THE UNIVERSITY OF QUEENSLAND

AUSTRALIA

CONTEXT AWARENESS IN OPPORTUNISTIC COMPUTING

Ranjana Pathak

B.Eng (Computer science & Engg.), M.Tech (Computer science & Engg.)

A thesis submitted for the degree of Doctor of Philosophy at
The University of Queensland in 2015
School of Information Technology & Electrical Engineering (ITEE)

Abstract

Hybrid wireless mesh networks are infrastructureless networks. These networks can self-configure and self-heal and are therefore preferred candidates for the dynamic adaptive networks such as emergency response services, military applications and so on. In addition to that these networks have been recommended as a complementary technology for offloading the ever increasing data traffic from cellular networks. Around the world, wireless mesh networks have been deployed by the public safety sector as a way to establish essential communications for law enforcement personnel and to provide city video surveillance. Most applications in these networks use traditional end-to-end routing protocols however the performance of such routing protocols degrades due to network conditions such as mobility, heavy load and interference. These network conditions can be termed as network context information. In these network conditions if any link fails between a pair of nodes then it can fail the whole communication path. This is because a traditional end-to-end protocol communicates over a multi-hop connection when all the nodes between a source and a destination are connected at the same time.

Another paradigm of routing that does not require infrastructure support is opportunistic routing. In this routing mechanism nodes communicate over multi-hops even though connection among them is intermittent. Whenever nodes come in contact with each other they store and forward data. These routing protocols have degraded performance as compared to the traditional end-to-end routing protocol and they are characterized by long delays and lowered packet delivery ratio. These routing protocols can be applied to network situations where end-to-end route is not possible e.g. providing Internet connections in rural areas.

This thesis proposes a protocol that is a context aware integration of traditional end-to-end and opportunistic routing. The resultant hybrid protocol can utilize capabilities of both routing mechanisms. Such a hybrid protocol has the capability of dynamically switching between both the routing modes depending upon the network situation. This hybrid protocol can improve the performance of the network compared to the existing end-to-end or opportunistic protocols. To evaluate the performance of the proposed protocol a wide range of network situations from the connected to sparse networks are considered and they show significant protocol performance improvement in most cases.

To explore the possible potential of the hybrid solution, two variants of this protocol are investigated i.e. broadcast and unicast approaches with distinct algorithm designs. Both

variants of the hybrid protocol use the connectivity metric which is introduced to identify the potential forwarders in the network so that overhead can be reduced in the network. Performance of both the approaches are evaluated over a wide range of network situations and the results compared with a representative protocol from each side of the communication paradigms. The results show the proposed improved version of hybrid protocols can achieve significant improvement in terms of packet delivery ratio in any network situation.

Declaration by author

This thesis is composed of my original work, and contains no material previously published or written by another person except where due reference has been made in the text. I have clearly stated the contribution by others to jointly-authored works that I have included in my thesis.

I have clearly stated the contribution of others to my thesis as a whole, including statistical assistance, survey design, data analysis, significant technical procedures, professional editorial advice, and any other original research work used or reported in my thesis. The content of my thesis is the result of work I have carried out since the commencement of my research higher degree candidature and does not include a substantial part of work that has been submitted to qualify for the award of any other degree or diploma in any university or other tertiary institution. I have clearly stated which parts of my thesis, if any, have been submitted to qualify for another award.

I acknowledge that an electronic copy of my thesis must be lodged with the University Library and, subject to the General Award Rules of The University of Queensland, immediately made available for research and study in accordance with the Copyright Act 1968.

I acknowledge that copyright of all material contained in my thesis resides with the copyright holder(s) of that material. Where appropriate I have obtained copyright permission from the copyright holder to reproduce material in this thesis.

Publication during candidature

Ranjana Pathak, Peizhao Hui, Jadwiga Indulska, Marius Portmann and Wee Lum Tan, **Towards efficient opportunistic communications: a hybrid approach**, Proc. of PerMoby 2013 (PerCom workshop), March 2013, San Diego, CA, USA.

Ranjana Pathak, Peizhao Hui, Jadwiga Indulska, and Marius Portmann, **Protocol for efficient opportunistic communication**, The 38th IEEE Conference on Local Computer Networks (LCN), October 2013, Sydney, Australia

Ranjana Pathak, Peizhao Hui, Jadwiga Indulska, Marius Portmann and Saaidal R. Azzuhri, **A Performance Study of Hybrid Protocols for Opportunistic Communications**, Proc. of The 38th IEEE Conference on Local Computer Networks (LCN) (P2MNET workshop, October 2013), Sydney, Australia

Saaidal R. Azzuhri, Peizhao Hu, Jadwiga Indulska, Marius Portmann, Ranjana Pathak. OLSR -Opportunistic: **Towards a Better Approach of Hybrid Protocols in Multihop Wireless Networks.** (2013) submitted to Malaysian Journal of Computer Science (MJCS)

Publications included in this thesis

Ranjana Pathak, Peizhao Hui, Jadwiga Indulska, Marius Portmann and Wee Lum Tan, **Towards efficient opportunistic communications: a hybrid approach**, Proc. of PerMoby 2013 (PerCom workshop), March 2013 San Diego, CA, USA.
Incorporated as Chapter 4 and 5.

Contributor	Statement of contribution
Author Ranjana Pathak (Candidate)	Problem identification and Concept design (55%) Designed methodology (80%) Conducted experiment (100%) Wrote the paper (50%)
Author Peizhao Hu	Problem identification and Concept design (20%) Designed methodology (20%) Wrote and edited paper (20%)
Author Jadwiga Indulska	Problem identification and Concept design (10%) Wrote and edited paper (10%)
Author Marius Portmann	Problem identification and Concept design (10%) Wrote and edited paper (10%)
Author Wee Lum Tan	Problem identification and Concept design (5%) Wrote and edited paper (10%)

Ranjana Pathak, Peizhao Hui, Jadwiga Indulska, and Marius Portmann, **Protocol for efficient opportunistic communication**, The 38th IEEE Conference on Local Computer Networks (LCN), October 2013, Sydney, Australia
Incorporated as Chapter 6.

Contributor	Statement of contribution
Author Ranjana Pathak (Candidate)	Problem identification and Concept design (60%) Designed methodology (80%) Conducted experiment (100%) Wrote the paper (60%)
Author Peizhao Hu	Problem identification and Concept design (20%) Designed methodology (20%) Wrote and edited paper (20%)
Author Jadwiga Indulska	Problem identification and Concept design (10%) Wrote and edited paper (10%)
Author Marius Portmann	Problem identification and Concept design (10%) Wrote and edited paper (10%)

Ranjana Pathak, Peizhao Hui, Jadwiga Indulska, Marius Portmann and Saaidal R. Azzuhri, **A Performance Study of Hybrid Protocols for Opportunistic Communications**, Proc. of The 38th IEEE Conference on Local Computer Networks (LCN) (P2MNET workshop, October 2013), Sydney, Australia

Incorporated as Chapter 4 and 5.

Contributor	Statement of contribution
Author Ranjana Pathak (Candidate)	Problem identification and concept design (60%) Designed methodology (80%) Developed hybrid model (100%) Conducted experiment (100%) Wrote the paper (60%)
Author Peizhao Hu	Problem identification and concept design (20%) Designed methodology (20%) Wrote and edited paper (20%)
Author Jadwiga Indulska	Problem identification and Concept design (10%) Wrote and edited paper (10%)
Author Marius Portmann	Problem identification and Concept design (10%) Wrote and edited paper (10%)
Author Saaidal R. Azzuhri	Provided help in OLSR simulation environment (100%)

Saaidal R. Azzuhri, Peizhao Hu, Jadwiga Indulska, Marius Portmann, Ranjana Pathak. OLSR -Opportunistic: **Towards a Better Approach of Hybrid Protocols in Multihop Wireless Networks.** (2013) submitted to Malaysian Journal of Computer Science (MJCS)

Incorporated as Chapter 4.

Contributor	Statement of contribution
Author Saaidal R. Azzuhri	Problem identification and concept design (20%) Designed experiment (50%) Wrote the paper (60%)
Author Peizhao Hu	Problem identification and Concept design (10%) Designed experiment (20%) Wrote and edited paper (20%)
Author Jadwiga Indulska	Problem identification and Concept design (10%) Wrote and edited paper (10%)
Author Marius Portmann	Problem identification and Concept design (10%) Wrote and edited paper (10%)
Author Ranjana Pathak (Candidate)	Problem identification and concept design (50%) Designed experiments (30%)

Contributions by others to the thesis

Assistant of statistical analysis, interpretation of results and protocol development was provided by Prof Jadwiga Indulska, A/Prof Marius Portmann, Dr Peizhao Hu, and Dr Wee Lum Tan).

Statements of parts of the thesis submitted to qualify for the award of another degree

None.

Acknowledgements

I would like to express the deepest and most sincere gratitude to my advisory team, Professor Jadwiga Induska and A/Prof Marius Portmann. In addition to that I am also thankful to Dr Peizhao Hu and Dr Wee Lum Tan for all the invaluable guidance and advice they have provided to me over the past few years. Without their encouragement and endless support, I would not have been able to finish this thesis.

In particular I thank my principal supervisor, Professor Indulska who helped me in applying for the PhD program at The University of Queensland. She also helped me in getting a living scholarship from the UQ and National ICT Australia (NICTA). She introduced me to the field of context awareness, opportunistic computing and provided the best support throughout this thesis.

I would also like to thank Dr Peizhao Hu. He is one of the past PhD students of Prof Indulska. He helped me in developing my logical thinking. His encouragement throughout my PhD is invaluable.

Prof Indulska and Dr Peizhao Hu were always willing to offer help when I had problem with design, implementation and evaluation of the proposed protocols. They both helped me in writing research papers.

I would also like to thank my associate supervisor A/Prof Marius Portmann who was one of the networking group leader at NICTA QRL lab. He offered me very sound advice on a range of aspects related to my thesis.

I would also like to thank Dr Wee Lum Tan who was the Adjunct fellow at UQ. He has provided help in understanding the simulation environment and running experiments.

On a more personal side, I would like to thank my husband Mr Prashant Pathak who supported me through the whole journey of my PhD. Finally thanks to my amazing parents who provided their moral support and always encouraged me to work hard.

Keywords

Context awareness, opportunistic computing, wireless mesh networks, hybrid wireless mesh networks, routing protoocls, DTN, adaptive routing.

Australian and New Zealand Standard Research Classifications (ANZSRC)

ANZSRC code: 080502, Mobile Technologies, 50%
ANZSRC code: 080503, Networking and Communications, 40%
ANZSRC code: 080504, Ubiquitous Computing, 10%

Field of Research (FoR) Classification

FoR code: 1005 , Communications Technologies, 100%

Table of contents

List of Figures

List of Tables

List of abbreviations

ACK	Acknowledgement
AODV	Ad hoc On-Demand Distance Vector
CBR	Constant Bit rate
CTS	Clear To Send
DTN	Delay Tolerant Networks
GPS	Global Positioning System
IFQ	Interface Queue
MAC	Medium Access Control
MANET	Mobile Ad hoc Network
MID	Multiple Interface Declaration
MPR	Multi Point Relay
NS2	Network Simulator 2
OGM	Originator Message
OLSR	Optimized Link State Routing
PDR	Packet Delivery Ratio
QoS	Quality of Service
RQUEUE	Repaired Queue
RREP	Route Reply
RREQ	Route Request
RSSI	Received Signal Strength Indication
RTS	Ready to Send
SNR	Signal to Noise Ratio
TCP	Transmission Control Protocol
TTL	Time to live
UDP	User Datagram Protocol
Wi-Fi	Wireless Fidelity

CHAPTER 1

Introduction and Motivation

1.1 Motivation

Hybrid wireless mesh networks (HWMNs) are self-configuring and self-healing wireless networks. These networks consist of mesh client nodes and mostly static routers supporting multi-hop communication as shown in Fig. 1.1. These networks are the preferred candidate networks for emergency response services and other network environments that require dynamically adapting networks.

Currently, most of the applications use traditional end-to-end communication. In this approach, communication happens between a source node and one or more destination nodes, using end-to-end protocols such as TCP (Transmission Control Protocol) or UDP (User Datagram Protocol), supported by IP (Internet Protocol). However, due to dynamic nature of HWMNs, in some situations like mobility, low battery power or heavy load, network performance can degrade. For example in case of mobile client nodes it is possible that acknowledgements are not received by the host node within specified time limits. This leads to doubling timeout and slow start phase in TCP, and hence has negative impact on the quality of service provided by the network and may even lead to network throughput that is close to zero.

Opportunistic communication provides another way of communication among network devices for which communication links are only available intermittently. Whenever devices come into some close proximity i.e. within a range, they can exchange information. This type of communication does not require any infrastructure support (i.e. routing devices) while the traditional end-to-end communication can provide better network performance when a path from the source to the destination exists and the link quality is good. An op-

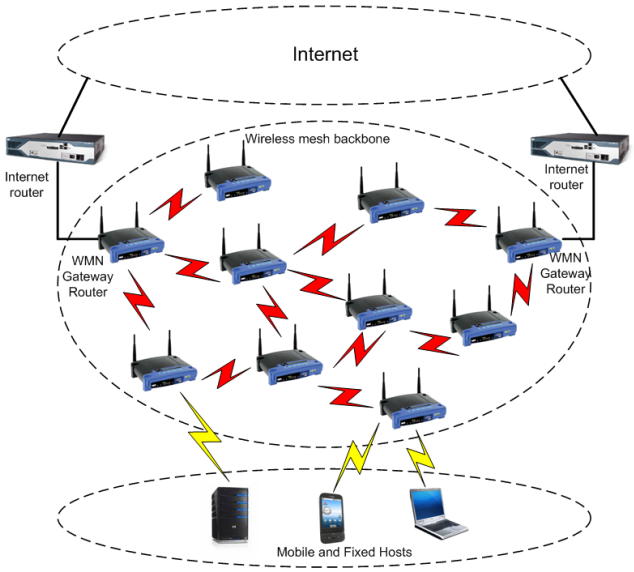


Figure 1.1: Wireless mesh network.

portunistic routing is better suitable for dynamic/sparse network situations when the path is often lost or cannot be established. The information like node and network mobility, battery level and network load, which is used to characterize the current network situation, can be considered as network context information as it describes the network situation.

There are existing protocols combining end-to-end and opportunistic routing. However they are designed for particular network situations and can not perform well in any network situation(s). The goal of my research is to develop a framework that can integrate opportunistic routing and traditional end-to-end routing for hybrid wireless mesh networks (HWMNs) in a context aware manner, and to design and develop a software infrastructure as a proof of concept prototype for this framework.

1.2 Problem statement

Due to the rapid growth in mobile device market, there is a significant growth in adoption of various wireless networking technologies. One of the technology, wireless mesh networks, has been recommended as a complementary technology for offloading the growing data traffic in cellular networks, deployed by the public safety sector to establish essential communications for law enforcement personnel [15] and to provide city video surveillance [31]. Recently the most popular use of these networks is to provide an alternate network model for Internet connectivity which is utilized in rural areas of the third world countries to help communities to access information[64].

The core component of these protocols is routing i.e. *the way data can be forwarded in the network to reach the destination*. In general, wireless links are sensitive due to multipath propagation in which a radio signal can reach the receiver in more than one path and over many tranmsissions from many wireless devices. This can cause interference which can then lead to lossy and unstable paths. Furthermore, nodes' mobility in wireless mesh networks causes link breakage when nodes move out of the transmission range. This causes connectivity pattern changes amongst nodes. Such volatile and intermittent connections can degrade the network performance if the traditional end-to-end routing protocols are applied. Such performance degradation is due to the fact that these protocols assume a connected path between a source node and a destination node such as AODV [70], OLSR [67], DSR [48] etc. In case of route failure when no alternate route is possible these protocols drop the packets destined for that route and this leads to lowered packet delivery ratio. In contrast opportunistic routing does not make any assumption regarding the existence of a contemporaneous path to the destination and also does not have any knowledge of destination's location or any other information related to it, in advance.

Opportunistic routing is a mode of communication in wireless networks in which data is transmitted in a **store-carry-forward** mechanism. For example, if two distant wi-fi enabled mobile users do not have any Internet connection and they want to exchange some data, then a wi-fi enabled bus moving on that way can be used as a carrier to deliver data. Here, movement of the bus becomes the opportunity to deliver data [25, 77]. Hence it is possible to take forwarding decision based on current network situation so that routing can minimize the impact of lossy and unstable links. Opportunistic protocols perform lower than a traditional end-to-end routing protocol due to the overhead involved in the data forwarding, however they are still capable of data forwarding when no end-to-end route is possible. Hence both routing modes have their own benefits and drawbacks. In this scenario another mode of routing emerges that is hybrid routing *that can combine capabilities of both routing modes together*.

There is existing research work in designing hybrid end-to-end and opportunistic protocols. It has been observed that these hybrid solutions are designed to handle only particular network situations. Hence they can not be applied to any network situation.

Primary goal of this research is to design and develop a hybrid protocol that can improve the performance of the network in a wide range of connection scenarios that are not covered by existing protocols. Such a hybrid protocol integrates the *store-carry-and-forward* capability

of opportunistic routing with the traditional end-to-end routing protocols in a context aware manner. Designing such hybrid protocol requires analysis of context information gathered from the various levels of the network. On the basis of gathered context information hybrid protocol can dynamically switch between routing modes whenever required.

1.3 Research contribution

The previous section discussed the problem statement that is the motivation to design and develop a hybrid protocol which can improve the performance of the network in a wide range of scenarios. To achieve this goal various research contributions are made in this thesis including the following:

1. Analysis of various types of routing protocols in wireless mesh networks and their performance under range of network scenarios,

   (a) Investigation of the functionality of the existing protocols in both modes i.e. traditional end-to-end routing and opportunistic routing.

   (b) Investigation of the network situations in which routing protocols degrades or upgrades their performance.

2. Analysis of context information that allows context aware integration of traditional end-to-end routing and opportunistic routing to enhance the performance of the protocol,

   (a) Examination of the factors at various levels of a network that can impact the routing decision.

   (b) Investigation of the network situations when switching between routing modes can enhance the protocol performance.

3. Design and development of a hybrid protocol that can improve the performance of the network due to dynamic switching capability between two routing modes i.e. end-to-end and opportunistic routing and can be a reactive or proactive routing protocols.

   (a) Investigation of the design principle of end-to-end routing protocols for reactive and proactive routing.

   (b) Design of generalized algorithms that can be applied to any end-to-end routing protocol to extend it to a hybrid protocol that has the capability of switching between routing modes to improve the performance of the protocol.

   (c) Development of a prototype in the form of a simulation model for reactive and proactive routing and its validation to verify the correctness of the simulation based protocols.

4. Design of a unique metric that can identify the potential forwarder(s) in the network that leads to the performance improvement of the hybrid protocol.

   (a) Analysis of the existing opportunistic approaches and their strategies to compute potential forwarder(s).

   (b) Investigation of the context information that can maximize probability to find the potential forwarder(s) in any network situation as compared to the existing approaches.

5. Design and analysis of broadcast and unicast versions of the hybrid protocol to further improve the performance of the protocol based on the proposed new metric.

   (a) Exploration of the potential of the proposed hybrid solution using the new metric.

   (b) Investigation of the impact of the transmission type (i.e. unicast or broadcast) on the protocol performance.

6. Development of a prototype of both broadcast and unicast based hybrid protocols in the form of a simulation model and its validation to verify the correctness of the simulation based protocol.

7. Systematic evaluation of the proposed protocols in various network situations using synthetic traces and real traces.

Remainder of the thesis is organised as follows. Chapter 2 discusses an overview of routing protocols and gives a brief introduction of context awareness that includes analysis of context information that can affect the protocol performance. Chapter 3 presents a critical literature review. Chapter 4 describes the initial concept of the hybrid protocol. Then subsequent Chapter 5 describes the protocol evaluation showing also the methodology and simulation environment required to evaluate the initial concept of the hybrid protocol. Chapter 6 describes the design of a new metric and also focuses on designing of two improved versions

of hybrid protocols and their systematic evaluation. Chapter 7 presents the conclusion and future work.

CHAPTER 2

Background

The previous chapter discussed the problem statement and research goal of this thesis work. As discussed the research goal of this thesis is the design and development of a hybrid routing protocol that can integrate end-to-end and opportunistic routing in a context aware manner to enhance the performance of the protocol. To properly introduce the research problem this chapter provides a background on end-to-end, opportunistic and hybrid routing protocols. In addition to that this chapter provides discussion on context awareness.

2.1 Routing protocols

One of the most important aspect of wireless communication is routing. Routing technique is responsible for finding the route/path from a source node to a destination node. There are many factors that encourage and obstruct routing decisions. All these factors depend upon the network situations and type of applications. In the field of wireless communication routing protocols can be categorised as traditional end-to-end, opportunistic and hybrid protocols. This section describes each of those routing protocols along with the factors that can affect them.

2.1.1 End-to-end protocols

Most of the applications use end-to-end routing protocols as they are suitable for fully connected wired networks. In wireless networks these routing protocols are also intended for mostly connected wireless networks. In these protocols the end-to-end route can be discovered between a source and a destination node. On the basis of route discovery these

protocols can be further classified as, proactive and reactive protocols.

2.1.1.1 Proactive routing protocols

In this class of routing protocols every node maintains a forwarding table which shows to which particular neighbour a packet with particular address should be forwarded. This table is periodically updated to reflect the network situation so that nodes have fresh list of possible routes in the network. To address the issue of scalability the number of nodes that store and forward global route information can be reduced in the network. Although these protocols have readily available routes they require high maintenance in high mobility scenarios. OLSR [23] is one of the proactive (table-driven) protocols, which maintains up-to-date link state information of nodes in the network. The routing table has route information for any destination. A technique, called *Link Sensing*, is employed to distribute the link state information (using periodic HELLO and Topology Control messages) of each node to the neighboring nodes. Alternatively, link-layer feedback is another way to populate the local link set. The link state information needs to be flooded through the network to keep each node's routing tables up-to-date.

In large networks, when each node frequently sends the topology information, it dramatically increases the protocol overhead. To reduce the overhead, OLSR nodes delegate the task of exchanging topology information (in the form of Topology Control messages) to a set of multi-point relays (MPRs). Each node chooses MPRs (supported by a MPR selection algorithm) from its one-hop neighbours that have *symmetric* connectivity to the node. Also, MPRs are those neighbours that completely cover the set of the two-hop neighbours of the node. The role of MPRs is to disseminate the topology information between other MPRs. Out of these control messages, HELLO messages are sent only to the one-hop neighbours, but the TC messages are forwarded by the MPRs in order to flood the entire network with topology information [46]. OLSR achieves optimal efficiency when the MPR set is as small as possible. Based on this topology information, any node in the network can compute the next-hop required by the routing table using the shortest path algorithm.

2.1.1.2 Reactive routing protocols

The purpose of these routing protocols is to minimize the control overhead of the proactive routing protocols. Therefore in this class of routing protocols route discovery is based on

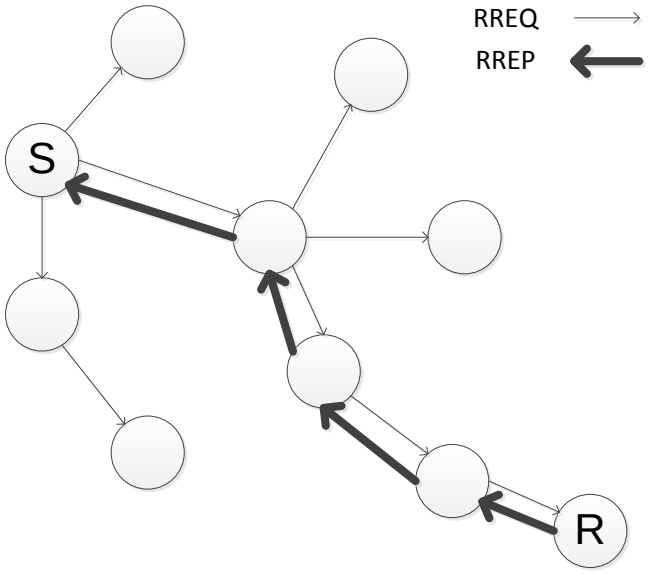


Figure 2.1: AODV route discovery

demand. Hence, if a node has packets to send, it initiates a route discovery process e.g. as in AODV (Ad-hoc on-demand distance vector)[70] designed for mobile ad-hoc networks (MANETs) or MESH networks. AODV performs routing in two phases; first phase is route discovery and other phase is route maintenance. For both of these phases it uses three control messages i.e. Route Request (RREQ), Route Reply (RREP) and Route Error (RERR).

When a node has a packet to send, it first checks the routing table. In case of no route, the router initiates a route discovery phase. In this phase the source node broadcasts RREQ. Upon receiving the RREQ, all one-hop neighbours of the source node create a reverse route to the source node and broadcast RREQ further until it reaches the destination. When RREQ reaches the destination node it generates RREP which is sent back to the source node along the reverse path (as shown in Figure 2.1). When a node receives RREP, router updates its routing table and marks route as active. RREQs that exceed their lifetime are discarded silently. Each routing entry has a lifetime, which is updated every time a packet passes through the route. When the lifetime of a route expires, the route is invalidated and subsequently removed from the routing table.

To detect link failures, AODV uses the periodic Hello messages (i.e., missing Hello messages). After detecting link failure, a local repair mechanism can be invoked. If an alternative route can not be created within a time window, Route Error (RERR) messages are sent along the affected path to invalidate the routing entry in all the affected nodes.

In AODV, the Link-layer detection is another approach to detect link failure. In this Link-Layer detection a node registers a callback function for each link a node has with its neigh-

bour(s). When the link-layer reports link failure the protocol uses the *local repair* mechanism, if possible. In case of no route being discovered, all the packets waiting for that route within the interface priority queue (IFQ) are dropped. While waiting for a route the packets are buffered in the *AODV's rqueue* (repaired queue), and a route repair is initiated. Although this class of routing protocols generates less overhead than proactive class, high latency time is present in the route discovery phase.

2.1.2 Opportunistic protocols

In the literature both terms such as DTN (Delay Tolerant Networks) and Opportunistic Routing both represents similar routing approaches. Hence, routing strategies designed for opportunistic or DTN networks address network situations in which end-to-end routing protocols can not work due to the lack of a route between source and destination nodes. This type of communication is suitable for the sporadically connected networks where a network experiences frequent network partitions as a result of high node mobility, stringent power management, load in the network, interference etc., [77, 69, 91]. Initially these protocols were designed for the network situations where connections among nodes were periodic e.g. satellite networks, underwater acoustic networks. Due to its intermittent behaviour it can be applied to sensor network applications such as wild life tracking and also for developing Internet connections for developing/rural areas.

In opportunistic routing a node which has a packet to send waits for an opportunity to exchange the packet with a neighbour and whenever possible the nodes continue sending the packet until it reaches the destination node. For example, if two distant Wi-Fi enabled mobile users do not have any Internet connection and they want to exchange some data, then a Wi-Fi enabled bus moving on that way can be used as a carrier. Here, movement of the bus becomes the opportunity to deliver the data [25].

There are routing protocols which do not need any knowledge of network topology and contacts. Essentially, these are flooding based routing strategies. Each encountered node attempts to deliver the packet and floods the entire network until the packet reaches the destination. This requires high usage of network resources like energy, bandwidth, storage etc., which severely degrades protocol performance in the presence of scarce resources. Some routing protocols take routing decisions based on their knowledge and don't replicate data packets unknowingly. In these routing protocols, the router estimates the delivery

probability for a destination node at each node. Whenever there is data to send, it selects the next node based on higher delivery probability for the desired destination node. This reduces significant overhead in the network by selecting forwarders for data transmission as compared to the flooding based routing.

Another issue involved in these routing protocols is *how many copies of a packet need to be transmitted to get the desired level of protocol performance*. To resolve this issue some protocols send only one copy of the packet. A copy of the packet is sent either to the destination or to the node that has a higher delivery probability to destination. These protocols can reduce significant overhead caused by flooding [81, 12, 63] i.e. by lowering the number of transmissions in the network, although an unsuitable forwarder could also degrade the protocol performance.

Unlike single copy, multi-copy routing forwards data on multiple paths that can maximize the packet delivery ratio or can get a desired level of delay. For example, routing protocols replicate packets whenever a node comes in contact with a node with higher delivery probability to reach desired destination node. This can maximise the chances of a successful delivery [16, 8]. Whereas some protocols limit the number of packet copies to get the desired level of delay [65, 56], to estimate the number of copies for a packet, requires context information related to the mobility pattern of the nodes. For example, in [83] the authors presented an analysis of various mobility models like, random direction, random way point and community based model in terms of contact time and waiting time among nodes. According to their study there is a direct relation between nodes mobility pattern and end-to-end packet delay because data transmission occurs when nodes come in contact with each other. An application can provide a delay tolerance factor and accordingly number of copies can be generated in the network.

The performance of opportunistic protocols is much lower than routing protocols that use end-to-end routes if a path from the source to the destination exists.

2.1.3 Hybrid protocol

Another growing category of routing are hybrid protocols. A hybrid protocol is the combination of different types of protocols. These routing protocols combine the capability of two different categories of protocols. Different categories of the protocols can address different network situations. Designing a hybrid protocol can address all the network situations.

Hence, it provides a more viable routing solution as compared to a stand alone routing protocol. Research on hybrid protocols already exists but the protocols have been designed with different motivations. For example, APTEEN [61], zone routing protocol (ZRP) [33] are hybrid designs of reactive and proactive protocols. Lakkakorpi et al. [53] also proposed an integrated routing protocol that combines DTN and AODV routing. According to this integrated protocol, before data transmission occurs, the router selects suitable routing mode either DTN and AODV, at the source node. In [21] authors proposed a hybrid of two different types of network coding techniques to improve the forwarding decision in opportunistic routing.

2.2 Context awareness

The previous Section 2.1 discussed different categories of routing protocols. It has been emphasized that routing protocols require information about the network. An end-to-end protocol wants to discover a route from a source to a destination either reactively or proactively by means of a multi-hop connection. On the other hand opportunistic protocols might require the knowledge of network so that they can select next node to route the data. All this information/knowledge is termed as context information in this thesis. In designing hybrid protocols, context information can play an important role because it can assist routers in selecting next node and/or particular routing mode.

This section briefly discusses context awareness and context models that have been developed for various kinds of applications in the field of networking. A widely used definition by Dey [29] defines context as *any information which can be used to characterise the situation of an entity*. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including themselves. Another definition of context given by Henricksen [40] says that *Context refers to the circumstance or situation in which a computing task takes place*. The term context is being treated as the environmental and/or situational information about an entity (e.g. user location at a given time, user activity) needed to complete relevant tasks.

Various types of applications require various types of context. For example in social relationship applications, a context is assumed to be a collection of information that describes the community in which users live and the history of social relationships among users [10]. In the field of opportunistic routing one of context aware routing protocols is a history based

opportunistic protocol (Hibop [8]). This protocol is designed for a social network. In this protocol router gathers users information and shares it with other contacts so that routers can take routing decisions based on that information.

2.2.1 Classification of Context information

Chen and Kotz[20] classified context information into three categories:

A. Physical context information

Refers to environmental factors that are usually captured using physical (e.g. GPS device to get the geographical position) or logical sensors [45]. Sensors can observe certain states of the physical world and produce raw data (e.g. a GPS position) in real time, which has to go through an interpretation process that transforms the sensor output into high-level information (such as a street name) i.e. logical sensor. One of the application of an Android application is *Travel&Local* which also gathers GPS co-ordinates and helps visitors/tourists to find the location of different places such as nearby fuel station, restaurant, etc.

B. Computing context information

Refers to the information that describes the resources available in the computing environment. This includes information such as the network connectivity and its characteristics (e.g. bandwidth, memory, etc.,), as well as available resources (e.g. projectors, printers, etc.,).

C. User context information

Reflects the characteristics and needs of users that are usually specified in the form of profiles or preferences. The user-supplied information is generally rich in semantics and updated rarely. Examples of user context information can include ownership of a PDA or a family relationship; the former changes infrequently and the latter is a persistent property, often maintained for a lifetime.

Context information [10, 41, 38] can also be classified based on the type of context information source as

A. Static e.g. Data of Birth, which is maintained for a life time.
B. Derived, generated from context information e.g. using GPS, measures the closeness from a certain destination.
C. Sensed, i.e. highly dynamic context which is prone to noise and sensing errors e.g. position on GPS, current network throughput, node mobility.

D. Profiled or user supplied, this information is initially reliable but later often out of date.

Context information can be imperfect because of possibility of failed sensor nodes, sensor error or communication problems. Imperfect context information may be of the following types:
A. Ambiguous: when similar kind of context is coming from different devices e.g. speed of vehicle measured by two different devices.
B. Unknown: if communication is broken then some context may be unknown. e.g. a reading from a sensor that could not be delivered.
C. Imprecise: sensors measurements always have some associated errors.
D. Erroneous: context may get erroneous due to noise in the network or due to human error e.g. measuring duration of any event.

Information on imperfection (quality of information) of context information is necessary for its management. For example outdated maps in the device can direct user towards the long route or no route.

Another important isue in context awareness is context modelling. Context models support design and implementation of context-aware applications. Context models describe types of context information required by the applications, the instances (facts) of these context types that need to be gathered and evaluated, and relationships between context information types. There exist models [10, 41, 38, 37, 39, 44, 6] for representing context information that attempt to capture types of context information, their relationships and also the quality of information. The model based approach for context aware-applications requires the designer to develop a context model for each application. These context models describe the context facts that need to be gathered and their relationships. These are mapped into internal data representations used by the repository managing context fact instances (e.g. a relational database used for object-role based context models). Context models can be categorised as: (i) Object based models, one of the languages used for these models is CML [19]. CML (Context Modelling Language) is an extended ORM (Object Role Modelling). Fact types are defined as relationships between entities. CML can also represent imperfect information, (ii) Spatial models, are fact based models for large scale spatial information. They require physical location as a context and this location can be a geographic location e.g. a GPS position or symbolic e.g. ID of a cell. This model is preferred for location based context

applications, (iii) Ontological models, these models are knowledge based and typically use OWL-DL to model context. (iv) Hybrid models, these models are constructed by integrating more than one type of models i.e. various types of context are modelled using appropriate modelling technique. In [6] C. Bettini et al. presented a comparison of context modelling approaches.

The existing models for modelling context information and user preferences can be improved. In particular, the research is required on context types (including user context, network context and device context) and their modelling and management techniques suitable for the wireless networks and opportunistic networking domains. The next section focuses on network context information that can affect routing decisions.

2.2.2 Context information affecting routing protocols

This section describes an analysis of various context types that can facilitate decision making on changing the routing mode from end-to-end to opportunistic routing in dynamic networks, such as Wireless Mesh Networks. Such a decision can be taken either at the beginning of communication or during the on going communications.

2.2.2.1 Nodes' mobility

In a wireless network, movement of nodes is an important factor that affects the routing performance. As already discussed in Section 2.1, a router discovers route from the source to the destination for end-to-end routing. In case of a proactive protocol a periodic update is required to update the link state whereas in a reactive protocol whenever there is data to send, router starts route discovery. Hence, proactive protocol is beneficial for those network situations where infrastructure is reasonably stable, for example the Internet, whereas reactive protocols are useful for those situations where nodes are mobile, for example routing in mobile ad hoc networks (MANETs). If nodes are highly mobile then both proactive or reactive protocols can degrade their performance because links among nodes change frequently due to the node movement.

Opportunistic communication is differently affected by mobility than end-to-end communication because movement generates contacts with other nodes and provides opportunity for exchanging information [82, 9]. Mobility can be classified based on mobility patterns and also on number of mobile nodes. These patterns can be of the following types:

Fixed pattern, where nodes follow a predetermined path, e.g. cars moving on a highway in vehicular networks, where cars follow the road.
Dynamic pattern, where nodes move around randomly, e.g. people roaming in a shopping mall.
Mixed pattern, where some of the nodes are static and others are mobile e.g. communication between a control centre of emergency services and a rescue crew (rescue team moves around randomly).

A network routing decision depends on the pattern of mobility. For example, if all nodes are moving in one direction at the same speed then it would not affect routing. On the other hand, if some of the nodes move in other direction at different speeds then it might result in broken links among nodes. Therefore mobility is an important context information that needs to be gathered to design a protocol that can cope with nodes' movements e.g. in CAR [63] the Kalman filter is used to predict next node.

2.2.2.2 Load

Load can be measured for a node and for a network. Load for a node refers to the traffic it experiences or applies to the network. Whereas, network load refers to the total amount of traffic applied to the network by all the nodes currently connected or currently communicating in the whole network [66], [22]. A node can be connected to many networks. Therefore, load on a node can be measured for a particular network or can be measured at each network [22]. Load on a node can be affected by resources available to that node like channel quality, CPU-cycle, bandwidth, etc. Quality of channel can vary due to communication range, environmental conditions, mobility and shadowing effect.

If a network experiences heavy load then it can lead to congestion resulting in longer delays. This affects the Quality of Service for real time traffic (video, audio) and also the delivery of non real time data. For example, performance of TCP can be severely affected by late or lost acknowledgements [84]. Therefore load measurements can play an important role in protocol adaptations.

In [36, 35] authors propose a hybrid routing strategy. According to authors, opportunistic communication can be used to offload the traffic from a cellular or mobile network. To decide what kind of data can be sent via opportunistic communication, router prioritises the information travelling in these networks. Lower priority data such as multimedia newspa-

pers, weather forecast, movie trailers etc. can be routed via opportunistic communication.

2.2.2.3 Link quality

In wireless networks a packet is transmitted via the broadcast medium. Hence that medium is shared by all the other devices and that causes problems, such as the hidden terminal and exposed terminal problems at the MAC layer. Due to such problems a transmission can be involved in a collision and hence packets can be lost. Sometimes collisions can create congestion that can partition the network. In these situations a routing protocol can degrade its performance.

Also with growing popularity of wireless devices most people carry their wireless devices with them to remain connected. This can lead to unavoidable situations where there is radio signal interference due to presence of other wireless device(s). Buildings and construction sites can also interfere with the communication system and degrade the signal quality. As a result, links among nodes become inconsistent and unpredictable which results in disconnections among nodes. To avoid such situations a router can gather context information which can assist it to avoid such paths with poor links [58, 88] like geographic location, RSSI, SNR etc. Thus, for wireless environments, channel quality depends on many factors [90] including: transmission range, signal strength, environmental factors, interference, mobility and shadowing effect. If a router is capable of gathering and using this context information then it will have a better quality connection.

2.2.2.4 Quality of Service

For any application, it is important to ensure that the desired Quality of Service (QoS) is provided by the communication protocol [60, 62]. Generally QoS is represented by bit rate, delay, and jitter. Different traffic classes have different QoS requirements and based on that traffic can be classified into four classes: (i) Conversational class, (ii) Streaming class, (iii) Interactive class, and (iv) Background class. Among them conversational and streaming classes represent real time traffic whereas interactive and background classes represent best effort traffic. As it was discussed, opportunistic routing is based on store-carry-and-forward principle. Therefore these routing protocols are not suitable for those applications which have stringent time requirements, For example, real time applications.

In order to design a hybrid protocol, it is important to know whether the application sup-

ports both the routing protocols as well as the ability to switch dynamically between these protocols. In dynamic switching a routing mode can be selected at the beginning of communication [53] or can be applied for an on-going communication session[73].

2.2.2.5 Network resources

Routing protocols are bounded by provided network resources[76] such as energy, bandwidth, computing capability, storage etc. In the situation of scarce resources a routing protocol cannot work well and this can negatively impact on the network performance. For example, if high bandwidth and storage are available then flooding based routing has no negative side effects, but with no knowledge of network situation it can congest the entire or part of the network. In real life scenarios, routing based on unlimited resources is unrealistic. A smart phone has limited battery power and storage so it is important to carefully manage its power and storage capacity. For hybrid protocols that have the capability of storing data packets and forwarding them in a hybrid manner, they must use network resources efficiently.

2.2.2.6 Application specific

Another important factor in selection of routing protocols is the type of application. The users might be aware of a network situation in advance and prefer a particular routing mode based on their experience. For example, in case of a fire, high temperatures can interrupt connection among nodes. The user might prefer opportunistic routing over end-to-end routing in this case. In another example, a user may prefer cheaper communication and is ready to compromise the Quality of Service. Recently the popular application **whatsApp** which is used by many mobile users is an example of such a service. Users use it to exchange messages without paying any extra money but the application does not give any guarantee of the delivery of these messages.

In another example, TCP based applications require an end-to-end routing protocol because it is a reliable protocol that gurantees packet delivery for applications and also has very poor performance if packets are lost or delayed, e.g Bootstrap (BOOTP) which is a dynamic method to associate workstations with a server.

2.3 Summary

This chapter presented an analysis of two routing modes i.e. end-to-end and opportunistic routing. In addition to that it also presented a brief introduction of context awareness in general and discussed the network context information that can be utilized when designing hybrid routing protocols. The next chapter will present a discussion on the existing routing protocols proposed in both routing modes.

CHAPTER 3

Literature review

As discussed in the last chapter, there are various end-to-end and opportunistic routing protocols for wireless mesh networks. However none of them has good performance in all network situations. Last chapter also provided a brief introduction to context awareness and how it can be useful in designing dynamically adaptive routing protocols. In this chapter, a critical literature survey is carried out to investigate how the state-of-the-art routing protocols in wireless mesh networks adapt to dynamic network situations. Discussion in this chapter addresses opportunistic routing, hybrid routing and routing metrics.

3.1 Opportunistic routing

As discussed in Section 2.1.2 opportunistic routing is designed for network situations when a complete route from a source to a destination is not possible. A node buffers a packet until it finds a suitable forwarder that can deliver that packet to the destination or towards the destination. Due to this, routing protocols have two important issues which need to be addressed. First: what context information is required to determine the forwarder node and second: how many forwarders need to be selected.

Traditional end-to-end routing protocols rely on their route information that can guarantee successful delivery. In these protocols only one forwarder is identified on the basis of route information. It is the reason for their degraded performance when that single copy of packet is lost due to a broken link which can be caused by various reasons (refer Section 2.2.2 for details). End-to-end protocols do not perform multipath routing because the primary goal of these protocols is to provide delivery while minimizing the overhead in the network i.e. AODV [70], OLSR [23]. Opportunistic protocols can also be classified as per their goals such

as to improve **packet delivery ratio (PDR)** or to improve **throughput**. Opportunistic protocols can be categorised as context oblivious or context based [10].

Context oblivious

Context oblivious routing can work without any knowledge of the network situation. It is based on the assumption that mobility in a network is random. The protocol floods the entire network with the copies of the packet to deliver at least one copy of the packet to the destination node. Flooding is performed to maximize the chances of successful delivery to the destination node. These strategies are also known as dissemination based routing.

Context based

These protocols gather context information from the different levels of the network protocols. Router determines potential forwarder based on the gathered information [7, 80, 55]. These routing protocols are intended to minimize the overhead generated due to a flooding based protocol that can severely affect the network performance. There is a large set of context information that can be gathered as per the routing protocol's strategies [8, 10, 13, 11, 14]. For example, protocols designed for social network gather context information such as a name of the person, address, age, interest etc. Some protocols gather other details regarding the contacts such as frequency of meeting, duration of the meeting etc.

3.1.1 Epidemic

The epidemic protocols represent a class of routing protocols for opportunistic routing designed for sparse/highly mobile networks. They adapt the store-carry-forward mechanism for delivering data packets, in a way similar to the spread of an infection. It is known that infections are transmitted from an infected person to an uninfected one by their coming in contact with one another. Similarly, in an epidemic protocol, packets are treated as an infection and when node encounters a neighbour it attempts to transmit stored packets to it.

In case of the low traffic network condition these protocols can give low end-to-end delay at the cost of other network resources such as buffer size, bandwidth, and transmission power. However in heavy load conditions they can congest the links or exhaust resources [89]. Various variations are proposed to trade-off such issues such as k-hop [32, 78] forwarding: a

packet can be transmitted to maximum k-hops for successful delivery to the destination node. In probabilistic routing [57, 34] a packet is transmitted with a delivery predictability. Here delivery predictability is the probability to reach a known destination. When nodes meet they exchange and update their probability information. When a node has a packet to send then it takes the forwarding decision based on delivery predictability of their neighbour nodes.

One of the proposed epidemic protocols [86] does not have any knowledge of network or contacts except the periodic or random node contacts. Whenever nodes meet they exchange packets for further transmission. This can get higher PDR gain but at the cost of high overhead in the network. This overhead requires high use of network resources i.e. bandwidth, energy and storage etc. In this protocol nodes do not accept packets they have seen in the past as a fail-safe mechanism.

These routing protocols are not only designed to improve the protocol performance in terms of the maximum packet delivery ratio but also apply different strategies to efficiently utilize network resources. For example, techniques to release the buffer space occupied by packets which are already delivered to the destination node as discussed in [86, 89].

3.1.2 PROPHET

PROPHET [57] is the evolution of the epidemic routing scheme. It reduces significant overhead in the network by introducing the concept of delivery predictability. Here, delivery predictability for a node is the probability to encounter a certain destination and node(s) which can be a forwarder if the delivery predictability is high. This protocol is a partially context-aware routing protocol and requires contact frequency.

PROPHET is advantageous in minimizing the network overhead as it tries to optimize selection of the next node. One of the drawbacks is that it identifies the next node on the basis of contact frequency which is not sufficient to identify the best forwarder in the network [56].

3.1.3 Spray and wait (SAW)

SAW [82] is one of the routing protocols designed to get the desired level of throughput by minimizing the number of transmissions in the network. It functions in two phases as suggested by its name:(i) spray and (ii) wait. In the spray phase, protocol initially spreads

L number of copies of a packet. In the wait phase, if that packet is not received by the destination then it can be kept for direct transmission to the destination.

To get the expected delay one of the important issues that need to be addressed in this protocol is: *how many copies of the packets should be initially generated in the network*. T. Spyropoulos et al. discussed different ways to determine that number L, with and without knowledge of network parameters in [82].

Compared to the flooding based routing this protocol can reduce overhead by reducing the number of transmissions due to limited copies of a packet that are generated in the network. One of the drawbacks of this protocol is that it relies on a high degree of node mobility. A packet can be directly delivered to the destination node via a relay node after the spray phase. To minimize the overhead, limited number of copies are sent in the network that causes lower PDR gain for the SAW protocol.

3.1.4 ExOR

ExOR [7] is a cross layered routing protocol that utilizes a MAC layer feedback. In this protocol source node selects the next node after transmitting a packet. For such selection this protocol maintains network wide information of connectivity among nodes.

This protocol transmits packets in a batch. Each packet has a map of batch as well as a list of candidate forwarders provided by the source node. When a packet is broadcasted then all the neighbour nodes add or update their batch maps for the packet accordingly. One of the neighbour nodes that has the highest priority to the destination can broadcast that packet further. At the same time other nodes know which packet is not yet forwarded and if it exceeds its time limit then it can be transmitted only via end-to-end route, if possible.

One of the drawbacks of this protocol is that it cannot support different kinds of multicast traffic. Another drawback is that it underutilizes the MAC layer feedback because only one forwarder is selected to broadcast packets. Inaccurate estimation can degrade its performance. This protocol is not suitable for dynamic network situations because each packet carries a batch map and a list of potential forwarders that require co-ordination among nodes.

3.1.5 MORE

The MAC independent opportunistic routing protocol (MORE) [17] is designed for stationary mesh networks to overcome the limitation of the ExOR [7] using network coding techniques. It leverages the ETX (expected number of transmissions) path metric proposed in [27] to determine the delivery probability to reach the desired destination node. According to this protocol, each node periodically checks its connections to other nodes and updates their delivery probability. It also works on batches of packets where packets are coded. Sender keeps sending coded packets until it receives ACK from the destination node. A relay node stores a new packet and can forward it, if it is the forwarder node for that packet. Similarly the destination receives only new packets and sends back ACK using best path routing.

This heuristic based approach is designed for stationary wireless mesh networks and requires coordination among nodes. This can not be applied for dynamic network situations.

3.1.6 HiBop

HiBop [8] (history based opportunistic routing protocol) is an opportunistic protocol completely relying on the gathered context information. It is specifically designed to improve the performance of the protocol in terms of PDR.

In this protocol, the router gathers information of contacts in an identity table (IT). Nodes share their ITs among each other at every encounter. Whenever a node wants to send some data to a destination then the router selects the best forwarder which has higher probability to get to the destination node. Here higher probability is assigned to a node that shares similar context information with the destination node. For example, if a destination belongs to place P and joins some social community C, then forwarder can be the one who is a member of the same social community or belongs to the same place. This routing protocol requires large set of context information and is dedicated for social networks.

3.1.7 CAR

CAR [63] is a partially context aware routing protocol designed for DTN. It is a utility based protocol where utility is computed by the Kalman filter based predictions. The Kalman filter prediction is a time series analysis based on a state space model that has capability of

evaluating DTN scenarios without storing the entire past history of the system. For this filter only locally available information is required. Nodes in the network proactively compute their delivery probability and at each encounter share it with neighbor(s).

CAR assumes underlying mobile ad hoc networks (MANET) and disconnected groups of nodes are known as clouds. In CAR, for transmitting a message from one node to another node where both nodes belong to different clouds, a sender node selects a node from the current cloud with the highest delivery probability which can successfully deliver message to the destination node.

However the CAR protocol also has a few drawbacks. Firstly, it only selects one forwarder. Hence, unsuitable forwarders can degrade the protocol performance because only one copy of the packet exists. Secondly it uses Kalman filter prediction technique which requires comprehensive mathematical computation and therefre requires resources.

3.1.8 Robust Replication Routing (R3)

R3 [85] is one of the replication based schemes that claims to cope with the disconnections in the network. In this protocol distribution of path delays is estimated as compared to the most DTN protocols that monitor expected delay for an application. On the basis of path delays a packet can be replicated in the network to meet the desired level of delay. If load condition changes at any path then replication can be affected. R3 would not replicate packets when the actual delay is considerably higher than the estimated delay.

Hence it can improve the delay parameter but can suffer in packet delivery ratio [65] due to limited copies of a packet as compared to flooding based routing e.g. epidemic protocols [86]. Analysis of mobility patterns to estimate the path delays needed for this protocol can also limit its scope.

3.1.9 SEDUM

SEDUM [56] is a recently proposed opportunistic protocol. This protocol is designed to get the desired level of end-to-end delay. It has the following issues that need to be addressed, (i) selection of next node (ii) number of packet copies to be generated.

In this protocol, the next node is selected on the basis of utility metric. For this metric nodes keep records of contacts frequency and their duration over a specified period of time. When-

ever node encounters other node(s) they compute and share their utility vector. They also have utility for an indirect contact using gossip i.e. multi-hop information. Hence its metric can measure next node with higher probability of successful delivery to the destination as compared to approaches where only one attribute i.e. contact frequency is considered. However sharing of such information also causes overhead.

As mentioned, the goal of SEDUM is to get a desired level of end-to-end delay. In order to achieve the goal it is required that it analyses the mobility pattern of nodes and delay tolerance factor of the application. The analysis of the mobility pattern measures contact frequency and contact duration so that end-to-end delay can be estimated on a path. So SEDUM can select a path as per the delay tolerance factor of the application and replicate packets on that path to reach the desired destination. SEDUM can improve on end-to-end delay but its packet delivery ratio suffers due to limited copies of a packet. One of the drawbacks is that it cannot cope with uncertain changes in the network that impact on delay estimation.

3.2 Hybrid routing

As discussed in Chapter 2 hybrid routing represents a class of routing protocols that can combine different routing protocols together. A hybrid protocol has a capability to address network situations which are not covered when stand alone protocols are used. For example, for traditional end-to-end routing a path from source to destination is mandatory and in absence of such route the protocol's performance diminishes. On the contrary, in opportunistic routing, data is transmitted on hop-by-hop basis when no route is possible. Opportunistic routing protocols generate overhead in the network that causes lowered performance as compared to an end-to-end routing protocol.

In the literature different types of hybrid protocols exist but our focus is on the routing protocols aiming to combine traditional end-to-end and opportunistic routing.

3.2.1 MaDMAN

A. Petz et al. [73] proposed a cross layer architecture MaDMAN (a Middle-ware for Delay-Tolerant Mobile Ad-hoc Networks). Authors have shown that MaDMAN has a set of protocol stacks so that DTN and traditional end-to-end routing can co-exist. Whenever protocol

performance degrades due to a dynamic network situation the application can switch to a different protocol stack. According to the design this architecture is made up of four components, (i) application interface: Allows selection of network interface according to the application, e.g. socket interface for TCP/IP based applications or bundle interface for handling DTN applications. (ii) Context aggregator: A component which can gather context information from various layers of network protocols and provide it to the session manager. (iii) Connection logic: A session manager has connection logic based on the provided context information so that it can switch connections whenever required. (iv) Transport, Network and Routing: MaDMAN has a collection of many protocols. With so many combinations of these protocols possible, the connection logic protocol can be selected for an application.

The MaDMAN design concludes that different protocol combinations can improve network performance as compared to a stand alone protocol. The basis of the presented hybrid model is the context information gathered from the different protocol layers. The authors also discussed open research issues such as switching of TCP connections, validation of protocol stacks, etc. Therefore one of the limitations of this architecture is that it is the preliminary approach where context and its sources are not discussed. Another limitation of this architecture is the selection of a suitable protocol stack. Although MaDMAN has the session manager that is responsible for engaging the suitable protocol stack - the decision of switching is initiated by either of the end nodes. Whereas proposed hybrid protocol in this thesis has the capability of dynamically switching between routing modes at any node in the network.

3.2.2 HYMAD

HYMAD [87] is also a hybrid approach that operates on groups of nodes. It combined two types of protocols. One of them is the mobile ad-hoc networks (MANET) protocols that address the routing in fairly stable or connected networks. Another is delay tolerant protocols that address the routing in sparse networks. Main focus of this approach is to determine the cluster of nodes i.e. group of connected nodes. Within a group, nodes use end-to-end routing protocols; whereas between disconnected groups an opportunistic/DTN routing protocol is used. Each connected group has one edge node that can communicate with other groups' edge nodes. These edge nodes use the Spray-and-Wait routing to exchange packets.

In HYMAD, each node shares the knowledge of what packets they have for all the edge

nodes. Whenever the edge nodes of two groups meet, they check whether there are packets that need to be sent to a node in the other group. If so, the edge node notifies the group member about the opportunity.

If connections amongst the nodes change then HYMAD needs to reform the grouping. Therefore it also uses a metric to limit the nodes group size so that appropriate number of nodes can be selected. This protocol needs to have updated group of nodes to reflect the current network situation. The grouping scheme of this solution means that the protocol works better in scenarios where mobility within a group is relatively low.

In the evaluation the authors have shown that HYMAD can outperform Spray-and-Wait in various mobility scenarios in terms of PDR and delay. But when the network becomes sparse then it can degrade the protocol performance as compared to Spray-and-Wait and epidemic protocols due to the grouping schemes.

3.2.3 Integrating DTN and MANET

Ott et al. [68] proposed a hybrid DTN-MANET approach. Such integration is to incorporate DTN routing in a MANET routing protocol i.e AODV [71] so that when a path to the destination breaks and cannot be repaired DTN routing can be performed.

To integrate DTN capability inside AODV, control packets of AODV are modified. The modified control packets not only discover end-to-end route but also discover DTN-capable nodes in the vicinity. DTN capable nodes are the mobile nodes willing to perform DTN routing. Hence whenever alternate route is not found data can be forwarded via those DTN-capable nodes.

Authors proposed integration of DTN-MANET but one of the drawbacks is that switching from AODV to DTN is always performed at the source node and switching back is not supported. Hence once the mode of communication is switched for a communication session, it remains in that mode for the whole communication session lifetime.

3.2.4 Adaptive routing

In [53] the authors proposed an adaptive routing method. According to this method a routing mode is selected before transmission. Selection of either end-to-end or opportunistic routing depends on metrics which indicate the estimated lifetime of the link and the re-

quired time for successfully completing the file delivery to the destination. Once the mode of communication is identified, packets are sent using that communication method only. In the case of link failure, the selection processes will be re-evaluated.

It is observed that these approaches tend to switch over to the opportunistic communication paradigm for the lifetime of the packet flow when packets are dropped due to link failures. Hence mode of communication switches only once in these approaches [68, 53].

3.2.5 Hybrid proactive protocols

SF-BATMAN [28] (Store and Forward BATMAN) is an attempt to extend BATMAN (Better Approach to Mobile Ad Hoc Network). BATMAN is a proactive MANET protocol in which each participating node periodically broadcasts Originator messages (OGMs). All nodes keep track of OGMs they have received in a specified time window, so that the next node can be selected based on maximum OGMs received from a node.

SF-BATMAN is the DTN extension of BATMAN. The extended BATMAN has capability to buffer packets that can be dropped by BATMAN due to a broken link. Whenever possible (i.e. link available) SF-BATMAN will forward buffered packets. It is a single copy scheme so only one copy of message is forwarded in the network.

The addition of the store-and-forward capability in SF-BATMAN is simple to implement and has lower overhead as compared to the BATMAN. Evaluation shows that PDR improvement due to the store-and-forward feature in BATMAN is about 10%.

Similarly authors in [74] proposed an extension of a proactive MANET routing protocol i.e OLSR and BATMAN by incorporating the store-and-forward functionality in the protocol. To evaluate the performance of the proposed hybrid versions of proactive protocols, initially the performance of OLSR and BATMAN is evaluated by varying MaxLinkTimeout in various mobility patterns. The authors have concluded that OLSR outperforms BATMAN because OLSR can more quickly repair route. To evaluate the impact of hybrid MANET-DTN the authors compared performance of unmodified and the hybrid versions of both protocols (OLSR and BATMAN). In this set of experiments the authors concluded that hybrid versions of the protocols can give higher PDR due to the store-and-forward mechanism where stored packets can be re-sent when link is available. They have also verified that the hybrid version of OLSR also outperforms the hybrid version of BATMAN.

Both the designs proposed in [28, 74] are preliminary - they leverage the buffering mechanism but presented evaluations are preliminary.

3.2.6 Dt-dymo

Dt-dymo [52] is one of the hybrid protocols that is an extension of the Dynamic MANET On-demand routing protocol (DYMO) [18].

DYMO is a reactive MANET routing protocol and considered an extension of AODV [71]. Unlike AODV, when a RREQ message is broadcasted to discover the route in DYMO, it also carries information about all the nodes it passes through in sequence. When nodes receive such RREQ they also update their route table. Route expiry is updated when data transmission occurs via that route. Such enhancement in DYMO are made to be incorporated within the mobile network. Still DYMO does not work well in highly mobile network situations.

Dt-dymo has DTN capability in addition to the end-to-end DYMO MANET protocol. It can buffer packets if end-to-end route is not available. Later on buffered packets can be sent towards the node which has higher delivery probability to reach the destination. Delivery probability is computed and shared among nodes whenever they come in contact with each other. Hence it is based on the assumption that nodes are mobile.

Dt-dymo has the capability of dynamic switching between routing modes. One of the drawbacks of this protocol is that the computation of delivery probability can not represent network situation accurately as it is relying on contact frequency. Existing research work in [24, 83, 56] suggested that estimation of delivery probability based only on contacts often mislead data forwarding decisions. This is due to the lack of capability in representing the current network situation or connection patterns.

3.3 Routing metrics

Hybrid wireless mesh networks are dynamic in nature and various routing metrics are proposed for these networks. A good routing metric designed for such networks needs to follow some of the criteria i.e. interference, locality information, load balancing, agility, isotonicity and throughput etc. In HWMNs interference can be intra-flow, inter-flow and external. Intra-flow which is due to radio links using the single path. Intra-flow interference can be reduced by increasing channel diversity. Inter-flow is due to various flows competing on same

channel which is harder to control. Whereas external interference is due to some external factors that can be controlled and uncontrolled. Locality information involves local information that is required to compute a metric e.g. noise level in the network. Load balancing also an important aspect that should be considered so that network resources are fairly utilized in the network. Agility is the capability of a metric to quickly and efficiently respond to the network situation. Isotonicity ensures that if a path is appended and prefixed by a common path than order of weights of two paths is preserved. Another criteria is throughput where metric should be able to consistently select higher throughput path.

This section describes the various existing routing metrics and presents their pros, cons and feasibility to implement a particular metric.

3.3.1 Hop-count [43]

One of the traditional routing metrics is the hop count which is used in most routing protocols i.e. AODV, DSDV, DSR etc.. It is the simplest metric that can find the shortest path with the smallest number of hops. This metric treats all the links alike therefore often leads to the poor performance of the protocol. However it can be easily implemented.

3.3.2 ETX (Expected Transmission Count)[47]

This metric counts expected number of transmissions required for a successful delivery over a wireless link. As a result it can select path with higher throughput and lower number of hops. ETX deals with the inter-flow interference indirectly. However it can only be applied for a single channel multi-hop wireless network. ETX does not consider difference in transmission rates, hence cannot reflect the actual traffic loss rate. ETX is based on the delivery ratio where each node remembers number of probes needed for a successful transmission. Therefore it is not as simple to implement as hop-count.

3.3.3 ETT (Expected Transmission Time)

It computes expected transmission time required for a successful transmission. For such computation it requires packet size and link bandwidth. As compared to ETX, it can increase the network performance. However, it retains many drawbacks of ETX. Also its implementation is not simple as it requires link's bandwidth and loss rates (for both forward

and reverse direction). To determine the bandwidth several methods are proposed in literature and none of them can accurately compute bandwidth. It is because the computation is based on several assumptions ignores several factors that affect packet delivery time.

3.3.4 WCETT (Weighted Cumulative Expected transmission Time) [30]

It is an extension of ETT metric and it can efficiently consider the channel diversity and intra-flow interference. As already discussed ETT computation is complex, evaluation of WCETT also retains this disadvantage.

There are some other routing metrics [47, 50, 54] such as Metric of interference and channel switching (MIC), Load Aware Expected Transmission Time (LAETT), Exclusive Expected Transmission Time (EETT), Interference Load Aware metric (ILA), Interference Aware metric (iAWARE), and Multi-Hop Effective Bandwidth Based Routing (MHEB). All the above mentioned routing metrics are designed for a proactive routing protocol in which node's movement is limited. Due to such limitations these metrics cannot perform well in a highly mobile network.

3.4 Summary

The goal of this thesis is to design and develop a context aware integrated routing protocol for wireless mesh networks that can operate across a wide range of network scenarios which are not addressed by existing protocols. Towards this aim this chapter discussed existing routing protocols and context information for their smooth functioning.

Traditional end-to-end and opportunistic routing protocols are tailored to address completely different network situations. Traditional end-to-end routing protocols cannot work without a complete route between a source node and a destination node. Contrary to this, in opportunistic routing a node transmits a packet to another node in a store-carry-and-forward fashion on the basis of opportunity defined by the protocol.

Key routing strategies and metrics discussed in this chapter are devised to address particular set of network situations with/without the requirement of specific context information. Opportunistic protocols which do not require any specific context information are mostly dissemination based protocols that can cause high overhead in the network. To trade off overhead opportunistic protocols apply different techniques to identify nodes in the net-

work allowed to forward packets. To define opportunities or to identify forwarding nodes, protocols measure delivery probability for a known destination so that node(s) with higher delivery probability can participate in packet forwarding. The next chapter presents the design of a hybrid protocol that can overcome the limitations of existing protocols.

CHAPTER 4

Hybrid protocol design

Previous chapters presented the discussion on traditional end-to-end routing and opportunistic routing protocols and the impact of network dynamics on the performance of the routing protocols. Chapter 3 discussed existing routing protocols for end-to-end and opportunistic routing to deal with particular network situations.

In this thesis work a hybrid routing protocol is proposed that can combine the capabilities of both the traditional and the opportunistic routing so that it can improve the performance of the routing in a wide range of network situations. To achieve this goal both the routing modes i.e. traditional end-to-end and opportunistic routing are integrated together so that the resultant hybrid protocol can dynamically switch between any of the routing modes as per the network situation. Hence it can widely address network situations that are not covered by the existing routing protocols. To systematically present the design of such hybrid protocol this chapter first discusses the issues related to traffic types and transport protocol. Later in the chapter detailed design of the proposed hybrid protocol is presented.

4.1 Traffic types and transport protocol

To design a routing protocol that has the capability of switching routing modes at the network layer it is important to know what kind of applications are suitable for such a switching mechanism. An application and its required traffic type can be analysed on the basis of its Quality of Service (QoS) requirements that includes required bandwidth, acceptable delay and jitter. As discussed in section 2.2.2.4 on the basis of QoS requirements alone, there are four classes of traffic:

Conversational class, e.g. voice applications.

Streaming class, e.g. video streaming applications.

Interactive class, e.g. web browsing.

Background class, e.g. data, emails.

Among these classes conversational and streaming classes belong to the real time traffic. Hence these applications have stringent Quality of Service (QoS) requirements for their communication. On the other hand interactive and background classes belong to the best effort traffic class. An application that supports best effort traffic does not have strict constraints on bandwidth, jitter, and delay.

In this thesis the proposed hybrid protocol is a context aware integration of end-to-end and opportunistic routing and the scope of this thesis is that the real time is traffic treated as one class. Traditional end-to-end routing is characterized by a routing mechanism in which a route exists between source/destination pair whereas in opportunistic routing no such route is discovered. As a result opportunistic routing causes delay. Hence if an application does not have any contraints over delay then it can use this proposed hybrid protocol. For example **WhatsApp** is a popular smart phone messenger service that has priority of delivering a message without any delay constraints.

After determining the suitable applications for such a hybrid protocol the next step is to find its suitability to the transport layer protocols. Many traditional end-to-end applications use TCP connections because TCP provides reliable and ordered delivery of the data. Due to opportunistic routing integrated into the proposed hybrid protocol it can not support timely or ordered data delivery. Therefore the hybrid protocol is not using TCP connections. While UDP is not reliable it can be modified to incorporate reliability [87].

4.2 Design principle

The goal of this hybrid protocol is to improve the performance on the network in any network situation. Such a hybrid protocol is designed by integrating opportunistic routing with end-to-end routing so that data can be transmitted via traditional end-to-end routing for the connected part of the network and for non connected part of the network packets can be routed via opportunistic routing. Such hybrid protocols can be designed for both

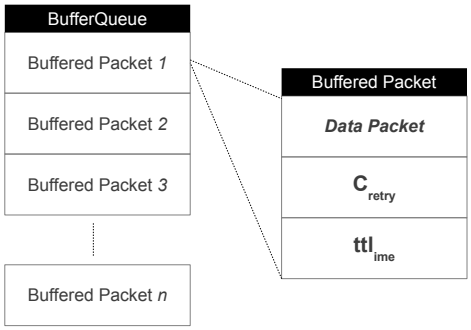


Figure 4.1: BufferQueue structure.

reactive and proactive routing. Discussion on reactive and proactive routing was presented in Chapter 2. By analysing behaviour of end-to-end routing protocols (e.g., AODV, OLSR), it can be seen that there are similarities in the way they detect and handle link failures. For example, they detect link failures either by the loss of periodic *Hello* messages or the mechanism called *Link-layer feedbacks*. If these protocols do not find any alternative routes to the destination in case of link failures then the packets (waiting to reach those destinations) are dropped. By extending traditional end-to-end routing protocols to support hybrid forwarding these packets can be prevented from being dropped. All the participating nodes can store these packet(s) in a queue (named **BufferQueue**), if required, to perform hybrid forwarding. When a packet is stored in the BufferQueue it has attributes as follows: (i) number of times this packet can be sent opportunistically i.e. C_{retry} , and (ii) life of the packet in the network i.e. ttl_{time}.

This hybrid protocol has features such as packet drop, meeting new neighbours and detecting a route to perform dynamic switching between routing modes. The detailed description of these features is presented in the following subsections.

4.2.1 Packet drop

When a packet is to be dropped due to the lack of route to the destination, the router first checks whether there is any one-hop neighbour. As shown in Fig. 4.2(a), if there are one-hop neighbours, a copy of the that packet will be sent to each neighbour and the C_{retry} is decreased. The use of C_{retry} is to provide a controlled flooding scheme, which limits the number of packets transmitted in the network (that can minimise the overhead). At the end, the packet is stored in the queue with the remaining C_{retry} and ttl_{time} (i.e., time-to-live in time unit for the packet in the queue). In case of no neighbours packet can be stored with its C_{retry} and ttl_{time}. The router then continues with the normal routing operations.

4.2.2 Meeting new neighbour

Another event that triggers the delivery of the buffered packets, as shown in Fig. 4.2(b), is when a node/router detects a new neighbour. The proposed hybrid protocol considers this event as an opportunity to deliver buffered packets. Hence, when a router detects a new one-hop neighbour, it checks whether there is any packet in the queue. If the queue is empty, then the router continues with the normal routing operation. Otherwise for each packet in the queue, the router first checks whether an end-to-end route exists or not. In this hybrid protocol end-to-end routes are always preferred, if they exist, because they have higher chances of a successful delivery. When a packet is sent to the destination via an end-to-end route (or to the one-hop neighbour if it is the destination), then this packet is removed from the queue. In case there are no end-to-end route, the packet is sent to the new neighbour and the associated C_{retry} is decreased. The process continues for every packet in the queue.

4.2.3 Detecting end-to-end route

An end-to-end route may be created as a result of a node more than one-hop away creating the routing path to a given destination. Therefore, the event that triggers it is when the router detects a route (as shown in Fig.4.2(c)) and tries to send buffered packets using that route. It is similar to the processes described in Fig.4.2(b) except that if there is no route for a packet, then the next packet is processed. Hence in case a packet destined for that route exists in the BufferQueue then the packet will be sent via that route and at the same time removed from the BufferQueue.

4.3 Algorithms

Previous section discussed the design principles of the hybrid protocols that can be applied to any reactive or proactive routing protocols to develop their hybrid extension. One of the most commonly used protocols are: the reactive protocol AODV and the proactive protocol OLSR. In this thesis hybrid extensions of AODV and OLSR are proposed and named as AODV-OPP and OLSR-OPP, respectively. This section describes algorithms required to design both of the above mentioned hybrid protocols.

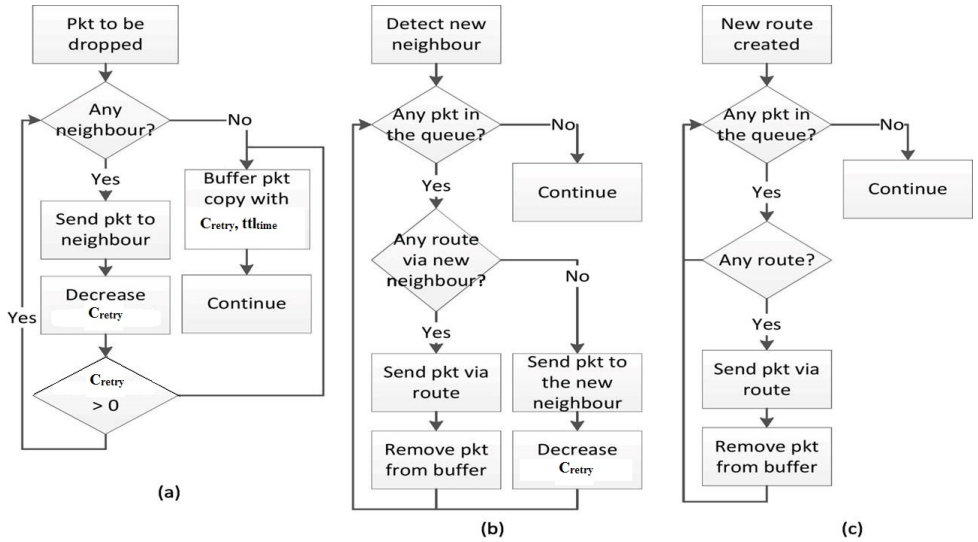


Figure 4.2: Processes for handling packet drops: (a) Packet drop, (b) Meeting new neighbour and (c) Detect route

4.3.1 AODV-OPP

As discussed in Chapter 2, AODV is an on-demand routing protocol for MANETs. In short, if there is any packet for sending, AODV first checks for an existing route. In case of no existing route being present, AODV initiates route discovery for the desired destination. When route is available AODV transmits the packet towards that route. If a link is broken then the protocol attempts to find alternate routes by means of local repair, if feasible, or re-send route request until permissible trials not finished or route request exceeds its time limit. At the end when no route is found AODV drops packets. On the basis of such functionality there are four crucial components required to design its hybrid version i.e. the AODV-OPP protocol, as described in the following subsections.

4.3.1.1 Detecting route and link failures

AODV supports two ways to detect the failure of a route or a link. The most common approach is by periodic exchange of *Hello* messages between neighbours. Upon receiving these heartbeat messages, a node refreshes the time-to-live timer of the respective neighbours. Any neighbour that does not refresh its timer is removed from the node's neighbour list. Subsequently, an AODV RERR message is propagated to all the nodes along the affected routing path. Any route that relies on the removed node is invalidated. As a result, packets that depend on these routes are dropped. At this point, the method of Fig. 4.2(a) is applied to handle the packets rather than dropping them in AODV-OPP.

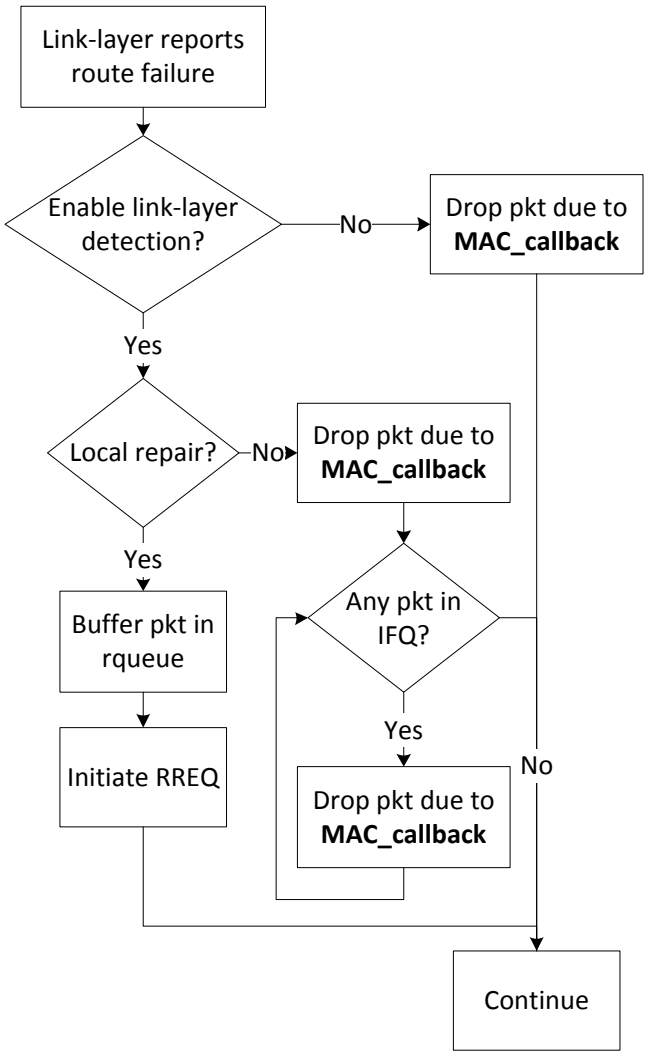


Figure 4.3: The link-layer detection in AODV.

Link-layer detection is another approach to detect link failure. As shown in Fig. 4.3, for each link a node has with its neighbours, the node registers a callback function. When the link-layer reports route failure, it tries to perform the *local repair* mechanism, which is possible only in case if the link-layer detection feature is enabled, otherwise, the packet is dropped. Similarly, if local repair is not supported then all packets within the interface priority queue (IFQ) are dropped. During the local repair, the packet is buffered in the *rqueue* (in AODV rqueue or repair queue temporarily stores packet(s) waiting to repair its route), and a route discovery is initiated.

When a packet is dropped due to the *MAC_callback*, the same method as shown in the previous approach (Fig. 4.2(a)), is applied.

Algorithm 1 Purge packet form BufferQueue

```
if BufferQueue.contains(Packets p) then
  for p in Packets do
    t = p->ttl_time;
    if t < CURRENT_TIME then
      Remove packet p from BufferQueue;
    end if
  end for
end if
```

Algorithm 2 Route probing for destinations

```
if BufferQueue.contains(Packets p) then
  for p in Packets do
    Destination d = p.getDst();
    if !RREQList.contain(d) then
      sendRREQ(p, d);
      RREQList.add(d);
    end if
  end for
end if
```

4.3.1.2 Buffering dropped packets

As shown in Fig. 4.2(a), the C_{retry} of a dropped packet will decrease by one after sending it to a neighbour. When this packet has been sent to all its neighbours and its C_{retry} is still left greater than zero (i.e., this dropped packet is still allowed to be sent), this packet is put into the *BufferQueue* in order to be delivered when a new neighbour is detected (as shown in Fig.4.2(b)). In addition to storing the packet ID and other forwarding information, each packet in the BufferQueue also keeps information about the C_{retry} and the ttl_{time} (time-to-live). Packets with expired ttl_{time} are purged from the BufferQueue.

4.3.1.3 Engaging route discovery

Since AODV is a reactive routing protocol, it does not initiate route discovery unless there is a packet to be sent from a source node (or a packet being buffered in the intermediate node due to no route). As shown in Fig. 4.2(b), when a new neighbour is detected, an end-to-end route that could be created via the new neighbour is preferred because it provides better guarantee of delivery compared to hop-by-hop delivery. Therefore, an approach is needed to initiate the route discovery process when a new neighbour is detected.

In this approach, the first packet of a particular destination in the BufferQueue is used to

Algorithm 3 Packet history record : Node n receives a data packet p

```
if PktID_record.contains(pkt ID of P) then
   IGNORE p;
else
   Insert packet ID in PktID_record;
   Process packet p at node n;
end if
```

trigger the process of sending a RREQ message for that destination (Algorithm 2 shows the pseudo-code of this process). After a RREQ for the packet's destination is sent, this packet along with other packets for the same destination are then sent to the new neighbour. Similarly for every packet in BufferQueue router raise RREQ (if not yet raised) to discover its route and subsequently sent packet(s), for the same destination, to the neibhbour node. Packets travel in a hop-by-hop fashion until an end-to-end route is found. It maximizes the number of packets to be sent to nodes that are closer to the destination, rather than waiting for the RREQ timeout of 10s by default. In addition, since these packets are already in the neighbouring node, they get a higher priority to be sent should a route be created. If an end-to-end route is possible, then the remaining packets in the BufferQueue are sent using the route as shown in Fig. 4.2(b).

In large-scale networks, there could be many traffic flows sending packets to the same destination. When this happens, there is a significant number of RREQ messages created for the same route. To reduce the overhead, each node keeps a record of the destinations for which RREQ has already been raised in the network so that multiple requests for the same destination node can be ignored.

4.3.1.4 Avoiding routing loop

Sequence numbers are used to avoid routing loops in AODV. As AODV-OPP incorporates the store-and-forward feature of opportunistic protocols, I introduced another component — *PacketHistory*, to keep a trace of all packet IDs that a node has seen over the last time window (Algorithm 3 shows the pseudo-code of this process). This way, a node can discard a packet if it has already processed that packet within a specified time limit. *PacketHistory* can be large when packet's IDs are saved for a long time or when network is congested. Therefore to control the size of *PacketHistory* one of the approaches can be to keep the time window smaller so that node(s) would not keep packet(s) record for a long time.

4.3.2 OLSR-OPP

A generic concept of the hybrid protocol is presented in Section 4.2 that can be applied to both reactive and proactive routing protocols. This section discusses the design of OLSR-OPP, a hybrid extension of the proactive routing protocol and also compares it with AODV-OPP.

4.3.2.1 Detecting route and link failures

In OLSR, the status of a link is managed very carefully. A link is considered to be bi-directional. When a node receives the HELLO message from a neighbour, it creates an entry in the neighbour information set (*nb_tuple*) to indicate its presence. Exchange of packets occurs after these two nodes create an entry in the link information set (*link_tuple*). Once the *link_tuple* is created it can either be *asymmetric* or *symmetric*. An asymmetric link is created when HELLO messages are received from one direction, and if both nodes exchange HELLO messages then that link_tuple becomes symmetric. OLSR can send data packets only to a symmetric link.

OLSR periodically broadcasts the HELLO message to update the link information. In case of any link failures it drops packets if it does not find any alternate routes. In this situation OLSR-OPP triggers the event shown in 4.2(a).

4.3.3 Buffering dropped packets

As shown in 4.2(a), the OLSR-OPP protocol also follows the similar procedure for packet drop as in case of AODV-OPP. This procedure is triggered in case of a broken link. In addition to the procedure OLSR-OPP also has the similar BufferQueue structure for storing the packets (if required) with meta-information, such as C_{retry} and ttl_{time}. C_{retry} is used to limit the number of times a packet can be sent opportunistically. In other words it can limit the overhead generated in the network. For a buffered packet ttl_{time} is time to live in the network. To remove the stale packets from the BufferQueue, a timer based event periodically monitors the ttl_{time} of each packet and removes the expired or stale packets from the BufferQueue in a timely fashion.

4.3.4 Meeting new neighbour/detecting new route

Every time an OLSR node receives a control packet, the routing table is recomputed after processing the non-duplicate message. A control packet could be
(i) A *HELLO message* about the local neighbour information,
(ii) A *TC message* that updates the topology information set, or
(iii) A *MID (Multiple Interface Declaration) message* that advertises the information about the node's interface association.
It can be observed that whenever there is a change of a neighbour or a link, every node in the network gets an update of the routing table. This update can be very frequent. Because of these characteristics in OLSR, the design of OLSR-OPP is slightly different from the design of AODV-OPP. For example, in OLSR-OPP, the process, as shown in Fig. 4.2(b) is triggered by an update on the *link_tuple*, rather than changes of the *nb_tuple*. Hence, if any link, direct or indirect, changes in the network, each node in the network, updates its routing table. Due to such functionality of OLSR, OLSR-OPP considers those link(s) changes that directly affect the node i.e. one hop neighbour nodes. Therefore, the process for handling new routing entry updates is not necessary in OLSR-OPP. This is due to two reasons: (i) routing table update is very frequent and majority of the new routes found are not for the buffered packets inside the BufferQueue, and (ii) neighbour set updates and recomputation of routing table are triggered by the received HELLO messages, but updating the neighbour set is always done first.

4.3.5 Avoiding routing loops

Due to the hybrid nature of the routing protocol packets may loop in the network. To prevent such situations, OLSR-OPP also keeps record of packet-IDs seen in past (Algorithm 3 shows the pseudo-code of this process).

4.4 Summary

This chapter discussed the design of a hybrid protocol that can be applied to both classes of reactive and proactive routing protocols in wireless mesh networks. It also presented a discussion on transport layer protocols and types of traffic which can be used in such a hybrid protocol. It also provided a detailed description of an extensions of both AODV to a reactive

hybrid protocol and OLSR to a proactive hybrid protocol. The next chapter discusses the systematic approach to evaluate both hybrid protocols presented in this chapter.

CHAPTER 5

Evaluation

The previous chapter described the design principles of the hybrid protocol that can be applied to both reactive and proactive routing protocols. This chapter presents the systematic evaluation of the proposed hybrid protocols. These hybrid protocols are developed in the NS2 simulation environment.

A simulation environment provides flexibility as compared to the real test-bed. For example, it is easier to build a complex network scenario from a basic building blocks, like, a variety of nodes, links, and protocols to perform a test. A wide range of networking technologies can be simulated without using any expensive resources. In addition, a fine-grained control of nodes' movement is also difficult in real test-beds due to the placement of dedicated machines and limited accessible area. A simulation environment can easily model a large scale topology where it is possible to control parameters of the network or nodes. For example, nodes' movement, their transmission range, or interference among nodes etc.

This chapter begins with a brief description of the simulation environment. After that in the context of systematic evaluation of the proposed hybrid protocols a detailed description of the methodology used is provided. At the end of this chapter performance of the protocols is discussed.

5.1 NS2 simulation environment

NS2 is the discrete time event simulator designed for networking research. It provides significant support for simulating various routing protocols (i.e. AODV, DSR, DSDV, TCP and other protocols for wired and wireless networks). This simulator is widely accepted by other researchers working in networking fields. It is an open source design and compatible

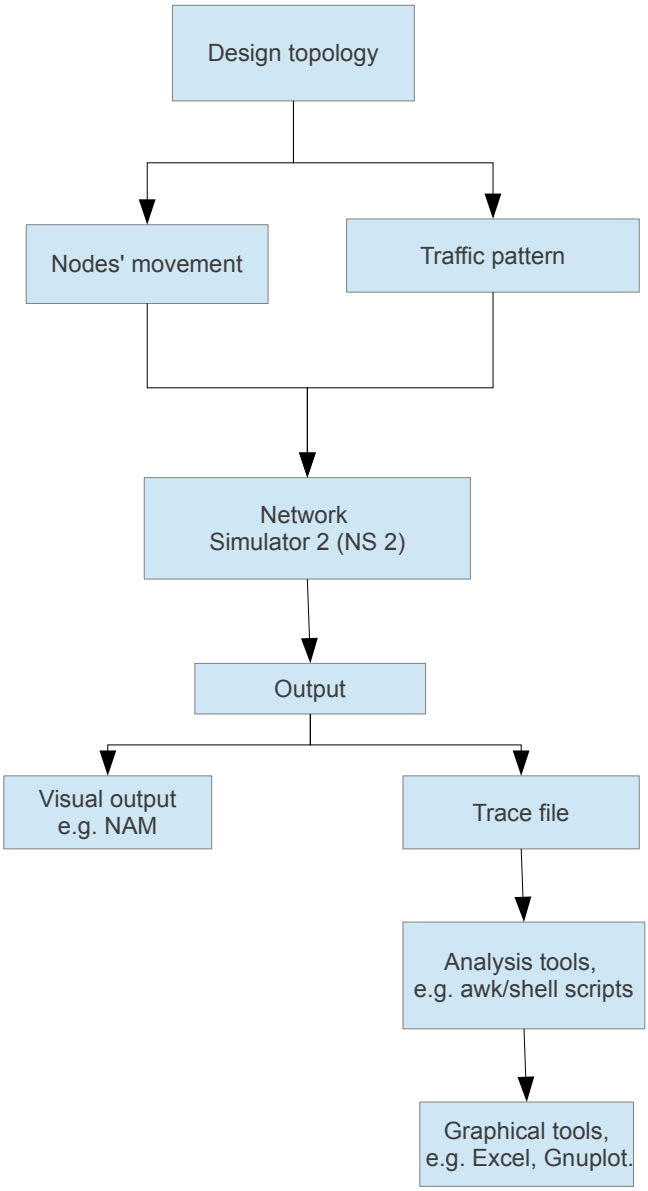


Figure 5.1: Overview of NS2 simulation.

with the Unix based Ubuntu operating system. A developed AODV-OPP version, which is an extension of the AODV protocol, is built on the CMU version in NS 2.34, whereas the OLSR-OPP is an extension of the UM-OLSR protocol [75].

NS2 is an object oriented simulator, written in C++. The Object Tool command language (OTcl) is used as a front end. In NS2 experiment results are recorded in a trace file. In addition to the trace file, users can also view the visual output.

High level overview of NS2 simulation environment is shown in Fig. 5.1. To simulate a network scenario NS2 requires two input files:
(i) Nodes' movement file that describes position of the nodes in terms of {x,y} coordinates with time stamp;
(ii) Traffic pattern that describes the flow of traffic among nodes.

Both of these files are applied to an oTcl code that can be executed in NS-2 environment. It generates two types of output files: Trace file and Network Animator (NAM) file. The NAM file has a graphical format to visualize the simulation experiment. The trace file has detailed descriptions of the events that occur during the simulation experiments. Before beginning the simulation, the parameters and their measurement techniques need to be selected. Scripts can analyse the trace files and compute parameters as per the applied technique. These scripts can be developed using shell, AWK or Python scripts. At the end, to statistically present output of the experiment various graphical tools like Excel, Matlab or gnuplot can be used.

5.2 Parameters to evaluate protocol performance

This section discusses all the parameters and their measurement techniques required to evaluate the performance of the protocol.

5.2.1 Packet delivery ratio (PDR)

PDR is defined as the ratio of number of packets received at the destination node and number of packets sent from the source node. Therefore, PDR (%) is calculated as

$$PDR(\%) = \frac{N_{received}}{N_{sent}} * 100(\%) \tag{5.2.1}$$

Where N_{sent} is the number of packets sent by the source node; $N_{received}$ is the number of packets received at the destination node without considering the received copies of the packet.

5.2.2 Normalised Routing Load (NRL)

NRL is defined as the number of control packets (such as RREQs, RREPs, HELLOs, and RERRs or TC updates etc) required for successful receipt of the number of data packets at the destination node.

$$NRL(\%) = \frac{N_{received}}{N_{contol\_pkts}} * 100(\%) \tag{5.2.2}$$

Where $N_{control\_pkts}$ is the number of control packets forwarded in the network; $N_{received}$ is the number of packets received at the destination node without considering the received copies of the packet.

5.2.3 Overhead

Overhead O is defined as the number of additional packets forwarded in the network for every packet successfully delivered to the destination. Therefore, it is calculated as

$$O = \frac{N_{forwarded}}{N_{received}} \tag{5.2.3}$$

Where $N_{forwarded}$ is the number of additional copies of the buffered packets forwarded in the network; $N_{received}$ is the number of buffered packets received at the destination without considering the received copies of the packet.

5.2.4 Average end-to-end packet delay

Average end-to-end packet delay is defined as the sum of delays each packet experiences when successfully received at the destination node from the source node. To measure the delay experienced by each packet the time difference is computed between when a packet is received at the destination node and when the packet is first sent from the source node.

To get the average end-to-end delay all the values are added and divide by the number of packets received at the destination (excluding the copies of the packet(s)).

5.3 Methodology

This section presents the methodology used to evaluate the performance of the hybrid protocols and to demonstrate its performance as compared to the existing protocols. The evaluation is carried out in three phases as listed below.

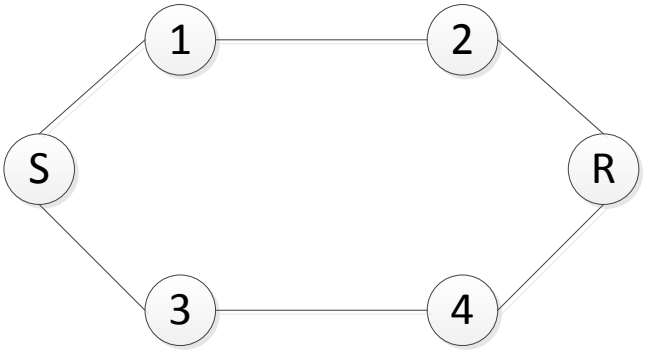


Figure 5.2: Validation tests topologies

The first phase is to validate the developed simulation model. For this set of experiments a 6 node topology is designed and 4 different scenarios are created by moving the nodes that can demonstrate the functionality of the protocol in four different network situations to verify the correctness of the developed simulation model.

The second phase is to evaluate the performance of the protocol in varying network characteristics using synthetically generated mobility scenarios.

The third phase is to verify the performance of the protocol using a real time trace.

5.3.1 Validation test

As discussed above, these tests are aimed to verify the basic operations of the hybrid protocols. The basis of these tests is to check that, if a route exists in the network, then the hybrid protocol behaves similar to the end-to-end routing protocols i.e. it does not introduce any additional overhead.

For these validation tests, six nodes are positioned in a diamond topology as shown in Fig 5.2. Each node is connected to two other nodes in a sequence. Among these nodes, S is the source node and R is the destination node. In this topology two routes are possible i.e. S-1-2-R and S-3-4-R. Subsequent subsections describe four different scenarios possible in this topology and their significance to evaluating the correctness of the developed simulation model.

5.3.1.1 Case 1: All nodes are static

In this scenario all the nodes are static and a route exists from the source node to the destination node. These are baseline tests, which aim to verify that AODV-OPP and OLSR-OPP

do not introduce any unnecessary overheads when normal AODV and OLSR operations should be used for packet delivery.

5.3.1.2 Case 2: Link breaks causing rerouting

This scenario, first verifies that the traffic from the sender (Node S) to the receiver (Node R) traverses via the route S-3-4-R. Node 4 then moves out of the range, which causes AODV and OLSR to reroute the traffic through alternative path S-1-2-R. Except Node 4 other nodes are static. Therefore, AODV should be able to repair the route quickly via S-1-2-R (as AODV requires at least 0.01 seconds to discover a route, if discovery is possible) and in case of OLSR it can use an alternate route already present in its route table.

5.3.1.3 Case 3: No alternative route causing packet drop

This scenario emulates the situation when Node 4 moves out of the range from the route S-3-4-R at around 50 s after that node 1 moves out of the range from the route S-1-2-R. This causes AODV to reroute via the alternative route, but the second link break causes packet drops. As a result, the delay-tolerant mechanisms in the AODV-OPP and OLSR-OPP should buffer a copy of these packets for delivery at a later time. The goal of this scenario is to confirm that AODV-OPP and OLSR-OPP are able to correctly buffer all the packets which are being dropped.

5.3.1.4 Case 4: Route can be re-established

This scenario tests AODV's ability to re-establish the route when a node on the routing path moves within range again, and all the buffered packets are delivered successfully through the new route. In this scenario, Node 4 moves back to its original position after 60 seconds. Due to Case 3, there are packets buffered at Nodes S and 3 because broken link causes the routing mode to change to opportunistic routing. Hence packets are sent from node 4 to the neighbour node Node 3, and from Node 3 packets are sent to Node S as shown in Fig 4.2(a).. A new route established through S-3-4-R will cause packets to be delivered via this route, as shown in Fig 4.2(b).

In case of OLSR-OPP when node 4 moves back to its original position, nodes update their link information and re-compute route tables at the nodes. When nodes update their routes

after that all the one hop nodes can be added as the neighbour nodes. Hence the event, shown in Fig. 4.2(b), is triggered, and all the buffered packets can be delivered via new neighbour to their destination i.e. 3-4-R.

5.3.2 Mobility model

For systematic evaluation of the performance of the protocol there is a need to have mobility scenarios that can represent the different network situations. These mobility scenarios can be captured from a real life situation or can be generated synthetically. Various ways exist to gather real life mobility traces [49]. For example the SanFrancisco cab trace is one of the real mobility traces which is gathered using a GPS device. To systematically evaluate the protocol$'s$ performance both kinds of traces are used.

There are various ways to generate synthetic traces. One of the synthetic ways is to use NS 2$'$s utility called **setdest** that allows to generate different kinds of mobility patterns such as random way point, random walk etc. The biggest drawback of setdest is that the resultant mobility scenarios are randomly generated and it is difficult to determine the behaviour of the protocol in a particular situation. Therefore Bonn motion mobility generator is used to generate synthetic trace. The subsequent section gives detailed description of Bonn motion tool and its generated traces.

5.3.2.1 Bonn motion

This section gives a brief introduction of the Bonn motion and a detailed description of the traces generated using this tool.

Introduction

BonnMotion [1] is a Java software **mobility scenario generation and analysis tool**. It was developed at the Institute of Computer Science IV of the University of Bonn, Germany in the Communication Systems group. The purpose of this tool is to investigate the characteristics of the network. The generated scenarios can also be exported for various simulators such as NS2, GloMoSim/QualNet, COOJA, MIXIM and ONE.

To generate a mobility pattern this tool functions in three steps. The first step is to generate a scenario by providing inputs such as (i) number of nodes in the network, (ii) required mobility models such as the Random Waypoint model, Random Walk model, Gauss-

Markov model, Manhattan Grid model, Reference Point Group Mobility model, Disaster Area model, Random Street model etc. (iii) duration of the scenario, in other words simulation time and (iv) cutoff period, which specifies the duration that should not be taken into account at the beginning of the generated scenario.

As a result of the first step a mobility file is created as per the specified parameters: number of nodes, type of mobility model, duration of the mobility scenario and cutoff periods.

The second step is to generate statistics of the generated mobility file from step 1. This requires parameters such as the mobility file created using Bonn motion and transmission range of the node. The result of this step is to create another file that contains statistics of the specified mobility file. Some of statistics provided by this application are as follows:
Average node degree i.e. how many other nodes is one node connected to?
Average number of partitions: This is an integer number where a value of 1 means the network is connected at all times. Any other value for this number indicates the opposite.
Partitioning degree/Degree of separation: How likely is it that two randomly chosen nodes are within a connected component at a randomly chosen point in time?
Average link duration: Only links that go up after the simulation start and go down before the simulation end are taken into account etc.

To use these mobility scenarios, the third step is to make the scenario compatible with the selected simulation environment using various applications. For example, the **NSFile** applications generate mobility patterns compatible with the NS2 simulation environment. The output of this application generates two separate files, one file specifies the parameters required to use the generated mobility file in NS2 such as area size, duration of mobility, and the second file, called the compatible mobility file. This tool generates mobility patterns with detailed statistical data that makes it easier to understand the behaviour of a protocol.

Synthetic trace

To evaluate the protocol performance Bonn motion tool's generated synthetic traces are used. Generated traces vary in their partitioning degree (PD) which represents the varying node density in the network i.e. dense to sparse. Its value varies from 0 to 1, where 0 represents the connected/dense network and 1 represents the sparse network. To evaluate the protocol in different network situations PD range from 0 to 1 is divided into three equal ranges (PD low: 0-0.33; PD medium: 0.34-0.66; PD high: 0.67-1).

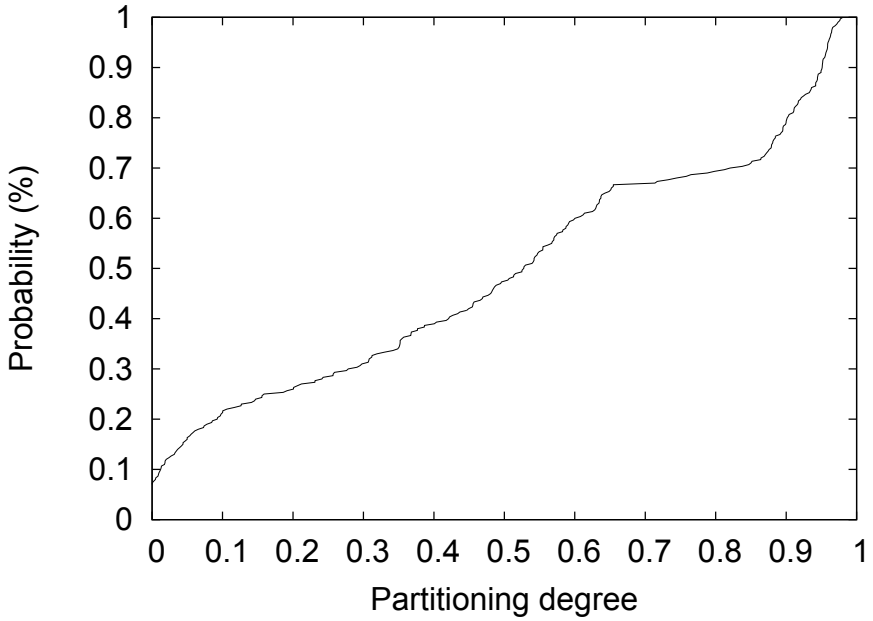


Figure 5.3: Scenario distribution.

To achieve statistical confidence in results, for each partitioning degree 100 different scenarios are generated. A total of 300 scenarios are generated for the entire PD range. To generate these 300 scenarios, first 2000 random scenarios with different area sizes are generated using Bonn motion. Then randomly 100 scenarios are selected for each partitioning degree range.

Cumulative distributive function (CDF) plot of PD values of selected 300 scenarios is shown in Fig. 5.3. It can be seen that these 300 scenarios are uniformly distributed across the whole range of partitioning degree values. It can be argued that this set of randomly generated scenarios should be representative for most of the application scenarios (including corner cases). It should be noted that fewer samples between PD value of 0.65-0.85 exist. This means there are not as much scenarios for this PD range as the other ranges. However, the whole point of systematic evaluation is to investigate the performance of each protocol using randomly selected scenarios. Therefore, generated set of scenarios are not artificially changed for the evaluation. By evaluating proposed protocols against these randomly selected scenarios, it is possible to analyse how the protocol performs under different characteristics of the network and the evaluation results should be comprehensive.

5.3.2.2 Real-life trace

The tool Bonn Motion used to generate synthetic traces that conform to a particular characteristic of the scenario (e.g., different densities or node connectivities), allows carrying out systematic tests for the proposed hybrid protocol. In addition to the synthetic tests, I have

used GPS based San Francisco taxi cabs mobility traces [2] gathered in real-life. This trace consists of GPS coordinates of 500 taxis over the period of 30 days. For the purpose of protocol evaluation, traces of 116 cabs are selected from the downtown area of size $5700x6600mtr^2$ over the period of 3600 seconds. I use this trace because it has high resolution of node positions when nodes were frequently moving. Hence, it serves as a means to validate observations from previous tests and to demonstrate how hybrid protocols perform in real time scenarios.

5.4 Results and discussion

This section discusses the results of proposed hybrid protocols in different network scenarios as described in Section 5.3 and they are as follows, (i) Validation test result, (ii) Performance of AODV and OLSR in 300 selected scenarios (varying in PD value), to demonstrate the performance of end-to-end routing protocols in these scenarios. The basis of this experiment is to create a base line when analysing the behaviour of their hybrid extensions. (iii) Performance of both hybrid protocols i.e. AODV-OPP and OLSR-OPP in 300 selected scenarios, (iv) Performance of the protocols in a real life trace.

5.4.1 Validation test result

This section presents the results of validation test using the scenarios discussed in Section 5.3.1. For this set of experiments simulation parameters are set as listed in Table 5.1.

5.4.1.1 Result of AODV-OPP's validation test

Fig. 5.4 shows the averaged results of all the simulation tests. In addition, Table 5.2 details (in one test run) the number of buffered packets (being dropped due to no route) that are *buffered* in the BufferQueue, *received* at the destination, and *lost* due to various reasons (e.g., IFQ being full). As expected, AODV-OPP achieves exactly the same PDR as the original AODV and no packets have been buffered when nodes are static (as in Case 1). This means AODV-OPP does not generate additional overhead when in the same situation AODV is capable of handling the traffic.

In the second case, when node 4 moves out of range, AODV will reroute the traffic flow from

Table 5.1: Simulation parameters for validation test

Hybrid protocol's parameters	C_{retry} Packet's TTL	10 copies 300 s
Traffic parameters	Traffic Type Traffic start time Traffic end time Packet size Data rate	CBR 20 s 120 s 512 bytes 4 packets per second
Network parameters	Simulation Time IFQ length Transmission Range Propagation Model 802.11 MAC Tx Rate RTS/CTS Queue Type Simulation Area	300 s 50 pkts 250 m Two Ray Ground 11 Mbps Enabled Drop Tail 1500 x 1500 mtr^2

Table 5.2: Analysis of dropped packets

	Buffered	Received	Lost
Case 1	0	0	0
Case 2	1	0	1
Case 3	73	0	73
Case 4	73	49	25

S-3-4-R to S-1-2-R. It is observed that most of the traffic goes through this alternative route. However it is also noticed that one packet was at node 4 when the node moved out of range. As a result, this packet is dropped and buffered on node 4 and lost due to TTL expiry.

In Case 3, when nodes 1 and 4 move away, there are no routes to the destination R. It is observed that PDR is almost the same for both protocols i.e. AODV and AODV-OPP, because neither any alternative route exists nor any other opportunity to deliver buffered packets. In AODV-OPP, there are 73 packets that have been buffered in the BufferQueue in nodes S and 3.

In Case 4, AODV achieves the same PDR as in Case 3. This is because the simulation traffic was stopped (at 120 s as shown in Table 5.1) when nodes 1 and 4 were out of range, and there is no new packet generated when node 4 moves back to its original position. In addition, it can be noticed that a significant PDR gain of around 12% is achieved by AODV-OPP (as shown in Fig. 5.4). This demonstrates the store-and-forward mechanism works well in AODV-OPP. Although 49 packets arrived at the destination among the 73 packets that were dropped and buffered, as shown in Table 5.2, 25 packets were lost. After investigation, it was uncovered that there were 24 packets lost due to IFQ being full and one packet lost

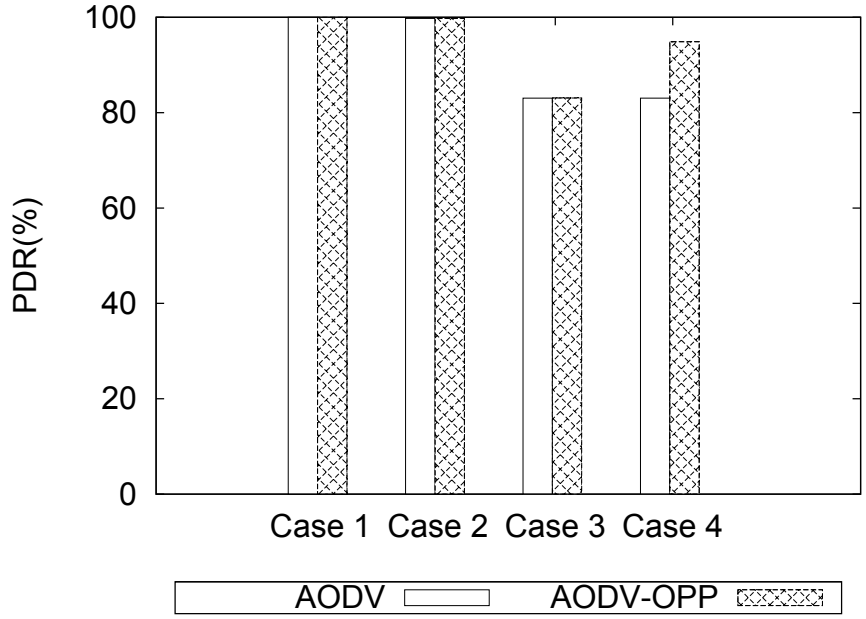


Figure 5.4: Performance comparison of the validation tests.

due to expired TTL. The reason for the loss due to IFQ is because of our small IFQ length (allowing only 50 packets). A larger IFQ can reduce the packet loss for this cause, but a very large IFQ is unrealistic. It can be argued that for a protocol to be practical, it should work under the practical settings.

The performance of the AODV and AODV-OPP over 100 runs for all four cases is shown in Fig. 5.4. As illustrated in this figure, for Case 1 (static nodes) and Case 2 (re-routing) PDR (100%) is similar for both AODV and AODV-OPP. This is because in both scenarios an end-to-end route is possible. Similarly, in Case 3 both protocols have similar PDR (80%) because in this case node 1 and node 4 are out of the transmission range. AODV can not discover a route. Also, AODV-OPP can buffer packets but does not send those packets opportunistically. In Case 4, AODV-OPP is able to show higher PDR (i.e. 92%) as compared to AODV (i.e. 80%) because in this scenario AODV-OPP can discover a route when node comes back to its original position. As a result all the buffered packets can be delivered to the destination node.

One more observation is that in Case 4 AODVOPP can get even higher PDR. As in this implementation packets that are dropped due to broken link are buffered in the BufferQueue, whereas there are some packets still waiting inside the "rqueue" (repair queue of AODV) for their route to get repaired as discussed in Section 2.1.1.2. Hence if no route is generated then these packets remain in the rqueue until packets' life (TTL) is not expires.

Another observation in case 4 is that when node 4 moves back, node 5 and node 3 add this

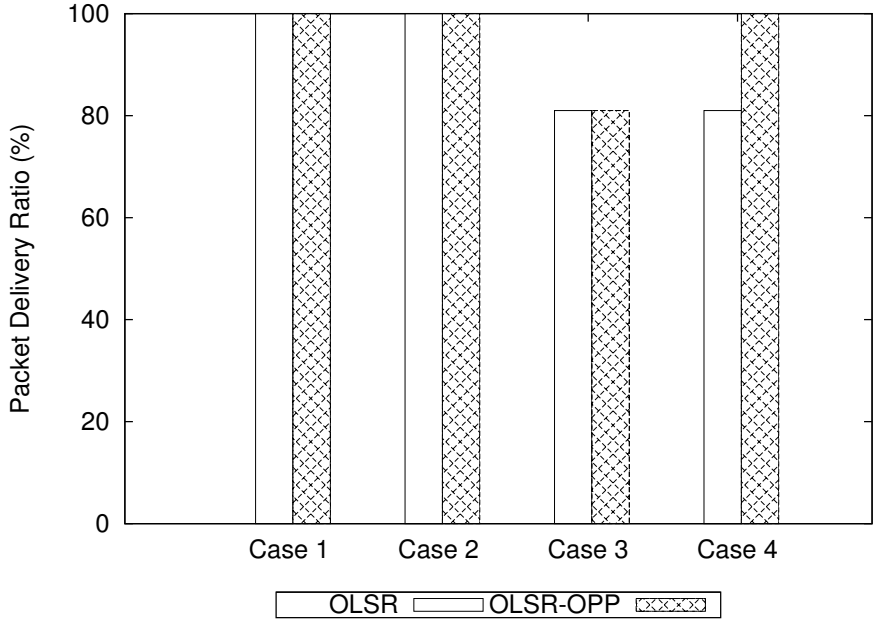


Figure 5.5: Performance comparison of the validation tests.

node as their neighbour node. As mentioned earlier nodes S and 3 have packets in their BufferQueue. Node 3 raise RREQ and when a route is generated via node 4 packets from nodes 3 are delivered to the destination. Buffered packets at node 1 are purged due to their TTL expiring and no route generated via node 1 to R because the RREQ expired before node 4 moved back.

It can be concluded based on the above that due to the hybrid feature of AODV-OPP it can efficiently make use of the network situation to deliver packets to the destination.

5.4.1.2 Result of OLSR-OPP's validation test

Fig. 5.5 shows the average result of 100 runs for all four network scenarios. Before collecting data for the validation tests it was ensured that initially data is transmitted via S-3-4-R route in diamond topology.

In case 1, nodes are static and end-to-end route is possible between the S-R pair. Packets can be transmitted via a route. This means that OLSR and OLSR-OPP perform similarly in terms of PDR and OLSR-OPP does not generate any additional overhead in the network.

In case 2, when Node 4 moves out of the transmission range,the link between node 3 and node R failed. In this case OLSR and OLSR-OPP sent packets through the alternate route i.e. S-1-2-R. That confirms that this protocol can re-route packets if possible. Hence, OLSR-OPP is able to perform similar to OLSR when an end-to-end route exists.

In case 3 no route is possible to the destination because node 4 and node 1 move out of the transmission range from their routes; this scenario verifies that when OLSR decides to drop packets, OLSR-OPP is capable of successfully buffering those packets. As shown in Fig. 5.5 for case 3 PDR is the same for both protocols. Although OLSR-OPP has buffered packets it does not have any alternative way to deliver packets to the destination node.

In case 4, after 50s, node 4 moves out of the range from the route S-3-4-R. After another 50s node 1 moves out of the range from the route S-1-2-R and as a result all the packets are buffered at node 1 and node 3. Later on when node 4 moves back to its original position at 110s routes are computed at all the nodes. When node 3 and node 4 update their neighbour table then nodes deliver packets from node 3 via route 3-4-R. Hence OLSR-OPP is able to transmit buffered packets. In a similar situation, OLSR cannot use this opportunity because packets are already dropped and traffic stops at 120s so there is no new packet to send. As a result OLSR-OPP is able to outperform OLSR as illustrated in the Fig. 5.5. For case 4 OLSR-OPP gives 18% PDR gain over OLSR.

5.4.2 Synthetic trace results

The validation tests aim to verify whether the operations of AODV-OPP conform to the design specifications. To study the performance in actual networks, there is a need to scale up the simulation with more concurrent data traffic.

The second set of synthetic tests use 50 mobile wireless nodes. Each of these 50 nodes are allowed to form connections with any other node in the network. These connections will be formed randomly at different times during the simulation and other parameters for this experiment are shown in Table 5.3. For each of the 300 scenarios, simulation runs 10 times and average is computed. To create a baseline for comparing performance of the original protocol and its hybrid version, in the first set of experiments using synthetic trace, performance of AODV and OLSR is evaluated and then hybrid extensions i.e. AODV-OPP and OLSR-OPP are evaluated over those scenarios.

5.4.2.1 Performance of AODV and OLSR

In highly mobile networks, AODV could perform better than OLSR, since it does not actively maintain routes for the entire network (i.e., smaller protocol overhead). Due to mobility, most routes might not be valid when they are needed. In addition, in OLSR any change in

Table 5.3: Simulation parameters for Synthetic test

Hybrid protocol's parameters	C_{retry} Packet's TTL	10 copies 500 s
Traffic parameters	Numberof nodes Traffic Type Packet size Data rate	50 CBR 512 bytes 4 packets per second
Network parameters	Simulation Time IFQ length Transmission Range Propagation Model 802.11 MAC Tx Rate RTS/CTS Queue Type	500 s 50 pkts 250 m Two Ray Ground 11 Mbps Enabled Drop Tail

part of the network will cause a global update in every node's routing table. However, OLSR outperforms AODV for some metrics, especially in terms of delay. As OLSR exchanges topology information with all nodes in the network, the route is ready for use whenever a node has packets to send. The responsiveness of OLSR is entirely up to the interval settings of the two link state messages (HELLO and TC). In comparison, AODV requires the time to initiate the route discovery process if the route does not exist. This potentially increases the packet delay.

With regard to handling packet drops, AODV introduces a buffering feature, i.e. rqueue, which can temporarily hold packets, to give a node the time to repair the route by its route discovery process. In contrast, in OLSR, if a packet cannot be sent due to no route to the destination, this packet will be dropped.

The first set of simulations investigates how AODV and OLSR perform in the 300 scenarios with different partitioning degrees. Fig. 5.6 shows the performance of the two protocols, in terms of packet delivery ratio (PDR). In this figure, I have also plotted the fitted curves (labelled as *curve*) of both protocols using second degree polynomial. As I expected, both protocols achieve lower PDR as the partitioning degree increases (i.e., the network becomes sparse where connection among nodes gradually decreases and router invalidates route(s) respectively). As highlighted by the fitted curves and the PDR difference, AODV outperforms OLSR more than 5% in all the scenarios from the low to medium PD ranges. In these scenarios the network is relatively dense, hence to get the updated link information every node sends HELLO and TC updates. Therefore frequent HELLO and TC messages exchanged in OLSR could become the source of interference that prevents nodes from suc-

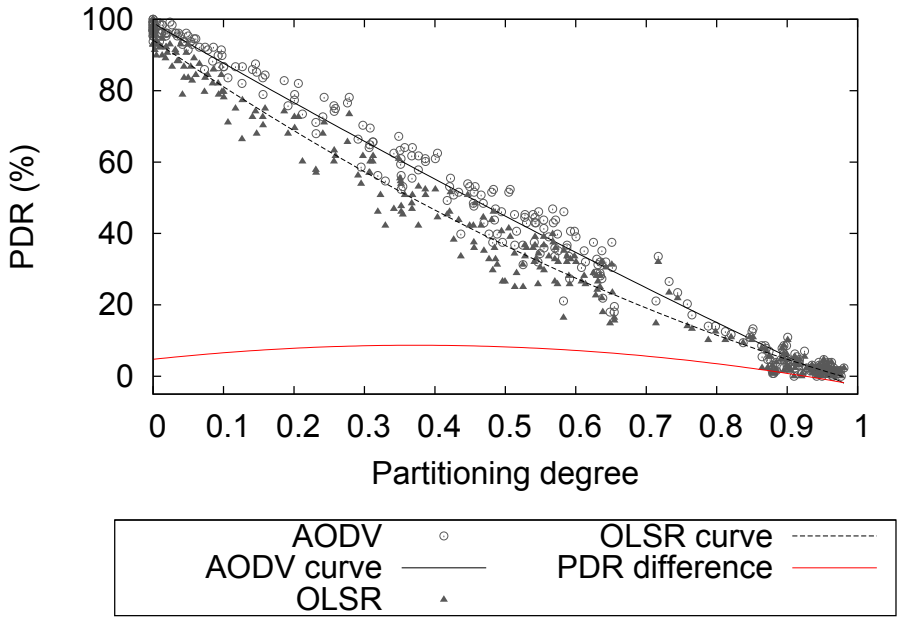


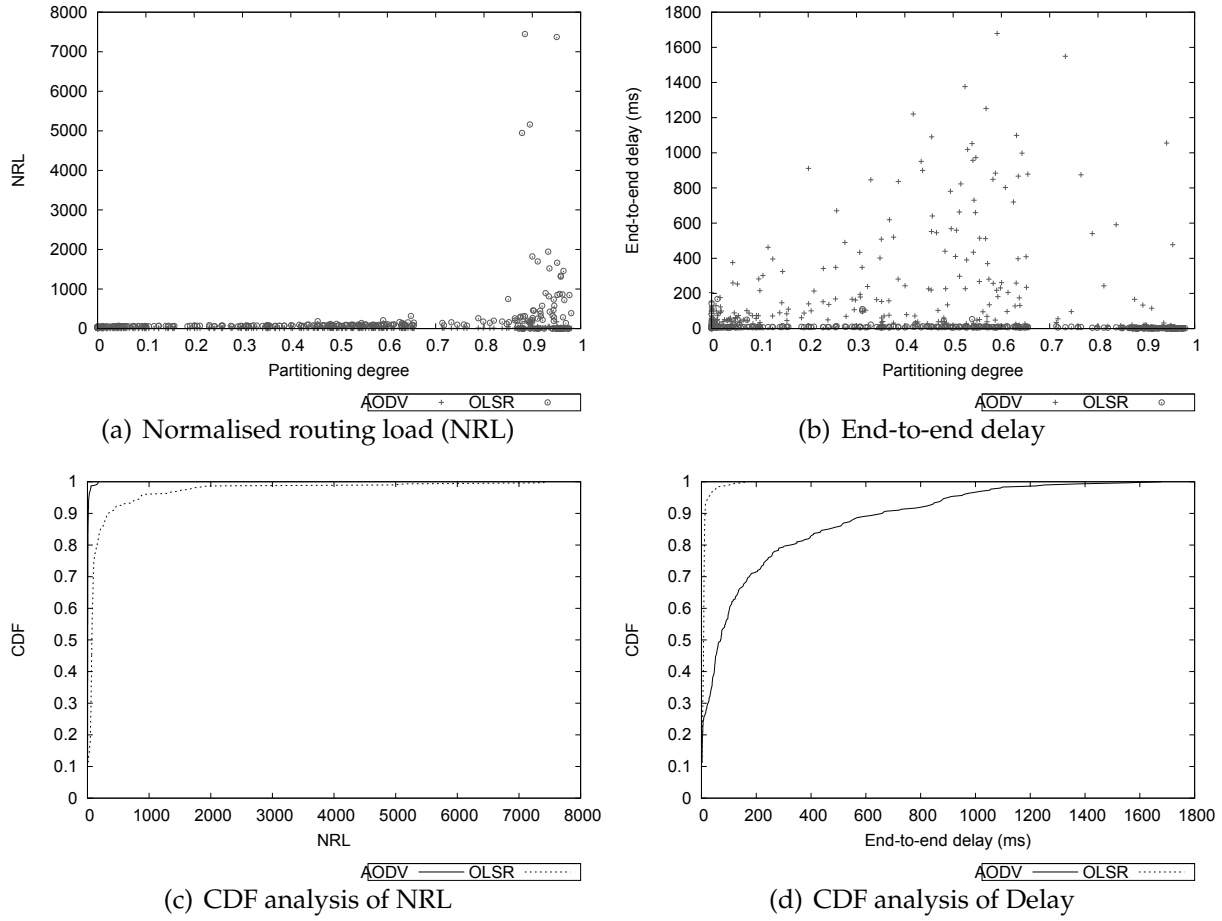
Figure 5.6: Performance of AODV and OLSR.

cessfully sending data packets, resulting in lower PDR than AODV.

I conjecture that the overhead in OLSR is the cause of this PDR difference. The excessive overhead is understandable as OLSR frequently exchanges control information to keep the routing tables of every node in the network up-to-date. Fig. 5.7(a) shows the Normalised Routing Load (NRL) [72] of both protocols for all the scenarios. Another observation is that the maximum PDR difference between the two protocols is achieved in the medium PD range. The reason behind such behaviour is that in the medium PD range connection among nodes are changing frequently as compared to low PD or high PD ranges and to cope with these changes OLSR generates more control packets as compared to AODV as discussed earlier. Hence, AODV achieves maximum PDR gain in that range.

As shown in the figure, OLSR generates significantly higher load of control packets for every successfully delivered data packet. Also, this routing load increases dramatically when the network becomes very sparse, as shown in Fig. 5.7(a). From the Cumulative Distribution Function (CDF) graph, as shown in Fig. 5.7(c), it can be seen that the overhead generated by AODV is almost negligible as compared to OLSR (about 5% of the cases with NRL more than 1000). The saving in overhead in AODV is due to its on-demand route discovery when a node has a packet to send, whereas OLSR needs frequent exchange of control information to maintain the up-to-date neighbour, link and topology information.

In addition to the overhead, Figs. 5.7(b) and (d) show comparison of the packet end-to-end delay of the two protocols and the corresponding CDF analysis. The figures show the



(a) Normalised routing load (NRL)

(b) End-to-end delay

(c) CDF analysis of NRL

(d) CDF analysis of Delay

Figure 5.7: Overhead and delay comparison of AODV and OLSR.

average end-to-end delay of AODV is significantly higher than OLSR. In about 30% of cases the averaged end-to-end delay is more than 200 ms, whereas the delay for OLSR is almost negligible. These results validate the discussion about the two protocols; that is, in OLSR every node always knows how to route a packet if the route exists, but in AODV the node will have to initiate the route discovery processes.

5.4.2.2 Performance of AODV-OPP in 300 PD scenarios

Fig. 5.8 shows all the PDR of AODV and AODV-OPP for the 300 scenarios (in points) and outlines the relationship between the partitioning degree and PDR (in fitted curves). The first observation is that both protocols achieve lower PDR as the partitioning degree increases. This is what is expected as the network becomes sparse as discussed earlier. Another observation is that AODV-OPP outperforms AODV in most cases, and the PDR gains in the medium PD are significantly higher than the other two ranges. This signifies that in medium PD scenarios nodes have partial connections that can be utilised by the hybrid protocol i.e. AODV-OPP.

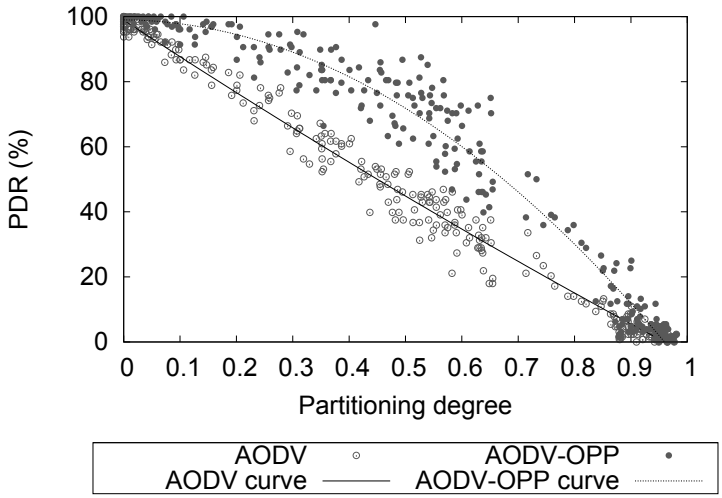


Figure 5.8: PDR of AODV and AODV-OPP for varying partitioning degrees.

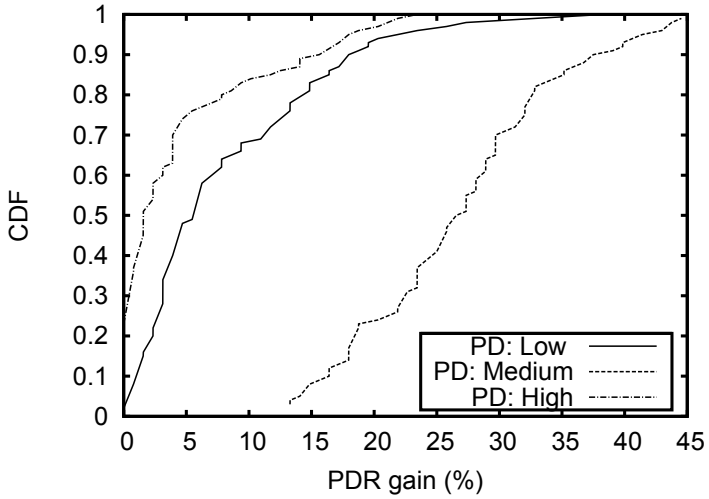


Figure 5.9: CDF of PDR gain achieved by AODV-OPP over AODV.

The cumulative distribution function (CDF) of PDR gains achieved by AODV-OPP for each partitioning degree range i.e. High, Medium and Low, as shown in Fig. 5.9, provides the same observation. For example, for achieving PDR gain greater than 20%, there is about 80% of chance in the medium PD range (but only around 1% and 10% in the high PD and low PD ranges). In addition, it shows that AODV-OPP outperforms AODV over all 300 scenarios, with the maximum improvement of 45% in medium PD cases (around 13% improvement even in the worst case). Surprisingly, AODV-OPP is able to outperform the original AODV over 5% in PDR with about 50% chance in low PD cases and about 25% chance in high PD cases.

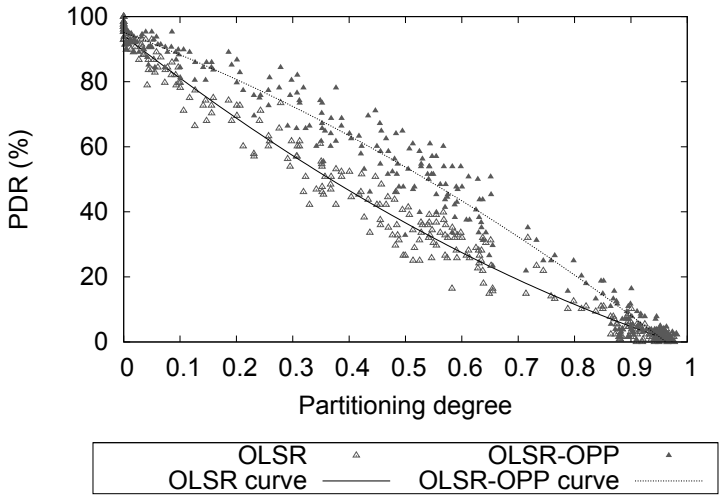


Figure 5.10: PDR of OLSR and OLSR-OPP for varying partitioning degrees.

5.4.2.3 Performance of OLSR-OPP in 300 PD scenarios

Fig. 5.10 shows the PDR of OLSR and OLSR-OPP for the 300 scenarios (in points) and outlines the relationship between the partitioning degree and PDR (in fitted curves using second degree polynomial). The first observation is that with the increase in partitioning degree (PD) PDR is reduced for both protocols, whether OLSR or OLSR-OPP, which is obvious due to increase in PD value network become sparse. Also OLSR-OPP is able to outperform over OLSR for around 93% scenarios. To evaluate performance in all three ranges a separate CDF plot for all the PD ranges is shown in Fig. 5.11. From this figure it can be verified that in the medium and high PD ranges OLSR-OPP outperforms OLSR for the entire range of scenarios. In some negligible cases (i.e. 1%) of low PD range OLSR shows slightly better performance than OLSR-OPP.

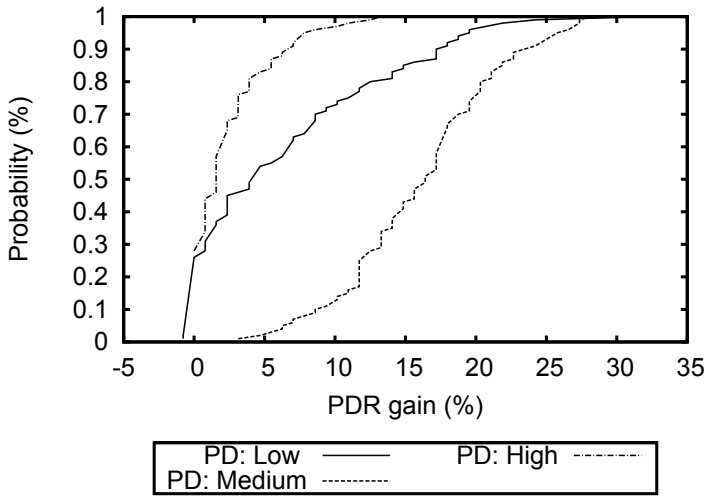


Figure 5.11: CDF of PDR gain achieved by OLSR-OPP over OLSR.

5.4.3 Realistic trace results

The tool Bonn Motion allows us to generate synthesised traces that conform to a particular characteristic of the scenario (e.g., different densities or node connectivities). This allows us to carry out systematic tests for the proposed hybrid protocol.

This section presents the performance evaluation of AODV-OPP and OLSR-OPP using realistic mobility patterns gathered by the GPS devices mounted on the San Francisco city cabs [3]. The use of these realistic traces serves as a means to validate observations from previous tests and to demonstrate AODV-OPP and OLSR-OPP performance in real life scenarios.

GPS traces are converted into the NS2 simulation and these traces are used as the node mobility model. As for the traffic model, 500 different traffic models are generated (e.g., each with different sender-receiver pairs and connections are formed at random times). The simulations involve 116 mobile wireless nodes.

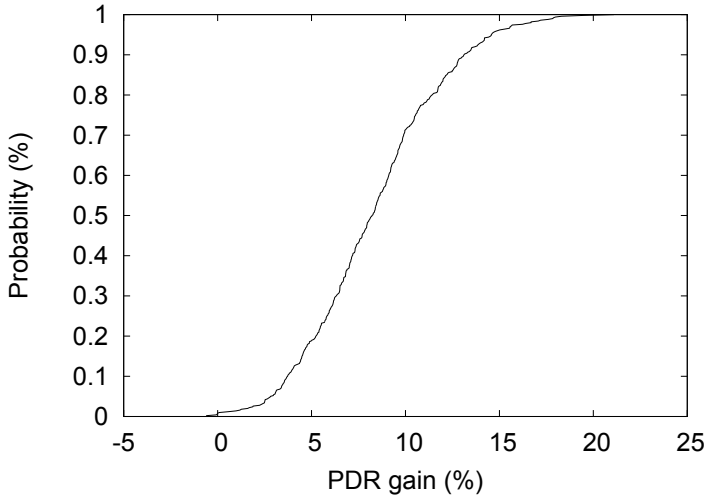


Figure 5.12: The CDF of PDR gain for AODV-OPP over AODV in realistic trace

Fig. 5.12 shows CDF of the PDR gain achieved by AODV-OPP over the original AODV. As shown in the figure, more than 99% of the 500 different scenarios, AODV-OPP outperforms the original AODV. The overall average gain in PDR is around 8%. There is around 70% chance that AODV-OPP can achieve less than and equal to 10% of PDR gain. It is observed that in a very small number of cases (less than 0.8%) AODV performs better than AODV-OPP. The negative PDR gain may be caused by the interference introduced when sending buffered packets hop-by-hop in AODV-OPP. These additional transmissions might compete with the data sent by AODV.

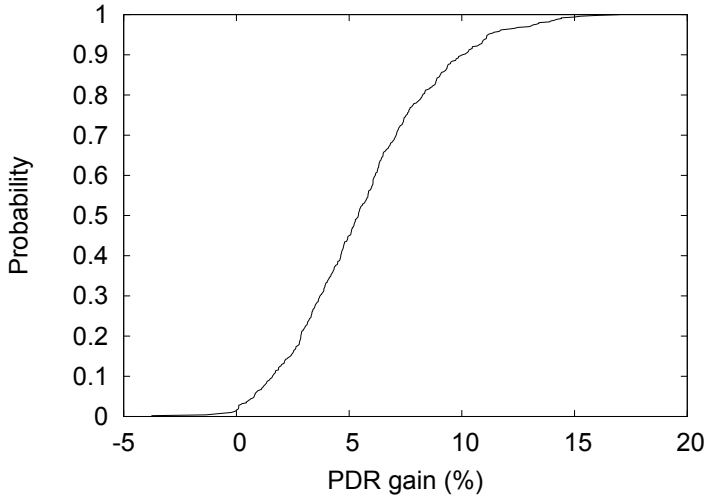


Figure 5.13: The CDF of PDR gain for OLSR-OPP over OLSR in realistic trace

Fig. 5.13 shows the CDF plot of PDR gain over 500 runs in case of OLSR and OLSR-OPP. As shown in figure OLSR-OPP outperforms OLSR in 99% cases. In around 20% cases PDR gain of OLSR-OPP is 10% over OLSR.

The result from this set of experiments verify that OLSR-OPP and AODV-OPP can also function in real time scenarios and are able to outperform OLSR and AODV, respectively, in most cases.

5.5 Summary

This chapter presented a systematic evaluation of the protocols in three folds. Firstly it discussed the validation tests that verify that the developed simulation model works correctly as expected in various network conditions. Secondly, it evaluated performance of the protocol in varying network conditions on the basis of their partitioning degree. Result of this experiment verifies that hybrid extension of AODV and OLSR can achieve higher PDR as compared to their traditional version. Third and last - the protocol performance was verified in real time traces.

It can be concluded that both hybrid protocols AODV-OPP and OLSR-OPP can outperform their traditional version i.e. AODV and OLSR, respectively.

The next chapter discusses a new metric which can be utilized to further improve the hybrid protocol's performance. With the use of this new metric two different designs are proposed for the reactive hybrid protocol in wireless mesh networks that extend the AODV-OPP i.e. AODV-OPP+ broadcast and AODV-OPP+ unicast.

CHAPTER 6

AODV-OPP+: hybrid protocol designs using metric

Basic principle of an end-to-end routing protocol is that there is always a route between a source and a destination pair. When a route is broken due to any link failure and no alternate route is possible then all the packets destined to that route are dropped. This leads to the degraded performance of an end-to-end routing protocols in terms of PDR. To improve the performance in such network situations previous chapters presented the design and development of a hybrid routing protocol that can be applied to both reactive and proactive routing in wireless mesh networks. As per the design principles of the hybrid protocol when no end-to-end route is possible in a similar network situation then packets are not dropped and on the contrary these packets are sent to all the neighbour nodes in the vicinity to increase PDR. For each packet received at a node, the node finds if it is not the destination node. If not then the hybrid protocol first finds possible partial path to the required destination node and then performs opportunistic routing if no route is possible. In this initial hybrid design all the neighbour nodes can participate in hybrid routing which can lead to significant overhead in the network. Therefore this chapter discusses a new metric, ***reachability***, that can determine the potential forwarder(s) in the network, in any network situation. So that hybrid protocol can allow only potential forwarder(s) to participate in opportunistic forwarding when no route is available.

This chapter presents two variations of the hybrid protocol on the basis of reachability i.e. AODV-OPP+ unicast and AODV-OPP+ broadcast. In both the designs packets are buffered at node when route to the destination is broken due to any reason. One of the major differences in both designs is that in broadcast approach a buffered packet is broadcasted to all its

one hop neighbours and only potential forwarders participate in opportunistic forwarding in case of no route. Whereas in unicast approach buffered packet is sent to only potential forwarder(s) in case of no route is found. This chapter discusses both the designs in detail.

These two proposed designs are extension of the AODV-OPP therefore they are named as **AODV-OPP+ broadcast** and **AODV-OPP+ unicast** where underlying base protocol is AODV as AODV is a more usable routing protocol in hybrid wireless networks with node mobility. To systematically present the designs of both of the extended hybrid protocols this chapter has the following subsections, (i) computation of reachability, (ii) common features of both designs, (iii) & (iv) detailed description of algorithms required for both designs along with an example scenario to give the insight of their forwarding mechanisms, (v) discussion on factors affecting delivery ratio of hybrid protocols and (vi) systematic evaluation of the protocols.

6.1 Reachability

In the initial concept of hybrid protocol as illustrated in Fig.4.2(a), when an end-to-end protocol decides to drop data packets in case of absence of route to the destination it forwards these packets to all one-hop neighbours for multiple times, as defined by a C_{retry} parameter. Sending buffered packets to all neighbours can lead to a higher delivery probability, but also results in higher overhead.

Goal of this chapter is to further improve the performance of the hybrid protocol that can lead to the higher delivery probability by minimizing the overhead. To achieve this goal a new metric reachability is proposed which measures the probability of a node having connection to the desired destination. Potential forwarder(s) can be selected in the neighbourhood on the basis of their reachability value. Whenever a packet is sent then packet's receiver must have higher reachability than the sender of the packet. This way the data packets are likely to travel towards the desired destination. Another advantage of such forwarding is that routing loop is prevented.

Due to the connectivity pattern of network nodes in disruptive networks, contact frequency and duration are commonly used as an indicator for predicting the delivery probability [57]. In our hybrid approach, a node can establish connection with the desired destination either by direct encounter or via an end-to-end route. Therefore, each node can have two reachabil-

Table 6.1: Example of reachability table

Destination	Neighbour	$R_{encounter}$	R_{route}
Node10	Node1	**35%**	23%
Node10	Node2	43%	**67%**
Node12	Node3	0%	**10%**

ity metrics for a desired destination node, namely $R_{encounter}$ and R_{route}. The highest one of the two values is used as the node's reachability to the destination, that is $max(R_{encounter}, R_{route})$. Table 6.1 shows an example of the neighbouring nodes' reachability to the destinations. The value in **bold** will be used as the reachability representing how likely a neighbour node will form a connection to these three destinations. In best of my knowledge this reachability metric is the first connectivity metric that has end-to-end route information along with contact duration and contact frequency in it. This metric provides higher probability of coming in contact with destination node as compared to the existing delivery predictability metric [56, 51, 57].

The reachability R (either $R_{encounter}$ or R_{route}) of a node are computed separately as

$$R = (1 - \alpha) * R_{old} + \alpha * R_{measured} \tag{6.1.1}$$

where $R_{measured}$ is the respective probability in the last measurement window; R_{old} is the historical probability (initialized to zero when node first bootup); and α is an adjustable parameter, which controls the weight between the history and new measurements.

To measure the reachability $R_{measured}$ of a node to other nodes in the network following equation is used.

$$R_{measured} = \frac{\sum T_{connection\_duration}}{T_{window}} \tag{6.1.2}$$

where $\sum T_{connection\_duration}$ is the sum of the duration (in time unit) the two nodes stay connected (with respect to having direct contact or having connection via a route) within a period of time T_{window}. During the time T_{window}, a background process records the total time any two nodes are connected, either through direct encounter or an end-to-end route. Here, background process refers to the re-computation of connection duration between nodes as per the assigned frequency of timer. T_{window} represents the measurement window and is a tunable parameter depending on the node mobility in a particular scenario. When the network is relatively mobile, then T_{window} needs to be relatively small to cope with the rapid

changes in the topology. Section 6.5 presents the discussion required to investigate the optimal value of α and T_{window}. Next section discusses the common features of both the variance of hybrid protocol i.e. AODV-OPP+ unicast and AODV-OPP+ broadcast.

6.2 AODV-OPP+ broadcast and unicast: common features

The previous section presented computation of reachability which can be used to further improve the performance of the hybrid protocol in any wireless mesh network's situation. Before going into details of both of the strategies separately, this section describes their common features. Motive behind this discussion is to create a baseline to compare both strategies. Common features of hybrid protocols are described in the following subsections.

6.2.1 Reachability (R) measurement

It is a routing metric which is computed for a particular destination. It is a probability measurement as defined in Section 6.1.

6.2.2 History of routes and direct contacts

It is a record of all the available end-to-end routes and direct meetings with the destination node. This information is (re)computed for every measurement window size (i.e. T_{window}). Measurement window defines a history period for updating the existing measurements for routes and direct contacts. These values are computed as a moving average where the moving factor can assign weights to past value and measured value in the current window. Size of the measurement window can also affect the measurement of the metrics that is explored later in this chapter.

6.2.3 BufferQueue: an improved rqueue

Most of the end-to-end routing protocols will simply drop packets when no route to the destination is available, AODV (used as base for the development of AODV-OPP+) has a limited support for buffering dropped packets due to temporary disconnection. This feature is called *rqueue* in AODV. To maximise compatibility with the existing protocol, AODV-OPP+

Algorithm 4 BufferQueue's storage management: n_s is a source node, n_{fwd} is a potential forwarder and n_d is a destination node. Reachability of both nodes n_s and n_{fwd} for the destination n_d are R_s and R_{fwd} respectively. R_{min} is the minimum reachability value for a packet's destination exists in the BufferQueue.

```
Packets p sent from n_s and received at n_fwd
if BufferQueue at n_fwd is not FULL then
  Buffer p at n_fwd;
else
  R_min = getMinReachabilityValue(BufferQueue at n_fwd )
  if R_fwd < R_min then
    Reject p;
  else
    Replace p with oldest packet with R_min;
  end if
end if
```

aims to replace the rqueue, redesigns packet buffering support and introduces a *BufferQueue* to incorporate features of delay-tolerant networks into the typically end-to-end communications. When a packet is stored in a BufferQueue it has attributes (shown in Fig. 4.1) as follows: (i) number of times this packet can be sent opportunistically i.e. C_{retry} , and (ii) life of the packet in the network i.e. ttl_{time}.

6.2.4 BufferQueue management

BufferQueue's size depends on the individual devices in the network. Therefore hybrid protocols support a storage management to efficiently utilise the storage at each node. To accommodate new packets when there is no space left in BufferQueue AODV-OPP+ follows replacement policy on the basis of strategy shown in Algorithm 4.

According to the algorithm when a packet p arrives at a node n_{fwd} and it has space to accommodate this packet in BufferQueue then protocol buffers it. When BufferQueue is FULL or in other words when its size reaches its limit then to accommodate new packet p hybrid protocol finds a packet in the BufferQueue for which node has lowest reachability value to its destination i.e. R_{min}. If forwarder's reachability R_{fwd} for destination n_d is lower than the existing lowest reachability R_{min} then this packet is ignored or rejected because the node has very small chances to meet its destination node. Vice versa if $R_{min} < R_{fwd}$ then packet p can be buffered at the potential forwarder. In this case packet p replaces the oldest packet in the buffer with lowest reachability value i.e. R_{min}. Hence nodes prefer to buffer only those packets for which the node has higher reachability so that it can maximize the delivery ratio.

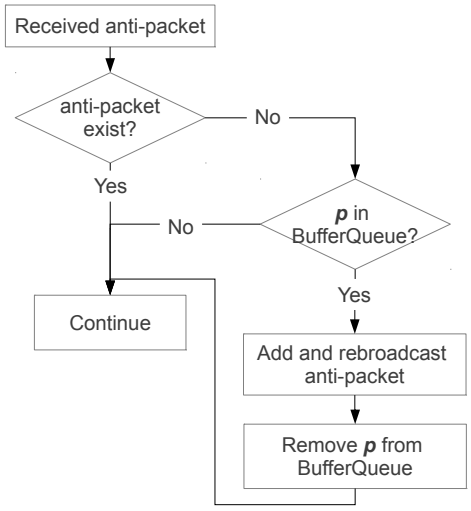


Figure 6.1: On receiving anti-packets in AODV-OPP+.

6.2.5 Anti-packets

To limit the number of buffered packets in the network, this design also introduces the anti-packets. The concept is based on the work described in [79] according to that efficient utilization of the available resources is also required to get the satisfactory performance of the network protocol. Therefore unnecessary packets are removed from the network. To accomplish this goal anti packets are used as an identifier so that packets that reached the destination are removed by the other nodes, so that more packets can be accommodated. As soon as a packet is delivered at the destination an antipacket is broadcasted in the network. Upon receiving such anti-packet nodes remove the packet's copy from their buffers, if it exists, and stores antipacket so that it can avoid storing this packet again in its buffer.

Fig. 6.1 shows the process of handling the anti-packets on receiving them in improved variants of hybrid protocol. In this protocol, the anti-packets are disseminated by broadcast when a packet is successfully received at the destination node. Upon receiving anti-packets the protocol checks their existence in the records and if they already exist it simply ignores them. Otherwise, protocol checks whether a BufferQueue has a packet corresponding to this anti-packet. If such a packet exists, the protocol records the anti-packet and rebroadcast it and also removes the packet from the BufferQueue. If such a packet does not exist in the BufferQueue then the antipacket is ignored. This way the anti-packets are likely being forwarded towards the nodes (holding copies of the packet) on the reverse path.

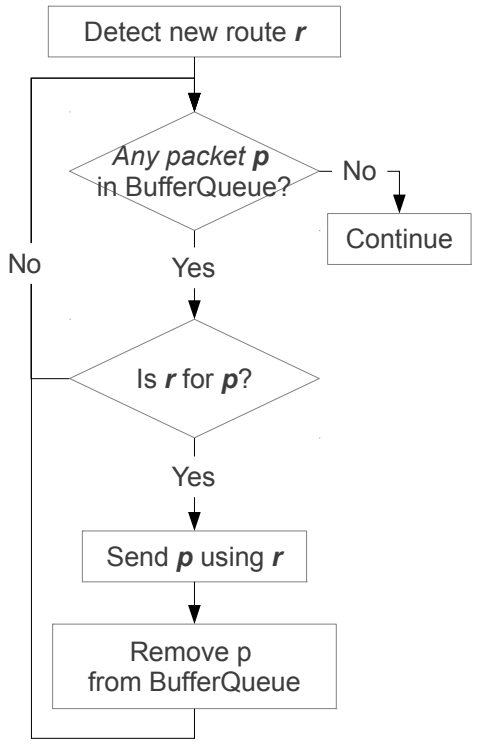


Figure 6.2: Detecting new route in AODV-OPP+.

6.2.6 Detecting a new route

A route may form due to nodes' mobility in the network. When detecting a new route r, both variations of AODV-OPP+ will try to send the buffered packets from the BufferQueue using r, as shown in Fig. 6.2. After sending p, AODV-OPP+ removes p from the BufferQueue and processes the next packet in the buffer.

6.2.7 Pending-send

This protocol has a provision to check its network interface queue (IFQ) before transmitting a packet so that if IFQ is full then it can hold the transmission until IFQ has space to accommodate another packet that can reduce packet drop in the network due to IFQ.

To accomplish this task the hybrid protocol keeps record of the packets a node wants to transmit but due to full IFQ they were ignored. These packets are added to a list *pending-send*. For each packet in the pending-send list hybrid router periodically performs attempt for transmitting that packet. This feature can reduce IFQ drops in the network. In the real implementation it requires access to the network interface queue that may or may not be provided by the manufacturer.

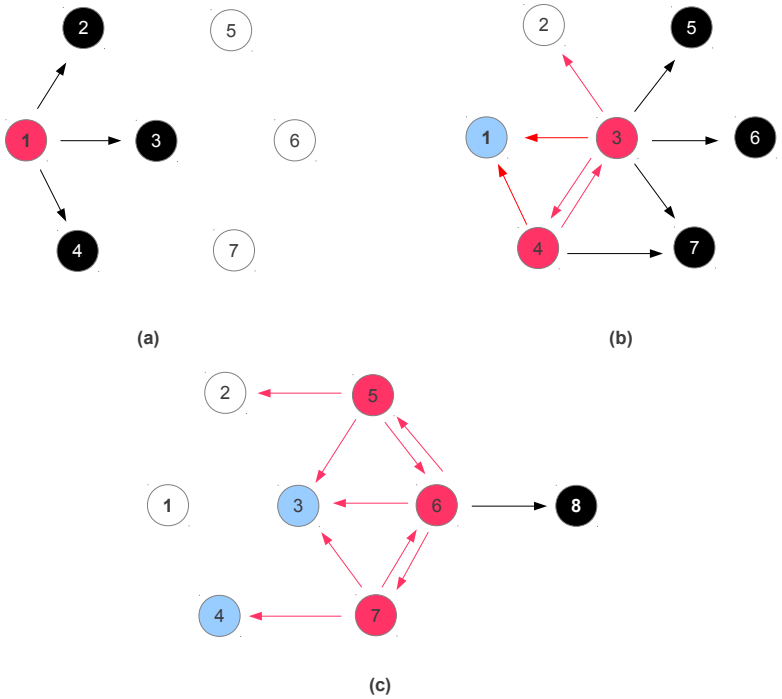


Figure 6.3: Example of AODV-OPP+ broadcast-based forwarding mechanism.

This section discussed common components of both variants of the dynamic switching forwarding methods i.e. AODV-OPP+ broadcast and AODV-OPP+ unicast. Subsequent subsections present the detailed description of their algorithms.

6.3 AODV-OPP+ broadcast

As compared to the initial design of the hybrid protocol (Section 4.2) in which packets dropped due to link failure are sent to all the neighbour nodes and all the neighbour nodes can participate in hybrid forwarding, AODV-OPP+ broadcast allows only potential forwarders to participate in the hybrid forwarding. These potential forwarder(s) are selected based on the reachability which is discussed in Section 6.1. This section presents an example scenario to show its forwarding mechanism and then detailed descriptions of **AODV-OPP+ broadcast** algorithms required in addition to the features described in Section 6.2.

6.3.1 Example of AODV-OPP+ broadcast

Having described the reachability metric and a discussion on some of the components in Section 6.2 this section presents an example to show the packet forwarding in AODV-OPP+ broadcast.

Fig. 6.3 illustrates an example scenario to show the AODV-OPP+ broadcast forwarding mechanism. To explain the mechanism first assumption is that a link from node 1 to the destination node 8 is broken and an alternative route could not be found resulting in packet drop. As shown in Fig. 6.3(a) node 1 buffers subsequent packets and broadcasts them to its one-hop neighbours (nodes 2, 3, 4) attaching its reachability (R_1) to the destination for this packet. Node 1 marks itself as in the "broadcast" state and waits to overhear rebroadcast from its neighbours. Upon receiving the broadcast packets as shown in Fig. 6.3(b), nodes 2, 3 and 4 receive the packets if they are the destination or can forward the packets if they have a route to the destination. Otherwise they check their respective reachability to the destination (R_2, R_3 and R_4). Nodes with greater reachability than R_1 buffer the packets in their *BufferQueue* and rebroadcast these packets with their own reachability. It is assumed that nodes 3 and 4 satisfy the conditions and broadcast the packets whereas node 2 ignores the packets. Because nodes can only participate in the packet forwarding if and only if they have greater reachability to the destination, the buffered packets are likely to be heading towards the destination.

When node 1 overhears the rebroadcast packets from nodes 3 and 4, it adds nodes 3 and 4 into the forwarder list and increases packet's C_{retry}, which was initialised to 0. When the value of $C_{retry} > 0$, it indicates corresponding packet has been received by the potential forwarder(s). To seek the opportunity to reduce packet delivery delay when $C_{retry} = 0$, the packets are kept for direct transmission to the destination or via a route. A packet can be purged when its ttl_{time} is expired.

In the same way the buffered packets are forwarded from node 6 to node 8 until they arrive at the destination or their ttl_{time} expires, as shown in Fig. 6.3(c). In this round nodes 3 and 4 were the senders of the packets. They record forwarders in their forwarder list upon receiving the rebroadcast packets from nodes 5, 6 and 7. AODV-OPP+ broadcast prefers end-to-end routes as they increase the protocol performance. Therefore, if any node in the forwarding path has an end-to-end route to the destination then packets can be delivered via that route. As discussed in Section 6.2 AODV-OPP+ broadcast also introduces *anti-packet* as a means to reduce overhead by removing packets along the packet forwarding path when it is already received by the destination.

6.3.2 Algorithms of AODV-OPP+ broadcast

Having described the forwarding mechanism of AODV-OPP+ broadcast, there are the following important functionalities that should be included in its design along with common features discussed in Section 6.2 to support dynamic switching between routing modes.

6.3.2.1 Handling packet drops

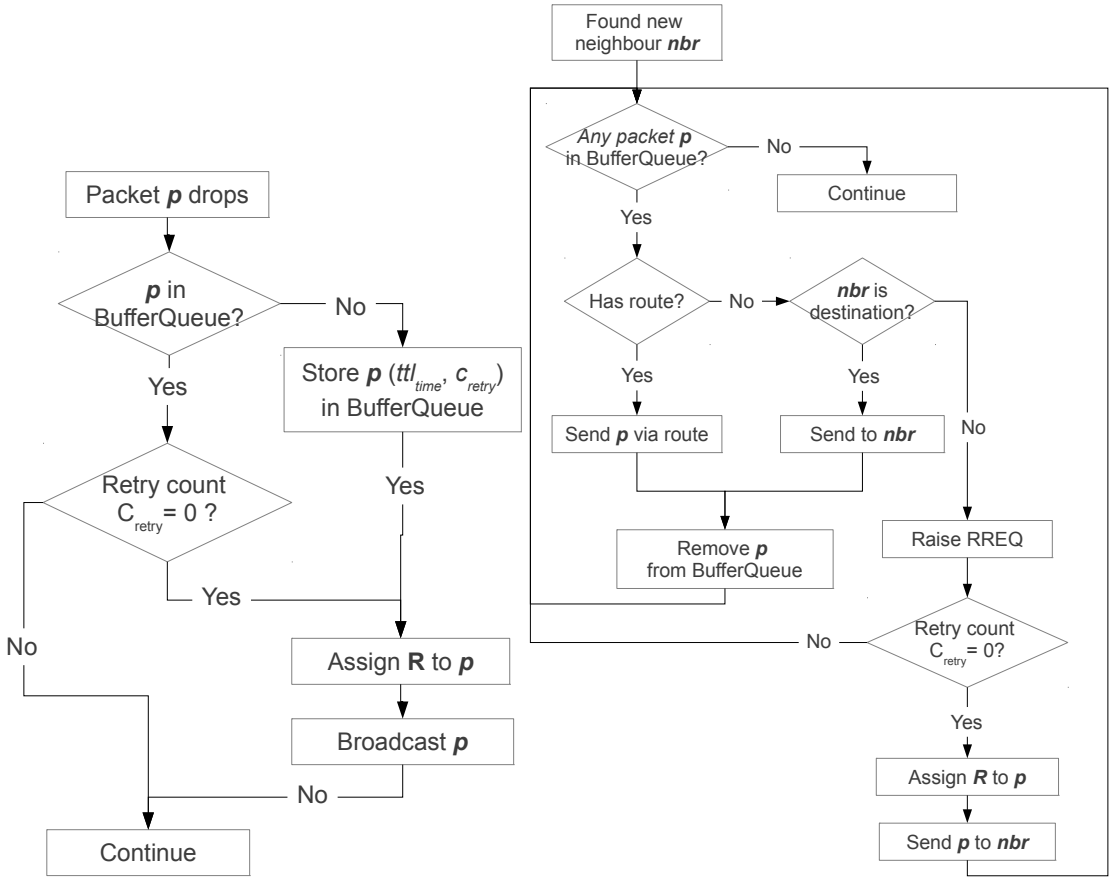
Most routing protocols (AODV, OLSR) detect link failures either by the loss of periodic Hello messages or a mechanism called *Link-layer feedback*. When a link fails and no alternative routes exist the subsequent data packets for the same unreachable destination are dropped.

As illustrated in Fig. 6.4(a) if dropped packet p does not exist in BufferQueue then it will be stored at the node with attributes C_{retry} and ttl_{time} as shown in Fig. 4.1. C_{retry} is initialized to 0. If $C_{retry} = 0$ then node's reachability R (to the desired destination of p) is attached to the packet and broadcasted to one hop neighbours.

When node ensures that this packet is received by any potential forwarder(s) then keeps p in BufferQueue to provide a backup in case none of the packets broadcasted to the neighbours arrives the destination. So that in future backup copy can be transmitted if node encounters destination node directly or via an end-to-end route. Strategy to keep records of packet's potential forwarders is discussed in a Section 6.3.2.3.

6.3.2.2 Meeting a new neighbour

As the nodes move around they form connections from time to time with the help of periodic Hello messages. When a node receives a Hello from other node it can record that node as a neighbour node. As shown in Fig. 6.4(b), when a node meets a new neighbour it checks whether the new neighbour can contribute in forming an end-to-end route to the destination so that the buffered packet p can be sent. If so, p can be removed from the BufferQueuee. However, in AODV when two nodes meet, routes from a node are not automatically populated to another node without a route request (RREQ). Therefore, at the time when two nodes meet and if the new neighbour is not the destination and no existing route can be found through it, then a RREQ can be raised for the destination (in the hope to discover a new route). In our approach rather than waiting for a route reply (RREP) or a RREQ timeout (default to 10 s), packets are sent to the new neighbour with the reachability R of the



(a) Handling packet drops. (b) Meeting a new neighbour.

Figure 6.4: AODV-OPP+ broadcast algorithm for dynamic switching

node given that this packet is not received by any potential forwarder. In other words, our approach tries to use the first packet for a particular destination to probe a new route. When a new route is formed via the neighbour then all the subsequent packets are delivered using the route. A RREQ list is also included to avoid sending multiple RREQ message for the same destination from a node.

6.3.2.3 On receiving a packet

As shown in Fig. 6.5, when a data packet p arrives at a node, AODV-OPP+ broadcast first checks whether the current node is the destination for p. If p arrives at the destination then it is sent to upper layers otherwise the node that receives p can check the broadcast flag and execute one of two operations: (i) If the node has broadcasted some data packets in the past (that means this node is a sender) then it checks whether its reachability is smaller than the reachability value specified in the received packet p (i.e., $R_{node} < R_p$). If $R_{node} < R_p$ holds true, the current node (as the sender) can add all those neighbours that re-broadcast the packet p to the forwarder list. In the design of AODV-OPP+ broadcast C_{retry} is initialised to

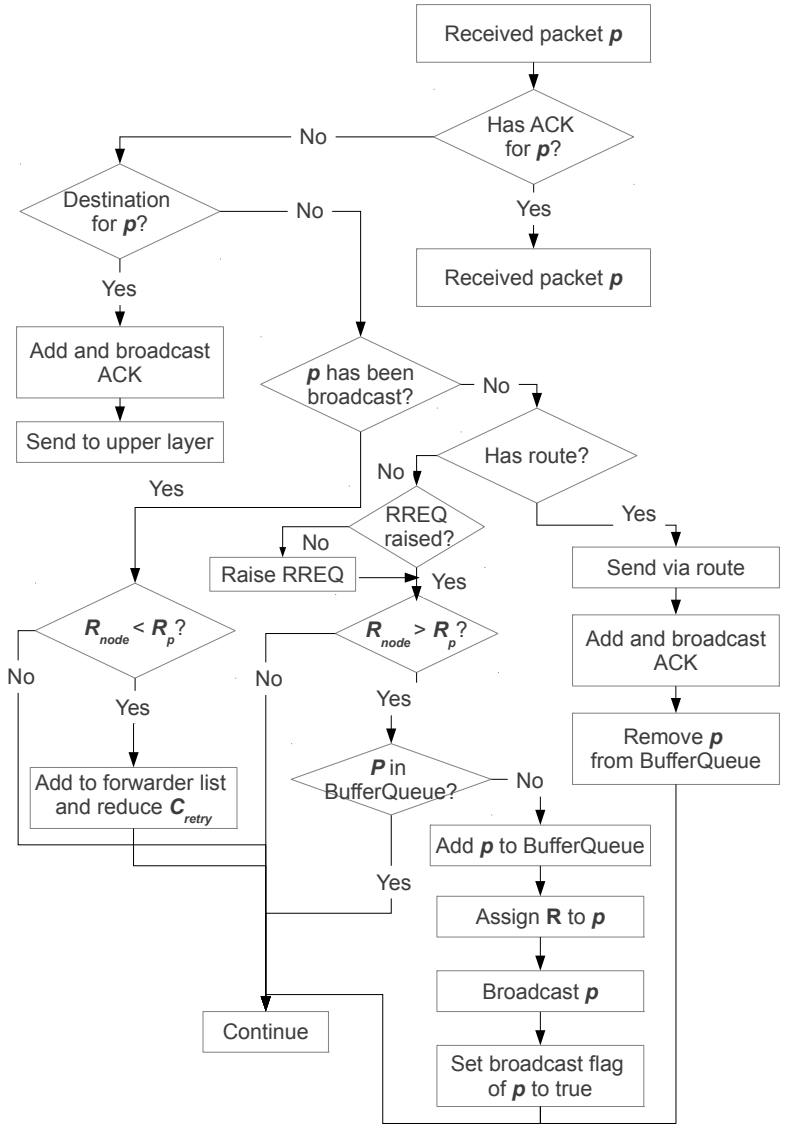


Figure 6.5: On receiving a packet in AODV-OPP+ broadcast.

0. Whenever a node adds a forwarder to the forwarder list, C_{retry} is simultaneously increased by one. (ii) If the node had not broadcasted any data packet in the past (that means this node is a forwarder) then it will first try to lookup an active route for the received packet (the packet will be sent via route if it exists otherwise RREQ is raised to discover one if not raised yet and the packet is removed from the BufferQueuee). If no active route is found, the receiving node checks $R_{node} > R_p$ or if the receiving node has 100% reachability. If the condition does not holds then it ignores the packet. Otherwise the receiving node tries to buffer the packet (according to buffer management described in Section 6.2) and rebroadcast it with its own reachability.

6.4 AODV-OPP+ unicast

This section presents another variance of AODV-OPP+ i.e. unicast approach. In this version a copy of a buffered packet is sent only to neighbours having higher reachability to the desired destination. Section 6.2 discussed common features in both variants of AODV-OPP+ designs. This section focuses on the detailed description of additional algorithms required in the design of AODV-OPP+ unicast which is followed by an example scenario to give insight into its forwarding mechanism.

6.4.1 Example of AODV-OPP+ unicast

Fig. 6.6 illustrates the packet forwarding mechanism of the AODV-OPP+ unicast for the similar network situation as discussed in Section 6.3.1 to show the AODV-OPP+ broadcast forwarding mechanism. Hence in the network packets are flowing to reach the destination node 8.

In the unicast design all the participating nodes share their *reachability* for the destination node 8 whenever they encounter each other and maintain their forwarder lists accordingly as shown in Fig. 6.6(a). A forwarder list is the list of neighbour nodes that have higher reachability than the node itself to reach the destination node. According to the Fig. 6.6(a) Node 1 has 20% reachability to node 8 and it has node 4 and node 3 as potential forwarders with reachability i.e., 34% and 40%, respectively.

As shown in Fig. 6.6(b), node 1 has packets to send to node 8. Due to a link failure node 1 buffers packets. For each packet in the buffer node 1 checks existing forwarder(s). As illustrated in Fig. 6.6(a) node 1 has two potential forwarders i.e. node 3 and 4. Hence two copies of a packet are sent to each of them and corresponding C_{retry} can also be reduced. In the next step when packet reaches node 3 and 4, router first checks for existing routes from these nodes, if possible. In case of no route packets are buffered at both the nodes and a copy of packet is sent to the potential forwarders. Also if a node encounters a neighbour that has higher reachability to the destination then the packets are delivered to that neighbour node if $C_{retry} > 1$. In this protocol C_{retry} is used to control the number of opportunistic attempts that can be performed on a packet. Unicast approach also keeps last copy of a packet in case packet is not delivered via opportunistic trials and can be delivered via direct/end-to-end route, if possible. This process continues until packet reaches the destination or packet

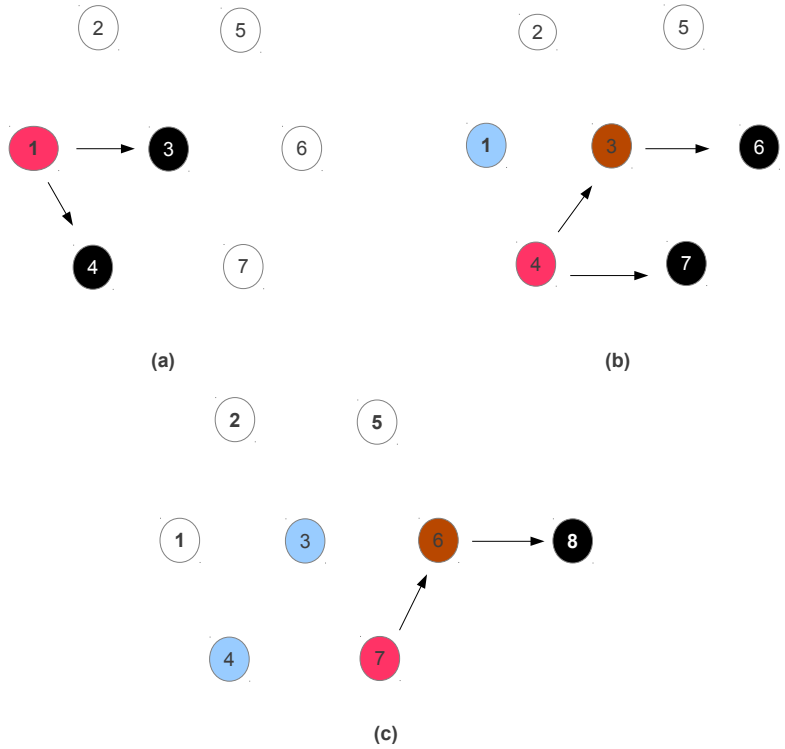
FWD_LIST: NODE 1 (20%)		
NODE 4	DST NODE 8	34%
NODE 3	DST NODE 8	40%

FWD_LIST: NODE 4 (34%)		
NODE 7	DST NODE 8	45%
NODE 3	DST NODE 8	40%

FWD_LIST: NODE 3 (40%)		
NODE 6	DST NODE 8	75%

FWD_LIST: NODE 7 (45%)		
NODE 6	DST NODE 8	60%

(a) Forwarder list.



(b) Packet forwarding.

Figure 6.6: Example of AODV-OPP+ unicast-based forwarding mechanism.

expires as per the value of ttl_{time} assigned in its IP header. Hence, in both hybrid protocols, AODV-OPP+ broadcast and AODV-OPP+ unicast, C_{retry} is computed differently. In AODV-OPP+ unicast, sender node reduces each packet's C_{retry} when packet packet is sent to the potential forwarder. On the other hand in AODV-OPP+ broadcast, when a node overhears the rebroadcast packet from the potential forwarders then node increases packet's C_{retry}. Therefore both protocols have different approaches to utilize C_{retry}. Detail description of these algorithms can be found in the next section.

6.4.2 Algorithms of AODV-OPP+ unicast

This section describes the algorithms that the AODV-OPP+ unicast requires to support dynamic switching between routing modes in addition to the features described in Section 6.2.

6.4.2.1 AODV-OPP+ unicast: forwarder list

One of the unique feature of this design is the forwarder list. Every participating node in the network maintains a list of potential forwarders in its vicinity. A forwarder list has three attributes i.e.

$< neighbour\_id, destination\_id, reachability >$ where neighbour with neighbour_id has specified value of reachability to reach the destination_id. In order to create and manage this list all the participating nodes share their neighbours reachability with other nodes in the form of a reachability vector. As a result nodes know the reachability of other nodes.

One of the goal of this design is to optimize the size of the reachability vector so that nodes do not share their complete reachability vector unnecessarily. Another goal is to minimize the additional transmissions to share this vector. In order to achieve these goals AODV-OPP+ unicast rely on its base protocol's functionalities i.e. AODV.

To optimize the size of the reachability vector

As discussed earlier the base protocol for AODV-OPP+ unicast is AODV which is a reactive protocol. In AODV [70], whenever source node has data to send it initiates route discovery process. In route discovery phase source node broadcasts route request control messages (RREQs). Whenever a node receives RREQ and is not the destination then it rebroadcasts RREQ. This process repeats until RREQ reaches the destination node or RREQ expires. When RREQ reaches the destination node, route reply control message (RREP) is generated by the destination node and unicast towards the source node. When RREP is traversing towards the source node all the nodes receive the RREP message updating their routing tables. Therefore unless a node receives RREQ it does not know about the possible routes in the network. If all the participating nodes keep record of all these RREQ's destinations then they can determine the demanded destinations in the network so that nodes can advertise reachability only for those destinations that can significantly reduce size of the reachability vector which is shared between nodes.

As a result AODV-OPP+ unicast can efficiently advertise the reachability vector as compared to the existing protocols [8, 52, 56] in which nodes unnecessarily share their complete delivery probability vector with their neighbours whether they are needed or not.

To minimize the transmissions to advertise reachability

In most of the routing protocols HELLO messages are used to update link state information (e.g. in OLSR) or neighbour information (e.g. in AODV). These messages are small in size and are broadcasted periodically from a node. In case of AODV, as soon as a node receives a HELLO it adds the source of HELLO as the neighbour node. Later on these neighbours can

cooperate in finding routes to the destination in route discovery process. In AODV-OPP+ unicast these HELLO messages are utilized to advertise the reachability vector. A technique to optimize the reachability information is discussed using the RREP message. To further reduce the size of the information that can be attached to these HELLO and its implications are discussed in this section.

Belding-Royer and Chakeres discussed issues related to accurate estimation of link-quality based on HELLO in [19]. According to the authors these messages should have similar characteristics as the data packets. For example, if the size of the HELLO packet is equivalent to the size of the data packet then reception of such messages not only determines the presence of a node but also indicates better link quality. Hence the use of the HELLO message to advertise reachability has two advantages. Firstly it can minimize the overhead because no additional transmission is required. Secondly it can maximize the chances of successful reception via that link [19].

In wireless network each frame has a limit on the maximum size of the packet (e.g. in IEEE 802.11 it is 2346 bytes). Therefore more than one HELLO are used if its size exceeds the average data packet size traversing in the network.

Whenever router add reachability to the *HELLO* and its size exceeds its average data packet size, then it can add rest of the information in the next *HELLO* message. This way router split information among more than one *HELLO* message by applying upper bound on the size of *HELLO* by average size of the data packets.

At the reception of such HELLO messages nodes' create/update their forwarder list. If a neighbour node has higher reachability to some destinations as compared to the node's own reachability then the neighbour node is added into the forwarder list. Similarly if a node receives a reachability for an existing forwarder then it updates the value to reflect the current network situation.

6.4.2.2 Meeting a new neighbour

Connection among nodes changes due to their movement. As a result node meets new neighbour which is not necessarily a potential forwarder and unless route discovery is initiated no route is generated via new neighbour. Therefore in AODV-OPP+ unicast whenever node encounters a neighbour and finds packets in BufferQueue then for each packet it checks the possible route. If route exists the packet can then be sent towards the route or

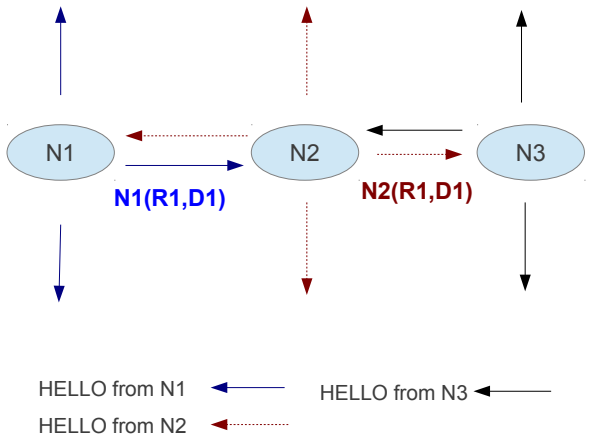


Figure 6.7: On receiving reachability.

if neighbour is the destination then packet can be sent to it directly. Those packets are also removed from the BufferQueue afterwards. But in case of no route a route discovery process is initiated if not yet raised.

In the network situation when new neighbour is a potential forwarder for a destination d, the node sends packets from its BufferQueue waiting to reach d. Although there are situations when BufferQueue does not have such packets then it finds ways to utilize such potential forwarder(s). As discussed in the previous subsection nodes have a record of destinations demanded in the network with help of RREQ raised/overheard in the network. Hence if a node has this destination node in its record then it assumes that one or more neighbours have packets to send to this destination. Therefore node broadcasts its neighbour's reachability (via HELLO) to invite other neighbour(s) so that they can send packets to it if they have any packets for that destination node. For example, as shown in Fig. 6.7 node N2 receives reachability R1 of N1 to destination D1, as N2 does not have any packet in BufferQueue it checks its RREQ records and finds the received request for this destination. If the node has received route request in past then it can broadcast its neighbour's reachability information. Upon receiving such HELLO, a node can identify either this node can be the potential forwarder or not. In case of potential forwarder N3 can transmit packets towards N2. In this way nodes have multi-hop information to transmit data in the opportunistic mode.

6.5 Analysis of factors affecting delivery ratio of both AODV-OPP+

Previous sections discussed the computation of reachability and also provided the detailed description of the algorithms that allows dynamic switching between routing modes. Both variants of the hybrid protocol have capability to switch between routing modes and they share similarity in a way that both protocols have a buffer space to store packets that are dropped due to link failure in a traditional end-to-end routing approach. This buffer is maintained by every participating node. One of the major differences between the protocols is that in the AODV-OPP+ broadcast approach buffered packet is broadcasted to all of its one hop neighbours whereas in the AODV-OPP+ unicast approach buffered packet is forwarded to only potential forwarders.

The IEEE 802.11 standards for wireless LANs include the distributed coordination function (DCF) that allows multiple nodes to access the medium. DCF is based on CSMA/CA with binary *backoff*. In case of unicast transmission the destination acknowledges successful transmission. Hence, for every unsuccessful transmission sender can (re)transmit frame. Therefore for each packet failure it performs trials at MAC layer [59]. In contrast to such mechanism, retransmission/recovery is not feasible in broadcast because frame is intended to all the stations within the transmission range. Hence, broadcast transmission assumes packet is delivered although it could have collided. Also AODV-OPP+ unicast can be performed at higher transmission rate whereas broadcast is performed at lower rate e.g. In the IEEE 802.11 standard unicast transmission rate can go up to 54Mbps(IEEE 802.11g) whereas basic rate is 6Mbps. Such difference in rate can result in a significantly faster transmission in the unicast protocol. Although higher rate allows faster transmission, unicast is also sensitive to the network conditions (e.g. interference, anomaly [42]). Hence, in a good channel condition unicast gives higher throughput, lower latency and higher PDR.
Before evaluating the performance of the hybrid protocol it is important to understand how various parameters involved in these protocols can affect its performance. This section presents an analysis of all those factors that can affect the performance of the hybrid protocol in terms of packet delivery ratio (PDR).

6.5.1 Window size (T_{window}) and EWMA weight parameter (α)

It is the time window that triggers the (re)computation of the reachability (R) at every node in the network. This metric is one of the important factors for selecting the potential forwarders in the network. Performance of the AODV-OPP+ protocol heavily depends on the accurate estimation of this metric. If connections among nodes are rapidly changing then nodes can compute reachability at a higher rate i.e. smaller value should be assigned to T_{window}. Whereas if nodes are mostly stable then lower rate of computation can give accurate estimation of reachability i.e. large value of T_{window}. For example, let us assume a 3 nodes scenario with nodes A, B, and C. Nodes A and C are not in direct range of each other (Distance between them is 1000m). However an intermediate node B moves back and forth between A and C at the speed of 25 m/s. Transmission range of all the nodes is 250 m. If initially node B is moving apart from node A and T_{window} is 10s (α = 1, no history involved) then for B reachability is 1 of node A at time = 10 s. Whereas if T_{window} is 40s (α = 1) then reachability of A will be 0.25 for node B. Hence, due to size of T_{window} the value of reachability changes significantly. Hence for a predictable network situation value of T_{window} can be appropriately selected. However for a dynamic network situation it is not a feasible solution. Hence mobility pattern is the useful context information to determine the appropriate window size (T_{window}).

In a dynamic situation a large value of T_{window} updates reachability record at a slow rate i.e. it produces stale information. As a result it can not accurately capture the network situation. Whereas frequent updates on reachability can capture the network situation at high rate. At high rate past and current measurement of reachability are close to each other, so that any change in the network situation can be exactly captured in the reachability whether static or mobile. Although in a static situation frequent updates are not needed as discussed but for an unpredictable situation frequent updates can closely represents the network dynamics. One of the consequences of frequent updates is that it requires faster computation capability at each participating node. In the worst scenario each node can have maximum (n-1) connections in a network of n nodes. Hence the reachability vector can have values which are in order of O(n-1). So that at every node the required computation depends linearly on the number of nodes. These computations are triggered at T_{window} size interval, however in the world of advanced technology with modern processors these frequent computations would not affect protocol performance. Also its computation relies on locally available context information which means that it is simple to gather as well as to compute.

According to the reachability computation another tunable parameter is α which assigns weights to both current and past measurements as shown in Eqs. 6.1.1.

If two nodes are not connected in the current time window or if two strangers meet in a time window then their reachability increases or decreases gradually by applying EWMA accordingly. Hence node is not completely relying on the current network situation because connection/disconnection among nodes can occur due to mobility or interference between them. In a real network situation total contact time among nodes over a specific time interval (T_{window}) depends upon the nodes encounter rate and lengths of each encounter, both of which depend upon the mobility of the nodes. If nodes move mostly among known nodes then they must benefit from their history, so more weight can be assigned to the R_{old}. Whereas if nodes mostly meet with new nodes then more weight can be assigned to the $R_{measured}$ in the current time window. Hence, it is important to know the movement pattern of the nodes before assigning weights to the current and past measurement components of reachability. Authors of [8, 26] assume 0.5 for α, so that equal weights are assigned to both current and past measurements. With $\alpha = 0.5$ the router assumes that in the next time window particular contact may or may not occur with equal probability unless any knowledge is provided regarding the application scenario.

6.5.2 Size of BufferQueue and packet's *ttl*

To perform the hybrid forwarding a node can buffer packets in the BufferQueue. Hence its size is also an important factor that contributes to the protocol's performance. If more packets can be accommodated then there are more chances to perform hybrid forwarding which leads to the higher PDR. However buffer size is limited and depends on the individual device specification. Hybrid protocols already have buffer management to efficiently utilize this space. Still larger buffer size can have higher probability to give higher PDR.

ttl_{time} is the life time of the buffered packet. Longer life of a packet provides higher chances to meet potential forwarders or to find an alternate route to reach the destination. On the other hand long packet life allows long delays in packet delivery and for some applications such packet could arrive after their expected time being of no use for the application. Therefore, packet delivery within a desired time limit is also a requirement in addition to getting high PDR. Hence, for a time sensitive application ttl_{time} can be set accordingly but if the delay is not an issue then longer ttl_{time} can give higher PDR in AODV-OPP+. As it is already

discussed *WhatsApp* is currently a very popular Android application. It is an alternative tool for text messaging via 3G or Wi-Fi where successful packet delivery is more important than the delay.

Another issue related to the higher ttl_{time} is overhead in terms of high resource consumption. Anti packets are already utilized by both variance of AODV-OPP+ so that it can limit the number of buffered packets by removing delivered packets from the BufferQueue of nodes.

6.6 Evaluation

This section presents the systematic evaluation of both types of AODV-OPP+ and shows their performance as compared to the other existing protocols. The section has two subsections. The first section discusses the methodology to systematically evaluate the proposed hybrid protocols and another section presents the evaluation results.

As discussed in Chapter 5, the NS2 simulation environment is chosen due to the lack of access to a large-scale testbed and fine-grain control of nodes' movement in the network. To validate the correctness of both AODV-OPP+ broadcast and AODV-OPP+ unicast, many iterations are performed on their simulation based models using diamond topology discussed in Section 5.3.1. Details of those validation tests are already described in that section. For both protocols (AODV-OPP+ unicast and broadcast) results of those validation tests were as expected that verifies the correctness of both the simulation based models.

6.6.1 Methodology

In this section protocol evaluation is carried out in three categories of simulation scenarios. Firstly tests are performed to support the analysis of parameters affecting delivery ratio of the hybrid protocol as discussed in Section 6.5. For this set of experiments synthetic mobility traces generated by varying speed of the nodes are used. Secondly performance of both the hybrid protocols is compared with the existing solutions in different network situations that characterise different aspects of the network scenario as described in Section 5.3.2.1. Thirdly tests are performed that use actual traces gathered from real life objects or human movement as described in Section 5.3.2.2 to validate the performance of the protocol. In these scenarios, extensive simulations are performed with a different scales and objectives. One of the objectives is to compare both variations of AODV-OPP+ broadcast and unicast with

Table 6.2: Simulation parameters

Propagation model	TwoRay ground
Antenna model	OmniAntenna
Nodes	50
Traffic	UDP CBR
Packet TTL	500 s
Data rate	4 pkts/s
Tx range	250 m
IFQ length	50 pkts
Simulation Time	500 s

the AODV, Spray-and-Wait [4] and AODV-OPP protocols. The proposed hybrid protocols are compared with the best representative protocols of both communication modes. Their source code is available in NS2.

Table 6.2 lists the common parameter settings that are used in the simulations. They are the default values for all simulations, unless discussed in the respective simulation scenarios. The rest of this section discusses the details of the synthetic trace with varying speed of the nodes.

As discussed in Chapter 5 for evaluating the protocol performance different types of mobility traces are used. Similarly synthetic traces are used to evaluate the AODV-OPP+ hybrid protocols. These synthetic mobility traces are of two types, one with varying speed of nodes and another with varying partitioning degree. Mobility traces with varying PD are already described in Section 5.3.2.1. Therefore in this subsection mobility traces with varying speed of nodes are discussed.

This set of mobility scenarios are generated to evaluate the optimum value of parameters involved in the hybrid protocol to support the analysis presented in Section 6.5. These parameters are heavily affected due to mobility of nodes as already discussed therefore for this set of experiments generated scenarios are varying in node speed. So that the parameters can be closely evaluated in different speed ranges.

Random Waypoint mobility patterns are generated using the setdest tool (version-2) which is a part of the ns-2 distribution. In all these scenarios, 20 mobile nodes move in the area of 1000m x 1000m for a period of 900s. These scenarios are categorised in 5 speed ranges. The first speed ranges from 1 m/s to 5 m/s, then the second ranges from 5 m/s to 10 m/s, after that 10m/s to 15 m/s, next from 15 m/s to 20 m/s, and the last from 20 m/s to 25 m/s. These speed ranges represent the walking speed of pedestrians to car moving on highways

[5]. In each category 100 mobility patterns are generated by varying the pause time of nodes (from 1s to 100s). For each of the 500 scenarios, simulation run for 10 seeds and average of packet delivery ratio (PDR) is computed. Evaluating the protocol performance on these scenarios demonstrate the impact of selecting different values for the parameters in different speed ranges.

For this set of synthetic tests, each of 20 nodes are allowed to form connections with any two other nodes in the network. These connections are formed randomly at different times during the simulation. UDP traffic is injected for the maximum duration of 100s (this duration is randomly assigned between 100s to 600s) and packet interval is 1s.

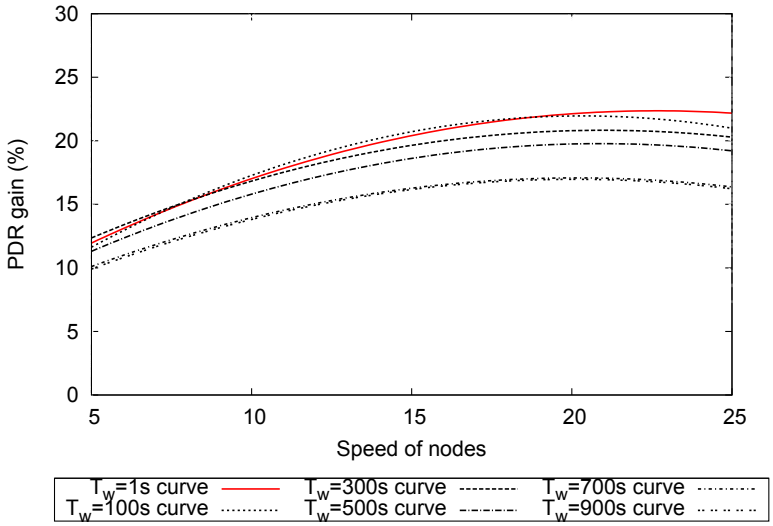
6.6.2 Results and discussions

This section presents the performance evaluation of the hybrid protocols based on the methodology described in the previous section and compares the performance of both types of AODV-OPP+ with AODV and Spray-and-Wait (SAW) in synthetic and real life mobility scenarios. Evaluation is carried out in three phases as follows: Firstly evaluation of the factors that can affect the protocol performance i.e. T_{window}, α, and ttl_{time}. Secondly evaluation of the protocols performance in varying network conditions, thirdly comparison of the performance of both proposed protocols over varying load conditions.

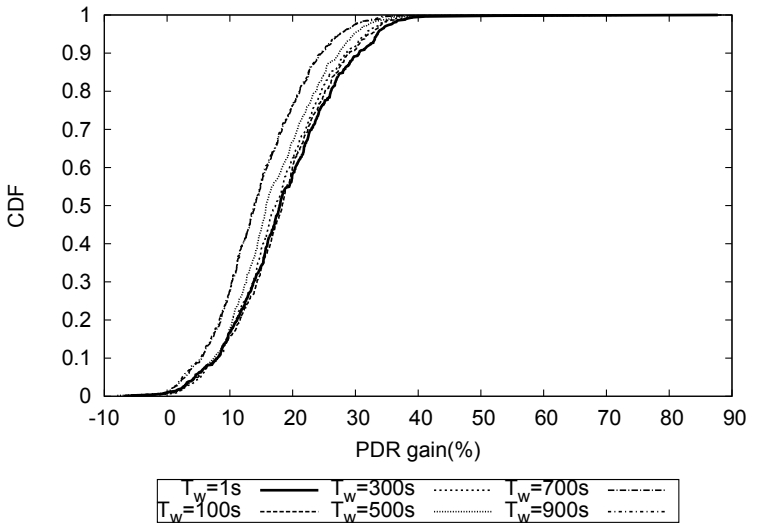
6.6.3 Evaluation of factors affecting delivery ratio of AODV-OPP+

For this set of experiments mobility scenarios varying in speed ranges are used (as discussed in Section 6.6.1). Both hybrid protocols AODV-OPP+ unicast and AODV-OPP+ broadcast are evaluated over these scenarios and both protocols generated similar results. Therefore only one of the results is discussed in this section i.e. AODV-OPP+ unicast.

Firstly the effect of T_{window} size is demonstrated. For this set of experiments α is set to 0.5 to provide equal probability to historic and current measurements (Eqs. 6.1.1) when reachability is computed. By using 500 scenarios (varying speed of nodes) discussed in Section 6.6.1, each mobility pattern simulated over 10 runs and average PDR over those 10 runs is computed for both AODV and AODV-OPP+ unicast. This experiment is repeated for the different values of T_{window} i.e. 1s, 100s, 300s, 500s, 700s and 900s. Hence in total 300,000 experiments are carried out to estimate the optimum value of T_{window} size.



(a) PDR in varying speed of the nodes.



(b) CDF of PDR gain in varying T_{window}.

Figure 6.8: Result for T_{window} using AODV-OPP+ unicast.

Fig. 6.8 (a) shows the fitted curves (using second degree polynomial) for the PDR gain over AODV in varying sizes of T_{window}. First observation is that AODV-OPP+ unicast outperforms AODV for all the mobility scenarios. This observation confirms that AODV-OPP+ unicast design gets benefit of its hybrid design where some of the packets are delivered via opportunistic mode.

Second observation is that with the increase in speed of nodes PDR gain also increases. As the speed of nodes increases connections among nodes change frequently. As a result AODV's route breaks and consequently packets are dropped in the network. As a result PDR decreases for AODV. Whereas due to the hybrid nature of AODV-OPP+ unicast those packets are buffered and delivered opportunistically. As a result PDR upgrades for AODV-OPP+ unicast.

Another observation is that for most of the mobility scenarios PDR gain is higher when T_{window}=1s. As shown in Fig. 6.8 (b) PDR gain reaches up to 23% when T_{window} is 1s and when T_{window}=100s for approximately 50% scenarios the protocols perform similarly. The reason for such behaviour is that in this experiment traffic is injected between 100s to 600s and due to on-demand nature of AODV no route can be found or established in the network until there is a packet to send. Even though when T_{window}=1s frequent updates can give better estimation of current network situation and because at this time no traffic is flowing in the network. Hence no end-to-end information is available for reachability until time reaches 100s when traffic started flowing in the network. It also verifies that if this context information (time at which traffic injected into the network) then T_{window}=1s can be set accordingly otherwise smallest possible window size is the optimum value for this.

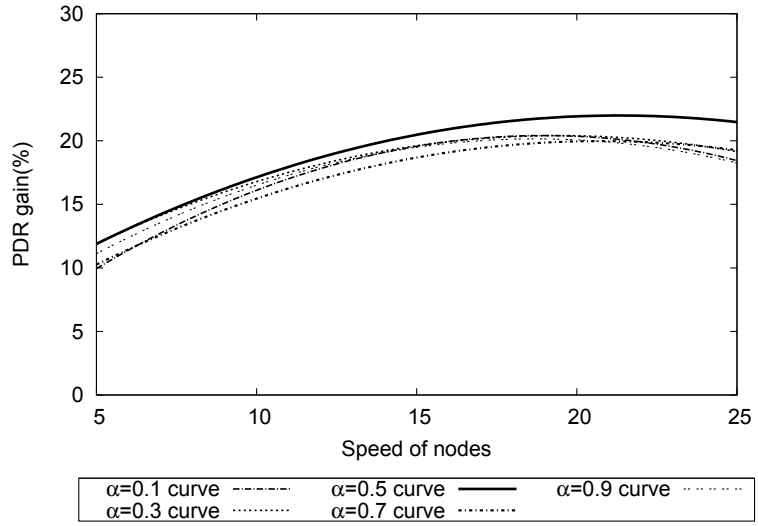
In other words even though analysis suggest frequent computation can estimate potential forwarders but if no route is found then reachability is only relying on the direct contact. Once traffic is injected at 100s, possible routes are created and because of that PDR gain is slightly higher (i.e. 0.2% to 0.3%) with T_{window}=100s in some cases. As soon as T_{window} increases, PDR gain of the protocol starts reducing which is noticeable.

Another observation is that when $T_{window} = 900s$ packets are delivered only via end-to-end route. As shown in Fig. 6.8 (a) AODV-OPP+ unicast at $T_{window} = 900s$ outperforms AODV even though no opportunistic delivery is involved. This PDR gain reflects the enhancement due to extended BufferQueue (replaced rqueue) where packets have higher chances to get delivered successfully. This set of simulations verifies our analysis that the *smaller the* T_{window} *size the higher the PDR gain* because the frequent updates can accurately reflect any network situation in reachability computation.

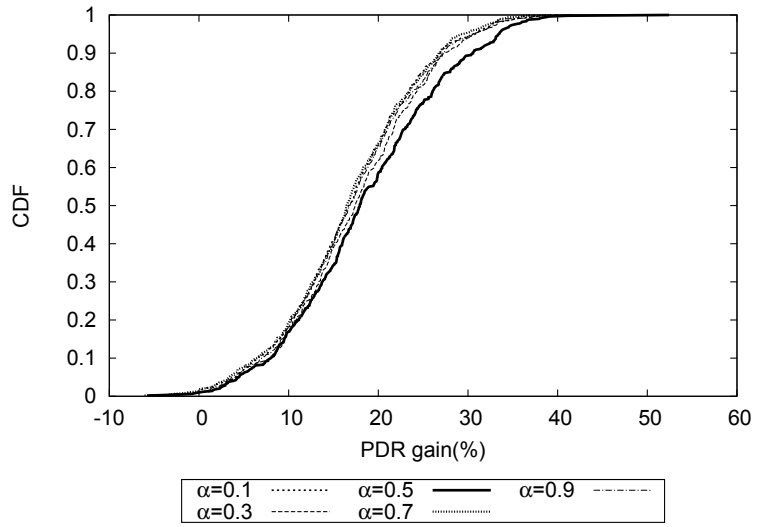
6.6.3.1 EWMA weight parameter (α)

After evaluating the T_{window}, another set of experiments is conducted for α. For this set of experiments similar 500 scenarios of varying speed were selected as discussed in Section 6.6.1. As a result of previous experiments for T_{window} its value is set to 1s, because the protocol gives maximum PDR gain for this window size and gathers results for different values of α i.e. 0.1, 0.3, 0.5, 0.7 and 0.9.

Fig. 6.9(a) shows the fitted curves (using second degree polynomial) for the PDR gain over AODV for varying speed of nodes with different values of α, similarly Fig. 6.9(b) shows the



(a) PDR in varying speed of the nodes



(b) CDF of PDR gain for different α

Figure 6.9: Performance of AODV-OPP+ unicast for different α

CDF for the PDR gain. First observation from the Fig. 6.9(a) is that the protocol outperforms AODV in all conditions whereas when speed of nodes is increasing PDR gain is getting lowered for $\alpha = 0.1, 0.3.0.7, 0.9$. Such behaviour shows that inappropriate weights assigned to the α can degrade protocol's performance. It is also clearly observed that when α is 0.5 the protocol has the highest PDR gain. Hence, when equal probability is assigned to both the components R_{old} and $R_{measured}$ then protocol can accurately measure the reachability and as a result achieves a higher PDR gain. Hence it can be concluded that higher PDR gain can be achieved when T_{window} is set to 1s and α is set to 0.5 for the hybrid protocols.

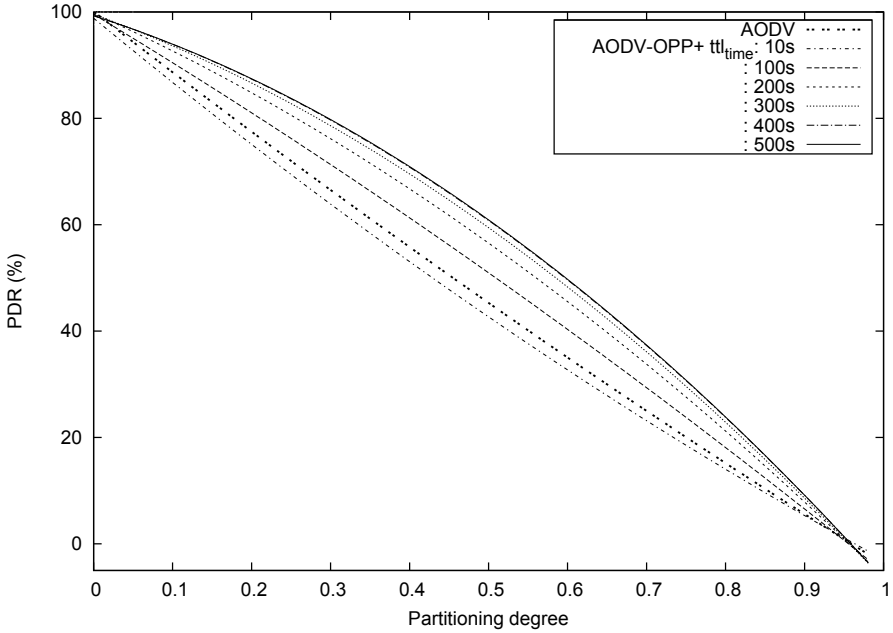


Figure 6.10: Performance of AODV-OPP+ unicast for different packet's ttl_{time}

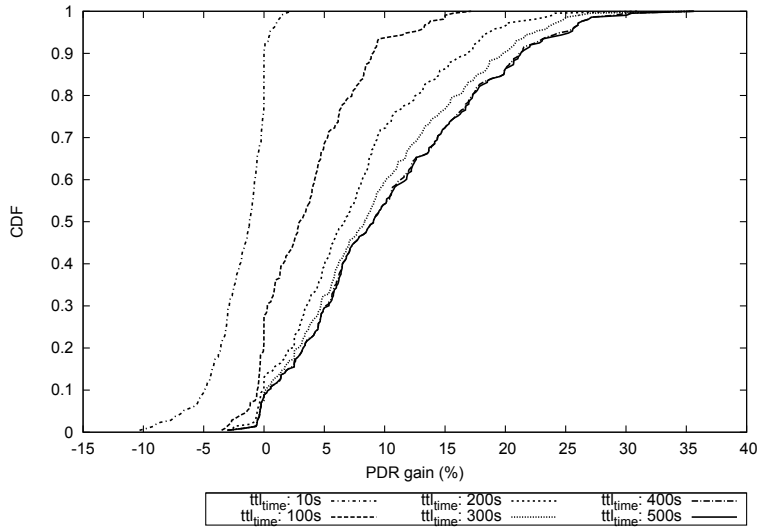
6.6.3.2 Buffered packet's ttl

For this set of experiments 300 mobility scenarios with varying PD values are used as described in Section 5.3.2.1 and the ns2 parameters shown in Table 6.2 except for packet's ttl_{time}. Each mobility pattern is simulated for 10 runs and average PDR is computed and it is repeated for different ttl_{time} of the packets i.e. 10s, 100s, 200s, 300s, 400s, 500s.

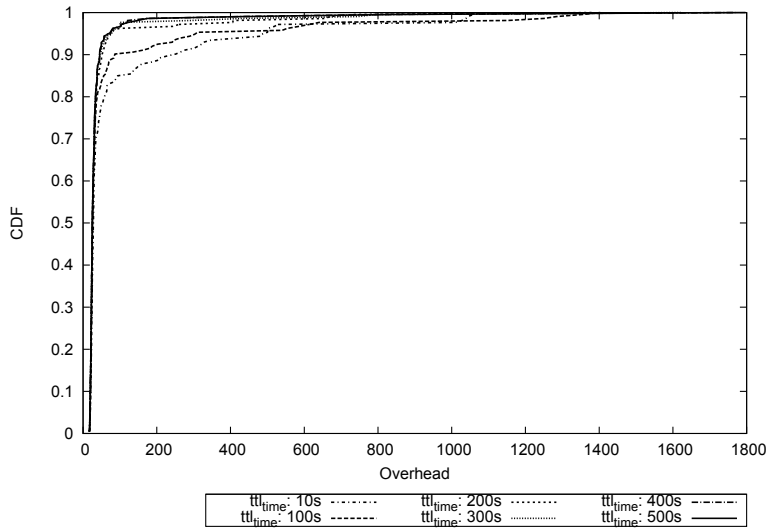
Fig. 6.10 shows curve-fitting plot (of 2nd degree polynomial) for all 300 PD mobility scenarios. The first observation is that as the PD value increases PDR decreases. As discussed in the Section 5.3.2.1 higher value of PD indicates the lower density of the nodes in the network. In low density there are smaller chances of nodes having connections with other nodes and this is the reason of lower PDR gain when PD increases.

The second observation is that even though hybrid forwarding is involved in AODV-OPP+ the protocol performance can degrade if packet's ttl_{time} is very short (as shown in Fig. 6.10 PDR gain when $ttl_{time} = 10s$). Such degraded performance is because packets were removed before they were successfully delivered to the destination. In other words, packets soon become stale and due to the hybrid feature of the protocol, the overhead generated in the network negatively impacts the PDR gain as shown in Fig. 6.11(a). The PDR gain is the highest when $ttl_{time} = 500s$, and in Fig. 6.11(b) overhead is maximum when $ttl_{time} = 10s$.

Overall this results show that the PDR gain is higher when packets live for a long time. As shown in the Fig. 6.10 for $ttl_{time} = 500s$ which is equivalent to the simulation time and



(a) CDF of PDR gain.



(b) CDF of overhead.

Figure 6.11: Performance analysis of AODV-OPP+ unicast for different packet's ttl_{time}

protocol achieves highest PDR gain at this value. The simulation verifies our analysis that for longer ttl_{time}, PDR gain is higher because buffered packets remain in the network and have higher chances to reach the destination.

6.6.4 Performance of AODV-OPP+: In varying PD value

For this set of experiments protocol parameters are selected as shown in Table 6.3 on the basis of results obtained from the previous experiments and used 300 mobility scenarios (varying in PD). For each protocol AODV, AODV-OPP+ broadcast and AODV-OPP+ unicast, average PDR is computed over 10 simulation runs for each PD value. Performance of these protocols is highlighted using fitted curves in the second degree polynomial form (hereafter they are labeled as curve in the figures) as shown in Fig. 6.12.

Measurement window size (T_{window})	1s
EWMA weight parameter (α)	0.5
Buffered packet's life (ttl_{time})	500

Table 6.3: AODV-OPP+ parameters

First observation from the results is that all three protocol's PDR decreases with increase in PD value. As it is already discussed this reduction in PDR value is due to lower node density in the network as a result lesser chances of having connection among nodes.

Second observation suggests that both AODV-OPP+ unicast and AODV-OPP+ broadcast outperform the original AODV across all different network densities. Also AODV-OPP+ unicast shows PDR slightly above the AODV-OPP+ broadcast for the entire PD range. Such behaviour confirms that AODV-OPP+ unicast design gets benefit due to one to one interaction between nodes where it verifies the presence of the potential forwarder before sending any data packets. Hence, it produces less overhead as compared to broadcast approach where in case of no route packet is broadcasted to all the one hop neighbour nodes. Therefore the overhead generated by both protocols is also analysed. As shown in Fig. 6.13 (a) increasing in PD value AODV-OPP+ broadcast generates significant overhead as compared to AODV-OPP+ unicast, because in the unicast design a packet is sent only to potential forwarders(s) whereas in the broadcast design packet is broadcast to all the neighbour nodes. From the CDF analysis of overhead as shown in Fig. 6.13 (b) AODV-OPP+ unicast generates approximately 25 packets over the entire PD range whereas AODV-OPP+ broadcast generates equivalent overhead for more than 25% network situations.

6.6.5 AODV-OPP+ and Spray-and-Wait

Having investigated the improvement of AODV-OPP+ broadcast and AODV-OPP+ unicast over AODV, next step is to carry out simulations to evaluate the performance of both of them against Spray-and-Wait (SAW) [82], a well known opportunistic communication protocol. SAW distributes buffered packets in a hop-by-hop manner. Whenever a SAW node X meets another node Y, it sends n of its buffered packets to node Y. Binary SAW [4] is used in our simulation; that is, half of the unseen packets are sent to a new node. Similar to most routing protocols, the HELLO interval is an important parameter in SAW. Fig. 6.14 shows the performance of SAW with different hello interval ranges over 300 PD mobility traces. When the HELLO interval is set to 0.75-1.25 s, which is default in AODV, SAW seems to

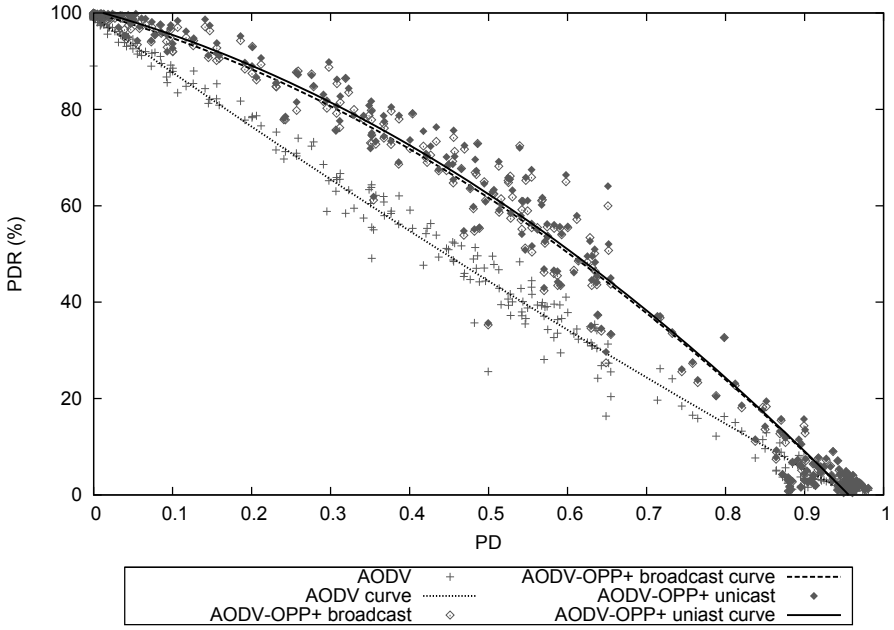
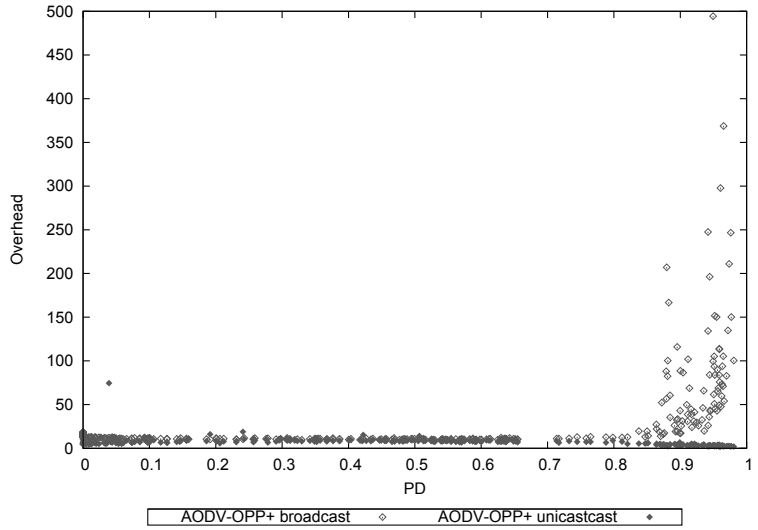


Figure 6.12: Performance of AODV-OPP+ broadcast and AODV-OPP+ unicast against AODV.

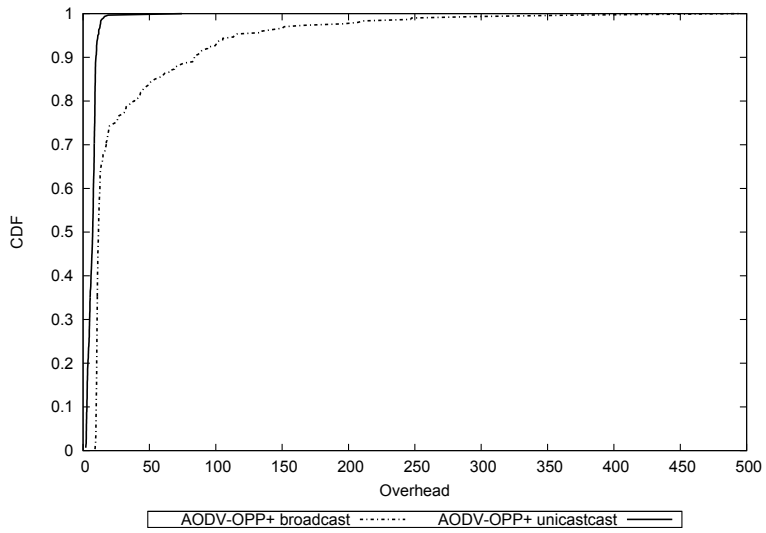
achieve relatively low PDR. Another observation is that PDR increases when the network becomes sparse and it decreases as the network density increases. The low PDR, when the network is dense, is caused by the high frequency of HELLO messages. These HELLO messages not only take up the transmission time for data packets - in a dense network they also create interference that stops other nodes from sending. This problem is reduced as the network becomes sparser, therefore an increase in PDR is also observed when the partitioning degree is medium. When the network becomes really sparse, PDR becomes low due to the lack of connections.

When the HELLO interval increases to 36-44 s (a default values used in the demo application that comes with the source code), it can be seen that SAW achieves higher PDR when the network is dense. Conjecture is that with such infrequent HELLO message exchanges, they have no impact on the data packet delivery. Also, a higher HELLO interval means an increase of delay in detecting new neighbours. The focus of this thesis is not on optimising SAW, therefore both results are used for the comparison.

Fig. 6.15 shows the performance comparison for all protocols (that is, combining the fitted curves from Fig. 6.12 and 6.14) including the result of the initial hybrid protocol proposed described in Chapter 4. It is also observed that the protocols which make use of end-to-end routes achieve significantly better performance when the network density is relatively high (partition degree 0-0.7), whereas SAW representing a DTN type of hop-by-hop only routing gets slight advantage over the original AODV when the network becomes really



(a) Overhead.



(b) CDF of overhead.

Figure 6.13: Overhead analysis over varying PD.

sparse (partitioning degree greater than 0.7).

6.6.6 Protocol performance in increasing load condition

As the mobile nodes need to forward buffered packets to their neighbours, overhead is one of the concerns in the proposed idea. In the initial approach AODV-OPP (presented in Chapter 4), buffered packets are sent to all the one-hop neighbours, which can result in significant increase in overhead. To trade off the PDR and overhead two variants of AODV-OPP+ are proposed that can reduce the overhead by selectively disseminating buffered packets to the neighbours that are more likely to have connections (or be part of a route) to the desired destination. As discussed overhead is defined as the number of additional packets forwarded in the network for every packet successfully delivered to the destination.

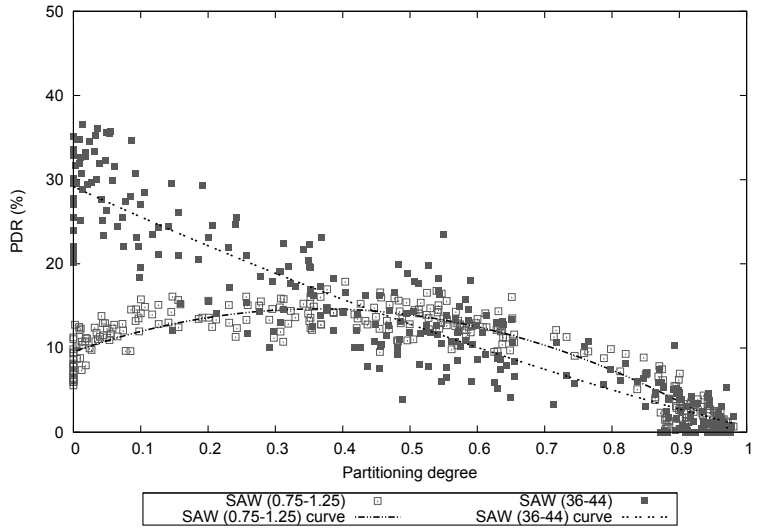


Figure 6.14: Performance of Spray and Wait for varying PD.

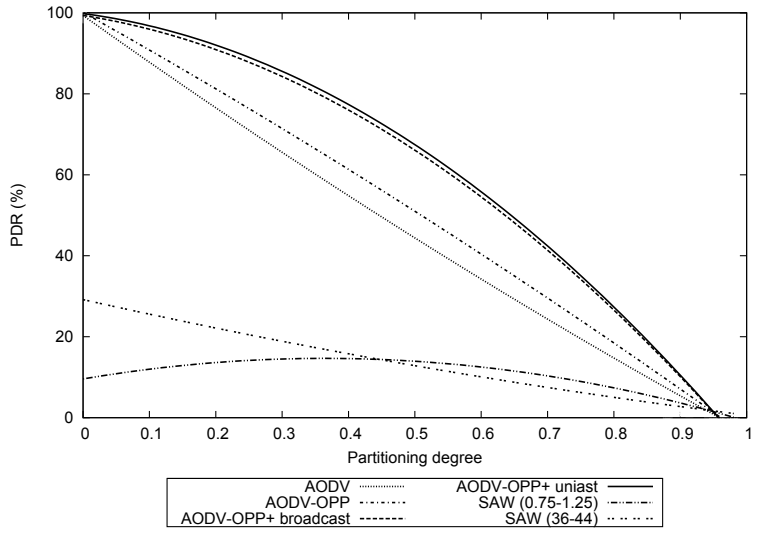


Figure 6.15: Performance comparison for all the protocols for varying PD.

It has been already discussed that AODV-OPP+ unicast is able to outperform AODV-OPP+ broadcast over the entire PD range. As the final step of the evaluation, another set of experiments is conducted using the San Francisco cab traces to emulate the network scenarios when network is heavily loaded. For this set of experiments the performance of AODV, AODV-OPP+ broadcast and AODV-OPP+ unicast is evaluated by increasing the number of flows in the network i.e., the number of connections per node from 1 to 20 for every 116 nodes. The purpose of this set of experiments is to identify which routing protocol can perform better in any network situation.

The result of this experiment is shown in Fig. 6.16, As illustrated in this figure as network load is increasing AODV-OPP+ unicast outperforms AODV-OPP+ broadcast and its PDR gain reaches up to 17%. Such PDR gain confirms that the even though AODV-OPP+ broad-

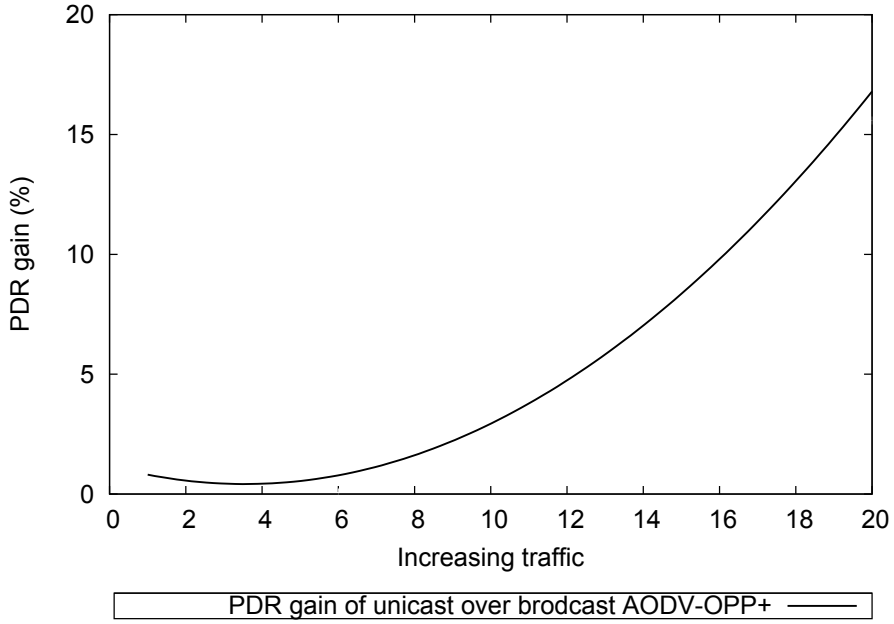


Figure 6.16: Performance of hybrid protocols in increasing load conditions.

cast uses only potential forwarders for data forwarding it still generates significant overhead in high load conditions. As a result it can be concluded that AODV-OPP+ unicast has superior performance as compared to the other protocols in any network situations.

6.7 Summary

This chapter presented a design of a new metric which is utilized to identify the potential forwarders in the network. Based on that metric two improved versions of the hybrid protocols i.e. AODV-OPP+ broadcast and AODV-OPP+ unicast are proposed in this chapter. Detailed descriptions of both protocols are provided along with the protocol comparisons. An analysis is also provided to discuss all the parameters that can affect the performance of both protocols.

At the end a systematic evaluation is provided that shows the superior performance of the AODV-OPP+ unicast in any network situation as compared to the traditional end-to-end routing protocol AODV and opportunistic routing protocol Spray-and-Wait (SAW). Next chapter discusses the conclusion and future directions.

CHAPTER 7

Conclusion and future directions

7.1 Conclusion

This thesis discussed context-aware integration of two different paradigms of routing in hybrid wireless mesh networks i.e. traditional end-to-end and opportunistic routing. Both the routing mechanisms are investigated in a broad range of network situations. Aim of such investigation was to find out their advantages and disadvantages. So that an integrated routing protocol can be designed that can combine the capabilities of both the routings so that the resultant protocol can perform in any network situation. In order to achieve this aim network context information is analysed that has impact on the protocol performance and which allows to dynamically switch between routing modes as per the network situation. On the basis of this analysis an initial model of a hybrid protocol is designed and developed that can be applied to either proactive or reactive end-to-end routing protocols. With the extensive set of systematic evaluation it is shown that the proposed hybrid version of the end-to-end routing protocol (reactive hybrid routing i.e. AODV-OPP and proactive hybrid routing i.e. OLSR-OPP) can significantly improve performance of its base protocol i.e. an end-to-end routing protocols (i.e. AODV and OLSR) in any network situation. It is worth noting that the performance enhancement in the hybrid protocol is due to utilizing partial paths in any network situation they were not utilized in any of the existing routing protocols such as an end-to-end or an opportunistic protocols.

The initially proposed hybrid protocol can dynamically switch between routing modes according to a network situation. According to its design principle if a packet can not be transmitted via a route due to link failure then it can be sent to all the one hop neighbours in a hope that they might get a route or directly encounter the destination. Although such a de-

sign principle leads to the maximum PDR gain but can also lead to the significant overhead in the network. Therefore various ways are investigated to further improve this hybrid protocol so that overhead can be minimized. One of the solutions was to allow only potential forwarders to participate in opportunistic forwarding, if required. Only those nodes having higher probability to reach the destination can forward data packets opportunistically. Hence this can maximize the PDR gain as well as lower overhead in the network. In order to achieve that a new reachability metric is proposed. It is the measure of the probability of meeting a node with the destination node via an end-to-end route or via direct contact. All the participating nodes compute and update reachability periodically to reflect the current network situation.

Two variants of the hybrid protocol are designed and developed using reachability i.e. AODV-OPP+ unicast & AODV-OPP+ broadcast and presented their detailed algorithms. Analysis is also presented based on the factors that can affect the protocol performance and on the selection of their optimum value so that the protocol can be tuned to get the maximum delivery ratio in any network situation.

Performance of the protocols is evaluated in various network conditions (using synthetic and real time mobility traces) and their performance is compared with AODV and Spray-and-Wait (SAW) using systematic evaluation techniques. It has been found that both variants i.e. AODV-OPP+ unicast and AODV-OPP+ broadcast outperform AODV and SAW. In addition, in increasing load conditions AODV-OPP+ unicast consistently outperforms AODV-OPP+ broadcast.

In conclusion, AODV-OPP+ unicast is the best hybrid protocol for dynamically switching between the routing modes as per the network situation. It also generates low overhead in the network.

7.2 Future directions

As it has been concluded that AODV-OPP+ unicast has superior performance as compared to the well known existing protocols. This section presents a discussion on how this research can be further extended.

In this hybrid protocol potential forwarders are selected based on the reachability that exploits the connectivity information among nodes. As a future work other context informa-

tion can also be considered so that protocol can avoid situations when potential forwarders are overloaded due to being selected by many nodes for routing their packets. Example of those context information can be available buffer size, energy level of nodes etc.

Applications requiring reliable communication prefer TCP connections. TCP provides reliable and ordered delivery of the data. If for each packet it does not receive "ACK" on time, it will shrink its contention window size, but will increase it if it does receive "ACK". In NS2, the default TCP implementation is TCP Tahoe. Whenever a packet is lost TCP Tahoe goes into slow start (contention window size reduced to 1). If TCP Tahoe receives "ACK" it increases the contention window exponentially. When the contention window(cwnd_) reaches to "1" TCP assumes link break. In our attempt to create the hybrid routing protocol, whenever application recognises that the TCP stopped sending packets due to link break, it will release TCP connection and application can switches to sending UDP packets. UDP can be modified to incorporate reliability [87]. The proposed hybrid protocol already has anti-packets. In the presented approach anti-packets are used to release packets from the buffer. To incorporate reliability these packets can be modified. So that they can be used as acknowledgements to support reliability in the hybrid protocol.

This hybrid protocol is based on the assumption that all the nodes are trustworthy. Hence as a future work another feature can be added to the hybrid protocol so that it can allow only trustworthy nodes in the routing by ignoring any malicious nodes.

This research can be extended to include sub-classes of real-time traffic to evaluate, (i) the performance gain by removing queued packets that are no longer meeting the requirement of maximum delay for sub-class of traffic to which they belong, (ii) overhead introduced by managing several classes of time window for traffic sub-classes.

References

[1] Bonnmotion. http://net.cs.uni-bonn.de/wg/cs/applications/bonnmotion/.

[2] Cabsplotting. http://cabsplotting.org.

[3] The san francisco cab traces. http://cabspotting.org.

[4] Spray and wait ns2 implementation. http://www.ida.liu.se/labs/rtslab/code/LAROD-LoDiS/.

[5] F. Bai, N. Sadagopan, and A. Helmy. Important: A framework to systematically analyze the impact of mobility on performance of routing protocols for adhoc networks. In *INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications. IEEE Societies*, volume 2, pages 825–835. IEEE, 2003.

[6] C. Bettini, O. Brdiczka, K. Henricksen, J. Indulska, D. Nicklas, A. Ranganathan, and D. Riboni. A survey of context modelling and reasoning techniques. volume 6, pages 161–180. Elsevier, 2010.

[7] S. Biswas and R. Morris. Exor: opportunistic multi-hop routing for wireless networks. In *ACM SIGCOMM Computer Communication Review*, volume 35, pages 133–144. ACM, 2005.

[8] C. Boldrini, M. Conti, J. Jacopini, and A. Passarella. Hibop: a history based routing protocol for opportunistic networks. In *World of Wireless, Mobile and Multimedia Networks, 2007. WoWMoM 2007. IEEE International Symposium on a*, pages 1–12. IEEE, 2007.

[9] C. Boldrini, M. Conti, and A. Passarella. Impact of social mobility on routing protocols for opportunistic networks. In *World of Wireless, Mobile and Multimedia Networks, 2007. WoWMoM 2007. IEEE International Symposium on a*, pages 1–6. IEEE, 2007.

[10] C. Boldrini, M. Conti, and A. Passarella. Autonomic behaviour of opportunistic network routing. volume 1, pages 122–147. Inderscience, 2008.

[11] C. Boldrini, M. Conti, and A. Passarella. Contentplace: social-aware data dissemination in opportunistic networks. In *Proceedings of the 11th international symposium on Modeling, analysis and simulation of wireless and mobile systems*, pages 203–210. ACM, 2008.

[12] C. Boldrini, M. Conti, and A. Passarella. Context and resource awareness in opportunistic network data dissemination. In *World of Wireless, Mobile and Multimedia Networks, 2008. WoWMoM 2008. 2008 International Symposium on a*, pages 1–6. IEEE, 2008.

[13] C. Boldrini, M. Conti, and A. Passarella. Modelling data dissemination in opportunistic networks. In *Proceedings of the third ACM workshop on Challenged networks*, pages 89–96. ACM, 2008.

[14] C. Boldrini, M. Conti, and A. Passarella. User-centric mobility models for opportunistic networking. In *Bio-Inspired Computing and Communication*, pages 255–267. Springer, 2008.

[15] R. Bruno, M. Conti, and E. Gregori. Mesh networks: commodity multihop ad hoc networks. *Communications Magazine, IEEE*, 43(3):123–131, March 2005.

[16] J. Burgess, B. Gallagher, D. Jensen, and B. N. Levine. Maxprop: Routing for vehicle-based disruption-tolerant networks. In *INFOCOM*, volume 6, pages 1–11, 2006.

[17] S. Chachulski, M. Jennings, S. Katti, and D. Katabi. *Trading structure for randomness in wireless opportunistic routing*, volume 37. ACM, 2007.

[18] I. Chakeres and C. P. Dynamic manet on-demand (dymo) routing. June 2008.

[19] I. D. Chakeres and E. M. Belding-Royer. The utility of hello messages for determining link connectivity. In *Wireless Personal Multimedia Communications, 2002. The 5th International Symposium on*, volume 2, pages 504–508. IEEE, 2002.

[20] G. Chen, D. Kotz, et al. A survey of context-aware mobile computing research. Technical report, Technical Report TR2000-381, Dept. of Computer Science, Dartmouth College, 2000.

[21] L.-J. Chen, C.-H. Yu, T. Sun, Y.-C. Chen, and H.-h. Chu. A hybrid routing approach for opportunistic networks. In *Proceedings of the 2006 SIGCOMM workshop on Challenged networks*, pages 213–220. ACM, 2006.

[22] P. Chimento and J. Ishac. Rfc 5136, defining network capacity. 2008.

[23] T. Clausen and P. Jacquet. Optimized link state routing protocol (olsr). October 2003.

[24] M. Conti, S. Giordano, M. May, and A. Passarella. From opportunistic networks to opportunistic computing. *Communications Magazine, IEEE*, 48(9):126–139, 2010.

[25] M. Conti and M. Kumar. Opportunities in opportunistic computing. volume 43, pages 42–50. Institute of Electrical and Electronics Engineers, Inc., 3 Park Avenue, 17 th Fl New York NY 10016-5997 USA, 2010.

[26] E. Davies, F. Consulting, and S. Grasic. Dtn research group a. lindgren internet-draft sics intended status: Experimental a. doria expires: November 23, 2012 consultant. 2012.

[27] D. S. De Couto, D. Aguayo, J. Bicket, and R. Morris. A high-throughput path metric for multi-hop wireless routing. volume 11, pages 419–434. Kluwer Academic Publishers, 2005.

[28] L. Delosiéres and S. Nadjm-Tehrani. Batman store-and-forward: the best of the two worlds. In *Proc. of PerNEM 2012 (PerCom workshop)*, pages 727–733, Lugano, Switzerland, March 2012.

[29] A. K. Dey, G. D. Abowd, and D. Salber. A conceptual framework and a toolkit for supporting the rapid prototyping of context-aware applications. volume 16, pages 97–166. L. Erlbaum Associates Inc., 2001.

[30] R. Draves, J. Padhye, and B. Zill. Routing in multi-radio, multi-hop wireless mesh networks. In *Proceedings of the 10th Annual International Conference on Mobile Computing and Networking*, MobiCom '04, pages 114–128, New York, NY, USA, 2004. ACM.

[31] Firetide. http://www.firetide.com/innercontent.aspx?taxid=16&id=3216.

[32] R. Groenevelt, P. Nain, and G. Koole. The message delay in mobile ad hoc networks. volume 62, pages 210–228. Elsevier, 2005.

[33] Z. J. Haas, M. R. Pearlman, and P. Samar. The zone routing protocol (zrp) for ad hoc networks. 2002.

[34] Z. J. Haas and T. Small. A new networking model for biological applications of ad hoc sensor networks. volume 14, pages 27–40. IEEE, 2006.

[35] B. Han, P. Hui, V. Kumar, M. Marathe, G. Pei, and A. Srinivasan. Cellular traffic offloading through opportunistic communications: a case study. In *Proceedings of the 5th ACM workshop on Challenged networks*, pages 31–38. ACM, 2010.

[36] B. Han, P. Hui, V. Kumar, M. Marathe, J. Shao, and A. Srinivasan. Mobile data offloading through opportunistic communications and social participation. Number 99, pages 1–1. IEEE, 2011.

[37] K. Henricksen and J. Indulska. Modelling and using imperfect context information. In *Pervasive Computing and Communications Workshops, 2004. Proceedings of the Second IEEE Annual Conference on*, pages 33–37. IEEE, 2004.

[38] K. Henricksen and J. Indulska. A software engineering framework for context-aware pervasive computing. In *Pervasive Computing and Communications, 2004. PerCom 2004. Proceedings of the Second IEEE Annual Conference on*, pages 77–86. IEEE, 2004.

[39] K. Henricksen and J. Indulska. Developing context-aware pervasive computing applications: Models and approach. volume 2, pages 37–64. Elsevier, 2006.

[40] K. Henricksen, J. Indulska, and A. Rakotonirainy. Modeling context information in pervasive computing systems. In *Pervasive Computing*, pages 167–180. Springer, 2002.

[41] K. Henricksen, J. Indulska, and A. Rakotonirainy. Generating context management infrastructure from high-level context models. In *In 4th International Conference on Mobile Data Management (MDM)-Industrial Track*. Citeseer, 2003.

[42] M. Heusse, F. Rousseau, G. Berger-Sabbatel, and A. Duda. Performance anomaly of 802.11 b. In *INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications. IEEE Societies*, volume 2, pages 836–843. IEEE, 2003.

[43] R. Hou, K.-S. Lui, F. Baker, and J. Li. Hop-by-hop routing in wireless mesh networks with bandwidth guarantees. *Mobile Computing, IEEE Transactions on*, 11(2):264–277, 2012.

[44] P. Hu, J. Indulska, and R. Robinson. An autonomic context management system for pervasive computing. In *Pervasive Computing and Communications, 2008. PerCom 2008. Sixth Annual IEEE International Conference on*, pages 213–223. IEEE, 2008.

[45] J. Indulska and P. Sutton. Location management in pervasive systems. In *Proceedings of the Australasian Information Security Workshop Conference on ACSW Frontiers 2003 -*

Volume 21, ACSW Frontiers '03, pages 143–151, Darlinghurst, Australia, Australia, 2003. Australian Computer Society, Inc.

[46] P. Jacquet, A. Laouiti, P. Minet, and L. Viennot. Performance of multipoint relaying in ad hoc mobile routing protocols. In *Networking 2002*, Pisa, Italy, 2002.

[47] N. Javaid, A. Javaid, I. A. Khan, and K. Djouani. Performance study of etx based wireless routing metrics. In *Computer, Control and Communication, 2009. IC4 2009. 2nd International Conference on*, pages 1–7. IEEE, 2009.

[48] D. B. Johnson and D. A. Maltz. Dynamic source routing in ad hoc wireless networks. pages 153–179. Springer, 1996.

[49] M. Kim, D. Kotz, and S. Kim. Extracting a mobility model from real user traces. In *INFOCOM*, volume 6, pages 1–13, 2006.

[50] V. Kisara. A new routing metric for wireless mesh networks. 2010.

[51] L. B. Koralov, Y. G. Sinai, et al. *Theory of probability and random processes*. Springer-Verlag Berlin Heidelberg, 2007.

[52] C. Kretschmer, S. Ruhrup, and C. Schindelhauer. Dt-dymo: delay-tolerant dynamic manet on-demand routing. In *Distributed Computing Systems Workshops, 2009. ICDCS Workshops' 09. 29th IEEE International Conference on*, pages 493–498. IEEE, 2009.

[53] J. Lakkakorpi, M. Pitkänen, and J. Ott. Adaptive routing in mobile opportunistic networks. In *Proc. of MSWiM2010*, pages 101–109, Bodrum, Turkey, October 2010.

[54] H. Li, Y. Cheng, and C. Zhou. Multi-hop effective bandwidth based routing in multi-radio wireless mesh networks. In *Global Telecommunications Conference, 2008. IEEE GLOBECOM 2008. IEEE*, pages 1–5. IEEE, 2008.

[55] Y. Li, W. Chen, and Z.-L. Zhang. Optimal forwarder list selection in opportunistic routing. In *Mobile Adhoc and Sensor Systems, 2009. MASS'09. IEEE 6th International Conference on*, pages 670–675. IEEE, 2009.

[56] Z. Li and H. Shen. Sedum: Exploiting social networks in utility–based distributed routing for dtns. volume 62, pages 83–97. IEEE, 2013.

[57] A. Lindgren, A. Doria, and O. Schelén. Probabilistic routing in intermittently connected networks. *ACM SIGMOBILE Mobile Computing and Communications Review*, 7(3):19–20, 2003.

[58] M.-H. Lu, P. Steenkiste, and T. Chen. Design, implementation and evaluation of an efficient opportunistic retransmission protocol. In *Proceedings of the 15th annual international conference on Mobile computing and networking*, pages 73–84. ACM, 2009.

[59] X. Ma and X. Chen. Performance analysis of ieee 802.11 broadcast scheme in ad hoc wireless lans. volume 57, pages 3757–3768. IEEE, 2008.

[60] K. Majumder, S. Sarkar, and S. Ray. Analysis of qos parameters for dsdv and dsr in hybrid scenario. In *Electronic System Design (ISED), 2010 International Symposium on*, pages 219–224. IEEE, 2010.

[61] A. Manjeshwar and D. P. Agrawal. Apteen: A hybrid protocol for efficient routing and comprehensive information retrieval in wireless sensor networks. In *ipdps*, volume 2, page 48, 2002.

[62] M. Mishra and A. Sahoo. A contention window based differentiation mechanism for providing qos in wireless lans. In *Information Technology, 2006. ICIT'06. 9th International Conference on*, pages 72–76. IEEE, 2006.

[63] M. Musolesi and C. Mascolo. Car: context-aware adaptive routing for delay-tolerant mobile networks. *Mobile Computing, IEEE Transactions on*, 8(2):246–260, 2009.

[64] ncape. http://allafrica.com/stories/201404110366.html.

[65] S. C. Nelson, M. Bakht, and R. Kravets. Encounter-based routing in dtns. In *INFOCOM 2009, IEEE*, pages 846–854. IEEE, 2009.

[66] L. Nguyen, R. Beuran, and Y. Shinoda. A load-aware routing metric for wireless mesh networks. In *Computers and Communications, 2008. ISCC 2008. IEEE Symposium on*, pages 429–435. IEEE, 2008.

[67] R. OLSR. 3626.

[68] J. Ott, D. Kutscher, and C. Dwertmann. Integrating dtn and manet routing. In *Proc. of ACM SIGCOMM workshop CHANTS*, 2006.

[69] L. Pelusi, A. Passarella, and M. Conti. Opportunistic networking: data forwarding in disconnected mobile ad hoc networks. *Communications Magazine, IEEE*, 44(11):134–141, 2006.

[70] C. Perkins, E. Belding-Royer, and S. Das. Ad hoc on-demand distance vector (aodv) routing. IETF. RFC 3561, July 2003.

[71] C. Perkins, E. Belding-Royer, S. Das, et al. Rfc 3561-ad hoc on-demand distance vector (aodv) routing. *Internet RFCs*, pages 1–38, 2003.

[72] C. Perkins, E. Royer, S. Das, and M. Marina. Performance comparison of two on-demand routing protocols for ad hoc networks. volume 8, pages 16–28, 2001.

[73] A. Petz, A. Bednarczyk, N. Paine, D. Stovall, and C. Julien. Madman: A middleware for delay-tolerant mobile ad-hoc networks. Technical report, Technical Report TR-UTEDGE-2010-010, 2010.

[74] C. Raffelsberger and H. Hellwagner. A hybrid manet-dtn routing scheme for emergency response scenarios. PerNEM 2013: Third International Workshop on Pervasive Networks for Emergency Management 2013-Program.

[75] F. J. Ros. Um-olsr ns2 implementation. http://masimum.inf.um.es/fjrm/?page_id=116.

[76] C. Sengul, A. C. Viana, and A. Ziviani. A survey of adaptive services to cope with dynamics in wireless self-organizing networks. volume 44, pages 23:1–23:35, New York, NY, USA, Sept. 2012. ACM.

[77] R. C. Shah, S. Wietholter, A. Wolisz, and J. M. Rabaey. When does opportunistic routing make sense? In *Pervasive Computing and Communications Workshops, 2005. PerCom 2005 Workshops. Third IEEE International Conference on*, pages 350–356. IEEE, 2005.

[78] G. Sharma, R. Mazumdar, and N. B. Shroff. Delay and capacity trade-offs in mobile ad hoc networks: A global perspective. volume 15, pages 981–992. IEEE Press, 2007.

[79] T. Small and Z. J. Haas. Resource and performance tradeoffs in delay-tolerant wireless networks. In *Proc. of SIGCOMM WDTN*, pages 260–267, Philadelphia, PA, USA, 2005. ACM.

[80] L. Song and D. F. Kotz. Evaluating opportunistic routing protocols with large realistic contact traces. In *Proceedings of the second ACM workshop on Challenged networks*, pages 35–42. ACM, 2007.

[81] T. Spyropoulos, K. Psounis, and C. S. Raghavendra. Single-copy routing in intermittently connected mobile networks. In *Sensor and Ad Hoc Communications and Networks, 2004. IEEE SECON 2004. 2004 First Annual IEEE Communications Society Conference on*, pages 235–244. IEEE, 2004.

[82] T. Spyropoulos, K. Psounis, and C. S. Raghavendra. Spray and wait: an efficient routing scheme for intermittently connected mobile networks. In *Proc. of SIGCOMM WDTN*, pages 252–259, Philadelphia, PA, USA, 2005. ACM.

[83] T. Spyropoulos, K. Psounis, and C. S. Raghavendra. Performance analysis of mobility-assisted routing. In *Proceedings of the 7th ACM international symposium on Mobile ad hoc networking and computing*, pages 49–60. ACM, 2006.

[84] W. Stevens, M. Allman, and V. Paxson. Tcp congestion control. *Consultant*, 1999.

[85] X. Tie, A. Venkataramani, and A. Balasubramanian. R3: robust replication routing in wireless networks with diverse connectivity characteristics. In *Proceedings of the 17th annual international conference on Mobile computing and networking*, pages 181–192. ACM, 2011.

[86] A. Vahdat, D. Becker, et al. Epidemic routing for partially connected ad hoc networks. Technical report, Technical Report CS-200006, Duke University, 2000.

[87] J. Whitbeck and V. Conan. Hymad: Hybrid dtn-manet routing for dense and highly dynamic wireless networks. *Comput. Commun.*, 33(13):1483–1492, Aug. 2010.

[88] W. Yin, P. Hu, J. Indulska, and K. Bialkowski. Performance of mac80211 rate control mechanisms. In *Proceedings of the 14th ACM international conference on Modeling, analysis and simulation of wireless and mobile systems*, pages 427–436. ACM, 2011.

[89] X. Zhang, G. Neglia, J. Kurose, and D. Towsley. Performance modeling of epidemic routing. volume 51, pages 2867–2891. Elsevier, 2007.

[90] J. Zhou, C. Guo, P. Pawelczak, and I. Niemegeers. Adaptable link quality estimation for multi data rate communication networks. In *Vehicular Technology Conference, 2009. VTC Spring 2009. IEEE 69th*, pages 1–5. IEEE, 2009.

[91] H. Zhu and K. Lu. Resilient opportunistic forwarding: Issues and challenges. In *Military Communications Conference, 2007. MILCOM 2007. IEEE*, pages 1–7. IEEE, 2007.