Multipath routing and QoS provisioning in mobile ad hoc networks
Li, Xuefei

For additional information about this publication click this link.
http://qmro.qmul.ac.uk/jspui/handle/123456789/3814

Information about this research object was correct at the time of download; we occasionally make corrections to records, please therefore check the published record when citing. For more information contact scholarlycommunications@qmul.ac.uk
Multipath Routing and QoS Provisioning
in Mobile Ad hoc Networks

Xuefei Li

Submitted for the degree of Doctor of Philosophy

Department of Electronic Engineering
Queen Mary, University of London

April 2006
To my family
Abstract

A Mobile Ad Hoc Networks (MANET) is a collection of mobile nodes that can communicate with each other using multihop wireless links without utilizing any fixed based-station infrastructure and centralized management. Each mobile node in the network acts as both a host generating flows or being destination of flows and a router forwarding flows directed to other nodes.

Future applications of MANETs are expected to be based on all-IP architecture and be capable of carrying multitude real-time multimedia applications such as voice and video as well as data. It is very necessary for MANETs to have an efficient routing and quality of service (QoS) mechanism to support diverse applications.

This thesis proposes an on-demand Node-Disjoint Multipath Routing protocol (NDMR) with low broadcast redundancy. Multipath routing allows the establishment of multiple paths between a single source and single destination node. It is also beneficial to avoid traffic congestion and frequent link breaks in communication because of the mobility of nodes. The important components of the protocol, such as path accumulation, decreasing routing overhead and selecting node-disjoint paths, are explained. Because the new protocol significantly reduces the total number of Route Request packets, this results in an increased delivery ratio, smaller end-to-end delays for data packets, lower control overhead and fewer collisions of packets.

Although NDMR provides node-disjoint multipath routing with low route overhead in MANETs, it is only a best-effort routing approach, which is not enough to support QoS. DiffServ is a standard approach for a more scalable way to achieve QoS in any IP network and could potentially be used to provide QoS in MANETs because it minimises the need for signalling. However, one of the biggest drawbacks of DiffServ is that the QoS provisioning is separate from the routing process. This thesis presents a Multipath QoS Routing protocol for
supporting DiffServ (MQRD), which combines the advantages of NDMR and DiffServ. The protocol can classify network traffic into different priority levels and apply priority scheduling and queuing management mechanisms to obtain QoS guarantees.
Acknowledgement

First and foremost, I would like to express my sincere gratitude to my supervisor Professor Laurie Cuthbert. I thank him for leading me into this exciting research area and giving me help whenever I need. I am grateful for his constant encouragement, support, supervision and guidance during my PhD at Queen Mary, University of London.

I would like to thank John Bigham, John Schormans, Yue Chen, Eliane Bodanese, Athen Ma, Eric Scharf and Karen Shoop for giving me help and valuable suggestions and comments during my PhD. I would also like to thank the kind help from administrative and technical staff, Melissa Yeo, Lynda Rolfe, Michele Pringle, Kok Ho Huen, George Cunliffe, Julian Rodaway and many others during the studies.

I would like to thank my friends at QMUL for the good time and the friendship. In particular, my thanks to Lin Du, Yulin Wang, Ning Lu, Yapeng Wang, Xuan Jin, Na Yao, Xiao Lin, Pengcheng Liang, Weizhi Luo and many others.

Finally, I would like to thank the support and understanding from my dear wife Ji Qi, my parents, my parents-in-law, my brothers and my sister.
Table of Contents

Abstract ................................................................................................... iii

Acknowledgement ................................................................................... v

Table of Contents ................................................................................... vi

List of Figures .......................................................................................... x

Glossary ................................................................................................. xii

Chapter 1  Introduction ........................................................................ 1
  1.1 Research Motivation ...................................................................... 1
  1.2 Research Contribution ............................................................... 3
  1.3 Author’s Publications .................................................................. 4
  1.4 Thesis Organization .................................................................... 5

Chapter 2  Routing in Mobile Ad hoc Networks ................................. 7
  2.1 Ad hoc Networks ......................................................................... 7
  2.2 Routing Classification in Ad Hoc Networks ................................. 10
    2.2.1 Proactive versus Reactive Approaches ................................. 11
    2.2.2 Clustering and Hierarchical Routing ..................................... 12
  2.3 Review of Ad hoc Proactive Routing Protocols ............................. 12
    2.3.1 Dynamic Destination-Sequenced Distance-Vector Routing .... 13
    2.3.2 The Wireless Routing Protocol (WRP) ................................. 14
    2.3.3 Global State Routing (GSR) ................................................ 15
    2.3.4 Fisheye State Routing (FSR) ............................................. 16
  2.4 Review of Ad hoc Reactive Routing Protocols ............................. 16
    2.4.1 Ad Hoc On-demand Distance Vector Routing (AODV) ....... 16
    2.4.2 Dynamic Source Routing (DSR) ......................................... 18
    2.4.3 Cluster based Routing Protocol (CBRP) ............................. 20
2.5 Ad Hoc On-demand Multipath Routing Protocols ........................................ 22
  2.5.1 Ad hoc On-demand Multipath Distance Vector (AOMDV) .................. 23
  2.5.2 Split Multipath Routing (SMR) .......................................................... 24
  2.5.3 Multipath Source routing (MSR) ........................................................ 25
  2.5.4 Ad hoc On-demand Distance Vector Multipath Routing .................. 26

2.6 Problem with Current Multipath Routing Protocols ................................. 27

Chapter 3 Node-Disjoint Multipath Routing with Low Overhead...29

  3.1 Introduction .......................................................................................... 29
  3.2 Node-Disjoint Multipath Routing (NDMR) ........................................... 30
    3.2.1 Protocol Message Formats ............................................................. 30
    3.2.2 Data Structures ............................................................................ 31
    3.2.3 Route Discovery ......................................................................... 33
    3.2.4 Route Maintenance ................................................................... 43
  3.3 Simulation Implementation ................................................................. 44
    3.3.1 Network Model ........................................................................... 44
    3.3.2 Node Model .............................................................................. 45
    3.3.3 Application Process Model ....................................................... 47
    3.3.4 Node-Disjoint Multipath Routing Process Model ....................... 49
    3.3.5 Medium Access Control Model ............................................... 52
    3.3.6 Mobility Process Model ........................................................... 52
  3.4 Simulation Validation .......................................................................... 54
  3.5 Simulation Environment Model .......................................................... 57
    3.5.1 Mobility and Traffic model ...................................................... 57
    3.5.2 Performance Metrics ............................................................... 58
  3.6 Simulation Results .............................................................................. 59
    3.6.1 Varying Velocity ...................................................................... 59
    3.6.2 Varying Number of Sources .................................................... 66
    3.6.3 Performance Comparisons between NDMR and AOMDV ....... 71
  3.7 Summary .............................................................................................. 74

Chapter 4 Multipath QoS Routing for supporting DiffServ ........... 75
4.1 Introduction .......................................................................................... 75
4.2 QoS Models .......................................................................................... 77
  4.2.1 Integrated Services (IntServ) .................................................... 77
  4.2.2 Differentiated Services (DiffServ) ........................................... 79
4.3 Existing QoS Models for MANETs ..................................................... 81
  4.3.1 Ad hoc QoS on-demand routing (AQOR) ................................ 81
  4.3.2 Flexible QoS Model for MANETs (FQMM) ......................... 82
  4.3.3 SWAN ...................................................................................... 82
  4.3.4 INSIGNIA ................................................................................ 83
4.4 A New QoS Routing Protocol: MQRD ................................................ 84
  4.4.1 Integration of NDMR and DiffServ ......................................... 85
  4.4.2 QoS and Resource Management of MQRD............................. 85
  4.4.3 Load Balance and Congestion Avoidance.............................. 87
4.5 Simulation Model ................................................................................. 88
  4.5.1 Mobility and Traffic model ...................................................... 88
  4.5.2 Performance Metrics ............................................................... 88
4.6 Simulation Results ................................................................................ 89
4.7 Summary .............................................................................................. 92

Chapter 5  Distributed Cross-Layer QoS ................................................. 93
  5.1 Introduction .......................................................................................... 93
  5.2 Medium Access Control .................................................................... 94
    5.2.1 Classification of Medium Access Control Schemes .............. 94
    5.2.2 IEEE 802.11 Medium Access Mechanisms ............................. 95
  5.3 Distributed Cross-Layer QoS (DCLQ) ............................................. 104
    5.3.1 QoS challenges .................................................................... 104
    5.3.2 Overview of DCLQ Architecture ........................................... 106
    5.3.3 Distributed QoS MAC ........................................................... 109
    5.3.4 QoS-aware Priority Scheduler ............................................. 111
    5.3.5 Node-disjoint multipath routing .......................................... 112
    5.3.6 Admission Control ............................................................... 113
List of Figures

Figure 2.1 Illustration of the infrastructure network model ................................. 8
Figure 2.2 Illustration of the infrastructure-less networks ............................... 8
Figure 2.3 Route discovery in AODV .............................................................. 17
Figure 2.4 Route discovery in DSR ................................................................. 19
Figure 3.1 The flow chart of initiating a discovery process ............................... 34
Figure 3.2 Path Accumulation in NDMR ......................................................... 35
Figure 3.3 Shortest Routing Hops of Loop-free Paths .................................... 37
Figure 3.4 Flowchart of reducing broadcast routing overhead ....................... 38
Figure 3.5 Route Request Process with Low Overhead ................................. 39
Figure 3.6 Node-Disjoint Paths .................................................................... 40
Figure 3.7 Flowchart of selecting node-disjoint paths ................................... 41
Figure 3.8 Flowchart of processing an incoming RREP packet ...................... 42
Figure 3.9 Flowchart of processing data packets .......................................... 43
Figure 3.10 Network Model ......................................................................... 45
Figure 3.11 Node Model ............................................................................. 46
Figure 3.12 Application Process Model ......................................................... 48
Figure 3.13 NDMR Routing Process Model .................................................. 50
Figure 3.14 Mobility Process Model ............................................................... 53
Figure 3.15 Packet Delivery Ratio vs. Velocity ............................................. 61
Figure 3.16 Average Delay vs. Velocity ......................................................... 63
Figure 3.17 Normalized Routing Load vs. Velocity ...................................... 65
Figure 3.18 Packet Delivery Ratio vs. Number of Sources ................................ 68
Figure 3.19 Average Delay vs. Number of Sources................................. 69
Figure 3.20 Normalized Routing Load vs. Number of Sources............... 70
Figure 3.21 Performance Comparisons between NDMR and AOMDV ....... 73
Figure 4.1 TOS and DSCP + ECN ............................................................... 79
Figure 4.2 Priority Scheduler ..................................................................... 86
Figure 4.3 Packet Delivery Ratio of NDMR ................................................. 89
Figure 4.4 Packet Delivery Ratio of MQRD ................................................ 90
Figure 4.5 Average Delay of NDMR .......................................................... 91
Figure 4.6 Average Delay of MQRD .......................................................... 91
Figure 5.1 Basic Access Scheme ............................................................... 97
Figure 5.2 Hidden and Exposed terminals ............................................... 98
Figure 5.3 RTS/CTS access scheme .......................................................... 100
Figure 5.4 EDCF channel access scheme with different priority .......... 103
Figure 5.5 DCLQ architecture ................................................................. 107
Figure 5.6 Packet Delivery Ratio ............................................................. 119
Figure 5.7 Average end-to-end delay ....................................................... 120
Figure 5.8 Jitter ...................................................................................... 121
## Glossary

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AF</td>
<td>Assured Forwarding</td>
</tr>
<tr>
<td>AIFS</td>
<td>Arbitration IFS</td>
</tr>
<tr>
<td>AODV</td>
<td>Ad hoc On-demand Distance Vector</td>
</tr>
<tr>
<td>AODVM</td>
<td>Ad hoc On-demand Distance Vector Multipath Routing</td>
</tr>
<tr>
<td>AOMDV</td>
<td>Ad hoc On-demand Multipath Distance Vector</td>
</tr>
<tr>
<td>AP</td>
<td>Access Point</td>
</tr>
<tr>
<td>AQOR</td>
<td>Ad hoc QoS On-demand Routing</td>
</tr>
<tr>
<td>BE</td>
<td>Best Effort</td>
</tr>
<tr>
<td>CBR</td>
<td>Constant Bit Rate</td>
</tr>
<tr>
<td>CSMA</td>
<td>Carrier Sense Multiple Access</td>
</tr>
<tr>
<td>CSMA/CA</td>
<td>Carrier Sense Multiple Access/Collision Avoidance</td>
</tr>
<tr>
<td>CSMA/CD</td>
<td>Carrier Sense Multiple Access/Collision Detection</td>
</tr>
<tr>
<td>CTS</td>
<td>Clear To Send</td>
</tr>
<tr>
<td>CW</td>
<td>Contention Window</td>
</tr>
<tr>
<td>DARPA</td>
<td>Defence Advanced Research Projects Agency</td>
</tr>
<tr>
<td>DCF</td>
<td>Distributed Coordination Function</td>
</tr>
<tr>
<td>DCLQ</td>
<td>Distributed Cross-Layer QoS</td>
</tr>
<tr>
<td>DIFS</td>
<td>DCF IFS</td>
</tr>
<tr>
<td>DiffServ</td>
<td>Differentiated Service</td>
</tr>
<tr>
<td>DSCP</td>
<td>DiffServ Code Point</td>
</tr>
<tr>
<td>DSDV</td>
<td>Destination Sequenced Distance Vector</td>
</tr>
<tr>
<td>DSR</td>
<td>Dynamic Source Routing</td>
</tr>
<tr>
<td>ECN</td>
<td>Explicit Congestion Notification</td>
</tr>
<tr>
<td>EDCF</td>
<td>enhanced DCF</td>
</tr>
<tr>
<td>EF</td>
<td>Expedited forwarding</td>
</tr>
<tr>
<td>ETSI</td>
<td>European Telecommunications Standards Institute</td>
</tr>
<tr>
<td>FQMM</td>
<td>Flexible QoS Model for MANETs</td>
</tr>
<tr>
<td>FSR</td>
<td>Fisheye State Routing</td>
</tr>
<tr>
<td>Acronym</td>
<td>Description</td>
</tr>
<tr>
<td>---------</td>
<td>-------------</td>
</tr>
<tr>
<td>GSR</td>
<td>Global State Routing</td>
</tr>
<tr>
<td>HCF</td>
<td>Hybrid Coordination Function</td>
</tr>
<tr>
<td>IEEE</td>
<td>Institute of Electrical and Electronics Engineers</td>
</tr>
<tr>
<td>IFS</td>
<td>Inter Frame Spacing</td>
</tr>
<tr>
<td>IntServ</td>
<td>Integrated Services</td>
</tr>
<tr>
<td>IP</td>
<td>Internet Protocol</td>
</tr>
<tr>
<td>LAN</td>
<td>Local Area Networks</td>
</tr>
<tr>
<td>LANMAR</td>
<td>Landmark Routing</td>
</tr>
<tr>
<td>MAC</td>
<td>Medium Access Control</td>
</tr>
<tr>
<td>MANET</td>
<td>Mobile Ad hoc Networks</td>
</tr>
<tr>
<td>MQRD</td>
<td>Multipath QoS Routing for supporting DiffServ</td>
</tr>
<tr>
<td>MRL</td>
<td>Message Retransmission List</td>
</tr>
<tr>
<td>MSR</td>
<td>Multipath Source Routing</td>
</tr>
<tr>
<td>NAV</td>
<td>Network Allocation Vector</td>
</tr>
<tr>
<td>NDMR</td>
<td>Node-Disjoint Multipath Routing</td>
</tr>
<tr>
<td>OFDM</td>
<td>Orthogonal Frequency Division Multiplexing</td>
</tr>
<tr>
<td>OSI</td>
<td>Open System Interconnection</td>
</tr>
<tr>
<td>PCF</td>
<td>Point Coordination Function</td>
</tr>
<tr>
<td>PDA</td>
<td>Personal Digital Assistant</td>
</tr>
<tr>
<td>PHBs</td>
<td>Per-Hop Behaviours</td>
</tr>
<tr>
<td>PRNet</td>
<td>Packet Radio Network</td>
</tr>
<tr>
<td>QoS</td>
<td>Quality of Service</td>
</tr>
<tr>
<td>RED</td>
<td>Random Early Detection</td>
</tr>
<tr>
<td>RREP</td>
<td>Route Reply packet</td>
</tr>
<tr>
<td>RREQ</td>
<td>Route Request packet</td>
</tr>
<tr>
<td>RSVP</td>
<td>Resource reservation Protocol</td>
</tr>
<tr>
<td>RTS</td>
<td>Request To Send</td>
</tr>
<tr>
<td>SIFS</td>
<td>Short IFS</td>
</tr>
<tr>
<td>SMR</td>
<td>Split Multipath Routing</td>
</tr>
<tr>
<td>SURAN</td>
<td>Survivable adaptive radio networks</td>
</tr>
<tr>
<td>ToS</td>
<td>Type of Service</td>
</tr>
<tr>
<td>Acronym</td>
<td>Description</td>
</tr>
<tr>
<td>---------</td>
<td>----------------------------------</td>
</tr>
<tr>
<td>WLAN</td>
<td>wireless local area network</td>
</tr>
<tr>
<td>WRP</td>
<td>wireless routing protocol</td>
</tr>
<tr>
<td>ZRP</td>
<td>Zone Routing Protocol</td>
</tr>
</tbody>
</table>
Chapter 1 Introduction

1.1 Research Motivation

In wireless networking, there are two main architectures: infrastructure (single-hop) networks and mobile ad hoc (multi-hop) networks (MANETs) [1]. Infrastructure networks include cellular networks and wireless local area networks. Users are connected via base stations/access points and backbone networks. Although users can handover between base stations or access points and roam among different networks, their mobility is limited within the coverage areas of the base stations or access points. Ad hoc networks [2, 3, 15] exclude the use of a wired infrastructure. Mobile nodes can form arbitrary networks “on the fly” to exchange information without the need of pre-existing network infrastructure. Ad hoc networks can extend communication beyond the limit of infrastructure-based networks.

A fundamental problem in ad hoc networking is how to deliver data packets among nodes efficiently without predetermined topology or centralized control, which is the main objective of ad hoc routing protocols. Each node in the network functions as both a host and a router, and changes of network topology are distributed among the nodes. Design of efficient and reliable routing protocols in such a network is a challenging issue.

On-demand routing protocols in particular, are widely studied because they consume less bandwidth than proactive protocols. Ad Hoc On-demand Distance Vector (AODV) [4] and Dynamic Source Routing (DSR) [5] are the two most widely studied on-demand ad hoc routing protocols. Previous work [6, 7, 8] has shown limitations of the two protocols. The main reason is that both of them build and rely on a unipath route for each data session. Whenever there is a link break on the active route, both of the two routing protocols have to invoke a route discovery process. On-demand multipath routing protocols can alleviate these
problems by establishing multiple paths between a source and a destination in a single route discovery. A new route discovery is invoked only when all of its routing paths fail or when there only remains a single path available. In this thesis, a practical Node-Disjoint Multipath Routing Protocol (NDMR) [9] is proposed. NDMR has two novel aspects compared to the other on-demand multipath protocols: it reduces routing overhead dramatically and achieves multiple node-disjoint routing paths.

Future applications of MANETs are expected to be based on all-IP architecture and be capable of carrying real-time multimedia applications such as voice and video as well as data. Multimedia applications place stringent requirements on networks for delivering real time audio and video packets. Compared to the requirements of traditional data-only applications, these new requirements generally include a high packet delivery rate, a low delay and a small jitter. 150 ms is specified in ITU-T G.114 recommendation [21] as the maximum desired one-way delay to achieve high-quality voice. A delay above 250 ms is felt as unacceptable. Jitter is also called the delay variation between consecutive packets. It is an important metric for real-time flows. A smaller jitter indicates a higher quality flow.

DiffServ [10] is a standard approach for a more scalable way to achieve QoS in any IP network and could potentially be used to provide QoS in MANETs because it minimises the need for signalling. However, one of the biggest drawbacks of DiffServ is that the QoS provisioning is separate from the routing process. This thesis presents a Multipath QoS Routing protocol for supporting DiffServ (MQRD) [11], which combines the advantages of NDMR and DiffServ. The protocol can classify network traffic into different priority levels and apply priority scheduling and queuing management mechanisms to obtain QoS supports.

Many very important aspects, such as the routing scheme, service model, admission control, resource reservation, packet scheduling, signalling techniques and MAC protocols need to be considered to support QoS in ad hoc networks [64,
Actually, every layer of mobile nodes of ad hoc networks should be made QoS aware because only when all these QoS-aware functions are considered together can an effective QoS be provided for the end-user applications [103, 104, 105]. The design of cross layer QoS-aware protocols in ad hoc networks requires perfect coordination between different layers of the protocol stack to support QoS for real-time traffic as well as best effort traffic. A distributed Cross-Layer QoS (DCLQ) architecture is proposed to provide QoS improvement for real-time flows in mobile ad hoc networks. Without any extra control overhead in network layer, DCLQ can schedule packets of real-time flows according to their per-hop QoS requirements. DCLQ implements per-hop delay QoS-aware priority scheduling and QoS consideration of MAC layer to ensure that real-time flows to achieve their desired service level.

1.2 Research Contribution

In this thesis, multipath routing and QoS provisioning of mobile ad hoc networks are addressed. The major contributions of the work are as follows.

- A stable Node-Disjoint Multipath Routing (NDMR) protocol with low control overhead is proposed to overcome the shortcomings of current unipath routing and multipath routing protocols. NDMR has two novel aspects in that it reduces routing overhead dramatically and achieves multiple node-disjoint routing paths.

- Multipath QoS Routing for supporting DiffServ (MQRD) combines the advantages of NDMR and DiffServ. The protocol can classify network traffic into different priority levels and apply priority scheduling and queuing management mechanisms to obtain QoS supports.

- A distributed Cross-Layer QoS (DCLQ) architecture is proposed to provide QoS guarantees for real-time flows in mobile ad hoc networks. DCLQ can make real time flows get their desired service level by
considering per-hop QoS requirement of the flow and the local resource in
every node to schedule real time packet flows.

1.3 Author’s Publications

[1] Xuefei Li and Laurie Cuthbert, “Node-Disjoint Multipath Routing and
Distributed Cross-Layer QoS Guarantees in Mobile Ad hoc Networks”,
received by the 7th IEEE ACIS International Conference on Software
Engineering, Artificial Intelligence, Networking, and Parallel/Distributed

Multipath Routing, DiffServ and Distributed Traffic Control in Mobile Ad
hoc Networks”, In Proceedings of International Conference on Mobile Ad-
hoc and Sensor Networks (MSN), Wuhan, China, December 2005.

Provisioning for Mobile Ad hoc Networks”, In Proceedings of the 19th
International Teletraffic Congress (ITC19), Beijing, China, August 29 –
September 2, 2005.

[4] Xuefei Li and Laurie Cuthbert, Multipath QoS Routing of supporting
DiffServ in Mobile Ad hoc Networks, In Proceedings of the 6th IEEE ACIS
International Conference on Software Engineering, Artificial Intelligence,
Networking, and Parallel/Distributed Computing (SNPD2005), Towson,
Maryland, May 23 ~ 25, 2005

[5] Xuefei Li and Laurie Cuthbert, QoS Provisioning By Integrating Multipath
Routing and DiffServ in Ad hoc Networks, In Proceedings of the ninth
Canadian Workshop on Information Theory, Montreal, Canada, June 5 –
June 8.


### 1.4 Thesis Organization

Chapter 2 gives a brief overview of routing in mobile ad hoc networks. This chapter provides background and describes related research efforts and existing problems in ad hoc routing protocols. Some basic concepts and classifications of ad hoc networking are also introduced.
In chapter 3 a stable Node-Disjoint Multipath Routing Protocol with low control overhead is proposed. The important components of the protocol, such as path accumulation, decreasing routing overhead and selecting node-disjoint paths, are explained. This chapter also presents simulation models and implementation of the protocol in OPNET.

Chapter 4 presents a Multipath QoS Routing protocol for supporting DiffServ (MQRD), which combines the advantages of NDMR and DiffServ. The protocol can classify network traffic into different priority levels and apply priority scheduling and queuing management mechanisms to obtain QoS supports.

Chapter 5 presents a distributed Cross-Layer QoS (DCLQ) architecture to provide QoS guarantees for real-time flows in mobile ad hoc networks. Without any extra control overhead in network layer, DCLQ schedules dynamically packets of real-time flows according to their per-hop QoS requirements. DCLQ implements priority scheduling of per-hop delay requirement and QoS consideration of MAC layer to ensure that real-time flows to achieve their desired service level.

Chapter 6 summarizes the work in this thesis, draws the conclusions and also mentions the future work.
Chapter 2 Routing in Mobile Ad hoc Networks

This chapter provides background and describes related research efforts and existing problems in ad hoc routing protocols. Section 2.1 gives a general introduction about ad hoc networks. Section 2.2 explains several important concepts, including proactive versus reactive routing approaches and hierarchical routing. Section 2.3 describes some typical ad hoc proactive routing protocols. Section 2.4 presents several typical ad hoc reactive routing protocols. Section 2.5 provides a review of current on-demand multipath routing protocols in wireless ad hoc networks. Section 2.6 describes existing problems of current multipath routing protocols.

2.1 Ad hoc Networks

There are two architectures that allow two wireless stations to communicate with each other. The first one relies on a third fixed party (a base station) that will hand over the offered traffic from a station to another, as illustrated in Figure 2.1. This same entity will regulate the allocation of radio resources. When a source node wishes to communicate with a destination node, the former notifies the base station, which eventually establishes the communication with the destination node. At this point, the communicating nodes do not need to know about the route from one to the other. All that matters is that both source and destination nodes are within the transmission range of the base station; if one of them loses this condition, the communication will abort.

The second approach, called ad-hoc, does not rely on any stationary infrastructure. All nodes in ad hoc networks are mobile and can be connected dynamically in an arbitrary manner. Each node in such networks behaves as a router and takes part in discovery and maintenance of routes to other nodes.
Figure 2.1  Illustration of the infrastructure network model

Figure 2.2  Illustration of the infrastructure-less networks

Figure 2.2 illustrates a simple 3-node ad-hoc network. In this figure, a source node S wants to communicate with a destination node D. S and D are not within transmission range of each other. Therefore, they both use the relay node R to forward packets from one to another. R functions as a host and a router at the same time. By definition, a router is an entity that determines the path to be used in order to forward a packet towards its final destination. The router chooses the next node to which a packet should be forwarded according to its current understanding of the state of the network.
Research in ad hoc networking has been going on for some time. The history of wireless ad hoc networks can be traced back to the Defence Advanced Research Projects Agency (DARPA) packet radio network (PRNet), which evolved into the survivable adaptive radio networks (SURAN) program [12]. Ad hoc networks have played an important role in military applications and related research efforts, for example, the global mobile information systems (GloMo) program [13] and the near-term digital radio (NTDR) [14] program. Recent years have seen a new spate of industrial and commercial applications for wireless ad hoc networks, as many portable computers and personal digital assistants (PDAs) equipped with wireless ports are becoming more compact and inexpensive.

Ad hoc networks have numerous potential applications. For example, the IEEE 802.11 wireless LAN standard family [36, 46, 47, 48] and the European Telecommunications Standards Institute (ETSI) HIPERLAN/2 standard [49, 81], or even Bluetooth [51], support an ad hoc mode of operation for building simple ad hoc networks. Ad hoc networks are very useful in battle-field, disasters (such as flood, fire and earthquake) recovery, emergency search-and-rescue operations, home networking, meetings or conventions in which people wish to quickly share information [106].

Wireless ad hoc networks can be broadly divided into two categories: quasi-static and mobile. In a quasi-static ad hoc network, nodes are static or portable. However, due to power controls and link failures, the resulting network topology may be dynamic. A typical sensor network [107] is an example of a quasi-static ad hoc network. In mobile ad hoc networks (MANETs), the entire network may be mobile, and nodes may move quickly relative to each other. A major technical challenge in a MANET is the design of efficient routing protocols to cope with the rapid topology changes.
2.2 Routing Classification in Ad Hoc Networks

Routing in wireless ad hoc networks is clearly different from routing found in traditional infrastructure networks. Routing in ad hoc networks needs to take into account many factors including topology, selection of routing path and routing overhead, and it must find a path quickly and efficiently. Ad hoc networks generally have lower available resources compared with infrastructure networks and hence there is a need for optimal routing. Also, the highly dynamic nature of these networks means that routing protocols have to be specifically designed for them, thus motivating the study of protocols that aim at achieving routing stability.

Designing a routing protocol for ad hoc networks is challenging because of the need to take into account two contradictory factors:

- a node needs to know at least the “reachability” information to its neighbours for determining a packet route; and
- the network topology can change quite often.

Furthermore, as the number of network nodes can be large, finding a route to the destinations also requires large and frequent exchange of routing control information among the nodes. Thus, the amount of update traffic can be quite high, and it is even higher when the network includes high mobility nodes, which can impact the route overhead of routing protocols in such a way that there might be no bandwidth leftover for the transmission of data packets.

In wireless ad hoc networks, the communication range of a node is often limited and not all nodes can directly communicate with one another. Nodes are required to relay packets on behalf of other nodes to allow communication across the network. Since there is no pre-determined topology or configuration of fixed routes, an ad hoc routing protocol is used to dynamically discover and maintain up-to-date routes between communicating nodes.
2.2.1 Proactive versus Reactive Approaches

Ad hoc routing protocols may generally be categorized as being either proactive or on-demand (reactive) according to their routing strategy [50]. Proactive protocols require that nodes in a wireless ad hoc network should keep track of routes to all possible destinations so that when a packet needs to be forwarded, the route is already known and can be used immediately. Any changes in topology are propagated through the network, so that all nodes know of those changes in topology. Examples include “destination-sequenced distance-vector” (DSDV) routing [17], “wireless routing protocol” (WRP) [18], “global state routing” (GSR) [16], and “fisheye state routing” (FSR) [20].

On-demand protocols only attempt to build routes when desired by the source node so that the network topology is detected as needed (on-demand). When a node wants to send packets to some destination but has no routes to the destination, it initiates a route discovery process within the network. Once a route is established, it is maintained by a route maintenance procedure until the destination becomes inaccessible or until the route is no longer needed. Examples include “ad hoc on-demand distance vector routing” (AODV) [4], “dynamic source routing” (DSR) [5], and “Cluster Based Routing protocol” (CBRP) [25].

Proactive protocols have the advantage that new communications with arbitrary destinations experience minimal delay, but suffer the disadvantage of the additional control overhead to update routing information at all nodes. To cope with this shortcoming, reactive protocols adopt the inverse approach by finding a route to a destination only when needed. Reactive protocols often consume much less bandwidth than proactive protocols, but they will typically experience a long delay for discovering a route to a destination prior to the actual communication. However, because reactive routing protocols need to broadcast route requests, they may also generate excessive traffic if route discovery is required frequently.
2.2.2 Clustering and Hierarchical Routing

Scalability is one of the important problems in ad hoc networking. Scalability in ad hoc networks can be broadly defined as the network’s ability to provide an acceptable level of service to packets even in the presence of a large number of nodes in the network. In proactive routing protocols, when the number of nodes in the network increase, the number of topology control messages increases non-linearly and they may consume a large portion of the available bandwidth. In reactive routing protocols, large numbers of route requests to the entire network may eventually become packet broadcast storms. Typically, when the network size increases beyond certain thresholds, the computation and storage requirements become infeasible. When mobility is considered, the frequency of routing information updates may be significantly increased, thus worsening the scalability issues.

One way to address these problems and to produce scalable and efficient solutions is hierarchical routing. Wireless hierarchical routing is based on the idea of organizing nodes in groups and then assigning nodes different functionalities inside and outside a group. Both the routing table size and update packet size are reduced by including in them only part of the network. For reactive protocols, limiting the scope of route request broadcasts also helps to enhance efficiency. The most popular way of building hierarchy is to group nodes geographically close to each other into clusters. Each cluster has a leading node (cluster head) to communicate with other nodes on behalf of these clusters. Examples of hierarchical ad hoc routing protocols include “zone routing protocol” (ZRP) [23] and “zone-based hierarchical link state” (ZHLS) routing protocol [24].

2.3 Review of Ad hoc Proactive Routing Protocols

This section presents brief descriptions for several existing proactive routing protocols.
2.3.1 Dynamic Destination-Sequenced Distance-Vector Routing

The Destination-Sequenced Distance-Vector (DSDV) Routing Algorithm [17] is a proactive hop-by-hop distance vector routing protocol, which is based on the idea of the classical Bellman-Ford Routing Algorithm with certain improvements. Every mobile station maintains a routing table that lists all available destinations, the number of hops to reach the destination and the sequence number assigned by the destination node. The sequence number is used to distinguish stale routes from new ones to avoid the formation of loops. The stations periodically transmit their routing tables to their immediate neighbours. A station also transmits its routing table if a significant change has occurred in its table from the last update sent. The update is both time-driven and event-driven.

The routing table updates can be sent in two ways:

- a “full dump” where the full routing table is sent to the neighbours (which could span many packets); or
- an incremental update where only those entries from the routing table that have had a metric change since the last update are sent (and these must fit in a single packet).

If there is space in the incremental update packet, then those entries whose sequence number has changed may be included. When the network is relatively stable, incremental updates are sent to avoid extra traffic and full dumps are relatively infrequent. In a fast-changing network, incremental packets can grow large so full dumps will be more frequent.

Each route update packet, in addition to the routing table information, also contains a unique sequence number assigned by the transmitter. The route labelled with the highest (i.e. most recent) sequence number is used. If two routes have the same sequence number then the route with the best metric (i.e. shortest route) is used. Based on past history, the stations estimate the settling time of
routes. The stations delay the transmission of a routing update by settling time so as to eliminate those updates that would occur if a better route were found very soon.

### 2.3.2 The Wireless Routing Protocol (WRP)

The Wireless Routing Protocol (WRP) [18] is a proactive distance-vector routing protocol. Each node in the network maintains a *distance table*, a *routing table*, a *link-cost table* and a *message retransmission list*.

- The distance table of a node $x$ contains the distance of each destination node $y$ via each neighbour $z$ of $x$. It also contains the downstream neighbour of $z$ through which this path is realized.

- The routing table of node $x$ contains the distance of each destination node $y$ from node $x$, the predecessor and the successor of node $x$ on this path. It also contains a tag to identify if the entry is a simple path, a loop or invalid. Storing predecessor and successor in the table enables loops to be detected.

- The link-cost table contains the cost of the link to each neighbour of the node and the number of timeouts since an error-free message was received from that neighbour.

- The message retransmission list (MRL) contains information to let a node know which of its neighbours has not acknowledged its update message and to retransmit update message to that neighbour.

Nodes periodically exchange routing tables with their neighbours using update messages as well as on link changes. The nodes present on the response list for the update message (formed using the MRL) are required to acknowledge the receipt of the update message. If there is no change in the routing table since last update, the node is required to send an idle “Hello” message to ensure connectivity. On receiving an update message, the node modifies its distance
table and looks for better paths using the new information. Information is sent back to the original nodes about any new paths found so that their tables can be updated. The routing table is also updated if the new path is better than the existing path.

2.3.3 Global State Routing (GSR)

Global State Routing (GSR) [16] is similar to DSDV in that it takes the idea of link state routing but makes an improvement by reducing the flooding of routing messages.

In this algorithm, each node maintains a neighbor list, a topology table, a next hop table and a distance table.

- The neighbour list of a node contains the list of its neighbours (all nodes that can be heard by it).

- The link state information for each destination is maintained in the topology table together with the timestamp of the information.

- The next hop table contains the next hop to which the packets for each destination must be forwarded.

- The distance table contains the shortest distance to each destination node.

The routing messages are generated on a link change as in all link state protocols. When it receives a routing message, the node updates its topology table if the sequence number of the message is newer than the sequence number stored in the table and it then reconstructs its routing table and broadcasts the information to its neighbours.
2.3.4 Fisheye State Routing (FSR)

Fisheye State Routing (FSR) [20] is an improvement of GSR. The large size of update messages in GSR wastes a considerable amount of network bandwidth, so to reduce this, FSR takes an approach where each update message does not contain information about all nodes. Instead, it exchanges information about closer nodes more frequently than it does about farther nodes, thus reducing the update message size. In this way, each node gets accurate information about near neighbours and accuracy of information decreases as the distance from the node increases.

Even though a node does not have accurate information about distant nodes, the packets are routed correctly because the route information becomes more and more accurate as the packet moves closer to the destination.

2.4 Review of Ad hoc Reactive Routing Protocols

Reactive protocols take a lazy approach to routing. In contrast to proactive routing protocols, all up-to-date routes are not maintained at every node, but instead the routes are created as and when required. When a source wants to send to a destination, it invokes the route discovery mechanisms to find the path to the destination. In this section several typical reactive (on-demand) routing protocols are introduced.

2.4.1 Ad Hoc On-demand Distance Vector Routing (AODV)

Ad hoc on-demand distance vector (AODV) routing [4] adopts both a modified on-demand broadcast route discovery approach used in DSR [5] and the concept of destination sequence number adopted from destination-sequenced distance-vector routing (DSDV)[17].

When a source node wants to send a packet to some destination and does not have a valid route to that destination, it initiates a path discovery process and
broadcasts a route request (RREQ) message to its neighbours. The neighbours in turn forward the request to their neighbours until the RREQ message reaches the destination or an intermediate node that has an up-to-date route to the destination. Figure 2.3(a) illustrates the propagation of the broadcast RREQs in an ad hoc network.

![RREQ messages propagation](image)

(a) RREQ messages propagation

![RREP message sent back to source](image)

(b) RREP message sent back to source

**Figure 2.3 Route discovery in AODV**

In AODV, each node maintains its own sequence number and a broadcast ID. Each RREQ message contains the sequence numbers of the source and destination nodes and is uniquely identified by the source node’s address and a
broadcast ID. AODV utilizes destination sequence numbers to ensure loop-free routing and use of up-to-date route information. Intermediate nodes can reply to the RREQ message only if they have a route to the destination whose destination sequence number is greater or equal to that contained in the RREQ message.

So that a reverse path can be set up, each intermediate node records the address of the neighbour from which it received the first copy of the RREQ message, and additional copies of the same RREQ message are discarded. Once the RREQ message reaches the destination (or an intermediate node with a fresh route) the destination (or the intermediate node) responds by sending a route reply (RREP) packet back to the neighbour from which it first received the RREQ message. As the RREP message is routed back along the reverse path, nodes along this path set up forward path entries in their routing tables (Figure 2.3(b)).

When a node detects a link failure or a change in neighbourhood, a route maintenance procedure is invoked:

If a source node moves, it can restart the route discovery procedure to find a new route to the destination.

If a node along the route moves so that it is no longer contactable, its upstream neighbour sends a link failure notification message to each of its active upstream neighbours. These nodes in turn forward the link failure notification to their upstream neighbours until the link failure notification reaches the source node.

### 2.4.2 Dynamic Source Routing (DSR)

Dynamic source routing (DSR) [5] is an on-demand routing protocol for wireless ad hoc networks. DSR is based on the concept of source routing, in which a source node indicates the sequence of intermediate routes in the header of a data packet. Like other on-demand routing protocols, the operation of DSR can be divided into two procedures: route discovery and route maintenance.
Each node in the network keeps a cache of the source routes that it has learned. When a node needs to send a packet to some destination, it first checks its route cache to determine whether it already has an up-to-date route to the destination. If no route is found, the node initiates the route discovery procedure by broadcasting a route request message to neighbouring nodes. This route request message contains the address of the source and destination nodes, a unique identification number generated by the source node, and a route record to keep track of the sequence of hops taken by the route request message as it is propagated through the network. When an intermediate node receives a route discovery request, it checks whether its own address is already listed in the route record of the route request message. If not, it appends its address to the route record.

Figure 2.4 Route discovery in DSR
record and forwards the route request to its neighbours. Figure 2.4(a) illustrates the formation of the route record as the route request propagates through the network.

When the destination node receives the route request, it appends its address to the route record and returns it to the source node within a new route reply message. If the destination already has a route to the source, it can use that route to send the reply; otherwise, it can use the route in the route request message to send the reply. The first case is for situations where a network might be using unidirectional links and so it might not be possible to send the reply using the same route taken by the route request message. If symmetric links are not supported, the destination node may initiate its own route discovery message to the source node and piggyback the route reply on the new route request message. Figure 2.4(b) shows the transmission of route record back to the source node.

Route maintenance uses route error messages and acknowledgement messages. If a node detects a link failure when forwarding data packets, it creates a route error message and sends it to the source of the data packets. The route error message contains the address of the node that generates the error and the next hop that is unreachable. When the source node receives the route error message, it removes all routes from its route cache that have the address of the node in error. It may initiate a route discovery for a new route if needed. In addition to route error message, acknowledgements are used to verify the correct operation of links.

To reduce the route search overhead, an important optimization is allowing an intermediate node to send a route reply to the source node if it already has an up-to-date route to the destination.

### 2.4.3 Cluster based Routing Protocol (CBRP)

In Cluster Based Routing protocol (CBRP) [25], the nodes are divided into clusters. To form the cluster the following algorithm is used. When a node comes up, it enters the "undecided" state and broadcasts a Hello message. When a
cluster-head gets this hello message it responds with a triggered hello message immediately. When the undecided node gets this message it sets its state to "member". If the undecided node times out, then it makes itself the cluster-head if it has bi-directional link to some neighbour otherwise it remains in undecided state and repeats the procedure again. Cluster-heads are changed as infrequently as possible.

Each node maintains a neighbour table. For each neighbour, the neighbour table of a node contains the status of the link and the state of the neighbour (cluster-head or member). A cluster-head keeps information about the members of its cluster and also maintains a cluster adjacency table that contains information about the neighboring clusters. For each neighbor cluster, the table has entry that contains the gateway through which the cluster can be reached and the cluster-head of the cluster.

When a source has to send data to destination, it floods route request packets (but only to the neighboring cluster-heads). On receiving the request a cluster-head checks to see if the destination is in its cluster. If yes, then it sends the request directly to the destination else it sends it to all its adjacent cluster-heads. The cluster-heads address is recorded in the packet so a cluster-head discards a request packet that it has already seen. When the destination receives the request packet, it replies back with the route that had been recorded in the request packet. If the source does not receive a reply within a time period, it backs off exponentially before trying to send route request again.

In CBRP, routing is done using source routing. It also uses route shortening that is on receiving a source route packet, the node tries to find the farthest node in the route that is its neighbor and sends the packet to that node thus reducing the route. While forwarding the packet if a node detects a broken link it sends back an error message to the source and then uses local repair mechanism. In local repair mechanism, when a node finds the next hop is unreachable, it checks to see if the next hop can be reached through any of its neighbor or if hop after next hop
can be reached through any other neighbor. If any of the two works, the packet can be sent out over the repaired path.

Table 2.1 shows and compares the unipath routing protocols for mobile ad hoc networks.

<table>
<thead>
<tr>
<th></th>
<th>DSDV</th>
<th>WRP</th>
<th>GSR</th>
<th>FSR</th>
<th>AODV</th>
<th>DSR</th>
<th>CBRP</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Routing Category</strong></td>
<td>Proactive</td>
<td>Proactive</td>
<td>Proactive</td>
<td>Proactive</td>
<td>Reactive</td>
<td>Reactive</td>
<td>Reactive</td>
</tr>
<tr>
<td><strong>Beaconing</strong></td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td><strong>Periodic Update</strong></td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td><strong>Flood Control</strong></td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td><strong>TTL Limitation</strong></td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td><strong>QoS Support</strong></td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td><strong>Multicast Support</strong></td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td><strong>Power Management</strong></td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td><strong>Security Support</strong></td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
</tbody>
</table>

Table 2.1 Comparison of the unipath routing protocols

2.5 Ad Hoc On-demand Multipath Routing Protocols

Standard on-demand routing protocols in ad hoc wireless networks, such as AODV and DSR, are mainly intended to discover a single route between a source and destination node. When the route disconnects, nodes of the broken route simply drop data packets because no alternate path to the destination is available.
until a new route is established. Multipath routing is useful for finding multiple paths between a source and destination in a single discovery. These multiple paths between source and destination can be used to compensate for the dynamic and unpredictable topology change in ad hoc networks. Recently, several different multipath routing mechanisms have been proposed. This section introduces some main characteristics of these multipath protocols. AOMDV [27] and AODVM [31] routing protocols are based on the AODV [4] routing protocol, whereas SMR [28] and MSR [29] are based on DSR [5]

2.5.1 Ad hoc On-demand Multipath Distance Vector (AOMDV)

Ad hoc On-demand Multipath Distance Vector (AOMDV) [27] is an extension to the AODV protocol for computing multiple loop-free and link-disjoint paths. The protocol computes multiple loop-free and link-disjoint paths. Loop-freedom is guaranteed by using a notion of “advertised hopcount”. Link-disjointness of multiple paths is achieved by using a particular property of flooding.

To keep track of multiple routes, the routing entries for each destination contain a list of the next-hops together with the corresponding hop counts. All the next hops have the same sequence number. For each destination, a node maintains the advertised hop count, which is defined as the maximum hop count for all the paths. This is the hop count used for sending route advertisements of the destination. Each duplicate route advertisement received by a node defines an alternative path to the destination. To ensure loop freedom, a node only accepts an alternative path to the destination if it has a lower hop count than the advertised hop count for that destination. Because the maximum hop count is used, the advertised hop count therefore does not change for the same sequence number. When a route advertisement is received for a destination with a greater sequence number, the next-hop list and advertised hop count are reinitialized.

AOMDV can be used to find link-disjoint routes. To find disjoint routes, each node does not immediately reject duplicate RREQs. Each RREQ carries an
additional field called firsthop to indicate the first hop (neighbour of the source) taken by it. Also, each node maintains a first hop list for each RREQ to keep track of the list of neighbours of the source through which a copy of the RREQ has been received. In an attempt to get multiple link-disjoint routes, the destination replies to duplicate RREQs regardless of their first hop. To ensure link-disjointness in the first hop of the RREP, the destination only replies to RREQs arriving via unique neighbours. The trajectories of each RREP may intersect at an intermediate node, but each takes a different reverse path to the source to ensure link-disjointness.

### 2.5.2 Split Multipath Routing (SMR)

Split Multipath Routing (SMR) proposed in [28] is an on-demand multipath source routing protocol that builds multiple routes using a request/reply cycle. SMR can find an alternative route that is maximally disjoint from the source to the destination. When the source needs a route to the destination but no route information is known, it floods the Route Request (RREQs) message to the entire network in order to find maximally disjoint paths, so the approach has a disadvantage of transmitting more RREQ packets. Because this packet is flooded, several duplicates that traversed through different routes reach the destination. The destination node selects multiple maximally disjoint routes and sends Route Reply (RREP) packets back to the source via the chosen routes. In order to choose proper maximally disjoint route paths, the destination must know the entire path of all available routes. Therefore, SMR uses the source routing approach where the information of the nodes that comprise the route is included in the RREQ packet.

SMR is similar to DSR, and is used to construct maximally disjoint paths. Unlike DSR, intermediate nodes do not keep a route cache, and therefore, do not reply to RREQs. This is to allow the destination to receive all the routes so that it can select the maximally disjoint paths. Maximally disjoint paths have as few links or nodes in common as possible. Duplicate RREQs are not necessarily
discarded. The algorithm only selects two routes. In the algorithm, the destination sends a RREP for the first RREQ it receives, which represents the shortest delay path. The destination then waits to receive more RREQs. From the received RREQs, the path that is maximally disjoint from the shortest delay path is selected. If more than one maximally disjoint path exists, the shortest hop path is selected. If more than one shortest hop path exists, the path whose RREQ was received first is selected. The destination then sends an RREP for the selected RREQ.

2.5.3 Multipath Source routing (MSR)

Multipath Source Routing (MSR) [29, 30] is an extension of the on-demand DSR [5] protocol. It consists of a scheme to distribute traffic among multiple routes in a network. MSR uses the same route discovery process as DSR with the exception that multiple paths can be returned, instead of only one.

When a source requires a route to a destination but no route is known (in the cache), it will initiate a route discovery process by flooding a RREQ packet throughout the network. A route record in the header of each RREQ records the sequence of hops that the packet passes. An intermediate node contributes to the route discovery by appending its own address to the route record. Once the RREQ reaches the destination, a RREP will reverse the route in the route record of the RREQ and traverse back through this route.

Each route is given a unique index and stored in the cache, so it is easy to pick multiple paths from there. Independence between paths is very important in multipath routing, therefore disjoint paths are preferred in MSR. As MSR uses the same route discovery process as DSR, where the complete routes are in the packet headers, looping will not occur. When a loop is detected, it will be immediately eliminated.

Since source routing is used in MSR, intermediate nodes do nothing but forward the packet according to the route in the packet-header. The routes are all
calculated at the source. A multiple-path table is used for the information of each different route to a destination. This table contains for each route to the destination: the index of the path in the route cache, the destination ID, the delay and the calculated load distribution weight of a route. The traffic to a destination is distributed among multiple routes. The weight of a route simply represents the number of packets sent consecutively on that path.

### 2.5.4 Ad hoc On-demand Distance Vector Multipath Routing

Ad hoc On-demand Distance Vector Multipath Routing (AODVM) [31] is an extension to AODV for finding multiple node disjoint paths. Instead of discarding the duplicate RREQ packets, intermediate nodes are required to record the information contained in these packets in the RREQ table. For each received copy of an RREQ message, the receiving intermediate node records the source that generated the RREQ, the destination for which the RREQ is intended, the neighbour that transmitted the RREQ, and some additional information in the RREQ table. Furthermore, intermediate relay nodes are precluded from sending an RREP message directly to the source.

When the destination receives the first RREQ packet from one of its neighbours, it updates its sequence number and generates an RREP packet. The RREP packet contains an additional field called “last hop ID” to indicate the neighbour from which the particular copy of RREQ packet was received. This RREP packet is sent back to the source via the path traversed by the RREQ. When the destination receives duplicate copies of the RREQ packet from other neighbours, it updates its sequence number and generates RREP packets for each of them. Like the first RREP packet, these RREP packets also contain their respective last hop nodes' IDs.

When an intermediate node receives an RREP packet from one of its neighbours, it deletes the entry corresponding to this neighbour from its RREQ table and adds a routing entry to its routing table to indicate the discovered route
to the originator of the RREP packet (the destination). The node, then, identifies the neighbour in the RREQ table via which, the path to the source is the shortest, and forwards the RREP message to that neighbour. The entry corresponding to this neighbour is then deleted from the RREQ table. In order to ensure that a node does not participate in multiple paths, when nodes overhear any node broadcasting an RREP message, they delete the entry corresponding to the transmitting node from their RREQ tables.

Intermediate nodes make decisions on where to forward the RREP messages (unlike in source routing) and the destination, which is in fact the originator of these messages, is unaware as to how many of these RREP messages that it generated actually made it back to the source. Thus, it is necessary for the source to confirm each received RREP message by means of a Route Confirmation message (RRCM). The RRCM message can, in fact, be added to the first data packet sent on the corresponding route and will also contain information with regards to the hop count of the route, and the first and last hop relays on that route.

2.6 Problem with Current Multipath Routing Protocols

Previous section introduces simply routing mechanisms and benefits of several existing multipath protocols. Although these protocols can build on demand multiple routing paths, all of them will encounter a broadcast storm of routing packets in the process of looking for multiple disjoint routing paths.

When a source in these multipath routing protocols needs a route to a destination but no route information is known, it floods the Route Request (RREQ) message to the entire network. In order to ensure that the destination can select disjoint paths, all the four multipath routing protocols do not discard duplicate RREQs at intermediate nodes. Also, they do not allow intermediate nodes, which know routing information to the destination, to reply the RREQ. These lead to dramatic increase of routing overhead in the ad hoc network.
Because bandwidth in wireless ad hoc networks is limited, how to reduce routing overhead has to be considered when designing a routing protocol.

None of current multipath routing protocols have taken measures to minimize routing flood overhead. In the next chapter, a novel Node-Disjoint Multipath Routing protocol (NDMR) with low control overhead is proposed to solve these problems.

Table 2.2 compares the main characteristics of existing multipath routing protocols.

<table>
<thead>
<tr>
<th></th>
<th>AOMDV</th>
<th>SMR</th>
<th>AODVM</th>
<th>MSR</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Routing Category</strong></td>
<td>Reactive</td>
<td>Reactive</td>
<td>Reactive</td>
<td>Reactive</td>
</tr>
<tr>
<td><strong>Loop-free Paths</strong></td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td><strong>Routing Overhead Control</strong></td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td><strong>Node-disjoint Paths</strong></td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td><strong>Complete Routes Known at Source</strong></td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td><strong>Paths Used Simultaneously</strong></td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td><strong>TTL Limitation</strong></td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td><strong>QoS Support</strong></td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td><strong>Multicast Support</strong></td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td><strong>Power Management</strong></td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td><strong>Security Support</strong></td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
</tbody>
</table>

Table 2.2 Comparison of the multipath routing protocols
Chapter 3 Node-Disjoint Multipath Routing with Low Overhead

3.1 Introduction

A mobile ad hoc network is a collection of mobile nodes that can communicate with each other using multihop wireless links without requiring any fixed based-station infrastructure and centralized management. Each node in the network acts as both a host and a router. As seen in Chapter 2, the design of an efficient and reliable routing protocol in such a network is a challenging issue.

On-demand routing protocols in particular, have been widely developed because they consume much less bandwidth than proactive protocols. Ad hoc On-Demand Distance Vector (AODV) [4] and Dynamic Source Routing (DSR) [5] are probably the two most widely studied on-demand ad hoc routing protocols. Previous work, and in particular [6, 7, 8, 33], has shown limitations of the two protocols. The main reason is that both of them build and rely on a unipath route for each data session and whenever there is a link break on the active route, there has to be a route discovery process, leading to more delay and overhead.

If multipath paths are set up between a source and a destination in a single route discovery process, then a new route discovery is invoked only when all of the routing paths fail or when there only remains a single path available. Multipath routing in ad hoc networks has been proposed in [27, 28, 29, 31]. Although these protocols build multiple routes on demand, most of them only discuss non-disjoint or link-disjoint paths. Furthermore, all of them flood route request packets to the whole network in order to discover multiple routing paths.

In this chapter a novel and practical route protocol, called node-disjoint multipath routing protocol (NDMR) [9], is proposed. The protocol modifies and extends AODV to enable the path accumulation feature of DSR in route request/reply
packets and discover multiple node-disjoint routing paths with a low broadcast redundancy.

The remainder of this chapter is organized as follows. Section 3.2 describes the novel node-disjoint multipath routing (NDMR) Protocol mechanism in detail. Section 3.3 presents a simulation model of NDMR and its implementation in OPNET. Section 3.4 presents simulation validation. In Section 3.5 simulation environment model is described. Section 3.6 presents the simulation results and compares performances between unipath and multipath routing protocols.

### 3.2 Node-Disjoint Multipath Routing (NDMR)

The Node-Disjoint Multipath Routing (NDMR) protocol proposed here is novel: it can efficiently discover multiple route paths between nodes desiring communication with minimal control overhead (low broadcast redundancy) and minimal routing latency. This section shows the protocol’s mechanism in detail.

#### 3.2.1 Protocol Message Formats

There are three types of control messages (RREQ, RREP and RERR) and a type of data message in NDMR.

- **RREQ (Type, Src, SrcSeqNb, Dest, DestSeqNb, BroadcastID, TTL, RoutePath):** A route request message from source Src to destination Dest. The field Type is the type of the message. SrcSeqNb is the sequence number of source; DestSeqNb is the sequence number of destination; BroadcastID is broadcast ID of the RREQ. TTL is Time to Live value of the RREQ. RoutePath is path accumulation list of the route path.

- **RREP (Type, Src, SrcSeqNb, Dest, DestSeqNb, RoutePath):** A route reply message from destination Dest to source Src along reverse route path. The field Type is the type of the message. SrcSeqNb is the sequence number of
source; DestSeqNb is the sequence number of destination; RoutePath is path accumulation list of the route path.

- REER (Type, PreviousHop, NextHop): A route error message includes the address of PreviousHop and the address of NextHop. The error message will be produced when a link to a neighbouring node is broken or forwarded towards the source along the reverse route path when an error message is received from a neighbouring node.

- DATA (Type, Src, Dest, NextHop, Payload): A data message includes message type, source address Src, destination address Dest, the address of next hop NextHop and data payload Payload.

### 3.2.2 Data Structures

Each node of the ad hoc network keeps and maintains a neighbours table, a source_broadcastID table, a route table and a reverse route table. The neighbours table is used to record neighbourhood information. The source_broadcastID table is used to record the current pair of source address and broadcast ID. The route table is used to store routing information towards every destination and the reverse route table is used to keep reverse routing information towards every source.

**Neighbours table:** Neighbours table includes an updated list of its neighbours, which is periodically updated.

**Source_broadcastID table:** The table is used to record the address of a source node and broadcast ID, which is incremented each time the source node initiates a RREQ.

**Route table:** The route table has an update list of all the possible routes to the desired destinations. Each element in the table is a six-tuple of the form
<src_ip_addr, dest_ip_addr, dest_seq_nb, next_hop_valid, next_hop, hop_count>:

- src_ip_addr and dest_ip_addr represent the unique addresses of the source and destination node, respectively;
- dest_seq_nb field represents the sequence number of the destination;
- next_hop_valid flag represents whether or not the route is currently valid;
- next_hop field contains the address of the neighbouring node to which data packets need to be forwarded; and
- hop_count field contains the number of intermediate nodes from the source to the destination node on this route.

Reverse route table: the reverse route table has an update list of all possible routes to the sources. Each element in the table is a seven-tuple of the form <src_ip_addr, dest_ip_addr, src_seq_nb, reverse_next_hop_valid, reverse_next_hop, hop_count, reverse shortest routing hopcount, routing_path_list>:

- src_ip_addr and dest_ip_addr represent the unique addresses of the source and destination node, respectively;
- src_seq_nb field represents the sequence number of the source;
- reverse_next_hop_valid flag represents whether or not the reverse route is currently valid;
- reverse_next_hop field contains the address of the neighbour node from which data packets come;
- hop_count field contains the number of intermediate nodes from the destination to the source node on this route;
reverse shortest routing hopcount field contains the shortest routing hopcount to source; and

routing_path_list field contains the whole routing path list.

3.2.3 Route Discovery

When a source node wants to communicate with a destination node, it checks its route table to confirm whether it has a valid route to the destination. If so, it sends the packet to the appropriate next hop towards the destination. However, if the node does not have a valid route to the destination, it must initiate a route discovery process. To begin such a process, the source creates a RREQ (Route Request) packet. This packet contains message type, source address, current sequence number of source, destination address, the broadcast ID and route path. The broadcast ID is incremented every time when the source node initiates a RREQ. In this way, the broadcast ID and the address of the source node form a unique identifier for the RREQ. Figure 3.1 shows the flow chart of initiating a discovery process.
Prepare to send data to a destination node

Check route table, but no route is available towards the destination

source sequence number + 1
Broadcast ID + 1

Save new source sequence number and new Broadcast ID to source_broadcastID table

Create a RREQ packet

Broadcast the RREQ packet to neighbouring nodes

End

**Figure 3.1 The flow chart of initiating a discovery process**

Finding node-disjoint multiple paths with low broadcast overhead is not an easy task when the network topology is unknown and changing dynamically. This section briefly describes the mechanism of NDMR based on AODV that enables path accumulation during a multipath route discovery cycle and records the shortest routing hops to minimize its routing overhead and achieve multiple node-disjoint routing paths. NDMR routing computation has three key components to avoid introducing a broadcast flood in MANETs:
• Path accumulation;
• Decreasing multipath broadcast routing packets;
• Selecting node-disjoint paths.

3.2.3.1 Path Accumulation

The main goal of NDMR is to build multiple node-disjoint paths with a low broadcast overhead during a route discovery. To achieve this goal, the destination must know the entire routing path list of all available routes so that it can select the right node-disjoint route paths from the candidate paths. Therefore, AODV is modified to include path accumulation in RREQ packets. When the RREQ packets are generated or forwarded by the nodes in the network, each node appends its own address to the routing request packets. When a RREQ packet arrives at its destination, the destination is responsible for judging whether or not the routing path is a node-disjoint path. After confirming a node-disjoint path, the destination generates a Route Reply (RREP) packet that contains the node list of the whole route path and unicasts it back towards the source that originated the RREQ message along the reverse route path. When an intermediate node receives a RREP, it updates its routing table entry and its reverse routing table entry by using the nodes list of the whole route path contained in the RREP.
As an example, consider five nodes A, B, C, D and E as shown in Figure 3.2. Node A wants to send data to node E. Since A does not have a route for E in its routing table, it broadcasts a route request. Node B receives the route request, appends its own address to the request, and forwards the request since it also has no route to E. Similarly, when node C and node D receive the RREQ, they append their address to the request and forward it. When the request reaches destination E, node E checks the path accumulation list (A-B-C-D) from the RREQ and judges whether or not the routing path is a node-disjoint path. If it is, node E generates a RREP packet that contains the path accumulation list of the whole route path and unicasts it back to the source that originated the RREQ message along the reverse route path. If not, node E discards the received RREQ.

### 3.2.3.2 Decreasing Broadcast Routing Overhead

In AODV, if a source node does not know a route to a destination, it will initiate a route discovery by flooding a Route Request (RREQ) message. The RREQ message carries the source ID and the RREQ sequence number. When an intermediate node receives a RREQ, if it is the first time that the node receives this RREQ message, then the node will broadcast the RREQ message again. Otherwise, the node will drop the RREQ packet.

In NDMR, using this method of broadcasting RREQ, the possibility of finding node-disjoint multiple paths is almost zero so a novel method has to be introduced. The reason is that later duplicate RREQ packets, which may come from a different path, are dropped. However, if all of the duplicate RREQ packets are re-broadcast, this will lead to a routing packet broadcast storm and decrease dramatically the performance of the ad hoc networks. In order to avoid this problem, a novel approach recording the Shortest Routing Hops of Loop-free Paths is implemented to decrease routing broadcast overhead.

When a node receives a RREQ packet for the first time, it checks the path accumulation list from the packet and calculates the number of hops from the
source to itself and records the number as the shortest number of hops in its reverse route table entry. If the node receives the RREQ duplicate again, it computes the number of hops from the source to itself and compares it to the number of the shortest hops recorded in its reverse route table entry. If the number of hops is larger than the shortest number of hops in its reverse route table entry, the node drops the RREQ packet. Otherwise (less than or equal to), the node appends its own address to the route path list of the RREQ packet and broadcasts the RREQ packet to its neighbouring nodes.

![Figure 3.3 Shortest Routing Hops of Loop-free Paths](image)

For example, in Figure 3.3, from source node S to node c there are five route paths: S-c, S-b-c, S-a-c, S-b-g-c, S-a-e-c. The numbers of hops are 1, 2, 2, 3 and 3 respectively. When node c receives the RREQ packet at the first time from path S-c, it records 1 as the shortest number of hops in its reverse route table entry. When the node c receives the RREQ duplicates from the other four route paths, it calculates the number of hops and compares it to the shortest number of hops in its reverse route table entry. Because the numbers of hops of route list of the four route paths are all greater than 1, the four RREQ duplicate packets are dropped. From the example it can be seen that “recording the shortest routing hops” approach results in most of the RREQ packets being discarded in the process of discovering multiple node-disjoint paths. Furthermore, the approach can also avoid forming loop paths. This is a novel and practical approach to guarantee
loop-free paths as well as to dramatically decrease the routing overhead. The flow chart of reducing routing overhead is illustrated in Figure 3.4.

Figure 3.4 Flowchart of reducing broadcast routing overhead
Figure 3.5 illustrates the route request process with low overhead in the entire network. Source S broadcasts a route request packet. Each intermediate node uses the approach with low routing overhead to propagate and discard packets. Therefore, only seven packets (S-c-f-D, S-a-i-g-D, S-b-e-h-D, S-c-i-g-D, S-c-e-h-D, S-c-f-g-D, S-c-f-h-D) can reach the destination D. Most of packets are discarded. However, not all of paths packets that arrive in destination are node-disjoint. In next section how to choose node-disjoint paths will be discussed.

3.2.3.3 Selecting Node-Disjoint Paths

In the algorithm of selecting node-disjoint paths, the destination is responsible for selecting and recording multiple node-disjoint route paths. In order to decrease the overhead of the route table in each node, the number of node-disjoint routing paths has been limited to three although more than three node-disjoint routes can be searched. In Figure 3.6, its three node-disjoint route paths are: S-a-i-g-D, S-c-f-D, S-b-e-h-D. When receiving the first RREQ packet (the shortest route path: S-c-f-D), the destination records the list of node IDs for the entire route path in its reverse route table and sends a RREP that includes the route path towards the source along the reverse route. When the destination receives a
duplicate RREQ, it will compare the whole route path in the RREQ to all of the existing node-disjoint route paths in its route table entry. If there is not a common node (except source and destination) between the route path from the current received RREQ and any node-disjoint route path recorded in the destination’s reverse route table entry, the route path of the current RREQ (such as S-a-i-g-D or S-b-e-h-D) satisfies the requirement of node-disjointness and is recorded in the reverse route table of the destination. Otherwise, the route path (such as paths: S-c-i-g-D, S-c-e-h-D, S-c-f-g-D, S-c-f-h-D) and the current received RREQ are discarded. The flow chart of selecting node-disjoint paths is shown in Figure 3.7.

Because the node IDs of the entire path are included in the RREP, each intermediate node receiving a RREP can record some necessary information from the path to its route table before forwarding the RREP. At first, the intermediate node sets up a forward path entry to the destination in its route table and a reverse path entry to the source in its reverse route table. According to the information in path IDs list, the forward path entry records the IP address of the destination and the IP address of the neighbour from which the RREP arrived. The reverse path entry records the IP address of the source and the IP address of the next hop to the source. Finally the intermediate node forwards the RREP towards the source.
node along the reverse route path. When the RREP arrives at the source node, it does not need to be forwarded. The source node records the next hop to destination into its multiple route forward path entry. A flowchart of processing an incoming RREP packet is illustrated in Figure 3.8. After the first RREP arrives at the source, the newly established route can now be used to send the data packets. Figure 3.9 shows the flowchart of processing data packets.

![Flowchart of selecting node-disjoint paths](image)

**Figure 3.7 Flowchart of selecting node-disjoint paths**
begin

Receive a RREP packet

Read source IP address from the RREP packet

Source IP address = my IP address?

Yes → Read node list of the route path from the RREP packet

No → Read destination sequence number from the RREP packet

The destination sequence number < destination sequence number in route table?

Yes → Update destination sequence number and clear nexthop field and its valid flag

No → Discard the RREP packet

The destination sequence number > destination sequence number in route table?

Yes → Send the data packet

No → No

Are some data packets waiting to being sent?

Yes → Send the data packet

No → End

Record next hop toward destination to route table entry and next hop toward source to reverse route table entry

Forward the RREP toward source node along reverse route path

End

Figure 3.8 Flowchart of processing an incoming RREP packet
3.2.4 Route Maintenance

In general, route links in ad hoc networks are broken frequently due to the mobility of nodes, congestion and packet collisions. Like AODV, each node of NDMR is dependent on sending out HELLO packets to maintain local connectivity. Failure to receive a HELLO packet from a neighbour is regarded as an indication that the link to the neighbour is broken. A Route Error (RERR) packet is propagated from the upstream node of the link failure to the source node.
for the route. When an intermediate node receives a RERR packet, it marks its
route to the destination invalid and then propagates the RERR to its precursor
node along the reverse route path. After receiving the RERR, the source
invalidates the route path to destination and chooses a valid node-disjoint routing
path as active routing path from the routing table to continue to forward data
packets. Additionally, the source needs to check each valid flag of the three node-
disjoint route paths. If only one of them is valid or all of three routing paths are
invalid, the source initiates a route discovery process.

3.3 Simulation Implementation

The commercial discrete event simulator OPNET [32] is used to simulate a
general framework of wireless ad hoc networks. The section gives detailed
descriptions of the NDMR network model, mobile node model and several
process models.

3.3.1 Network Model

The network model is shown in Figure 3.10. Previous work [6, 28, 103] in the
literature was 50 nodes, so the value is used here for comparison. The 50 mobile
nodes can move around a wide area. Nodes communicate over wireless links with
a transmission range of 250m.
3.3.2 Node Model

As shown in Figure 3.11, the NDMR node model simulates the protocol stack. Each node within the network is uniquely identified by its IP address. Below is the list of the different modules that make up the NDMR node model:

- *src* module: This is the packet source module. It generates packets according to specific packet size and inter-arrival distributions. Once generated, packets are sent to the immediate lower layer.

- *application* module: The application module sets a random destination address to the incoming packet from *src* module.

- *routing* module: This module is deployed to discover and maintain routing information of a mobile ad hoc network. Receiving a data packet from the application layer, the module firstly checks its route table. If there is a route
path towards the destination node in route table, the module forwards the data packet to the next node. Otherwise, the module executes the node-disjoint multipath routing algorithm (NDMR) to discover multiple route paths to a destination node.

![Node Model](image)

**Figure 3.11 Node Model**

- `wlan_mac` module: This module is an implementation of the IEEE 802.11[36] standard *medium access control* (MAC) protocol.

- `radio_tx` module: This module receives packets from `wlan_mac` module and sends these packets on the radio channel through antenna.

- `radio_rx` module: This module receives packets from the antenna and forwards packets to `wlan_mac` module.
- *antenna* module: This module sends and receives packets from the defined channel. The antenna is an isotropic pattern.

\[ \text{radio}_{\text{tx}} + \text{radio}_{\text{rx}} + \text{antenna} \] modules are implemented by OPNET to satisfy specifications of the IEEE 802.11 standard *physical layer*.

- *mobility* module: Each mobile node has a position and velocity and moves around on a wide area. This module performs the movement of the current node by changing its position according to the actual movement scheme.

### 3.3.3 Application Process Model

The main function of the application process (Figure 3.12) is to allocate a destination IP address for each incoming packet from *src* module.

**Inputs**

- The number of available flows within the network: This value is assigned at the simulation level and indicates the maximum authorized number of active source/destination pairs. Changing the number of available flows can result in a change of network load.

- Node’s communicator (enumerated values: either None, Random, or a specific Node’s address): this value is defined at the process level and indicates the state of activity of the current node. If the *communicator* attribute is set to None, the node is not allowed to communicate with other nodes. If the *communicator* attribute is set to a specific node’s address, then the node is only authorized to communicate with that same node. If *communicator* attribute is set to Random, the current node may (under the condition of flow availability) pick a random destination IP address and initiate a conversation with the corresponding node.
Figure 3.12 Application Process Model

Description

Each node, at the pre-init state, randomly picks a waiting period before transiting to the init state. The idea is to introduce some sort of differentiation between the existing nodes. Thus, as nodes consecutively transit into the init state, remaining flows are progressively granted to arriving nodes until no more flows are available. Of course, nodes with a communicator attribute set to a specific node’s IP address automatically transit to the init state (waiting period at the pre-init state equals 0) and are consequently guaranteed to occupy a free flow.

Once in the init state, each node checks its corresponding communicator attribute:

- If it is set to none, the current node automatically transits to the idle state and no actions are undertaken.

- If it is set to a specific node’s IP address, the current node occupies a flow by decrementing the number of available flows.
• If it is set to Random, the node first checks the number of remaining flows. If the number of remaining available flows is not equal to zero, the current node reserves a flow and picks up a random destination node as its communicator during the simulation time. The communicator attribute is then switched from Random to that specific IP address and the node transits to the idle state. If all flows have been reserved, the current node is not lucky and its communicator attribute is switched from Random to None.

The current node is in the idle state and can transit either to the rx (upon packet arrival from the route layer) state or tx (upon packet arrival from the source layer) state.

When a data packet is received from the src module, the current node checks its communicator attribute. At this stage, only two values are possible for the communicator attribute: None or a specific destination IP address. If the value is None, the current nodes silently discards the data packet and returns to an idle state. For the other case, the current node sends the data packet from src module and the destination IP address to routing module.

In the rx state, the current node has just received a packet from the routing module. As the packet has reached its destination, the current node simply destroys it. Also, if the received packet requires a response, the current node creates a new data packet and generates a service request primitive as described above.

### 3.3.4 Node-Disjoint Multipath Routing Process Model

The node-disjoint multipath routing process (Figure 3.13) implements the NDMR routing algorithm that is proposed in previous section.

**Inputs**

- Current node’s IP address
- Data buffer size: This variable indicates the maximum number of data packets waiting for routes that can be stored in the internal queue. If an extra packet is received while the buffer is full, the packet just received is discarded.

**Description**

*Init* state: This state consists of the initialization of the process model. User defined attributes are loaded and routing information tables are initialized (route table, reverse route table, source_broadcastID table, etc). The value of Time to Live (TTL) threshold is set to 12 (in AODV the value of TTL is set to 8 [4, 6]). The threshold value represents that the maximum transmission hop count is 12. The value of TTL can be adjusted according to different scenario situation. A self-interrupt is scheduled to initiate the first Hello Interval. Once the initialization step is accomplished, the process transits to the idle state.

![Figure 3.13 NDMR Routing Process Model](image)
*Recv_App* state: The *routing* process transits to this state when a service request is received from the application layer to transmit a data packet to a given destination. The current state first extracts the destination IP address from the received packet, then checks the route table. If a route path exists in its route table, the current state inserts the IP address of the next hop to the data packet and forwards it to next hop node. Otherwise, the current state saves the data packet to a waiting queue and initiates a route discovery process.

*Recv_Mac* state: This state receives the incoming packet stream from the MAC layer. It first checks the type of the received packet and then calls an appropriate function to proceed. If a packet has reached its final destination, the current state unencapsulates its payload and sends it to the application module.

*Rebroadcast* state: This occurs when a RREP_WAIT_TIMEOUT timer expires for a given destination. This means that the current node still did not receive a route reply to its request. In this case, the current state checks whether a rebroadcast is possible or not. If the maximum authorized number of retries is reached, the discovery process for that destination is aborted. Consequently, any data packet waiting for this route is dropped from the buffer. In the other case, a RREQ packet is rebroadcast.

*Update_Route_Table* state: This state occurs when the timer of an entry expires.

*Say_Hello* state: Node should broadcast a hello message in order to advertise its presence to the neighbourhood.

*Link_Break* state: When a node detects that there is a link break to a neighbouring node, it sends a RERR packet to its upstream node.

*Congestion_Notification* state: When a node detects traffic congestion, it sends a congestion notification packet to its upstream node.
Stat state: The current process periodically transits to this state in order to collect different global statistics. These statistics are written into a file which is created at the beginning of each simulation run.

### 3.3.5 Medium Access Control Model

OPNET supports models for simulating ad hoc wireless networks on physical layer model and medium access control layer model. The IEEE802.11 MAC protocol with Distributed Coordination Function (DCF) [36] is deployed as the MAC layer in the simulations reported here. DCF is the basic access method used by mobiles to share the wireless channel and avoid hidden and exposed terminator problems [58]. The access scheme is Carrier Sense Multiple Access/Collision Avoidance (CSMA/CA) with acknowledgements. The nodes can make use of Request To Send / Clear To Send (RTS / CTS) channel reservation control frames for unicast, virtual carrier sense, and fragmentation of packets larger than a given threshold. In the model, the RTS/CTS and virtual carrier sense are deployed to minimize the effect of collisions over the wireless medium.

### 3.3.6 Mobility Process Model

The mobility process model, shown in Figure 3.14, implements a random waypoint mobility scheme [7] that is described below. The literature in ad hoc networks generally uses this mobility model [103, 6, 28].

The general motion of a particular node is simulated through a set of discrete small step intervals. A node in motion updates its position every time step period of time. In the simulations here, the duration of each step was set to a value of 0.2 seconds.
Figure 3.14 Mobility Process Model

Inputs

- Mobility attribute (enumerated values: Enable, or Disabled): Indicates whether the current node is fixed or mobile.
- Grid dimensions: Each mobile node moves around the specified area.
- Speed limit: Maximum speed that a node in motion may reach.
- Pause time: After reaching a target position, a moving node must stop for a period of time. Then it begins to move towards a new target position.

Description

In the init state, each node picks a random position within the specified grid. After that, each node checks the mobility attribute in order to determine whether it should move or not: if the mobility attribute is set to Disabled, the current node
transits immediately to the *idle* state and remains at the same position during the whole simulation time. In the other case, the *mobility* attribute is set to Enabled, the current node transits to the *init_mvt* state in order to initialize its next movement parameters.

Basically, a moving node chooses a random target position within the grid and a random speed between 0 and the *speed limit* value. Given these two parameters, the moving node periodically (every step time period) transits from the *idle* state to the *move* state until it reaches the target position. While in the *move* state, a moving node progressively travels by a *step time * speed* amount of distance towards the target position. After each step movement, the node checks if it has reached its target or not. If it has, the current node transits to the *idle* state and enters a pause time phase. When the pause period finishes, the END_PAUSE condition becomes true and the current node transits to the *init_mvt* state in order to plan the next trip. On the other hand, if the target position has still not been reached yet, the current node returns to the *idle* state and waits for the next step time before returning to the *move* state.

### 3.4 Simulation Validation

In order to demonstrate that the simulation model and implementation of NDMR are performing correctly, some validations need to be carried out to verify and validate the simulation. OPNET ODB functionality is used to trace and monitor the process of the simulation. The overall simulation results and intermediate results can be achieved in traces, breakpoints and files. The validations are listed as follows and the results are “True” for all cases:

1. **Validation rule:** If a node prepares to send data packets to a destination and no route is available towards the destination, it creates a RREQ packet and broadcasts the RREQ packet to its neighbouring nodes.
2. **Validation rule:** The value of Time to Live (TTL) from any received RREQ packet should be less than TTL threshold value. Otherwise, the received RREQ packet should be discarded.

3. **Validation rule:** If the broadcast ID of a received RREQ is greater than the Request ID in its source_broadcastID table, the value of the Request ID should be updated into the value of the broadcast ID.

4. **Validation rule:** Intermediate node should append its node address to the route path and increment the value of TTL before it broadcasts a RREQ packet to its neighbouring nodes.

5. **Validation rule:** If the broadcast ID of a received RREQ is less than the Request ID in source_broadcastID table, the RREQ should be discarded.

6. **Validation rule:** If the hopcount of the route path of a received RREQ is less than or equal to the reverse shortest routing hopcount of reverse routing table, the intermediate node appends its address to the RREQ and forwards the RREQ.

7. **Validation rule:** If the hopcount of the route path of a received RREQ is greater than the reverse shortest routing hopcount of reverse routing table, the RREQ should be discarded.

8. **Validation rule:** When a RREQ reaches its destination and the broadcast ID in the RREQ is greater than Request ID of its source_broadcastID table, the destination creates a RREP and sends it towards source node along the reverse routing path.

9. **Validation rule:** When a RREQ reaches its destination node and the broadcast ID in the RREQ is equal to Request ID of its source_broadcastID table, the destination node judges whether the routing path is a node-disjoint path.
10. **Validation rule:** If the routing path of a received RREQ is a node-disjoint path, the destination node creates a RREP and sends it towards the source node along the reverse routing path. Otherwise, the RREQ is discarded.

11. **Validation rule:** An intermediate node records next hop towards destination to route table entry and next hop towards source to reverse route table entry before it forwards a RREP towards source node along the reverse route path.

12. **Validation rule:** When a RREP arrives in source node and the destination sequence number in the RREP is greater than the destination sequence number in route table, source node updates destination sequence number and records next hop towards destination node in its route table.

13. **Validation rule:** When a RREP arrives in source node and the destination sequence number in the RREP is equal to the destination sequence number in route table, the source node records next hop and its valid flag to its route table.

14. **Validation rule:** When a RREP arrives in source node and the destination sequence number in the RREP is less than the destination sequence number in route table, source node discards the RREP.

15. **Validation rule:** When an intermediate node receives a data packet, it extracts destination address from the data packet and checks its route table to get nexthop’s address and forwards the data packet to next hop node.

16. **Validation rule:** Every node sends HELLO packets to maintain local connectivity.

17. **Validation rule:** Failing to receive a HELLO packet from a neighbour, the upstream node of the link failure sends a RERR packet towards source.
18. **Validation rule:** When an intermediate node receives a RERR packet, it marks its route to the destination invalid and then propagates the RERR to its precursor node along the reverse route path.

19. **Validation rule:** After receiving a RERR, source invalidates the route path to destination and chooses a valid node-disjoint routing path as active routing path from routing table to forward data packets.

20. **Validation rule:** When there only remains a routing path available to send data packets in the route table of a source node, the source initiates a route discovery process to get a new set of multiple node-disjoint paths.

### 3.5 Simulation Environment Model

OPNET 8.1 Modeller [32] was used to create a simulation environment to develop and analyze the proposed NDMR and compare its performance with the already existing AODV and DSR on-demand unipath ad hoc routing protocols.

#### 3.5.1 Mobility and Traffic model

The mobility and traffic models similar to those previously reported [6, 38] are used. The random waypoint model [7] is used to model mobility of nodes. This model was first used by Johnson and Maltz in the evaluation of DSR, and was later refined by the same research group. Each node starts its journey from a random location to a random destination point with a specific speed. Once the destination is reached, another random destination point is targeted after a pause. Field configurations of 1000m x 1000m field with 50 nodes [28, 103] and 1500m x 1500m field with 100 nodes are used. The two field configurations have almost the same node density. Each node uses the IEEE 802.11 [36] with a 250m transmission radius. The pause time is kept constant at 30 seconds for all the simulation experiments.
Traffic sources with 512 bytes data packets (as used widely in the literature [6, 27, 28, 103]) are CBR (constant bit rate). The source-destination pairs are spread randomly over the network and the number of sources is varied to change the offered load in the network. The sending rate is set to 10 packets per second.

Nodes in all the three protocols maintain a send buffer which can contain 100 packets. Each node buffers all data packets while waiting for a route. All packets (both data and routing) sent by the routing layer are queued at the buffer until the MAC layer can transmit them. Routing packets are given higher priority than data packets in the buffer.

Simulations are run for 800 simulated seconds. Each data point represents an average of five runs with identical traffic models, but different randomly generated mobility scenarios by using different seeds. The maximum and minimum values are also shown on the graphs.

### 3.5.2 Performance Metrics

The following metrics [39] are used in varying scenarios to evaluate the three different protocols:

- **Packet delivery ratio:** The ratio of the data packets delivered to the destinations to those generated by the CBR sources.

- **Average delay of data packets:** This includes all possible delays from the moment the packet is generated to the moment it is received by the destination node.

- **Normalized routing load:** The number of routing control packets transmitted per data packet delivered at the destination. Normalized routing load gives a measure of the efficiency of the protocol.
3.6 Simulation Results

In order to compare and evaluate performances of the three protocols (NDMR, AODV and DSR) in different network conditions, three parameters are varied in the simulations:

- Number of nodes (50 nodes and 100 nodes)
- Maximum velocity of the nodes
- Maximum number of sources

At first, simulations are carried out by keeping the number of sources constant and varying the velocity in 50 nodes and 100 nodes networks. 20 and 40 sources are modelled respectively to study the effect of varying mobility in networks of 50 and 100 nodes. Then, the number of sources is varied from 10 to 50 in intervals of 10 for 50 nodes and from 10 to 100 for 100 nodes. When varying the number of sources, velocity is kept at a uniform rate of 0-20m/s.

3.6.1 Varying Velocity

The first set of experiments varies the velocity for 20 sources of 50 nodes network and 40 sources of 100 nodes network. The mobility was varied to see how it affects the different metrics that are measured. The packet sending rate is fixed at 10 packets / sec. The results are collected at constant speeds of 0, 1, 5, 10 and 20 m/s.

3.6.1.1 Packet Delivery Ratio

Packet delivery ratio is defined as ratio of the data packets delivered to the destination to those generated by the CBR sources. Packet delivery ratio is a very important metric since it shows the loss rate, which in turn affects the maximum throughput of the network. The packet delivery ratio of the three protocols is shown in Figure 3.15. The Figure depicts the variation of the packet delivery ratio
as a function of velocity of nodes. As the velocity of the nodes increases, the probability of link failure increases and hence the number of packet drops also increases. NDMR has much higher packet delivery ratio than both AODV and DSR. More than 90% data packets of NDMR can be delivered to specified destinations in all of mobility conditions in both 50-node and 100-node networks. AODV and DSR have a similar low delivery ratio situation in that only 50% sent packets are received at higher speeds. The reason is that NDMR has multiple paths with node-disjointness. When an active routing path is broken due to mobility of nodes, the source node of the data flow will receive a notification of link break. The source node at once invalidates the broken routing path in its route table and selects another valid node-disjoint routing path from its route table to continue to keep communication between source and destination without pause or interrupt. In addition, when there only remains a routing path available to forward data packets in the route table of a source node, the source will initiate a discovery process to get a new set of multiple paths. This mechanism of NDMR guarantees high packet delivery ratio. NDMR maintains a high packet delivery ration compared to AODV and DSR. This indicates that the robust nature of the protocol to mobility of nodes.
Figure 3.15 Packet Delivery Ratio vs. Velocity


3.6.1.2 Average end-to-end delay of data packets

The average end-to-end delay includes all possible delays from the moment the packet is generated to the moment it is received by the destination node. Generally, there are three factors affecting end-to-end delay of a packet:

(1) Route discovery time, which causes packets to wait in the queue before a route path is found;

(2) Buffering waiting time, which causes packets to wait in the queue before they can be transmitted;

(3) The length of routing path. The more number of hops a data packet has to go through, the more time it takes to reach its destination node.

Figure 3.16 depicts the variation of the average end-to-end delay as a function of velocity of nodes. It can be seen that the general trend of all curves is an increase in delay with the increase of velocity of nodes. The reason is mainly that high mobility of nodes results in an increased probability of link failure that causes an increase in the number of routing rediscovery processes. This makes data packets have to wait for more time in its queue until a new routing path is found. The delay of NDMR remains approximately equal at all mobile velocities. Delay in DSR and AODV increases quickly as velocity increases. When the velocity is more than 10m/s, the delay in NDMR is almost half of that in AODV and DSR. This is because availability of alternate node-disjoint routing paths in NDMR eliminates route discovery latency that contributes to the delay when active route fails. In addition, when a congestion state occurs in a routing path, the source node can distribute incoming data packets to the other node-disjoint routing paths to avoid congestion. This reduces the waiting time of data packets in queue.
Figure 3.16 Average Delay vs. Velocity
3.6.1.3 Normalized routing load

Normalized routing load can be measured by the number of routing control packets transmitted per data packet delivered at the destination. Normalized routing load is an important metric to compare the performance of different protocols since it can give a measure of the efficiency of protocols, especially in a low bandwidth and congested wireless environment. Protocols that transmit a large number of routing packets can also increase the probability of packet collisions and waiting time of data packets in transmission buffer queues. Figure 3.17 presents the normalized routing load characteristics of the 50-node and 100-node networks. It can be seen that the normalized routing load in NDMR performs much better than that of both AODV and DSR. The metrics increases slowly with the increase of velocity. The normalized routing load in AODV and DSR increases more quickly than that in NDMR with the increase of velocity. There are three reasons for this result:

1. NDMR can find multiple node-disjoint route paths in a route discovery process, so the protocol decreases tremendously the number of route rediscovery process. On the contrary, since AODV and DSR encounter more link failures with the increase in mobility, they have to trigger more new route discovery processes which cause more routing control packets to be sent to the whole networks.

2. NDMR reduces dramatically the number of control packets by using the shortest routing hops of loop-free paths concept to search multiple routing paths.

3. NDMR has higher packet delivery ratio than AODV and DSR in high mobility of nodes.
Figure 3.17 Normalized Routing Load vs. Velocity
3.6.2 Varying Number of Sources

The second set of experiments varies the number of sources with a random velocity of 0-20 m/s for 50 and 100 nodes. The network load is varied by changing the number of sources. The packet sending rate is still fixed at 10 packets / second. The number of sources is varied from 10 to 50 in intervals of 10 for 50 nodes and from 10 to 100 for 100 nodes.

3.6.2.1 Packet Delivery Ratio

The packet delivery ratio of the three protocols is shown in Figure 3.18. The Figure describes the variation of the packet delivery ratio as a function of the number of sources. It can be seen that the packet delivery ratio for NDMR has much better performance than those of both AODV and DSR with the increase in the number of sources. When the number of sources increases, AODV and DSR drop a larger fraction of the packets. Although the delivery ratio of NDMR is more than 80%, it decreases more quickly with larger numbers of sources. The reason is that there are more collisions in the air and congestion in node buffers when the number of sources increases.

3.6.2.2 Average end-to-end delay of data packets

Figure 3.19 depicts the variation of the average end-to-end delay as a function of the number of sources. It can be seen that NDMR has a lower average delay than both AODV and DSR under almost all possible numbers of source. The primary reason is that the number of route discoveries is reduced in NDMR. Although NDMR has a low number of route discoveries, its delay also increases gradually with the increase of number of source. The reason is that increase of the numbers of sources leads to higher network load traffic in the ad hoc networks. Because of the limitation of a constrained wireless bandwidth, packets that will be sent or forwarded have to stay in buffers and wait for a longer time to get a radio channel available in order to avoid collisions in the air.
3.6.2.3 Normalized routing load

The normalized routing load of the three protocols is shown in Figure 3.20. The Figure depicts the variation of normalized routing load as a function of the number of sources. With the increase of the number of sources, the probability of packet collision and packet congestion increases. This leads to the increase of normalized routing load. It is seen that NDMR has much lower normalized routing load than both AODV and DSR in all possible numbers of sources. The normalized routing load in AODV and DSR increases more quickly than that in NDMR with the increase of the number of sources. The reason is that NDMR has multiple node-disjoint paths. When an active routing path encounters packet congestion due to high network load traffic, the source node will receive a congestion notification. To avoid the loss of data packets, the source node can at once select another valid node-disjoint routing path from its route table to send data packets towards the destination.
Figure 3.18  Packet Delivery Ratio vs. Number of Sources
Figure 3.19 Average Delay vs. Number of Sources
Figure 3.20 Normalized Routing Load vs. Number of Sources
In order to compare network performance between NDMR and AOMDV [27], the simulation environment and assumptions of AOMDV from [27] is used with a simulation of NDMR. The random waypoint model is used to model mobility. The pause time is set to zero. A 100 node network in a field with dimensions 2200m x 600m is used. Traffic sources are CBR. The source-destination pairs are spread randomly over the network. Only 512 byte data packets are used. Simulations are run for 500 simulated seconds. The maximum speed of the nodes is varied from 0 m/s to 30 m/s to change mobility of nodes. The number of sources and packet rate are fixed at 25 and 4 packets / sec, respectively. These values are the same as reported in [27].

Figure 3.21 shows the performance comparison results between NDMR and AOMDV. The blue curves are simulation results of AOMDV and AODV based on ns-2 from [27]. The red curves are simulation results of NDMR and AODV based on the OPNET Modeller. Because the simulation environments are not completely identical, plotting AODV results from both allows the common baseline to be compared.

Figure 3.21(a) shows the comparison results of packet delivery ratio. The results show that they have the same trend: as the velocity of the nodes increases, the probability of link failure increase and hence the number of packet drops also increase. The figure also shows that NDMR has higher packet delivery ration than AOMDV. The reason is that NDMR can discover multiple node-disjoint route paths and AOMDV can only obtain link-disjoint paths. This makes the mobility of nodes have less effect on NDMR. The source node of NDMR can forward data packets to another routing path when it receives a notification of a route link break. To reduce the pause or interrupt of communication between source and destination, the source node in NDMR will reinitiate a discovery process to get a new set of multiple node-disjoint paths when there only remains a
routing path available to forward data packets. These mechanisms of NDMR improves packet delivery ratio.

Figure 3.21(b) shows the comparison results of average end-to-end delay of data packet transfer. The figure shows that AOMDV and NDMR provide smaller end-to-end delays than AODV. This is because the NDMR and AOMDV have backup routes and need smaller route discovery overheads. Another observation is that NDMR has lower average delay than AOMDV. The reason is because NDMR take a measure to decrease dramatically the number of routing request packets and save wireless network bandwidth. This results in the decrease of waiting time of data packets in the transmission buffer of all nodes.

Figure 3.21(c) shows the comparison results of normalized routing load. It can be seen that the normalized routing load in NDMR performs better than of AOMDV. The reason is that NDMR reduces dramatically the number of control packets by recording the shortest routing hops of loop-free paths to search multiple node-disjoint routing paths.
Figure 3.21 Performance Comparisons between NDMR and AOMDV

(a) Packet Delivery Ratio

(b) Average Delay

(c) Normalized Routing Load
3.7 Summary

In this chapter, a novel and practical Node-Disjoint Multipath Routing (NDMR) protocol with low routing overhead is proposed in mobile ad hoc networks. The protocol can reduce routing control overhead dramatically and achieves multiple node-disjoint routing paths. The simulation model of NDMR in OPNET simulator is also implemented. Performance results for unipath routing protocols (AODV and DSR) and Node-Disjoint Multipath Routing Protocol (NDMR) are compared in different scenarios. Simulation results show that performance of NDMR is much better than those of AODV and DSR.

There is no QoS support in NDMR. A new QoS routing protocol will be presented in the next chapter.
Chapter 4 Multipath QoS Routing for supporting DiffServ

Future mobile Ad hoc networks (MANETs) are expected to be based on all-IP architecture and be capable of carrying multitude real-time multimedia applications such as voice and video flows as well as data flows. It is necessary for MANETs to have an efficient routing and quality of service (QoS) mechanism to support diverse applications. Providing multipath routing is beneficial to avoid traffic congestion and frequent breaks in communication due to mobility in MANETs. Differentiated Services (DiffServ), which have simple, efficient and scalable characteristics, can be used to classify network traffic into different priority levels and apply priority scheduling and queuing management mechanisms to obtain QoS supports. In this chapter, a node-disjoint Multipath QoS Routing protocol for supporting DiffServ (MQRD) is proposed. Simulation results show that MQRD achieves better performance in terms of packet delivery ratio and average delay.

4.1 Introduction

With the growth in potential use of MANETs, a lot of research is being focused on providing QoS [40, 41, 60, 61, 66, 72, 73, 74]. There are a number of technical challenges because of the network restrictions such as dynamically and unpredictably variable topology resulting from nodal mobility and bandwidth constraints caused by the shared wireless medium.

For the current Internet there are two different models to obtain a QoS guarantee: the Integrated Services (IntServ) [42] and Differentiated Services (DiffServ) [43]. IntServ uses the RSVP protocol [44, 45] to carry the QoS parameters from the sender to the receiver to make resource reservations along the path. IntServ/RSVP provides for a rich end-to-end QoS solution, by way of end-to-end signalling, state-maintenance (for each RSVP-flow and reservation),
and admission control at each network element. DiffServ on the other hand, does not have any end-to-end signalling mechanism and works on a service level agreement between the provider and the user. All packets from a user are marked to specify the service level and are treated accordingly. Multiple flows in DiffServ model are mapped to a single service level and state information about every flow need not be maintained along the path.

The IntServ-based model on per-flow resource reservation is not particularly suitable for MANETs because of the frequently changing topology and limited resources in MANETs, resulting in more signalling overhead and unaffordable storage and computing power for mobile nodes. However, the DiffServ-based approach is a lightweight model using a relative-priority scheme to soften the requirements of hard QoS models like IntServ. The service differentiation is based on per-hop behaviours (PHBs), so no flow states need to be maintained within the network. Thus the model could be a potential QoS model in MANETs.

Multipath routing allows the establishment of multiple paths between a single source and single destination node during a single route discovery. Some multipath routing protocols [9, 26, 28, 29, 71] in MANETs have been proposed to provide load balancing, fault-tolerance and higher aggregate bandwidth as well as eliminate route discovery latency after a link break by making use of the availability of multiple route paths. However, these multipath routing protocols lack QoS support in the process of transmission of data packets.

This chapter presents a node-disjoint Multipath QoS Routing protocol for supporting DiffServ (MQRD) [11], which makes DiffServ readily available over a multipath routing protocol, Node-Disjoint Multipath Routing (NDMR) [9], for QoS support in MANETs.

The chapter is organized as follows. In section 2, an overview of QoS Models is presented. Section 3 gives a simple overview about existing QoS Models for MANETs. Section 4 presents MQRD QoS model for MANETs. In section 5 a
 simulation model based on OPNET is proposed. Performance evaluation and comparison of NDMR and MQRD are presented in Section 6 and concluding remarks are made in Section 7.

4.2 QoS Models

RFC2386 [52] characterizes Quality of Service (QoS) as a set of service requirements to be met by the network while transporting a packet stream from the source to the destination. For the current Internet there are two different models to obtain a QoS guarantee: the Integrated Services (IntServ) [42] and Differentiated Services (DiffServ) [43]. In the section the basic concepts of the two models are introduced.

4.2.1 Integrated Services (IntServ)

The Integrated Services (IntServ) approach [42] aims to provide applications with a guaranteed share of bandwidth. IntServ operates on a per-flow basis, and the requested QoS for a flow is either fully granted or denied.

Three main services are provided to applications: (1) Guaranteed services [53] provide an assured amount of bandwidth, strict end-to-end delay bounds, and minimal queuing delay to packets, (2) Controlled load services [54] give a service that is as close as possible to a best-effort service in a lightly loaded network, and (3) Best effort services are characterized by the absence of a QoS specification. The first two services classes use parameters, such as token bucket rate and size, peak data rate, and minimum and maximum packet size. These provide detailed information about the intended packet stream, so that routers are able to produce detailed reservations.

The IntServ approach assumes that an explicit setup mechanism is used to convey resource requests to routers so that they can provide the requested services to flows that require them. Moreover, the signaling must establish and keep the reservation state in order to guarantee the resources promised. Resource
Reservation Protocol (RSVP) [44] can be used to create and maintain the required flow-specific states in network elements allowing them to provide the requested services. RSVP is a signaling protocol that applications may use to reserve resources for all kinds of flows in an IP network. The network routers respond by explicitly admitting or rejecting RSVP requests.

The main message types in RSVP are the Path message, which is transmitted by the sender to initialize a new flow, and the Resv message, which comes back upstream to the sender, applying the actual resource reservations at the routers. The sender includes the wanted QoS with the Path message, which causes the Path-state to be initialized at every RSVP-aware router receiving the message. The Resv message follows exactly the same route as the Path message and sets the reservation if possible. The Path and Resv messages are refreshed periodically, and if a router does not receive a refreshing message within a specified time, it will remove the reservation state and the allocated resources.

The components of integrated services are admission control, classifier, packet scheduler, and resource reservation protocol. Admission control implements the decision algorithm that a router or host uses to determine whether a new flow can be granted the requested QoS without impacting earlier guarantees. When a host requests a real-time service along some path, the admission control is invoked at every intermediate node to make a local accept/reject decision. The classifier determines what classes the packets should be placed in based on the information in the header, such as IP and port source and destination addresses. The purpose of the classifier is to map each incoming packet into some class for the purpose of traffic control. All packets in the same class get the same treatment from the packet scheduler. The packet scheduler manages the forwarding of different packet streams using a set of queues and other mechanisms such as timers. The scheduler determines the order in which the packets should be serviced by placing them into priority queues. Resource reservation protocol (RSVP) is used as an IP signaling protocol to create and maintain the state in the routers along the path of a flow.
4.2.2 Differentiated Services (DiffServ)

DiffServ [43] does not define any signalling mechanisms, but instead, it provides QoS by dividing traffic into a small number of classes and allocating network resources on a per-class basis. The class is marked directly on the packet, in the 6 bit DiffServ Code Point (DSCP) field. The DSCP field is part of the original type of service (ToS) field in the IP header. The IETF redefined the meaning of the little-used ToS field, splitting it into the 6-bit DSCP field and a 2-bit unused field. The unused field is being allocated to the Explicit Congestion Notification (ECN) mechanisms [55], as shown in Figure 4.1.

![TOS and DSCP + ECN](image)

**Figure 4.1 TOS and DSCP + ECN**

The basic goal of the Differentiated Services architecture is to meet the performance requirements of the users. Different traffic classes have different priority levels and scheduling algorithms have to ensure high priority packets are forwarded before low priority ones.

The DSCP determines the QoS behaviour of a packet at a particular node in the network. This is called the per-hop behaviour (PHB) and is expressed in terms of the scheduling and drop preference that a packet experiences. From an implementation point of view, the PHB translates to the packet queue used for forwarding, the drop probability in case the queue exceeds a certain limit, the resources (buffers and bandwidth) allocated to each queue, and the frequency at which a queue is serviced. The IETF defined a set of 14 standard PHBs as follows:
- **Expedited forwarding (EF)** [56, 59]. Traffic encounters minimal delay and low loss. From a practical point of view, this means a queue dedicated to EF traffic for which the arrival rate of packets is less than the service rate, so delay, jitter and loss due to congestion is unlikely. Voice and video streams are typical examples of traffic mapped to EF: they have constant rates and require minimal delay and loss.

- Twelve **assured forwarding** (AF) PHBs [57]. Each PHB is defined by a queue number and a drop preference. The IETF recommends using four different queues with three levels of preference each, a total of twelve distinct AF PHBs. The convention for naming the AF PHBs is AFxy, where x is the queue number and y is the level of drop preference. All packets from AFxy will be put in the same queue for forwarding, so that packets from an application cannot be reordered if they differ only in the drop preference. The AF PHBs are applicable for traffic that requires rate assurance but that does not require bounds on delay or jitter.

- **Best effort** (BE). There is no guarantee for QoS. Traffic receives no special treatment. Every packet gets the service that the network is able to provide.

DiffServ provides differential forwarding treatment to traffic, thus enforcing QoS for different traffic flows. It is a scalable solution that does not require per-flow signalling and state maintenance.

DiffServ is a fully distributed and stateless model. No state information is required to be maintained at any node. The model aims at pushing the complexity to the edge nodes of the network so that the process in intermediate nodes can be as simple and fast as possible. Instead of providing QoS at per flow granularity, DiffServ differentiates the traffic into a fixed number of classes.
4.3 Existing QoS Models for MANETs

The current existing solutions for QoS provisioning in MANETs are mainly based on the IntServ or DiffServ model. AQOR [62] uses a reservation-oriented method to decide admission control and allocate bandwidth for each flow. FQMM [63] is designed to provide QoS in ad hoc networks by mixing the IntServ and DiffServ mechanisms. High priority applications are provided by IntServ per-flow QoS guarantee, while lower priority applications are provided with per-class differentiation based on DiffServ. INSIGNIA [64] employs an in-band signalling protocol rather than out-of-band signalling protocol like RSVP to decrease reservation overhead. SWAN [65] is based on a reservation-less approach. By avoiding signalling, it simplifies the whole architecture and provides a differentiation between real-time and best effort in spite of not being able to guarantee the QoS needs of each flow for the whole session due to frequently changing topology and limited wireless bandwidth restriction.

The methods in the following section 4.3.1 – 4.3.4 represent the most practical solutions so far.

4.3.1 Ad hoc QoS on-demand routing (AQOR)

AQOR [62] uses a reservation-oriented method to provide QoS guarantees. The protocol provides a strategy for dynamically constructing paths between mobile nodes that form a MANET. The signalling of AQOR allows for both route discovery and end-to-end QoS reservation (minimum bandwidth and maximum delay). AQOR developed detailed computations to estimate the available bandwidth and end-to-end delay in unsynchronized wireless environment. By using the proposed mechanisms it is possible to make an admission control of flows based on the available resources (bandwidth and end-to-end delay), and to easily apply fast recovery on QoS violation situations.
The protocol works in several ways to allow QoS routing: neighbour discovery and maintenance, route exploring, route registering (for explored routes), a bandwidth reservation mechanism based on the arrival of the first packet of a flow, releasing of registered resources (but not reserved), a loop-free routing mechanism and the already mentioned mechanisms for admission control and bandwidth calculation.

4.3.2 Flexible QoS Model for MANETs (FQMM)

Flexible QoS Model for MANETs (FQMM) [63] is to combine knowledge from the solutions offered in the wire-based networks and apply them to a new QoS Model that will consider the characteristics of MANETs. The basic idea is that it uses both the per-flow state property of IntServ and the service differentiation of DiffServ. In other words, this model proposes that highest priority is assigned per flow provisioning and other priority classes are given per-class provisioning. This model is based on the assumption that not all packets in the network are actually seeking the highest priority because then this model would result in a similar model with IntServ where we have per-flow provisioning for all packets. FQMM model has the following features: nodes have dynamic roles, a hybrid provisioning scheme that combines the per-flow granularity in InServ and per-class granularity in DiffServ, and a relative and adaptive traffic profile to maintain consistent differentiation between traffic types and keep up with the dynamics of the network. In FQMM, both the wired schemes are used separately for different priority classes. Therefore, the drawbacks related to IntServ and DiffServ remain in this model.

4.3.3 SWAN

The SWAN model [65] was developed by the Comet team at Columbia University. The model differentiates traffic into real time UDP traffic and best effort TCP traffic. It is a stateless and fully distributed model that provides soft QoS assurances to real-time traffic. It uses admission control for real-time traffic,
rate control of TCP traffic and _ECN congestion control mechanisms_ to ensure that real-time packets meet QoS bounds. Each node comprises an admission controller that maintains information about the status of the outgoing link in terms of the available bandwidth and amount of congestion. It does this by promiscuously listening to all packet transmissions within its range. The admission controller located at the source node sends a _probe message_ toward the destination when a new real-time flow requires servicing. The probe message returns carrying the value of the bottleneck bandwidth along the path. If this value is greater than the requirements plus a threshold value, the flow will be admitted. Otherwise it is rejected and marked as best effort. All TCP flows are considered as best-effort. The best-effort traffic passes through a _rate-controller_ that shapes the traffic according to the rate based on the feedback from the MAC layer. The admitted real-time traffic bypasses the rate controller and has a scheduling priority over best-effort traffic. The admitted real-time flows only have soft QoS assurances, so that some of the flows may be dropped or downgraded to best effort if network traffic conditions change due to rerouting of traffic.

### 4.3.4 INSIGNIA

INSIGNIA [64] is an _in-band signaling_ system that supports adaptive reservation-based services in ad hoc networks. Thus all the control information is carried within the header of the data packet itself, without the need for a separate control channel.

The signalling system supports a number of protocol commands that drive fast-reservation, fast restoration and end-to-end adaptation mechanism. These commands are carried in-band with the data and encoded using the IP option field in datagram. This in-band information is snooped as data packets traverse intermediate nodes/routers and used to maintain _soft-state reservations_ in support flows. To establish reservation-based flows between source-destination pairs, source nodes initiate fast reservations by setting the appropriate fields in the
INSIGNIA IP option field before forwarding packets. Reservation packets (i.e. data packet with the appropriate IP option set) traverse intermediate nodes, executing admission control modules, allocating resources and establishing soft-state reservation at all intermediate nodes between source-destination pairs. The reservations need to be periodically refreshed by the packets of the flows. In the event of a change in the path resulting from movement of the nodes, the first packet along the new path makes fresh reservations along this path thereby performing a fast restoration. Reservations made along the old path are removed on a timeout. Flows in the network are expected to be adaptive to bandwidth availability. A flow that was allocated a maximum amount of bandwidth initially could be downgraded to minimum amount or even to best-effort in the event of rerouting of a flow or if network conditions change. The source node continues to send packets with the reservation bit set until the destination node completes the reservation setup phase by informing the source node of the reservation status using a QoS reporting mechanism.

4.4 A New QoS Routing Protocol: MQRD

This section describes a novel QoS routing protocol that comes from the research presented in this thesis. The IntServ-based model on per-flow resource reservation is not particularly suitable for MANETs because of the frequently changing topology and limited resources in MANETs, resulting in more signalling overhead and unaffordable storage and computing process for mobile nodes. DiffServ is a lightweight model using a relative-priority scheme to soften the hard requirements of hard QoS models like IntServ. The service differentiation is based on per-hop behaviours (PHBs), so no flow states need to be maintained within the core of the network. Thus the model should be a potential QoS model in MANETs.

Although the NDMR protocol already presented in this thesis provides node-disjoint multipath routing with low route control overhead in MANETs, it is only a best-effort routing approach, which is not enough to support QoS. MQRD
combines the advantages of NDMR and DiffServ and makes it suitable for the environment of MANETs with QoS support.

### 4.4.1 Integration of NDMR and DiffServ

Both NDMR and DiffServ operate at the network layer, so it is easy to combine them. Although NDMR was designed without taking QoS into consideration, it and DiffServ can be complementary techniques that can be implemented in MANETs to support an end-to-end QoS solution. When used together, DiffServ provides the standardized QoS mechanisms and NDMR provides node-disjoint multipath routing techniques increasing the network resource optimization and decreasing routing overhead. In MANETs, the source node classifies data packets and then marks them with the corresponding DiffServ Code Point (DSCP). The intermediate mobile nodes use per-hop behaviour (PHB) to determine the scheduling treatment and drop probability for each packet.

### 4.4.2 QoS and Resource Management of MQRD

Effective QoS mechanism can be used to provide better service to certain flows in the environments of limited wireless bandwidth. In MQRD this is done by either raising the priority of a flow or limiting the priority of another flow. In order to support service differentiation, scheduling and queue management are thought to be two important aspects of resource management [67, 68, 69, 75]. The former is done by the scheduler which decides the opportunities of flows for link access and the latter holds the valid packets and, when necessary, drops some packets from the buffer in case of network congestion.

#### 4.4.2.1 Priority Scheduling

In MANETs, when a mobile node is receiving traffic faster than it can transmit, the node may buffer the extra traffic until bandwidth is available. Using a queuing algorithm to sort the traffic has been deployed to handle an overflow of
arriving traffic in wired networks. In MQRD, priority queuing is used to build a priority scheduler. The priority scheduler includes two queues: a high-priority queue and a low-priority queue. The high-priority queue must be emptied before packets are scheduled from low-priority queues.

Like most of the work in the literature, this research does not take into account the quality of the wireless link. However, underlying NDMR ensures that the network only includes links that are operational.

Although DiffServ has 14 classes defined, it usually provides support for only the two most common applications [103, 109,110]:

- Real-time (Voice and Video) traffic.
- Best effort data.

![Figure 4.2 Priority Scheduler](image)

Denote the two classes as A and B. Class A applications require generally low loss, low latency, low jitter and assured bandwidth service, so packets of class A are classified as Expedited Forward (EF) traffic. Class B is classified as Best Effort (BE) traffic which offers a lower priority service. The priority scheduler (see Figure 4.2) is designed to transmit any available Class A packets ahead of Class B packets. The priority scheduling is commonly used to support voice and video application as the bandwidth, delay and jitter requirements of voice and video packets are strict, and the quality must not be degraded by background data.
traffic. On the other hand, Class B packets are not sensitive to delay, as the application which they service are primarily HTTP and FTP sessions.

4.4.2.2 Queue Management

While the scheduling does play a big role in the QoS provided by the network, it is only effective if there is sufficient queue space to hold incoming packets. Because queues are not of infinite size, they can fill and overflow. When a queue is full, any additional packets cannot get into the queue and will be dropped. This is a tail drop. The issue with tail drops is that the router cannot prevent this packet from being dropped (even if it is a high-priority packet). So, the purpose of queue management is to make sure the queue does not fill up so that there is room for high-priority packets.

The random early detection (RED) algorithm [70] is implemented to avoid congestion before it becomes a problem. The minimum threshold specifies the number of packets in a queue before the queue considers discarding packets. The probability of discard increases until the queue depth reaches the maximum threshold. After a queue depth exceeds the maximum threshold, all other packets that attempt to enter the queue are discarded.

4.4.3 Load Balance and Congestion Avoidance

As mentioned before, MQRD can discover multiple node-disjoint route paths with low routing overhead, so it can provide load balancing and higher aggregate bandwidth. Load balancing function can be triggered to avoid congestion by spreading the traffic along multiple routes when the RED algorithm judges the queue depth to reach the maximum threshold at which the queue begins to discard packets. The mobile node needs to send a Congestion Notification packet (CN) to the source of the data packet along the reverse route path. When the source receives the CN, it distributes part of the traffic to the other node-disjoint routing paths. In this way congestion and bottleneck are avoided or alleviated.
4.5 Simulation Model

OPNET 8.1 Modeller [32] was used to create a simulation environment to develop and analyze the proposed node-disjoint Multipath QoS Routing protocol for supporting DiffServ (MQRD) and compare performances with NDMR, which does not take QoS into account.

4.5.1 Mobility and Traffic model

The random waypoint model [7] is used to model mobility. Each node starts its journey from a random location to a random destination with a random velocity of 0-20 m/s. Once the destination is reached, another random destination is targeted after a pause. A field configuration of 1000m x 1000m field with 50 nodes is used and each node uses the IEEE 802.11 protocol [36] with a 250m transmission radius. The pause time is kept constant at 30 seconds for all the simulation experiments. Traffic sources with 512 byte data packets are CBR (constant bit rate). The source-destination pairs are spread randomly over the network and the number of sources is varied to change the offered load in the network.

In order to investigate the usage of network ability, the number of EF ( Expedited forwarding) sources with 80kbit/s (20pkt/s) bandwidth requirement is varied from 5 to 20 in intervals of 5. 20 other nodes are randomly chosen to send background BE (Best Effort) traffic with 2 pkt/s. Simulations are run for 800 simulated seconds.

4.5.2 Performance Metrics

The following metrics are used in varying scenarios to evaluate the two protocols:

- **Packet delivery ratio:** The ratio of the data packets delivered to the destinations to those generated by the CBR sources.
• *Average delay of data packets:* This includes all possible delays from the moment the packet is generated to the moment it is received by the destination node.

### 4.6 Simulation Results

Comparing Figure 4.3 and Figure 4.4, it can be seen that the packet delivery ratio of MQRD has better performance than that of NDMR with the increase in the number of EF sources. In order to show clearly and compare simulation results of different type of packets, packet delivery ratios of EF packets, BE packets and ALL packets (combination of EF and BE packets) are depicted respectively in the two figures. In Figure 4.4 EF packets have higher delivery ratio than BE packets because priority scheduler is used in MQRD. When the number of EF sources increases, NDMR drops a larger fraction of the packets than that of MQRD. The reason is that there exists more congestion in mobile node buffers when the number of EF sources increases.

![Figure 4.3 Packet Delivery Ratio of NDMR](image-url)
From Figure 4.5 It can be seen that EF packets and BE packets in NDMR have little difference in End-to-End average delay. The reason is that there is no priority policy to deal with the incoming EF and BE packets in mobile nodes. Figure 4.6 shows that EF packets of MQRD has a much lower average delay than BE packets because priority scheduler in MQRD makes EF packets be forwarded more quickly. With the increase in the number of EF sources, the average delay of BE packets in MQRD increases more quickly than that of EF packets. The reason is that an increase in the number of EF sources leads to higher network load traffic. Because of the limitation of a constrained wireless bandwidth, BE packets that will be sent or forwarded have to stay in buffers and wait for a longer time to get a radio channel available than EF packets in order to avoid traffic congestions.
Figure 4.5 Average Delay of NDMR

Figure 4.6 Average Delay of MQRD
4.7 Summary

In this chapter, a new QoS provisioning, which makes DiffServ over node-disjoint multipath routing protocol for MANETs to overcome the shortcomings of the best-effort model, is proposed. A solution for a reliable multipath routing and resource management for QoS issues of real-time multimedia applications in ad hoc networks is also presented. The performance evaluation and comparison between NDMR and MQRD are studied by extensive simulations using OPNET Modeler. Simulation results show that MQRD achieves better performance than NDMR by providing end-to-end QoS support in MANETs. It can be concluded that MQRD has a good potential to serve as a QoS model to provide real-time multimedia applications under the dynamically changing environment of ad hoc networks.

In the next chapter a novel distributed cross-layer QoS protocol is proposed to make real-time flows obtain their desired service requirements.
Chapter 5 Distributed Cross-Layer QoS

5.1 Introduction

In the previous chapter, DiffServ is applied in the network layer of MQRD to obtain service differentiation. Flows are classified into real-time class and best effort class. Service differentiation is achieved by assigning different classes with different priority in buffer queues. Although MQRD may guarantee that real time flows has better service quality than best effort flows, it can not completely guarantee whether a real time flow can get its desired service requirement such as end-to-end delay and jitter. In addition, the medium access control (MAC) layer of MQRD can not distinguish real-time traffic and best effort traffic since it deploys IEEE 802.11 DCF mode [36] which has no support of service differentiation in the shared channel contention.

In this chapter, a novel Distributed Cross-Layer QoS (DCLQ) architecture is proposed to provide QoS guarantees for real-time flows in mobile ad hoc networks. Without any extra control overhead in network layer, DCLQ can schedule packets of real-time flows according to their per-hop QoS requirements. DCLQ implements per-hop delay QoS-aware priority scheduling and QoS consideration of MAC layer to ensure that real-time flows to achieve their desired service level.

The chapter is organized as follows. In section 2, an overview of IEEE 802.11 medium access control is presented. Section 3 gives details about the implementation of DCLQ. In section 4 a simulation model based on OPNET is proposed. Performance evaluation and comparison of MQRD and DCLQ are presented in Section 5 and concluding remarks are made in Section 6.
5.2 Medium Access Control

Before the cross layer QoS mechanisms of DCLQ are presented, it is necessary to know the classifications of MAC and to understand how IEEE 802.11 medium access control works. In this section, the mechanisms of IEEE 802.11 DCF and IEEE802.11e EDCF are introduced.

5.2.1 Classification of Medium Access Control Schemes

Various MAC schemes have been proposed in the literature for MANETs. Based on different techniques adopted, MAC schemes are classified into two categories: contention-free and contention-based.

5.2.1.1 Contention-free scheme

In contention-free scheme, the system time is synchronized and slotted. Each mobile node accesses the channel in a specific time slot, which is determined by a schedule. The schedule can be determined in a centralized or distributed manner. The protocols using contention-free scheme [76, 77, 78] are generally designed for a TDMA-based MAC layer. QoS scheduling is achieved by reserving dedicated time slots for real time flows according to their service requirements. Admission control is performed by looking for a sequence of free time slots, while ensuring that nodes in each other’s contention range are allocated with different time slots to avoid collisions. However, contention-free MAC requires effective time synchronization between all nodes in the network. Applying synchronization in an ad hoc network is expensive and synchronization can fail when the nodes are mobile.

5.2.1.2 Contention-based scheme

The contention-based scheme is based on random channel access techniques, such as CSMA/CA (Carrier Sense Multiple Access / Collision Avoidance). Since a wireless channel is a broadcast channel, when only one mobile node transmits,
it can successfully send its packets. When several neighbouring nodes transmit simultaneously, a collision will occur. When a node detects a collision, it tries to access the channel again after a random back off time. The protocols using contention-based scheme [79, 80] avoid the cost of time synchronization in MAC layer.

5.2.2 IEEE 802.11 Medium Access Mechanisms

The IEEE 802.11 wireless local area network (WLAN) is a shared-medium communication network that transmits information over wireless links for all nodes in its transmission range to receive. The 802.11 WLAN standard has been accepted widely and rapidly for many different environments. The main characteristic of the 802.11 is its simplicity, scalability, and robustness against failures due to its distributed nature. IEEE802.11 WLAN can be configured into two different modes: ad hoc and infrastructure. In ad hoc mode, all wireless stations within the communication range can communicate with each other, whereas in infrastructure mode, an access point (AP) is needed to connect all stations to a distribution system (DS), and each station can communicate with others through the AP.

IEEE802.11 standard actually include a family of standards. The original standard, IEEE802.11 provides data rates up to 2 Mb/s in the 2.4 GHz industrial, scientific and medical (ISM) band. Later, the IEEE802.11 working group published several enhanced versions. IEEE802.11b extends the data rate up to 11 Mb/s in the ISM band. IEEE802.11a can achieve a data rate up to 54 Mb/s using orthogonal frequency division multiplexing (OFDM). In order to support video, audio real-time voice over IP, IEEE802.11e [82] enhances the original 802.11 MAC [36] sub-layer to support QoS. The original 802.11 WLAN MAC sub-layer employs a distributed coordination function (DCF) based on carrier sense multiple access with collision avoidance (CSMA/CA) for medium access, and is best known for its asynchronous best effort data transfer. IEEE802.11e standard adds a new function called a hybrid coordination function (HCF) that includes
both controlled contention-free and contention-based channel access methods in a single channel access protocol. The HCF uses a contention-base channel access method called enhanced DCF (EDCF) that operates concurrently with a controlled channel access mechanism based on a central polling mechanism. HCF supports both prioritized and parameterized medium access.

This section will briefly review the main features and functions of the original 8.2.11 and 802.11e standards.

5.2.2.1 IEEE802.11 DCF

The IEEE 802.11 standard covers the MAC layer and the physical layer of OSI (Open System Interconnection) reference model. In this section, only MAC layer is discussed (the physical layer part is described in [36]).

The basic services provided by the MAC layer are the mandatory asynchronous data service and an optional time-bounded service. Due to non-centralized management mechanism, only the asynchronous service (contention-based) can be deployed in ad hoc networks. Both service types can be used simultaneously in infrastructure-based networks. The asynchronous service supports broadcast and multicast packets, and packet exchange is based on a best effort model, no delay bounds can be given for transmission.

The following three basic access mechanisms have been defined for IEEE 802.11: the mandatory basic method based on a version of CSMA/CA, an optional method avoiding the hidden terminal problem and exposed terminal problem in ad hoc networks, and finally a contention-free polling method for time-bounded service. The first two methods are also summarized as distributed coordination function (DCF), the third method is called point coordination function (PCF). DCF uses contention-based approach to access the medium and only provides asynchronous service. PCF provides both asynchronous and time-
bounded service but needs an access point to control medium access and to avoid contention.

As mentioned earlier, the DCF is an asynchronous data transmission function and it is the only possible function in ad hoc networks. Each node gets an equal share of the channel through contention. The basic scheme for DCF is Carrier Sense Multiple Access (CSMA) [83]. This protocol has two variants: Collision Detection (CSMA/CD) and Collision Avoidance (CSMA/CA). A collision can be caused by two or more stations using the same channel at the same time after waiting for the channel to become idle, or (in wireless networks) by two or more hidden terminals transmitting simultaneously.

![Diagram of the Basic Access Scheme](image)

**Figure 5.1 Basic Access Scheme**

CSMA/CD is deployed in Ethernet (IEEE 802.3) wired networks. Whenever a node detects that the signal it is transmitting is different from the one on the channel, it stops the transmission, saving useless collision time. This mechanism is not possible in wireless communications because a mobile node cannot listen to the channel while it is transmitting, due to the big difference between transmitted and received power levels. To deal with this problem, the sender should wait for
an acknowledgment (ACK) from the receiver after each frame transmission, as shown in Figure 5.1. If no ACK is returned, a collision must have occurred and the frame is retransmitted. The inter-frame spacing DIFS and SIFS will be explained later in the section.

In MANETs, when a node transmits packets, only neighbouring nodes that are within its transmission range can receive them. However, this characteristic leads to the hidden terminal and exposed terminal problems [108]. This is illustrated in Figure 5.2 where there are three nodes. The transmission range of node A can reach node B, but not node C; the transmission range of node C can reach node B, but not node A; the transmission range of node B can reach node A and node C. Both node A and node C cannot reach each other directly.

![Figure 5.2 Hidden and Exposed terminals](image)

- **Hidden Terminal Problem:**

  A begins sending packets to B, C cannot receive the transmission. C also wants to send a packet to B and senses the medium. Because the medium appears to be free, C starts sending the packet to B. This causes a collision at B. A cannot detect this collision at B and continues with its transmission. A is hidden for C and vice versa.
- **Exposed Terminal Problem:**

  When B is sending something to A and C wants to transmit data to some other node. C senses the carrier and detects that the carrier is busy. C has to delay its transmission until it detects that the carrier is idle again. Because A is outside the interference range of C, the waiting of C is not necessary. In this situation, C is exposed to B.

  To solve the hidden terminal and exposed terminal problems, an optional RTS/CTS (Request To Send and Clear To Send respectively) scheme is used in addition to the previous basic scheme, as shown in Figure 5.3: a node sends an RTS before each frame transmission to reserve the channel. Note that a collision of RTS frames (20 bytes) is less severe and less probable than a collision of data frames (up to 2346 bytes). The receiver replies with a CTS packet if it is ready to receive and the channel is reserved for the packet duration. When the sender receives the CTS, it starts transmitting its frame, being sure that the channel is reserved for itself during all the frame duration. All other nodes update their Network Allocation Vector (NAV) whenever they hear an RTS, a CTS or a data frame. Not only does the RTS/CTS scheme use physical carrier sensing, it also introduces the concept of virtual carrier sensing. This is implemented in the form of the NAV that is maintained by every node. The NAV contains a time value that represents the duration up to which the wireless medium is expected to be busy because of transmissions by other nodes. Since the packet contains the duration information for the remainder of the message, every node overhearing a packet continuously updates its own NAV.
Not all packet types have the same priority. For example, ACK packets should have priority over RTS or data frames. This is done by assigning to each packet type a different Inter Frame Spacing (IFS), after the channel turns idle, during which a packet cannot be transmitted. In DCF two IFSs are used: Short IFS (SIFS) and DCF IFS (DIFS), where SIFS is shorter than DIFS (See Figure 5.1 and Figure 5.3). As a result, if an ACK and a new data packet are waiting simultaneously for the channel to become idle, the ACK will be transmitted before the new data packet because the ACK only has to wait SIFS whereas the data has to wait DIFS.

Carrier sensing can be done on both the physical and MAC layers. On the physical layer, physical carrier sensing is performed by sensing any channel activity caused by other sources. On the MAC layer, virtual carrier sensing can be performed by updating a local NAV with the value of other terminals’ transmission duration. The duration is declared in data, RTS and CTS frames. From the local NAV a mobile node can know when the current transmission ends. NAV is updated upon hearing an RTS from the sender and/or a CTS frame from the receiver, so the hidden node and exposed node problems are avoided.
The collision avoidance part of CSMA/CA consists of avoiding packet transmission immediately after the channel is sensed idle for DIFS time, so it does not collide with other “waiting” packets. Instead, a wireless terminal with a packet ready to be transmitted waits the channel to become idle for DIFS time, then it waits for an additional random time, backoff time, after which the packet is transmitted, as shown in Figure 5.1 and 5.3. Each node chooses a random backoff time within a contention window and delays medium access for this random amount of time. The node continues to sense the medium. As soon as a node senses the channel is busy, it has lost this cycle and has to wait for the next chance, i.e., until the medium is idle again for at least DIFS. But if the randomized additional waiting time for a node is over and the medium is still idle, the node can access the medium immediately. The additional randomly distributed delay helps to avoid collisions. Otherwise, all nodes in a virtual carrier sensing area would try to transmit data after waiting for the medium becoming idle again plus DIFS. To provide fairness, each node selects a random waiting time within the range of the contention window. If a certain node does not access to the medium in the first cycle, it stops its backoff timer, waits for the channel to be idle again for DIFS and starts the counter again. As soon as the counter expires, the node accesses the medium. This means that deferred stations do not choose a randomized backoff time again, but continue to count down. Nodes that have waited longer have higher priority than nodes that have just entered, in that they only have to wait for the remainder of their backoff timer from the previous cycle.

If the packet collides with another frame, the node computes a new random backoff time with a higher range to retransmit the packet with lower collision probability. This range increases exponentially as $2^{2^i}$, where $i$ (initial equal to 1) is the transmission attempt number. Therefore, the backoff time equation is:

$$Backoff\_time = [2^{2^i} \times rand()] \times Slot\_time$$

Where Slot_time is function of physical layer parameters, and rand() is a random function with a uniform distribution in [0,1]. There is a higher limit for i, above
which the random range remains the same. The packet is dropped after a given number of retransmissions.

IEEE 802.11 DCF does not differentiate the data traffic. All node and traffic classes have the same priority to access the wireless medium. Thus, different delay and bandwidth requirements of applications are not supported with the use of DCF.

5.2.2.2 IEEE 802.11e EDCF

To support applications with QoS, IEEE802.11 working group developed IEEE802.11e [82, 84, 85], which enhances the original 802.11 MAC to support applications with QoS requirements. The basic approach of enhanced DCF (EDCF) includes two differences from DCF: (1) assignment of different $CW_{min}$ values to different priority classes, resulting in high priority traffic with smaller $CW_{min}$ values; (2) assignment of arbitration IFS (AIFS), instead of DIFS, to different traffic classes, resulting in high priority classes with smaller AIFS values.

The IEEE 802.11e standard adds a new medium access mechanism, the hybrid coordination function (HCF). HCF concurrently exists with basic DCF/PCF for backward compatibility and combines functions from the DCF and PCF with some enhanced QoS-specific mechanisms.

The EDCF in 802.11e is the contention-based medium access method for HCF. QoS support is realized with the introduction of traffic categories (TCs). The EDCF provides differentiated distributed access to the wireless medium for eight priorities of nodes. EDCF channel access defines the access category (AC) mechanism that provides support for the priorities at the nodes. Each station may have up to four ACs to support eight user priorities. Each AC is an enhanced variant of the DCF. It contends for transmission opportunities (TXOPs) using a set of EDCF channel access parameters. An AC with higher priority is assigned a
shorter contention window (CW) in order to ensure that, in most cases, a higher-priority AC will be able to transmit before lower-priority ones. This is performed by setting the CW limits CWmin[AC] and CWmax[AC], from which CW[AC] is computed, to different values for different ACs. For further differentiation, different interframe space (IFS) is introduced for different ACs. Instead of DIFS, an arbitration IFS (AIFS) is used. The AIFS is at least DIFS, and can be enlarged individually for each AC. Similar to DCF, if the medium is sensed to be idle in the EDCF mechanism, a transmission begins immediately. Otherwise, the station defers, until the end of current transmission. After deferral, the node waits for a period of AIFS (AC) to start a backoff procedure. The backoff interval is now a random number drawn from the interval \([1, CW(AC)+1]\). Each AC contends for access to the wireless medium and independently starts its backoff time after sensing the medium is idle for at least AIFS. Figure 5.4 illustrates the EDCF channel access.

![EDCF channel access scheme with different priority](image)

Figure 5.4  EDCF channel access scheme with different priority
5.3 Distributed Cross-Layer QoS (DCLQ)

By improving MAC layer performance and supporting service differentiation, many routing frameworks can provide QoS supports for ad hoc networks. For example, studies in [90, 95, 96, 100] propose to tune the contention windows sizes or the inter-frame spacing values to improve network performances, while studies in [79, 92, 98, 101, 102] propose priority-based scheduling to provide service differentiation. Most of these frameworks deploy different backoff mechanisms, different DIFS lengths, or different maximum frame lengths. Although they may guarantee that real time flows have better service quality than best effort flows, they cannot completely guarantee whether a real time flow can get its desired service requirement such as end-to-end delay and jitter.

In the section, a novel practical Distributed Cross-Layer QoS (DCLQ) framework is proposed to provide QoS guarantees to real-time multimedia traffic. The cross-layer QoS architecture uses distributed per-hop QoS-aware scheduling in the high priority queue of network layer and QoS consideration in MAC layer to ensure real-time flows to get their desired service requirements.

5.3.1 QoS challenges

The major challenge of providing multimedia services in ad hoc networks is that some QoS metrics must be satisfied. Much significant research on providing QoS in wired networks has been done, for example, IntServ and DiffServ. In ad hoc networks, two unique characteristics impose two major challenges for provisioning QoS: (1) the shared wireless medium and (2) mobility of nodes.

The first challenge is due to the shared nature of the wireless medium. In wired networks most of the QoS solutions rely on the availability of precise link utilization information. However, in ad hoc networks all traffic within a mobile node’s transmission range contends for medium access. Transmissions from a mobile node not only use local resources, but also consume the bandwidth of
neighbours in contention range. Therefore, medium access must consider not only the relevant service level of a flow, but also the impact of the flow on the neighbouring flows, which greatly add to the complexity of medium contention. In addition, for many widely available protocols, including IEEE 802.11, virtual carrier sensing is deployed to avoid the hidden and exposed terminal problems. In these protocols, the contention range of a node is equal to its carrier sensing range, which often is more than twice the transmission range. Therefore, how to improve the utilization ratio of the shared medium and to satisfy the requirements of low end-to-to delay to the real-time flows must be considered when designing a QoS architecture for ad hoc networks.

The second challenge is caused by the mobility of nodes, which results in dynamic changes of network topology. It also brings new obstacles to QoS provisioning in ad hoc networks. In general, node mobility has two impacts on network performance. First, the movement of a node on an active routing path can lead to a link break, and subsequent loss of packets. This effect is even more serious in a large scale network with long communication paths. After a link is broken, a new routing discovery process has to be initiated. This brings not only higher routing overhead consuming limited wireless bandwidth but also increased packet transmission delay. Therefore, reducing routing overhead and mitigating the number of routing are important to maintain the service quality. The second effect is dynamic change of load traffic due to node movement. Therefore, admission control and congestion control have to be considered to avoid congestion in ad hoc networks.

Many issues, such as the routing approaches, service model, admission control, resource reservation, packet scheduling and MAC protocols need to be considered in the context of providing QoS in ad hoc networks. Actually, every layer of ad hoc networks should be made QoS aware because only when all these components are considered together in the overall scenario can effective QoS be provided for the end-user applications.
The design of cross layer QoS-aware protocols under these challenges is not easy and requires perfect coordination between different layers of the protocol stack to support QoS for real-time traffic. Cross layered approaches are becoming more common in modern communication networking environments. Their main functions are to decrease design complexity by modularizing and to improve maintainability compared to monolithic stacks. Since each layer has a well defined functionality, designing each layer can be done without considering specific functionality of upper or lower layers. The modularity of cross layer makes the combination of some different protocols easier, thereby helping to construct network stacks tailored towards different networking environments.

5.3.2 Overview of DCLQ Architecture

Due to the lack of centralized control mechanism and the shared nature of the wireless medium in an ad hoc network, distributed medium access control must be used to allocate resources along the routes of flows to provide flow-based per hop QoS guarantees. Since there is no fixed infrastructure in an ad hoc network, every node in the network have to participate in the contention of the shared medium access and must be equipped with QoS support, which requires five necessary components: node-disjoint multipath routing with low routing overhead, admission control, QoS-aware priority scheduling, congestion control and distributed QoS MAC. In order to provide effective QoS guarantees in an ad hoc network, every node must implement the five necessary components

Figure 5.5 illustrates the Distributed Cross-Layer QoS (DCLQ) architecture, which includes network layer, MAC layer and physical layer.
Figure 5.5  DCLQ architecture
Applications can be categorized into real-time and best effort applications based on their sensitivity to packet delay. Real-time applications have strict requirements on the packet delay. Therefore, packet retransmission is not allowed. The applications that fit into this category are voice over IP, on-line movies and video conferencing. Many video compression technologies, such as MPEF-4, H.263 can compress video with different coding rates to meet different channel conditions.

After data flows enter the network layer, they are first classified into real-time flows or best effort flows. Then they enter multipath routing component to achieve multipath routing information towards the destination node. Before entering the QoS-aware scheduler, a flow must be checked by the admission control component to confirm whether or not there have enough resources to support the flow. Once admitted, a real-time packet can enter the QoS aware scheduler. According to the maximum delay of a real time packet and the number of hops along a chosen routing path, the delay allowed for every hop can be achieved. For example, the maximum allowed delay of a real time packet is 100 ms and the number of hops along a chosen routing path from source to destination is 5. The per-hop maximum allowed delay (waiting time) is 20ms. The real-time packet and its every hop waiting time are inserted into the high priority queue and wait for being scheduled to the MAC layer. Best effort traffic can be admitted to the low priority queue of QoS aware scheduler if the admission control component confirms that the low priority queue in the scheduler is not full. However, to guarantee the QoS requirements of existing admitted real-time flows, the best effort flows are rate-controlled to make sure they only use free bandwidth left by real-time traffic.

Distributed QoS MAC is based on an enhanced IEEE 802.11e EDCF protocol. It can provide service differentiation to different flows. Service differentiation is achieved by assigning different classes with different contention related parameters such as contention window size, frame size and inter-frame spacing. Once the MAC layer sends out a packet successfully, it will return at once a
control signal to the network layer to update the residual waiting time of all real-
time packets in the high priority packet queue. The priority scheduling chooses a 
real time packet whose residual waiting time is the smallest among all real time 
packets and sends the packet to MAC layer.

The congestion control component is unique to mobile ad hoc networks. Even 
if admission control is performed to guarantee enough available bandwidth before 
accepting any real-time flow, the network can still experience congestion due to 
mobility or connectivity changes. Therefore, congestion control is extremely 
important to the QoS architecture here. It monitors the network bandwidth 
utilization continuously and processes upcoming link congestion.

5.3.3 Distributed QoS MAC

Distributed QoS MAC in DCLQ is based on IEEE 802.11e EDCF mode with 
simple modifications. In this section, IEEE802.11 DCF mode and Service 
differentiation are briefly reviewed. Service differentiation is achieved by 
assigning different classes with different contention related parameters such as 
contention window size, frame size and inter-frame spacing.

IEEE 802.11 DCF Mode:

In IEEE 802.11 DCF mode, the transmission of each packet invokes an RES-
CTS-DATA-ACK handshake between the sender and the receiver to avoid the 
hidden and exposed terminal problems. A node desiring to transfer a data packet 
first invokes the carrier sense mechanism to determine the busy / idle state of the 
medium. If the medium is idle, the node defers a DCF inter-frame space (DIFS). 
If the medium is still idle after the DIFS period, the node may transmit its RTS 
packet. If the medium is busy, the node waits until the medium is determined to 
be idle for DIFS time. After this DIFS idle time, the node defers for an additional 
backoff period before transmitting an RTS. If the backoff timer is not yet set, the 
backoff period is generated as $\text{BackoffTime} = \text{Random()} \times a\text{SlotTime}$, where 
$\text{Random()}$ is a pseudo random number uniformly distributed between 0 and
contention window (CW). CW is an integer within the range of values of the PHY characteristics CWmin and CWmax. For 802.11b, CWmin = 31 and CWmax = 1203. aSlotTime is a very small time period (9µs). The backoff time is decremented by aSlotTime if the channel is idle during this period and stopped when a transmission is detected on the channel. The backoff timer is reactivated when the channel is sensed idle again for more than DIFS time units. The node transmits when the backoff timer reaches zero. After each failed transmission attempt, the contention window size is doubled to avoid congestion.

The backoff process in IEEE 802.11 provides a simple mechanism to minimize collisions during contention between multiple nodes by spreading out their channel access attempts. During the contention, the node with the smallest backoff time always wins the next transmission right.

*Service differentiation:*

During contention for the channel, the node with the smallest backoff time always wins the right to use the channel, the backoff process providing a distributed method to differentiate the service that a node receives. By decreasing the contention window size, a node essentially decreases its average backoff time and hence increases the chances that it wins the channel when competing with other nodes, affecting the node’s service quality in terms of bandwidth and packet delay. Based on the proportional relationship between contending nodes’ contention window sizes and services in terms of bandwidth and delay, IEEE 802.11e allocates different contention window sizes to different classes of traffic [86, 87, 94] so that class-based proportional fairness can be achieved.

In order to improve the utilization ratio of medium and transmission efficiency, the contention window (CW) parameter takes an initial value of CWmin. The CW will take the next value in the series after each unsuccessful transmission until the CW reaches the value of CWmax. Once it reaches CWmax, the CW remains at the value of CWmax until it is reset. This improves the transmission
efficiency under high load conditions. The CW is reset to CWmin after each successful attempt to transmit a packet.

### 5.3.4 QoS-aware Priority Scheduler

The QoS-aware Priority Scheduler of DCLQ can guarantee the QoS of admitted real-time flows under the condition that the capacity of the network is larger than the requirements of all admitted flows. The design of the QoS-aware priority scheduler is based on IEEE 802.11e, since it is simple and robust without requiring any centralized control and is well standardised. This section gives more details about the design of the QoS-aware scheduler and shows how it can guarantee that the admitted real time flows achieve their required QoS.

From Figure 5.5, the QoS-aware Scheduler includes a high priority packet queue used to buffer real-time flows and a low priority packet queue used to buffer best effort flows. The scheduler deploys priority scheduling approach to guarantee that admitted real time flows receive their desired services and control the rate of best effort flows so that they fill the bandwidth left by real time flows. Priority Scheduling first schedules real time packets of the high priority queue if the high priority queue is not empty; only when the high priority queue is empty, priority scheduling schedules best effort packets of low priority queue.

Since flows may have different QoS requirements based on the type of application data carried in the flow, DCLQ classifies flows into two types: real time flows and best effort flows. The real time flows, also called delay-sensitive flows, such as conversational audio/video conferencing, require that packets arrive at the destination within a certain delay bound. The best effort flows, such as HTTP and FTP sessions can adapt to changes in bandwidth and delay.

For a real time flow, the main QoS requirement is end-to-end packet delay. To control delay, the end-to-end delay requirement \( d \) is broken down into a per-hop delay requirement. Each hop locally limits packet delay below its per-hop requirement to maintain the aggregated end-to-end delay below \( d \). Every node
uses the same per-hop delay requirement, \( d/m \), where \( m \) is the hop count of the flow. When a source node prepares to communicate with a destination node, it chooses a right route path from multipath route table and calculates the number of hops to destination node along the route path. The source can get the per-hop delay requirement by breaking down the end-to-end delay requirement \( d \) to each hop. The allowed waiting time in every node should be below its per-hop delay requirement. Before the real-time packet is sent out, the source node should append a per-hop delay requirement field to the real-time packet so that every intermediate node knows the per-hop delay requirement. Before the real-time packet enters the high priority packet queue of QoS aware Scheduler, the per-hop delay requirement should be extracted from the real-time packet and be assigned to its residual waiting time field. In order to monitor the waiting state of all real-time packets in high priority queue, the residual waiting time field of every real-time packet should be updated frequently. Once the MAC layer sends out a packet successfully, it will return at once a control packet to the network layer to update the residual waiting time of all of real-time packets in the high priority packet queue by cutting down their waiting time. Then priority scheduling of QoS aware Scheduler extracts a real-time packet whose residual waiting time is the smallest and sends the packet to MAC layer.

By the scheduling approach the real-time packet corresponding to the smallest residual waiting time is always first scheduled to MAC layer to ensure its per-hop delay requirement.

5.3.5 Node-disjoint multipath routing

Providing multipath routing is beneficial to avoid traffic congestion and frequent breaks in communication due to mobility in MANETs. How to decrease routing overhead and avoid the broadcast storm problem which consumes the limited wireless bandwidth in process of routing discovery has to be considered in designing ad hoc route algorithms. Chapter 3 gives more details on why and how to implement the component.
5.3.6 Admission Control

With the supports from the multipath routing and congestion control, admission control component of source node can decide whether to admit a new real-time flow. Admission control component must ensure that the total resource requirements of admitted flows can be handled by the network. If there are not enough resources for admitted flow packets in a route path, some of packets may be distributed to multiple route paths. If the resource requirements of all of admitted flows are greater than network capacity, some of low priority flows have to be rejected to maintain the guarantees made to other flows.

When a new real-time flow with certain delay requirement arrives, the admission control component of the source node first consults the routing table and congestion control component. If there is an effective routing path to the destination in the routing table and the high priority packet queue of QoS aware scheduler is not full and the buffering delay of current sent real-time packet is smaller than the per-hop delay requirement of the new real-time flow, the real-time flow is accepted. Otherwise, the real time flow is rejected.

When a new best effort flow arrives, the admission control component of the source node first consults the routing table and congestion control component. If there is an effective routing path to destination in routing table and low priority packet queue of QoS aware scheduler is not full, the best effort flow is accepted. Otherwise, the best effort flow is rejected.

5.3.7 Congestion Control

In mobile ad hoc networks, admission control in source nodes can not guarantee QoS since the network topology is often changed due to mobility of nodes after flows are admitted. Network congestion can still occur frequently under mobility. Therefore, congestion control is very important to the QoS architecture here. When network congestion occurs, best effort traffic is expected to first reduce their transmission rate to give bandwidth to real-time flows.
Congestion control component monitors the high priority packet queue and low priority packet queue of the QoS aware scheduler continuously and processes upcoming link congestion. Because queues are not of infinite size, they can fill and overflow. When a queue is full, any additional packets cannot get into the queue and will be dropped. The random early detection (RED) algorithm [70] is implemented to avoid congestion before it becomes a problem. The minimum threshold specifies the number of packets in a queue before the queue considers discarding packets. The probability of discarding packets increases until the queue depth reaches the maximum threshold. After a queue depth exceeds the maximum threshold, all other packets that attempt to enter the queue are discarded. To decrease the number of discarded packets, the mobile node needs to send some Congestion Notification packets (CN) to the sources of these data packets along the reverse route paths. When these sources receive a CN, they may distribute part of traffic to the other node-disjoint routing paths. In this way congestion and bottleneck are able to be avoided or alleviated.

5.4 Simulation Model

To evaluate the effectiveness of DCLQ’s QoS support, the performance of DCLQ is compared with that of MQRD. OPNET 8.1 Modeller was used to create a simulation environment to develop and analyze the proposed distributed cross layer QoS protocol and compare performances with MQRD, which do not take into account about per-hop QoS requirement and QoS support in MAC layer.

Table 1 presents the differences of the two different network architectures. MQRD uses multipath routing, QoS-aware, priority scheduler, and the IEEE 802.11 DCF mode. DCLQ uses multipath routing, cross layer QoS-aware, per-hop QoS guarantee, and Enhanced IEEE 802.11 EDCF mode.
Table 5.1 Architectures used in simulation

<table>
<thead>
<tr>
<th>Architecture</th>
<th>Routing</th>
<th>QoS</th>
<th>Queue</th>
<th>MAC</th>
</tr>
</thead>
<tbody>
<tr>
<td>MQRD</td>
<td>Node-Disjoint multipath Routing</td>
<td>QoS-aware in network layer</td>
<td>Priority Queue</td>
<td>IEEE 802.11 DCF mode</td>
</tr>
<tr>
<td>DCLQ</td>
<td>Node-Disjoint multipath Routing</td>
<td>Cross layer QoS-aware and per-hop QoS guarantee</td>
<td>Priority Queue and per-hop QoS consideration</td>
<td>Enhanced IEEE 802.11 EDCF mode</td>
</tr>
</tbody>
</table>

5.4.1 Performance Metrics

The following metrics are used in varying scenarios to evaluate the two protocols:

- **Packet delivery ratio:**
  
The ratio of the data packets delivered to the destinations to those generated by the CBR sources.

- **Average delay of data packets:**
  
  This includes all possible delays from the moment the packet is generated to the moment it is received by the destination node.

- **Jitter or variability of delay:**
  
  While network latency affects how much time a real-time packet spends in the network, jitter controls the regularity in which real-time packets arrive.
Typical real-time sources generate packets at a constant rate. The destination expects incoming real-time packets to arrive at a constant rate. However, the transmission delay by the hop-by-hop network may be different for each packet. The result is that packets that are sent with equal intervals from a source node arrive with irregular intervals at a destination node.

Jitter is calculated based on the inter-arrival time of successive packets. Frequently, two numbers are given: the average inter-arrival time, and the mean deviation. On a good network, the average inter-arrival time will be the packet transmission interval at the sender, and the mean deviation will be low - pointing at a consistent inter-arrival time.

### 5.4.2 Mobility and Traffic model

The random waypoint model [7] is used to model mobility. Each node starts its journey from a random location to a random destination point with a specific speed. Once the destination is reached, another random destination point is targeted after a pause. A field configuration of 1000m x 1000m field with 50 nodes is used and each node has a 250m transmission radius. The pause time is kept constant at 30 seconds for all the simulation experiments.

All traffic sources with a packet size of 512 bytes are CBR (constant bit rate). The source-destination pairs are spread randomly over the network and the number of sources is varied to change the offered load in the network. In order to investigate the network load ability, two different patterns are used corresponding to 10 and 20 CBR real-time sources. The sending rate of every real-time source is set to 25 pkts/s. The other twenty nodes are randomly chosen to send background BE (Best Effort) traffic with 2pkt/s.

Simulations are run for 800 simulated seconds. The hop counts of flows range from 1 to 8. Each real time flow has a delay requirement of 100ms. If the number of hops from a source to a destination is 5, the per-hop delay requirement of flows
is 20ms. Namely, the average waiting time of real time packet in every intermediate node is not more than 20ms. If the number of hops from a source to a destination is 2, the per-hop delay requirement of flows is 50ms.

### 5.5 Simulation Results

Three key performance metrics: packet delivery ratio, average end-to-end delay and jitter are evaluated under various traffic loads and various mobility rates.

The packet delivery ratio for 10 and 20 real-time flow sources is presented in Figure 5.6. For 10 real-time flow sources, it is observed that MQRD and DCLQ show similar trends. The packet delivery ratio is very high in this case because there is low number of real-time flow sources in the network which in turn reduces the buffering time in queue and probability of congestion and collisions. As node nodes become more mobile, the probability of link failure increases and hence, the packet delivery ratio decreases. As the number of real-time flow sources is increased, the real-time packet delivery ratio of DCLQ is better than MQRD. The reason is that DCLQ uses per-hop QoS-aware scheduler in network layer and differentiated process (Figure 5.4) in MAC layer which improves the contention efficiency of medium access.

Figure 5.7 shows that real-times packets of DCLQ has a lower average end-to-end delay than real-time packets of MQRD because the priority scheduler in DCLQ can schedule real-time packets according to their per-hop delay requirement. This makes some real-time flow packets which have long routing path be forwarded more quickly. When the number of real-time flow sources is increased to 20, the average delay of Best Effort packets of MQRD and DCLQ increases more quickly than that of real-time packets. The reason is that an increase in the number of real-time flow sources leads to higher network load traffic. Because of the limitation of a constrained wireless bandwidth, Best Effort packets that will be sent or forwarded
have to stay in buffers and wait for a longer time to get a radio channel available than real-time packets in order to avoid traffic congestions.

The jitter is a very important metric since it describes the consistency of inter-arrival time of real-time packets. The metric has a quite affection on qualities of voice or video processing in destination nodes. In the simulation the average inter-arrival time in destinations should be the packet transmission interval of senders. The value of the packet transmission interval of senders is 40ms. Figure 5.8 exhibits that the jitter (mean deviation of delay) of real-time packets in DCLQ is smaller than that of real-time packets in MQRD in all scenarios even with low mobility and low network load. DCLQ makes use of per-hop delay requirement of real-time packets to update dynamically allowed waiting time of real-time packets in high priority queue. Then QoS aware priority scheduler chooses a real-time packet with the smallest waiting time from the queue and forwards it to MAC layer. Using the approach, DCLQ can not only help the real-time packets to get their desired delay requirements, but also reduce jitter.
Figure 5.6 Packet Delivery Ratio

(a) 10 real-time sources

(b) 20 real-time sources
Figure 5.7 Average end-to-end delay
10 real-time sources

20 real-time sources

Figure 5.8  Jitter
5.6 Summary

In this chapter, a novel Distributed Cross-Layer QoS (DCLQ) architecture is proposed to provide QoS guarantees for real-time flows in mobile ad hoc networks. Without any extra control overhead in the network layer, DCLQ can schedule packets of real-time flows according to their per-hop QoS requirements. DCLQ implements per-hop QoS-aware priority scheduling and QoS consideration of MAC layer to ensure that real-time flows to achieve their desired service level. The performance evaluation and comparison between DCLQ and MQRD are studied by extensive simulations using OPNET Modeller. Simulation results show that DCLQ achieves better performances than MQRD by providing end-to-end QoS support in MANETs.
Chapter 6 Conclusions

6.1 Conclusions

An ad hoc wireless network is a collection of mobile nodes that communicate with each other by forming a multi-hop radio network and maintaining connectivity management without an existing network infrastructure. Such networks are expected to play increasingly important roles in future civilian and military applications. Design of efficient and reliable routing protocols and QoS provisioning in such network are challenging issues. The goal of this research is to explore efficient multipath routing and QoS provisioning protocols in mobile ad hoc networks.

In chapter 3, a novel Node-Disjoint Multipath Routing (NDMR) Protocol with low control overhead is proposed and implemented to overcome the shortcomings of on-demand existing unipath and multipath routing protocols. NDMR has two novel aspects in that it reduces routing overhead dramatically by recording the shortest number of hops of loop-free paths in the route table entry and achieves multiple node-disjoint routing paths. It is evident from simulation results that NDMR outperforms AODV, DSR and AOMDV because NDMR can decrease routing overhead dramatically and save wireless network bandwidth. NDMR has a higher packet delivery ratio, lower end-to-end delay and routing overhead than AODV, DSR and AOMDV. These characteristics make the protocol suitable for reliable real time data and multimedia communication applications in ad hoc networks.

In chapter 4 a node-disjoint Multipath QoS Routing protocol for supporting DiffServ (MQRD) is proposed to provide QoS provisioning. A solution for a reliable multipath routing and resource management for QoS issues of real-time multimedia applications in ad hoc networks is also presented. The performance evaluation and comparison between NDMR and MQRD are studied by extensive
simulations using OPNET Modeler. Simulation results show that MQRD achieves better performance than NDMR by providing end-to-end QoS support in MANETs. It can be concluded that MQRD has a good potential to serve as a QoS model to provide real-time multimedia applications under the dynamically changing environment of ad hoc networks.

In chapter 5 a novel practical Distributed Cross-Layer QoS (DCLQ) framework is proposed to provide QoS improvement to real-time multimedia traffic. Although MQRD may guarantee that real time flows has better service quality than best effort flows, it can not completely guarantee whether a real time flow can get its desired service requirements such as low end-to-end delay and jitter. The medium access control (MAC) layer of DCLQ can distinguish real-time traffic and best effort traffic by deploying IEEE 802.11e EDCF mode which has support of service differentiation in the shared channel contention. Without any extra control overhead in the network layer, DCLQ can schedule packets of real-time flows according to their per-hop QoS requirements. DCLQ implements per-hop delay QoS-aware priority scheduling and QoS consideration of MAC layer to ensure that real-time flows to achieve their desired service level. Simulation results show that DCLQ achieves better performance than MQRD.

6.2 Future Work

The research work focuses on three important aspects: node-disjoint multipath routing with low routing overhead, QoS routing for support DiffServ and distributed Cross-Layer QoS in mobile ad hoc networks. Other important aspects, which need to be further investigated, are:

- Multicast Routing

Multicast is the process of sending packets from a transmitter to multiple destinations identified by a single address. The packets of each multicast group are forwarded according to a multicast tree. Multicast routing in MANET is also hard since the network topology changes quite frequently.
Therefore, frequent maintenance of the multicast tree will result in a substantial amount of control overhead. How to reduce routing overhead has to be considered when designing multicast routing.

- Distributed Power Control

Most wireless devices are battery-powered and hence it is desirable that protocols for wireless networking should be energy-efficient. A distributed power control scheme should be taken into account to reduce energy consumption of nodes so that the battery life can be extended longer.

- Distributed Security

Due to the broadcast nature of radio communication, wireless networks are susceptible to eavesdropping, malicious jamming and interference, which a well-designed physical layer should be able to avoid. Because usually there are no central control and no trusted authorities in an ad hoc network, how to secure key distribution and manage data encryption and authentication has to be considered when designing a secure mechanism of ad hoc networks.

- Effect of quality of wireless links

Because nodes move in and out of each other’s range, the network topology changes frequently. The network’s dynamic nature, combined with adverse wireless link’s effects, raises issues that are difficult to address. In the physical layer, some techniques are needed to adapt to rapidly changing channel characteristics to make wireless link quality less sensitive to node performance.
References


based routing in the internet. Request for Comments 2386, Internet

service. Request for Comments (Standards, Track) 2212, Internet

Request for Comments (Standards, Track) 2211, Internet Engineering Task

[55] K. K. Ramakrishnan, Sally Floyd, and D. Black. The addition of explicit
congestion notification (ECN) to IP. Request for Comments (Standards Track)

[56] Anna Charny, Jon Bennett, Kent Benson, Jean-Yves Le Boudec, Angela Chiu,
William Courtney, Shahram Davari, Victor Firoiu, Charles Kalmanek, and K.
K. Ramakrishnan. Supplemental information for the new definition of the EF
PHB. Request for Comments (Informational) 3247, Internet Engineering

PHB Group, Request for Comments (Standards Track) 2597, Internet
Engineering Task Force, June 1999


[59] Bruce Davie, Anna Charny, Jon Bennet, Kent Benson, Jean-Yves Le Boudec,
William Courtney, Shahram Davari, Victor Firoiu, and Dimitrios Stiliadis.
An expedited forwarding PHB. Request for Comments (Standards Track)


