a considered forwarding node  $v$  selects the node in the set  $B(v) = N(v) - N(u)$  that can cover the highest number of nodes in  $U(v)$  at every iteration until all nodes in  $U(v)$  are covered. This could result in a sub-optimum number of forwarding nodes being selected by  $v$  due to the existence of nodes in  $U(v)$  that are reachable only through a certain node in  $B(v)$ . The nodes in  $B(v)$  that can reach these nodes (nodes in  $U(v)$  only reachable by a certain node in  $B(v)$ ) might have very low cover count of nodes in  $U(v)$  resulting in them not being selected in earlier iterations of the greedy algorithm. Nevertheless, they are in  $F(v)$  because certain nodes in  $U(v)$  are only reachable through them. If these forwarding nodes are selected first and then nodes in  $U(v)$  covered by them are removed before the greedy set cover algorithm is run, a potentially smaller set of forwarding nodes could be constructed. The improved forwarding node selection algorithm is given in Algorithm 4.2.

---

Input:  $U(v), N(v)$

Output:  $F(v)$

---

1. Initialization.  $F(v) = \emptyset$
  - 2.
  3. For (all  $i \in U(v)$ )
    - a. Determine  $N(v) \cap N(i)$
    - b. If ( $|N(v) \cap N(i)| = 1$ ) // nodes in  $U(v)$  that are only reachable by one node in  $N(v)$ 
      - i.  $F(v) \leftarrow F(v) + (N(v) \cap N(i))$
      - ii.  $U(v) \leftarrow U(v) - (N(N(v) \cap N(i)))$
    - c. End if
  4. End for
  - 5.
  6. Goto line 3 of Algorithm 4.1.
- 

Algorithm 4.2: The improved forwarding node selection algorithm

In PDP, two termination criteria were specified. One is to assign a mark/unmarked status to each node in a network. For a particular broadcast, a node is marked if it received a copy of the broadcast packet; otherwise, it is unmarked. However, how nodes are able to determine the marked/unmarked status of neighboring nodes was not detailed. The second approach is to assign a relayed/unrelayed status to each node in the network. For a particular broadcast, a node that has forwarded a packet of the broadcast labels itself as relayed. The second enhancement that we propose for IPDP is to make a forwarding node deduces and records (marks) nodes that have been covered in its broadcast ID cache. For example, when a considered forwarding node  $v$  receives a packet from its previous hop  $u$  with source  $s$ , then  $v$  knows that the one-hop neighbors of  $s$  and  $v$  were covered and therefore can be excluded from  $U(v)$ . When the relayed/unrelayed termination criterion is used, often a node is not selected as a forwarding node at a time but might be selected as a forwarding node at a later time. If the marking of covered nodes heuristic is used, then  $v$  could prune recorded covered nodes from  $U(v)$  to further reduce the size of  $U(v)$  when it selects its forwarding nodes. If the marking of covered nodes heuristic is used,  $U(v)$  is finally given as  $U(v) =$

$$N(N(v)) - N(v) - N(u) - N(N(u) \cap N(v)) -$$

$\cup_{addr(v) < addr(w), w \neq v, w \in F(u)} N(N(v) \cap N(w)) - \mathcal{C}(v)$ , where  $\mathcal{C}(v)$  is the set of nodes that are deduced as covered and recorded (marked) by  $v$ . After forwarding a packet,  $v$  also marks all nodes in its two-hop neighborhood,  $N(N(v))$ , as covered.

## 4.6 Simulation Studies

To evaluate the performance of IPDP, we performed simulation studies using an event simulator called network simulator 2 (ns-2) (“The Network Simulator - ns-2,” n.d.). The models used in ns-2 are summarized in Table 4.4.

Table 4.4: Models used in ns-2

<b>Model</b>	<b>Value</b>
Channel Type	Wireless Channel
Radio Propagation Model	Two Ray Ground
MAC Protocol	IEEE 802.11
Interface Queue Type	Drop Tail
Antenna Model	Omnidirectional Antenna
Node Mobility Model	Random Waypoint Model

The following metrics are used for comparing the various protocols:

1. *Number of transmissions*: The number of transmissions required for a unique broadcast (note: all packets of the same broadcast carry the same data and are actually different copies of the same packet).
2. *Packet reachability (%)*: The ratio of the number of nodes in the network other than the source that receive at least one copy of the packet of a broadcast to the number of nodes in the network minus one. It takes a value in the interval [0, 100]. A value of 0 means poor reachability and no node other than the source has received the packet when a broadcast has ended. A value of 100 means perfect reachability and a packet has reached all other nodes in the network.
3. *Transmissions to nodes ratio*: The ratio of the number of transmissions required in a broadcast to the number of nodes in the network.
4. *Average number of packets received per node*: The average number of times a packet of a broadcast is received by a node.



The settings used in both static and mobile scenarios are summarized in Table 4.5.

Table 4.5: Settings used in static and mobile scenarios

Parameter	Description	Value
LENGTH	The length of the simulation area.	1000 m
WIDTH	The width of the simulation area.	1000 m
HELLO_INTERVAL	The time interval between successive HELLO packets.	1 s
ALLOWED_HELLO_LOSS	The number of times the HELLO packets of a neighbor is allowed to be lost before the neighbor is considered disconnected.	2
BROADCAST_JITTER	The maximum random time a packet is delayed to avoid collisions.	20 ms

#### 4.6.1 Static Scenario

In the static scenario, 60-150 nodes were randomly placed in the simulation area, in increments of 30 nodes. A minimum number of 60 nodes were used to avoid generating a disconnected network, which could make analyzing of the results more difficult. Using a simulation framework, we were able to obtain the network connected probability with the number of nodes deployed in a network. From Figure 4.11, it can be observed that a network has a high connected probability of over 90% when 60 nodes with transmission range of 250 meters are randomly deployed in a 1000 meters by 1000 meters area. For a treatment on the network connectivity in wireless ad hoc networks, we recommend the excellent tutorial in (Bettstetter & Bettstetter, 2004). Node 0 was chosen as the source and made to broadcast a packet at exactly 7 seconds simulation time. A conservative time of  $7 - 0 = 7$  seconds was used to allow for several rounds of HELLO packet exchanges to take place for steady neighborhood information. Simulation was stopped at exactly 10 seconds simulation time for a conservative time of  $10 - 7 = 3$  seconds to allow the packet from the source to reach all other nodes in the network.

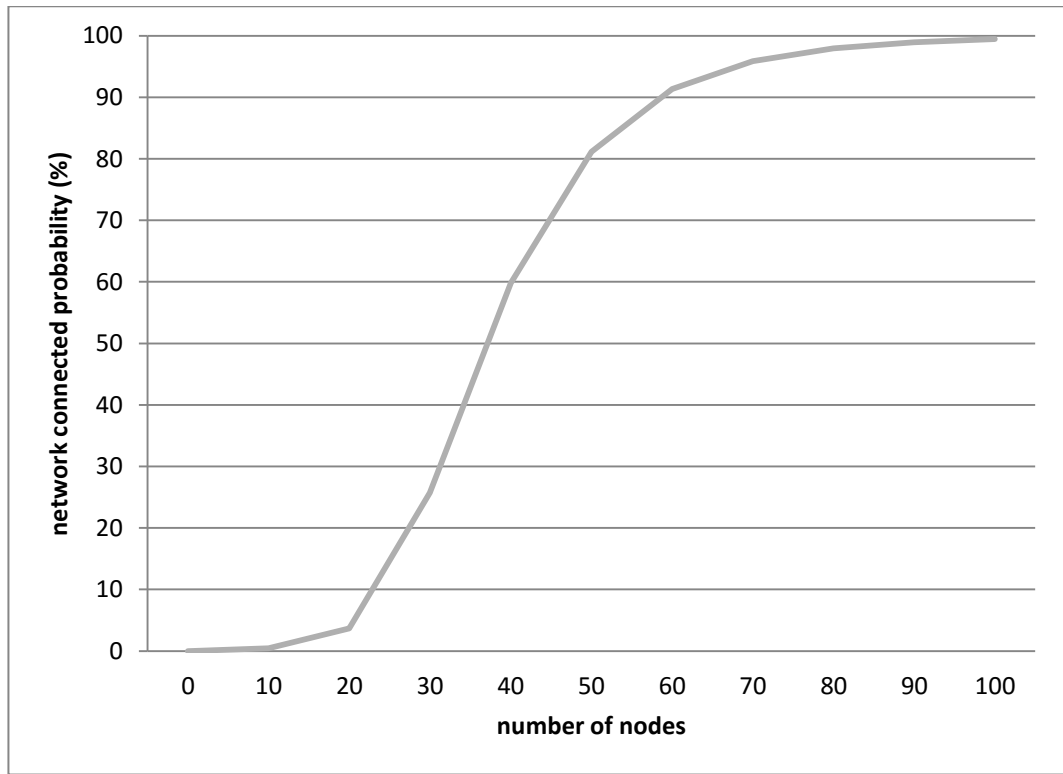


Figure 4.11: Network connected probability

For a certain number of nodes in the network, we ran the simulation 20 times, each time using a different seed number for generating the node placement. We compare different protocols using the averages of the results obtained. The settings used in this test are summarized in Table 4.6.

Table 4.6: Settings used in the static scenario

Parameter	Value
Number of nodes	60, 90, 120, 150
Number of broadcasts	1
Time of broadcast	7 s
Simulation time	10 s
Number of simulation runs for each value of the number of nodes parameter	20

Figure 4.12a shows the number of transmissions required for the single broadcast. IPDP was effective in reducing the number of transmissions required in a broadcast compared to PDP. On average, IPDP produced approximately 11.89%, 15.01%, 20.12%, and 15.63% fewer transmissions than PDP when there were 60, 90, 120, and

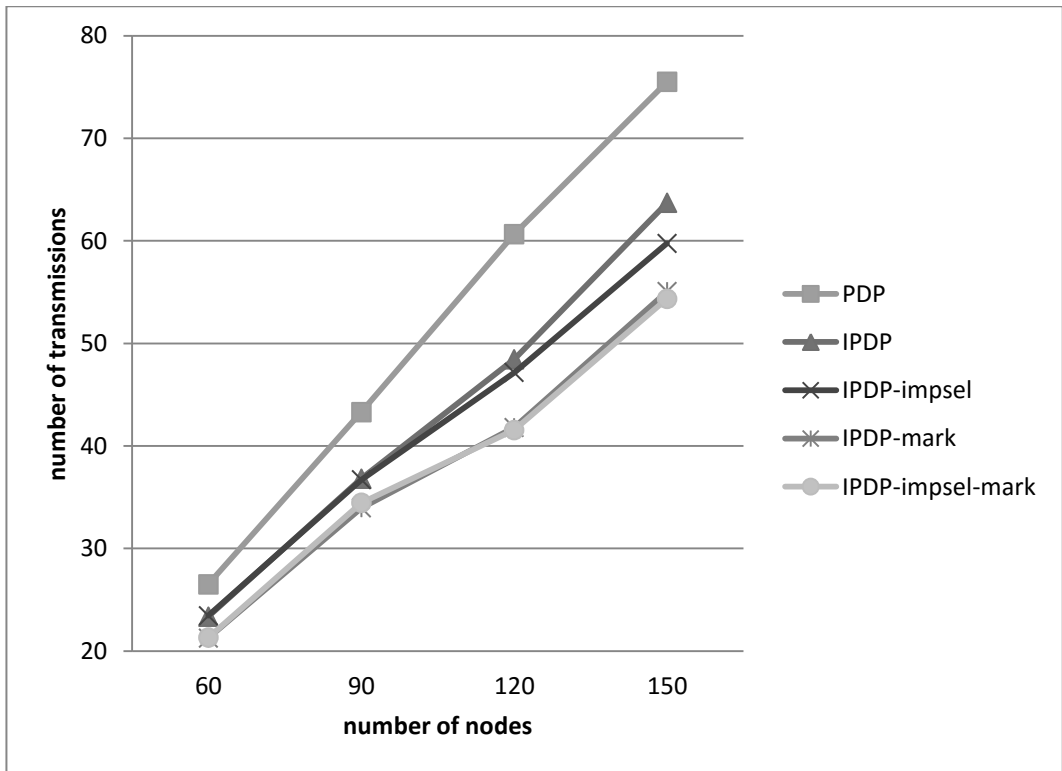
150 nodes in the network, respectively. In dense scenarios (more than or equal to 120 nodes in the network), using the improved forwarding node selection algorithm (IPDP-imp sel) further reduced the number of transmissions required for the broadcast due to more redundancy in denser networks. Redundant transmissions were further reduced when the marking of covered nodes heuristic was used (IPDP-mark, IPDP-imp sel-mark). IPDP-mark and IPDP-imp sel-mark have similar performance. It can also be observed that enabling the marking of covered nodes heuristic made the improved forwarding node selection algorithm less effective, as can be seen by comparing IPDP with IPDP-imp sel, and IPDP-mark with IPDP-imp sel-mark. The improved forwarding node selection algorithm reduces the number of forwarding nodes selected by a forwarding node in a forwarding node selection process by making a considered forwarding node  $v$  first select forwarding nodes to cover nodes in  $U(v)$  that are only reachable by certain nodes in  $N(v)$ . These forwarding nodes might have low cover count of nodes in  $U(v)$  which may make them not selected in earlier iterations of a forwarding node selection process if the improved forwarding node selection algorithm was not used. In contrast, the marking of covered nodes heuristic reduces redundant transmissions by reducing the size of  $U(v)$  and potentially allows a broadcast to terminate earlier. In the regular IPDP, there was more room for either enhancement to reduce redundant transmissions. However, when both were used simultaneously, less improvement was achieved. This shows that both the enhancements are effective in reducing redundant transmissions. Out of the two enhancements, the marking of covered nodes heuristic seems to be more effective.

From Figure 4.12b, we found that all protocols obtained full packet reachability at all node densities.

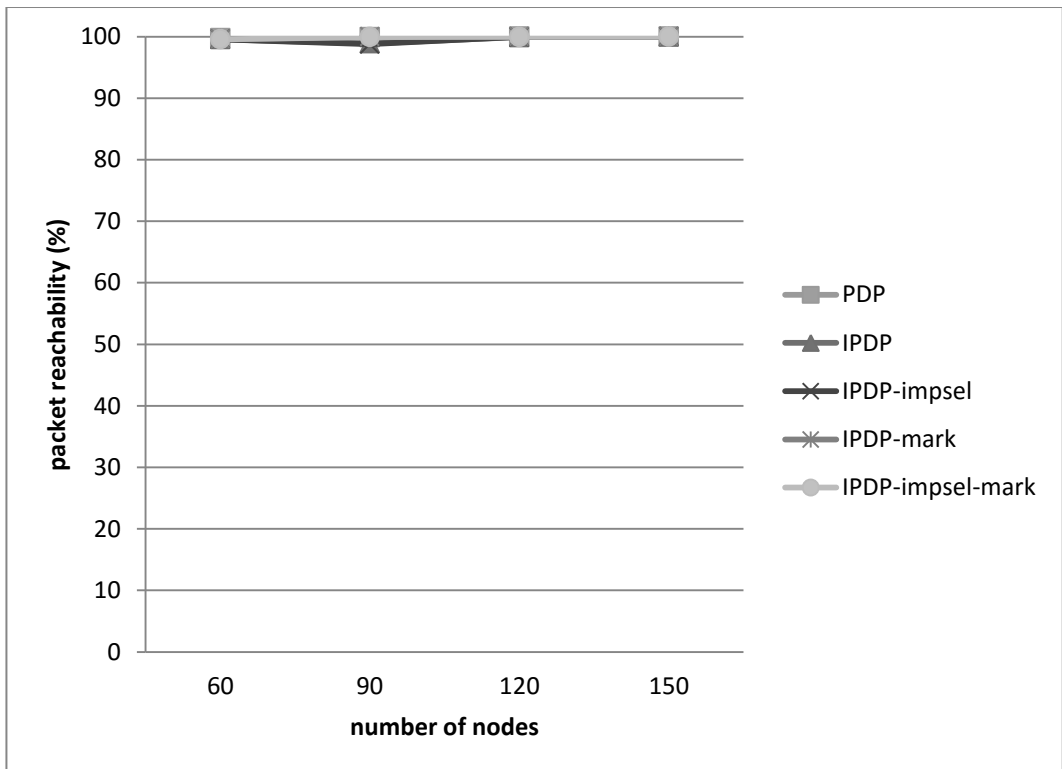
Figure 4.12c shows the transmissions to nodes ratio, which gives us an idea of how effective a broadcast protocol is relative to blind flooding. In blind flooding, a node

forwards a broadcast packet when it receives the packet for the first time; hence, if blind flooding is simulated, it would obtain the value of 1.0 in this metric. We observed that PDP was able to reduce the number of transmissions required in a broadcast from about 50% (120 and 150 nodes) to 56% (60 nodes) compared to blind flooding. IPDP improves these numbers further to from about 57.5% (150 nodes) to 61% (60 nodes). Further improvement can be observed when the improved forwarding node selection algorithm (-imp sel) or the marking of covered nodes heuristic (-mark) was used.

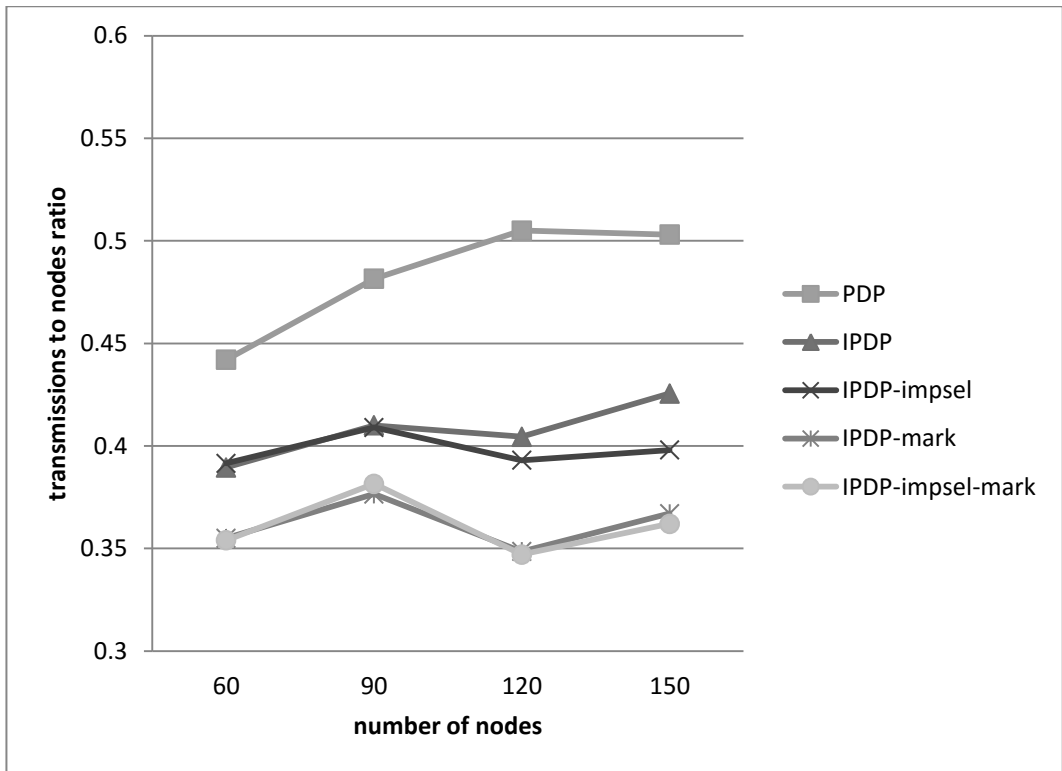
Figure 4.12d shows the average number of copies of the packet of the single broadcast is received by a node. This number increases as node density increases regardless of the broadcast protocol used, which is reasonable considering that with a higher node density, a transmission will generally cover more nodes. A node will receive more copies of a packet when more transmissions are required in a broadcast. Nodes received the highest number of packets when PDP was used. IPDP reduced the number of packets a node receives. Using the improved forwarding node selection algorithm (IPDP-imp sel) further reduced the number of packets a node receives. Using the marking of covered nodes heuristic produced the least number of redundant transmissions. This can be seen as nodes received the least number of packets when IPDP-mark and IPDP-imp sel-mark were used.



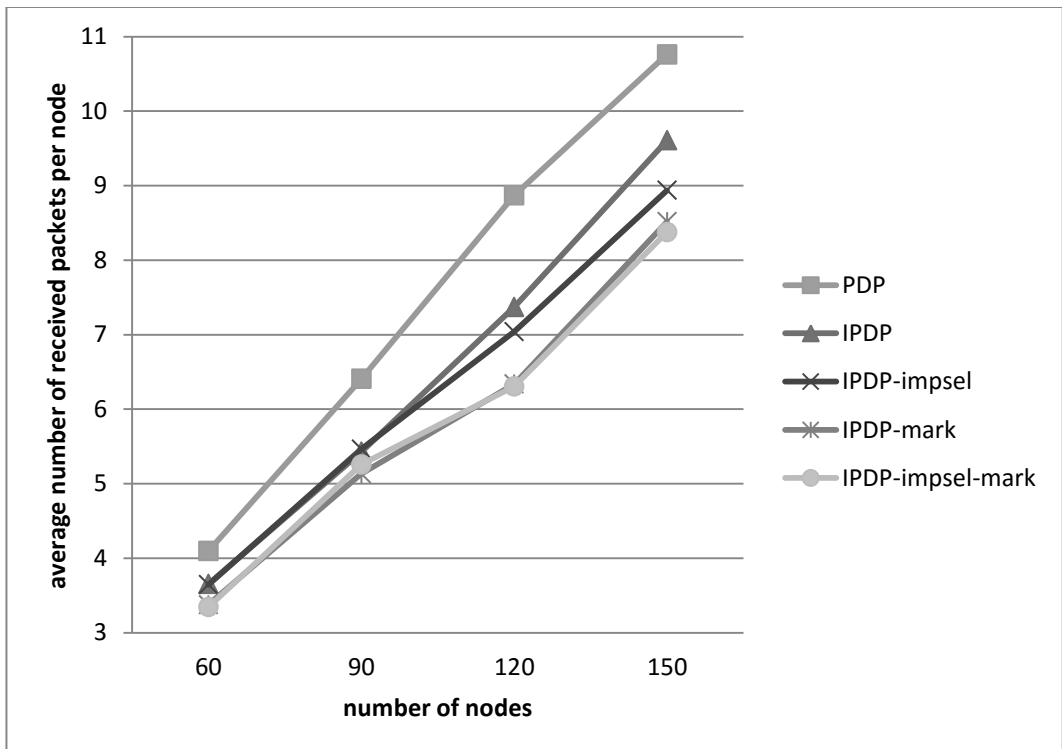
(a)



(b)



(c)



(d)

Figure 4.12: Static scenario: (a) number of transmissions, (b) packet reachability, (c) transmissions to nodes ratio, (d) average number of packets received per node

#### 4.6.2 Mobile Scenario

Similar to the static scenario, in the mobile scenario, 60-150 nodes were randomly placed in the network, in increments of 30 nodes. Node 0 was selected as the source and made to broadcast a packet periodically at 3 seconds intervals starting at exactly 0.5 seconds simulation time. We stopped the simulation at exactly 303 seconds simulation time for exactly 100 broadcasts. The maximum node speed was varied from 10 m/s to 30 m/s in increments of 10 m/s to evaluate the performance of the broadcast protocols at various levels of node mobility. As a reference, the speed of 10 m/s corresponds to the speed of 100 meter sprinters while the speed of 30 m/s or equivalently 108 km/h corresponds to vehicle highway cruising speed. For a certain maximum node speed, we ran the simulations 10 times. The settings used in this test are summarized in Table 4.7.

Table 4.7: Settings used in the mobile scenario

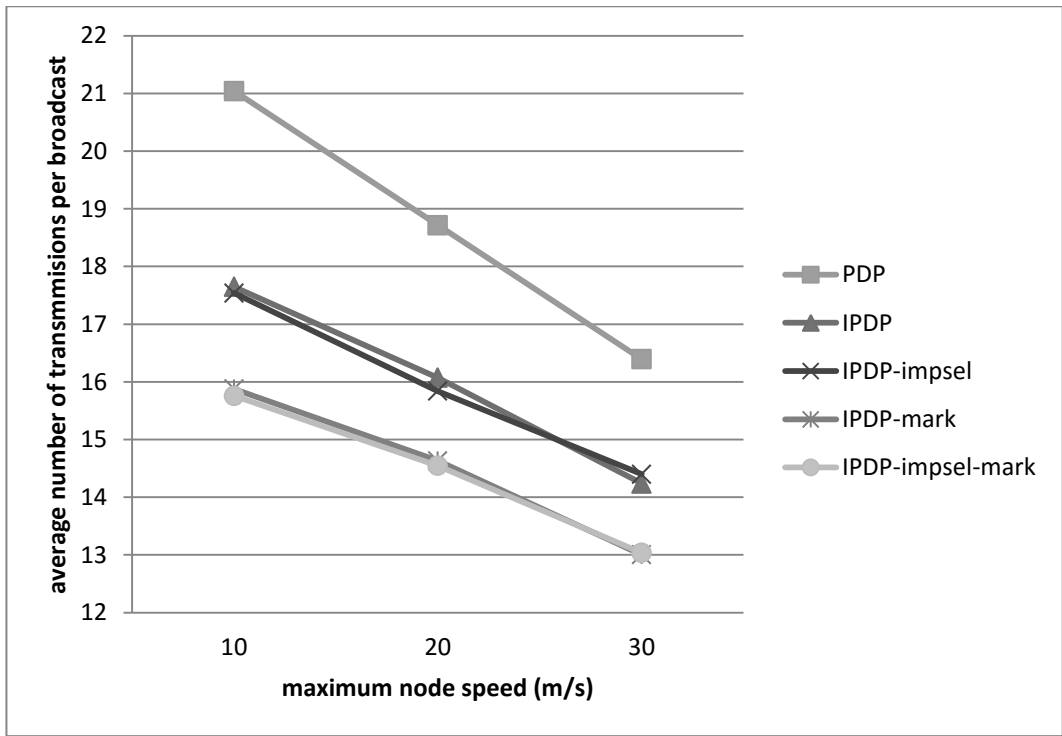
Parameter	Value
Number of broadcasts	100
Broadcast interval	3 s
Broadcast start time	0.5s
Simulation time	303 s (for 100 broadcasts)
Number of nodes	60, 90, 120, 150
Maximum node speed	10, 20, 30 m/s
Node pause time	0 s
Number of sources	1
Number of simulation runs	10

Figures 4.13, 4.14, 4.15, and 4.16 show the results obtained when there were 60, 90, 120, and 150 nodes in the network, respectively. Figures 4.13a, 4.14a, 4.15a, and 4.16a show the average number of transmissions required per broadcast over the 100 broadcasts, when there were 60, 90, 120, and 150 nodes in the network, respectively. The average number of transmissions required per broadcast reduces as the maximum node speed increases, regardless of the broadcast protocol used. We believe this could be due to the fact that the random waypoint model was used as the node mobility model. In the random waypoint model, a node moves from its initial position to a random point

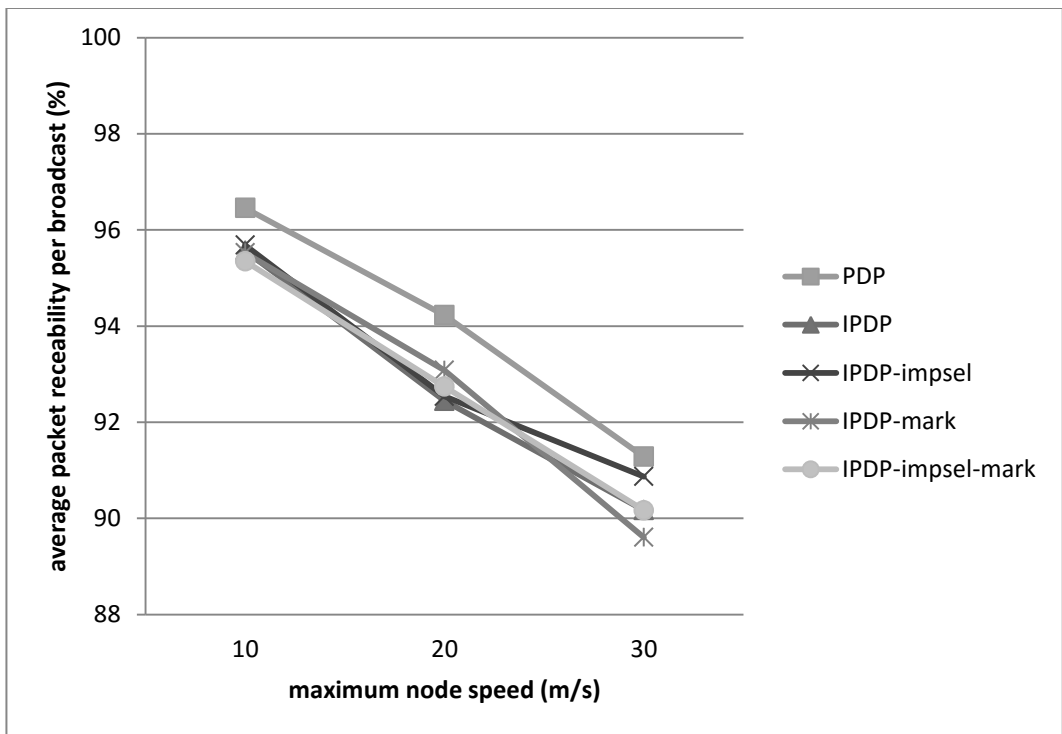
within the simulation area, called a waypoint, at a random speed. When it reaches its waypoint, it pauses for a short moment (the amount of time equal to the value of the node pause time parameter) before moving to another random waypoint at another random speed. The random waypoint model is known to generate a node spatial distribution which is independent of the initial node positions and which nodes are concentrated in the center of the deployment region (Santi, 2005). While random positions are used as waypoints, nodes spend more time in the center of the deployment region moving from one waypoint to another. This phenomenon is known as the border effect (Santi, 2005). As nodes became more concentrated in the center of the deployment region when node mobility is higher, fewer transmissions were needed for a broadcast to cover all the nodes in a network.

From Figures 4.13a, 4.14a, 4.15a, and 4.16a, we observed that IPDP and its variants produced fewer transmissions in the network compared to PDP. Figures 4.13b, 4.14b, 4.15b, and 4.16b show the average packet reachability of the 100 broadcasts when there were 60, 90, 120, and 150 nodes in the network, respectively. From these figures, it can be observed that neither PDP nor IPDP managed to obtain perfect packet reachability. We believe this to be attributed to node mobility and the use of stale neighborhood information. However, as the nodes were confined within the simulation area, every protocol obtained good packet reachability. We also observed that packet reachability decreases as node mobility increases. This is to be expected as higher node mobility means less accurate neighborhood information were used by the nodes. Due to fewer redundant transmissions, all variants of IPDP (IPDP, IPDP-impself, IPDP-mark, and IPDP-impself-mark) obtained lower packet reachability than PDP. However, from Figure 4.17, it can be observed that the difference in packet reachability between PDP and the worst performing variant of IPDP is generally low (5.35% at 150 nodes, 30 m/s maximum node speed).



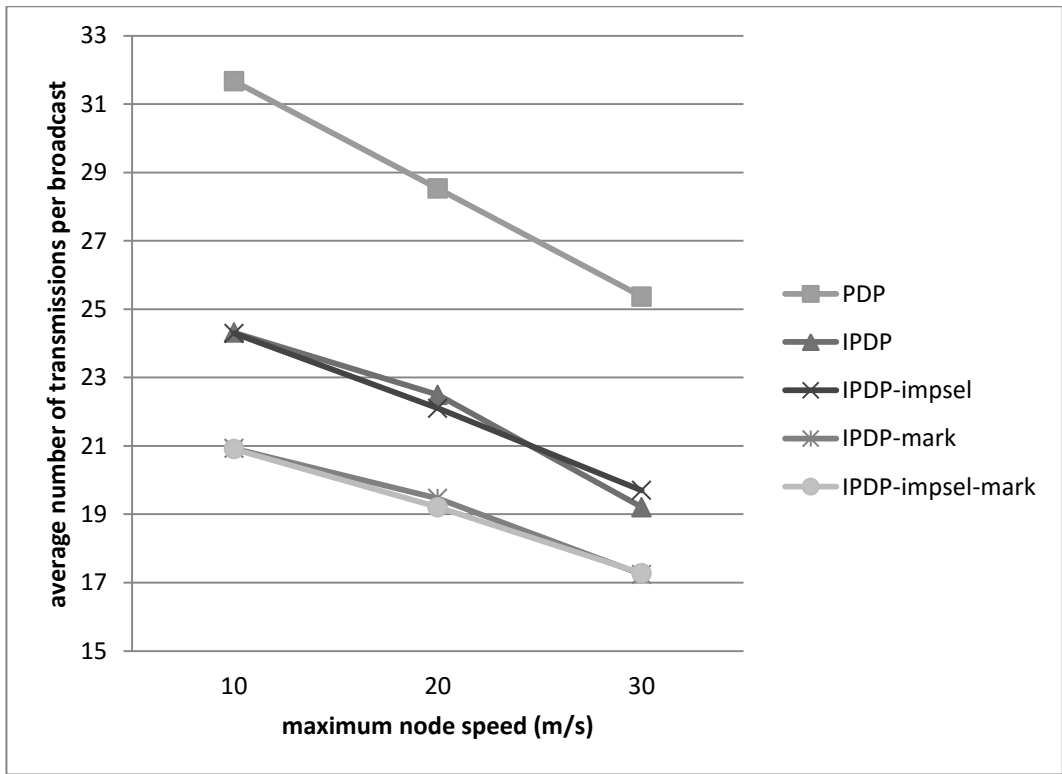


(a)

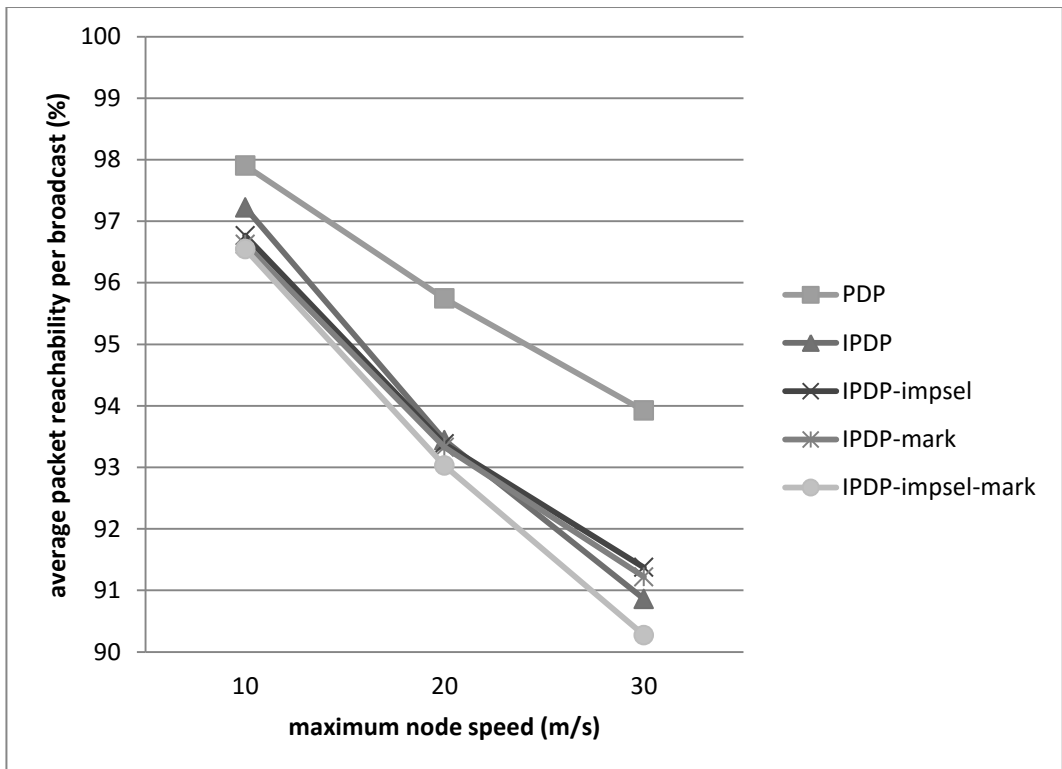


(b)

Figure 4.13: Sixty nodes: (a) average number of transmissions per broadcast, (b) average packet reachability per broadcast

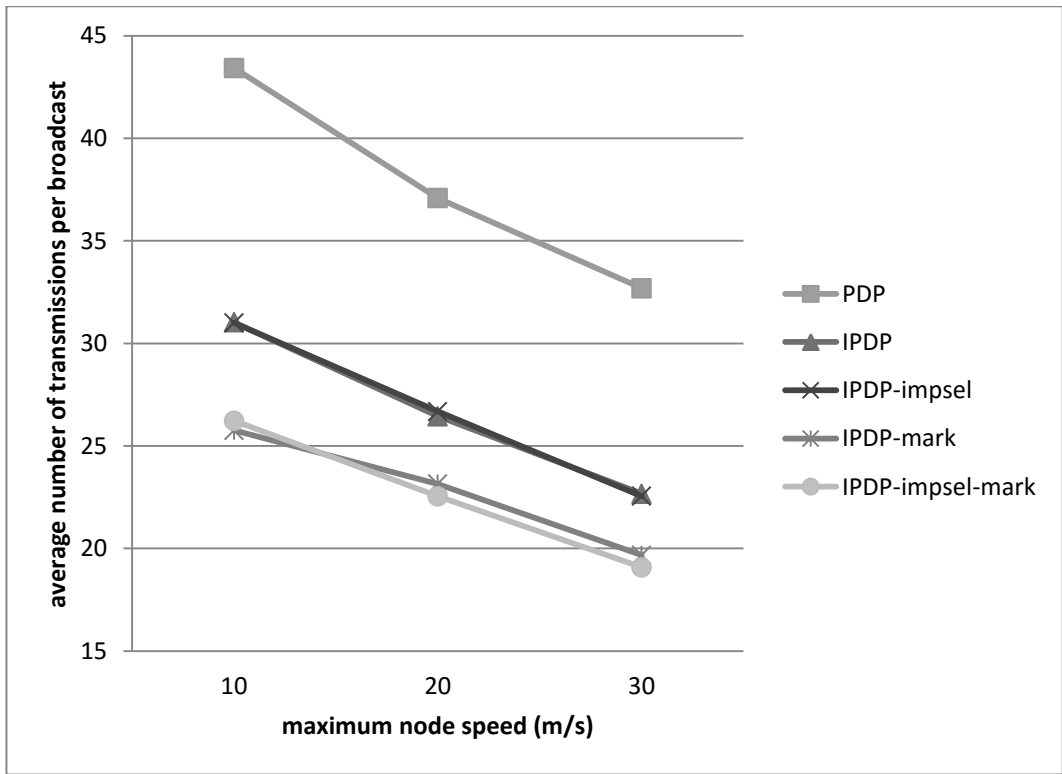


(a)

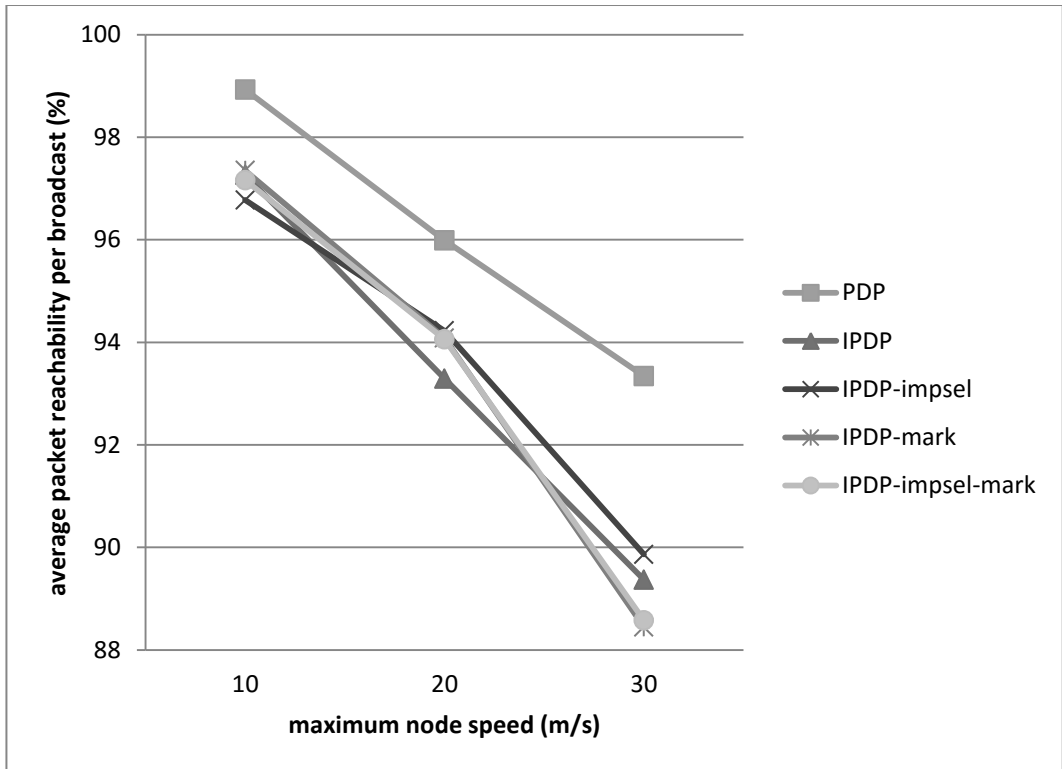


(b)

Figure 4.14: Ninety nodes: (a) average number of transmissions per broadcast, (b) average packet reachability per broadcast

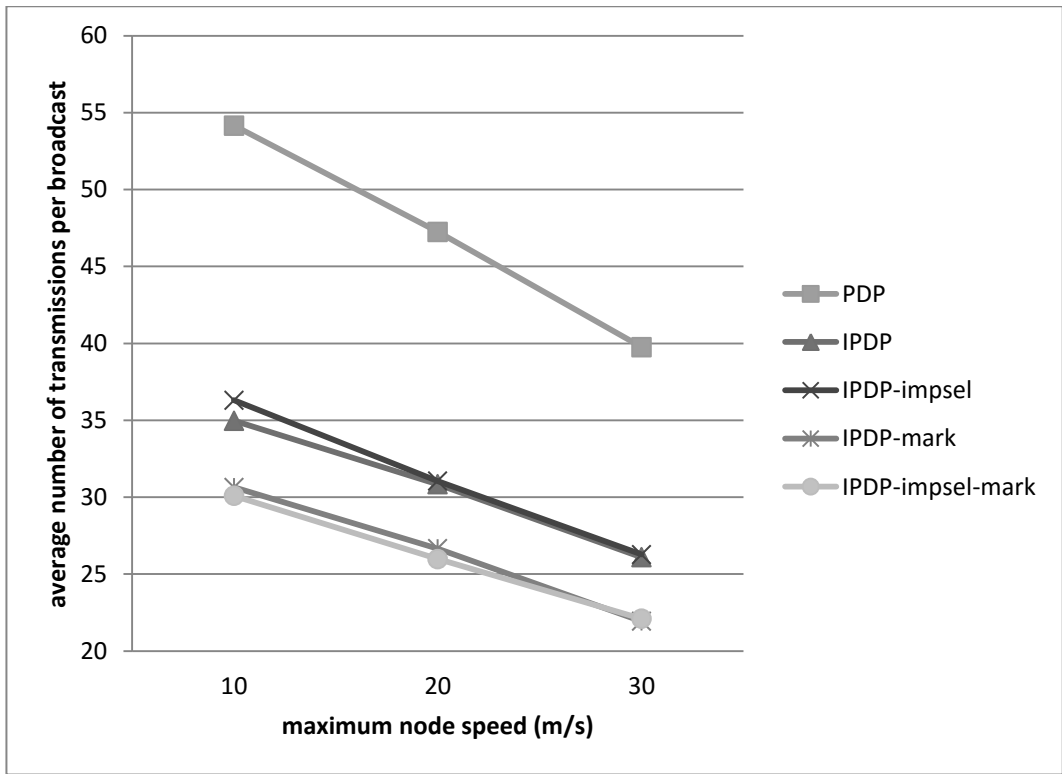


(a)

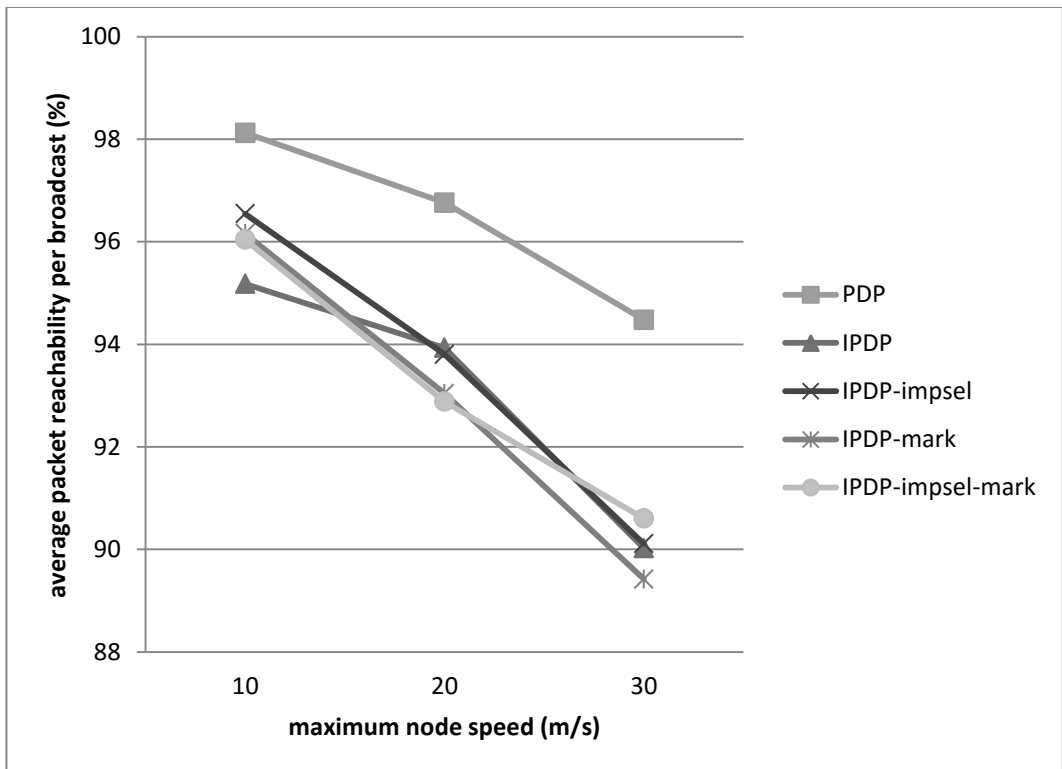


(b)

Figure 4.15: One hundred and twenty nodes: (a) average number of transmissions per broadcast, (b) average packet reachability per broadcast



(a)



(b)

Figure 4.16: One hundred and fifty nodes: (a) average number of transmissions per broadcast, (b) average packet reachability per broadcast

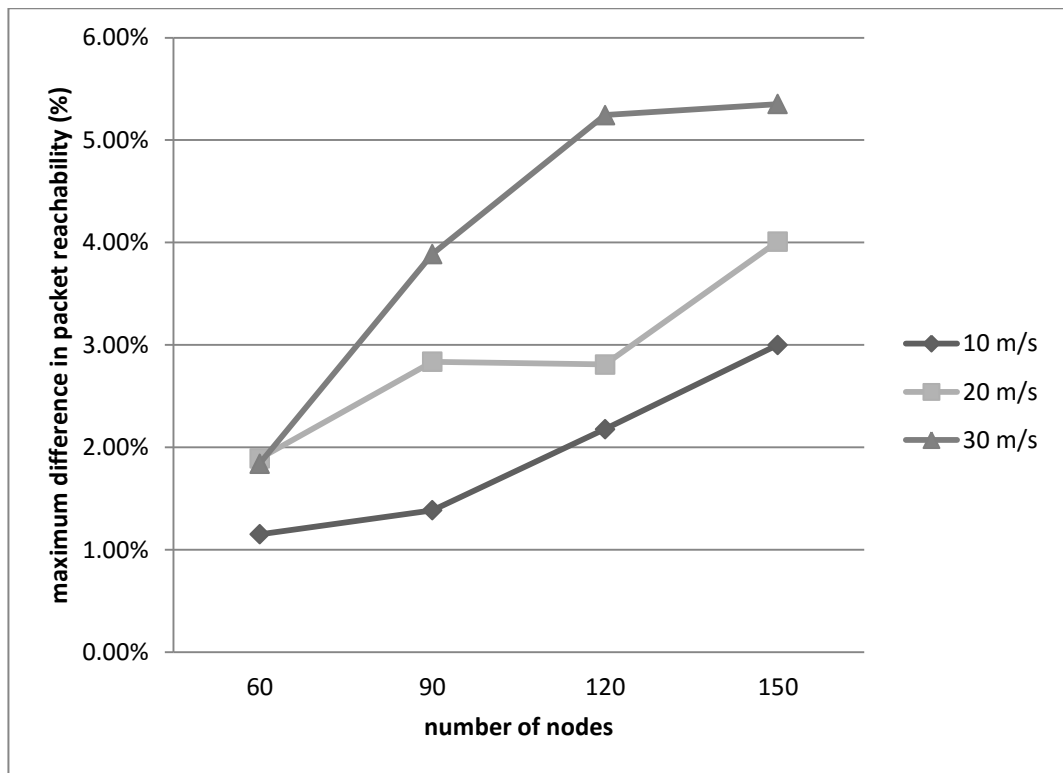


Figure 4.17: Maximum difference in packet reachability between PDP and the worst performing variant of IPDP

## 4.7 Conclusions

In this chapter, we proposed an efficient broadcast protocol for wireless ad hoc networks called Improved Partial Dominant Pruning (IPDP), which was extended from the Partial Dominant Pruning (PDP) protocol without introducing additional overhead. We make a forwarding node aware of other forwarding nodes of the same packet when it determines the set of nodes its neighbors must cover. Two enhancements were further proposed to be used together with IPDP for further gain: (1) the improved forwarding node selection algorithm (impSel), and (2) the marking of covered nodes heuristic (mark). The improved forwarding node selection algorithm allows a forwarding node to construct a potentially smaller set of forwarding nodes. The marking of covered nodes heuristic makes the set of nodes the neighbors of a forwarding node must cover smaller, and potentially allowing a broadcast to terminate earlier. In the static scenario, we found IPDP and its variants to be very effective in reducing redundant transmissions while maintaining packet reachability to all nodes in a network. In the mobile scenario, we found IPDP and its variants to perform closely to PDP in terms of packet reachability in low to high node movement speeds. Even at very high speeds, the difference in packet reachability between PDP and the worst performing variant of IPDP is low.

## CHAPTER 5: SUMMARY AND FUTURE WORK

### 5.1 Summary

In Chapter 2, we proposed the RS1 and RS2 routing metrics to aid the discovery and establishment of stable routes in mobile ad hoc networks. We implemented and evaluated the routing metrics in the popular AODV routing protocol. Due to nodes selecting routes with shorter links or links formed through lower mobility nodes, our routing metrics significantly improves the network performance (packet delivery ratio, network throughput, etc.) when compared to other routing metrics. To avoid using additional hardware (sensors) to measure the routing metric values of the paths, a link length estimation method and a node mobility estimation method were also proposed. When the routing metrics are combined with these estimation methods, even when less accurate information was used, the routing metrics still significantly outperformed other sensor-free routing metrics. As a reference, the RS1 routing metric, which is the lower performer of the two routing metrics, produced 3.1%, 3.8%, 5.8%, 3.3%, and 5.2% higher packet delivery ratio than the hop count routing metric at 5, 10, 15, 20, and 25 m/s maximum node speed, respectively. On the contrary, the Path Encounter Rate (PER) routing metric gave only negligible gains of 0.4%, 0.7%, 1.4%, -0.2%, and 1.8% over the hop count metric at 5, 10, 15, 20, and 25 m/s maximum node speed, respectively. By discovering and establishing stable routes, the RS1 routing metric managed to reduce the number of routes discoveries by approximately 25.1%, 19.9%, 24.0%, 19.2%, and 22.5% when compared to the hop count routing metric at 5, 10, 15, 20, and 25 m/s maximum node speed, respectively. On the contrary, the PER routing metric managed to reduce the number of route discoveries by only -1.3%, 0%, 4.8%, 7.5%, and 4.7% when compared to the hop count routing metric at 5, 10, 15, 20, and 25 m/s maximum node speed, respectively.

In Chapter 3, we proposed to increase the throughput of wireless ad hoc networks using the idea of network load reduction. We proposed the Network Coding Routing (NCRT) protocol, which consists of a new set of coding conditions, a new route discovery process that allows the nodes to gather the necessary information to determine if they can encode packets together based on the coding conditions, and a routing metric that allows the source nodes to select ideal paths for sending their packets on considering coding opportunities and network load. We implemented NCRT and compared it against a state-of-the-art network coding routing protocol called FORM and also DSR and found that NCRT outperforms the two, especially in terms of higher network throughput and packet delivery ratio, due to more packet encodings from the proposed set of coding conditions and better path selection from the proposed routing metric. In random networks, NCRT provided a maximum network throughput gain of about 39.9% over DSR. In contrast, FORM managed only a lower gain of about 32.8%.

In Chapter 4, we proposed a method to reduce network load (redundant transmissions) during broadcast. We improved upon an efficient broadcast protocol called Partial Dominant Pruning (PDP) by removing the overlap in the coverage areas of two forwarding nodes of the same packet. This is done without introducing new overhead. From our investigation, we found that the enhanced protocol called Improved Partial Dominant Pruning (IPDP) outperforms PDP in reducing redundant transmissions while ensuring packet reachability to all nodes in a network in the static scenario. In the static scenario, for a single broadcast, IPDP produced 11.9%, 15.0%, 20.1%, and 15.6% fewer packet transmissions than PDP when there were 60, 90, 120, and 150 nodes in the network, respectively. In the mobile scenario, due to fewer redundant transmissions, IPDP obtained slightly lower packet reachability than PDP. However, even the worst performing variant of IPDP was outperformed by PDP by only a small margin.



In this thesis, we tackled several performance issues in wireless ad hoc networks. By improving the performance of wireless ad hoc networks, we hope to prepare wireless ad hoc networks to handle the numerous challenges that it may face in the future. For instance, with a higher network throughput, wireless ad hoc networks can work more effectively in supporting the excess load from cellular networks, or aid in the realization of the Internet of Things (IoT) vision, which is expected to cause serious scalability issues as a huge amount of objects are envisioned to be able to self-organize into wireless ad hoc networks and connected to the Internet infrastructure at all times.

## 5.2 Future Work

In this thesis, we investigated the performance aspect of wireless ad hoc networks and proposed solutions for improving network performance. In Chapter 2, we proposed routing metrics for improving route stability in wireless ad hoc networks. Separately and independently in Chapter 3, we proposed to reduce the workload in a network by using the concept of network coding. These two methods could be combined for a unified method for high performance mobile ad hoc networks.

In Chapter 3, we proposed to improve network throughput using the concept of network load reduction. Besides network load reduction, there are other methods that can increase network throughput, such as using multiple network interface cards and channels simultaneously within a single network. The channel assignment problem is concerned with how the channels are assigned in a multi-channel network so that the network performance is maximized. In multi-channel networks, channel assignment and routing are heavily intertwined and must be jointly optimized for optimal network performance. This is extremely challenging to do in wireless ad hoc networks due to their distributed nature and frequent topology changes. A possible future work direction is to investigate into this issue and to design a joint channel assignment and routing protocol with network coding for ultra-high network performance.

In Chapter 4, we improved upon an existing broadcast protocol to make it more effective in reducing redundant transmissions while maintaining packet reachability to all nodes in a network. In the work, we investigated only from the perspective of a single broadcast source. Although not likely, there could be several broadcasts from different sources happening at the same time, and the number of transmissions required to complete the simultaneous broadcast operations may be reduced. We could investigate into this issue as a potential future work.

## REFERENCES

- Abbas, A. M., & Kure, O. (2010). Quality of Service in mobile ad hoc networks: a survey. *International Journal of Ad Hoc and Ubiquitous Computing*. doi:10.1504/IJAHUC.2010.034322
- Abedi, A., & Hariri, N. (2010). RSMG: Coding-aware Routing Based on Stable Matching Game. In *2010 International Conference on Wireless Networks (ICWN 2010)* (pp. 577–583). Retrieved from <http://digitalcommons.library.umaine.edu/wisenet/29/>
- Ahlswede, R., & Cai, N. (2000). Network information flow. *IEEE Transactions on Information Theory*, 46(4), 1204–1216. Retrieved from [http://ieeexplore.ieee.org/xpls/abs\\_all.jsp?arnumber=850663](http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=850663)
- Akyildiz, I. F., Su, W., Sankarasubramaniam, Y., & Cayirci, E. (2002). Wireless sensor networks: a survey. *Computer Networks*, 38(4), 393–422. doi:10.1016/S1389-1286(01)00302-4
- Akyildiz, I. F., Su, W., Sankarasubramaniam, Y., & Cayirci, E. (2002, August). A survey on sensor networks. *IEEE Communications Magazine*, (August), 102–114. Retrieved from [http://ieeexplore.ieee.org/xpls/abs\\_all.jsp?arnumber=1024422](http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1024422)
- Akyildiz, I. F., & Wang, X. (2005, September). A survey on wireless mesh networks. *IEEE Radio Communications*, (September), 23–30. Retrieved from [http://ieeexplore.ieee.org/xpls/abs\\_all.jsp?arnumber=1509968](http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1509968)
- Al-Akaidi, M., & Alchaita, M. (2007). Link stability and mobility in ad hoc wireless networks. *IET Communications*, 1(2), 173. doi:10.1049/iet-com:20060273
- Al-Kanj, L., & Dawy, Z. (2011). Offloading wireless cellular networks via energy-constrained local ad hoc networks. In *GLOBECOM - IEEE Global Telecommunications Conference*. doi:10.1109/GLOCOM.2011.6133570
- Alotaibi, E., & Mukherjee, B. (2012). A survey on routing algorithms for wireless Ad-Hoc and mesh networks. *Computer Networks*, 56(2), 940–965. doi:10.1016/j.comnet.2011.10.011
- Arango, J., Degermark, M., Efrat, A., & Pink, S. (2004). An efficient flooding algorithm for mobile ad-hoc networks. In *Proc. of WiOpt* (pp. 1–7). Retrieved from <http://scholar.google.com/scholar?hl=en&btnG=Search&q=intitle:An+Efficient+Flooding+Algorithm+for+Mobile+Ad-hoc+Networks#3>
- Aziz, A. A., Sekercioglu, Y. A., Fitzpatrick, P., & Ivanovich, M. (2012). A Survey on Distributed Topology Control Techniques for Extending the Lifetime of Battery Powered Wireless Sensor Networks. *IEEE Communications Surveys & Tutorials*, 1–24. doi:10.1109/SURV.2012.031612.00124
- Bettstetter, C., & Bettstetter, C. (2004). On the connectivity of ad hoc networks. *The Computer Journal*, 47(4), 432–447. doi:10.1093/comjnl/47.4.432

- Boukerche, A., Turgut, B., Aydın, N., Ahmad, M. Z., Bölöni, L., & Turgut, D. (2011). Routing protocols in ad hoc networks: A survey. *Computer Networks*. Elsevier B.V. doi:10.1016/j.comnet.2011.05.010
- Campista, M. E. M., Esposito, P. M., Moraes, I. M., Costa, L. H. M. K., Duate, O. C. M. B., Passos, D. G., ... Rubinstein, M. G. (2008). Routing metrics and protocols for wireless mesh networks. *IEEE Network*, 22(1), 6–12. Retrieved from [http://ieeexplore.ieee.org/xpls/abs\\_all.jsp?arnumber=4435897](http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4435897)
- Chachulski, S., Jennings, M., Katti, S., & Katabi, D. (2007). Trading structure for randomness in wireless opportunistic routing. In *SIGCOMM '07 Proceedings of the 2007 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications* (p. 169). Kyoto, Japan: ACM Press. doi:10.1145/1282380.1282400
- Chen, L., & Heinzelman, W. (2005). QoS-aware routing based on bandwidth estimation for mobile ad hoc networks. *Selected Areas in Communications*, ..., 23(3), 561–572. doi:10.1109/JSAC.2004.842560
- Chlamtac, I., Conti, M., & Liu, J. J. N. (2003). Mobile ad hoc networking: Imperatives and challenges. *Ad Hoc Networks*. doi:10.1016/S1570-8705(03)00013-1
- Chowdhury, K. R., & Akyildiz, I. F. (2011). CRP: A Routing Protocol for Cognitive Radio Ad Hoc Networks. *IEEE Journal on Selected Areas in Communications*, 29(4), 794–804. doi:10.1109/JSAC.2011.110411
- Cisco. (2015). *Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2014 – 2019*. Cisco. Retrieved from [http://www.cisco.com/en/US/solutions/collateral/ns341/ns525/ns537/ns705/ns827/white\\_paper\\_c11-520862.html](http://www.cisco.com/en/US/solutions/collateral/ns341/ns525/ns537/ns705/ns827/white_paper_c11-520862.html)
- Couto, D. S. J. De, Aguayo, D., Bicket, J., & Morris, R. (2005). a High-Throughput Path Metric for Multi-Hop Wireless Routing. *Wireless Networks*, 11(4), 419–434. doi:10.1007/s11276-005-1766-z
- De Couto, D. S. J., Aguayo, D., Bicket, J., & Morris, R. (2003). A high-throughput path metric for multi-hop wireless routing. In *Proceedings of the 9th annual international conference on Mobile computing and networking - MobiCom '03* (p. 134). New York, New York, USA: ACM Press. doi:10.1145/938998.939000
- Djenouri, D., Khelladi, L., & Badache, A. N. (2005). A survey of security issues in mobile ad hoc and sensor networks. *IEEE Communications Surveys & Tutorials*, 7(4), 2–28. doi:10.1109/COMST.2005.1593277
- Do, N. M., Hsu, C., Singh, J. P., & Venkatasubramanian, N. (2011). Massive live video distribution using hybrid cellular and ad hoc networks. In *2011 IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks* (pp. 1–9). doi:10.1109/WoWMoM.2011.5986380
- Doppler, K., Rinne, M., Wijting, C., Ribeiro, C. B., & Hug, K. (2009). Device-to-device communication as an underlay to LTE-advanced networks. *IEEE Communications Magazine*, 47(12), 42–49. doi:10.1109/MCOM.2009.5350367

- Draves, R., Padhye, J., & Zill, B. (2004). Routing in multi-radio, multi-hop wireless mesh networks. *Proceedings of the 10th Annual International Conference on Mobile Computing and Networking - MobiCom '04*, 114. doi:10.1145/1023720.1023732
- Dutkiewicz, E. (2006). A New Method of Selecting Stable Paths in Mobile Ad Hoc Networks. In *IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing - Vol 2 - Workshops* (Vol. 2, pp. 38–45). IEEE. doi:10.1109/SUTC.2006.11
- Even, S. (1979). *Graph Algorithms*. Computer Science Press. Retrieved from [http://books.google.com.my/books/about/Graph\\_Algorithms.html?id=adtQAAAAMAAJ&pgis=1](http://books.google.com.my/books/about/Graph_Algorithms.html?id=adtQAAAAMAAJ&pgis=1)
- Fitzek, F. H., Popovski, P., & Zorzi, M. (2005). A symbolic perspective on low-cost cellular and multihop WLAN interworking solutions. *IEEE Wireless Communications*, 12(6), 4–10. doi:10.1109/MWC.2005.1561939
- Gerla, M. (1999). IPv6 flow handoff in ad hoc wireless networks using mobility prediction. In *Seamless Interconnection for Universal Services. Global Telecommunications Conference. GLOBECOM'99. (Cat. No.99CH37042)* (Vol. 1a, pp. 271–275). IEEE. doi:10.1109/GLOCOM.1999.831647
- Guha, S., & Khuller, S. (1998). Approximation Algorithms for Connected Dominating Sets. *Algorithmica*, 20(4), 374–387. doi:10.1007/PL00009201
- Guo, B., Li, H., Zhou, C., & Cheng, Y. (2011). Analysis of general network coding conditions and design of a free-ride-oriented routing metric. *IEEE Transactions on Vehicular Technology*, 60(4), 1714–1727. doi:10.1109/TVT.2011.2121097
- Gupta, P., & Kumar, P. R. (2000). The capacity of wireless networks. *IEEE Transactions on Information Theory*, 46(2), 388–404. doi:10.1109/18.825799
- Han, S., Zhong, Z., Li, H., Chen, G., Chan, E., & Mok, A. K. (2008). Coding-Aware Multi-path Routing in Multi-Hop Wireless Networks. In *2008 IEEE International Performance, Computing and Communications Conference* (pp. 93–100). Ieee. doi:10.1109/PCCC.2008.4745146
- Hur, I., Le, T., Jo, M., & Choo, H. (2012). Regular Tiling Based Flooding Scheme for Energy Efficiency in Wireless Ad Hoc Networks. *Journal of Internet Technology*, 13(2), 257–272. Retrieved from <http://embedded.korea.ac.kr/esel/paper/international/Minho Cho/18.pdf>
- Ibm, C., & Perkins, W. E. (1994). Highly Dynamic ( DSDV ) for Mobile Computers Routing. In *SIGCOMM Comput Commun Rev* (pp. 234–244).
- Iqbal, M. A., Dai, B., Huang, B., Hassan, A., & Yu, S. (2011). Survey of network coding-aware routing protocols in wireless networks. *Journal of Network and Computer Applications*, 34(6), 1956–1970. doi:10.1016/j.jnca.2011.07.012
- Jacquet, P., Muhlethaler, P., Clausen, T., Laouiti, A., Qayyum, A., & Viennot, L. (2001). Optimized link state routing protocol for ad hoc networks. In *Proceedings*.

- IEEE International Multi Topic Conference, 2001. IEEE INMIC 2001. Technology for the 21st Century.* (pp. 62–68). Ieee. doi:10.1109/INMIC.2001.995315
- Jain, K., Padhye, J., Padmanabhan, V. N., & Qiu, L. (2005). Impact of Interference on Multi-Hop Wireless Network Performance. *Wireless Networks*, *11*(4), 471–487. doi:10.1007/s11276-005-1769-9
- Jianzhen Sun, Yuan'an Liu, Hefei Hu, & Dongming Yuan. (2010). Link stability based routing in mobile ad hoc networks. In *2010 5th IEEE Conference on Industrial Electronics and Applications* (pp. 1821–1825). IEEE. doi:10.1109/ICIEA.2010.5515377
- Johnson, D. B., Maltz, D. A., & Broch, J. (2001). DSR: The dynamic source routing protocol for multi-hop wireless ad hoc networks. In *Monarch Papers* (pp. 1–25). Retrieved from <http://cseweb.ucsd.edu/~snoeren/cse291wi07/presentations/verkaik.pdf>
- Kadi, N., & Agha, K. Al. (2008). Optimized MPR-based flooding in wireless ad hoc network using network coding. In *2008 1st IFIP Wireless Days* (pp. 1–5). Ieee. doi:10.1109/WD.2008.4812903
- Kajioka, S., Wakamiya, N., Satoh, H., Monden, K., Hayashi, M., Matsui, S., & Murata, M. (2011). A QoS-aware routing mechanism for multi-channel multi-interface ad-hoc networks. *Ad Hoc Networks*, *9*(5), 911–927. doi:10.1016/j.adhoc.2010.10.008
- Kannhavong, B., Nakayama, H., Nemoto, Y., Kato, N., & Jamalipour, A. (2007). A survey of routing attacks in mobile ad hoc networks. *IEEE Wireless Communications*, *14*(5), 85–91. doi:10.1109/MWC.2007.4396947
- Karp, B., & Kung, H. T. (2000). GPSR. In *Proceedings of the 6th annual international conference on Mobile computing and networking - MobiCom '00* (pp. 243–254). New York, New York, USA: ACM Press. doi:10.1145/345910.345953
- Katti, S., Rahul, H., Hu, W., Katabi, D., Medard, M., & Crowcroft, J. (2008). XORs in the air: Practical wireless network coding. *IEEE/ACM Transactions on Networking*, *16*(3), 497–510. doi:10.1109/TNET.2008.923722
- Kim, W. (2001). A reliable route selection algorithm using Global Positioning Systems in mobile ad-hoc networks. In *ICC 2001. IEEE International Conference on Communications. Conference Record (Cat. No.01CH37240)* (Vol. 10, pp. 3191–3195). IEEE. doi:10.1109/ICC.2001.937260
- Ko, Y., & Vaidya, N. (2000). Location-Aided Routing (LAR) in mobile ad hoc networks. *Wireless Networks*, *6*(4), 307–321. doi:10.1023/A:1019106118419
- Kwon, T. J., & Gerla, M. (2002). Efficient flooding with passive clustering (PC) in ad hoc networks. In *ACM SIGCOMM Computer Communication Review* (pp. 44–56). Retrieved from <http://dl.acm.org/citation.cfm?id=510730>
- Lal, C., Laxmi, V., & Gaur, M. S. (2011). Performance analysis of MANET routing protocols for multimedia traffic. In *2011 2nd International Conference on*

*Computer and Communication Technology (ICCCT-2011)* (pp. 595–600). IEEE.  
doi:10.1109/ICCCT.2011.6075103

- Le, J., Lui, J. C. S., & Chiu, D. M. (2010). DCAR: Distributed coding-aware routing in wireless networks. *IEEE Transactions on Mobile Computing*, 9(4), 596–608.  
doi:10.1109/TMC.2009.160
- Lim, H., & Kim, C. (2000). Multicast tree construction and flooding in wireless ad hoc networks. In *Proceedings of the 3rd ACM international workshop on Modeling, analysis and simulation of wireless and mobile systems - MSWIM '00* (pp. 61–68).  
doi:10.1145/346855.346865
- Lim, H., & Kim, C. (2001). Flooding in wireless ad hoc networks. *Computer Communications*, 24(3-4), 353–363. doi:10.1016/S0140-3664(00)00233-4
- Liu, H., Wan, P., Jia, X., Liu, X., & Yao, F. (2006). Efficient Flooding Scheme Based on 1-Hop Information in Mobile Ad Hoc Networks. In *Proceedings IEEE INFOCOM 2006. 25TH IEEE International Conference on Computer Communications* (Vol. 00, pp. 1–12). IEEE. doi:10.1109/INFOCOM.2006.17
- Lou, W., & Wu, J. (2002). On reducing broadcast redundancy in ad hoc wireless networks. *IEEE Transactions on Mobile Computing*, 1(2), 111–122.  
doi:10.1109/TMC.2002.1038347
- Lovász, L. (1975). On the ratio of optimal integral and fractional covers. *Discrete Mathematics*, 13(4), 383–390. Retrieved from  
<http://www.sciencedirect.com/science/article/pii/0012365X75900588>
- Lu, T., & Feng, K. (2005). Predictive mobility and location-aware routing protocol in mobile ad hoc networks. In *GLOBECOM '05. IEEE Global Telecommunications Conference, 2005.* (p. 5 pp.). IEEE. doi:10.1109/GLOCOM.2005.1577768
- Macone, D., Oddi, G., & Pietrabissa, A. (2013). MQ-Routing: Mobility-, GPS- and energy-aware routing protocol in MANETs for disaster relief scenarios. *Ad Hoc Networks*, 11(3), 861–878. doi:10.1016/j.adhoc.2012.09.008
- Marina, M. K., & Das, S. R. (2006). Ad hoc on-demand multipath distance vector routing. *Wireless Communications and Mobile Computing*, 6(7), 969–988.  
doi:10.1002/wcm.432
- Miorandi, D., Sicari, S., De Pellegrini, F., & Chlamtac, I. (2012). Internet of things: Vision, applications and research challenges. *Ad Hoc Networks*, 10(7), 1497–1516.  
doi:10.1016/j.adhoc.2012.02.016
- Moussaoui, A., Semchedine, F., & Boukerram, A. (2014). A link-state QoS routing protocol based on link stability for Mobile Ad hoc Networks. *Journal of Network and Computer Applications*, 39, 117–125. doi:10.1016/j.jnca.2013.05.014
- Murugesan, M., & Krishnan, A. (2010). Reliable and Efficient Algorithm for Broadcasting in Asymmetric Mobile Ad hoc Networks. *International Journal of Computer Applications*. doi:10.5120/822-1164

- Ng, P. C., & Liew, S. C. (2007). Throughput analysis of IEEE802.11 multi-hop ad hoc networks. *IEEE/ACM Transactions on Networking*, 15(2), 309–322. doi:10.1109/TNET.2007.892848
- Ni, B., Santhapuri, N., Zhong, Z., & Nelakuditi, S. (2006). Routing with opportunistically coded exchanges in wireless mesh networks. In *2006 2nd IEEE Workshop on Wireless Mesh Networks* (pp. 157–159). Ieee. doi:10.1109/WIMESH.2006.288636
- Paruchuri, V., Duresi, A., & Jain, R. (2003). Optimized flooding protocol for ad hoc networks. In *arXiv preprint cs/0311013* (pp. 1–10). Retrieved from <http://arxiv.org/abs/cs/0311013>
- Peng, H., & Shao, L. (2010). Energy Saving Routing Algorithm Based on Stable Zone in Mobile Ad Hoc Networks. In *2010 Third International Symposium on Information Processing* (pp. 181–185). IEEE. doi:10.1109/ISIP.2010.36
- Peng, W., & Lu, X. (2001). AHBP: An efficient broadcast protocol for mobile Ad hoc networks. *Journal of Computer Science and Technology*. doi:10.1007/BF02950416
- Peng, Y., Yu, Y., Wang, X., Song, Q., Yu, Y., & Yu, C. (2013). A new coding- and interference-aware routing protocol in wireless mesh networks. *Computers & Electrical Engineering*, 39(6), 1822–1836. doi:10.1016/j.compeleceng.2013.02.007
- Penz, S. (2007). A DSR Extension for Connection Stability Assessment in Mobile Ad-Hoc Networks. In *Fifth Annual IEEE International Conference on Pervasive Computing and Communications Workshops (PerComW'07)* (pp. 509–513). IEEE. doi:10.1109/PERCOMW.2007.6
- Perkins, C. E., & Royer, E. M. (1999). Ad-hoc on-demand distance vector routing. In *Proceedings - WMCSA '99: 2nd IEEE Workshop on Mobile Computing Systems and Applications* (pp. 90–100). doi:10.1109/MCSA.1999.749281
- Qayyum, A., Viennot, L., & Laouiti, A. (2002). Multipoint relaying for flooding broadcast messages in mobile wireless networks. In *Proceedings of the 35th Annual Hawaii International Conference on System Sciences* (pp. 3866–3875). IEEE Comput. Soc. doi:10.1109/HICSS.2002.994521
- Rahman, A., Endadul Hoque, M., Rahman, F., Kundu, S. K., & Gburzynski, P. (2009). Enhanced partial dominant pruning (EPDP) based broadcasting in ad hoc wireless networks. *Journal of Networks*, 4(9), 895–904. doi:10.4304/jnw.4.9.895-904
- Santi, P. (2005, June 1). Topology control in wireless ad hoc and sensor networks. *ACM Computing Surveys*. doi:10.1145/1089733.1089736
- Sasson, Y., Cavin, D., & Schiper, A. (2003). Probabilistic broadcast for flooding in wireless mobile ad hoc networks. In *2003 IEEE Wireless Communications and Networking, 2003. WCNC 2003*. (Vol. 2, pp. 1124–1130). IEEE. doi:10.1109/WCNC.2003.1200529



- Sengupta, S., Rayanchu, S., & Banerjee, S. (2010). Network coding-aware routing in wireless networks. *IEEE/ACM Transactions on Networking*, 18(4), 1158–1170. Retrieved from [http://ieeexplore.ieee.org/xpls/abs\\_all.jsp?arnumber=5438834](http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5438834)
- Sesay, S., Yang, Z., & He, J. (2004). A survey on mobile ad hoc wireless network. *Information Technology Journal*, 3(2), 168–175. Retrieved from <http://www.docsdrive.com/pdfs/ansinet/itj/2004/168-175.pdf>
- Sharef, B. T., Alsaqour, R. a., & Ismail, M. (2014). Vehicular communication ad hoc routing protocols: A survey. *Journal of Network and Computer Applications*. doi:10.1016/j.jnca.2013.09.008
- Sheng, M., Li, J., & Shi, Y. (2005). Relative degree adaptive flooding broadcast algorithm for Ad hoc networks. *IEEE Transactions on Broadcasting*, 51(2), 216–222. doi:10.1109/TBC.2005.847624
- Son, T. T., Minh, H. Le, Sexton, G., & Aslam, N. (2014). A novel encounter-based metric for mobile ad-hoc networks routing. *Ad Hoc Networks*, 14(2), 2–14. doi:10.1016/j.adhoc.2013.10.012
- Sun, J., Liu, Y., Hu, H., & Yuan, D. (2010). On-demand coding-aware routing in wireless Mesh networks. *The Journal of China Universities of Posts and Telecommunications*, 17(5), 80–92. doi:10.1016/S1005-8885(09)60511-2
- Tan, W. C.-W., Bose, S. K., & Cheng, T.-H. (2011). Location-aided power and mobility aware routing in wireless ad hoc network. In *2011 IFIP Wireless Days (WD)* (pp. 1–3). IEEE. doi:10.1109/WD.2011.6098148
- Tan, W. C.-W., Bose, S. K., & Cheng, T.-H. (2012). Power and mobility aware routing in wireless ad hoc networks. *IET Communications*, 6(11), 1425. doi:10.1049/iet-com.2011.0451
- The Network Simulator - ns-2*. (n.d.). Retrieved November 14, 2012, from <http://www.isi.edu/nsnam/ns/>
- Toh, C.-K. (1997). Associativity-Based Routing for Ad Hoc Mobile Networks. *Wireless Personal Communications*, 4(2), 103–139. doi:10.1023/A:1008812928561
- Utsu, K., & Ishii, H. (2010). Load-aware Flooding over Ad Hoc Networks enabling High Message Reachability and Traffic Reduction. In *The Fifth International Conference on Mobile Computing and Ubiquitous Networking (ICMU 2010)* (pp. 3–4). Retrieved from <http://www.icmu.org/icmu2010/Papers/6-4.pdf>
- Vassis, D., Kormentzas, G., Rouskas, A., & Maglogiannis, I. (2005). The IEEE 802.11g standard for high data rate WLANs. *IEEE Network*, 19(3), 21–26. doi:10.1109/MNET.2005.1453395
- Ververidis, C. N., & Polyzos, G. C. (2008). Service discovery for mobile ad hoc networks: A survey of issues and techniques. *IEEE Communications Surveys and Tutorials*, 10(3), 30–45. doi:10.1109/COMST.2008.4625803

- Villaseñor-González, L. (2007). A Performance Study of the IEEE 802.11g PHY and MAC Layers over Heterogeneous and Homogeneous WLANs. *Ingeniería, Investigación Y Tecnología*, 8(1), 45–57. Retrieved from <http://www.scielo.org.mx/scielo.php?pid=S1405-77432007000100005>
- Wang, X., & Li, J. (2012). Network Coding Aware Cooperative MAC Protocol for Wireless Ad Hoc Networks. *IEEE Transactions on Parallel and Distributed Systems*, XX(XX), 1–13.
- Wang, Y., Zhou, Y., Yu, Y., Wang, Z., & Du, S. (2012). AD-AODV: A Improved Routing Protocol Based on Network Mobility and Route Hops. In *2012 8th International Conference on Wireless Communications, Networking and Mobile Computing* (pp. 1–4). IEEE. doi:10.1109/WiCOM.2012.6478385
- Williams, B., & Camp, T. (2002). Comparison of broadcasting techniques for mobile ad hoc networks. In *Proceedings of MobiHoc '02* (pp. 194–205). New York, New York, USA: ACM Press. doi:10.1145/513824.513825
- Wu, C., Kumekawa, K., & Kato, T. (2009). A MANET protocol considering link stability and bandwidth efficiency. In *2009 International Conference on Ultra Modern Telecommunications & Workshops* (pp. 1–8). IEEE. doi:10.1109/ICUMT.2009.5345608
- Xu, S., & Saadawi, T. (2001). Does the IEEE 802.11 MAC protocol work well in multihop wireless ad hoc networks? *IEEE Communications Magazine*, 39(6), 130–137. doi:10.1109/35.925681
- Yan, Y., Zhang, B., Mouftah, H. T., & Ma, J. (2008). Practical Coding-Aware Mechanism for Opportunistic Routing in Wireless Mesh Networks. In *2008 IEEE International Conference on Communications* (pp. 2871–2876). Ieee. doi:10.1109/ICC.2008.541
- Yan, Y., Zhang, B., Zheng, J., & Ma, J. (2010). Core: a coding-aware opportunistic routing mechanism for wireless mesh networks. *IEEE Wireless Communications*, 17(3), 96–103. Retrieved from [http://ieeexplore.ieee.org/xpls/abs\\_all.jsp?arnumber=5490984](http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5490984)
- Yang, H., Luo, H., Ye, F., Lu, S., & Zhang, L. (2004). Security in Mobile Ad Hoc Networks: Challenges and Solutions. *IEEE Wireless Communications*. doi:10.1109/MWC.2004.1269716
- Yang, Y., Wang, J., & Kravets, R. (2005). Designing routing metrics for mesh networks. In *IEEE Workshop on Wireless mesh Networks (WiMesh)*. Retrieved from <http://mars.cs.kent.edu/~peyravi/Ref/WiMesh05/p6.pdf>
- Yang, Y., Zhong, C., Sun, Y., & Yang, J. (2010). Network coding based reliable disjoint and braided multipath routing for sensor networks. *Journal of Network and Computer Applications*, 33(4), 422–432. doi:10.1016/j.jnca.2010.02.003
- Yi, Y., Gerla, M., & Kwon, T.-J. (2003). Efficient Flooding in Ad Hoc Networks using On-demand (passive) Cluster Formation. In *Second Annual Mediterranean Ad Hoc*

*Networking Workshop (Med-hoc-Net 2003)* (pp. 1–10). Retrieved from [http://nrlweb.cs.ucla.edu/publication/download/326/yjyi\\_medhoc\\_final2003.pdf](http://nrlweb.cs.ucla.edu/publication/download/326/yjyi_medhoc_final2003.pdf)

Yih-Chun, H., & Perrig, A. (2004). A survey of secure wireless ad hoc routing. *IEEE Security & Privacy Magazine*, 2(3), 28–39. doi:10.1109/MSP.2004.1

Yoon, H. (2002). Link stability and route lifetime in ad-hoc wireless networks. In *Proceedings. International Conference on Parallel Processing Workshop* (pp. 116–123). IEEE Comput. Soc. doi:10.1109/ICPPW.2002.1039720

Younis, O., Krunz, M., & Ramasubramanian, S. (2006). Node clustering in wireless sensor networks: Recent developments and deployment challenges. *IEEE Network*, 20(3), 20–25. doi:10.1109/MNET.2006.1637928

Yu, Q., & Heijenk, G. (2008). Abiding geocast for warning message dissemination in vehicular ad hoc networks. In *IEEE International Conference on Communications* (pp. 400–404). doi:10.1109/ICCW.2008.81

Zhang, J., & Zhang, Q. (2009). Cooperative Network Coding-Aware Routing for Multi-Rate Wireless Networks. In *IEEE INFOCOM 2009 - The 28th Conference on Computer Communications* (pp. 181–189). Ieee. doi:10.1109/INFCOM.2009.5061920

Zhao, J., Zhang, Y., & Cao, G. (2007). Data pouring and buffering on the road: A new data dissemination paradigm for vehicular ad hoc networks. *IEEE Transactions on Vehicular Technology*, 56(6 D), 3266–3277. doi:10.1109/TVT.2007.906412

## LIST OF PUBLICATIONS AND PAPERS PRESENTED

- Gin-Xian Kok, Chee-Onn Chow, Hiroshi Ishii. 2015. Improving Network Coding in Wireless Ad Hoc Networks. Ad Hoc Networks (available online).
- Gin-Xian Kok, Chee-Onn Chow, Hiroshi Ishii. 2014. Reducing Broadcast Redundancy in Wireless Ad Hoc Networks with Implicit Coordination among Forwarding Nodes. Wireless Personal Communications, 81: 253-278.
- Gin-Xian Kok and Chee-Onn Chow. Implementation of an Energy-efficient Broadcast Protocol for Wireless Sensor Networks. MJIIT-JUC Joint Symposium (MJJS 2012). Kuala Lumpur, Malaysia, 21-23, November, 2012.
- Gin-Xian Kok and Chee-Onn Chow. A Framework of Mobile Ad Hoc Networks for Disaster Mitigation. 5th AUN/SEED-Net Regional Conference on Information and Communications Technology (RCICT), Manila, Philippines, 18-19, October, 2012.