Technische Universität Ilmenau

Doctoral Thesis

Quality of Service Aware Routing Protocol for a Self-Organized Communication Network

Submitted by

M.Sc.Aymen Dawood Salman

Born in Baghdad on October 13, 1975

Dissertation submitted in fulfillment of the requirements for the degree of Doktor-Ingenieur (Dr.-Ing.)

in the

Faculty of Electrical Engineering and Information Technology
Institute of Information Technology
Communication Networks Group

First adviser: Prof. Dr. rer. nat. Jochen Seitz

Second adviser: Prof. Dr.-Ing. habil. Kai-Uwe Sattler

Third adviser: Assi. Prof. Dr.-Ing. Mohammed A. Abdala

Date of submission: 19.04.2016
Date of scientific debate: 13.10.2016

URN: urn:nbn:de:gbv:ilm1-2016000494

“Printed with the support of the German Academic Exchange Service”

TECHNISCHE UNIVERSITÄT ILMENAU

DOKTORARBEIT

Ein dienstgütebasiertes Routingprotokoll für ein selbstorganisiertes Kommunikationsnetz

Eingereicht von
M.Sc.AYMEN DAWOOD SALMAN
Geboren in Bagdad am Oktober 13, 1975

Dissertation Zur Erlangung des akademischen Grades Doktoringenieur (Dr.-Ing.)

in dem

FAKULTÄT FÜR ELEKTROTECHNIK UND INFORMATIONSTECHNIK
INSTITUT FÜR INFORMATIONSTECHNIK
KOMMUNIKATIONSNETZE

erster Gutachter: Prof. Dr. rer. nat. JOCHEN SEITZ

zweiter Gutachter: Prof. Dr.-Ing. habil. KAI-UWE SATTLER

dritter Gutachter: Assi. Prof. Dr.-Ing. MOHAMMED A. ABDALA

Tag der Einreichung: 19.04.2016
Tag der wissenschaftlichen Aussprache: 13.10.2016

URN: urn:nbn:de:gbv:ilm1-2016000494

“Gedruckt mit Unterstützung des
Deutschen Akademischen Austauschdienstes”

Erklärung

Ich versichere, dass ich die vorliegende Arbeit ohne unzulässige Hilfe Dritter und ohne Benutzung anderer als der angegebenen Hilfsmittel angefertigt habe. Die aus anderen Quellen direkt oder indirekt übernommenen Daten und Konzepte sind unter Angabe der Quelle gekennzeichnet.

Weitere Personen waren an der inhaltlich-materiellen Erstellung der vorliegenden Arbeit nicht beteiligt. Insbesondere habe ich hierfür nicht die entgeltliche Hilfe von Vermittlungsbzw. Beratungsdiensten (Promotionsberater oder anderer Personen) in Anspruch genommen. Niemand hat von mir unmittelbar oder mittelbar geldwerte Leistungen für Arbeiten erhalten, die im Zusammenhang mit dem Inhalt der vorgelegten Dissertation stehen.

Die Arbeit wurde bisher weder im In- noch im Ausland in gleicher oder ähnlicher Form einer Prüfungsbehörde vorgelegt.

Ich bin darauf hingewiesen worden, dass die Unrichtigkeit der vorstehenden Erklärung als Täuschungsversuch bewertet wird und gemäß § 7 Abs. 10 der Promotionsordnung den Abbruch des Promotionsverfahrens zur Folge hat.

Unterschrift: ______________________

Ort, Datum: ______________________

Acknowledgments

I would like to express my sincere gratitude to my supervisor Prof. Dr. rer. nat. Jochen Seitz for the continuous support of my Ph.D. study and research, his patience, motivation, enthusiasm, and immense knowledge. I would also like to thank my colleagues and friends at Communication Networks group for their help and support.

I would also like to extend my gratitude to all the organization that contributed financially to my PhD thesis. Especially I would to thank the Ministry of Higher Education and Scientific Research/IRAQ (MoHesr), and German Academic Exchange Service (DAAD) for offering me this great opportunity. This work also came as an opportunity to meet new people and make good friends, who helped me learn new things.

I would like to thank my family, my mother, and my father, who have been, from a distance, a great support. Finally, I owe my thanks to my wife "Ruaa" and my children "Yazen and Mohammed", without their encouragement and understanding, it would have been impossible for me to finish this work.

Abstract

Quality of Service Aware Routing Protocol for a Self-Organized Communication Network

Mobile Ad-hoc Networks (MANETs) are characterized by two dimensions namely, anywhere and anytime. The freely moving participating nodes can form an ad hoc network anywhere, and the mobile nodes can join or leave the network anytime. A particular mobile node in a MANET can communicate with all the other nodes using the multihop communication. Thus, MANETs offer a vast range of applications in various domains like entertainment, military, emergency, etc. However, the implementation of real-time applications like voice/video calling that demands stringent quality requirements over MANETs is a major challenge. This challenge arises due to the unplanned and dynamic nature of MANETs, due to the unreliability of wireless links, due to the scarcity of resources like battery, bandwidth, processing power, due to the large-scale nature of MANETs, etc. This issue can be addressed at the network layer or the routing protocol, which establishes multiple routes from source to destination and adapts to the dynamicity of MANETs without compromising on the quality requirements. The primary goal of this work is the investigation and development of such a routing algorithm that supports real-time applications over MANETs. For adaptive multipath routing, we studied Ant Colony Optimization (ACO) algorithms originat from the fields of Swarm Intelligence (SI) while Quality of Service (QoS) computation is carried out by cleverly utilizing the monitoring feature of the Simple Network Management Protocol (SNMP). So, combining these two mechanisms we propose a powerful adaptive multipath **Qo**S-aware **R**outing protocol based on **A**CO (**QoRA**). We discuss and investigate the internal working of QoRA and perform detailed simulation studies in the network simulator ns-3. Finally, we discuss the implementation of QoRA routing algorithms in a real world testbed.

Zusammenfassung

Ein dienstgütebasiertes Routingprotokoll für ein selbstorganisiertes Kommunikationsnetz

Mobile Ad-hoc-Netze (MANETs) ermöglichen eine Kommunikation überall zu jedem Zeitpunkt. Frei sich bewegende Knoten können überall ein solches Netz bilden, wobei die Teilnehmer zu jeder Zeit dem Netz beitreten oder es wieder verlassen können. Ein teilnehmender Knoten in einem MANET kommuniziert mit allen anderen über Multi-Hop-Kommunikation. So ermöglicht ein MANET viele unterschiedliche Anwendungen aus verschiedenen Domänen wie beispielsweise Unterhaltungskommunikation, Notfallkommunikation oder Einsatzkommunikation. Allerdings benötigen Echtzeitanwendungen wie Telefonie oder Videokommunikation eine stringente Kommunikationsdienstgüte, was für MANETs eine große Herausforderung darstellt. Diese Herausforderung hat viele Gründe: das dynamische und unvorhersehbare Verhalten der Knoten im MANET, die Unzuverlässigkeit der drahtlosen Kommunikation, die Beschränkung der zur Verfügung stehenden Kommunikationsressourcen (wie Batterielaufzeit, Bandbreite oder Prozessorleistung), die relativ große Abdeckung durch ein MANET. Die Herausforderung kann in der Vermittlungsschicht durch ein spezielles Routingprotokoll gelöst werden, das mehrere gleichzeitige Pfade von der Quelle zum Ziel verwendet, sodass die Dynamik in einem MANET Berücksichtigung findet ohne dass die Dienstgüte kompromittiert werden muss. Das vorrangige Ziel dieser Arbeit ist die Erforschung und Entwicklung eines solchen Routingverfahrens, das Echtzeitanwendungen in einem MANET unterstützt. Für das adaptive Mehrwegerouting wurde ein Ameisenalgorithmus (Ant Colony Optimization, ACO) angewendet, der das Prinzip der Schwarmintelligenz ausnutzt. Die Bestimmung der aktuell möglichen Kommunikationsdienstgüte erfolgt über die Informationen, die das Netzmanagementprotokoll Simple Network Management Protocol SNMP standardmäßig zur Verfügung stellt. Durch die Kombination dieser beiden Ansätze wurde das adaptive Mehrwegeroutingprotokoll "**Qo**S-aware **R**outing Protocol based on **A**CO" (**QoRA**) vorgeschlagen. In der vorliegenden Dissertation werden das Konzept von QoRA vorgestellt und die interne Funktionsweise erläutert. Anhand umfangreicher Simulationen auf Basis des Simulationswerkzeug ns-3 werden die Vorteile des Verfahrens nachgewiesen. Den Abschluss bildet die Diskussion einer Implementierung von QoRA in einer realen Testumgebung.

Contents

Chapter 1

Introduction

In this thesis, we propose an adaptive QoS-aware multipath routing protocol for Mobile Ad-hoc Networks (MANETs) that utilizes the evolutionary computation techniques from the fields of Swarm Intelligence (SI) and the functionalities of Simple Network Management Protocol (SNMP), the prevalent monitoring, and management tool for the Internet. In our work, the monitoring ability of SNMP is cleverly used to determine the QoS parameters without inducing any additional control messages. At the same time, from the network point of view, the SI methodology utilizes this information about QoS parameters and determines the multiple QoS-aware routes satisfying the multimedia application criteria. Also, from the intelligence point of view, the proposed routing protocol dynamically adapts to the varying environmental conditions without hindering the quality of the real-time applications implemented over the MANETs.

1.1 Motivation

Mobile Ad hoc Networks (MANETs) [1–3] are the networks in which the freely moving participating nodes can communicate with each other at any given instance without worrying about maintaining a direct connection to any particular fixed access point. In MANETs, the nodes concede a network by cooperating mutually in a distributed manner and any node becomes a part of the network by locally contacting any other member of the network and can communicate with all the other nodes using multihop. So, MANETs are self-organized networks and follow the distributed algorithms. Also, MANETs are highly scalable in terms of the number of participating nodes, as the nodes only require the local awareness of the network.

MANETs are becoming highly popular with the increase in the use of mobile communicating devices, such as smart phones, personal digital assistants (PDAs), and laptops.

To determine an exhaustive list of all the possible application scenarios in MANETs is out of the scope of this thesis. In some scenarios like network extension, where an infrastructure network is extended to the areas where raising an infrastructure is not feasible either for monetary reasons or for geographical conditions. In other application domains, MANETs compete with the other possible technologies.

However, we would like to focus on the application domains which requires audio/video streaming over MANETs e.g. in the stressful rescue operations performed during the fire, floods, earthquakes, war, etc. The audio and video services between the rescuers will provide the rescue leaders with a better overview of the situation and will empower him/her to take the appropriate steps. In the rescue scenarios, the timely receptions of these safety messages are required. But in reality, the implementation of these real-time multimedia applications over MANETs is a big challenge. These applications demands very stringent and inflexible QoS (as compared to the streaming of stored content) in terms of parameters like loss, delay, jitter, and bandwidth, which are difficult to met on MANETs due to its several constraints like dynamic topologies caused by nodes mobility, limited resource availability, and energy constraints, poor quality of the wireless link due to fading and interference, and shared radio channel, etc. So, the estimation of the QoS parameters on the links in the presence of the above-mentioned issues becomes the major hindrance in providing QoS.

The other issue faced in such scenarios is related to the node density. A high node density situation leads to the packet collision causing high end-to-end delays while low node density leads to the decreased signal strength causing frequent link failures and high communication disruptions. To overcome this issue, we have used adaptive multirate mechanism that considers the link quality between the nodes and adapts the transmission rate accordingly.

As discussed earlier, to ensure the fast, reliable and in time delivery of the real-time multimedia data, an efficient routing protocol with an optimal QoS mechanism is required. The purpose of this QoS routing protocols is to identify the routes from source to destination that satisfies the applications requirements and avoid congestion.

Population-based global search meta-heuristic approaches are promising approach for solving the problem of QoS routing. They efficiently explore the search place and provide excellent solutions for optimization problems. Ant Colony Optimization (ACO) algorithms are one such approach originated from the fields of SI or bio-inspired computing. ACO algorithms are distributed and self-organized in nature.

ACO routing protocols are derived from the foraging behavior of ants. Ants while hunting for the food deposit a particular amount of volatile substance called as pheromone on the path. The other ants after detecting a pheromone follow the trail and reinforce it. The pheromone acts as an indicator towards the food source and an indirect medium for communication between the ants. In terms of routing, the pheromone concentration corresponds to the available resources on a particular path, and this information is used to enable QoS awareness. But, in order to determine these QoS values most QoS-aware routing protocols requires an additional message exchange between neighbor nodes and also does not consider the congestion problem during data forwarding. These shortcomings increase the routing overhead, waste time and energy during path discovery, and increase packet loss ratio.

In our approach, we utilize the existing monitoring features of the SNMP to obtain required values locally. Based on these local values and the features of self-organization, we can estimate the global pattern of the relevant QoS parameters and also utilize the monitoring features of SNMP to avoid congestion problems during data packet forwarding.

In this work, we performed all the simulations in network simulator ns-3. The main reason behind the selection of ns-3 is its close resemblance to the real-world networking architecture. The nodes, interfaces, and objects in ns-3 are based on the Linux networking architecture. Since in this work we are dealing with the deployment of real-time multimedia applications over MANETs, ns-3 is an optimal choice considering its real-world features.

1.2 Thesis Objectives and Contributions

The main objective of our **Qo**S-aware **R**outing protocol based on **A**CO (QoRA) framework is to handle the dynamics of MANETs and real-time multimedia traffic simultaneously, without conceding additional control overheads and compromising on the received audio/video quality, thus maximizing the network resource utilization.

To achieve the above-mentioned objective, we have designed and implemented a version of SNMP Manager especially for MANETs that is integrated into the QoRA entity. In addition to the monitoring tasks, it determines the local QoS values. In our approach, we named it as QoS Manager and it is a software module. The necessary information required for QoS calculations is provided by the SNMP agent, which is also a software module, and runs on every networked device. This mechanism computes the values for QoS parameters like Bandwidth, Delay, Packet loss, and QoS Thresholds for all the links on the path.

Next, we designed and implemented a self-organizing routing protocol that uses the locally available QoS information using distributed control mechanisms to develop the global pattern of the multiple QoS parameters and thus establishing a QoS-aware route. The routing protocol is based on the meta-heuristic characteristics of the ACO. This is the novel routing protocol design that combines the monitoring capabilities of SNMP and intelligence of ACO.

1.3 Organization of the Thesis

The thesis is organized as follows:

Chapter 2: Mobile Ad hoc Networks
In this chapter, we give the general introduction on MANETs and the characteristics of self-organization. Later we describe the IEEE 802.11 standard, the concept of multi-rate adaptation techniques followed by the classification of routing protocols in MANETs.

Chapter 3: Quality of Service Routing
In this chapter, we defined the QoS-metrics and the challenges in providing QoS routing in MANETs followed by the survey of the QoS-aware routing protocols in MANETs. Furthermore, the chapter covers the topic of ACO and survey of the QoS-aware ACO routing protocol in MANETs.

Chapter 4: QoRA Approach
This chapter introduces our proposed adaptive QoS-aware multipath routing protocol. Later, the protocol is discussed in-depth and detailed information is provided about the physical and logical components, the architecture and different elements of the QoRA.

Chapter 5: QoRA Implementation
This chapter gives the implementation details of every single component used in the architecture from the simulation point of view. Also, the overview of ns-3 is provided in this chapter.

Chapter 6: Results and Discussion
In this chapter, the results obtained after the implementation of QoRA over different scenarios are discussed. Also, the performance of QoRA routing mechanism is compared with the other existing routing protocol for MANETs.

Chapter 7: Conclusion and Future work
This chapter concludes the thesis and provides guidelines for future research.

Chapter 2

Mobile Ad hoc Networks

In this chapter, we introduce the Mobile Ad hoc Networks (MANETs) and describe the characteristics of Self-Organized Networks (SON) to overcome the shortcomings in MANETs. In the beginning, we describe the MANETs applications followed by the IEEE 802.11 standard and its multi-rate characteristics. Later we present the classification of routing protocols in MANETs. Finally, we conclude the chapter with the summary.

2.1 Wireless Networks

The wireless communication networks have been experiencing exponential growth as compared to the traditional wired networks. Also, due to the rapid advancement in wireless technologies, the wireless devices are becoming smaller and powerful, simultaneously. This technological advancement has lead to the exploitation of the use of mobile electronic devices such as smart phones, Personal Digital Assistants (PDAs), laptops, etc. which has paved the way for humongous applications and wireless network services. The wireless networks can be classified into the following two categories. Infrastructure based network require an architectural framework to utilize the network services. Cellular networks, Wireless Local Area Network (WLAN), Wi-Fi, etc. are the examples of infrastructure based network. The other category of wireless networks is ad-hoc networks. As the name suggests these networks are on demand networks and does not require any infrastructure for network service utilization. Mobile Ad-hoc Networks (MANETs), Vehicular Ad-hoc Networks (VANETs), and Wireless Personal Area Networks (WPANs) are the example of such networks.

Infrastructure-based networks are the networks that are planned well ahead of time i.e. preconfigured and demands the proper placements of the essential physical network elements through which the services are delivered. A very good example of such kind of networks is Cellular networks. It fully utilizes the Public Switched Telephone Network (PSTN) backbone network that consists of network switches, base stations, and mobile hosts. Similarly, WLANs are also realized on the fixed wireless access points that connect the mobile stations to an existing network backbone.

Ad-hoc networks, on the other hand, do not require any infrastructure for its existence. Here, short-range radio devices like IEEE 802.11 (Wi-Fi) cooperate with other Wi-Fi devices in their communication range to form a dynamic temporary network. The other short-range radio devices include Bluetooth, HiperLan, and infrared transmission.

Any of these short-range communication technologies can be used for ad-hoc networking. However, IEEE 802.11 is more advantageous than Bluetooth from an ad-hoc networking point of view. The Bluetooth has shorter communication range and smaller bandwidths as compared to the IEEE 802.11. Also, HiperLan and infrared works at very short transmission range and low mobility. Therefore, we decided to choose IEEE 802.11 technology for the implementation of ad-hoc networks in the network simulator. In section 2.2 we describe this standard in more details.

Figure 2.1 shows an ad-hoc network that is formed using different devices like tablets, notebooks, smart phones, smart vehicles, etc. These devices can communicate directly with the other devices that are in their communication range using single hop communication. However, to communicate with the device that lies outside this range, the sending device can relay the message via different intermediate devices, thus facilitating multihop communication. The networks exhibiting these characteristics are also called as multihop wireless ad-hoc networks. MANETs have gained a lot of popularity in the current decade where the emphasis is on to provide real-time multimedia applications and thus are expected to become an important part of the future wireless communication networks. As, discussed earlier, relaying techniques can be implemented using MANETs to extend the network in the areas where networking services are not available due to the lack of infrastructure. MANETs are distributed systems, i.e. it can be deployed anywhere, anytime without any fixed infrastructure or the central management and on the basis of mutual cooperation between the devices the applications are executed over the network. So, in MANETs, the network topology is determined in a self-organized manner using the local mutual cooperation between the devices. Localization property enhances the scalability and dynamic adaptation due to the varying environmental conditions in MANETs.

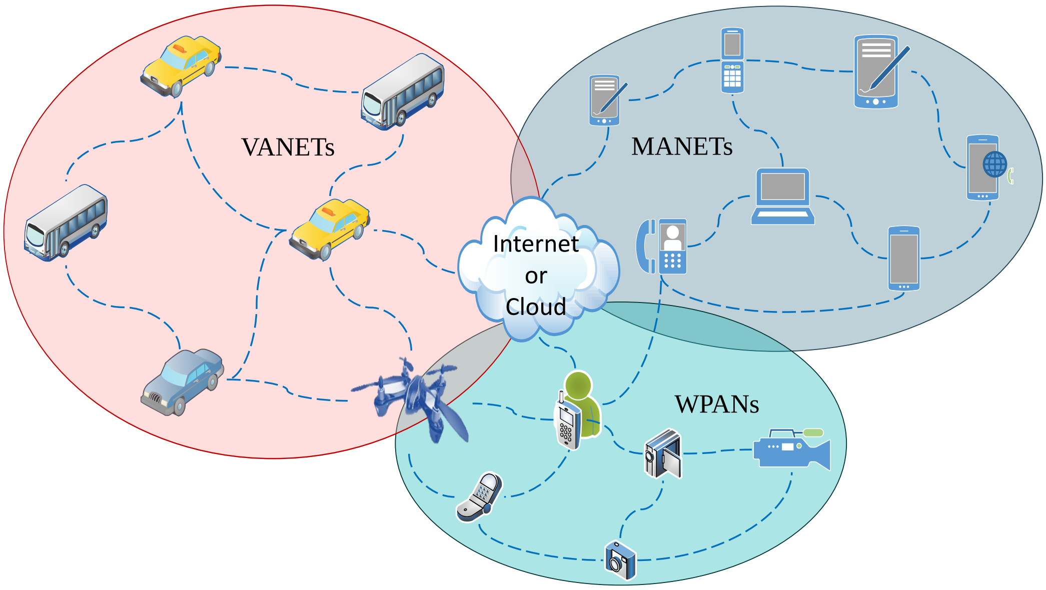


Figure 2.1: Examples of Mobile Ad hoc Networks

2.1.1 Characteristics of Self-Organize Networks

Self-Organization is a process in which the numerous low-level components interact mutually to determine the global high-level pattern of the system. These mutual interactions are carried out locally without taking any reference from the global pattern. Thus, in self-organized systems, the application with simple behavior at the lower level determines the sophisticated overall system pattern. This phenomenon is called emergent behavior. Why and how emergent behavior occurs is still not well understood. But, these behavior exhibited by the Self-Organized Networks (SON) can be used to overcome the shortcomings caused by mobility and to provide the features of adaptability, robustness, and scalability in MANETs. Next, we will explain the capabilities of Self-organizing networks [4–6]:

- **Self-management** is the capability to maintain the network based on the current system parameters. This function is required to keep the network operational. It is very closely related to the self-optimization.
- **Self-optimization** functions aim to extract the optimal performance and quality from the current state of the network. These functions are simply the self-optimization routine that administers, analyzes and triggers the optimization action on the affected network elements when necessary. This capability enhances the network performance by making the network adaptable to the varying network parameters.

- **Self-configuration** is a method to generate necessary configurations that depend on the current environmental conditions and the application requirements e.g. connectivity, quality of service parameters, power control, etc. This feature aims to enable the device essentially with Plug and Play capability thus, reducing the human intervention in the configuration process. This capability is executed using SON software and it provides fast network deployment and reduced human intervention leading to the reduction in network operator costs and a system that is less prone to human errors.

- **Self-healing** known also as Self-stabilizing, the notion of self-organization also appears in the context of failure resilience and network restoration. This mechanism detects different failures according to their cause e.g. link and node failure, network overload or congestion, etc.; locates this failure in the network and provide automatic repair mechanism. This capability is performed by the collection of SON procedures that detects and solves problems thereby avoiding its impact on user experience.

- **Adaptability** refers to the flexibility exhibited by the network in response to the changes observed in the environmental conditions. All the entities proactively adapt to the changes in an organized manner, so that the system as a whole can adapt or reorganize itself in response to these changes.

- **Scalability** reflects the fact that the system works efficiently in the presence of a large number of communicating entities. This characteristic ensures the ease with which the networking tasks such as addressing, routing, location managements, configuration managements, interoperability, security, etc. can be carried out in the presence of a large number of entities.

All these features and characteristics make MANETs a very important choice for a large number of applications domains as discussed in next section.

2.1.2 Applications of MANETs

Mobile ad-hoc networks, that is defined in the previous section as the wireless network without infrastructure, offers a large range of potential applications. The entire list of possible applications fields can only be limited by one's imagination. In certain situations as described below, MANETs are the only possible solutions. In other application fields, MANETs compete with the other possible technical solutions or must be part of a combination of technologies. In this section, we discuss the obvious and popular MANTEs applications [7, 8].

- **Network Extension:** As the name suggests, this application is used to extend the infrastructure network to the areas where infrastructure is not present. The ad-hoc networks are used to provide the networking connectivity between the access points and the nodes that are outside its range of communication. Typical scenarios, where network extension is the only means to communicate, include disaster scenarios where infrastructure has collapsed or archeological excavation in remote areas where no network connectivity is available.

- **Personal Area Networks (PAN):** Nowadays users are associated with more than one communication device while on the go e.g. smart phones, laptops, tablets, PDAs, etc. The ad-hoc networking between all these portable devices, within a personal operating space, typically up to a range of 10-meters is considered as Wireless PANs (WPANs). Even if the user is in motion, the relative position of all the devices to the user remains the same. Most of the times, one or more of these devices are connected to the internet. Bluetooth and infrared are two key WPAN technologies. WPAN are characterized by low complexity and low power consumption. WPANs also provide interoperability with 802.11 networks that have made WPANs very popular in current times. Typical applications that can be considered over WPANs include infotainment and assistance for old people, e.g. automatic heart rate monitoring or health monitoring and sending the information to the physician [5, 6].

- **Internet of Things (IoT):** IoT refers to the collective use of computers available in the physical environment of users, perhaps in a form invisible to users. It is an idea where communication is made to appear anywhere, anytime and between any entities. So, the communication design is totally based on the physical location of the devices or the proximity of the communicating nodes. Ad-hoc networks are very sensible and logical approach for such complex communication environment. The importance of ad-hoc networking can be explained via the following example. In IoT, a smartphone controlling a TV set-top box need to traverse the control information via dozens of routers through an internet connection even when the two devices just lie several feet away from each other, while in ad-hoc networking the two devices can simply communicate with each other avoiding the unnecessary routing complexities.

- **Vehicular Networking:** In these applications one communicating entity is a vehicle. With this definition, the wide range of technologies involving vehicular communication can be summed up. The first type of vehicular networking involves the short-range communication between the vehicle and the devices that are part of PANs. Such communication system relies on Bluetooth, infrared and

Wi-Fi technologies. Also, many devices in the local network of the vehicle can also maintain long-range wireless connections, e.g. via 3G cellular telephony, Worldwide Interoperability for Microwave Access (WiMAX) or Long Term Evolution (LTE). The second class of vehicular networking is Vehicle-to-Vehicle (V2V) communication. The vehicles have short interaction with their peers due to their high relative speed, which gives rise to highly mobile ad-hoc networks. The typical usages of this technology are in traffic management, accident avoidance, lane changes with pre-negotiated lane clearance, etc. The final class of vehicular networking is Vehicle-to-Infrastructure (V2I) communication. The vehicles on the move communicate with the fixed access point in the infrastructure. This brief communication constitutes to highly mobile ad-hoc network. The most popular applications of V2I are currently toll collection systems. V2V and V2I together are also called as Vehicular Ad-hoc Networks (VANETs).

Use cases of VANETs are divided into following three main groups [9].

1. Safety applications: These application aims to decrease the road accidents and save the lives of thousands of people. The shared information between V2V and V2I can be used to predict the collision.
2. Traffic monitoring and management applications: This application provides traffic assistance so that drivers can have better speed management and efficient navigation.
3. Infotainment applications: The last VANET application offers entertainment and useful information for the drivers such as parking and fuel stations, cinema, shopping centers, etc.

2.2 Standard IEEE 802.11

IEEE 802.11 standard [10] represents the Physical (PHY) Layer and Media Access Control (MAC) Layer specifications for implementing the WLAN communication. This standard has many advantages like availability, low cost, and most importantly its ability to create ad-hoc networks without previous planning. Therefore, the MAC protocol defined by 802.11 standard is the basic and most commonly used MAC protocol for ad-hoc wireless networks.

IEEE 802.11 defines two access methods using Distributed Coordination Function (DCF) and one access method using Point Coordination Function (PCF). The DCF adopts carrier sense multiple access with the collision avoidance (CSMA/CA) mechanism and utilizes the randomized back-off scheme in order to avoid the collision. This approach is the default and compulsory MAC method in the IEEE 802.11 standard. There are two methods used for packet transmitting; the default method uses DATA/ACK (Acknowledgment) scheme i.e. ACK is transmitted on the reception of correct data packet while the lack of ACK indicates the transmission failure. The other optional method is based on Request-To-Send (RTS) and Clear-To-Send (CTS) [11] extension with DCF in order to avoid hidden node [12] and exposed node [13] problems in WLAN. RTS and CTS are small signaling broadcast packets that contain all the information regarding the size of the upcoming data frame and the amount of time for which the channel will be occupied. The third optional access method depends on PCF where the Point Coordinator or Access Point (AP) polls the nodes for medium access.

The access methods using DCF provides asynchronous data service and can be implemented in both ad-hoc mode and infrastructure mode while access method using PCF provides time bounded data service and can be implemented only in the presence of AP. However, the basic idea behind these protocols is to control the medium by managing and reducing contention in the wireless communication medium, both in the infrastructure mode and in ad-hoc mode [14, 15].

In IEEE 802.11 MAC, the priorities are set by the different Inter-Frame Spaces (IFSs). IEEE 802.11 MAC defines three IFSs namely Short Inter-Frame Spacing (SIFS) with the highest priority, PCF IFS or PIFS with the medium priority and DCF IFS or DIFS with the lowest priority. SIFS are used for sending ACK, CTS and polling responses while DIFS are used for asynchronous data service. PIFS are used only in the PCF polling mode for synchronous data service.

The node ready to send data starts sensing the medium using Clear Channel Assessment (CCA). If the medium is free for DIFS, the node starts sending the data as shown in figure 2.2. The receiving node waits for SIFS and sends ACK. However, if the medium is busy, the sending node has to wait for the random back-off time in addition to the free DIFS. During this random back-off time, if another node occupies the channel, then the back-off timer stops as a fairness measure. If another collision happens after the random back-off time has elapsed and during the waiting for DIFS, the sending node has to increase its random back-off time exponentially. In the case of errors, the data packets are automatically retransmitted.

Figure 2.3 shows the DCF CSMA/CA Method with RTS/CTS extension. All nodes must wait for DIFS before sending RTS packet, which contains the reservation parameters as defined in Network Allocation Vector (NAV). If the receiving node is ready to receive, it will reply with the CTS packet with NAV after SIFS interval. The NAV also determines the amount of time the channel is needed by the data packet. The sending node then uses the channel for sending the data after SIFS and receiving node send ACK on correct

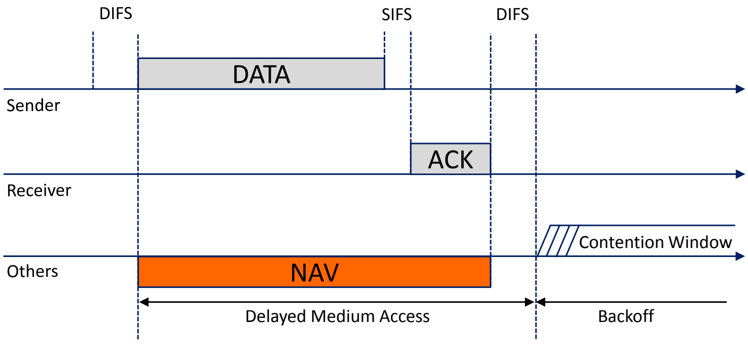


FIGURE 2.2: Frame and delay sequence diagram for CSMA/CA [15]

reception of data after SIFS. However, if the receiver is not ready to receive, it will not reply back to the CTS message, and this indicates that the medium is busy. The sending node on sensing that the medium is busy has to choose a random back-off time from the given range and has to wait for this back-off time to elapse to zero. During this back off time if the sending node discovers that the channel is busy again, then as a fairness measure, it stops its timer until the channel is idle again. Once the countdown reaches to zero, the sending has to wait for DIFS again before sending RTS packet. In case it senses the channel is still busy, it has to choose another exponentially increased back-off time, and the process then continues. In this approach, the collision may occur only during the transmission of RTS. However, this collision is very small as compared to the data transmission.

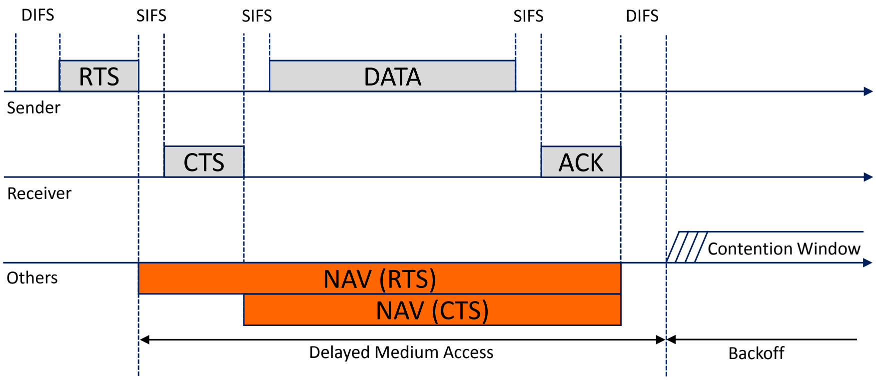


FIGURE 2.3: Frame and delay sequence diagram for CSMA/CA with RTS/CTS [15]

The IEEE 802.11 physical layer supports different coding and modulation schemes that enable it with the multi-rate transmission capability. Table 2.1 shows the different IEEE 802.11 standards namely 802.11a [16], 802.11b [17], 802.11g [18], and 802.11p [19] along with their operating frequency band, supported multi-rate stream data rates and modulation schemes.

Table 2.1: Types of IEEE 802.11 standard

Type	**Frequency (GHz)**	**Stream data rate (Mbit/s)**	**Modulation**
IEEE 802.11a	5	6, 9, 12, 18, 24, 36, 48, 54	OFDM
IEEE 802.11b	2.4	1, 2, 5.5, 11	DSSS, OFDM
IEEE 802.11g	2.4	6, 9, 12, 18, 24, 36, 48, 54	OFDM
IEEE 802.11p	5.9	3, 4.5, 6, 9, 12, 18, 24, 27	OFDM

- Orthogonal Frequency-Division Multiplexing (OFDM)
- Direct Sequence Spread Spectrum (DSSS)

The quality of the physical parameters of the wireless channel is observed to be inversely proportional to the distance between the two communicating nodes. So, the trade-off is observed between the transmission range and the transmission rate. This trade-off has lead to the inclusion of multi-rate capability in the IEEE 802.11 standard. In multi-rate data transmissions, the data rate is varied according to the bit error rate observed on the channel so that the bit error rate can be kept under the threshold in order to facilitate the reliable communication on the available channel.

2.3 Multi-rate Adaptation

The consumer expectations from the wireless devices always demand both, higher transmission rate and higher transmission range [20]. The transmissions at high data rate lead to shorter channel consumption or small packet duration and thus cause less interference on the channel. This means an increase in the transmission rate provides higher efficiency. However, if the distance between the two communicating nodes increases then the communication is highly prone to errors due to the decreasing signal to noise ratio. These errors trigger the retransmission over the already resource deprived links causing a steep decrease in the application throughput. On the other hand, the lower transmission data rates are robust against the errors but do not utilize the available channel resources effectively. This leads to the poor performance of the applications that require higher bandwidths [21]. This trade-off can be addressed by the Multi-rate Adaptation mechanism supported by IEEE 802.11 standard.

Multi-rate adaptation is the mechanism that determines the dynamic switching of the data rates according to the unstable channel conditions caused by mobility and interference. The goal of multi-rate is to select the data rate such that the optimal throughput can be achieved for the given channel conditions with the acceptable bit error rate. However, as stated earlier the IEEE 802.11 standards supports the multi-rate, but the mechanism to select the optimal rate in the multi-rate capable networks is not specified in the standards. The users can manually choose the desired data rate or the device can automatically choose the data rate depending on the channel conditions. Several auto rates protocols have been proposed in the literature to use the multi-rate capability offered by the physical layer [22–30]. The basic idea of multi-rate adaptation can be defined as the two-step process. The first step is to estimate the channel quality and the next step is to determine the appropriate rate selection mechanism. Figure 2.4, shown the 802.11b PHY transmission rate (1 $\sim$ 11 Mbps).

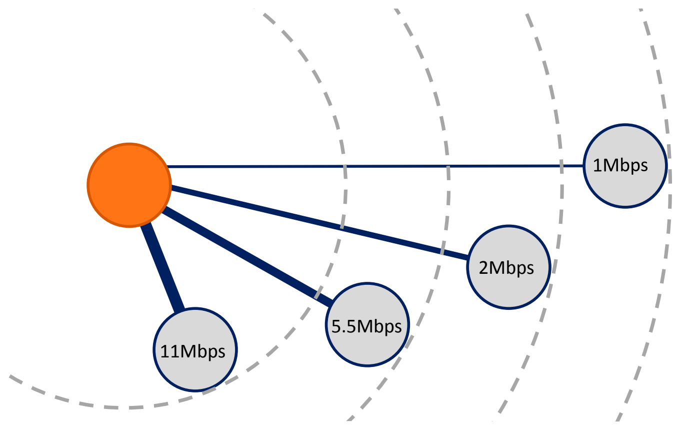


FIGURE 2.4: The PHY transmission rate of 802.11b

Channel quality estimation defines the measurement of the time-varying states of the wireless medium. On the basis of these measurements, the quality of the channel in the future is predicted and then the transmission rate is adjusted accordingly. The following methods are proposed for estimation of channel quality that collects information from the different layers of the protocol stack.

The first method uses the physical-layer approach. This approach measures the physical parameters like signal-to-noise ratio (SNR), receive signal strength (RSS), bit error rate (BER), etc. in order to estimate the channel quality. In this method, the signaling frames utilize RTS/CTS handshake. In the next step, the measured SNR is mapped to this information and on the basis of this mapping the optimal transmission rate is determined, Receiver-Based Auto Rate (RBAR) [23] and Opportunistic Auto Rate (OAR) [24] are the examples of physical-layer approach. However, this approach increases the implementation complexity, as it demands certain modifications in 802.11 standard. In [25] proposed another dynamic rate adaptation mechanism. In this method, the channel and receiver conditions are determined by the RSS of received frames and the number of frame retransmissions. As compared to the previous mechanisms, this algorithm does not require any advancement in the MAC procedures. Using such physical-layer metrics, in theory, determines nearly perfect rate estimation as it reacts quickly to the fluctuating wireless channel.

The second method is the Link-layer approach, the results of frame transmission indirectly provide the information about the channel quality. Automatic Rate Fallback (ARF) [22], Adaptive Auto Rate Fallback (AARF) [26], Collision-Aware Rate Adaptation (CARA) [28], Robust Rate Adaptation Algorithm (RRAA) [29], Adaptive Auto Rate Fallback with Collision Detection (AARF-CD) [30] are the examples of Link-layer approach.

In this approach, the channel quality information is generated using the data or the signaling frames. The data frame approach (ARF and AARF) uses only the data frames to determine the channel quality. The successful consecutive frames transmission determines the good channel quality while the failure observed in consecutive frame transmission determines the bad channel quality. This good/bad channel quality decides the optimal rate estimation. However, this approach is not suitable for multi-user scenarios as the transmission rate is decreased sharply when the frames failure occurs due to frames collision. The problem of multi-client contention in a single collision domain is overcome by making an optional use of RTS/CTS exchange. This method helps to differentiate between the frame failures due to the frame collisions and frame failures due to the channel error (CARA, RRAA, AARF-CD). So, after observing the frame loss the RTS is turned ON to determine the decrease in the transmission rate and turned OFF upon successful frame transmission. This mechanism has two major advantages. Firstly, it does not require any modification in the 802.11 standard and secondly the overhead observed due to optional RTS/CTS is very small.

Rate selection mechanism defines the optimal rate on the basis of the channel predictions made using the methods mentioned above. The most common rate selection technique is *threshold* selection. Here, the indicator value is compared with the list of threshold values that indicates the boundaries between the variable rates. The threshold mechanism can be implemented using two techniques namely sequential rate adjustment and best rate adjustment. In the sequential rate adjustment, the current rate can be increased or decreased only by one level corresponding to the good or bad channel condition. In the best rate adjustment, the current rate can be jumped over the multiple levels yielding the optimal performance.

2.4 Routing Protocol for MANETs

Routing protocols serve the two purposes in any communication scenario namely, *route discovery* and *route maintenance* so that the data can be transferred smoothly over the network. The direct adaptation of the routing protocols developed for static environment (wired networks) to the dynamic environment exhibited by MANETs is not advisable, as it has significantly high overheads causing poor performance, slow route convergence, low throughput with possibility of route loops during node failure and network partition and network congestion problems [1, 31].

To overcome these problems numbers of optimization solutions have been proposed to adapt traditional proactive internet-based protocols to MANETs. Also, new algorithms are proposed such as reactive or on-demand routing, hybrid routing, location-based routing, etc. Figure 2.5 depict the classification of the MANETs routing protocol [8, 32, 33].

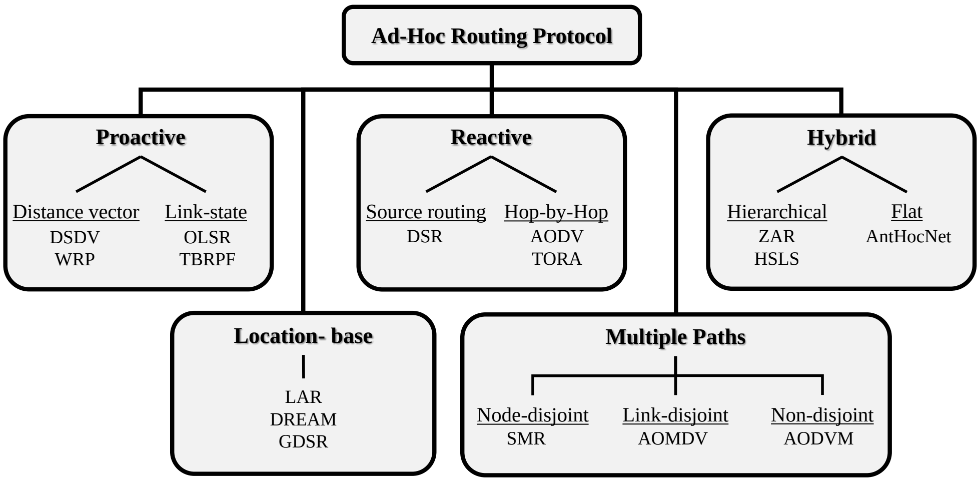


FIGURE 2.5: The classification of the MANETs routing protocol

2.4.1 Proactive Routing Protocols

In proactive protocols, every node has the unicast route to all the other nodes in the network irrespective of the fact that whether these routes will be used in future or not. So, the routes are readily available to all destinations and the source can start sending data without waiting for the route discovery. In order to maintain these routes, the proactive protocols periodically exchange control messages and maintains this information in the routing table. Since the routing information is represented in terms of tables, these protocols are also called as table-driven protocols.

Proactive protocols are further classified into *distance vector* and *link-state* algorithms [32]. The routing protocols for wired networks are developed on these two algorithms. In distance vector protocols, every node exchanges a vector with its neighbors. This vector contains the information about the node's current distance to all the known destinations. This mechanism helps to spread out the distance information about the different destinations in the network and enables a node to calculate the routes in a distributed manner. Destination-Sequenced Distance-Vector (DSDV) [34] and Wireless Routing Protocol (WRP) [35] are the examples of distance vector protocols.

In link-state protocols, every node floods the network with the link-state update messages that carries the information about the current condition of all the outgoing links of that node. However, the flooding of this update by all the nodes leads to high network overheads that can overwhelm the network. In order to overcome this problem, the concept of Multipoint Relays (MPRs) is adopted as an optimization parameter. Some nodes are selected as MPRs by the other nodes in the network and only the MPRs are allowed to generate and send these updates. Optimized Link State Routing (OLSR) [36] and Topology Broadcast based on Reverse-Path Forwarding (TBRPF) [37] are the examples of link state protocols.

Though proactive protocols do not cause delays during route discovery, their performance suffers mainly due to the frequent topology changes caused by the mobility of nodes that leads to slow convergence to optimal paths. In addition, the topology changes incur high route maintenance cost and large overhead in terms of routing table's maintenance in the ad-hoc network. Also, if there is low activity in the network, the high information exchange mechanism of proactive protocols is not desirable as it will be simply waste of precious network resource [38]. So, proactive protocols are not suitable for MANETs.

2.4.2 Reactive Routing Protocols

In contrast to proactive protocols discussed in the earlier section, reactive protocols discover and maintain routes between particular source-destination pair only when it is necessary i.e. on demand. This approach has following advantages over the proactive routing, low overheads as global routing table maintenance is not required and fast response to the topology change in the network. Due to this, reactive protocols have the upper hand in MANETs as compared to the proactive routing protocols. However, as routes are discovered on demand, latency is observed during route discovery phase and also high flooding during this phase may overwhelm the network [39].

Reactive protocols are further classified into two categories: *source routing* and *hop-by-hop routing* [33]. In source routing, all the data packets have the complete path information from the source to the destination node. The intermediate nodes read the information in the header of the packet and forward it accordingly. Thus, the intermediate nodes are not required to maintain the up-to-date routing information and the neighbor connectivity via periodic Hello messages. However, this protocol does not scale well in larger networks i.e. with a high number of intermediate nodes as explained below. Firstly, the probability of route failure increases with the increase in the number of intermediate nodes. Secondly, the increase in the number of intermediate nodes increases the amount of overhead in each data packet. Dynamic Source Routing (DSR) [40, 41] is an example of source routing protocol.

In hop-by-hop routing, each data packet carries only two addresses, namely the destination address and the next hop address. Here, each node maintains the neighbor node connectivity through Hello messages and on the basis of this information it updates the routing table. The intermediate node forwards the data packet on the basis of the information in the routing table. This approach is well suited for the dynamic environment observed in the MANETs, as the route updates are periodic, easy and flexible. The disadvantage of this technique is that all the nodes are required to store and maintain the routing information and neighbor connectivity via periodic Hello messages [33]. Ad hoc On-Demand Distance Vector (AODV) [42, 43] and Temporally Ordered Routing Algorithm (TORA) [44] are the examples of hop-by-hop routing protocol.

The **Ad hoc On-Demand Distance Vector (AODV)** routing protocol is the most widely used protocol and is based on DSDV and DSR algorithms. It has adopted periodic Hello messages and sequence numbering mechanism from DSDV and route discovery using flooding mechanism from DSR. The use of sequence numbering overcomes the DSR's shortcoming of identifying the stale routes and thus, AODV can discover fresh routes to the destination. AODV is used as the base routing mechanism for the simulations, tests and validation purposes within this work, as it exhibits better performance in highly mobile scenarios with high load. Therefore, we will discuss this protocol in greater details.

The reactive routing protocol operates in two phases: a *route discovery phase* and a *route maintenance phase*. The route discovery phase is initiated when a source node wants to send data to a particular destination and the route to this destination is not available in its routing table. The source node floods the network with Route Request (RREQ) message. Each intermediate node on reception of RREQ creates reverse route entry for the source node in their routing tables and checks if it has the route to the destination. In case the route to the destination is not available, the intermediate node rebroadcasts the RREQ. In this manner, the reverse path is created in route discovery phase by the flooding of RREQ messages. The destination node after receiving RREQ responds by sending unicast Route Reply (RREP) message to the source node. So, forward path is created using unicast RREP message. After the route establishment, the data is unicasted on the available route.

The sequence numbers are used to avoid the counting-to-infinity problem and to determine the freshness of the route. A larger destination sequence number indicates the most recent route. All RREQ and RREP packets carry the destination sequence number. On reception of these packets, the node checks for the sequence number and update the routing information only if its value is larger than what it has in its routing table. So, the route with the highest sequence number is preferred. In case, multiple routes are discovered to the same destination with the same sequence number, the route having a minimum number of hops is preferred.

During the route maintenance phase, the nodes monitor the connectivity with the next hops in active routes using periodic Hello message. If any node moves out of existing active rout, the link failure is detected while attempting to forward a data packet to the next hop or using Hello message. Upon link failure detection, Route Error (RERR) is initiated by the node in the forward path closer to the source of the link breakage. RERR is then forwarded to all the affected nodes that were using this broken link. If a source receives an RERR it can reinitiate the route discovery process if it still needs to transmit data. Also, if the routes are not used within active route timeout interval, they are deleted from the routing table.

Advantages: The main advantage of this protocol is that routes are discovered on demand that contributes to very low overhead and it is adaptable to dynamic networks. Also, the local movement of nodes has local effects i.e. during link breakage only affected nodes are informed and no global broadcast is needed. The sequence numbers are used for loop prevention and as route freshness criteria to the destination. Also, distance vector routing is simple and doesn't require large memory or complex computations.

Disadvantages: The initial communication during the route establishment is heavier than some other approaches and source node may experience large delays during this phase. AODV lacks an efficient route maintenance technique. If the intermediate nodes have a sequence number higher than the sequence number as that of the source node but smaller than the latest destination sequence number, the intermediate nodes lead to inconsistent routes due to the stale route entries. The high control overhead is observed while handling multiple RREQ messages in response to single RREP messages. Another disadvantage of AODV is that the periodic transmission of Hello messages leads to unnecessary bandwidth consumption [39]. However, the advantages of AODV falls on the heavier side outweighing its disadvantages and many multipath routing protocols have been proposed in the literature as an extension of either DSR or AODV protocols.

2.4.3 Hybrid Routing Protocols

The next category is hybrid routing protocols that combine the advantages of both proactive and reactive routing protocol and balances the latency and control overheads. The first group is the *flat hybrid routing protocol* where the route discovery is done using reactive approach and route maintenance employs proactive approach throughout the communication session. AntHocNet [45, 46] is an example of flat hybrid routing protocols. The second group is *hierarchical routing protocol* that increases the network scalability. The nodes in the close proximity work together and form a zone to reduce the route discovery overheads. This connectivity between the nodes in a zone is maintained proactively while routes to the node in a faraway zone are determined using reactive approach. A large number of hybrid protocols have been proposed based on this mechanism. However, the zone-based approach causes partitions in the network.

The main disadvantage of hybrid routing protocols is that being hierarchical in nature, the nodes that have high-level topological information has to maintain more routing information, which demands more memory requirements and power consumption by these nodes. Zone Routing Protocol (ZRP) [47, 48] and Hazy Sighted Link State (HSLS) [49] are the examples of hierarchical hybrid routing protocols.

2.4.4 Location-based Routing Protocols

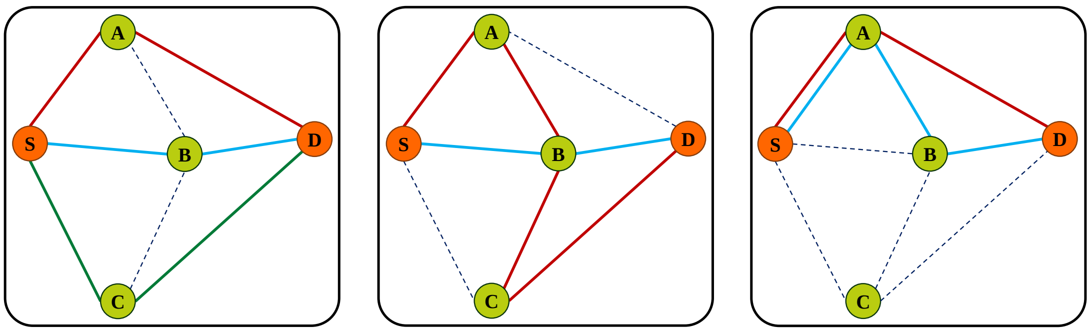
In this approach, the routing decisions are made on the basis of the actual geographic location of nodes that is determined by using the Global Positioning System (GPS) [50]. This routing approach does not require the route establishment, route maintenance or any storage of routing information. The source node typically selects the next hop for forwarding the data based on the physical position of the destination node. The information about the physical position of the nodes in the network is obtained through location-based services. The main advantage of this approach is that it doesn't require network wide searches unlike other protocols as the data can be transmitted directly towards the known geographical coordinates of the destination node. These enable location-aware routing protocols to quickly adapt to the route changes and provide a high degree of scalability. Location-Aided Routing (LAR) [51], Distance Routing Effect Algorithm for Mobility (DREAM) [52], and Greedy Perimeter Stateless Routing (GPSR) [53] are the examples of location based routing protocols. The disadvantage of this type of protocol is the retrieval of the location information to route packets towards the destination becomes very expensive in the mobile network [5, 6, 54].

2.4.5 Multiple Path Routing Protocols

The standard routing protocols in ad-hoc wireless networks determine a single path between source and destination. However, the discovered route may not be reliable due to node mobility, resulting in transmission failures and increase the delay at the nodes. This time delay observed while recovering from the link failures may not be acceptable to many time-sensitive applications. Also, the performance of single path routing protocols degrades sharply with the increase in the number of nodes [55].

To overcome these problems, multipath routing technique is proposed that aim to discover and utilize multiple paths between source and destination pairs. This approach increases the reliability of data transmission as it provides fault tolerance, reduces end-to-end delay, support load balancing and guarantees QoS assurance.

Multipath routing mechanism consists of three phases: *route discovery*, *route maintenance*, and *traffic allocation*. Route discovery and maintenance phases are responsible for finding and maintaining multiple paths between source and destination node and works in a similar way as discussed in earlier section. Multipath routing protocols can be implemented with node-disjoint, link-disjoint, or non-disjoint routing mechanisms [56]. In the node-disjoint technique, the multiple routes between source and destination do not share any node or link. In a link-disjoint approach, the multiple routes do not share any link, but may share nodes on the path. In the non-disjoint method, the multiple routes can share nodes and links. Figure 2.6 depicts the above three methods. Figure 2.6a represents the node-disjoint approach where the routes SAD, SBD, and SCD have no links or nodes in common. Figure 2.6b represents the link-disjoint approach where routes SABCD and SBD have node B in common but does not share any link. Figure 2.6c represents the non-disjoint approach where routes SAD and SABD have both, node A and link SA in common.



(A) Node-disjoint (B) Link-disjoint (C) Non-disjoint

FIGURE 2.6: Type of Multipath Routing Protocol [56].

Amongst the three multipath approaches, the multipath routes can be easily discovered in the non-disjoint method, as there are very fewer restrictions. It has been shown that in moderately dense networks, the number of node-disjoint routes between any two arbitrary nodes is small, especially as the distance between the nodes increases [57].

Load balancing is another very important criterion in MANETs. The traffic allocation phase defines the data distribution strategy over the discovered multiple paths in order to provide load balancing. Intelligent path selection enhances the performance of multipath routing. Basically, the multiple routes to the destination are segregated in terms of their characteristics e.g. from the fault tolerance point of view, more reliable paths should be selected to reduce the chance of routes failures. Such intelligent path selection plays an important role in providing QoS-aware routing. On the other hand, high complexity and overhead are the common disadvantages of multipath routing protocols over single path protocols. Split Multipath Routing (SMR) [58], On-demand Multipath Distance Vector Routing in Ad Hoc Networks (AOMDV) [59], and A Framework for Reliable Routing in Mobile Ad Hoc Networks (AODVM) [57] are the examples of multipath routing protocols.

2.5 Summary

In this chapter, we discussed the central idea behind the MANETs and its applications. Later we discussed the classification of the routing protocols in MANETs, which is the key section reviewed in this chapter. The proactive routing protocols provide the route to the destination without causing any latency. However, this approach is not suitable for MANETs due to the high control overheads. The reactive routing protocol overcome this problem and discovers the route on demand, but the latency observed during route discovery can be very high due to dynamic nature of MANETs. The hybrid routing protocols combine the advantages of the proactive and reactive routing protocols. However, it demands high power and memory requirements due to the hierarchical nature, as the nodes at a higher level have to maintain more routing information. The next discussed class was the location-based routing protocol that provides higher performance. However, the main disadvantage of this class is that the retrieval of the location information is an expensive affair from the MANETs point of view. The final approach is the multi-path routing protocols where multiple paths are established between source and destination. This routing mechanism increases the reliability, reduces end-to-end delay and, therefore, is well suited for the real-time applications where high QoS is required. So, we extended the idea of multi-path routing protocols to provide guaranteed QoS in MANETs that overcome the shortcomings of the routing protocols discussed above. In the next chapter, we discussed the concept of QoS in MANETs great details.

Chapter 3

Quality of Service Routing

In this chapter, we introduce the Quality of Service (QoS) routing in MANETs. In the beginning, we will discuss the QoS metrics, the NP-Complete problem and the major challenges in providing QoS routing in MANETs. In the next section, we reviewed QoS-aware routing protocol in MANETs and then the routing protocol based on ACO algorithm for MANETs, followed by the QoS-aware routing protocol for MANETs based on ACO algorithm. Lastly, the general drawbacks for QoS routing are presented followed by the summary that concludes the chapter.

3.1 Quality of Service

The widespread of communications and electronic mobile devices such as mobile phones, PDAs, tablet and laptops has popularized the idea of MANETs. These devices are capable to communicate and provide real-time multimedia applications such as audio/video calls, audio/video streaming, etc. So, with the ever-increasing scope of deploying real-time applications over MANETs, it has become very crucial to provide the Quality of Service (QoS) in these networks. This notion has transferred the research interests from providing best-effort service to a more complicated approach where QoS can be guaranteed in MANETs.

Various mechanisms have been proposed to estimate the QoS parameters based on different layers in the internet protocol stack. For example, at the physical layer, QoS strictly depends on the observed quality of the fluctuating wireless channel. The quality of the channel then determines the data rate and the packet loss rate as the QoS constraints on the physical layer. At the MAC layer, the QoS is determined by the fraction of time for which a node can successfully access and transmit a packet. The routing layer deals with the end-to-end communication. The QoS metric, therefore, depends on the combination of the individual metric value at each node of a multi-hop route. It is the responsibility of the network layer to find and maintain paths that satisfy the QoS constraints. In case if the network layer could not provide the QoS requirements then the transport layer and the upper layers have to include support for QoS [60]. Thus, QoS routing is very vital in order to provide QoS in MANETs.

3.2 QoS Routing in MANETs

According to RFC 2386 [61], "*QoS routing is defined as a routing mechanism under which paths for flows are determined based on some knowledge of resource availability in the network as well as the QoS requirement of flows*". QoS routing is the essential part of the network layer since it is responsible for collecting QoS parameters of the network and finding a suitable path between the source and destination that meet the QoS requirements for the given flow. On the other hand, a flow is defined as a stream of data packets traveling from a source to a destination with an associated QoS requirement. The QoS routing is much more complex since the QoS requirements vary significantly for different applications. So, it is essential that a QoS routing protocol supports various metrics reflecting the quality of the node, also known as QoS parameters

3.2.1 QoS Metrics

QoS metrics are defined as a set of the unit measurements that reflects the state of the node in terms of the available bandwidth, delay, jitter, packet loss probability, etc. Alternatively, the value of a QoS metric along the path is determined by the individual values of that metric at every intermediate node that forms the path. These metrics are broadly classified into three primary categories: *additive*, *multiplicative*, and *concave* [62]. Let $p(i, j, d)$ denote the path from node i to destination d through a neighbor node j as shown in figure 3.1. The symbols A, M, and C are the additive, multiplicative, and concave metrics respectively. These metrics are described as follows.

FIGURE 3.1: An example of the path from nod i to destination d via neighbor node j

- Additive metrics: The law of additive metrics is that the value of this metric is the summation of the values observed at each node in the path. Delay, delay variation (Jitter), and a number of hops are examples of additive metrics.

$$A_{ijd} = \sum_{n \in p(i,j,d)} A(n) \tag{3.1}$$

- Multiplicative metrics: The multiplicative metric is defined as the metric whose value over a path is the product of the values observed at each node corresponding to the path. Packet loss, stability and reliability are examples of multiplicative metrics.

$$M_{ijd} = \prod_{n \in p(i,j,d)} M(n) \tag{3.2}$$

- Concave metrics: The value of a concave metric over a path corresponds to the minimum value observed at the node on that path. Bandwidth, data rate (channel transmission speed), maximum transmission unit (MTU), and power are examples of concave metrics.

$$C_{ijd} = min(C(n)) \qquad \forall n \in p(i,j,d) \tag{3.3}$$

3.2.2 NP-Complete Problem

Optimization of QoS routing while considering only one QoS metric from additive, multiplicative, or concave at a time is achievable i.e. the underlying routing algorithm satisfies the given QoS metric in a reasonable time. On the other hand, finding QoS routing considering multiple QoS metrics is inherently hard. The problem of finding a path considering two or more additive and multiplicative metrics in any possible combination is NP-Complete (non-deterministic polynomial time) problem. The constraints in the NP-complete problem cannot be satisfied within a reasonable time, i.e. the algorithm may keep on running forever without converging. Thus, the only feasible combination is the concave metric with any one of the additive/multiplicative QoS metric; and such a problem is said to be a polynomial (P) class that can be satisfied within a reasonable time [63, 64].

One approach with respect to the combination of two or more constraints is to use mixed metric approach [63]. It defines a function i.e. a single metric is generated that represents the combination of multiple QoS metrics and this single metric is then used as the base for making the routing decisions. For example, a mixed metric *MIX* defined as a single metric that combines bandwidth B, delay D and packet loss probability L and is represented with a formula $MIX_{ijd} = D_{ijd}/(B_{ijd} \times L_{ijd})$. Thus, the path $p(i, j, d)$ having the metric with a higher value is likely to be a better choice in terms of bandwidth, delay and loss probability. Though, the mixed metric approach is a tempting heuristic, it can be best used to generate hints that will ease the path selection stage.

Apart from the NP-Complete computation problem, MANETs faces a large number of challenges in providing QoS as discussed in the next section.

3.2.3 Challenge in Providing QoS Routing in MANETs

For providing QoS in MANETs, the wireless link state information should be available and manageable. However, link state information is affected by the neighboring links activity. Also, MANETs must guarantee QoS on a multi-hop wireless networks with mobile nodes. In this section, we summarize these unique characteristics that act as major challenges in providing QoS routing in MANETs. We classify these challenges into the following three groups; challenges related to multimedia applications requirements, challenges imposed by the wireless medium, and the traditional challenges [62, 65–67].

1. **Multimedia application challenges**: The strict requirements set by the applications to provide a satisfying experience to the user act as the challenge for the network. The multimedia applications demand the stringent bandwidth, delay, and jitter requirements. The excessive delay, jitter and bandwidth constraints are compensated by providing the buffering functionality at receiving end point. However, it introduces an initial delay and this delay should be kept minimum. For one-way streaming of stored audio/video on demand, such delays do not cause considerable problem. But it is a big challenge for two-way conversational streaming that relies on strict delay constraints. Also, the multimedia contents differ from general data due to the large content of information. So, while transmitting such a large audio/video file the sender application tries to utilize all the available bandwidth. This causes the problem of network congestion.

   Mixed traffic is another problem that needs to be addressed in order to provide QoS communication. Different applications like voice and video communication, File Transfer Protocol (FTP), etc. generates traffic flow having variable characteristics like packet size, priority, sampling rate, etc. As the scale of MANETs increases, this nature of traffic becomes more obvious and might act as the hindrance in achieving QoS.

2. **Wireless medium challenges**: The wireless channels are susceptible to the phenomena such as shadowing and fading both of which reduces the signal quality and

thereby increase packet error rates. Fading is caused by multiple interfering reflections of the same signal. All these phenomena lead to another central challenge; the estimation of time-varying link state characteristics. The other challenge observed is the channel contention problem. This problem occurs when the packets from two or more senders that are not in the communication range of each other collide at the common intermediate node. This situation mandates the retransmission of packets as the packets are discarded on collision. Thus, it is highly desired to ensure that the minimum number of nodes contend for the low-bandwidth link while the comparatively large number of nodes can contend for the stable high-bandwidth link. The problem of contention leads to the network congestion, waste of precious network resources and energy drainage of the device.

3. **Traditional challenges**

   - **Mobility induced challenges**: When nodes move, the link parameters might change or they might even break. So re-routing is required in case of route failure while maintaining the stringent application requirements. Also, the mobile nodes may join, leave or rejoin at any location and at any time. This might interfere with the ongoing communication between the adjacent nodes.

   - **Multihop induced challenges**: Multihop causes the linear increment in the end-to-end delay with the addition of every hop on the path. Multihop also causes interference between the nearby links.

   - **Resource constrained devices**: The mobile nodes in MANETs are resource constrained and heterogeneous in nature. The efficient mechanisms are required to choose amongst the heterogeneous nodes with different capabilities and resource constraints, so as to meet the requirements set by the resource-demanding multimedia application.

   - **Heterogeneity**: MANETs by definition is a heterogeneous network that constitutes various types of mobile nodes possessing diverse communication technologies. These nodes differ in their capacities and computational abilities. Therefore, each node might have different routing responsibilities, network activities and energy consumption rates.

   - **Lack of central control**: MANETs does not have provision for central control. So, the nodes have to manage the network by mutual negotiations which increase the overhead for the already resource constraints nodes. Also, this negotiation process is further complicated due to the frequent topology changes.

Due to the above mentioned inter dependent challenges that are caused by the unpredictable conditions, providing truly hard QoS guarantees in MANETs are not practical.

3.2.4 QoS-aware Routing Protocol for MANETs

The quality of the communications service in MANETs has been tackled in many different approaches so far. The most significant challenge can be summarized as the estimation of link QoS parameters that depends on different parameters like node mobility, lack of precise state information, fading and shared radio channel [68].

The Ad hoc On-Demand Distance Vector (AODV) [43] routing protocol provides quick adaptation to dynamic topology, less memory overhead and low network resource utilization. Considering these characteristics, Pradeep et al. [69] propose some extension to AODV to support QoS and load balance features (QoS-AODV). The QoS-AODV extension includes cost and delay fields to the route request and route reply message. During route discovery phase, the intermediate node rebroadcasts route request packet if the node traversal time is less than the remaining delay in the Delay field. The load balance is achieved when source node receives multiple copies of route replay and selects the route with minimum cost. Each intermediate node along the establish path determines whether the requested QoS requirements cannot be maintained, or the route is not available anymore. Depending on the situation, the node originates a route error message to inform the source node about the QoS violation or link failure occurrence. However, the QoS parameter calculation and the detection mechanism for QoS violation are not illustrated thoroughly.

Sarma et al. [70] developed a Route Stability based QoS Routing (RSQR) protocol for MANETs with the following key parameters. The link stability model that depends on node mobility, computes the link stability by measuring the received signal strengths (RSS) from the MAC layer. The QoS-aware routing is done with throughput and delay as a constraints. Here, the routing incorporates the stability model and Admission Control to find the QoS route with highest stability. Admission Control ensures that at each hop of the path, the requested minimum throughput and maximum end-to-end delay constraints are satisfied. The available bandwidth at each node is estimated by subtracting the total consumed bandwidth in the interference range from the raw bandwidth. On the other hand, the pair of time stamped Hello messages are used to calculate per hop delay. Lastly, QoS violation and detection model keeps the check on QoS requirements. During the data transmission session, the destination node checks end-to-end delay violation through computation of one-way delay encountered by the data packets. If one-way delay for the received packet is greater than maximum end-to-end delay, the delay violation of the packet is detected by destination and it is confirmed after identifying certain number of continuous delay violation. The possible throughput violation is discovered by observing the absence of data packets in a flow at destination for a time interval called throughput reservation timer and it is triggered if no data packet of a flow is received within this interval. This proposed protocol is based on an enhancement of AODV routing protocol for MANETs. However, this protocol demands usage of pair of Hello messages for the calculation of QoS parameters that incurs an additional overhead. Also, the data packets are appended with additional packet field for conveying the time stamp values. Finally, the correct setting of the preemptive threshold for received signal strength for calculating stability is the main challenge of this algorithm.

Balachandra et al. [71] presented a multi-constrained and multipath QoS aware routing protocol (MMQARP) for MANETs. MMQARP is an extension of AODV routing protocol. The routing decision depends on three QoS constraints, which are the route reliability, delay observed on the link, and energy efficiency of nodes. These QoS constraints are used to compute node-disjoint multiple paths from source to destination. The timestamp associated with every received RREQ packets determines the average delay, whereas the link lifetime corresponds to the reliability. The link lifetime or link expiration time is calculated using the geographical information such as position and movement information. During the route discovery phase, the node drops the RREQ packet to avoid unnecessary flooding if the RREQ does not satisfy QoS constraints. However, the protocol has assumed the time synchronization between the nodes to calculate the average delay and the availability of the geographical information to estimate the correct position and speed of the mobile node.

Palaniappan et al. [72] proposed a reliable and energy-efficient routing mechanism for MANETs. This mechanism uses the QoS monitoring agents to calculate the following listed QoS information that in turn is used to determine the reliability of the link. The QoS metrics considered in this mechanism are link expiration time (LET), probabilistic link reliable time (PLRT), link received signal strength (LRSS) and link packet error rate (LPER) to estimate the link reliability, calculation of the residual battery power at each node for determining energy conservation. The next step is the initiation of the stable multipath route discovery phase that is based on the source routing and adaptive ad hoc on-demand multipath distance vector (AOMDV). In this phase, multiple paths between source and destination are established. Finally, the optimal path is determined by the route selection probability (RSP). RSP is based on the Fuzzy Logic System (FLS) that uses the estimated value of the QoS parameters mentioned above as input, and the RSP is given as output. The preemptive route repair phase detects the mobile node that might cause the communication failure and as the precautionary measure, it selects an appropriate node and initiates fast local route repair. However, for this preemptive route repair mechanism, the position information of the mobile nodes and the exchange of Hello messages are employed to collect the topological information that might cause high consumption of the resources of the network.

Katsaros et al. [73, 74] proposed a position-based routing protocol for urban vehicular environments, named Cross-Layer Weighted Position based Routing (CLWPR). As the name suggests, the protocol uses the position information of the nodes and cross-layer mechanisms between the PHY layer and MAC layer to improve the efficiency and reliability, respectively, of the routing protocol. The cross-layering mechanism keeps track of the PHY layer parameter like signal-to-interference-plus-noise ratio (SINR) value of the received packet using hello message, and the frame error rate is calculated in the MAC layer. The protocol supports traffic balancing by considering MAC queuing information in terms of node utilization for providing better QoS. The protocol also addresses the problem of network disconnection due to high mobility by buffering the packets with the carry-n-forward mechanism. Although this mechanism increases the packet delivery ratio, it also increases the end-to-end delay and, therefore, is not recommended for QoS-sensitive services. However, the protocol depends on electronic maps to predict the position of the mobile nodes.

Obaidat et al. [75, 76] proposes a QoS-aware Multipath Routing protocol (QMRP) for MANETs based on the single path AODV routing protocol. The proposed protocol aims to minimize the end-to-end delay in order to implement QoS multimedia applications in MANETs. The feedback from cross-layer communications between PHY, MAC and routing layers is used to compute the node-disjoint multiple paths. This method helps to achieve soft QoS [77] against the network and channel dynamics without requiring any additional resources. QMRP enhances the regular route broadcasting packet with an addition of the following two fields: the Expected Path Delay field (EPD) and a load field. EPD, which is the cumulative delay, is initialized to zero while the load field, which is the new load that will be added to the network, is initialized to the new amount of traffic that will be introduced into the network. The EPD contains various parameters from the MAC layer such as received data rate and current queue size and SNR from the physical layer. The delay computation is calculated using the average queuing delay instead of only using the current queue size. QMRP improves the AODV significantly as it modifies the stages of route discovery, route selection and route maintenance. However, QMRP incurs additional routing overhead as compared to AODV due to the discovery of more than one path in each route discovery process.

Yang et al. [78] presented Bandwidth-aware Multi-path Routing (BMR) protocol that can construct two node-disjoint parallel paths for a source-destination pair based on the available bandwidth constraint. The bandwidth available at a node is computed at MAC layer using cross-layer mechanism. The objective function in selecting optimal two node-disjoint paths is path's maximum available bandwidth. Yang and Kravets [79] present a method to calculate path's available bandwidth. Here, each node computes local available bandwidth and CS-neighbor (carrier sensing neighbor node) available bandwidth. Local available bandwidth is calculated by observing the fraction of channel being idle for a constant period of time. The channel is considered to be idle if the node is not present in the following states. In the first state, the node is receiving or transmitting a packet. Secondly, the node receives RTS or CTS messages from the other node that reserve the channel for a period of time. Finally, the node senses the busy carrier with received signal strength larger than the Carrier-sensing Threshold. The c-neighborhood available bandwidth defined as the minimal local available bandwidth of all the nodes in a node's carrier sensing range. During route discovery phase, the CS-neighbor available bandwidth is computed after receiving RREP by intermediate node and the source node selects the two parallel paths that have the maximal path's available bandwidth. However, to compute the available bandwidth at CS-neighboring nodes, two kinds of approaches that can be used. The first approach is the active approach; CS-neighbors proactively exchange available bandwidth information between each other. Since the CS-neighbors may not be able to communicate directly with each other, such exchanges may impose relatively high message overhead. The second approach is the passive approach; a node passively monitors the channel to estimate its CS-neighbors local available bandwidth. Thus, the estimations of the local available bandwidth at CS-neighbors may not be accurate.

Kwon et al. [80] proposed reactive QoS routing algorithm using IEEE 802.11 multi-rate with the minimum overall timeslot consumption for video streaming through MANETs. The protocol is an extended version of DSR routing protocol. The protocol considering the number of contention neighbors for selecting optimal path that minimize consumed timeslots. It uses the ratio of the consumed timeslots to the total timeslots as a routing QoS metric. Thus, it is important to estimate how many timeslots will be available at each node at a fixed monitoring interval. Instead of exchanging control packets, the author used a local monitor that will periodically monitor the wireless channel to find busy time interval in which the received carrier signal strength is larger than the carrier-sensing threshold. During route discovery mechanism, a hello packet is used to maintain contention neighbor table, which include a 1-hop contention neighbor list and a 2-hop carrier sensing neighbor list. Thus, the estimations of the local available bandwidth using passive approach may not be accurate as explained in the previous approach.

Ali et al. [55] introduced the QoS-aware multipath threshold routing (QMTR) protocol, which aim to enhance QoS by avoiding congestion in MANETs. The multiple paths are selected based on three QoS parameter, which are link bandwidth, forwarding delay, and average load. The QoS parameter computed based on MAC layer using cross-layer approach. The local available bandwidth computed based on the channel's idle time ratio [81]. The load of the node estimated based on MAC layer channel contention information and the number of packets in the interface queue [82]. Finally, the forwarding delay estimated based on the method proposed in [83] that include the MAC layer contention and transmission delay. In this approach, if the available link bandwidth, the forwarding delay, or the average load at the node did not satisfy QoS threshold, the traffic is distributed over fail-safe multipath to reduce the load of a congested node. The fail-safe multiple paths are establish using scalable multipath on-demand routing (SMORT) protocol [84]. SMORT protocol aims to minimize the route break recovery overhead by finding fail-safe multipath. Here, all the nodes in the primary path including the source node are provided with an additional route to the destination. However, the proposed protocol relies on local QoS constraints and does not consider the end-to-end QoS constraints for multimedia application. Thus, the higher number of hopes constituting the fail-safe multipath may increase the end-to-end delay and exceed the threshold recorded by the requirements of the data stream. So, the local QoS exist but the end-to-end QoS, as a whole on the route is not satisfied. Thus, the availability of local QoS constraints is not sufficient to carry out the real-time multimedia application.

Lal et al. [85, 86] proposed a reactive QoS-aware routing protocol (QARP), which is a bandwidth-aware node-disjoint multipath routing protocol. The proposed approach finds paths that satisfy the bandwidth requirement of the requested application. The protocol adapts session admission control and cross-layer communication to locally estimate the residual bandwidth for each node. The estimated value is the product of the channel's idle time ratio (CITR) for two-hop neighbor and the maximum transmission rate of the channel. During the route discovery phase, the local available bandwidth at any node is calculated dynamically and their carrier-sensing neighbors are estimated by using the query messages. The protocol determines the routes by considering the effects of both inter-contention (contention between nodes on active paths of different data sessions) and intra-contention (contention between nodes belongs to the same path). QARP handles the violation created by temporary congestion, channel interference and mobility by keeping a backup path, performing local route recovery, avoiding routing through short-lived, low-quality links and regular monitoring of the active transmission routes. However, the protocol considers only one QoS parameter and therefore may be is not sufficient for multimedia communication with various requirements. Additionally, the route discovery phase produces high overhead and delay, since collection information from two-hop neighbor to compute available bandwidth at each intermediate node for the selected paths.

Table 3.1 contain summary of characteristic of QoS-aware routing protocol.

TABLE 3.1: The summary of characteristic of QoS-aware routing protocol

Cite	Algorithm	Based on	Type	Req. Type	Using Hello	Path Type	QoS Optimization	Drawback	QoS Maintenance
Pradeep et al. (2010) [69]	QoS-AODV	AODV	Reactive	Flood	Yes	Single	• Cost • Delay	No method given	Yes
Sarma et al. (2010) [70]	RSQR	AODV	Reactive	Flood	Yes	Single	• Stability • Delay • Bandwidth	Set signal threshold Overhead	Yes
Yang et al. (2011) [78]	BMR	n.a.	Reactive	Flood	n.a.	Multipath node-disjoint	• Bandwidth	Based on one QoS parameter	No
Kwon et al. (2011) [80]	n.a.	DSR	Reactive	Flood	Yes	Single	• Timeslots	Based on one QoS parameter	No
Katsaros (2011) [73]	CLWPR	OLSR	Proactive	n.a.	Yes	Single	• SINR • Queue length • Error rate	Position req.	Yes
Obaidat et al. (2013) [76]	QMRP	AODV	Reactive	Flood	Yes	Multipath node-disjoint	• Delay • Load	Overhead	No
Balachandra et al. (2014) [71]	MMQARP	AODV	Reactive	Flood	Yes	Multipath node-disjoint	• Reliability • Delay • Energy	Synchronization Position req.	No
Ali et al. (2014) [55]	QMTR	AODV	Reactive	Flood	Yes	Multipath fail-safe	• Bandwidth • Delay • Load	Overhead	Yes
Lal et al. (2015) [86]	QARO	AODV	Reactive	Flood	Yes	Multipath node-disjoint	• Bandwidth	Overhead Based on one QoS parameter	Yes
Palaniappan et al. (2015) [72]	n.a.	AOMDV	Reactive	Flood	Yes	Multipath node-disjoint	• LET • PLRT • LRSS • LPER • RBP	Overhead Position req.	Yes

n.a.: not available

3.3 Ant Colony Optimization

Ant Colony Optimization (ACO) is one of the most successful fields in swarm intelligence (SI) algorithms. In general, SI is a collective behavior of decentralized and self-organized systems, which include positive feedback, negative feedback, randomness, and multiple interactions. The ant colonies, flocks of fishes, and swarms of birds are some examples of SI [87, 88].

ACO algorithm, proposed by Dorigo and Stützle [89], is a metaheuristic approach from the family of multi-agent algorithms for solving combinatorial optimization problems such as QoS routing in MANETs. The main idea of this optimization algorithm is derived from the observation of how real ants optimize food gathering in the nature. ACO use artificial agents called ants that interacting locally with one another and with their environment. These agents coordinate their activities via *stigmergy*, which is defined as a method of indirect communication used by ants to modify the environment. Thus, the communication between agents, or between agents and the environment, is made by using chemicals substance produced by the ants called *pheromones*. The pheromone is a positive feedback mechanism when the ants found the food source and return to its nest, it will lay more pheromone on the ground. Ants are likely to follow paths identified by strong pheromone concentrations and the quality of the path is reinforced than from other paths that are less visited. The pheromone trails allow ants to find their way to the food source, or back to the nest. The evaporation of pheromone is adapt as a negative feedback mechanism that used to reduce the pheromone concentration automatically over the time. Ants in the colony have the same goal and walk randomly to search for the food sources. This behavior enables ants to find suitable and shorter paths from food sources to their nest.

Figure 3.2 shows a scenario where multiple routes exist from the nest to the location where food is placed. At the junction point (A), the first ants will randomly select their path to the food and at the same time will trail pheromone in order to guide the ants following them. The pheromone concentration on the path determines the freshness of the path i.e. the following ants will take the path with highest pheromone concentration. In our scenario, the straight path is the shortest path. So after some time the pheromone concentration on this path will be higher than the other existing paths as more and more ants will be taking this path. Thus, the shortest path will be determined and ultimately all the ants will use this path to reach to their destination. The same steps are applicable on their way back to the nest. Particularly, the dynamic nature of this approach exhibits high adaptation to the dynamic network topology, observed in mobile ad-hoc networks, where the link changes occur very often and therefore the link existence cannot be guaranteed [90].

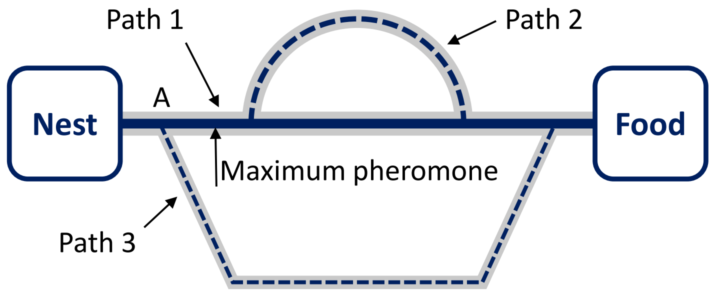


Figure 3.2: Pheromone concentration along the shortest path used by ants [90]

3.3.1 ACO Routing Protocol for MANETs

ACO algorithm demonstrated many desirable properties for MANETs such as flexibility adaptivity, robustness, and scalability. Additionally, ACO algorithms are highly distributed and self-organized in nature. Thus, ACO becomes an interesting approach to deal with the challenges in providing QoS routing in MANETs, as described in section 3.2.3.

ACO algorithm uses the colony of artificial ants and pheromone table. Ants are generated by the source node to find a path to a particular destination. While traveling from the source to the destination, ants collect information about the quality of the path and use this information to update the pheromone table at the intermediate nodes, on its way back to the source node. The pheromone table contains the probability values that reflect the goodness of the path to the specific destination. In terms of routing, the pheromone value could be easily related to the available resources on a particular path. As ACO provides traffic-adaptive and multipath routing, the data packets are forwarded according to a stochastic decision policy, i.e. hop-by-hop fashion, based on the information included in the pheromone tables. The other important aspect of ACO is its flexibility that is apparent, as it has been successfully implemented for a wide range of problems such as finding solutions to NP-Complete problems. All these features and qualities make routing based on ACO very desirable and suitable for MANETs.

Table 3.2 provides a detail comparison of various ant-based routing algorithms with respect to different parameters. The first ACO routing algorithm designed for wired network such as ABC [91] and AntNet [92]. Both these algorithms use proactive routing mechanism and are distributed, scalable and robust in nature. However, these algorithms lack adaptability when executed on dynamic networks like MANETs.

ACO routing algorithm can be classified into three main classes as explained in chapter 2 section 2.4: proactive, reactive and hybrid. In the proactive routing protocol (e.g., [91–97]), all the nodes continuously search and maintain the routes to all the other nodes by periodically flooding the network with control message. On the downside, proactive algorithms have a disadvantage in a highly dynamic network, as this algorithm becomes quite inefficient when many changes need to be tracked causing excessive overheads. Thus, this approach to achieve QoS routing in MANETs are not suitable. The second class is reactive routing protocols (e.g., [98–102]). It shows better performance for MANETs, since the flooding of route request packets is done only when a route is required, unlike the periodic flooding done in proactive routing protocols. The third class is a hybrid routing protocol (e.g., [46, 103–110]) that uses reactive route discovery mechanism and proactive route maintenance. Another group of hybrid routing protocol (e.g., [111]) is based on zone routing framework. However, hybrid routing protocol generates high overheads for maintenance of the available routes or the zone.

Another classification can be made according to router discovery mechanism, unicast where the next hop is selected randomly based on the probability or broadcast i.e. using flooding. However, the method of random selection suffers from the problem of stagnation [112]. Stagnation occurs when ants attracted the one optimal path that will lead this path to be a gradually congested and the dramatic decrease of the probability of selecting other paths. Thus, the likelihood to select another path witnesses high diminution. Another tradeoff is observed while executing the algorithm for predefined number of iterations. The small value might lead to no topology convergence while the large value will unnecessarily consume large amount of time to determine the topology [113]. Furthermore, each node uses Hello messages to locally maintain the connectivity between each neighbor for unicast mechanism.

Finally, one of the most important observations that can be made from the study of these algorithms is that none of these algorithms support QoS requirements. The optimization method adopted in these algorithms is based only on one or two parameter such as the number of hops, ant trip time, queue length, or number of ants that selects the path. But the QoS issues like end-to-end delay and jitter, available bandwidth, loss probability and error rate are not considered in this algorithms; this leads to the poor QoS performance and makes them unsuitable for real-time data and multimedia communications. Furthermore, in [114, 115] concluded that the minimum number of hops on a path does not determine the best route in terms of QoS requirement and exhibited low reliability and increase the possibility that they are broken. These paths may contain long-range links i.e. channels with low SNR values causing high packet loss and higher retransmissions packet at the node because of more reception to bit error.

Table 3.2: The detail comparison of various ant-based routing algorithms

Cite	Algorithm	Req. Type	Ant sending	Using Hello	Type Path	Optimization criteria
Schoonderwoerd et al. (1997) [91]	ABC	Proactive	Random	No	Single	Pheromone (The ant age)
Caro et al. (1997) [92]	AntNet	Proactive	Random	No	Single	Pheromone (Number of ant)
Gunes et al. (2002) [90, 98]	ARA	Reactive	Flood	No	Multipath	Pheromone (Hop count)
Marwaha et al. (2002) [103]	Ant-AODV	Hybrid	Flood	Yes	Single	Pheromone (Hop count)
Hussein et al. (2003) [93]	ARAMA	Proactive	Random	Yes	Single	Pheromone (Hop count and Energy)
Baras et al. (2003) [94]	PERA	Proactive	Flood	Yes	Single	Pheromone (Hop count or delay)
Roth et al.(2003) [95]	Termite	Proactive	Random	Yes	Multipath	Pheromone (Number of ant)
Caro et al. (2004) [46]	AntHocNet	Hybrid	Flood	Yes	Single	Pheromone (Delay and hop count)
Liu et al. (2004) [99]	EARA	Reactive	Flood	Yes	Multipath	Pheromone (Hop count and link capacity)
Kamali et al. (2007) [100]	POSANT	Reactive	Random	Yes	Multipath	Pheromone (Hop count, position information)
Rosati et al. (2008) [96]	DAR	Proactive	Random	Yes	Single	Pheromone (Number of ant)
Osagie et al. (2008) [97]	PACONET	Proactive	Random	Yes	Single	Pheromone and link age time
Yu et al. (2008) [104]	ACO-AHR	Hybrid	Flood	Yes	Multipath	Pheromone (Hop count or delay)
Wang et al. (2009) [111]	HOPNET	Hybrid	Random	Yes	Single	Pheromone (Number of ant) and link age time
Martins et al. (2010) [105]	Ant-DYMO	Hybrid	Random	Yes	Single	Pheromone (Hop count)
Villalba et al. (2010) [106]	AntOR	Hybrid	Flood	Yes	Multipath	Pheromone (Hop count)
Sujatha et al. (2010) [101]	PBANT	Reactive	Random	Yes	Single	Pheromone (Hop count)
Raval et al. (2011) [107]	Ant-CAMP	Hybrid	Flood	No	Multipath	Average queue length
Samadi et al. (2012) [108]	AMAR	Hybrid	Flood	Yes	Multipath	Pheromone (Hop count)
Kanani et al. (2013) [109]	Ant-AOMDV	Hybrid	Flood	Yes	Multipath	Pheromone (Hop count)
Canãs et al. (2013) [110]	AntOR-v2	Hybrid	Flood	Yes	Multipath	Pheromone (Hop count)
Gupta et al. (2014) [102]	E-DYMO	Reactive	Flood	Yes	Multipath	Pheromone (Number of ant)

3.3.2 QoS-aware Routing Protocol for MANETs using ACO

A number of protocols have been proposed to achieve QoS using ACO. These protocols are enabled with additional feature of adaptability that makes them flexible towards the changing environmental conditions and increases their capability against the failures and damages occurring in the network [88].

Liu et al. [116] proposed reactive routing protocol that combines SI and link-disjoint multipath routing protocol, Ant colony based Multipath QoS-aware Routing (AMQR). The Link-disjoint Multipath method is a stronger approach and can provide better QoS than that achieved by single paths. The main idea of this protocol is that the source node collects and stores the information about the paths followed by the ants and uses this information to construct a topology database of the network. This central database empowers the source node to construct link-disjoint multipath over which the data packets are sent. In addition, each node monitors the congestion state based on the time delay of the return acknowledge ants from the destination node. The protocol identifies the route failure through a missing acknowledgment at the MAC layer. The proposed protocol is based on delay (arrival time at each node) and number of hops to compute the amount of pheromone. However, the major limitation of this protocol is that it does not provide thorough information for delay calculation. Also, the use of acknowledges ant incurs the overhead and might overwhelm the network. Finally, the source node has to manage the central database for network topology graph whose maintenance becomes difficult with the increase in the number of nodes.

Wang et al. [117] proposed Ant Colony Optimization and Ad-Hoc On-Demand Multipath Distance Vector (ACO-AOMDV) routing protocol for MANETs. This protocol is the improved version of Ad-hoc On-demand Multipath Distance Vector (AOMDV) [59] protocol. The additional overhead is incurred while computing the pheromone values on multiple paths. The pheromone value calculation on the new path depends on the following three routing parameters; the average link count of the path, the average load of the path, and the hops of the path. The average link count of the path is the ratio of summation of neighbor node-link count (NLC) to the number of intermediate nodes in the path. The NLC is continual connection time, where the node can get partial link information by periodically sending Hello message and receiving other control messages. The second parameter is the average load of the path, which is computed as the ratio of the summation of the difference between the max queue length and the current queue length at the node to the number of intermediate hops in the path. The last and most important parameter is the number of intermediate hops in the path that allows the routing protocols to find the path with minimum numbers of hops. The paths with least number of hops have higher reliability against link failures as compared to the path with large number of hops. During the route maintenance phase, the periodic generation of the control packets and the exchange of Hello message for NLC computation increases the overhead and consumes the resource of the network.

Liu et al. [118, 119] presents an Emergent Ad Hoc Routing Algorithm with QoS provision (EARA-QoS). It is a hybrid routing algorithm for MANETs that combines reactive approach for finding multipath with periodic ant generation for path maintenance. The protocol extended based on EARA routing protocol [99]. However, EARA-QoS uses the biological concept of stigmergy to reduce the amount of control traffic. This algorithm utilizes the cross-layer multiple criteria metrics. These multi criteria routing decisions provide better usage of network characteristics in choosing best routes among multiple available routes and avoid forwarding data packet through the congested areas. The proposed protocol computes QoS parameters based on MAC layer. The first parameter is the average MAC layer utilization defined as the time interval, during which the channel is busy in a given time window. The second parameter is the heuristic transmission queue that is the ratio of the number of packets in the interface queue to the maximum capacity of the queue. Lastly, the average MAC layer delay defined as the period of time between transmission of RTS frame from node i to the successful reception of the data frame at node j during a given time window. The degree of interference is determined by this average MAC delay. EARA-QoS has an integrated lightweight QoS provision scheme. It classifies the traffic flows into different service classes that are defined by their relative delay bounds. Therefore, the delay-sensitive traffic such as real-time traffic is given a higher priority than other less sensitive traffic flows. However, this protocol has some restraints. The local connectivity managements for all neighbor nodes is done using Hello packet. Additionally, during route maintenance phase, the periodic ant packets are used for the maintenance of the available routes from source to destination. All these packets increase the overhead and might also interfere with the regular data packet.

Belkadi et al. [120] proposed adaptive and intelligent QoS routing protocol using ACO. The proposed protocol combines QoS routing protocol with a flow control mechanism to avoid congestion and packets losses. The QoS routing protocol is responsible to find a path that has a larger bandwidth, smaller delay and better stability. The link life duration reflects the stability of the path that depends on the direction, speed and the energy of the node. The protocol based on [121], which provides a method to compute delay and bandwidth. The transmission delay is computed by considering the time difference between transmission and reception of Hello message between the neighboring nodes. For this computation the protocol assumes that all the nodes are synchronized. The local available bandwidth is computed considering the idle time period of wireless channel. The protocol uses random selection method to forward ant packet, and this method requires dynamic maintenance of the neighbor nodes. Therefore, this approach incurs significant overheads. To avoid congestion, a flow control mechanism is employed to adapt the transmission rate for each route. This mechanism informs the source node about the smallest bandwidth value available on the link constituting the path.

Deepalakshmi et al. [122] proposed the ant-based multi-objective QoS routing (AMQR) protocol for MANETs to support multimedia communications. This is reactive routing approach that includes two phases namely route exploration and route maintenance. In the routing exploration phase, the protocol floods the network with control messages to determine the multi-paths based on different QoS parameters like shortest path length (number of hops), minimum delay, and higher bandwidth. The path with the highest preference probability is selected to send the data to the desired destination. The available bandwidth of the link is determined as the ratio of the size of Hello message to the time interval observed between its transmission and reception. The end-to-end delay between source and destination is calculated using the round trip time taken by the route request packet. In route maintenance phase, the protocol deals with the link failures. For this AMQR proposes three tables, one for neighbor nodes other for routing and the third table for the path preference. In neighbor table, the amount of pheromone and available bandwidth of outgoing links for each neighbor nodes are saved. The routing table has the information about the sources, destinations and best next hop to the indicated destination. Finally, the path preference table contains an entry for a destination associated with a number of neighbor nodes. The best neighbor node that has a higher probability will be copied to the routing table. However, the protocol utilized the periodic Hello messages to detect link failure and to evaluate the available bandwidth for each neighbor nodes that increase the overhead in network. In addition, it assumes that all nodes in MANET are synchronized for computing QoS parameters.

Kim [123] introduced a multi-path routing strategy based on ACO algorithm. During the path discovery phase, each node sends an ant packet randomly from one node to another node. The forward ant selects next hop based on the distance and queue length available at the neighbor node. In addition, each node keeps a routing table and periodically transmits Hello message to maintain local connectivity with neighbor nodes. The proposed routing protocol chooses the adaptable paths that satisfy the QoS constraints in term of bandwidth and delay. For transmitting the data packets through established multiple paths, the source node can make a decision about a particular path on the basis of its adaptability and energy level. This multipath routing mechanism can help to spreads the traffic load along multiple available paths, thus, providing load balancing which in turn improves network reliability. For the sake of providing dynamic network management, the control parameter decisions in the proposed algorithm are dynamically modified by using real-time approach. However, the proposed algorithm relies on the fact that the bandwidth and delay information are available beforehand. Additionally, the protocol needs to periodically look after the maintenance in order to ensure local connectivity with neighboring nodes.

Krishna et al. [124] proposed a QoS-enabled ant colony based multipath routing (QAMR) protocol for MANETs. The path selection algorithm is based on the next hop availability (NHA) and the path preference probability. The NHA considers both, mobility and energy factor to find the goodness of the links and the nodes; and these has to satisfy the threshold set up by NHA. For the path preference probability the different parameters such as delay, bandwidth and number of hops are measured. The measured value is then used to determine the path preference probability. Thus, the path with given QoS constraint and with the optimal path preference probability is selected for transmitting data between source and destination. The paths are interpreted in a pheromone table that shows their respective quality. In addition, the protocol maintains two tables, routing table that contains information about next hop for data forwarding and pheromone values, and neighbor table that contain information about all the neighboring nodes. However, the main drawback of this protocol is that the algorithm does not provide the detail information for calculating the available bandwidth. The end-to-end delay is again computed based on the trip time taken by the route request packet traveling from the source to destination and therefore assumes that all the nodes are synchronized. In addition, the periodic information needs to be exchanged between neighbor nodes in order to compute the link stability that incurs overhead.

Balaji et al. [125] Proposed MANET routing protocol based on AODV and ACO, named AODV-ACO. This protocol offers a new link quality metric to handle link quality between nodes to evaluate routes, as an enhancement to the existing AODV routing protocol. Link quality between the two neighbors can be assessed based on received signal strength. Here, the regular Hello messages are extended to a new packet Link Quality Format (LQF). It is a link quality integer metric that defines the link quality between the neighboring nodes. This extension of Hello messages enables the ACO technique to discover the local minima and the local maxima that help to identify the best route among the available routes. However, the local link quality does not guarantee end-to-end QoS for the path since the chosen neighbor with high quality may not produce a path with high quality to destination.

Nivetha et al. [126] proposes the combination of two stochastic optimization methods namely, Ant Colony Optimization (ACO) and Genetic Algorithms (GA); and named it as ACO GA Hybrid Meta-heuristic (AGHM) algorithm in order to reduce the complexities in the dynamic environment. For the given network topology, all the probable routes from source to destination are found using ACO. In the next step the set of all the routes is formed based on the pheromone concentration on the routes deposited by the artificial ants. This set of routes will act as the initial population to be used by GA. Now, for any source destination pair in the network, the set of optimal paths are identified from the initial population on the basis of the fitness function and genetic operations. This GA iteration is continued for either of the following two events. The first case is if the predefined number of generations is not reached and in the second instance is if there are no exclusive offspring included in the new population for three consecutive times. As the algorithm ensues, it rejects the weaker solutions and therefore the subsequent population will have the optimal set of paths needed for multipath routing. The amount of pheromone is calculated on the basis of bandwidth, delay and hop count to calculate the amount of pheromone, while the heuristic information is based on the distance between adjacent nodes. However, the proposed algorithm does not provide the information about the delay and bandwidth calculation. Also, the ACO and GA algorithms might have to go through large number of iterations in order to find the optimal path and thereby consuming enough amount of time, which is not desirable for the multimedia applications.

Table 3.3 contain summary of characteristic of QoS-aware based on ACO.

3.4 General Drawback for QoS Routing Protocol

This section provides general shortcomings observed in the measurement techniques adopted by the QoS-aware routing protocols presented in table 1 and 3.

1. A significant group of QoS routing protocol (e.g., [69, 116, 123, 124, 126]) assume that the QoS parameters are readily available i.e. the methodology to measure or estimate QoS parameters is not provided. Thus, we cannot assess the effort used in calculating the achievable QoS.

2. In a contention-based MAC protocol the node transmits the data only when it senses that the channel idle. The local available bandwidth is then determined as the product of the channel idle time and the transmission rate. In this context, Yang et al. [79] explain the problem of available bandwidth estimation in a CS-range so that all the nodes in the CS-range must be considered when estimating a node's available bandwidth. The following two approaches are employed to calculate the channel idle time that will then provide the available bandwidth at CS-range. In active method (e.g., [55, 86, 119, 120]), 2-hop Hello messages are used to determine the channel idle time. The drawback of this approach is the small hop count that may not cover all the nodes in the CS-range in some topologies. So, by taking advantage of the power control capabilities, the Hello messages send with

TABLE 3.3: The summary of characteristic of QoS-aware based on ACO

Cite	Algorithm	Based	Type	Req. Type	Using Hello	Path Type	QoS Optimization	Drawback	QoS Maintenance
Liu et al. (2005) [116]	AMQR	n.a.	Reactive	Random	Yes	Multipath link-disjoint	• Delay • Hop count	No method given Central databases	Yes
Wang et al. (2008) [117]	ACO-AOMDV	AOMDV	Reactive	Flood	Yes	Multipath link-disjoint	• Link count • Load • Hop count	Overhead	No
Liu et al. (2010) [119]	EARA-QoS	EARA	Hybrid	Flood	Yes	Multipath	• Utilization • Delay • Queue Size	Overhead	Yes
Belkadi et al. (2010) [120]	n.a.	n.a.	Reactive	Random	Yes	Multipath	• Delay • Bandwidth • Stability	Synchronization Overhead Position req.	Yes
Deepalakshmi et al. (2011) [122]	AMQR	n.a.	Reactive	Flood	Yes	Multipath used one	• Delay • Bandwidth • Hop count	Synchronization Overhead	No
Kim (2011) [123]	n.a.	n.a.	Reactive	Random	Yes	Multipath	• Delay • Bandwidth • Energy • Distance	No method given Overhead Position req.	Yes
Krishna et al. (2012) [124]	QAMR	n.a.	Reactive	Flood	Yes	Multipath	• Delay • Bandwidth • Hop count	Synchronization No method given	Yes
Balaji et al. (2014) [125]	AODV-ACO	AODV	Reactive	Flood	Yes	Single	• Signal strength	Overhead	No
Nivetha et al. (2014) [126]	AGHM	n.a.	Reactive	Random	Yes	Multipath	• Delay • Bandwidt • Hop coun • Distance	No method Position req.	No

n.a.: not available

higher power level than normal data transmission. The outcome of this solution, the Hello message from the sender can reach all the nodes in CS-range in the cost of high power consumption. Moreover, the method of the exchange Hello message results in large message overhead. In the passive approach (e.g., [78, 80]), the node passively monitors the channel. This approach is promising as compared to active approach as it does not incur any additional overheads; the cost is also low and does not require high power transmissions. However, this approach lacks accuracy while estimating the bandwidth.

3. Another approach assumes all the nodes in MANETs are synchronized to compute QoS parameters (e.g., [71, 120, 122, 125]). However, synchronization signaling incurs extra overhead, and as stated in previous work.

4. More routing protocol uses periodic Hello message for computation of QoS parameters and for discovery and maintenance of the neighbor nodes. The use of Hello messages consumes available bandwidth and energy. Also, the computations based on the control message exchange are too slow to react to a fast topology change. This makes the estimation or prediction of QoS parameter difficult or stale. Furthermore, these messages might conflict with regular data packets causing increasing delay and congestion in the network.

5. Position-based routing protocols (e.g., [73, 100, 101]) utilize the geographical position of the nodes. A group of protocols selects the next hop based on the distance between two nodes (e.g., [123, 126]) while another group exploits the information about the node's position, its speed and direction to compute the reliability (e.g., [71, 72, 120]). This geographical information is typically gathered via Global Positioning System (GPS) and other location services. However, to determine the mentioned physical information is expensive in terms of the necessary equipment or message exchange [88].

6. The local link quality information (e.g., [55, 73, 125]) does not guarantee the end-to-end QoS constraint for the path since the chosen neighbor with high quality may not produce a path with high quality to the destination.

7. The routing protocols discussed in [78, 80, 86] consider only one QoS parameter, and therefore, these routing protocols may be not suitable for carrying out multimedia applications that demand variable QoS constraints.

8. Another set of routing protocols determines the single path [69, 70, 73, 80, 125] from source to destination. However, for high communication session on this path, it might suffer from congestion and delay, thus, reducing the throughput and

reliability of the network. Also, in the case of link failures on the path, high overheads are incurred for route reestablishment.

9. Most of the QoS-aware routing protocols establishes the path that satisfies QoS constraints and does not consider the problem of congestion that might occur during the communication session.

3.5 Summary

In this chapter, we discussed the QoS metrics in MANETs that is followed by the brief review of the NP-Complete problem. Next we discussed the challenges in providing QoS routing in MANETs and classified these challenges into three groups namely, challenges related to multimedia applications requirements, challenges imposed by the wireless medium, and the traditional challenges. In the next section, we provided working synopsis of different QoS-aware routing protocols in MANETs and their shortcomings. Later we discussed Ant Colony Optimization and reviewed a number of Ant Colony Optimized routing protocols and presented the findings in the Table. In the next section, the literature survey on QoS-aware routing using ACO for MANETs is discussed and presented in the Table. Finally, we discussed the general drawbacks observed in the measurement techniques adopted by the various QoS-aware routing protocols. In the next chapter, we will discuss our proposed QoS-aware routing protocol and explain its advantages over the discussed QoS-aware routing protocols.

Chapter 4

QoRA Approach

This chapter presents an overview of our QoS-aware routing approach called as "**Qo**S-aware **R**outing based on **A**nt Colony Optimization" or QoRA. The QoRA architecture consists of two primary components QoRA Entity and SNMP Entity. The chapter begins by pointing out the most important aspects of QoRA and its components. In the later sections we will describe the cooperation between QoRA entity and SNMP agent followed by the in depth description of the functioning of QoRA protocol. Finally, we conclude the chapter with the summary.

4.1 QoRA Approach: Overview

QoRA [127–130] is a reactive, non-disjoint multipath routing protocol. In QoRA, the routing decisions are based on the three QoS constraints, namely, bandwidth, delay and expected success rate. These parameters are readily available from the SNMP agent. Also, the QoRA approach is highly flexible as it can consider more QoS parameters. These three QoS constraints together determine the objective function. The objective function is then used to find the appropriate outgoing link for the given QoS requirements. The information required to identify the characteristics of the link in terms of objective function could be obtained using the measurements technique. However, this approach would generate additional traffic influencing the characteristics of the links significantly. Thus, we decided to retrieve information from an already existing entity used for network management, i.e., the SNMP agent.

4.2 QoRA Approach: Architecture

Figure 4.1 demonstrates the QoRA architecture and the information exchange between its components. As displayed in the schematic diagram the architecture has two main components. The first element is the QoRA Entity, which runs on each node to identify suitable paths as per the given QoS requirements. The second component is the SNMP Entity consisting of SNMP agent and Management Information Base (MIB). We describe both these entities and the corresponding sub-components in the later sections.

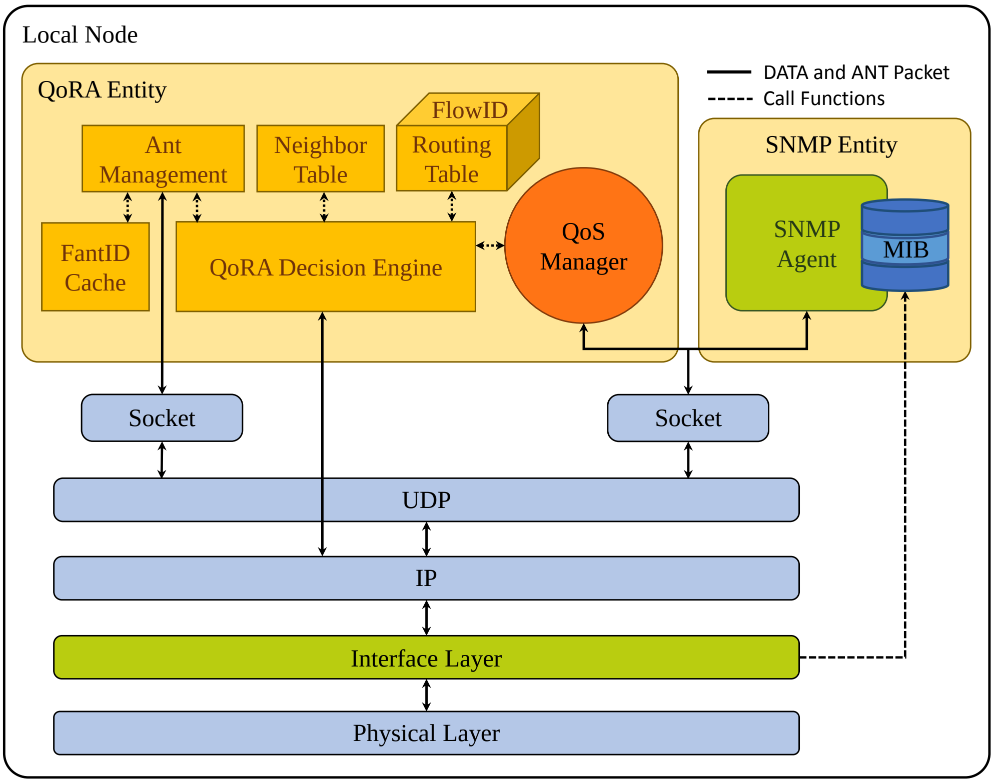


FIGURE 4.1: The QoRA Architecture

4.2.1 QoRA Entity Components

As mentioned earlier, each node chooses the best route to forward the packets as determined by QoRA Entity. The QoRA entity relies on the following sub-components to make this decision. These sub-components are Routing Table, Neighbor Table, Ant Management, FantID Cache, QoRA Decision Engine, and QoS Manager.

4.2.1.1 FlowID

Flow is defined as the successive transmission of packets from a source to a particular destination. The source node decides the appropriate label for each flow. This concept of flow label was originally developed to provide special services to real-time applications [131]. The IPv6 packet, by default, consist the flow label field. Here, the source node uses 20 bits flow label to label the packets of a flow. In IPv4, we have utilized the optional field to define the FlowLabel using 32 bits. So, with the help of the flow label the traffic is classified i.e. it facilitates the packets reordering. The flow label, when set to zero, indicates that the packets are not a part of any flow.

In QoRA, we use the 3-tuple parameters namely, flow label as discussed above, source address and destination address that determines the identity of a flow (FlowID) as shown in the below representation. Thus, FlowID is a unique representation consisting of flow label for a given source-destination pair.

$$\text{FlowID} < \text{SourceAddress}, \text{DestinationAddress}, \text{FlowLabel} >$$

4.2.1.2 QoRA Routing Table

The QoRA Routing Table is data structure used to forward the data packets. It stores multiple routes towards the known destinations according to the different flows.

Figure 4.2 demonstrates the routing table maintained at each node. An entry in the routing table represents the information about the route from node i to destination d over neighbor j associated with the given FlowID. The routing table entry contains the following fields: FlowID that specifies the flow, next hop address toward the destination, the QoS Parameter QP, the QoS threshold $\text{QP}_{\textbf{threshold}}$, the heuristic factor η_{ijd} related to the QoS parameters and a probability value P_{ijd} that reflects how likely it is that a data packet is forwarded using this neighbor.

The probability value or the objective function is calculated based on the pheromone and the heuristic factors according to the equation 4.1. Here, N_i is a set of neighbor nodes of i, and j is a neighbor node of i through which a route is available to destination d. The parameters α and β determines the weights of pheromone τ_{ij} and the heuristic factor η_{ijd}, respectively. The probabilities P_{ijd} of all neighbors with an available path to d sum up to 1.

$$P_{ijd} = \frac{[\tau_{ij}]^{\alpha}[\eta_{ijd}]^{\beta}}{\sum_{l \in N_i}[\tau_{il}]^{\alpha}[\eta_{ild}]^{\beta}} \tag{4.1}$$

The heuristic factor is calculated according to equation 4.2. Where β_B, β_S, and β_D denote the weights of each QoS parameter namely, bandwidth (B), expected success rate (S), and delay (D), respectively on the path from node i to destination d through neighbor node j. The equation to calculate the probabilities and the one to calculate the heuristic factor are derived from versions given in [132].

$$\eta_{ijd} = \frac{[\mathrm{B}_{ijd}]^{\beta_B} [\mathrm{S}_{ijd}]^{\beta_S}}{[\mathrm{D}_{ijd}]^{\beta_D}} \tag{4.2}$$

Thus, the entries in the routing table are solely based on the quality of the route. Also, every entry in the routing table comes with a lifetime. This means that after every usage of the given routing entry, its lifetime is updated for another constant period. In case if the routing entry is not utilized within the set time frame it is deleted from the routing table.

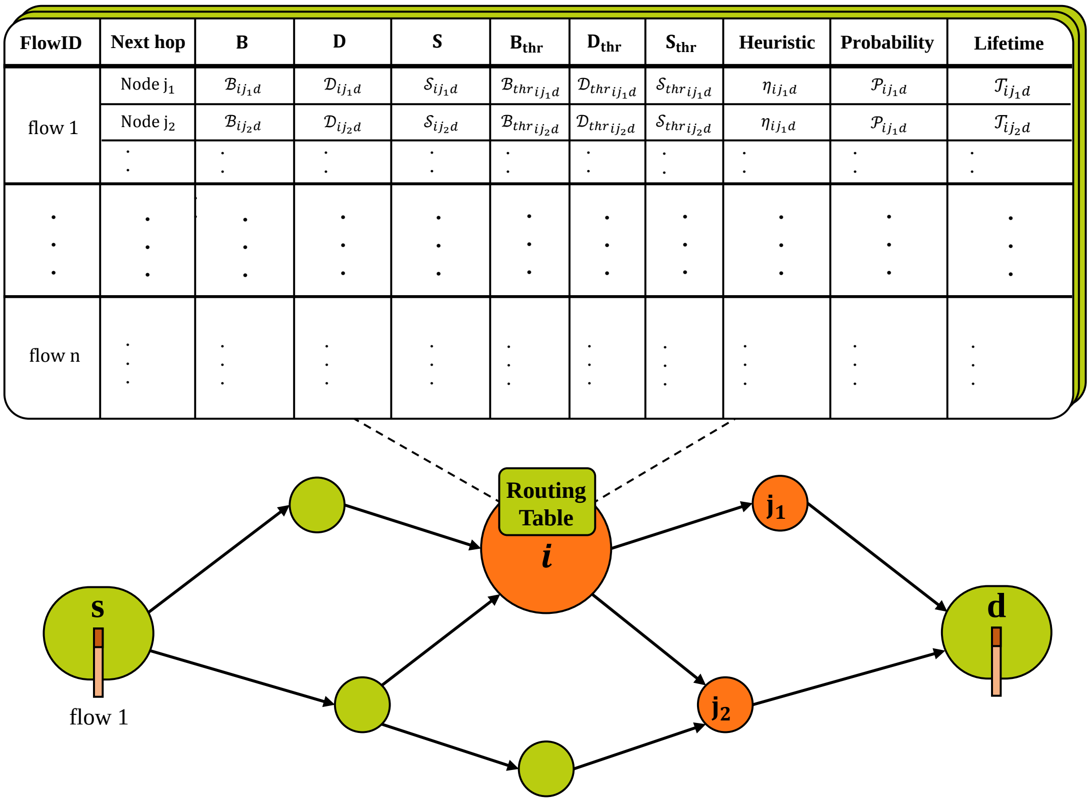


FIGURE 4.2: Example explaining the QoRA Routing Table

4.2.1.3 QoRA Neighbor Table

The QoRA Neighbor Table is a data structure as shown in figure 4.3, where each one-hop neighbor is registered along with the amount of pheromone indicating the experienced goodness of the link to that neighbor. Additionally, each entry for neighbor node contains the mapping between the neighbor IP address and its MAC address (hardware address).

In QoRA approach, the link failure notification is derived from the MAC layer. Based on this mapping, the neighbor node is identified and it is notified of the occurrence of the link failure. Also, each entry in the neighbor table comes with a lifetime. This value is updated after usage of the given entry in the neighbor table, its lifetime is updated for another constant period. In case if the entry is not utilized within the set time frame it is deleted from the neighbor table.

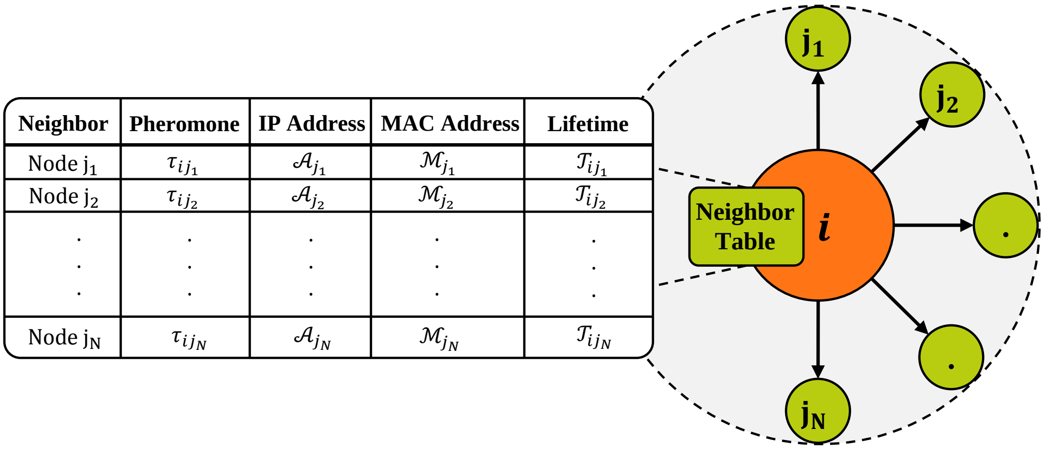


FIGURE 4.3: Example explaining the QoRA Neighbor Table

The equation 4.3 derived from [132], gives the current pheromone value τ_{ij} available on the link. High pheromone value on the link indicates that a large number of packets have selected the link from i to j recently, because each ant or data packet will increase the accumulated pheromone amount by $\triangle\tau_{ij}$. Whereas, the smaller pheromone value indicates that the link has bad quality, and the pheromone available on the path is decaying i.e. constantly evaporating by the factor ρ, thus reducing its attractive strength. The pheromone value is limited between the initial pheromone value 0.01 to maximum value 1.

$$\tau_{ij}^{new} = (1-\rho)\tau_{ij}^{old} + \triangle\tau_{ij} \tag{4.3}$$

The pheromone concentration τ_{ij} on a given link as discussed above depends on the number of packets traveling through that link and the pheromone diffusion factor ρ. In the QoRA approach, we have used the adaptive multi-rate mechanism that considers the link quality between the nodes and adapts the data rate accordingly.

Figure 4.4 represents this strategy in terms of the datarate adaptation. Here node S is the source node and the node D is the destination node whereas node B is the neighbor node of node S. In the subfigure 4.4.1, the distance between the nodes S and B is small and the link is estimated of high quality. So the data is sent on this link with higher data rates i.e. large number of packets are forwarded through this link with smaller transmission delays. As a result, the pheromone concentration on this link increases while it is continuously evaporating by the factor ρ. In the subfigures 4.4.2 to 4.4.4, the distance between the nodes is increasing and; therefore, the link quality is decreasing as explained in chapter 2, section 2.3. As a result, the data is transmitted with lower and lower data rates and increasing transmission delays, causing less pheromone accumulation on the link. In the subfigure 4.4.5, the two nodes move out of the communication range of each other, and there is no communication between them. All the pheromone available on the link is evaporated while no new pheromone deposition will take place on the link.

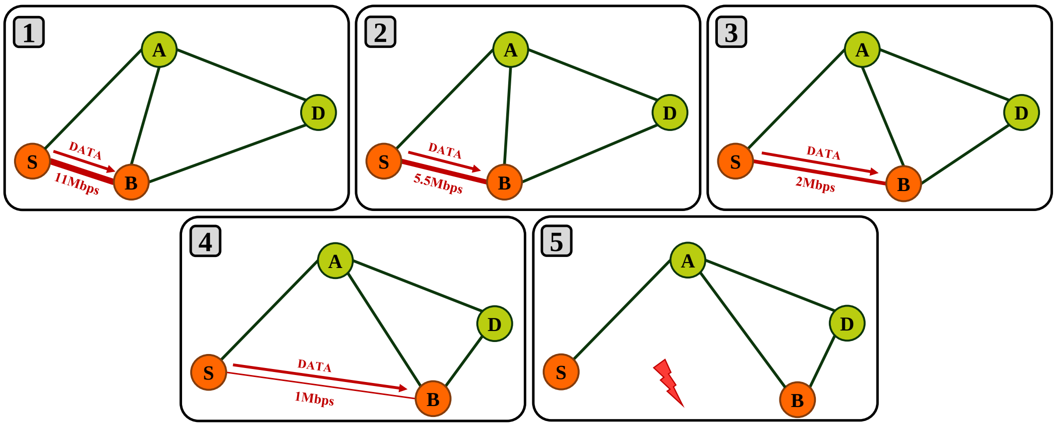


FIGURE 4.4: Adaptive of Datarate

Now, we define the diffusion factor ρ, such that it is adaptive to the transmission delay D_{trans}. The D_{trans} is given by the equation 4.9 and ρ is calculated as ten times of D_{trans}. It is evident from the Table 4.1 for the higher transmission delays the ρ will be higher indicating the poor link quality, while for lower transmission delays it will be smaller reflecting good link quality. So, the pheromone value is adaptive to the link quality and the maximum pheromone value is limited by the diffusion factor as explained in the table.

The graph in 4.5 shows the relation between the pheromone value and the number of packets. According to the equation 4.3, the new value of pheromone on the given link is determined by the value of ρ and the $\triangle\tau$ that is the pheromone deposition by each packet. In the beginning, the pheromone curve rapidly increases and stays at the maximum value of 1 (represents the scenario in subfigure 4.4.1) until the link quality

TABLE 4.1: The adaptive pheromone value based on transmission delay.

Datarate	D_{trans}	$\rho = D_{trans} \times 10$	Max Pheromone
11 Mbps	0.0017	0.017	1
5.5 Mbps	0.0033	0.033	0.5
2 Mbps	0.0092	0.092	0.2
1 Mbps	0.0184	0.184	0.1

$D_{trans} \leq \frac{\text{ifMtu (Octet)} \times 8}{\text{ifSpeed}}$ where ifSpeed = Datarate and ifMtu = 2296 equation 4.9.

starts deteriorating. At this instance, the pheromone value decreases and stays constant at value 0.5 (represents the scenario in figure 4.4.2) until a further dip (represents the scenario in subfigures 4.4.3 and 4.4.4) in link quality is observed, where pheromone value decreases to 0.2 and 0.1 respectively. Thus, the decrease in the amount of pheromone on the given link indicates the decrease in the probability for this link and an increase in the probability for other links. Once the link quality is estimated to be unsuitable for the communication, the pheromone value is dropped to 0 (subfigure 4.4.5). So, the pheromone value is adaptive to the link quality.

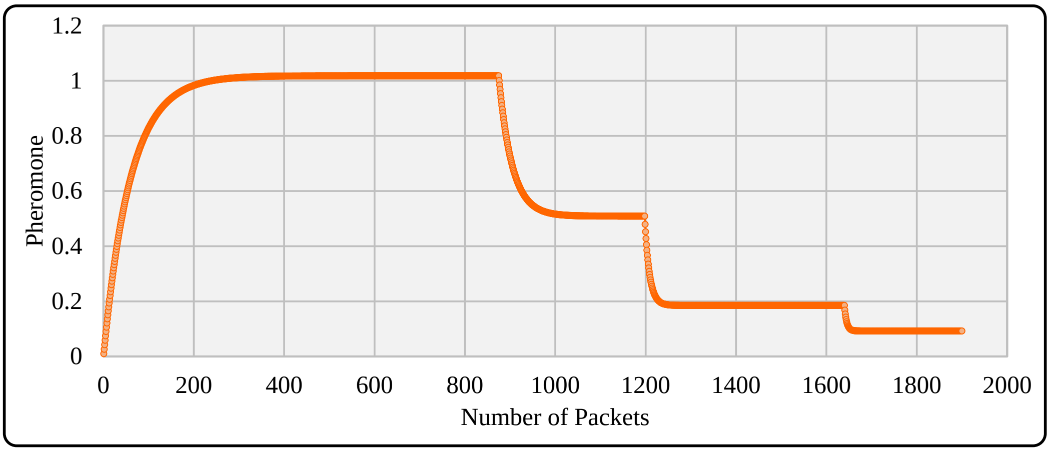


FIGURE 4.5: Adaptive pheromone concentration

4.2.1.4 Ant Management

The Ant Management is responsible for generating three types of ants namely the forward ant (FANT), the backward ant (BANT) and the error ant (EANT). The ant management serializes and deserializes these ants when transmitting or receiving them. These ants are autonomous and asynchronous agents, which indirectly communicate with each other using *stigmergy*. These ants contain specific information to provide QoS-aware routing.

- **Forward ant (FANT):**

The FANT packet is used to search the path from source to destination using the flooding technique. The FANT as shown in figure 4.6 has three packet fields namely QP, FlowID and FantStack.

- The first packet field is QP that represents the set of QoS Parameters, namely, the end-to-end delay (D) and expected success rate (S) aggregated over all visited nodes while traversing from source to destination. The desired QoS values for the application in terms of minimum bandwidth (B_{min}), maximum delay (D_{max}), and minimum expected success rate (S_{min}) are stored in the corresponding fields with the same name. The Type field determines the type of ant and the AntSeq field represents a unique sequence number for an ant in the flow.
- The second packet field is FlowID that contains the IP address of the source/destination pair of the communication session and the FlowLabel, the counter values that identifies the type of data stream or flow defined by the source node.
- The third packet field is the FantStack that contains the IP addresses of all the nodes visited by the FANT packet.

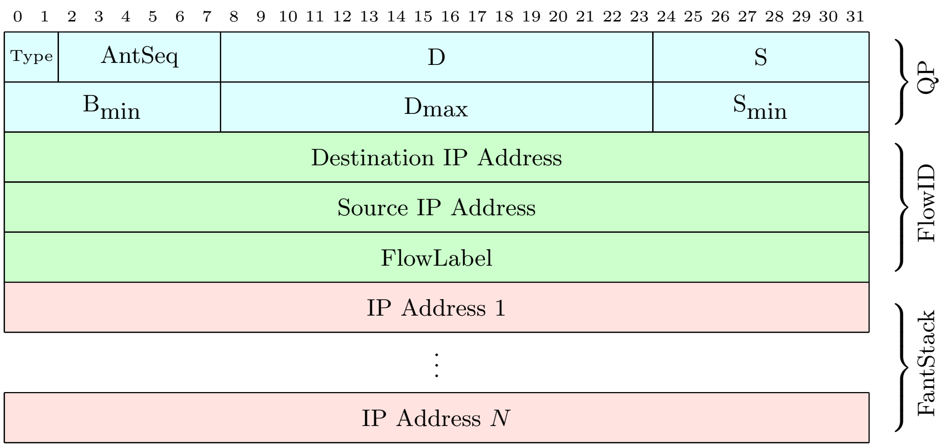


FIGURE 4.6: Forward ant packet format

- **Backward ant (BANT):**

The BANT packet is used to validate all the available routes from source to destination using the unicast technique. The BANT packet is shown in figure 4.7, also has three packet fields namely, QP, FlowID and BantStack.

- The first packet field is QoS parameter or QP that represents the minimum available bandwidth (B), the end-to-end delay (D), and expected success rate (S) aggregated over all the visited nodes. The fields QD, QB, and QS represent the residual QoS parameter in terms of residual delay, residual bandwidth, and residual expected success rate. The Type field determines the type of ant.

- The FlowID packet field is the same as discussed in FANT packet. BANT gets the value of this field from FANT to identify the flow.
- The BantStack packet field contains the IP addresses of all the intermediate nodes that are visited by the FANT during route discovery from source to destination. BANT utilizes these IP addresses stored in BantStack to validate and establish the route from destination to the source node.

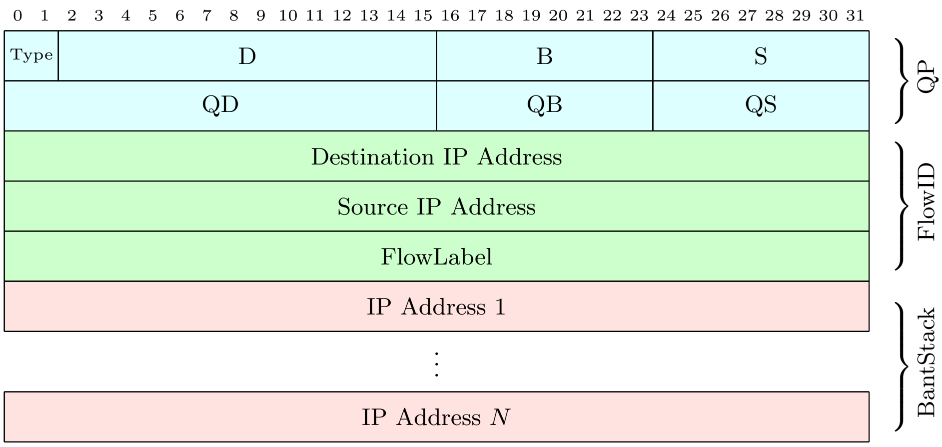


FIGURE 4.7: Backward ant packet format

- **Error ant (EANT):**

The EANT packet is shown in figure 4.8, contains only one packet field called EantStack. This field contains the FlowID of all the flows that are affected. The EANT packet is used to notify about the link failure or QoS violation (congestion problem) occurring on the selected route.

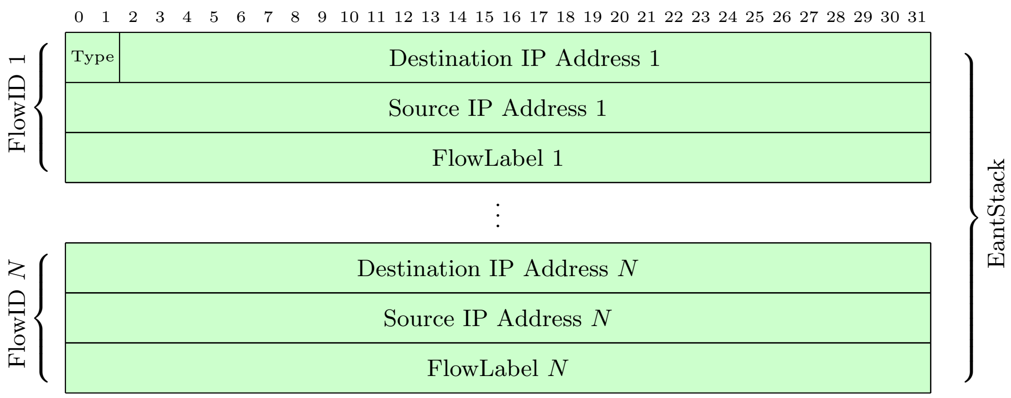


FIGURE 4.8: Error ant packet format

4.2.1.5 FantID Cache

The flooding mechanism causes the FANTs to multiply quickly over the network. So, a node might receive a number of FANTs from the same generation and same previous node. In this situation, a node compares the goodness value (ψ_{sji}) of the path traveled by the FANT to that of the previously received FANT. The node will forward the FANT only if its goodness value is equal or greater than the goodness value of the best FANT of the same generation that it has already received from the same previous node. Using this mechanism the overhead is put to limit by dropping FANTs that has followed bad paths. Additionally, based on this policy reliability increase by creating a mesh network, non-disjoint multipath. The goodness value (ψ_{sji}) is calculated using equation 4.4

$$\psi_{sji} = \frac{[\mathrm{S}_{sji}]^{\beta_S}}{[\mathrm{D}_{sji}]^{\beta_D}} \tag{4.4}$$

Figure 4.9 represents the FantID cache table with the different flow entries stored in the FlowID. The next parameter is the AntSeq i.e. unique for every ant generation and is given by the source node. FantID is a unique identifier that recognizes the particular FANT. It is generated using AntSeq in conjunction with the FlowID. The goodness value (ψ_{sji}) that identifies the quality of the incoming FANT. As discussed earlier all the entries come with a Lifetime and the information is deleted often the Lifetime is elapsed.

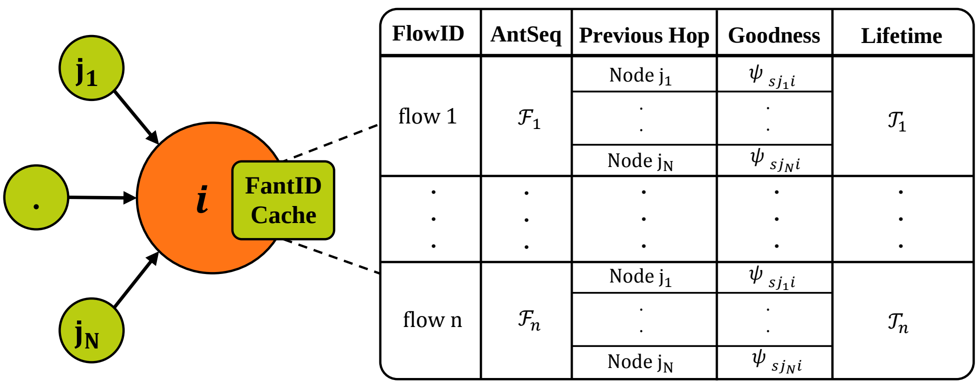


FIGURE 4.9: Example explaining the FantID Cache

4.2.1.6 QoRA Decision Engine

The QoRA Decision Engine is the most important block in the QoRA entity. It handles the execution of the following tasks. It triggers the generation of different ants through the Ant Management. It is responsible for updating the Neighbor Table and the Routing Table based on the QoS parameters, the information gathered by the ants, and the corresponding local QoS values. These local values are obtained by calling the QoS Manager to communicate with the local SNMP Agent. It takes a decision about forwarding the received data packets to upper layer or to other nodes by reading the information on the packet to determine its FlowID. Using the information in the routing table, it directs the packet to the next node according to the probability method and for local delivery it directs the packet to the upper layer.

4.2.1.7 QoS Manager

The QoS Manager can be described as a software module that works in line with SNMP Manager and generates QoS information. It is described in detail in section 4.2.2.1.

4.2.2 SNMP Entity Components

Simple Network Management Protocol (SNMP) is a member of the TCP/IP protocol structure and an application layer protocol that was developed to simplify the tasks of network management and monitoring. The Internet Engineering Task Force (IETF) has published three versions of SNMP as defined in the respective Requests for Comments (RFCs). The first version of SNMP was introduced in 1988 and is defined in RFC 1157 [133] and was called SNMPv1. The second version is called SNMPv2, and is defined in RFC 1901 [134], and RFC 1908 [135]. The latest version is called SNMPv3, and it is accepted as full standard by IETF and is defined together by a number of RFCs, namely in the series, RFC 3410 to RFC 3417, RFC 3584, RFC 3826, and RFC 5343 [136–146].

The primary elements of the SNMP management system are the monitor application, the manager, the agent, and the Management Information Base (MIB), as shown in figure 4.10. These entities together perform the task of network monitoring, configuration, controlling, accounting, etc. by defining a set of Managed Objects on the managed device.

4.2.2.1 SNMP Manger and Agent

The **SNMP Manager** is a software module that handles the task of network management and generates commands for the SNMP operations namely, Get, Set, and notification receive (trap). This manager communicates with SNMP agent to retrieve or

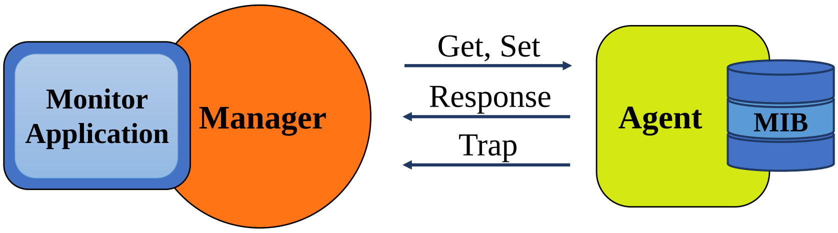


FIGURE 4.10: The central parts of SNMP [147]

modify management information. On the basis of this information, the Manager take the necessary preconfigured actions.

For the QoRA approach, we have designed a lightweight implementation of the SNMP Manager that is integrated into the QoRA entity and which we would like to call QoS Manager. In addition to the tasks performed by SNMP Manager, the QoS Manager calculates the values of QoS parameters locally on the basis of the gathered information and takes optimal decisions.

The **SNMP Agent** is a software module that runs on every networked device. Enabling the SNMP agent allows it to maintain the management information database from the device locally. A management information exchange can be polling, where the manager asks the agent for a particular value, or it can be event reporting (trap) by the agent, where the agent tells the manager that something important has happened. So, the trap originates from the agent and are sent to Manager, where it is handled appropriately.

The management information is available through Managed Objects, which are stored in a standardized tree-based MIB using a template called Structure of Management Information (SMI). The device/agent is responsible for maintaining MIB. The first version of SMI is called SMIv1 and is defined in RFC 1155 [148]. The second and latest version is called SMIv2, and it is described in RFC 2578 [149]. The SMIv2 defines a set of rules for organizing and naming of Managed Objects. However, SMI relies on ITU's Abstract Syntax Notation One (ASN.1) to define the structure of these objects. The ASN.1 language defines a set of rules and structures for representing, transmitting, encoding, and decoding SNMP messages. Additionally, it is used to specify the format of MIB models such as model definition, object definition, and notification definition. The SNMP message contains an SNMP Protocol Data Unit (PDU). The SNMP PDU is used for exchange information between managers and agents. The SNMP PDU is defined using ASN.1 for each of the following SNMP operations [150]:

- *GetRequest*: Generated by the SNMP manager to retrieve one or more requested MIB Management Objects defined in the PDU.
- *GetNextRequest*: Generated by the SNMP manager to retrieve the next MIB Management Objects defined in the PDU. This PDU is essentially used by the SNMP manager to walk through the SNMP agent MIB.
- *GetBulkRequest*: Generated by the SNMP manager to retrieve a large amount of MIB Management Objects, particularly from large tables. The GetBulkRequest operation performs a continuous GetNextRequest operation based on the max-repetition number.
- *SetRequest*: Generate by the SNMP manager to set one or more MIB Management Objects defined in the PDU with the value specified in the PDU.
- *GetResponse*: Generated by the SNMP agent in response to a GetRequest, GetNextRequest, GetBulkRequest, or SetRequest PDU.
- *Trap*: Generated by the SNMP agent to notify the SNMP manager about an important event that happened in the agent.

4.2.2.2 Management Information Base

The first version of Management Information Base (MIB) is defined in RFC 1156 and is called as MIB-I [151]. The latest version is called as MIB-II, and it is defined in RFC 1213 [152]. MIB-II contains the most significant objects used to define the TCP/IP stack, which is divided into groups as shown in figure 4.11.

The MIB-II interface group with Object IDentifier (OID: 1.3.6.1.2.1.2) includes managed objects about the physical interface, such as a configuration parameter and an interface statistics. The physical interface is responsible for the packet transfer over the network. This group was extended to include more Management Objects (e.g. use of Counter 64) in RFC 2863 [154]. The interface MIB group includes the object ifNumber that contains the number of the available network interfaces. Additionally, it contains these four tables:

- The first table is *ifTable*, which has one entry for each interface. It contains a collection of objects used to define the physical interface such as name, MTU, interface speeds, and the counter for bytes and packets received, sent, and dropped.

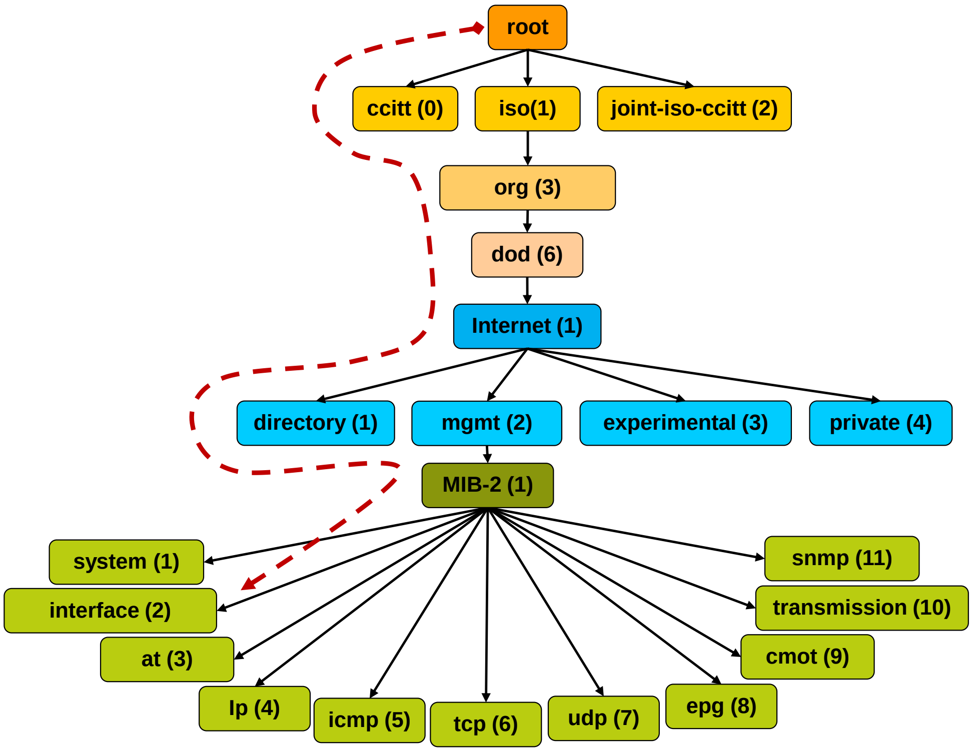


FIGURE 4.11: Object identifiers in the Management Information Base-II [153]

- The second table is *ifXTable*, which contains the additional objects that have been added or replaced in the ifTable.
- The third table is *ifStackTable*, which includes the objects used to define the relations between the various sub-layers of network interfaces.
- Finally, the fourth table is *ifRcvAddressTable*, which includes objects that are used to contain the MAC address (unicast, broadcast, or multicast) for which an interface will receive packets/frames.

The management information is available through Managed Objects, which are stored in a standardized tree-based MIB [155]. This tree-structure serves to uniquely address Managed Objects.

To give some examples of the standardized Managed Objects useful to assess the quality of different outgoing links, Table 4.2 lists the relevant objects for our approach. The short descriptions are taken from the relevant RFC 2863 [154].

Table 4.2: Relevant SNMP Managed Objects

Object	*Short Description*
ifSpeed	An estimate of the interface's current bandwidth in bits per second.
ifMtu	The size of the largest packet which can be sent/received on the interface.
ifOutQLen	The length of the output packet queue (in packets).*
ifInDiscards	The number of inbound packets which were chosen to be discarded.
ifInErrors	For packet-oriented interfaces, the number of inbound packets that contained errors.
ifInUnknownProtos	For packet-oriented interfaces, the number of packets received via the interface which were discarded because of an unknown or unsupported protocol.
ifOutDiscards	The number of outbound packets which were chosen to be discarded.
ifOutErrors	For packet-oriented interfaces, the number of outbound packets that could not be transmitted because of errors.
ifHCInUcastPkts	The number of Unicast packets, delivered by this sub-layer to a higher (sub-)layer.
ifHCInMulticastPkts	The number of packets, delivered by this sub-layer to a higher (sub-)layer, which were addressed to a multicast address at this sub-layer.
ifHCInBroadcastPkts	The number of packets, delivered by this sub-layer to a higher (sub-)layer, which were addressed to a broadcast address at this sub-layer.
ifHCOutUcastPkts	The total number of (unicast) packets that higher-level protocols requested be transmitted.
ifHCOutMulticastPkts	The total number of packets that higher-level protocols requested be transmitted, and which were addressed to a multicast address at this sub-layer.
ifHCOutBroadcastPkts	The total number of packets that higher-level protocols requested be transmitted, and which were addressed to a broadcast address at this sub-layer.

* This object is deprecated. Nevertheless, most equipment manufacturers have this information in their MIB.

4.3 Cooperation QoRA and SNMP

As discussed in the previous section, QoS Manager is inspired from the SNMP Manager that determines the quality of the links locally. For calculating the QoS values of the outgoing links it relies on the measurements of the various link parameters performed by the SNMP agent. These values are stored in the Managed Objects. The following QoS parameters are considered to determine the quality of the outgoing link.

4.3.1 Bandwidth

Bandwidth or transmission speed is one of the most important QoS parameters and is given in [b/s]. The Bandwidth is a measure of the transmission speed at which the data can be sent through the network. In multi-rate ad hoc networks, the maximum bandwidth for a given path is limited by the minimum transmission speed of the link on that path. Thus, each node must determine the transmission speed of its outgoing links. This value is stored in the Managed Object "ifSpeed" (cf. table 4.2) and is obtained from the SNMP agent. The management object ifSpeed is a read-only gauge that estimates the current capacity at the interface in [b/s], which is often expressed as the maximum bandwidth that the node can provide. The transmission speed is automatically selected based on the signal strength received as explain in the Receiver-Based Auto-Rate (RBAR) [23]. Thus, the minimum available bandwidth (B_{ijd} for the path $p(i, j, d)$) is determined according to equation 4.5.

$$B_{ijd} = min\{\text{ifSpeed}(n), \quad \forall n \in p(i, j, d)\} \tag{4.5}$$

In the figure 4.12 the bandwidth estimated on the link between node i and node j is 11 Mbps, between node j and node k is 2 Mbps while it is 5.5 Mbps on the link between node k and node d. The maximum available bandwidth for the path between node i to node d through the neighbor node j is the minimum bandwidth available (B_{ijd}) on the path i.e. 2 Mbps, as bandwidth is a concave metric.

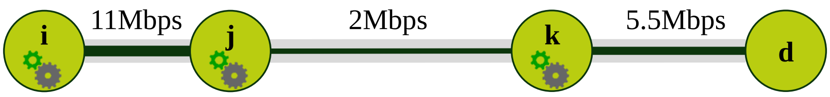


FIGURE 4.12: Example explaining the available Bandwidth of the path $p(i, j, d)$

The local available bandwidth (B_{local}) is calculated using the following equations that employs the periodic passive monitoring of the Channel Idle Time Ratio (CITR). The CITR is the ratio of the time for which the channel is observed to be idle (T_{idle}) to the given period of time , known as monitoring window (ΔT) as explained in the [79]. Thus, to determine whether the node n in the path $p(i, j, d)$ can satisfy bandwidth requirement B_{min} $[b/s]$, they must fulfill equation 4.6.

$$B_{\text{local}}(n) = \text{CITR}(n) \times \text{ifSpeed}(n) \geq B_{\text{min}} \tag{4.6}$$

Where, the CITR is calculated periodically based on the previous values using the equation 4.7 and θ is a weight moving factor for CITR.

$$\mathrm{CITR}^{new}(n) = (1-\theta)\mathrm{CITR}^{old}(n) + \theta\left(\frac{\mathrm{T}_{\mathrm{idle}}(n)}{\Delta T}\right) \tag{4.7}$$

4.3.2 Delay

The next QoS parameter is delay and it indicates the time required to transfer the data packets from the sender node to the receiver node. The average delay observed on the node is determined by the following components: the propagation delay ($\mathrm{D}_{\mathrm{prop}}$), which indicates the time taken by the physical signal to travel from the sender to the receiver, the transmission delay ($\mathrm{D}_{\mathrm{trans}}$), which is the time required by the sender to transmit the first and the last bit of the packet, and the queuing delay ($\mathrm{D}_{\mathrm{queue}}$) is observed at the outgoing interface queue of the sending node. So the total delay ($\mathrm{D}_{\mathrm{total}}$) in a node n is given by equation 4.8.

$$\mathrm{D}_{\mathrm{total}}(n) = \mathrm{D}_{\mathrm{prop}} + \mathrm{D}_{\mathrm{trans}} + \mathrm{D}_{\mathrm{queue}} \tag{4.8}$$

The propagation delay $\mathrm{D}_{\mathrm{prop}}$ depends on the physical distance between the sender and the receiver. The distance between the two nodes varies significantly in MANETs due to high mobility. However, assuming the radio communication with the signal speed close to the speed of light (299,792,458 m/s), the propagation delay between two MANET nodes that are less than 1000 m apart from each other, is observed to be approximately 3.33×10^{-6} s. This propagation delay is least significant and does not show any noticeable effect in comparison to the other delay factors. Thus, it is neglected and set to 0.

The transmission delay $\mathrm{D}_{\mathrm{trans}}$ depends on the size of the packet and the bandwidth of the link. For estimation, the value of the maximum size of a packet is retrieved from the SNMP agent, which is stored in the Managed Object "ifMtu". Since this object gives the size in octets, it needs to be multiplied by 8 to get the packet size in bits. The bandwidth again can be found in the Managed Object "ifSpeed". Therefore, $\mathrm{D}_{\mathrm{trans}}$ is estimated by equation 4.9.

$$\mathrm{D}_{\mathrm{trans}} \leq \frac{\mathrm{ifMtu} \times 8}{\mathrm{ifSpeed}} \tag{4.9}$$

The queuing delay $\mathrm{D}_{\mathrm{queue}}$ depends on the number of packets waiting in a queue at the node to be transmission on an outgoing link. This number is available with the SNMP agent in the Managed Object “ifOutQLen”. The queuing delay can be assumed as given in equation 4.10.

$$\mathrm{D_{queue}} \leq \mathrm{ifOutQLen} \times \mathrm{D_{trans}} \tag{4.10}$$

Then the total delay $\mathrm{D_{total}}$ at node n calculate using equation 4.11.

$$\mathrm{D_{total}}(n) \leq \frac{\mathrm{ifMtu\ (Octet)} \times 8}{\mathrm{ifSpeed}} \times (1 + \mathrm{ifOutQLen}) \tag{4.11}$$

The end-to-end delay (D_{ijd}) for the path $p(i, j, d)$ is calculated according to equation 4.12 and must satisfy the delay constraints ($\mathrm{D_{max}}$). If D_{ijd} greater than the desired maximum delay $\mathrm{D_{max}}$, then FANT will be dropped. Otherwise FANT will be forwarded with the updated D field.

$$\mathrm{D}_{ijd} = \sum_{n \in p(i,j,d)} \mathrm{D_{total}}(n) \leq \mathrm{D_{max}} \tag{4.12}$$

In the figure 4.13 the delay observed at node i is 2 ms, at node j is 4 ms and at node k is 6 ms. The end-to-end delay for the path from node i to node d through the neighbor node j is given as the summation of delays observed at each node in the path, since delay is an additive metric. In this scenario the delay on the given path is estimated to be $\mathrm{D}_{ijd} = 12$ ms.

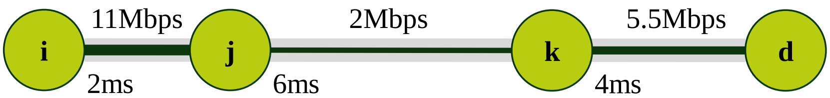


FIGURE 4.13: Example explaining the end-to-end delay of the path $p(i, j, d)$

4.3.3 Packet Loss

Packet loss can be assessed by the packet loss ratio, which is the number of lost packets divided by the total number of packets. The high packet loss value indicates towards the problem of congestion, bad channel quality due to fading and interference on wireless links, the software problem i.e. the node does not support the protocol and/or the hardware problem i.e. the net device does not work well. These values can again be retrieved from the SMNP agent (cf. table 4.2).

Each node can compute the correctly received incoming packets (Pkts_{Rx}) using the Managed Objects “ifHCInUcastPkts”, “ifHCInMulticastPkts”and “ifHCInBroadcastPkts” according to equation 4.13.

$$\begin{aligned}\text{Pkts}_{Rx} = \text{ifHCInUcastPkts} + \text{ifHCInMulticastPkts} \\ + \text{ifHCInBroadcastPkts}\end{aligned} \tag{4.13}$$

The number of successfully sent packets (Pkts_{Tx}) can be computed accordingly based on the Managed Objects “ifHCOutUcastPkts”, “ifHCOutMulticastPkts” and “ifHCOutBroadcastPkts” according to equation 4.14.

$$\begin{aligned}\text{Pkts}_{Tx} = \text{ifHCOutUcastPkts} + \text{ifHCOutMulticastPkts} \\ + \text{ifHCOutBroadcastPkts}\end{aligned} \tag{4.14}$$

For the number of dropped packets, there are also some Managed Objects, again divided into incoming and outgoing. Equation 4.15 shows how to compute the number of incoming packets that are dropped (DropPkts_{in}).

$$\text{DropPkts}_{in} = \text{ifInDiscards} + \text{ifInErrors} + \text{ifInUnkownProtos} \tag{4.15}$$

The number of dropped outgoing packets (DropPkts_{out}) is given by equation 4.16.

$$\text{DropPkts}_{out} = \text{ifOutDiscards} + \text{ifOutErrors} \tag{4.16}$$

With these computed values, each node n is able to compute the packet loss ratio (PLR) shown in equation 4.17. Additionally, all the above values are update each periods of times, known as monitoring window (ΔT), by calculating the difference between successive Managed Object (MO) values ($\text{MO} = \text{MO}_{new} - \text{MO}_{old}$).

$$\text{PLR (n)} = \frac{\text{DropPkts}_{in} + \text{DropPkts}_{out}}{\text{Pkts}_{Rx} + \text{Pkts}_{Tx} + \text{DropPkts}_{in}} \tag{4.17}$$

The success rate (S) is calculated periodically using equation 4.18. Where, φ is a weight moving factor for S.

$$\mathrm{S}^{new}(n) = (1-\varphi)S^{old}(n) + \varphi(1 - \mathrm{PLR}(n)) \tag{4.18}$$

The FANT carries the desired success rate ($\mathrm{S_{min}}$) for packet transmission and the expected success rate (S_{ijd}) for the visited path. Each node refreshes the S_{ijd} according to equation 4.19. If S_{ijd} falls below the desired success rate $\mathrm{S_{min}}$, then FANT will be dropped. Otherwise FANT will be forwarded with the updated S field.

$$\mathrm{S}_{ijd} = \prod_{n \in p(i,j,d)} \mathrm{S}(n) \geq \mathrm{S_{min}} \tag{4.19}$$

In the figure 4.14 the packet loss expected at node i is 1%, node j is 3% and node k is 2%. This means the packet success rate at node i, j and k is 99%, 97% and 98% respectively. So, the expected success rate for the path from node i to node d through the intermediate node j is given as the product of the success rates observed at each node in the path, since success rate is a multiplicative metric. Here, the expected success rate for the given path is 94%.

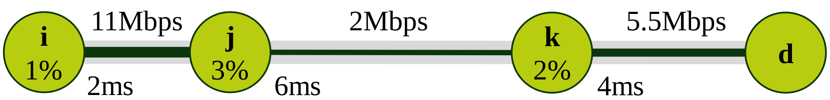


FIGURE 4.14: Example explaining the packet loss for the path $p(i, j, d)$

4.3.4 Estimate QoS Threshold

Once the data flow begins from the source to the destination, the path needs a constant monitoring in order to keep the track of its quality. If the data is forwarded on the link that no longer provides the desired minimum quality then it will cause packet loss and congestion. A specific threshold is required to indicate that the node does not satisfy the QoS requirements or it has a congestion problem. Each FANT arrives at the destination d containing the end-to-end QoS parameter, such as D_{ijd}, S_{ijd} and QoS desired, such as $\mathrm{B_{min}}$, $\mathrm{D_{max}}$, and $\mathrm{S_{min}}$. On the basis of these values, the residual value of the QoS parameters, such as residual bandwidth (QB), residual delay (QD), and residual expected success rate (QS) can be calculated based on the subsequent equations 4.20, 4.21 and 4.22. The calculated residual values are then set in BANTs and it is transmitted on the path.

$$\mathrm{QB} = \mathrm{B}_{\mathrm{min}} \tag{4.20}$$

$$\mathrm{QD} = \frac{\mathrm{Dmax} - \mathrm{D}_{ijd}}{hop + 1} \tag{4.21}$$

$$\mathrm{QS} = \frac{\mathrm{S}_{ijd} - \mathrm{S}_{\mathrm{min}}}{hop + 1} \tag{4.22}$$

In the above equations *hop* is the number of intermediate nodes from the source node to the destination. After receiving BANTs at the intermediate node, the node can compute the QoS threshold for each flow f and store these values in the routing table. QoS threshold for bandwidth ($\mathrm{B}_{\mathrm{thr}}$), delay ($\mathrm{D}_{\mathrm{thr}}$), and expected successful rate ($\mathrm{S}_{\mathrm{thr}}$) at node n are given by following equations 4.23, 4.24 and 4.25 respectively.

$$\mathrm{B}_{\mathrm{thr}}(n, f) = \mathrm{QB} \tag{4.23}$$

$$\mathrm{D}_{\mathrm{thr}}(n, f) = \mathrm{D}_{\mathrm{total}}(n) + \mathrm{QD} \tag{4.24}$$

$$\mathrm{S}_{\mathrm{thr}}(n, f) = \mathrm{S}(n) - \mathrm{QS} \tag{4.25}$$

4.4 QoRA Approach: Functioning

This section describes the functioning of the proposed routing protocol QoRA. The QoRA protocol consists of five phases namely the route forward phase, the route backward phase, packet forwarding phase, the monitoring phase and, the link failure phase. All these phases are discussed in details in the following sections.

4.4.1 Route Forward Phase

In the route forward phase, the source node of the session queries its routing table for the route availability to the requested destination. If the route is not found in the routing table, the source node triggers the forward phase. In the meantime, the source node saves the application data packets in the output queue until the desired QoS route is discovered with the destination.

In this phase, the source node broadcasts FANT in the network to find the routes to the destination. The source node set the fields, namely, type of ant, generation number, AntSeq, FlowID, and desired QoS parameters such as B_{min}, D_{max}, S_{min} in the FANT packet. The intermediate node after receiving FANT creates or updates an entry and link Lifetime of the node from which it has received the FANT in its neighbor table. The node then calls QoS Manager that calculates QoS parameters expected delay and success rate using appropriate Managed Objects.

The broadcasting of packets in the IEEE 802.11 MAC is done at the basic data rate for example 802.11b equal 1 Mbps so that the broadcast packet can cover larger area in the WLAN environment. However, before forwarding the FANT packet the QoRA approach makes the following checks:

- First, it looks into the FantStack to confirm that this node has not received FANT before in order to avoid the loops. In the figure 4.15, the loop exist for example between nodes l, m and j.
- Second, if the node cannot satisfy the required QoS in terms of expected delay and success rate, it discards the FANT. This process is repeated until the sent FANT reaches the desired destination.
- Finally, the FANT is killed once it arrives at the destination.

The multiple paths are established between the source and the destination by forwarding numerous copies of the same ant over the underlying network. But, the flooding mechanism causes the FANTs to multiply quickly over the network. So, a node might receive a number of FANTs from the same generation and same previous node. In the figure for the paths (s, l, j, m) and (s, i, j, m), node m receives the FANT from the same previous node i.e node j and from the same generation (FantID). In this situation, a node compares the goodness value (ψ_{sjm}) of the path and forwards the ant packet as explained in the section 4.2.1.5. After forwarding the FANT, the FantID cache is updated. The source node after broadcasting the FANTs waits for FantExpirationTime for the establishment of the desired path. In case the path is not discovered within this time frame, the source node makes another attempt but now with twice the value of the previous FantExpirationTime. One more failure will cause the node to inform the application layer about the impossibility to establish the desired path. The pseudocode in Algorithm 1 describes the route forward phase.

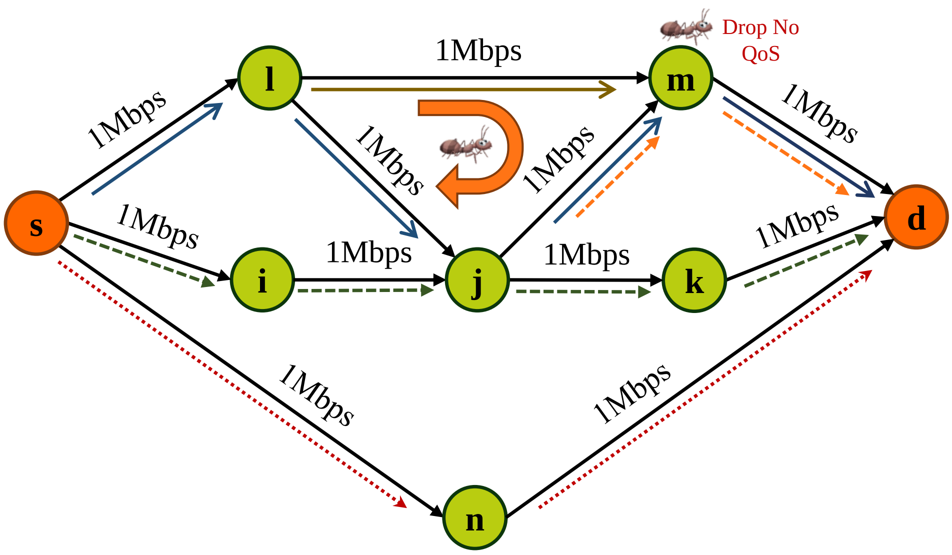


FIGURE 4.15: Example explaining the Route Forward Phase

Algorithm 1 Forward Phase

procedure ROUTE FORWARD PHASE

Node = Source node

    if No route available in the Routing table **then**

        Queuing data packet

        Broadcast FANT message ▷ Broadcast at the base data rate of 1 Mbps

    end if

Node = Intermediate node

    Receive FANT

    Add or update neighbor table entry

    Call SNMP agent using QoS Manager

    Calculate goodness ψ_{sji} ▷ using (4.4)

    if ψ_{sji} ‖ FAntStack Loop fails **then**

        Drop FANT packet

        Return

    end if

    if $(D_{sji} > D_{max}) \parallel (S_{sji} < S_{min})$ **then** ▷ using (4.12, 4.19)

        Drop FANT packet

        Return

    end if

    if Destination node condition fails **then**

        Update FantID Cache and FantStack

        Broadcast FANT

    else

        Kill FANT

        Start Route Backward Phase

    end if

end procedure

4.4.2 Route Backward Phase

The route backward phase start when the FANTs arrive at the destination. The FANT is then automatically converted into a BANT. Then, the destination node copies the FlowID and FantSack into the corresponding fields in the BANT packet. The destination node also computes the QoS residual value for the bandwidth (QB), delay (QD), and success rate (QS) as explained in section 4.3.4 and sets in BANT. The destination node then unicasts the BANT to the source node using the IP addresses stored in BantStack at the data rate determined by RBAR as shown in the figure 4.16.

Throughout the route, the BANT determines the bandwidth constraints and if the local available bandwidth (B_{local}) on the link is less than B_{min} then the node will drop BANT. In the figure the BANT is dropped at node n, since the bandwidth is not sufficient, when the B_{min} = 2 Mbps. Also, the BANT collects information about the quality of each link in the path and refreshes the routing tables using this information, calculates the QoS threshold B_{thr}, D_{thr}, and S_{thr} as discussed in section 4.3.4 and the probabilities using equation 4.1. The probabilities are used to determine the next hop for the data packets forwarding in the QoS aware multipaths to the destination. The probabilistic approach for the data forwarding on the QoS-aware multipath provides the features of load balancing and congestion control. The entire process is repeated if the source node loses valid paths for the destination while data still need to be sent.

Again, the pheromone on the path is updated using the equation 4.3. Also, the Lifetime field for the entries in routing table and neighbor table is updated and set to the current time plus the ActiveTimeOut. The pseudocode in Algorithm 2 describes the route backward phase.

Figure 4.17 depicts a simple scenario that explain the calculation of residual QoS values on the path. Here node i would like to communicate with node d and node j and node k are the intermediate nodes. We assume the following values are desired by an application in order to provide QoS, namely, B_{min} = 2 Mbps, D_{max} = 150 ms, S_{min} = 90%. In the forward phase the FANT is broadcasted at the base data rate i.e. the base data rate for IEEE 802.11b equal 1 Mbps. The delay observed at node i, node j and node k is 22 ms, 12 ms and 22 ms respectively. So, the end-to-end delay (D_{ijd}) observed on the path is the summation of the individual delays and in our case it is 56 ms according to equation 4.12. The success rate observed at node i, node j and node k is 0.99, 0.97 and 0.98 respectively. So the success rate (S_{ijd}) on the path is the multiplication of individual success rates and in our case it is 0.94 according to equation 4.19.

In the backward phase the BANT is unicasted from the destination d to the node i. At node d the residual QoS values are calculated as follows. The residual bandwidth (QB)

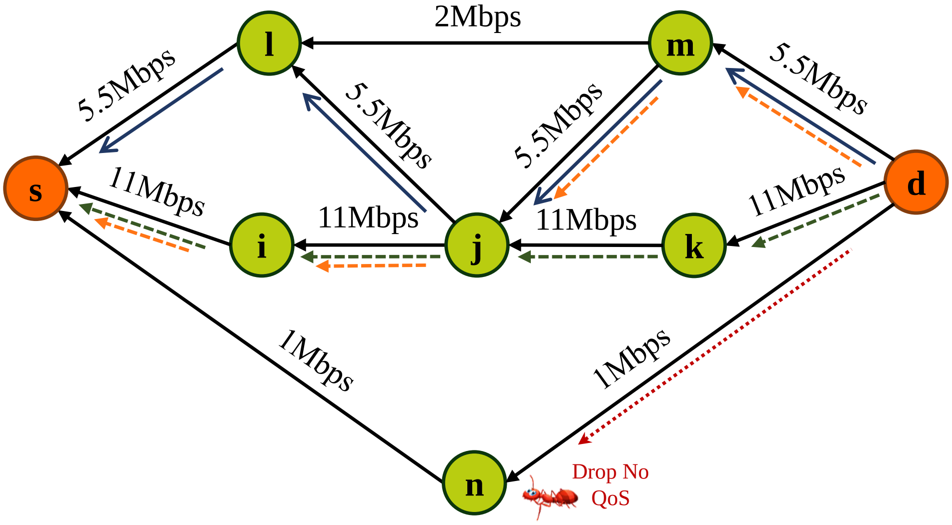


Figure 4.16: Example explaining the Route Backward Phase

Algorithm 2 Route Backward Phase

procedure Route Backward Phase
  Node = Destination node
    Generate BANT to all received FANT
    Calculate QoS residual and set in BANT ▷ using (4.20, 4.21, 4.22)
    Unicast BANT message to its neighbor ▷ At data rate Selected based on RBAR
  Node = Intermediate node
    Receive BANT
    Call SNMP agent using QoS Manager
    if ($B_{local} < B_{min}$) **then** ▷ using (4.6)
      Drop BANT packet
      Return
    end if
    Calculate QoS threshold ▷ using (4.23, 4.24, 4.25)
    Update Routing Table (Lifetime and calculate probability P_{ijd}) ▷ using (4.1)
    Update Neighbor Table (Lifetime and calculate pheromone τ_{ij}) ▷ using (4.3)
    if Source node condition fails **then**
      Unicast BANT
    else
      Kill BANT
      Start Packet Forwarding Phase
    end if
end procedure

is always equal to the desired minimum bandwidth (B_{min}) in our approach according to the equation 4.20. Residual delay (QD) is given as the ratio of the difference between the observed delay on the path and the maximum delay specified by an application to the number of hops (Hop count is 2 in our scenario). In our case it comes out 31.33 ms according to the equation 4.21. The residual success rate (QS) is give as the ratio of the difference between the observed success rate on the path and the minimum success rate specified by an application to the number of hops. In this situation it comes out to be 0.0133 according to the equation 4.22. Now on the link between node d and node k (in the backward phase) the Bandwidth is observed as 5.5 Mbps. Since the QoRA approach is adaptive, high bandwidth corresponds to smaller delays. Therefore at node k the delay is observed as 4 ms. Thus, the threshold delay D_{thr} (k) at node k is 35.33 ms according to the equation 4.24. The threshold success rate S_{thr} (k) at node k is 0.9667 obtained according to the equation 4.25. The process is iterated for all the nodes in the backward phase.

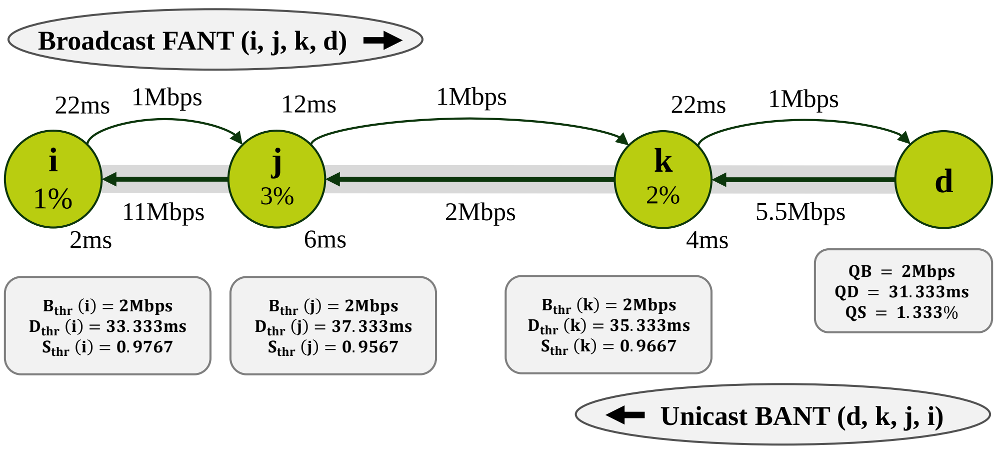


FIGURE 4.17: Example explaining the calculation of residual and threshold of QoS values [130]

4.4.3 Packet Forwarding Phase

After receiving the BANT, the source node starts the packet forwarding phase. Data packet forwarding in QoRA is based on probability (P_{ijd}) that decides the QoS path to be selected from node i to destination d through neighbor node j. When a data packet arrives at an intermediate node, the QoRA reads the flow information in the packet to determine its FlowID. Then, using the entries in its routing table, it randomly directs the packet to the next node based on the probabilistic roulette-wheel selection (RWS) [156]. Where, all data packets contain a unique FlowID that identifies the source, destination, and flowLabel. The flowLabel used as a counter value to identify the data stream/flow at the source node. The intermediate node forwards packets by looking at the FlowID of a data packet. For each flow, multiple paths may be recorded in the routing table. Through data packet forwarding the pheromone on the path and the Lifetime field for the entries in routing table and neighbor table is updated as discussed in earlier section.

The figure 4.18, explains the concept of FlowID in QoRA approach. We consider the situation where the source has identified (n) flows and assigned each flow with FlowIDs as shown in the figure and all these flows are directed towards the same destination. For the situation mentioned above, during the QoRA route discovery mechanism, multiple routes with different QoS are discovered between the given source and the destination pair. The FlowID acts as the decision maker in choosing the appropriate QoS route for the particular flow. For multiple routes that satisfy the quality determined by the FlowID, a route is chosen using RWS probability approach. It is clear from the figure, that the source node defines flow 1, flow 2 and flow 3 and mapped these flows to the different routes that satisfy their QoS requirements, despite the fact that all the traffic is intended for the same destination by the single source. The FlowID and the required QoS parameters are recorded in the routing tables to identify different QoS routes.

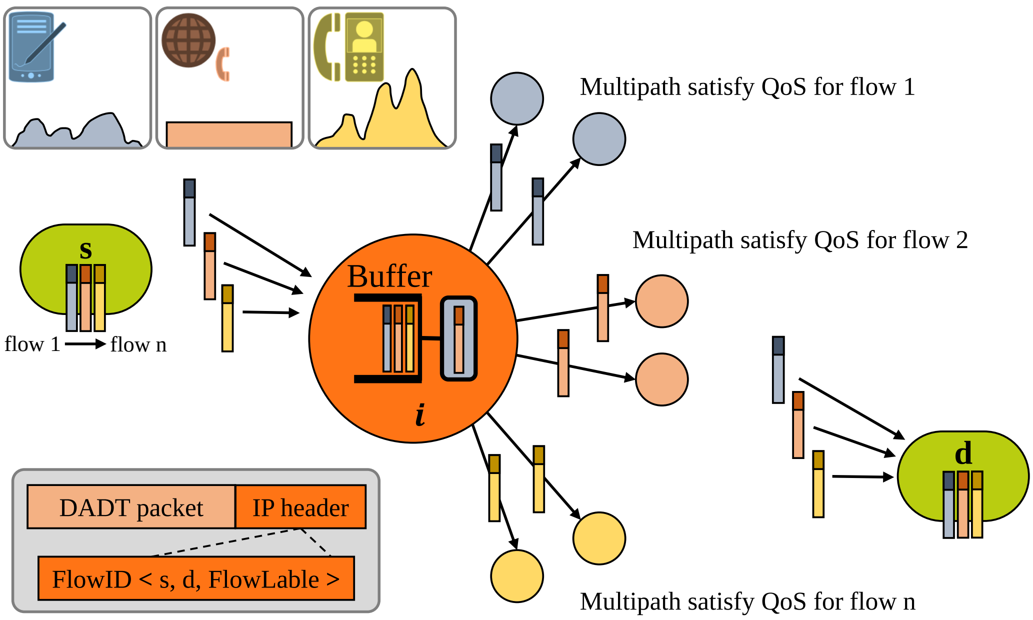


FIGURE 4.18: Example explaining the concept of FlowID in QoRA approach

The RWS is used to select the next neighbor from the pool of neighbors that provide QoS link to forward the data packet from the source to the destination. The individual nodes present in each routing table are mapped on consecutive segments to form a line. The segment size represents the probability to forward the packets to the respective neighbor node as shown in the figure 4.19. The sum of the probabilities is 1. Now, a random number (RNG) is generated between 0 and 1 for every link to these neighbors. The node segment having the generated random number is selected for forwarding the data. This mechanism is analogous to the roulette wheel with each segment size proportional to the probability.

The figure shows the example of the next hop selection process using RWS. Node s has entries for four neighbors i, j, k and l in its routing table to destination d. The quality of each link to these nodes is represented on a scale as shown. Since the segment from node j, spans the random number 0.5 it is selected to forward the data packet. The pseudocode in Algorithm 3 describes the packet forward phase.

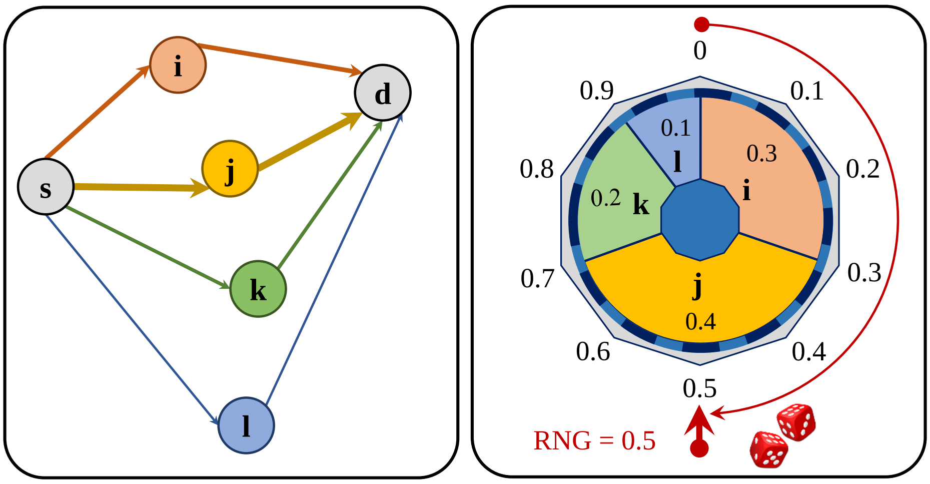


FIGURE 4.19: Example explaining the Roulette Wheel Selection (RWS) mechanism

Algorithm 3 Packet Forwarding Phase

procedure PACKET FORWARDING PHASE
Node = Source node
  RWS select next hop based on random method
  Unicast data packet to its neighbor ▷ At data rate selected based on RBAR
  Update routing table Lifetime
  Update neighbor table (Lifetime and pheromone τ_{ij}) ▷ using (4.3)
Node = Intermediate node
  Receive data packet, read FlowID
  RWS select next hop based on random method
  Unicast data packet to its neighbor ▷ At data rate selected based on RBAR
  Update routing table Lifetime
  Update neighbor table (Lifetime and pheromone τ_{ij}) ▷ using (4.3)
end procedure

4.4.4 QoS Monitoring Phase

The suitability of a route should be monitored once the data flow begins from source to destination. Congestion is a very common problem in the communication networks.

It occurs when the network fails to handle the offered external load. In the figure 4.20, the congestion is observed at the outgoing link of node i, since it does not withstand the packets coming at higher rates from nodes g and h. In this situation, if the necessary steps are not taken to minimize the traffic flow into the network then longer queue sizes are observed at the bottleneck links that will subsequently cause an increase in the packet delays. As a result, the buffer size may get exhausted forcing it to drop the incoming packets. This packet drop then might lead to the violation of maximum-delay-loss specifications. Thus, a specific threshold is sufficient to show that the node has a congestion problem or does not satisfy the QoS requirements.

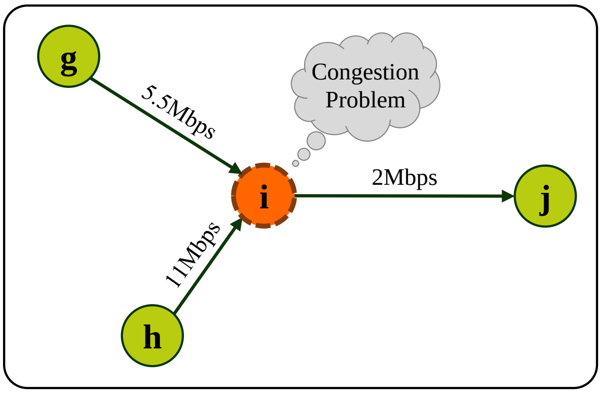


FIGURE 4.20: Example explaining the congestion problem at node i

The bandwidth, delay and success rate QoS violation are determined by continuous monitoring of these parameters in the successive monitoring time window (ΔT).

The auto rate protocol reduces the link transmission speed when the distance between two adjacent nodes increases. As a result, QoRA can avoid and predict QoS violation by monitoring the local available bandwidth $\mathrm{B}_{\mathrm{local}}$ using equation 4.6, as explain in section 4.3.1. Now, the $\mathrm{B}_{\mathrm{local}}$ is compared with bandwidth threshold $\mathrm{B}_{\mathrm{thr}}$ using equation 4.26 for QoS violation.

$$\mathrm{B}_{\mathrm{local}}(n) \geq \mathrm{B}_{\mathrm{thr}}(n, f) \tag{4.26}$$

For the delay QoS violation, the delay observed by each packet $\mathrm{D}_{\mathrm{total}}$ is calculated using equation 4.8, as explained in section 4.3.2. In the next step, the average delay calculated $\mathrm{D}_{avrg} = \sum_{i=1}^{x} \mathrm{D}_{\mathrm{total}}(i)/x$ is calculated in ΔT and compared with the delay threshold $\mathrm{D}_{\mathrm{thr}}$ as shown in equation 4.27. Where, x is the number of packet send through this time window and ϑ is a weight moving factor for average delay.

$$\mathrm{D}_{avrg}^{new}(n) = (1-\vartheta)\mathrm{D}_{avrg}^{old}(n) + \vartheta \mathrm{D}_{avrg}(n) \leq \mathrm{D}_{\mathrm{thr}}(n, f) \tag{4.27}$$

In the case of bandwidth and delay QoS violation, the node looks for another path that satisfies the QoS requirements for the specific flow. If the affected node does not have another path it will delete the routing table entry for the affected flow and indicate the QoS violation by broadcasting the EANT.

Finally, the packet success rate S at the node determined by equation 4.18, as explain in section 4.3.3, and in compared with $\mathrm{S_{thr}}$ as shown in equation 4.28. If the desired packet success rate is not satisfied, then the affected node informs the previous node about congestion problem by broadcasting EANT.

$$\mathrm{S}(n) \geq \mathrm{S_{thr}}(n, f) \tag{4.28}$$

The pseudo code in Algorithm 4 describes the QoS monitoring phase.

Algorithm 4 QoS Monitoring Phase

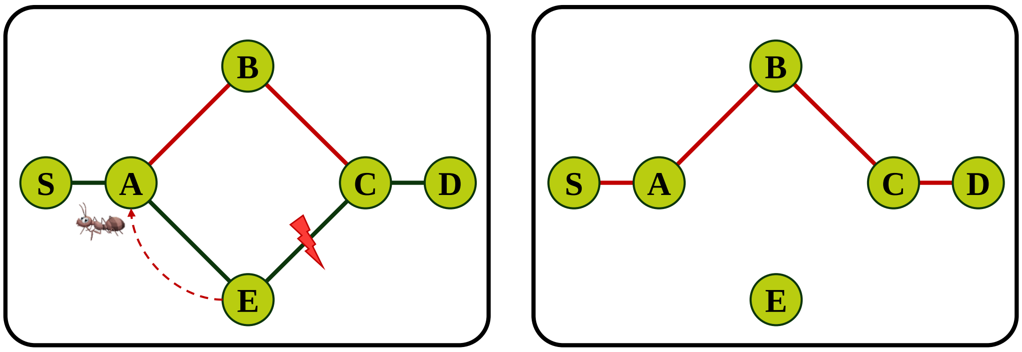
```
procedure QoS Monitoring Phase
Node = Intermediate node
    if ((B_local(n) < B_thr(n, f)) || (D_avrg(n) > D_thr(n, f))) through ΔT then
        Remove routing table entry for the affected flow
        if No other route available in the Routing Table then
            Broadcast EANT message to its neighbors
        end if
    else if (S(n) < S_thr(n, f)) through ΔT then
        Remove routing table entry for the affected flow
        Broadcast EANT message to its neighbor
    end if
end procedure
```

4.4.5 Link Failure Phase

This phase keeps the track of connectivity with their neighbors. Every intermediate node that forwards the data packets to the destination should keep a track of its connectivity to the immediate operational next hops. This connectivity information can be obtained by the available link or network layer mechanisms.

The IEEE 802.11 uses acknowledgment (ACK) and optional handshaking mechanism (RTS/CTS) to indicate successful communications. The MAC layer considers the link failure after attempting to transmit a packet for a limited number of times and notifies the higher layers accordingly. In situations where link layer notification is not available, the acknowledgments at the network layer should be used, i.e. the next hop should acknowledge the reception of packet from the current hop. After detecting the loss of a link or QoS violation to a neighboring node, QoRA deletes the information about this neighbor node from the neighbor table. Then, the node updates its routing table to identify the routes that become invalid because of the link failure or QoS violation and checks another path for the affected flow.

The figure 4.21 shows the utilization of the alternate path for the data transmission in the situations where congestion or link failure is observed on the currently active path. Consider the data is transmitted on the discovered path S-A-E-C-D as shown in figure 4.21a. After a certain time, the node E suffer from the problem of congestion or link failure with the neighbor node C as shown in the figure. In the mean time, the node E will refresh its routing table and check for the alternative path to the destination so that the affected communication can be resumed. In the situations where the node does not have another path to the destination, it will add the flow information (FlowID) to EantStack and will broadcast the EANT to inform the previous nodes in the path about link failures or QoS violation. The neighbor node after receiving EANT will look in its routing table for an alternate path to the destination D for the ongoing communication. In case, the neighbor node does not have the alternate path to the destination, it will rebroadcast the EANT until it is received by the node that has a path to the destination. In our example, the node B has an alternate path to the destination D, so it will drop the EANT, and resume the communication on the new alternate path S-A-B-C-D, as shown in figure 4.21b. If the EANT is received at the source node, the source node has to update its routing table for the affected flow and has to broadcast new FANT to identify new paths to the destination.



(A) Congestion/Link failure according (B) Alternate available path

FIGURE 4.21: The utilization of the alternate path

The pseudo code in Algorithm 5 describes the link failure phase.

Algorithm 5 Link Failure Phase

```
procedure LINK FAILURE PHASE
Node = affected node
    Receive link failure notification or QoS violation
    Update Neighbor table by delete entry with neighbor node
    Update Routing table
    if No other route available in the Routing table then
        Broadcast EANT message to its neighbors
    end if
Node = Source or Intermediate node
    Receive EANT message
    Update routing table
    if No other route available in the Routing table then
        if Source node then
            Start Route Forward Phase
        else
            Broadcast EANT message to its neighbor
        end if
    else
        Drop EANT and continue to send data packet
    end if
end procedure
```

4.5 Summary

In this chapter we discussed the mathematical model of our proposed routing protocol, QoRA in great details. In the beginning, we discussed the two key components of the QoRA architecture namely, QoRA entity and SNMP entity. The QoRA entity decides upon the best route for data forwarding, and it relies on the following sub-components namely FlowID, Routing Table, Neighbor Table, Ant Management, FantID Cache, QoRA Decision Engine, and QoS Manager to make this decision. The SNMP entity performs the task of network monitoring, controlling, accounting, etc. For these tasks, SNMP entity utilizes the services provided by its sub-components namely SNMP Manager and Agent, and Management Information Base. In QoRA, these two key entities are integrated together via the QoS Manager. The SNMP agent provide the information about the quality of the outgoing links to the QoS Manager on the basis of which QoS Manager takes the optimal routing decisions. The SNMP agent consider the following QoS parameters namely, bandwidth, delay, and packet loss to determine the quality of the outgoing link. Finally, we discussed the functioning of QoRA according to the different phases occurring during the data forwarding namely, route forward phase, the route backward phase, packet forwarding phase, the monitoring phase and, the link failure phase. In the next chapter, we will discuss the implementation of QoRA in the network simulator ns-3.

Chapter 5

QoRA Implementation

This chapter gives a short introduction about ns-3, a very popular network simulator among the students and academicians, which is developed especially for networking research purposes. In the following sections, we give the brief overview and the important features of the simulator in general followed by its key components for network modeling. In the later sections, we describe the modeling of QoRA architecture implemented in ns-3 and the major configuration parameters. Finally, we conclude the chapter with a summary.

5.1 The ns-3: Overview

The network simulator-3 (ns-3) [157] is a discrete-event network simulator and as the name suggests is designed to replace its predecessor, network simulator-2 (ns-2). ns-3 is free software, licensed under the GNU GPLv2 (General Public License, Version 2) and is publicly available. It is designed to run on Linux-based systems. However, it can also be run on the Windows system using Cygwin that provides an Unix-like environment for Windows. ns-3's development began in 2006 while its first public version was released in March 2007 and the first stable version came out in June 2008. By the time of writing this thesis the latest version is ns-3.24 released on September 2015 and can be downloaded from the ns-3 homepage.

In addition to ns-2, ns-3 models and implementations are inspired from other legacy network simulators like the Georgia Tech Network Simulator (GTNetS) [158], and Yet Another Network Simulator (YANS) [159]. Hence, ns-3 design addresses the issues and shortcomings of the predecessor network simulation tools making it the most favorable choice for the implementation of our QoRA approach.

5.1.1 The ns-3: Features

As compared to other simulators where the models are described using the domain-specific modeling language, the ns-3 core is written in C++ with an optional Python scripting interface. This design feature enhances the coding style, provides modularity and extends the basic C++ object system to provide additional functionalities and features as listed below. Also, the ns-3 developers provide a well-maintained documentation.

The default values of simulation parameters in ns-3 are managed by an integrated attribute-based system. This system is integrated with other sub systems like command line argument processing. This system empowers the users to vary the parameters value in the simulation without making any modifications to the source code. The user can set the member variables of the modifying object as an attribute and can easily provide the desired values to these variables in the high-level simulation script at run-time.

ns-3 uses function calls to perform a simulation event at given simulation time. With the help of given callback function, a particular event can be created and executed at the scheduled time. The callbacks in ns-3 are also used heavily for reducing the compile-time dependencies between simulation objects. This advantage makes the callbacks a powerful design function in ns-3, employed to decouple layers and models.

The provision of tracing facility is another distinctive feature of ns-3. With the help of tracing the user can discover which events are happening in the simulation and under which conditions at any given time. This feature acts as an important tool for the researcher in deriving and quantifying the values of the important metrics of the simulated model. Also, the output of particular events in ns-3 can be stored in a TEXT file and for packet transmit/receive events it can be stored in PCAP files.

The ns-3 design provides layered Application Programming Interface (API) approach. The powerful APIs are placed at low-level and provides with the higher flexibility in configuration. The helper layer APIs placed above are the simple APIs with easy to use functionality. The users can mix between the upper simpler APIs and the lower powerful APIs to get the desired functionality.

The ns-3 simulator is designed such that it is easily convertible to run in an emulation environment allowing many existing real-world protocol implementations to be reused within ns-3. Also, ns-3 design alleviates the code reuse, encourages the development of realistic models, and eases the simulator control flow enabling the ease in comparison with the real systems.

Also, the networking architecture and the interfaces in ns-3 are inspired by real-world hardware and software properties. The nodes, interfaces, and objects in ns-3 are based on the Linux networking architecture [160]. This feature allows easy integration of certain models from ns-2 or other network simulator to ns-3 and easy reuse of open source software in ns-3.

However, a large number of models have been implemented in ns-2. But, unlike in ns-3, ns-2 does not maintain a coding standard, consistent model verification, and software testing. Therefore, ns-3 does not support backward compatibility to ns-2. The [161] shows that ns-3 is the high-performance simulation tool capable of carrying out the simulation of the large-scale network in highly efficient manner. In [162] it has been shown that ns-3 requires less computation and low memory as compared to other network simulators. Today, ns-3 is a well-supported network simulator where the most critical network models like wireless models have been tested, verified and maintained continuously by the research community.

5.1.2 The ns-3: Simulation Models

The nodes, packets, and channels are the essential objects for the network modeling in ns-3 [163]. Before going into the details, we will first review some common networking terms and their interpretation in ns-3. Later we will describe the architecture of a simple sender-receiver scenario in ns-3 using these abstractions.

- **Node:** It is the abstraction of the basic computing device that can be connected to the network. The applications, protocol stacks, and NetDevices are attached to the node to enable it to do useful work. It is represented in C++ by the class Node.
- **Application:** : It is the abstraction of the user programs that initiates some activity using generation of the packets at the sender and its consumption at the receiver. The application runs on a node and communicates with the network stack. It is represented in C++ by class Application. Each node possesses the list of applications.
- **Packet:** The network packet consists of a byte buffer, a set of byte tags, a set of packet tags, and metadata. Byte buffer stores the serialized information of the headers and trailers added to a packet. The tags contain simulation specific information that cannot be stored in the byte buffer. The differences between these two tags are explained in detail in the ns-3 documentation. The metadata describes the types of serialized headers and trailers.

- **Socket:** It is an interface between an application and the internet stack. Two types of socket APIs are supported by ns-3. The first is native API while the other API is determined with the help of native API to offer a POSIX-like API.

- **Internet Stack:** It provides the implementation of Layer 4 and Layer 3 protocols. Layer 4 transport protocols, sockets, and applications are tied together, and the transport protocols are implemented using a socket factory. Layer 3 protocols like IPv4, IPv6, ARP, etc. are placed above the net devices. Both IPv4 and IPv6 are supported by ns-3. `Ipv4L3Protocol` is the main class for IPv4 and `Ipv6L3Protocol` is the main class for IPv6. Dual stacked nodes i.e. nodes with both IPv4, and IPv6 functionality implementations are also supported.

- **NetDevice:** It is the abstraction of the network card plugged into the IO interface of a node so that it can be connected to the network and consists of both hardware and software driver. It is installed in a node and like real computers it can be connected to multiple communication channels via multiple NetDevices. It is represented in C++ by the class NetDevice. The communication channels determine the NetDevices.

- **Channels:** It is the abstraction of the media over which the information flows between the nodes. It is represented in C++ by the class Channel. The nodes in ns-3 are connected to an object representing the communication channel.

ns-3 uses object aggregation to install an object like applications, protocol stack, net devices, mobility, etc. onto a node. It allows users to install/uninstall the objects according to the required additional functionality without modifying the node class. For example, `MobilityModel` object when attached to a `Node` object will enable the node with a functionality to determine its own position. However, only those objects that belong to a particular class can be attached to a node.

The figure 5.1 represents a scenario for sending and receiving a packet through an IPv4/6 network stack using the above-discussed basic objects in ns-3 [164, 165]. At the sender side, the application running on the top of the node generates the traffic in the form of packets. The data packets are passed to the API. The socket abstraction `[Tcp/Udp]SocketImpl` implements the socket specific aspects and queries the IPv4/6 routing system to find the right source address that matches with the required destination address of the given packet.

`Ipv4/6RoutingProtocol` is the routing protocol abstraction for IPv4/6, and it offers two important interfaces, `RouteOutput()` and `RouteInput()` as shown in the figure. The Linux equivalent of `RouteOutput` is `ip_route_output()`. The `RouteOutput` is used for the packets that are generated locally and does not provide packet forwarding. It queries the routing cache and returns the transport protocol with a valid route to the destination by using pointer `IPv4/6Route`. Using this information the `[Tcp/Udp]L4protocol` implements the socket-independent protocol logic. The packet then flows along the downward arrows via the stack in ns-3 and the necessary features are invoked on its way. Next `IPv4/6Protocol` adds the IP header to the packet and sends the packet to an `IPv4/6Interface`, in our case it is `ArpIPv4/6Interface`. The interface then looks up the MAC address and passes the packet to the NetDevice and then to the channel.

At the receiver, the packet flows upward along the arrow as shown in the figure. Depending on the packet type, the protocol handler passes the packet either to the ARP instance or the IP layer. At Layer 3 the forwarding decision is made using a `RouteInput()` interface. The Linux equivalent of `RouteInput` is `ip_route_input()`. Once the `IPv4/6L3Protocol` receives a packet from a NetDevice, the `IPv4/6L3Protocol` calls the `RouteInput` for doing the further processing on the packet. It either pushes the packet to the upper layers to make local delivery or forwards it to the other node.

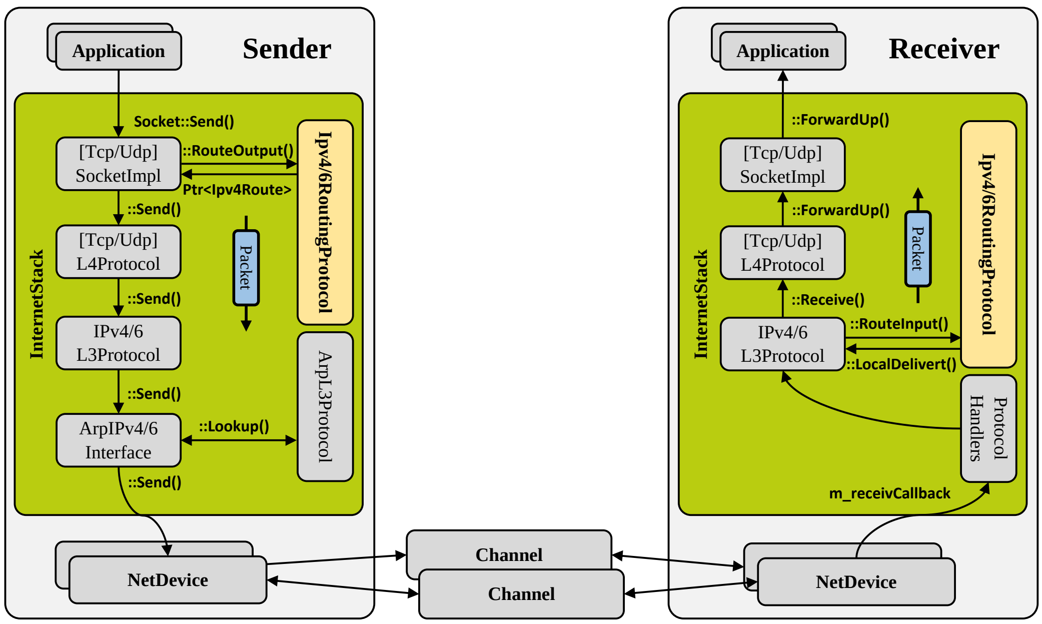


Figure 5.1: The ns-3 node architecture

5.2 The ns-3: QoRA Architecture

The figure 5.2 represents the implementation of the QoRA routing architecture based on the ns-3 framework. The key objects of the ns3 routing protocol architecture are `Ipv4L3Protocol`, `Ipv4RoutingProtocol`, and `Ipv4Route` [166].

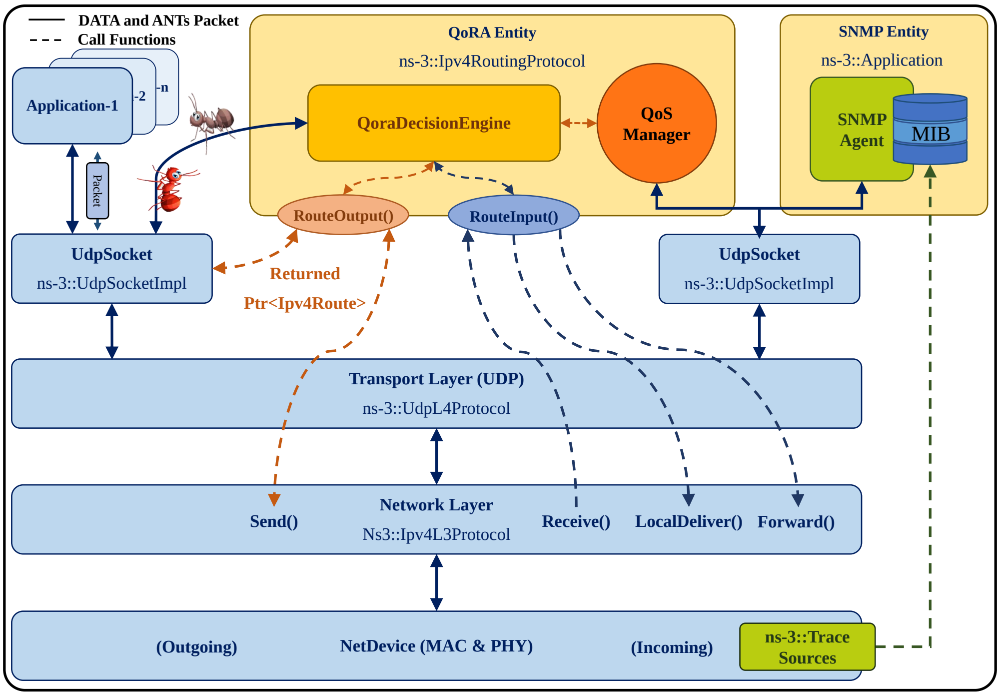


FIGURE 5.2: The QoRA Architecture based on ns-3 framework

At the sender, the application generates the data packets to be forwarded to the destination. This prompts the `UDPSocketImpl` to call the `RouteOutput` for determining the routing protocol. The `RouteOutput` pushes the packet to the `QoraDecisionEngine`. The QoRA then checks for the type of packet, as shown in the figure 5.3 the flowchart of QoRA `RouteOutput`. For the data packet, it issues a query in the routing table and checks for the route availability to the destination. If the route to the destination is not available, the `RouteOutput` tags the packet, creates loopback route with the local address and return `IPv4Route`. The tagged packets are pushed down to the network layer and again `Send` function is called for transferring the packet to the NetDevice. But since the packet has a loopback address the network layer calls the `Receive` function and accepts the tagged packet. The function `Receive` then calls the function `RouteInput` to deliver the tagged packets to `QoraDecisionEngine` for queuing.

In another case, where a valid route to the destination is available, the QoRA returns a valid route from the available multiple routes using the RWS mechanism as discussed in chapter 4 section 4.4.3. The packet is then pushed to the NetDevice by calling the function `Send` and subsequently, it is redirected to the appropriate channel.

In the situations where ant generation is essential, the flow of ant packets in the stack is similar to that of the data packet as discussed above. The `QoraDecisionEngine` after receiving the ant packet determines its type and broadcasts or unicasts the packet accordingly and returns the `IPv4Route`. Also, the `RouteOutput` is called directly from `IPv4L3Protocol` in case of IPv4 raw sockets.

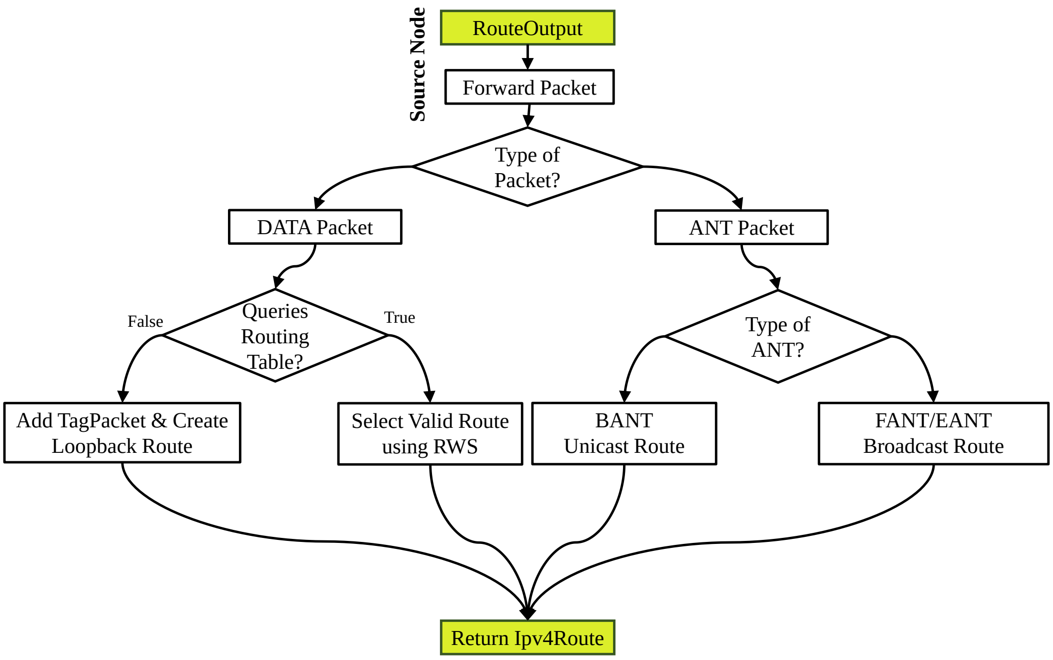


Figure 5.3: Flowchart for QoRA RouteOutput

At the source node if the route to the destination is not available, the `RouteInput` queues the data packets for `QoraDecisionEngine` and the Route Forward Phase is initiated. In the other situation, the intermediate node receives the data packet and calls the function `RouteInput`. The `RouteInput` then chooses one of the following valid outcomes using the appropriate callback.

- Callback for forwarding the packet as unicast.
- Callback for forwarding the packet as multicast.
- Callback for delivering the packet locally.
- Callback for error forwarding.

If the current node is the destination, then `RouteInput` calls the Local Delivery Callback and return. Otherwise, the `RouteInput` calls the Forwarding Callback and forwards the packet on a path selected using RWS mechanism. Figure 5.4 depict the flowchart of QoRA RouteInput.

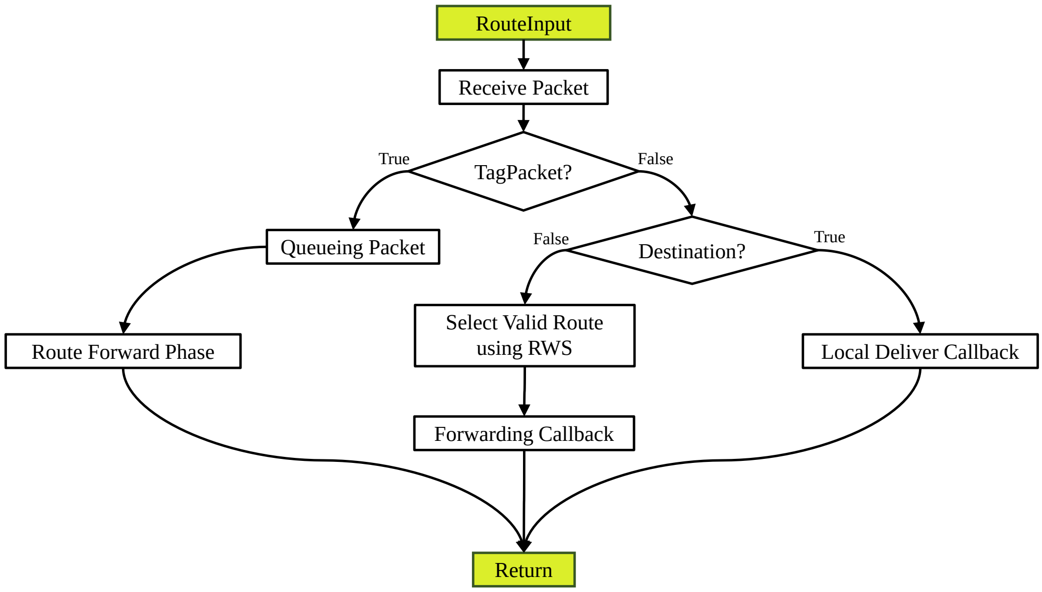


FIGURE 5.4: Flowchart for QoRA RouteInput

5.2.1 QoRA Entity Implementation in ns-3

In this section, we explain the operation of QoRA entity. The implementation consists of the number of classes as shown in the figure 5.5 (A).

The `QoraDecisionEngine` is the main class in the QoRA approach. It is inherited from the ns3 base class `Ipv4RoutingProtocol`. This class implements all the necessary protocol operations like the exchange of ants/packets, handling of link failures, network congestion, QoS violation, etc. In line to the base class, we outlined two virtual functions `RouteOutput` and `RouteInput`, as explained in the previous section. The Qora Decision Engine implementation is highly flexible, as all the dependent parameters are fully adjustable.

The next class is `QoraQueue`. As, the name suggests this class buffers all the data packets in a queue until the QoRA model discovers the demanded routes to the destination. The packet, IP header and the corresponding expiration time are stored in this class using the `QoraQueueEntry` objects. The packet queue also implements the garbage collection for the old packets that are no longer valid, and also puts a limit on the queue size.

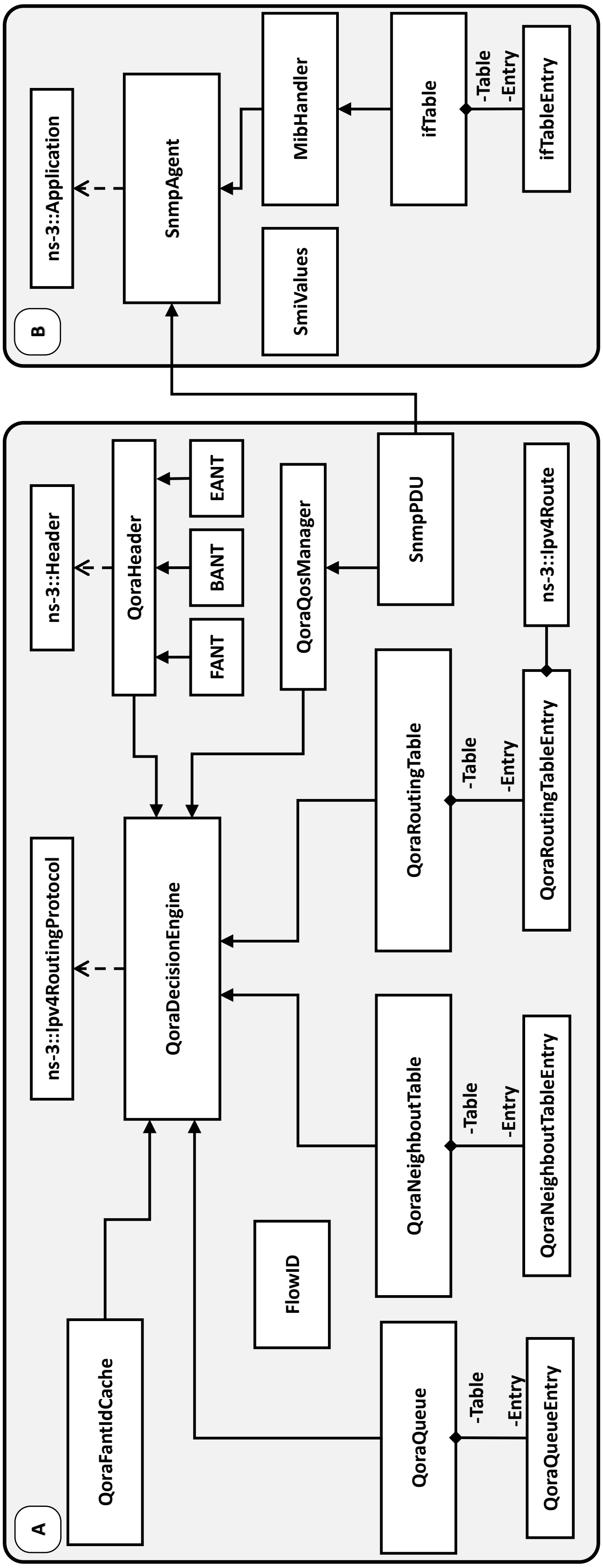


Figure 5.5: UML class diagram for QoRA Approach

In `QoraNeighbourTable` class, the pheromone trail value for each neighbor is stored. The pheromone value indicates the experienced goodness of this link to several destinations. Also, the information representing the mapping between the IP address and MAC address are stored in this class. All these information are collected and accessed through the `QoraNeighbourTableEntry` objects. The required objects can be added, removed, updated and fetched using the methods offered by this class. It also provides callback functionality for handling link failures, conveyed through the notifications from the lower layer.

The `QoraRoutingTable` class contains the route entry for the specific flows and is used to perform the actual routing. Each entry in this table contains information about the route from node i to destination d over neighbor j, along with the corresponding flowID, QoS values, heuristic values, and probabilities values. All these information are collected and accessed through the `QoraRoutingTableEntry` objects. The required objects can be added, deleted, updated and searched using the methods offered by this class.

The `QoraHeader` class defines the valid communication packets for the QoRA routing protocol. It is inherited from the base class `ns3::Header` and consists of three subclasses namely, forward ant `FANT`, backward ant `BANT`, and error ant `EANT`. This class handles the serialization and deserialization methods, manages different field parameters like set and get, and also manages the ant-stack using the functions pop and push.

The `QoraFantIdCache` is used to put a limit to the overhead caused by the broadcast of FANT packet. The table carries entries for each FANT generation, which is based on the unique key that is developed from the combination of FlowID and AntSeq fields. After receiving a FANT packet, the QoRA entity queries the table and compares the quality of the received FANT. Based on this information the node takes the decision about forwarding or dropping the received FANT packet.

The `QoraQosManager` class provides the functionality similar to that of the SNMP manager. It is responsible for creating the SNMP communication sessions for the following operations namely, Get, GetNext, Set, Trap, etc. It also determines the values of QoS parameter by exchanging information with the `SnmpAgent`. The `SnmpAgent` probes the management objects to retrieve the information about the desired QoS parameter. The `SnmpAgent` implementation is described in more details in next section.

5.2.2 SNMP Entity Implementation in ns-3

In this section, we explain the execution of SNMP entity in ns-3 as it has not been available in the ns-3 yet, which includes the SNMP agent and the MIB interface. The SNMP realization utilizes the various classes and the objects as shown in the figure 5.5 (B). The SNMP protocol design and implementation in ns-3 is simple and covers only those features that are required by QoRA approach. Also, to develop full working model of SNMP is out of the scope of this thesis.

The functionalities of the SNMP agent are realized in the `SnmpAgent` class. It includes command responder for replying to the queries raised by SNMP Manager, the notification originator, and an interface to provide accessibility to the MIB.

The `SnmpPDU` class defines the standard SNMP Protocol Data Unit (PDU). This PDU determines the message format of the information that exchanges between the managers and agents. This class consists of two sub-classes namely, `SnmpVB` and `SnmpTarget`. The `SnmpVB` class defines the SNMP variable binding. It binds a pair of elements, where the first element is an object identifier (OID) and the second elements is a value. The variable binding list is represented as the vector of variable binding objects. The `SnmpTarget` class contains the information about the version and community fields for SNMP packets.

The `SmiValues` class represents the Structure of Management Information (SMI), an adapted subset of ASN.1 that defines the sets of related managed objects in a Management Information Base (MIB). e.g. OID, Counter32, Counter64, Gauge, Octet, etc.

The MIB interface implementation consists of various classes. The first is `MibHandler` class, which is implemented using the tracing system in ns-3 [157]. The tracing system consists of trace source and trace sink. Trace sources are entities that provide access to interesting underlying data that is generated when an occurrence of event happens in a simulation. This data is collected using the callback functions. Trace sources must be connected to other pieces of code that actually do something useful with the information provided by the source, such as counting the number of incoming packets like Pkts_{Rx}, outgoing packet Pkts_{Tx}, or dropped packets DropPkts_{in} and DropPkts_{out}. A trace source may also inform about an interesting state change, such as the change in the data rate speed ifSpeed. For the implementation of `MibHandler`, we used various trace sources and callback functions as explained in Table 5.1. The second class is `ifTable`, which contains a list of interface entries that are defined in the class `ifTableEntry`. Each interface in `ifTable` is represented by an `ifTableEntry`, which contains some of the management objects useful to assess the quality of different outgoing links.

Table 5.1: Trace sources and callback functions used by MibHandler

Trace sources	Short description
Interface up/down	Used to add an `ifTableEntry` based on NotifyInterfaceUp and to delete the `ifTableEntry` based on NotifyInterface-Down.
MacTx	Provides notification about the packet being received from upper (internet) layer and queued for transition. This trace source connects to another function called PacketTx which can count the number of outgoing packets (Pkts_{Tx}) from higher layer.
MacRx	Provides notification about the packet being received from lower (physical) layer and forwarded up to the local protocol stack. This trace source connects to another function called PacketRx which can count the number of incoming packets (Pkts_{Rx}) from lower layer.
MacTxDrop	Provides indication about the packet being dropped in this device. This trace source is connected with another sink function called DropPacketOut to provide the number of transmitted packets that were dropped (DropPkts_{out}).
MacRxDrop	Provides an indication about the packet that has been dropped in this device after it has been received from the physical layer. This trace source is connected with another sink function called DropPacketIn to provide the number of received packets that were dropped (DropPkts_{in})
WifiEnqeue	Provides an indication about the packet that has been en-queued and returns the number of packets in the queue.
WifiDeqeue	Provides an indication that a packet has been dequeued and returns number of packets in the queue.
WifiQueueDrop	Provides an indication that a packet has been dropped and returns number of packets in the queue.
PhyRxDrop	Provides a notification that the device has dropped a packet during the reception. e.g. due to the error in the received packet.
PhyTxDrop	Provides a notification that the device has dropped a packet during transmission. e.g. due to some hardware/software issues in the net device.
MTU	The `WifiNetDevice` class provides a function called GetMtu which returns the maximum size of a packet sup-ported by this interface in byte.
MonitorSniffer	Theses trace sources provide a notification that a frame/-packet was transmit/received and returns the current inter-face speed in [b/s].
State	provide the information about the state of the PHY layer such as idle state, busy state i.e. TX (sending packet) or RX (receiving packet), Sleep state, CCA Busy (the medium busy through the CCA mechanism).

5.3 Summary

In this chapter, we discussed the implementation of our proposed routing protocol in the network simulator. In the first of the chapter we discussed the network simulator ns-3 in details and reviewed its features and simulation models. The thorough explanation of node architecture in ns-3 aided in establishing a direct analogy with the real world Linux networking architecture. Then, we discussed the implementation of the QoRA architecture. Here, the route input and route output are the key mechanisms in the routing protocol from the implementations point of view. Later, all the classes utilized for the QoRA implementation in ns-3 are explained using the UML class diagram. In the end, we discussed the implementation of lightweight SNMP agent in ns-3. In the next chapter, we compared QoRA with the other routing protocols and compared its performance under different real-world scenarios.

Chapter 6

Simulation Results and Discussion

In this chapter, we present the in-depth analysis of the QoRA routing protocol and evaluate it under extensive simulation tests within the framework of the network simulator (ns-3). Also, we compared the QoRA to a number of state-of-the-art MANTEs routing protocols for different real-life scenarios. Firstly, we will discuss the method of performance evaluation for QoRA. Later, we describe the basic test scenario on the basis of which different simulations setups can be derived by changing the network parameters. Following this, we test the QoRA over different test scenarios and discuss the results. Finally, we conclude the chapter with a summary.

6.1 Performance Evaluation

In this section, we discuss the different routing protocols that will be used for direct comparison with QoRA. Later, we describe the applications that we will be common for all the scenarios for measuring the performance of QoRA. Finally, we discuss the evaluation metrics that will quantify the performance measurement.

6.1.1 Protocol Used for Comparison

For MANETs, AODV [42, 43] routing protocol is considered as the standard routing protocol for the purpose of comparisons by a large number of authors (e.g., [69, 71, 74, 76, 110, 119, 122, 124, 125, 167]). Also, most of the QoS-aware routing protocols are proposed as an extension of the AODV routing protocol as discussed in chapter 3. So, following the tradition we choose AODV as the first routing protocol for the purpose of comparison. Also, as we use the network simulator ns-3, which does not include these QoS-aware routing protocols, an indirect comparison with AODV seems the practical approach. The second routing protocol that we choose for the purpose of comparison is CLWPR [73, 74]. The protocol as explained in chapter 3 section 3.2.4 is position based QoS routing protocol, and the author has compared this approach with all other MANETs routing protocols already implemented in ns-3. So, by comparing QoRA with CLWPR, we are indirectly comparing all other MANETs routing protocol implemented in ns-3. Thus, three types of routing protocols are examined in this section namely, the traditional routing protocol AODV, the QoS-aware QoRA, and the position based routing protocol CLWPR.

6.1.2 Application Used in Simulation

To verify our proposed algorithm we used the following multimedia applications. First, the constant bit rate (CBR) audio streaming that uses G.729 and G.726 schemes and second the variable bit rate (VBR) video streaming that uses H.264 codecs.

G.729 is a speech codec that operates at 8 kbps and G.726 is another speech codec that operates at 32 kbps. As this codec has low bandwidth requirements, it is used in Voice over IP (VoIP) applications. We mimicked G.729 and G.726 voice flows using an `OnOffApplication` model in ns-3, as real time protocol (RTP) is not implemented and also developing accurate models of VoIP is outside the scope of this thesis. We simulate worst case scenario for VoIP. In the On period, the source transmits data at the rate of chosen audio codec and during Off period no data is transmitted.

H.264 is a video codec that is used for the recording, compression, and distribution of high-definition video. For the video application, we have used the H.264 encoded video trace files. The trace files are simple files generated from the real-time video traffic that depicts the behavior of the source node. It contains the number of bits required to encode each video frame and the quality level of the encoding [168, 169]. The major advantages of using trace files are it does not involve any copyright issues and the easy usability from the networking researcher point of view, as it neither requires a detailed understanding of video coding nor any video coding equipment [170].

The following performance metrics are evaluated during the simulations of these two applications over MANETs and are compared using the routing protocols discussed in previous section.

6.1.3 Evaluation Metrics

We consider the following quantitative QoS metrics suggested by IETF MANET standardization group RFC 2501 [171] to assess the performance of the QoRA routing protocol. The first four parameters are an external measure of the performance and, therefore, determine the effectiveness of the protocol. The next two parameters are the internal measures that define the efficiency of the protocol in terms of the overhead generation and reliability.

- **Packet delivery ratio:** It is given as the ratio of the number of data packets successfully delivered at the destination nodes to the number of data packets generated by the source nodes. This parameter is associated to the reliability of the network and indicates the robustness of the protocol. A packet may be lost due to e.g. congestion, bit error, or link failure.

- **Throughput:** It is given as the ratio of the number of data packets delivered to the destination to the data packets delivery time and this value measure in bits per second. The nodes generating high-speed data streams like live telecasting requires high throughput. It is observed that the high network throughput leads to the better performance of the system.

- **Average end-to-end delay:** It indicates the cumulative delays observed by the data packets while traveling from the source node to the destination node. This term includes the buffering delay caused during the route discovery phase, queuing delay, contention delay, and propagation delay. The real-time applications are highly sensitive applications in terms of delays.

- **Average delay jitter:** It is the variation observed in the delivery time of the subsequent packets due to network congestion, route changes, queuing, etc. For real-time applications, it is an important QoS measure that indicates the robustness and adaptivity of the algorithm in the presence of the disruptive events in the network.

- **Overhead in number of packet:** It is given as the ratio of the average number of control packets transmitted to the number of the data packet deliver at destination. This ratio determines the overhead observed in terms of the number of packets or in the other words it indicates the packet efficiency of the protocol. The packet efficiency determines the protocol's channel access efficiency in contention-based links.

- **Route repair frequency:** It indicates the number of control packets generated to perform the route repairs per second after the communication is interrupted on the given path.

The next section discusses the simulation scenarios where the QoRA routing protocol is evaluated by the above-discussed external and internal measures that will determine its effectiveness and efficiency, respectively.

6.2 Test Simulation Scenarios

In this section, we describe the basic simulation scenario for the testing of the routing protocols. The different test scenarios are derived from this basic setup by varying different network parameters like the size of the network, the speed of the node, etc. The fundamental simulation scenario is designed considering the most common simulation scenarios so that an unbiased comparison can be made between the different routing protocols under the test. In the later section, we discuss the number of derived scenarios and the performance of the routing protocols.

6.2.1 Basic Simulation Setup

The radio propagation is simulated using the Friis propagation loss model [172] as described in ns-3. The Friis free-space model overshadows the two-ray model [173] for small distances, as two-ray model suffers from the constructive and destructive combination of the two-rays. The physical layer is configured in line to the IEEE 802.11b standard as described by the Yet Another Network Simulator (YANS) in ns-3 [159]. The MAC layer is modeled using the IEEE 802.11 DCF protocol as discussed in chapter 2 section 2.2. The channel speed over the wireless link was controlled depending on RBAR [23] using ns-3 `IdealWifiManager`. The channel rates varied with different ranges, including 1, 2, 5.5, and 11 Mb/s according to the signal-to-noise ratio between the neighbor nodes. Finally, we use the UDP protocol at the transport layer as TCP protocol behaves badly in mobile ad-hoc networks and real time multimedia application.

The simulation is carried out in an open area as shown in figure 6.1 with different size based on the number of nodes. The nodes mobility is characterized by random waypoint mobility (RWP) model [40]. In this model, every node starts from a random initial position and moves towards the random destination with a random speed selected from the given range of minimum and maximum speed. After reaching the destination, the nodes remains at rest for the random time selected from the given pause-time range and then it chooses a new speed and destination. In our setup, the nodes choose the random speed between 1 m/s (3.6 km/h) to 10 m/s (36 km/h) and a constant pause time 30 s. To test the performance of QoRA protocol for grid and urban mobility scenarios we used the mobility model BonnMotiona [174] and Simulation of Urban MObility (SUMO) [175].

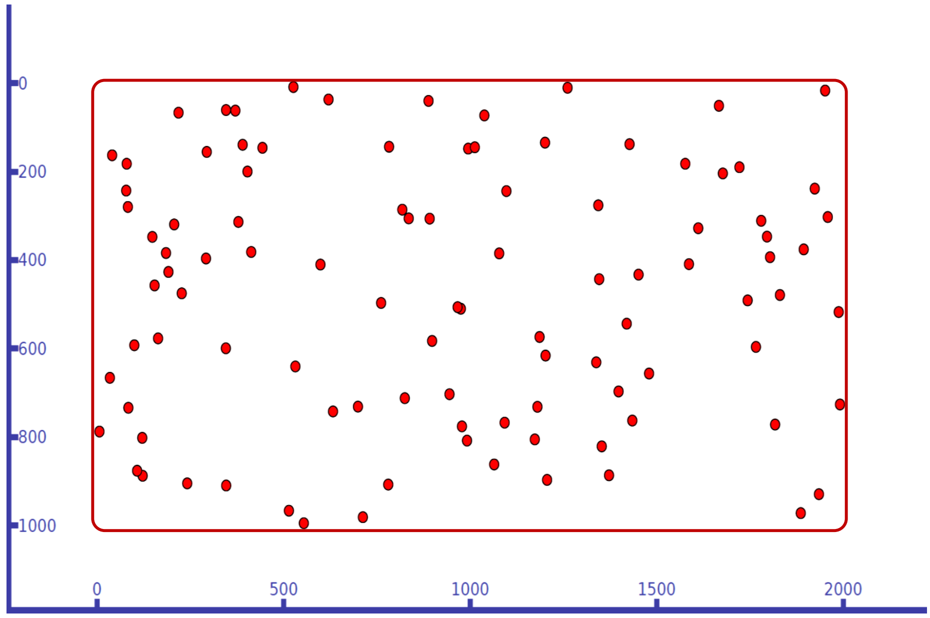


FIGURE 6.1: Examples of the open area RWP mobility with 100 nodes

The application data session starts at a random time between 10 s and 40 s and continues until the end of the simulation time at 500 s. We repeat every simulation for 20 times and the randomness across each of these multiple simulation is configured using a pseudo-random number generator that produces random results.

Table 6.1 presents the settings of the different weight factors used by the QoRA routing algorithm that will remain unchanged throughout all simulation scenarios. The following QoRA parameters are determined through numerous simulation trials. Also, the type of application determines the appropriate QoS constraints values. The bandwidth B_{min} is determined by the application data rate so that it covers the UDP/IP headers and for that we have increased this value to 4 kbps. The delay D_{max} for each communication flow is selected randomly from the range 150 ms to 250 ms. Finally, the desired success rate S_{min} is also selected randomly between 97% to 100%. These are the selection and monitoring parameters considered by QoRA. In order to make each communication flow heterogeneous requirements, we choose random constraints for each of the flows.

Table 6.1: The settings of the different weight factors used by QoRA

Parameter	Value
Initial pheromone value τ_{ij}	0.02
Pheromone increment factor $\triangle\tau_{ij}$	0.01
Pheromone weight factor α	0.7
Heuristic weight factor β	0.3
Bandwidth weight factor β_B	1.0
Success rate weight factor β_S	1.0
Delay weight factor β_D	1.0
T_{idle} weight moving factor θ	0.5
Delay weight moving factor ϑ	0.5
Success rate weight moving factor φ	0.5
Monitoring window ΔT	3 s

6.2.2 Test Scenario 1: Varying numbers of nodes and communication flows

In the first scenario, we test the routing protocol by varying the three network variables namely the number of nodes, the communication flows and the size of the simulation area, as the same scenario in conference paper [129]. The simulated network consists of a number of nodes ranging from 10 to 100. On the basis of the number of nodes, we choose between 4 to 10 communication sessions to send voice traffic at a constant bit rate (CBR) of 8 Kbps with a small packet size of 64 bytes using the ns-3 `OnOffApplication`. Also, the simulation area is regularly increased with an increase in the number of the participating nodes and the data sessions to maintain the constant node density. However, an increase in the size of simulation area also causes an increase in the length of the path between nodes. The results for this scenario are shown in figure 6.2.

Table 6.2 presents the settings information of the network variables and QoS constraints for the first scenario.

The significant difference between the first version i.e. QoRA I and final version i.e. QoRA is the enhancement in the mechanism for forwarding of FANT as explained in chapter chapter 4 section 4.2.1.5 and QoS monitoring phased in section 4.4.4. In QoRA I, the FANT packets are broadcasted using the blind method. In this method, each node forwards the constant number of FANT and does not consider the goodness value

TABLE 6.2: The parameters for the first scenario and QoS constraints

Parameter	Value			
Number of Nodes	10-20	30-50	60-80	90-100
Number of Flows	4	6	8	10
Size of Area (m^2)	400×800	700×1250	800×1700	1000×2000
Application data rate (CBR)	8 Kbps			
Packet size	64 bytes			
Bandwidth constraint B_{min}	12 Kbps			
Delay constraint D_{max}	150-250 ms			
Desired success rate S_{min}	97%-100%			

(ψ) of incoming FANT. Therefore, a node may forward the bad FANTs and discard the new FANT having a high goodness value. This method creates an additional overhead that leads to the increase in the delay and jitter. In the final version QoRA, the FANTs are forwarded purely on the basis of their goodness value. If the goodness value of an incoming FANT is higher than that of the previously received FANT from the same previous node, it will be forwarded and if not the node will just discard it. This leads to the creation of mesh network that improves the reliability and throughput. Also, the final version QoRA is equipped with QoS monitoring mechanism that mitigates the congestion problem by reducing the load at the congested node through eliminating the flow that does not satisfy QoS constraint based on the local QoS threshold.

Figure 6.2a shows the average packet delivery ratio curve plotted against the increasing number of nodes in the simulation network. It is evident from the curve that both the versions of QoRA routing protocol outperform the AODV protocol. This is because QoRA provides multiple paths between the given source-destination pair while AODV provides the single path, which is susceptible to link failures, congestion, etc.

Figure 6.2b shows that QoRA throughput is higher than AODV throughput, a behavior similar to the one observed in figure 6.2a. This is because QoRA finds multipath that satisfies the QoS requirements and spreads the data packets over these paths and yielding high network throughput.

Figure 6.2c shows that our QoRA mechanism performs better than AODV considering the average end-to-end delay. With the increase in the number of nodes, the delay observed by AODV protocol increases rapidly whereas by QoRA it is comparatively very low. The reason for this is that QoRA selects paths with minimum delay and faster transmission speed. Additionally, QoRA does not use any periodic control message, thereby significantly reducing network overhead and avoiding the probability of packet collision and interference with regular data transmission.

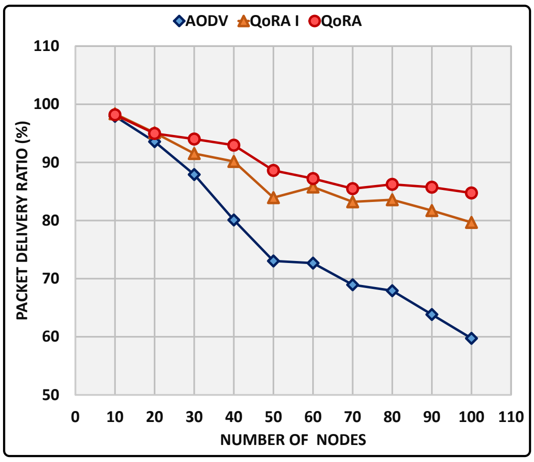
Figure 6.2d shows that QoRA mechanism performs better than AODV considering the jitter constraint. With the increase in the number of nodes, the jitter observed by QoRA and AODV increases. This is because QoRA provides multiple paths and that increase the jitter as the packet select different paths. On the other hand, AODV provides the single path that might suffer from the link failures and requires reestablishment of the path.

Figure 6.2e shows that the overhead in the number of packets is much higher in AODV than in QoRA. This is because QoRA uses three types of packets FANT, BANT, and EANT while AODV uses four types of packets RREQ, RREP, RERR, and Hello message that is transmitted every second.

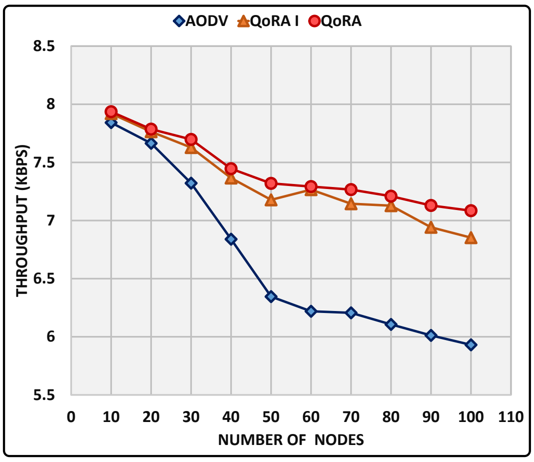
Figure 6.2f gives an idea about the reliability of the protocol. The frequency to discover the routes increases with the increase in the number of nodes and the number of flows. AODV shows high route discovery frequency as compared to the QoRA. This is because with the increase in the number of nodes the single path determined by AODV protocol suffers from frequent link failures whereas QoRA provides multiple QoS-aware paths.

The most significant drawback of AODV over QoRA routing protocol is its path selection mechanism. AODV selects the path from source to destination on the basis of the number of hops. This means for the given application it chooses only one path that has least number of hop counts or intermediate nodes between the given source-destination. However, as shown in figure 6.3a, for multi-rate wireless networks, this path selection mechanism followed by AODV has the tendency to pick the paths with low reliability and low effective throughput. The other issue faced in shortest path mechanism is their short lifetime. This means that on these paths the physical distance of the links is very large. So, a small movement of the nodes involved in this single path will lead to packet loss due to link failure and consequent heavy AODV path maintenance or path reestablishment procedures. On the other hand, the QoRA approach as shown in the figure 6.3b chooses the path that satisfies the QoS requirements and does not involve longer physical links, which ultimately increase the throughput and reliability.

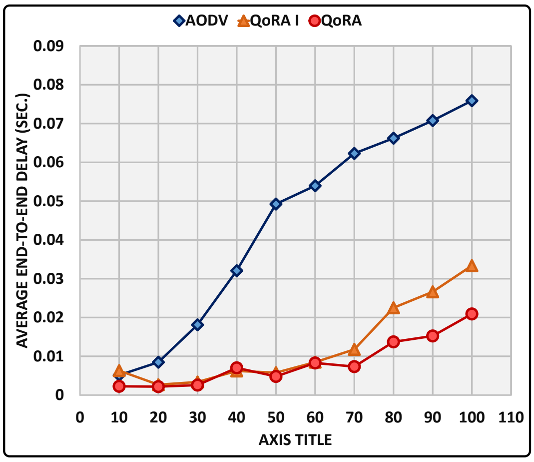
Also, in AODV the route discovery phases, every node that receives RREQ message has to maintain a routing table entry indicating the reverse path availability to the source node. However, any link failure in this reverse path is conveyed using RERR message. Thus, the routing tables are maintained by broadcasting of RERR messages, which increases the overhead. On the other hand, the QoRA uses the AntStack for tracking the reverse path and does not require to create a routing table entry to source node after receiving FANT. Therefore, all the other QoS-aware routing protocols (e.g. [55, 69–71, 76, 86, 117, 125]) derived from AODV protocol are inherited with these problems and may suffer from higher overhead.



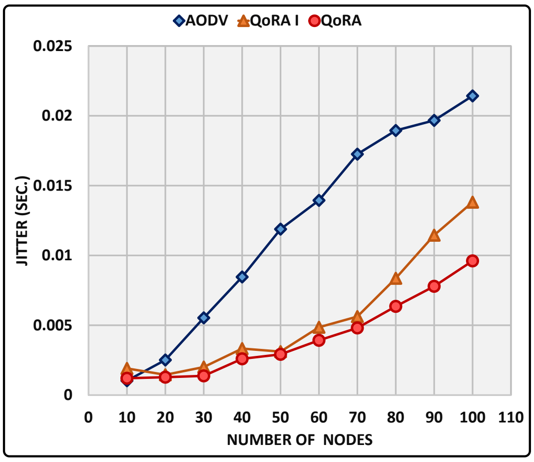
(A) Packet delivery ratio



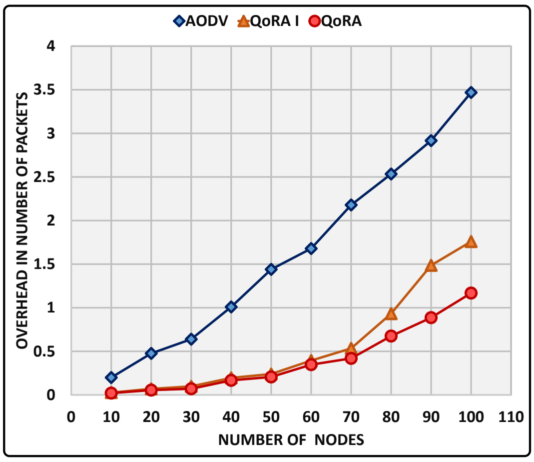
(B) Throughput



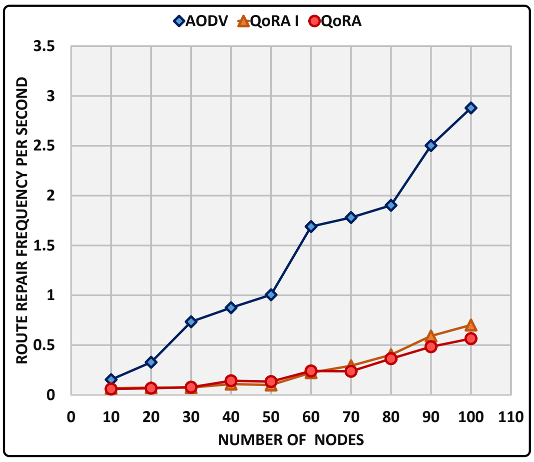
(C) Average end-to-end delay



(D) Jitter

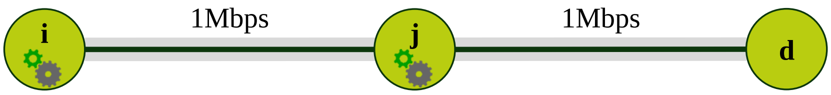


(E) Overhead in number of packets

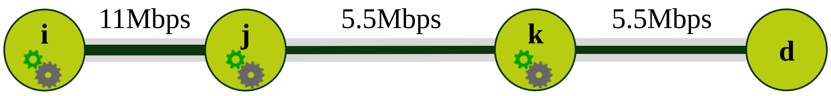


(F) Route repair frequency per second

Figure 6.2: The performance of QoRA using different number of nodes and flows



(A) AODV selection mechanism



(B) QoRA selection mechanism

FIGURE 6.3: Selection based Multirate QoS-aware

6.2.3 Test Scenario 2: Varying application data rates

The experimental analysis described here and in the next section investigates the performance of the protocols against the varying network loads. The load in the network can be varied by varying the application data rate or by varying the number of simultaneous communication flows. Here, we will discuss the effect of varying the application data rates. In this scenario, we simulated 100 nodes with 10 communication flows and varied the application data rate from 4 kbps to 40 kbps in steps of 4 kbps and a packet size of 160 bytes which is more than from the first scenario using ns-3 `OnOffApplication`. However, the increasing data rate leads to the high probability of the occurrence of the congestion and the interference caused by the channel contention in the network. The results for this experimental setup are shown in figure 6.4.

Table 6.3 presents the settings information of the network variables and QoS constraints for the second scenario.

TABLE 6.3: The parameters for the second scenario and QoS constraints

Parameter	Value
Number of Nodes	100
Number of Flows	10
Size of Area	1000×2000
Application data rate (CBR)	(4 – 40) Kbps
Packet size	160 bytes
Bandwidth constraint B_{min}	(8 – 44) Kbps
Delay constraint D_{max}	150 – 250 ms
Desired success rate S_{min}	97% – 100%

As shown in the figure 6.4a, it is evident that the QoRA performs better than the AODV. At a higher data rate of 24 kbps, the performance of AODV experiences a steep fall. However, the QoRA is also affected with the increase in the data rate but unlike AODV, it does not experience a sudden drop in performance. The average packet delivery ratio is observed to be 83% in QoRA. The average packet delivery ratio decreases below the desired success rate S_{min} due to mobility and link failures. This behavior can be explained by the inability of AODV protocol in handling congestion problem. In this simulation, the increase in the data loads causes the congestion in the network on the resource-constrained links leading to the packet loss. AODV fails to understand this problem and considers it as the link failure and triggers the route repair or route setup mechanism. This mechanism, in turn, worsens the network situation by initiating high flooding of RREQ messages to the already loaded and congested network. On the other hand, QoRA chooses the QoS path for the communication that constitutes of links working at higher data rates that help to avoid the problem of congestion. Also, QoRA protocol triggers the congestion avoidance mechanism as discussed in chapter 4 section 4.4.4 and in the next section of this chapter.

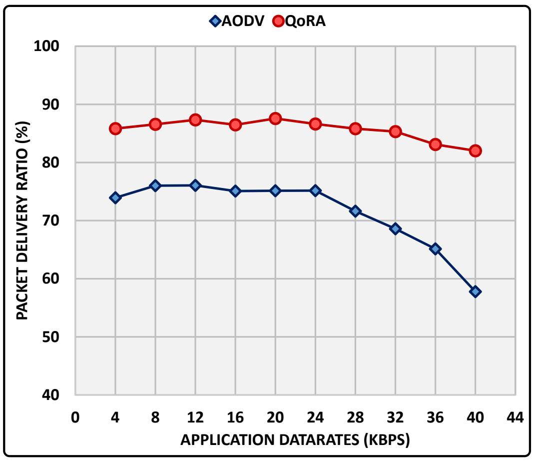
As shown in figure 6.4b the throughout increase linearly with increase of the data rate. The QoRA provide high network throughput performance as compared to AODV.

The end-to-end delay as shown in the figure 6.4c follows the same trend as that of the average packet delivery ratio. For QoRA, the delay is observed to be almost constant because QoRA provides QoS monitoring phase that monitors the QoS parameters and helps to keep this value constant, while, for AODV, a steep rise is observed at high data rates.

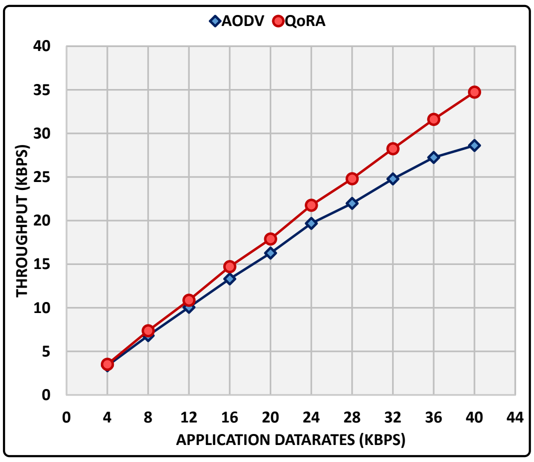
The performance of QoRA for jitter constraint is showed in figure 6.4d. At lower data rates QoRA and AODV demonstrates high jitter. This is because, at lower data rates, the network undergoes through comparatively fast topology changes, causing variations in delay between the consecutive data packets that aggregates to high jitter. At higher data rates the performance becomes stable but QoRA provides the lowest jitter.

The figure 6.4e shows the overhead in the number of packets curve. This is because, at higher data rates, more packets are transferred before the information gathered by routing algorithms gets outdated. However, at higher data rates the AODV suffers from the congestion and experiences a rapid increase in the control packets as explained before for the results obtained for average packet delivery ratio.

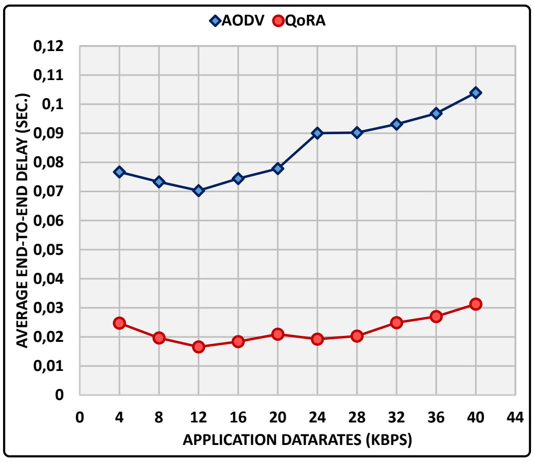
Figure 6.4f shows the curve for route discovery frequency. For QoRA, it is observed to be almost constant as the protocol does not suffer from congestion and channel contention. In contrast, AODV suffers from higher link failures because of mobility and channel contention and, therefore, causing larger route discoveries at higher data rates.



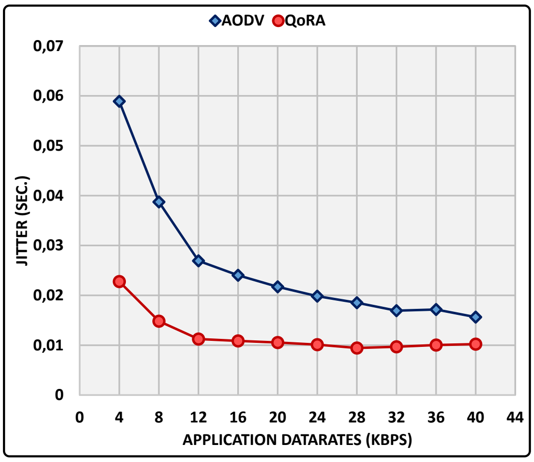
(A) Packet delivery ratio



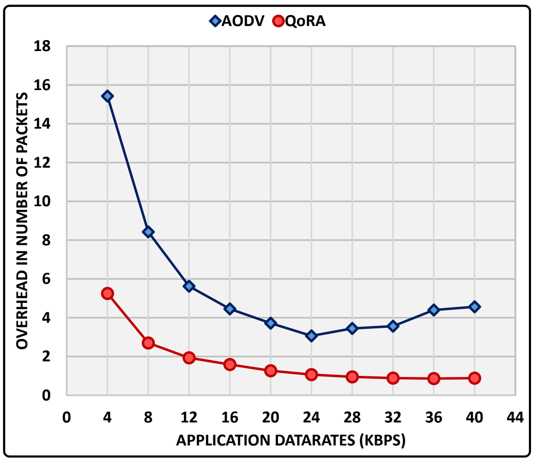
(B) Throughput



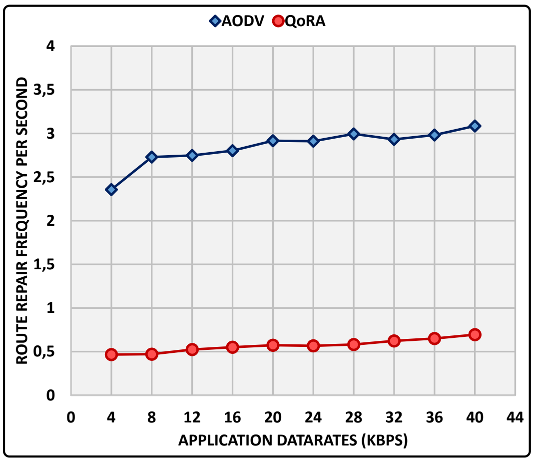
(C) Average end-to-end delay



(D) Jitter



(E) Overhead in number of packets



(F) Route repair frequency per second

FIGURE 6.4: The performance of QoRA using different application data rates

6.2.4 Test Scenario 3: Varying numbers of communication flows

This is another experimental setup to test the performance of the protocols for the varying network load in terms of multiple communication flows. We use from 2 to 20 communication flows (sessions) varying in the step of 2 flows at a time with different sender and receiver to send voice traffic (G.726) at a constant bit rate (CBR) of 32 kb/s (25 packets/s) and a packet size of 160 bytes using ns-3 `OnOffApplication`. Increasing the network load through multiple communication sessions creates a different effect than sending data at higher rates. In multiple sessions, the increased data is spread out in the network, however, the establishment of multiple paths are required for multiple sessions. Thus, in this scenario, the problem of congestion is more severe than the previous scenario. The results for this simulation setup are as shown in figure 6.5.

Table 6.4 presents the settings information of the network variables for the third scenario and QoS constraints.

TABLE 6.4: The parameters for the third scenario and QoS constraints

Parameter	Value
Number of Nodes	100
Number of Flows	2 – 20
Size of Area	1000 × 2000
Application data rate (CBR)	32 Kbps
Packet size	160 bytes
Bandwidth constraint B_{min}	36 Kbps
Delay constraint D_{max}	150 – 250 ms
Desired success rate S_{min}	97% – 100%

The figure 6.5a shows the performance curve for average packet delivery ratio, which is very similar to the curve obtained in the previous scenario. As the number of flows increases, the AODV protocol exhibits a high sudden drop in the performance as compared to the previous scenario. This is because the increase in the number of flows demands more route setup and also makes the network more susceptible to congestion. A similar drop is observed in the curve for QoRA, however, its performance is much better than AODV and the QoRA protocol still to deliver data packets up to 70%. Additionally, the packets delivery ratio decreases not only because of congestion but also due to the mobility and link failure. Similarly, the throughput curve as shown in figure 6.5b also sees a declivity with the increase in the number of communication flows. However, the QoS-aware multi-paths discovered by QoRA spread the data packets stochastically on these paths and improve the value of these metrics .

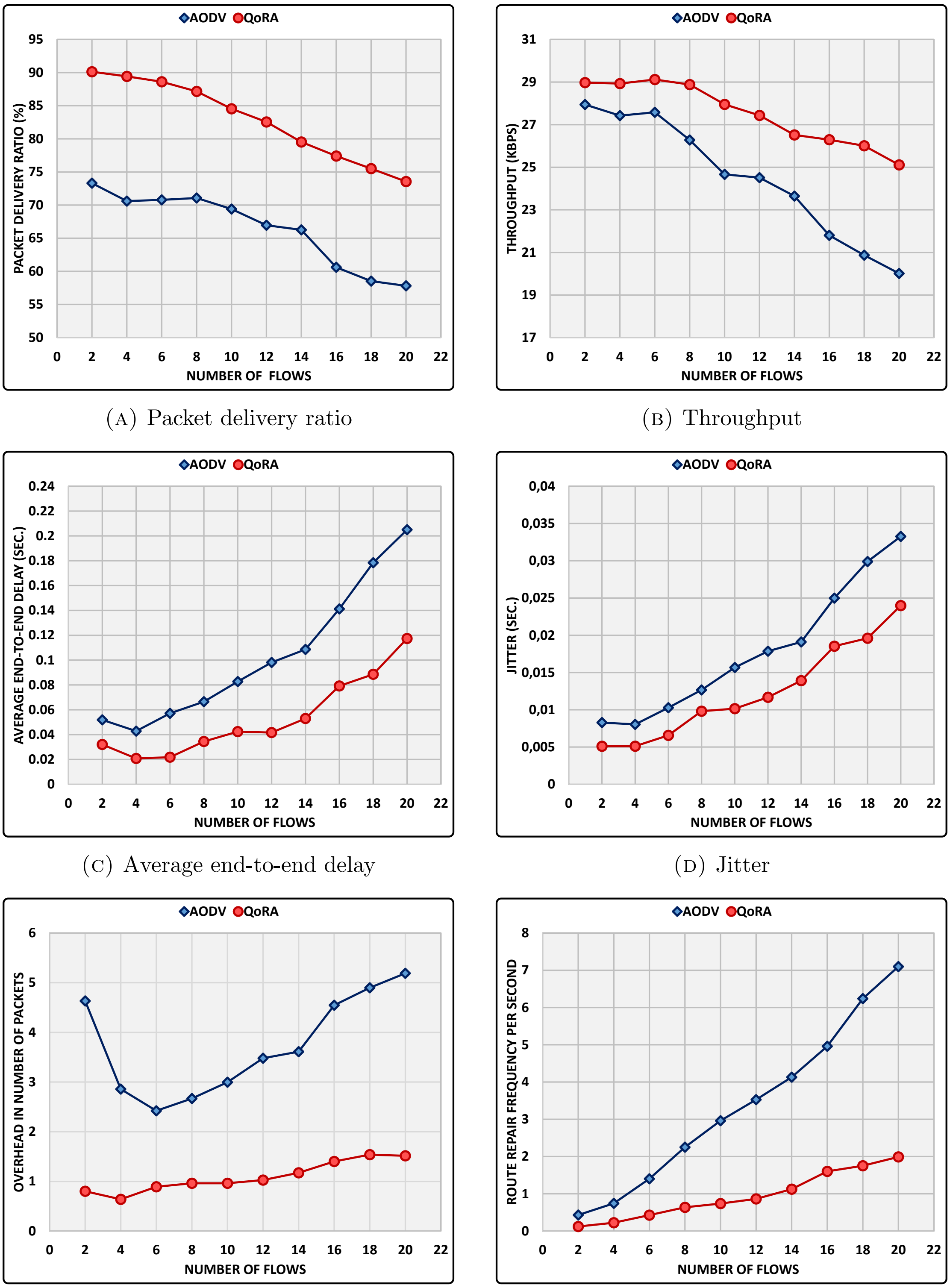
For the average end-to-end delay as shown in figure 6.5c, the obtained curve is again similar to the one from in the previous scenario. The delay increases with the increase in the number of communication flows due to the busy links causing longer interface queues and congestion problem. However, QoRA outperforms the AODV routing protocol.

The results obtained for jitter as shown in figure 6.5d are significantly different than those obtained in the previous scenario. However, the increase in the number of communication flows also increases the number of route setups, link failures and channel contention that leads to the increase in the jitter.

Figure 6.5e shows the curve for overhead in the number of packets. For the small number of flows, the overhead incurred by AODV is much higher than QoRA. This is because the number of received data at the destination is low when the numbers of flows are minimum. However, for QoRA, the overheads are observed to be still low even at higher communication flows. This is because unlike AODV, the QoRA does not use any periodic message or other proactive communication methodology when the nodes do not have any data to forward.

Figure 6.5f shows the curve for routes discovery frequency. For large communication flows, the route frequency increases drastically for AODV. This is because the increase in the number of flow increases the contention and collision. Also, as discussed earlier AODV employs periodic Hello message to track the neighbor nodes activity, which adds to increase the problem of contention and collision. Due to this AODV fails to deliver the data packet and after a certain number of retries the node assumes the link failure and triggers the path reestablishment mechanism. However, QoRA provides multiple routes and it is not affected by these problems, as it is evident from the figure.

In figure 6.6, we explain the QoRA congestion adaptation mechanism. Where in subfigure 1, we have one communication flow between source S and destination D happening over multiple paths as discovered by QoRA. In subfigure 2, the new communication flow starts between source node X and destination node Z. So now we have 2 communication flows happening in the network over multiple paths. As shown in subfigure 3, both the communication flows 1 and 2 are using the intermediate node B and because of that it now suffers from the problem of congestion. The node B confirms the congestion by comparing the local QoS value with the QoS threshold value determined for each flow. So node addresses the problem of congestion by sending EANT to the neighbor nodes (A, F, X, Y). The nodes A and F after processing the EANT will now remove the routing entry for node B for flow 1 as shown in subfigure 4. However, the flow 2 will continue using the communication via node B. Similarly, in subfigure 5, the node C experience the problem of congestion and again it addresses it in the same manner as discussed above and removes the routing entry for node Z for flow 2 as shown in subfigure 6.



(A) Packet delivery ratio

(B) Throughput

(C) Average end-to-end delay

(D) Jitter

(E) Overhead in number of packets

(F) Route repair frequency per second

Figure 6.5: The performance of QoRA using different number of flows

Thus, using this adaptation mechanism QoRA can adapt to the increase in the number of communication sessions strongly and can keep the QoS constraint for each flow.

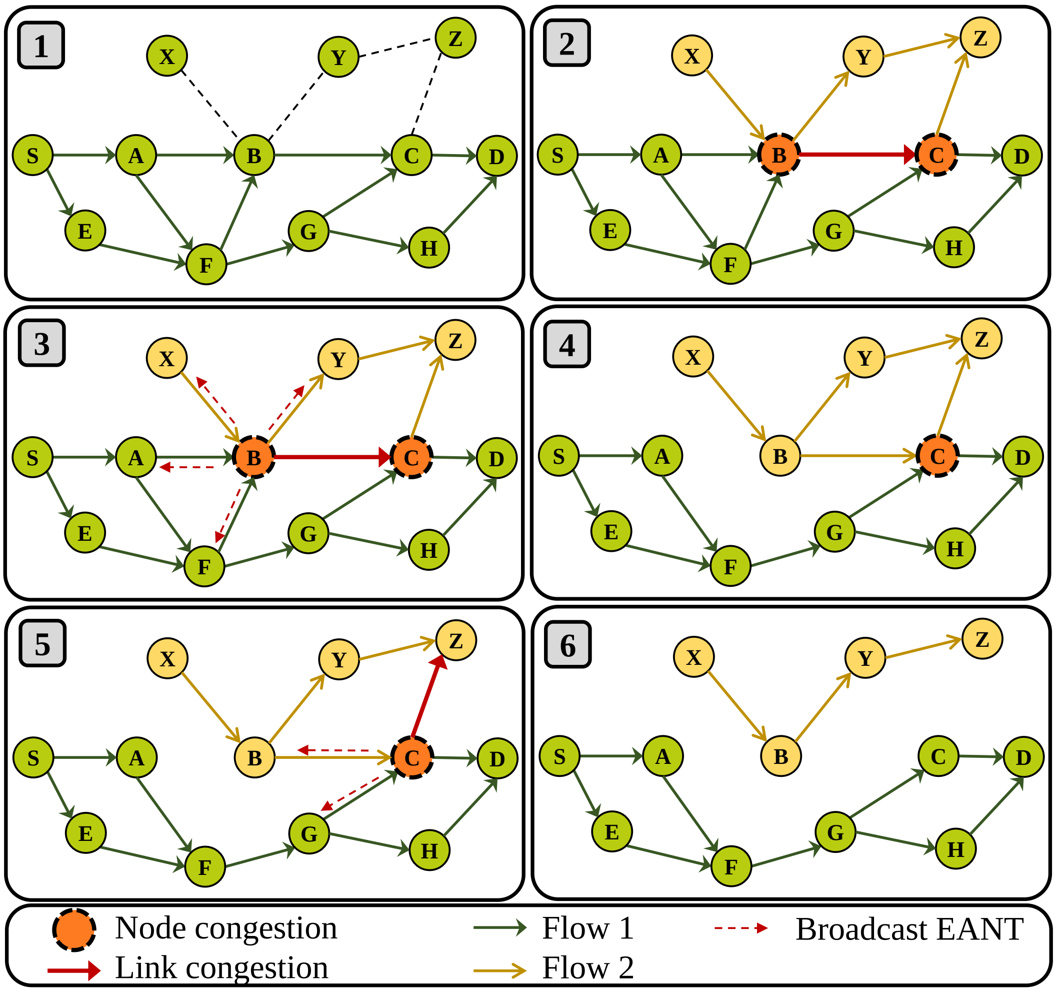


FIGURE 6.6: Example of congestion adaptivity based on QoS threshold

6.2.5 Test Scenario 4: Varying the pause time of the mobile nodes

Increasing and decreasing the pause time in RWP mobility has interesting effects in the performance of the routing protocol. In this scenario, we vary the pause time of the RWP mobility model from 30 s up to 480 s, with intermediate points having value, twice of the previous one. When the pause time of the nodes is low, the nodes become highly mobile and the network becomes more dynamic. Similarly, when the pause time of the nodes is high, the nodes become less mobile and the network becomes static. However, due to the high pause time, the scenario becomes less difficult. In this scenario, we will test the sensitivity of QoRA for these situations. The results for this simulation scenario are as shown in the figure 6.7.

Table 6.5 presents the settings information of the network variables for the fourth scenario and QoS constraints.

Table 6.5: The parameters for the fourth scenario and QoS constraints

Parameter	Value
Number of Nodes	100
Number of Flows	10
Size of Area	1000×2000
Pause time	30 – 480 s
Application data rate (CBR)	32 Kbps
Packet size	160 bytes
Bandwidth constraint B_{min}	36 Kbps
Delay constraint D_{max}	150 – 250 ms
Desired success rate S_{min}	97% – 100%

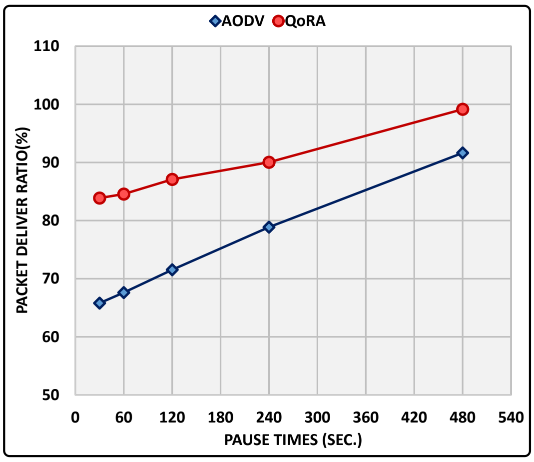
The figure 6.7a shows the average packet delivery ratio for QoRA and AODV. As the pause time increases, both the protocols show improvement in the average packet delivery ratio. QoRA shows the good performance and the protocol can deliver 99% when the pause time is 480 s. However, the average packet delivery ratio exhibited by AODV is lower than QoRA since AODV provide single path susceptible to congestion while QoRA provides multiple QoS-aware paths. Additional, QoRA can adaptive to congestion problem as experienced in provisos section.

The throughput curve shown in figure 6.7b shows the similar behavior for the average packet delivery ratio. The high pause time improves the network throughput as the network converges to the static state.

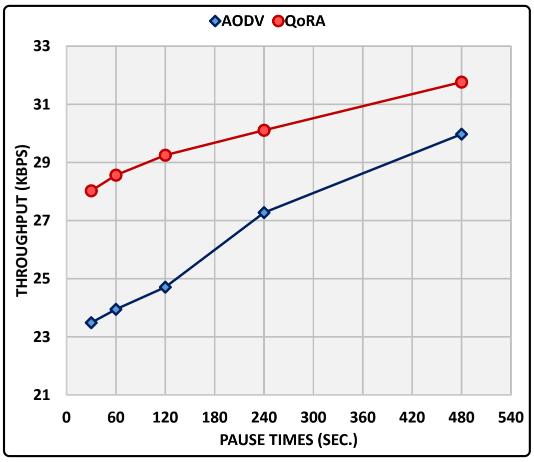
The average end-to-end delay and jitter curve are shown in figures 6.7c and 6.7d. The value for these parameters decreases with the increase in pause time. QoRA continues to show good results in comparison with AODV for the delay and jitter.

Figure 6.7e shows the overhead in the number of packets. For high pause times, these values decrease drastically. This is because high pause time provides high packet delivery ratio and, therefore, curbs the need for control packets.

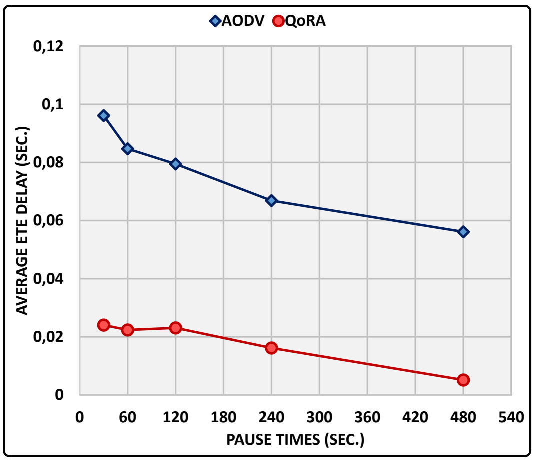
Finally, figure 6.7f shows the route discovery frequency curve. The curve follows the similar trend to that of the overhead in the number of packets. The AODV shows high route discovery frequency as compared to the QoRA as it suffers from the packet collision on the single paths between the data packet and control packet, which causes it to trigger route reestablishment process.



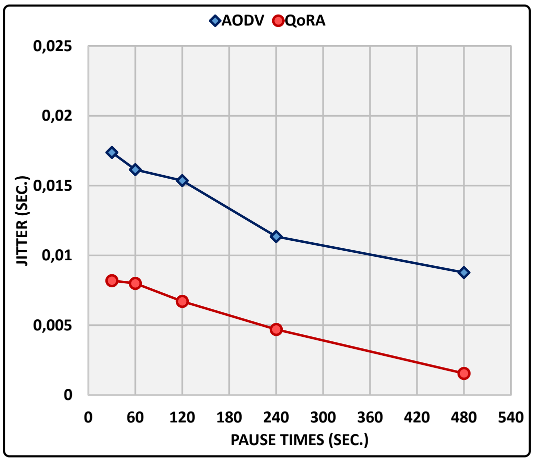
(A) Packet delivery ratio



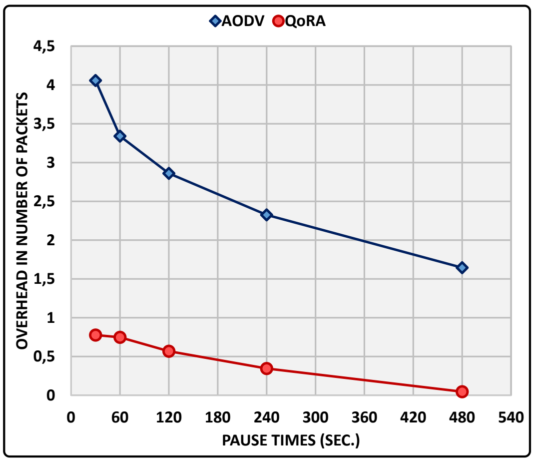
(B) Throughput



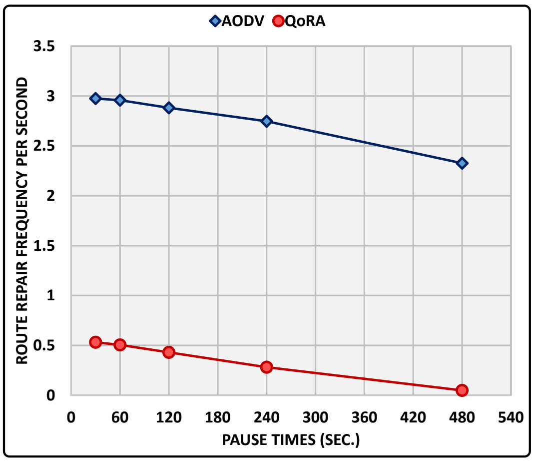
(C) Average end-to-end delay



(D) Jitter



(E) Overhead in number of packets



(F) Route repair frequency per second

Figure 6.7: The performance of QoRA using different pause times

6.2.6 Test Scenario 5: Varying the speed of the mobile nodes

In this experimental setup, we compare the performance of the proposed routing protocol for MANETs and VANETs scenarios [130]. For MANETs, the network consists of 100 nodes with RWP mobility model, while for VANETs we have 200 vehicles with 5×5 Manhattan Grid Mobility as shown in figure 6.8. The speed of the node is varied in both the cases from 5 m/s (18 Km/h) to 30 m/s (108 Km/h) in step 5 m/s. The variation in the maximum speed affects both these mobility model in the following ways. The higher speed means high mobility, which causes frequent topology changes and thus, making the scenario more difficult. We used 10 communication sessions to send voice traffic (G.729) at a Constant Bit Rate (CBR) of 8 kb/s and a packet size of 218 bytes using the ns-3 `OnOffApplication`. Additionally, we use Friis propagation loss model for small transmission range in MANETs scenario and for large transmission range in VANETs scenario we use Two Ray Ground propagation loss model. As explain in section 6.2.1 Friis perform well for small distance as compared to Two Ray Ground .

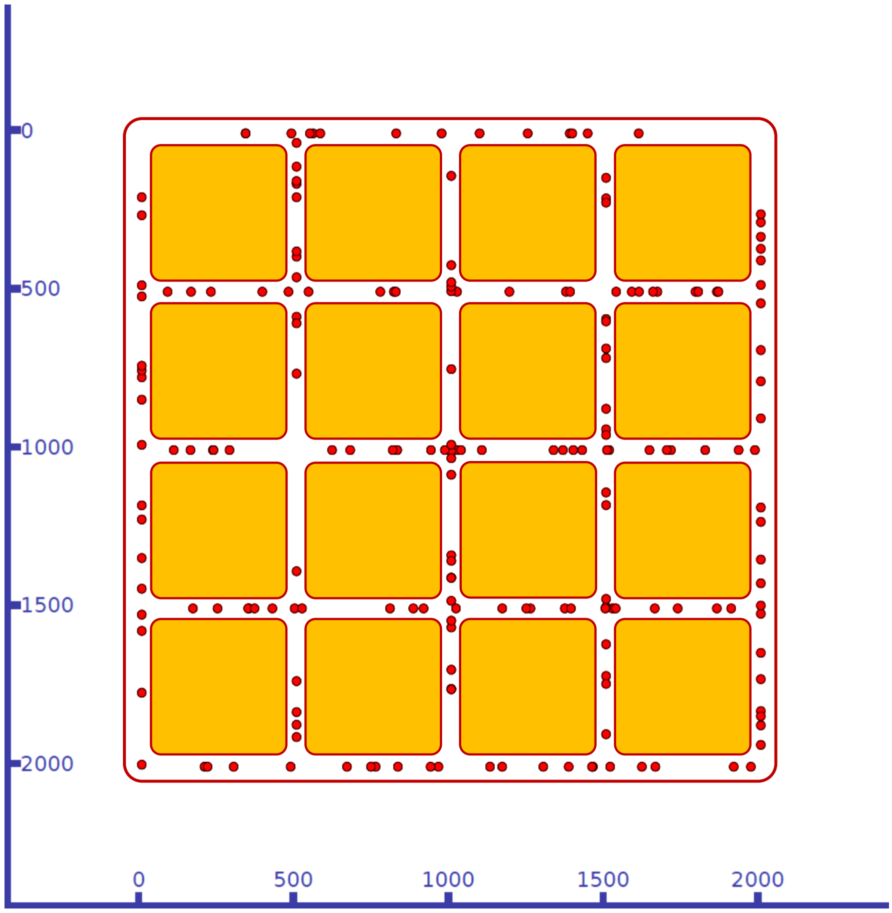


Figure 6.8: Examples of Manhattan Grid Mobility with 200 vehicles

Table 6.6 shows the parameters settings for MANETs and VANETs scenarios and the QoS constraints. We compared the results between AODV, CLWPR, and QoRA for both MANETs and VANETs scenarios.

TABLE 6.6: The parameters for the fifth scenario and QoS constraints

Parameter	MANETs scenario	VANETs scenario
Number of Nodes	100	200
Size of Area	1000×2000	2000×2000
MAC protocol	IEEE 802.11b	IEEE 802.11p
channel rates (Mbps)	1, 2, 5.5, 11	3, 4.5, 6, 9, 12, 18, 24, 27
Transmission Range	250 m	450 m
Propagation loss model	Friis	Two Ray Ground
Speed of the node	5 – 30 m/s	
Number of Flows	10	
Application data rate (CBR)	8 Kbps	
Packet size	218 bytes	
Bandwidth constraint B_{min}	12 Kbps	
Delay constraint D_{max}	150 – 250 ms	
Desired success rate S_{min}	97% – 100%	

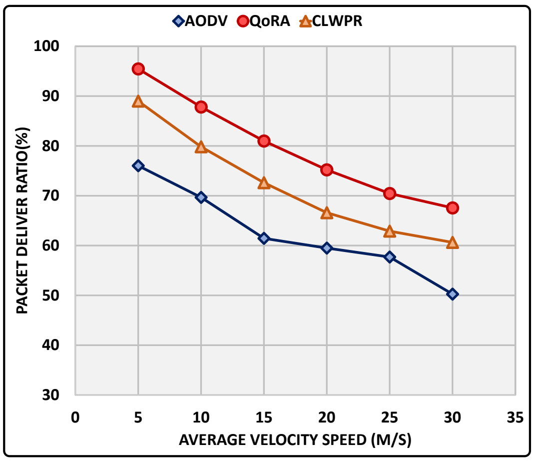
Figures 6.9a and 6.10a show the average packet delivery ratio for MANETs and VANETs scenarios. It is evident that the QoRA approach performs significantly better than the AODV and CLWPR protocols. This is because the QoRA approach provides multiple paths to the same destination, whereas AODV and CLWPR only provide single path resulting in frequent link failures due to dynamic topology changes. With the increase in the speed of the mobile nodes, the average packet delivery ratio reduces as expected, due to the frequent link failures. Additional, CLWPR also proposes the carry-n-forward mechanism as a link repair strategy. However, this method increases the average packet delivery ratio but it comes at the cost of high end-to-end delay and jitter, which is not suitable for QoS restricted real-time multimedia applications as explained in [73]. The CLWPR protocol is observed to perform well in high node-density scenarios. The protocol introduces network partitions due to the nodes mobility as the number of nodes decreases.

Figures 6.9b and 6.10b show the throughput observed in MANETs and VANETs respectively. It is clear from the graphs that the QoRA approach yields higher throughput as compared to the AODV and CLWPR protocols. It is due to the fact that the QoRA approach establishes multiple QoS-aware paths to the destination distribute the data packets over the multiple paths and improving the throughput. As the mobility increases, the throughput decreases due to frequent link failures.

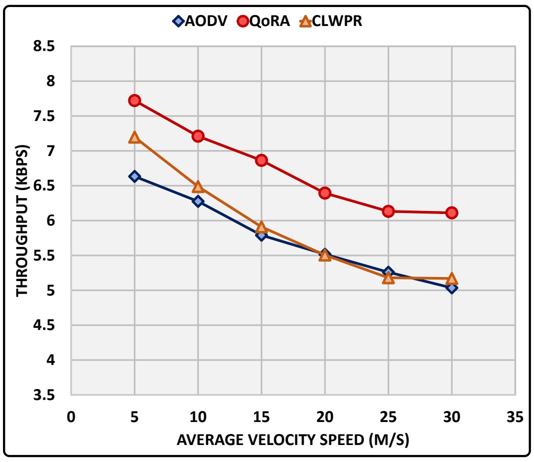
Figures 6.9c and 6.10c indicate that QoRA performs much better than AODV for end-to-end delays. However, CLWPR outperforms QoRA and AODV. Figures 6.9d and 6.10d show the similar curve for the jitter parameter. The QoRA approach again performs better than the AODV protocol but CLWPR shows the best performance. The reason that CLWPR outperforms QoRA for the end-to-end delay and jitter is due to the fact that CLWPR is a position based proactive routing protocol. So, CLWPR offers minimum delay and minimum jitter. As QoRA and AODV are on-demand routing protocols, the packets generated by the source node are stored in the queue until the route to the destination is established. The queuing of packets increases delay and jitter. Also, QoRA is a multipath routing protocol; so different paths contribute to the variable delay.

Figures 6.9e and 6.10e show the overhead in AODV is higher than in QoRA and CLWPR. The AODV protocol, as explained in the earlier section, is single path routing protocol and involves periodic exchange of Hello messages. At high mobility, CLWPR performs better than QoRA. It is again due to its proactive mechanism that uses the periodic hello message to adapt to the neighbor node activity. In QoRA, the overhead is low but with the increase in the mobility this value increase with the high occurrence of the link failure. Also, CLWPR uses an adaptive approach to keep the track of neighbor nodes. So, at high node speed time interval between the Hello messages is decreased that makes the curve for overhead to go up for higher velocity.

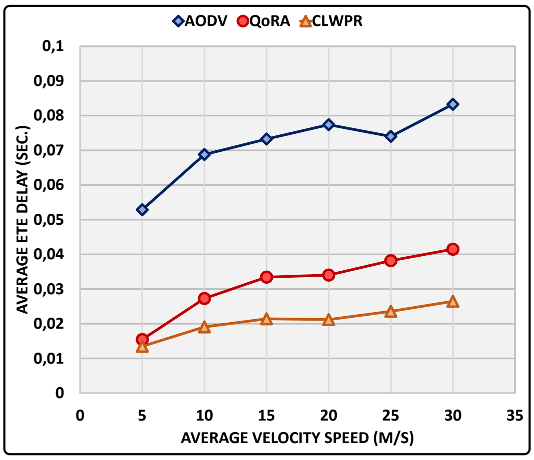
Figures 6.9f and 6.10f show the route discovery frequency curve. For QoRA, the curve shows rise with the increase in the node velocity due to highly dynamic topology. However, as show in figure 6.9f the curve for AODV initially decreases at high velocity, as AODV uses periodic Hello message that helps the nodes to find the destination directly. However, in CLWPR, the protocol does not have route discovery phase as it is a position based proactive mechanism for the route discovery.



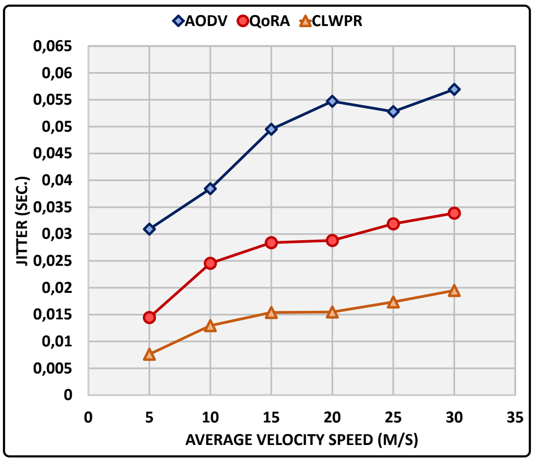
(a) Packet delivery ratio



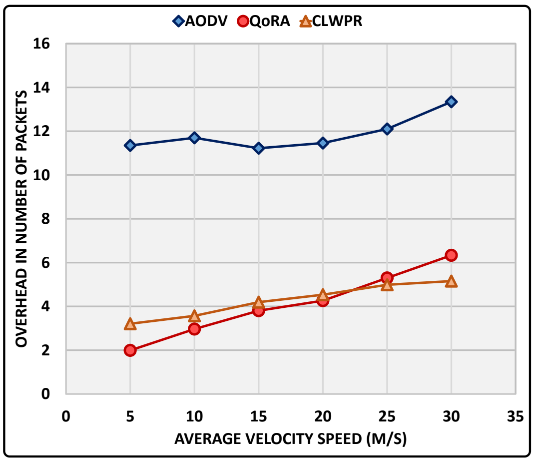
(b) Throughput



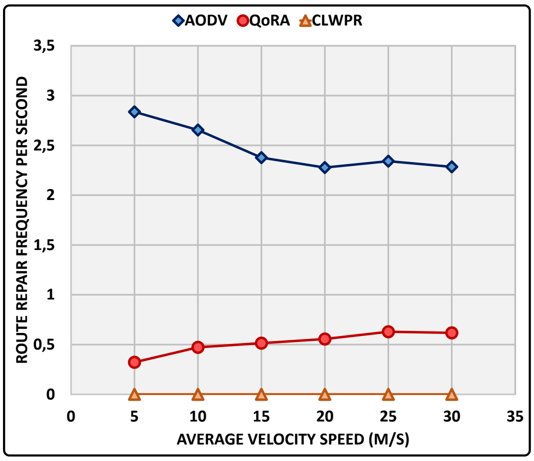
(c) Average end-to-end delay



(d) Jitter

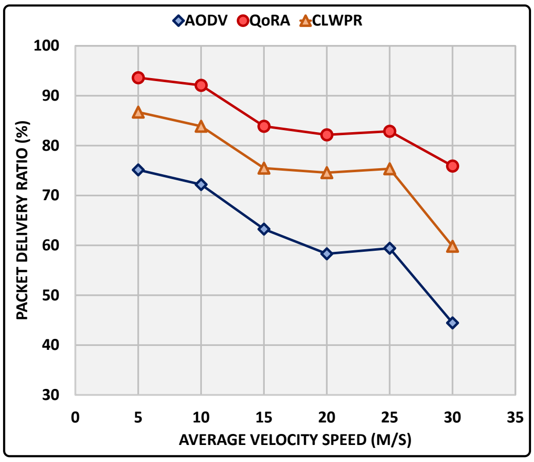


(e) Overhead in number of packets

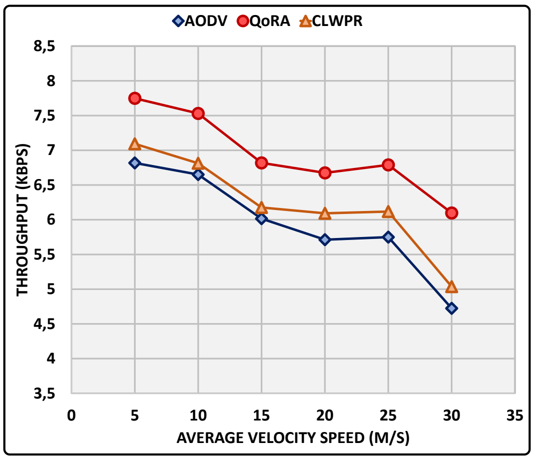


(f) Route repair frequency per second

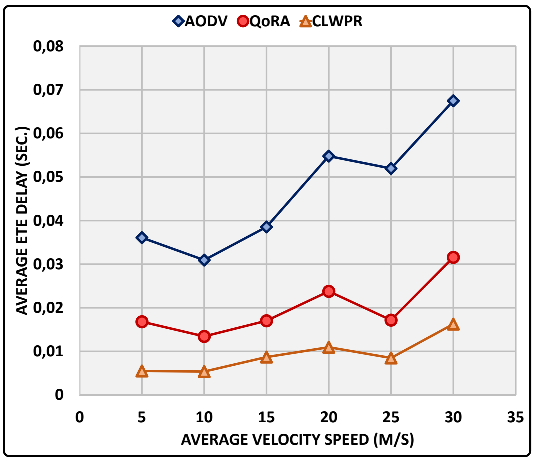
Figure 6.9: The performance of QoRA using different values for speed using RWP mobility



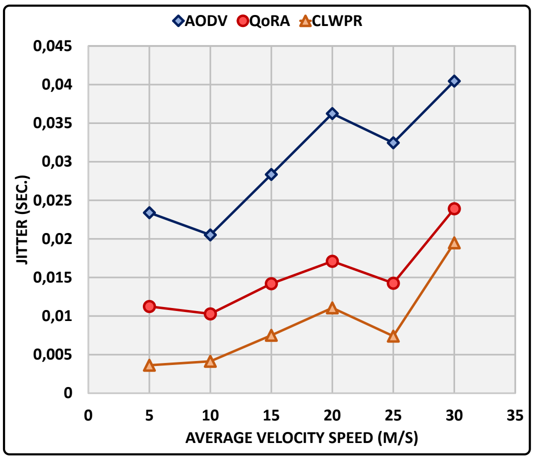
(A) Packet delivery ratio



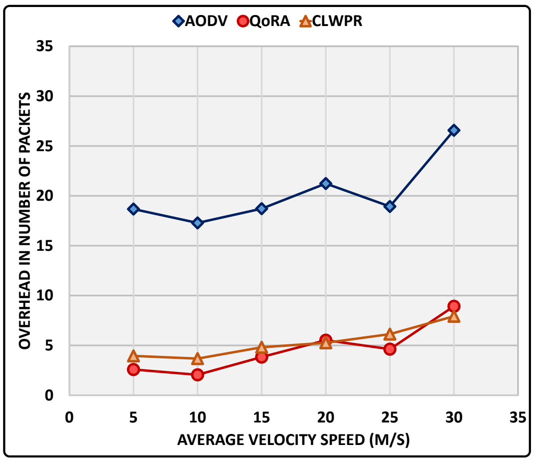
(B) Throughput



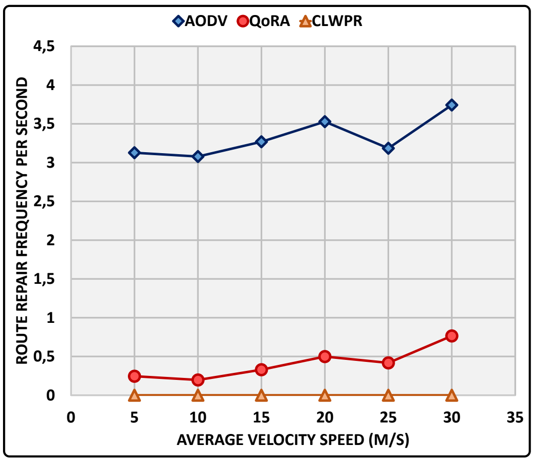
(C) Average end-to-end delay



(D) Jitter



(E) Overhead in number of packets



(F) Route repair frequency per second

Figure 6.10: The performance of QoRA using different values for speed using Bonn-motion mobility

6.2.7 Test Scenario 6: Video streaming in urban scenario

In the final scenario, we test the performance of QoRA for video transmission over highly realistic mobility setups. Here, we have used another simulator called SUMO for real-time road traffic scenario simulation. The speed of the cars or mobile nodes in SUMO are influenced by a wide range of real life parameters like lane speed, car following the traffic, lane changing, intersections, signals, etc. while in the previous scenarios the nodes are moving with a constant speed and do not support these parameters. Figure 6.11 shows the mobility network map for Ilmenau city. For the simulation, we choose 100 vehicles moving according to the traffic model in the area of size 1000×2000 m^2 .

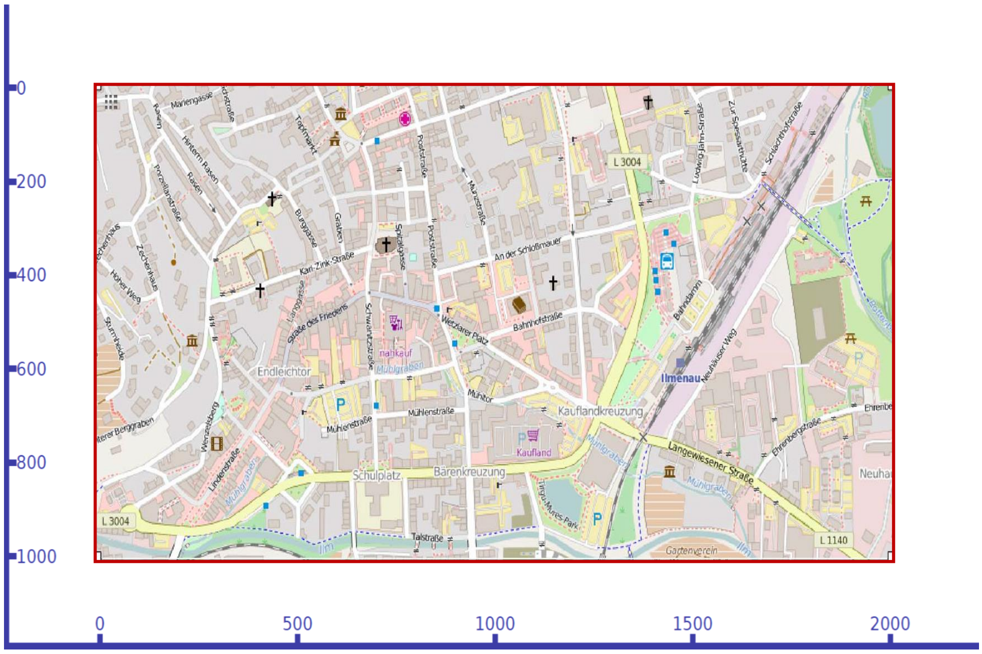


FIGURE 6.11: Ilmenau city mobility model

For the video application, we have used H.264/SVC (Scalable Video Coding) as it allows the scaling of the video transmissions so that the video can be delivered without any degradation in the quality between the devices. This property makes SVC an apt choice for MANETs/VANETs that are characterized by scarcity of resources like bandwidth, poor channel conditions, etc. Also, SVC benefits a large number of video applications like streaming, conferencing, broadcast, surveillance, etc. We choose between 2 to 12 video communication sessions varying in the step of 2 flows at a time to send video traffic that is modeled using video trace file [176]. In this way, we can stress the network or check the protocol with the real-time video data. The simulation results shown in the figure 6.12.

Table 6.7 presents the settings information of the network variables and QoS constraints.

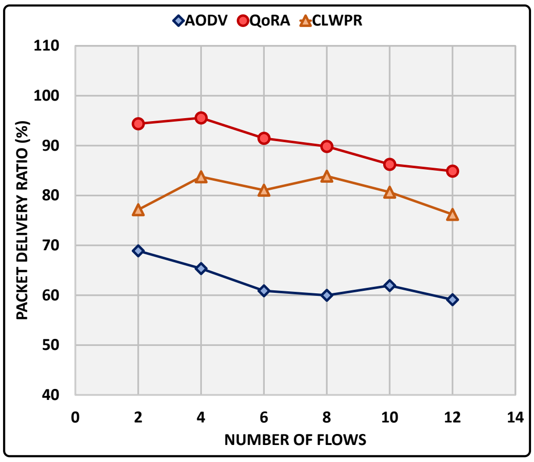
Table 6.7: The parameters for the sixth scenario and QoS constraints

Parameter	Value
Number of Nodes/Vehicles	100
Number of video source	2, 4, 6, 8, 10, 12
Video	Sony Demo CIF (352 × 288)
Minimum/Maximum frame Size	(22 – 66773) Bytes
Mean Frame Bit Rate	2.3 Mbps
Size of Area	$1000 \times 2000\ m^2$
Type of Mobility	Ilmenau city using SUMO
MAC protocol	IEEE 802.11p
channel rates (Mbps)	3, 4.5, 6, 9, 12, 18, 24, 27
Transmission Range	450 m
Propagation loss model	Two Ray Ground
Maximum speed of the nodes	27.78 m/s (100 km/h)
Pause time	0 s
Bandwidth constraint B_{min}	3 Mbps
Delay constraint D_{max}	150 – 250 ms
Desired success rate S_{min}	97% – 100%

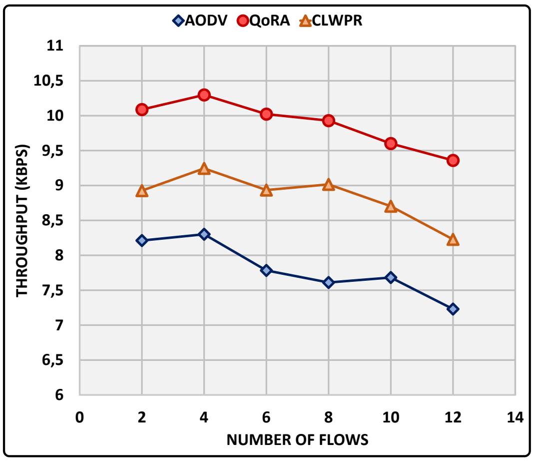
The figure 6.12a shows the performance curve for average packet delivery ratio. As the number of flows increases, the average packet delivery ratio decreases for all the three protocols. This is because the increase in the number of flows requires more route setup and lot of exchange of control information. QoRA outperforms CLWPR and AODV and provides average packet delivery ratio of about 85%. Similarly, the throughput curve as shown in figure 6.12b also sees declination with the increase in the number of communication flows.

For the average end-to-end delay and jitter as shown in figures 6.12c and 6.12d, the QoRA protocol shows the best performance. The increase in communication flows increases the load on the network. This means that the links become busy and longer queues at interfaces are observed causing higher delays. It is evident from the graph that AODV protocol performs better than the CLWPR protocol. In AODV, the RREQ packet contains "DestinationOnly" field, which is set to false by default. This means that any node having the route to the destination can reply with RREP after it has received the RREQ message. The source node then selects the first RREP message and discards all the later RREP messages. However, this approach increases the overhead with the increase in the number of flows. So, in order to avoid these overheads, we set the DestinationOnly field to true and now only destination is allowed to send one RREP message to the sender. This approach has allowed AODV to perform better than CLWPR in terms of delay and jitter.

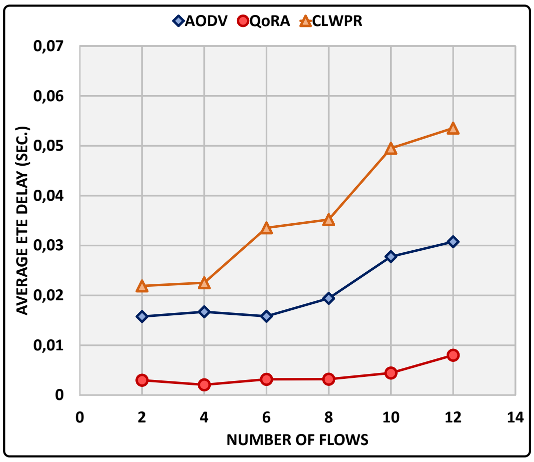
On the other hand, CLWPR routing protocol uses the adaptive method for sending



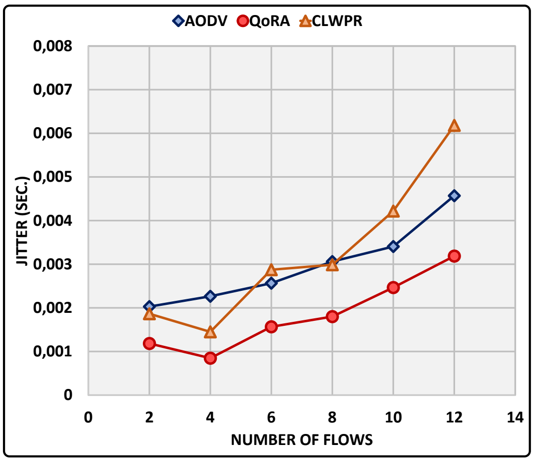
(A) Packet delivery ratio



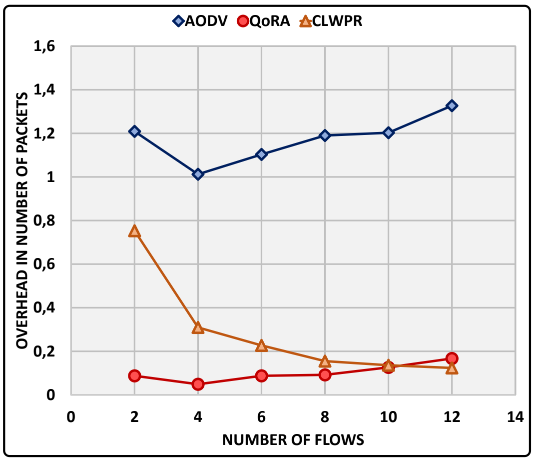
(B) Throughput



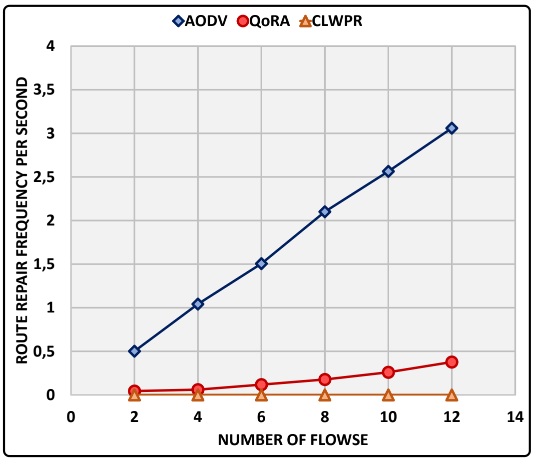
(C) Average end-to-end delay



(D) Jitter



(E) Overhead in number of packets



(F) Route repair frequency per second

Figure 6.12: The performance of QoRA using different number of video communication flows

Hello messages to keep track of the neighbor node. This means that for nodes moving at high speed the interval between periodic Hello message exchange is low while for slowly moving nodes it is high. This method increases delay and jitter, as the source nodes have to wait for the Hello message from the neighbor in the direction of the destination, before forwarding the data; thus, increasing the queue length at the interface. On the other hand this method provides a reduction of overhead. Also, CLWPR does not provide end–to-end QoS as compared to QoRA.

Figure 6.12e shows the curve for overhead in number of packets. For a small number of flows, AODV incurred much higher overhead than CLWPR and QoRA. However, for QoRA, the overhead is still low even at higher communication flows. This is because unlike AODV and CLWPR, QoRA does not employ proactive communication methodology like the use of periodic Hello messages, to keep track of its neighbors when the nodes do not have any data to forward.

Figure 6.12f shows the curve for routes discovery frequency. For large communication flows, the route frequency increases drastically for AODV. This is because the increase in the number of flows increases the probability of contention and collisions. Also, periodic usage of Hello message by AODV adds to the problem of contention and collision. Due to this AODV fails to deliver the data packet and assumes a link failure, which triggers the path reestablishment mechanism. For CLWPR, this value is observed to be 0. On the other hand, QoRA provides multiple routes and it is not affected by these problems, as it is evident from the figure.

6.3 Summary

In this chapter, we have provided an evaluation study of the QoRA routing protocol and studied its behavior over the common MANET scenarios. In all the test scenarios, we compared QoRA with AODV, which is considered as the benchmark routing protocol for MANETs. For VANETs environment we compared it with state-of-the-art position based CLWPR routing protocol. The results showed that QoRA has outperformed AODV and CLWPR over a wide range of scenarios. QoRA performed well in the tests with increasing levels of mobility, dealing with varying network load according to the number of communication flows and application data rates, scaling to networks of different sizes and managing scarce network conditions for optimal services. Finally, for high mobility VANET scenario, QoRA shows high reliability and adaptivity to the mobility.

Chapter 7

Conclusion and Future work

In this thesis, we addressed the challenges in implementing QoS-aware applications over MANETs. We proposed an adaptive QoS-aware multipath routing protocol that combines the monitoring capabilities of SNMP and the computation techniques from the field of swarm intelligence. We first introduced the concept of MANETs and different routing methodologies. Next, we reviewed the QoS routing in MANETs and the challenges, which hinder the implementation of real-time applications. Then we discussed the QoRA routing algorithm and examined its behavior and performance under different test scenarios. Finally, we give an overview of the contributions and results of this thesis, and then discuss the potential future research guidelines in this concluding chapter.

7.1 Conclusion

The main contribution of this thesis is the development of the QoRA routing protocol. QoRA has a completely reactive routing architecture and it establishes multiple paths for each communication session. A reactive route setup process is initiated at the start of each new communication session in order to obtain multiple paths for data transmission. The QoRA protocol utilizes the network monitoring and management functionalities of the SNMP. The SNMP is used to determine the QoS parameters locally without incurring any additional control messages on the network, which is the major highlight of our work. These locally determined QoS parameters are then collaborated with the population based global search meta-heuristic approaches having self-organizing characteristics. One such approach is the ACO, which efficiently explore the search place and provide optimal routing solutions needed for QoS-aware routing.

Also, the adaptive multi-rate mechanism in QoRA overcomes the problem of high end-to-end delays caused by the packet collision in the high-density environment and avoids transmission on the poor links in the low-density environment. In addition, the monitoring capabilities of SNMP are employed to tackle the congestion problems during data packet forwarding.

Later, we evaluated the performance and studied the behavior of the QoRA routing algorithm in a wide range of different test scenarios for MANETs. To quantify the results we evaluated the performance on the basis of six evaluation metrics that describes the efficiency and effectiveness of the routing protocol. For the sake of fair comparisons, in all the test scenarios we compared the QoRA with the AODV routing protocol, which is considered as the benchmark routing protocol for MANETs. For VANETs environment we compared it with state-of-the-art position based CLWPR routing protocol. We tested QoRA for different realistic scenarios and for multimedia transmission, where it outperforms the position based routing protocol CLWPR. CLWPR relies on the following two assumptions. Firstly, that every node is able to know its position that makes the use of expensive devices like GPS mandatory for this protocol. Secondly, the protocol requires that every node in the network knows the position of every other node/destination all the time. The QoRA protocol on the other hand as discussed, is a simple and effective QoS-aware routing mechanism that sidelines such assumptions for multimedia transmission. The results showed that for all the scenarios under test, the QoRA routing protocol outperforms the AODV and CLWPR routing protocol. QoRA showed high-performance results in terms of adapting to the increasing levels of mobility, dealing with the varying network load according to the number of communication flows and the application data rates, scaling to the networks with a large number of nodes and the optimal usage of the scarce network resources.

7.2 Future Research Directions

Here, we discuss the future research directions as the proposed extension work for the thesis. These includes the testing and deployment of the QoRA on the hardware testbeds, addition of security measures to secure the communication, the use of idea behind QoRA in other networks, the performance evaluation for other wide range of real-time applications like HD-video streaming, etc.

Since QoRA is highly adaptive to the dynamic topology, additional QoS parameters can be incorporated in QoRA approach. Also, a new model can be developed where different applications can be directly associated to different weight factors representing combination of various QoS parameters.

The deployment on the real network will bring up the new challenges that are unexplored in the simulation environment. For the optimal functioning in real implementations, further enhancements might be needed in the algorithm.

The other interesting point that could be explored in the future work is the implementation of QoRA over heterogeneous networks like Internet of Things.

From the security perspective, the security extension can be developed for QoRA. It is well-known fact that almost all the routing protocols in MANETs are developed without considering the security features. Some malicious nodes might break down the network and use the precious resources for self-benefit.

Also, as we developed the light version of SNMP protocol in ns-3 for QoRA approach and embed it to all participating nodes. It would be really interesting to implement the full version of SNMP protocol with all the features and techniques. This could lead to the new insights in the fields of MANETs.

Appendix A

Appendix A

Bibliography

[1] S. Basagni, M. Conti, S. Giordano, and I. Stojmenovic. *Mobile Ad Hoc Networking*. IEEE, John Wiley & Sons. Inc, San Francisco, CA, USA, 2004.

[2] P. Mohapatra and S. V. Krishnamurthy. *Ad Hoc Networks Technologies and Protocols*. Springer Science and Business Media, Inc., Boston, 2005.

[3] S. K. Sarkar, T. G. Basavaraju, and C. Puttamadappa. *Ad Hoc Mobile Wireless Networks Principles, Protocols, and Applications*. Taylor & Francis Group, New York: Auerbach, 2008.

[4] C. Prehofer and C. Bettstetter. Self-Organization in Communication Networks: Principles and Design Paradigms. *Communications Magazine, IEEE*, 43(7):78 – 85, July 2005. doi: 10.1109/MCOM.2005.1470824. URL http://dx.doi.org/10.1109/MCOM.2005.1470824.

[5] F. Dressler. Self-Organization in Ad Hoc Networks: Overview and Classification. Technical Report 7, University of Erlangen, Department of Computer, March 2006.

[6] F. Dressler. A study of self-organization mechanisms in ad hoc and sensor networks. *Computer Communications, Elsevier*, 31(13):3018–3029, August 2008. doi: 10.1016/j.comcom.2008.02.001. URL http://dx.doi.org/10.1016/j.comcom.2008.02.001.

[7] J.. Hoebeke, I. Moerman, B. Dhoedt, and P. Demeester. An Overview of Mobile Ad hoc Networks: Applications and Challenges. *JOURNAL OF THE COMMUNICATIONS NETWORK*, 3(3):60–66, 2004. URL http://hdl.handle.net/1854/LU-317876.

[8] A. Boukerche, B. Turgut, N. Aydin, M. Z. Ahmad, L. Bölöni, and D. Turgut. Routing protocols in ad hoc networks: A survey. *Computer Networks*, 55(13):3032–3080, 2011. doi: 10.1016/j.comnet.2011.05.010. URL http://dx.doi.org/10.1016/j.comnet.2011.05.010.

[9] F. D. da Cunha, A. Boukerche, L. Villas, V. Aline, and A. A. F. Loureiro. Data Communication in VANETs: A Survey, Challenges and Applications. Technical report, Research Report RR-8498, Version 2, INRIA, January 2014. URL https://hal.inria.fr/hal-00981126.

[10] ANSI/IEEE Std 802.11. Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications. *IEEE Standards 802.11*, June 1999. doi: 10.1109/IEEESTD.2003.95617. URL http://dx.doi.org/10.1109/IEEESTD.2003.95617.

[11] V. Bharghavan, A. Demers, S. Shenker, and L. Zhang. MACA: A Media Access Protocol for Wireless LAN's. *ACM SIGCOMM Computer Communication Review*, 24(4):212 – 225, Octoper 1994. doi: 10.1145/190809.190334. URL http://dx.doi.org/10.1145/190809.190334.

[12] F. A. Tobagi and L. Kleinrock. Packet Switching in Radio Channels, Part II: The Hidden Terminal Problem in Carrier Sense Multiple-Access Modes and the Busy-Tone Solution. *IEEE Transactions on Communications*, 23(12):1417 – 1433, December 1975. doi: 10.1109/TCOM.1975.1092767. URL http://dx.doi.org/10.1109/TCOM.1975.1092767.

[13] D. Shukla, L. Chandran-Wadia, and S. Iyer. Mitigating the Exposed Node Problem in IEEE 802.11 Ad Hoc Networks. In *Proceedings of the 12th International Conference on Computer Communications and Networks (ICCCN)*, pages 157 – 162, October 2003. doi: 10.1109/ICCCN.2003.1284164. URL http://dx.doi.org/10.1109/ICCCN.2003.1284164.

[14] G. Bianchi. Performance Analysis of the IEEE 802.11 Distributed Coordination Function. *IEEE Journal on Selected Areas in Communications*, 18(3):535 – 547, March 2000. doi: 10.1109/49.840210. URL http://dx.doi.org/10.1109/49.840210.

[15] M. M. Carvalho and J. J. G. Luna-Aceves. Delay analysis of IEEE 802.11 in single-hop networks. In *Proceedings of the 11th IEEE International Conference on Network Protocols*, pages 146–155, 2003. doi: 10.1109/ICNP.2003.1249764. URL http://dx.doi.org/10.1109/ICNP.2003.1249764.

[16] IEEE Std 802.11a. Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications: High-speed Physical Layer in the 5 GHz band. *Supplement to IEEE 802.11 Standard*, September 1999.

[17] IEEE Std 802.11b. Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications: High-speed Physical Layer Extension in the 2.4 GHz Band. *Supplement to IEEE 802.11 Standard*, September 1999.

[18] IEEE Std 802.11g. Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications: Further Higher Data Rate Extension in the 2.4 GHz Band. *Supplement to IEEE 802.11 Standard*, June 2003.

[19] IEEE Std 802.11p. Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications Amendment 6: Wireless Access in Vehicular Environments. *Supplement to IEEE 802.11 Standard*, July 2010.

[20] B. Awerbuch, D. Holmer, and H. Rubens. High Throughput Route Selection in Multi-Rate Ad Hoc Wireless Networks. In R. Battiti, M. Conti, and R. Cigno, editors, *Wireless On-Demand Network Systems*, volume 2928, pages 253–270. Springer Berlin Heidelberg, 2004. doi: 10.1007/978-3-540-24614-5_19. URL http://dx.doi.org/10.1007/978-3-540-24614-5_19.

[21] C. Campolo and A. Molinaro. Data Rate Selection in WBSS-Based IEEE 802.11p/WAVE Vehicular Ad Hoc Networks. In *Proceedings of the 7th International Symposium on Communication Systems Networks and Digital Signal Processing (CSNDSP)*, pages 412–416, July 2010.

[22] A. Kamerman and L. Monteban. WaveLAN®-II: A High-Performance Wireless LAN for the Unlicensed Band. *Bell Labs technical journal*, 2(3):118–133, 1997. doi: 10.1002/bltj.2069. URL http://dx.doi.org/10.1002/bltj.2069.

[23] G. Holland, N. Vaidya, and P. Bahl. A Rate-adaptive MAC Protocol for multi-Hop Wireless Networks. In *Proceedings of the 7th Annual International Conference on Mobile Computing and Networking*, pages 236–251, 2001. doi: 10.1145/381677.381700. URL http://doi.acm.org/10.1145/381677.381700.

[24] B. Sadeghi, V. Kanodia, A. Sabharwal, and E. Knightly. Opportunistic Media Access for Multirate Ad Hoc Networks. In *Proceedings of the 8th Annual International Conference on Mobile Computing and Networking*, pages 24–35, Atlanta, Georgia, USA, 2002. doi: 10.1145/570645.570650. URL http://doi.acm.org/10.1145/570645.570650.

[25] J. del Prado Pavon and S. Choi. Link Adaptation Strategy for IEEE 802.11 WLAN via Received Signal Strength Measurement. In *Proceedings of the IEEE International Conference on Communications (ICC'03)*, pages 1108–1113, 2003. doi: 10.1109/ICC.2003.1204534. URL http://dx.doi.org/10.1109/ICC.2003.1204534.

[26] M. Lacage, M. Hossein Manshaei, and T. Turletti. IEEE 802.11 Rate Adaptation: A Practical Approach. In *Proceedings of the 7th ACM international symposium on Modeling, analysis and simulation of wireless and mobile systems*, pages 126–134, Venice, Italy, 2004. doi: 10.1145/1023663.1023687. URL http://dl.acm.org/citation.cfm?id=1023687.

[27] D. Qiao and S. Choi. Fast-Responsive Link Adaptation for IEEE 802.11 WLANs. In *Proceedings of the IEEE International Conference on Communications (ICC'05)*, volume 5, pages 3583–3588, May 2005. doi: 10.1109/ICC.2005.1495085. URL http://dx.doi.org/10.1109/ICC.2005.1495085.

[28] J. Kim, S. Kim, S. Choi, and D. Qiao. CARA: Collision-Aware Rate Adaptation for IEEE 802.11 WLANs. In *Proceedings of the 25th IEEE International Conference on Computer Communications. Proceedings (INFOCOM)*, pages 1–11, Barcelona, Spain, April 2006. doi: 10.1109/INFOCOM.2006.316. URL http://dx.doi.org/10.1109/INFOCOM.2006.316.

[29] S. H.Y. Wong, H. Yang, S. Lu, and V. Bharghavan. Robust Rate Adaptation for 802.11 Wireless Networks. In *Proceedings of the 12th Annual International Conference on Mobile Computing and Networking (MobiCom'06)*, pages 146–157, Los Angeles, CA, USA, 2006. doi: 10.1145/1161089.1161107. URL http://doi.acm.org/10.1145/1161089.1161107.

[30] F. Maguolo, M. Lacage, and T. Turletti. Efficient Collision Detection for Auto Rate Fallback Algorithm. In *Proceedings of the IEEE Symposium on Computers and Communications (ISCC)*, pages 25–30, Marrakech, July 2008. doi: 10.1109/ISCC.2008.4625769. URL http://dx.doi.org/10.1109/ISCC.2008.4625769.

[31] C.-K. Toh. *Ad Hoc Mobile Wireless Networks: Protocols and Systems*. Prentice Hall PTR, 2002.

[32] M. K. Marina and S. R. Das. Routing in Mobile Adhoc Networks. In P. Mohapatra and S. V. Krishnamurthy, editors, *Ad Hoc Networkd*, chapter 3, pages 63–90. Springer US, 2005. doi: 10.1007/0-387-22690-7_3. URL http://dx.doi.org/10.1007/0-387-22690-7_3.

[33] M. Abolhasan, T. Wysocki, and E. Dutkiewicz. A review of routing protocols for mobile ad hoc networks. *Elsevier, Ad hoc networks*, 2(1):1 – 22, 2004. doi: 10.1016/S1570-8705(03)00043-X. URL http://dx.doi.org/10.1016/S1570-8705(03)00043-X.

[34] C. E. Perkins and P. Bhagwat. Highly Dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for Mobile Computers. In *Proceedings of the ACM*

SIGCOMM, pages 234 – 244, 1994. doi: 10.1145/190809.190336. URL http://dx.doi.org/10.1145/190809.190336.

[35] S. Murthy and J. J. Garcia-Luna-Aceves. An Efficient Routing Protocol for Wireless Networks. *Mobile Networks and Applications*, 1(2):183–197, 1996. doi: 10.1007/BF01193336. URL http://dx.doi.org/10.1007/BF01193336.

[36] T. Clausen and P. Jacquet. Optimized Link State Routing Protocol. Request for Comments RFC 3626, IETF Network Working Group, October 2003. URL http://dx.doi.org/10.17487/RFC3626.

[37] R. Ogier, M. Lewis, and F. Templin. Topology Dissemination Based on Reverse-Path Forwarding (TBRPF). Request for Comments RFC 3684, IETF Network Working Group, February 2004. URL http://dx.doi.org/10.17487/RFC3684.

[38] Q. Minhas, H. Mahmood, and H. Malik. The Role of Ad Hoc Networks in Mobile Telecommunication. In J. P. Maicas, editor, *Recent Developments in Mobile Communications - A Multidisciplinary Approach*, chapter 10, pages 179–200. InTech, 2011. doi: 10.5772/1368. URL http://dx.doi.org/10.5772/1368.

[39] F. Lee. Routing in Mobile Ad Hoc Networks. In X. Wang, editor, *Mobile Ad-Hoc Networks: Protocol Design*, chapter 16, pages 299–322. InTech, 2011. doi: 10.5772/548. URL http://www.intechopen.com/books/mobile-ad-hoc-networks-protocol-design.

[40] D. B. Johnson and D. A. Maltz. Dynamic Source Routing in Ad Hoc Wireless Networks. In T. Imielinski and H. F. Korth, editors, *Mobile computing*, volume 353, chapter 5, pages 153–181. Springe Addison-Wesley, 1996. doi: 10.1007/978-0-585-29603-6_5. URL http://dx.doi.org/10.1007/978-0-585-29603-6_5.

[41] D. B. Johnson, D. A. Maltz, Y. Hu, and J. G. Jetcheva. The Dynamic Source Routing Protocol (DSR) for Mobile Ad Hoc Networks for IPv4. Request for Comments RFC 4728, IETF Network Working Group, February 2007. URL http://dx.doi.org/10.17487/RFC4728.

[42] C. E. Perkins and E. M. Royer. Ad-hoc On-Demand Distance Vector Routing. In *Proceedings of the Second IEEE Workshop on Mobile Computing Systems and Applications (WMCSA)*, pages 90 – 100, Febryary 1999. doi: 10.1109/MCSA.1999.749281. URL http://dx.doi.org/10.1109/MCSA.1999.749281.

[43] C. E. Perkins, E. M. Belding-Royer, and S. R. Das. Ad hoc On-Demand Distance Vector (AODV) Routing. Request for Comments RFC 3561, IETF Network Working Group, July 2003. URL http://dx.doi.org/10.17487/RFC3561.

[44] V. D. Park and M. S. Corson. A Highly Adaptive Distributed Routing Algorithm for Mobile Wireless Networks. In *Proceedings of the Sixteenth Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*, volume 3, pages 1405 – 1413, April 1997. doi: 10.1109/INFCOM.1997.631180. URL http://dx.doi.org/10.1109/INFCOM.1997.631180.

[45] G. D. Caro, F. Ducatelle, and L. M. Gambardella. AntHocNet: an Ant-Based Hybrid Routing Algorithm for Mobile Ad Hoc Networks. In X. Yao, E. K. Burke, J. A. Lozano, J. Smith, J. J. Merelo-Guervós, J. A. Bullinaria, J. E. Rowe, P. Tiño, A. Kabán, and H. Schwefel, editors, *Parallel Problem Solving from Nature - PPSN VIII*, volume 3242 of *Lecture Notes in Computer Science*, pages 461–470. Springer Berlin Heidelberg, 2004. doi: 10.1007/978-3-540-30217-9_47. URL http://dx.doi.org/10.1007/978-3-540-30217-9_47.

[46] G. D. Caro, F. Ducatelle, and L. M. Gambardella. Anthocnet: an adaptive nature-inspired algorithm for routing in mobile ad hoc networks. *European Transactions on Telecommunications*, 16(5):443–455, 2004. doi: 10.1002/ett.1062. URL http://dx.doi.org/10.1002/ett.1062.

[47] M. R. Pearlman and Z. J. Haas. Determining the Optimal Configuration for the Zone Routing Protocol. *IEEE Journal on Selected Areas in Communications*, 17 (8):1395 – 1414, August 1999. doi: 10.1109/49.779922. URL http://dx.doi.org/10.1109/49.779922.

[48] Z. Haas and M. Pearlman. The Performance of Query Control Schemes for the Zone Routing Protocol. *IEEE/ACM Transactions on Networking*, 9(5):427 – 438, August 2001. doi: 10.1109/90.944341. URL http://dx.doi.org/10.1109/90.944341.

[49] C. A. Santiváñez, R. Ramanathan, and I. Stavrakakis. Making Link-State Routing Scale for Ad Hoc Networks. In *Proceedings of the 2nd ACM international symposium on Mobile ad hoc networking & computing*, pages 22–32, Athens, Greece, 2001. doi: 10.1145/501417.501420. URL http://dx.doi.org/10.1145/501417.501420.

[50] E. Kaplan and C. Hegarty. *Understanding GPS: Principles and Applications*. Artech house, 2005.

[51] Y. Ko and N. H. Vaidya. Location-Aided Routing (LAR) in mobile ad hoc networks. *Springer Wireless Networks*, 6(4):307–321, 2000. doi: 10.1023/A:1019106118419. URL http://dx.doi.org/10.1023/A:1019106118419.

[52] S. Basagni, I. Chlamtac, V. R. Syrotiuk, and B. A. Woodward. A Distance Routing Effect Algorithm for Mobility (DREAM). In *Proceedings of the IEEE/ACM MobiCom,*, pages 76–84, 1998. doi: 10.1145/288235.288254. URL http://dx.doi.org/10.1145/288235.288254.

[53] B. Karp and H. T. Kung. GPSR: Greedy Perimeter Stateless Routing for Wireless Networks. In *Proceedings of the IEEE/ACM MobiCom*, pages 243–254, 2000. doi: 10.1145/345910.345953. URL http://dx.doi.org/10.1145/345910.345953.

[54] R. C. Shah, S. Wiethölter, A. Wolisz, and J. M. Rabaey. Modeling and analysis of opportunistic routing in low traffic scenarios. In *Proceedings of the 3rd International Symposium on Modeling and Optimization in Mobile Ad Hoc and Wireless Networks (WiOpt0́5)*, pages 294–304, Trento, Italy, April 2005. doi: 10.1109/WIOPT.2005.30. URL http://dx.doi.org/10.1109/WIOPT.2005.30.

[55] M. Ali, B. G. Stewart, A. Shahrabi, and A. Vallavaraj. QoS Aware Multipath Threshold Routing for Mobile Ad hoc Networks. *International Journal of Applied Information Systems (IJAIS)*, 7(1):8 – 15, April 2014. doi: 10.5120/ijais14-451118. URL http://dx.doi.org/10.5120/ijais14-451118.

[56] S. Mueller, R. P. Tsang, and D. Ghosal. Multipath Routing in Mobile Ad Hoc Networks: Issues and Challenges. In G. Goos, J. Hartmanis, and J. van Leeuwen, editors, *Performance Tools and Applications to Networked Systems*, chapter 10, pages 209–234. Springer-Verlag Berlin Heidelberg, 2004. doi: 10.1007/b97129. URL http://dx.doi.org/10.1007/b97129.

[57] Z. Ye, S.V. Krishnamurthy, and S. K. Tripathi. A Framework for Reliable Routing in Mobile Ad Hoc Networks. In *Proceedings of the Twenty-Second Annual Joint Conference of the IEEE Computer and Communications (INFOCOM)*, volume 1, pages 270–280, March 2003. doi: 10.1109/INFCOM.2003.1208679. URL http://dx.doi.org/10.1109/INFCOM.2003.1208679.

[58] S.-J. Lee and M. Gerla. Split Multipath Routing with Maximally Disjoint Paths in Ad hoc Networks. In *Proceedings of the IEEE International Conference on Communications*, volume 10, pages 3201–3205, 2001. doi: 10.1109/ICC.2001.937262. URL http://dx.doi.org/10.1109/ICC.2001.937262.

[59] M. K. Marina and S.R.: S. R. Das. On-demand Multipath Distance Vector Routing in Ad Hoc Networks. In *Proceedings of the International Conference on Network Protocols*, pages 14–23, 2001. doi: 10.1109/ICNP.2001.992756. URL http://dx.doi.org/10.1109/ICNP.2001.992756.

[60] P. Sinha. QoS Issues in Ad-hoc Network. In P. Mohapatra and S. V. Krishnamurthy, editors, *Ad Hoc Networkd*, chapter 8, pages 229 – 247. Springer US, 2005. doi: 10.1007/0-387-22690-7_8. URL http://dx.doi.org/10.1007/0-387-22690-7_8.

[61] E. Crawley, R. Nair, B. Rajagopalan, and H. Sandick. A Framework for QoS-based Routing in the Internet. IETF Network Working Group RFC2386, IETF Network Working Group, August 1998. URL http://dx.doi.org/10.17487/RFC2386.

[62] T. B. Reddy, I. Karthigeyan, B. ManojS. Manoj, and C. S. R. Murthy. Quality of service provisioning in ad hoc wireless networks: a survey of issues and solutions. *Ad Hoc Networks*, 4(1):83–124, 2006. doi: 10.1016/j.adhoc.2004.04.008. URL http://dx.doi.org/10.1016/j.adhoc.2004.04.008.

[63] Z. Wang and J. Crowcroft. Quality-of-service routing for supporting multimedia applications. *IEEE Journal on Selected Areas in Communications*, 14(7):1228–1234, 1996. doi: 10.1109/49.536364. URL http://dx.doi.org/10.1109/49.536364.

[64] A.M. Abbas and O. Kure. Quality of Service in mobile ad hoc networks: a survey. *International Journal of Ad Hoc and Ubiquitous Computing*, 6(2):75–98, 2010. doi: 10.1504/IJAHUC.2010.034322. URL http://dx.doi.org/10.1504/IJAHUC.2010.034322.

[65] L. Hanzo and R. Tafazolli. A Survey of QoS Routing Solutions for Mobile Ad Hoc Networks. *IEEE Communications Surveys & Tutorials*, 9(2):50 – 70, 2007. doi: 10.1109/COMST.2007.382407. URL http://dx.doi.org/10.1109/COMST.2007.382407.

[66] N. Zhang and A. Anpalagan. Comparative review of QoS-aware on-demand routing in ad hoc wireless networks. *Wireless Sensor Network*, 2(4):274–284, 2010. doi: 10.4236/wsn.2010.24038. URL http://dx.doi.org/10.4236/wsn.2010.24038.

[67] M. Lindeberg, S. Kristiansen, T. Plagemann, and V. Goebel. Challenges and techniques for video streaming over mobile ad hoc networks. *Multimedia Systems*,

17(1):51 – 82, 2011. doi: 10.1007/s00530-010-0187-8. URL http://dx.doi.org/10.1007/s00530-010-0187-8.

[68] M. K. Gulati and K. Kumar. Qos routing protocols for mobile ad hoc networks: a survey. *International Journal of Wireless and Mobile Computing*, 5(2):107–118, 2012. doi: 10.1504/IJWMC.2012.046783. URL http://dx.doi.org/10.1504/IJWMC.2012.046783.

[69] B. S. Pradeep and S. Soumya. A new method for load balancing and qos in on demand protocols in the manet's perspective. *International Journal of Advanced Networking and Applications*, 1(4):275–281, 2010. URL http://www.ijana.in/papers/4.11.pdf.

[70] N. Sarma and S. Nandi. Route Stability Based QoS Routing in Mobile Ad Hoc Networks. *Wireless Personal Communications*, 54(1):203–224, 2010. doi: 10.1007/s11277-009-9718-z. URL http://dx.doi.org/10.1007/s11277-009-9718-z.

[71] M. Balachandraand, K.V. Prema, and K. Makkithaya. Multiconstrained and multipath QoS aware routing protocol for MANETs. *Wireless Networks*, 20(8):2395–2408, 2014. doi: 10.1007/s11276-014-0754-6. URL http://dx.doi.org/10.1007/s11276-014-0754-6.

[72] S. Palaniappan and K. Chellan. Energy-efficient stable routing using QoS monitoring agents in MANET. *EURASIP Journal on Wireless Communications and Networking*, 2015(1):1–11, December 2015. doi: 10.1186/s13638-014-0234-9. URL http://dx.doi.org/10.1186/s13638-014-0234-9.

[73] K. Katsaros, M. Dianati, R. Tafazolli, and R. Kernchen. CLWPR - A Novel Cross-Layer Optimized Position Based Routing Protocol for VANETs. In *Proceedings of the IEEE Vehicular Networking Conference (VNC)*, pages 139–146, Amsterdam, Netherlands, November 2011. doi: 10.1109/VNC.2011.6117135. URL http://dx.doi.org/10.1109/VNC.2011.6117135.

[74] K. Katsaros, M. Dianati, and R. Kernchen. A Position-based Routing Module for Simulation of VANETs in Ns-3. In *Proceedings of the 5th International Conference on Simulation Tools and Techniques (ICST)*, pages 345–352, Brussels, Belgium, 2012. URL http://dl.acm.org/citation.cfm?id=2263019.2263074.

[75] M. Obaidat, M. A. Ali, I. Shahwan, M. S. Obaidat, and S. Obeidat. QoS-aware multipath routing protocol for delay sensitive applications in MANETs a cross-layer approach. In *Proceedings of the International Conference on Wireless Information Networks and Systems (WINSYS)*, pages 41–46, July 2011.

[76] M. Obaidat, M. Ali, I. Shahwan, M. S. Obaidat, and S. Obeidat. Qos-aware multipath communications over manets. *Journal of Networks*, 8(1):26–36, 2013. doi: 10.4304/jnw.8.1.26-36. URL http://dx.doi.org/10.4304/jnw.8.1.26-36.

[77] S. Chen and K. Nahrstedt. Distributed quality-of-service routing in ad hoc networks. *IEEE Journal on Selected Areas in Communications*, 17(8):1488–1505, August 1999. doi: 10.1109/49.780354. URL http://dx.doi.org/10.1109/49.780354.

[78] W. YANG, X.YANG, G. LIU, and C. DONG. A bandwidth aware multipath routing protocol in mobile ad hoc networks. *Journal of Computational Information Systems*, 7(3):685–696, 2011. URL http://www.jofcis.com/publishedpapers/2011_7_3_685_696.pdf.

[79] Y. Yang and R. Kravets. Contention-aware admission control for ad hoc networks. *IEEE Transactions on Mobile Computing*, 4(4):363–377, July 2005. doi: 10.1109/TMC.2005.52. URL http://dx.doi.org/10.1109/TMC.2005.52.

[80] O. C. Kwon, H. R. Oh, H. Song, and G. C. Lee. QoS routing algorithm consuming minimum overall timeslots for video streaming over MANET. In *11th International Symposium on Communications and Information Technologies (ISCIT)*, pages 256–261, 2011. doi: 10.1109/ISCIT.2011.6089744. URL http://dx.doi.org/10.1109/ISCIT.2011.6089744.

[81] F. Qin and Y. Liu. Multipath routing formobile ad hoc network. In *Proceedings of the International Symposium on Information Processing (ISIP)*, 2009. URL http://www.academypublisher.com/proc/isip09/papers/isip09p237.pdf.

[82] X. Gao, X. Zhang, D. Shi, F. Zou, and W. Zhu. Contention and Queue-aware routing protocol for mobile ad hoc networks. In *Proceedings of the International Conference on Wireless Communications, Networking and Mobile Computing (WiCom)*, pages 1628–1631, 2007. doi: 10.1109/WICOM.2007.410. URL http://dx.doi.org/10.1109/WICOM.2007.410.

[83] S. Sheu and J. Chen. A novel delay-oriented shortest path routing protocol for mobile ad hoc networks. In *Proceedings of the IEEE International Conference on Communications (ICC)*, volume 6, pages 1930–1934, 2001. doi: 10.1109/ICC.2001.937126. URL http://dx.doi.org/10.1109/ICC.2001.937126.

[84] L. Reddeppa Reddy and S.V. Raghavan. SMORT: Scalable multipath on-demand routing for mobile ad hoc networks. *Ad Hoc Networks*, 5(2):162–188, 2007. doi:

10.1016/j.adhoc.2005.10.002. URL http://dx.doi.org/10.1016/j.adhoc.2005.10.002.

[85] C. Lal, V. Laxmi, and M. S. Gaur. QoS-aware Routing for Transmission of H.264/SVC Encoded Video Traffic over MANETs. In *Proceedings of the 19th Asia-Pacific Conference on Communications (APCC)*, pages 104 –109, August 2013. doi: 10.1109/APCC.2013.6765924. URL http://dx.doi.org/10.1109/APCC.2013.6765924.

[86] C. Lal, V. Laxmi, M. S. Gaur, and Seok-Bum Ko. Bandwidth-aware routing and admission control for efficient video streaming over manets. *Wireless Networks*, 21 (1):95–114, 2015. doi: 10.1007/s11276-014-0774-2. URL http://dx.doi.org/10.1007/s11276-014-0774-2.

[87] E. Bonabeau, M. Dorigo, and G. Theraulaz. *Swarm intelligence: from natural to artificial systems*. Oxford University Press, 1999.

[88] F. Dressler and O. B. Akan. A survey on bio-inspired networking. *Computer Networks*, 54(6):881–900, April 2010. doi: 10.1016/j.comnet.2009.10.024. URL http://dx.doi.org/10.1016/j.comnet.2009.10.024.

[89] M. Dorigo and T. Stützle. *Ant Colony Optimization*. MIT Press, 2004.

[90] M. Günes, U. Sorges, and I. Bouazizi. ARA-the ant-colony based routing algorithm for MANETs. In *Proceedings of the ICPP Workshop on Ad Hoc Networks (IWAHN)*, pages 79–85, 2002. doi: 10.1109/ICPPW.2002.1039715. URL http://dx.doi.org/10.1109/ICPPW.2002.1039715.

[91] R. Schoonderwoerd, O. Holland, J. Bruten, and L. J. Rothkrantz. Ant-based load balancing in telecommunications networks. *Adaptive behavior*, 5(2):169–207, 1997. doi: 10.1177/105971239700500203. URL http://dx.doi.org/10.1177/105971239700500203.

[92] G. D. Caro and M. Dorigo. Antnet: Distributed stigmergetic control for communications networks. *Journal of Artificial Intelligence Research (JAIR)*, 9:317–365, 1998. URL http://dl.acm.org/citation.cfm?id=1622797.1622806.

[93] O. Hussein and T. Saadawi. Ant routing algorithm for mobile ad-hoc networks (ARAMA). In *Proceedings of the IEEE Performance Computing and Communications Conference (IPCCC)*, pages 281–290, 2003. doi: 10.1109/PCCC.2003.1203709. URL http://dx.doi.org/10.1109/PCCC.2003.1203709.

[94] J. S. Baras and H. Mehta. A probabilistic emergent routing algorithm for mobile ad hoc networks. In *Proceedings of the WiOpt03: Modeling and Optimization*

in Mobile Ad Hoc and Wireless Networks, Sophia Antipolis, France, 2003. URL https://hal.inria.fr/inria-00466600/document.

[95] M. Roth and S. Wicker. Termite: ad-hoc networking with stigmergy. In *Proceedings of the IEEE Global Telecommunications Conference (GLOBECOM)*, volume 5, pages 2937–2941, December 2003. doi: 10.1109/GLOCOM.2003.1258772. URL http://dx.doi.org/10.1109/GLOCOM.2003.1258772.

[96] L. Rosati, M. Berioli, and G. Reali. On ant routing algorithms in ad hoc networks with critical connectivity. *Ad Hoc Networks*, 6(6):827–859, 2008. doi: 10.1016/j.adhoc.2007.07.003. URL http://dx.doi.org/10.1016/j.adhoc.2007.07.003.

[97] E. Osagie, P. Thulasiraman, and R. K. Thulasiram. Paconet: improved ant colony optimization routing algorithm for mobile ad hoc networks. In *Proceedings of 22nd International Conference on Advanced Information Networking and Applications (AINA)*, pages 204–211, March 2008. doi: 10.1109/AINA.2008.77. URL http://dx.doi.org/10.1109/AINA.2008.77.

[98] M. Günes, M. Kähmer, and I. Bouazizi. Ant-routing-algorithm (ARA) for mobile multi-hop ad-hoc networks-new features and results. In *Proceedings of the Second Mediterranean Workshop on Ad-Hoc Networks*, 2003.

[99] Z. Liu, M. Z. Kwiatkowska, and C. Constantinou. A swarm intelligence routing algorithm for MANETs. In *Proceedings of the International Conference on Communications Internet and Information Technology (IASTED)*, pages 484–489, November 2004. URL http://qav.comlab.ox.ac.uk/bibitem.php?key=LKC04.

[100] S. Kamali and J. Opatrny. POSANT: A Position Based Ant Colony Routing Algorithm for Mobile Ad-hoc Networks. In *Proceedings of the Third International Conference on Wireless and Mobile Communications (ICWMC)*, pages 21–21, 2007. doi: 10.1109/ICWMC.2007.68. URL http://dx.doi.org/10.1109/ICWMC.2007.68.

[101] B. R. Sujatha and M. V. Sathyanarayana. PBANT-Optimized ANT Colony Routing Algorithm for MANETs. *Global Journal of Computer Science and Technology*, 10(3):29–34, 2010. URL http://computerresearch.org/index.php/computer/article/view/910.

[102] A. Gupta, H. Sadawarti, and A. K. Verma. Performance Enhancement of DYMO Routing Protocol with Ant Colony Optimization. *International Journal of Electronics and Electrical Engineering*, 2(3):188–194, 2014. doi: 10.12720/ijeee.2.3.188-194. URL http://dx.doi.org/10.12720/ijeee.2.3.188-194.

[103] S. Marwaha, C. K. Tham, and D. Srinivasan. Mobile agents based routing protocol for mobile ad hoc networks. In *Proceedings of the IEEE Global Telecommunications Conference (GLOBECOM)*, pages 163–167, 2002. doi: 10.1109/GLOCOM.2002.1188062. URL http://dx.doi.org/10.1109/GLOCOM.2002.1188062.

[104] W. Yu, G. Zuo, and Q. Li. Ant colony optimization for routing in mobile ad hoc networks. In *Proceedings of the International Conference on Machine Learning and Cybernetics*, volume 2, pages 1147–1151, July 2008. doi: 10.1109/ICMLC.2008.4620576. URL http://dx.doi.org/10.1109/ICMLC.2008.4620576.

[105] J. A. P. Martins, S. L. O. B. Correia, and J. Celestino. Ant-DYMO: A bio-inspired algorithm for MANETs. In *Proceedings of the IEEE 17th International Conference on Telecommunications (ICT)*, pages 748–754, April 2010. doi: 10.1109/ICTEL.2010.5478808. URL http://dx.doi.org/10.1109/ICTEL.2010.5478808.

[106] L. J. G. Villalba, D. R. Cañas, and A. L. S. Orozco. Bio-inspired routing protocol for mobile ad hoc networks. *IET communications*, 4(18):2187–2195, 2010. doi: 10.1049/iet-com.2009.0826. URL http://dx.doi.org/10.1049/iet-com.2009.0826.

[107] C. Raval, S. Hegde, and M. P. Tahiliani. Ant-CAMP: Ant Based Congestion Adaptive Multi-Path Routing Protocol for Wireless Networks. In *Proceedings of the International Conference on Emerging Trends in Networks and Computer Communications (ETNCC)*, pages 463–468, Udaipur, Rajasthan, India, April 2011. doi: 10.1109/ETNCC.2011.6255882. URL http://dx.doi.org/10.1109/ETNCC.2011.6255882.

[108] S. Samadi and H. Beigy. An Adaptive Multipath Ant Routing algorithm for Mobile Ad Hoc Networks. *International Journal of Computer and Electrical Engineering*, 4(2):175–180, 2012. doi: 10.7763/IJCEE.2012.V4.474. URL http://dx.doi.org/10.7763/IJCEE.2012.V4.474.

[109] C. Kanani and A. Sinhal. Ant Colony Optimization based Modified AOMDV for Multipath Routing in MANET. *International Journal of Computer Applications*, 28(10):14–19, 2013. doi: 10.5120/14151-2320. URL http://dx.doi.org/10.5120/14151-2320.

[110] D. R. Cañas, L. J. G. Villalba, A. L. S. Orozco, and T. Kim. Adaptive routing protocol for mobile ad hoc networks. *Computing*, 96(9):817–827, 2014. doi: 10.1007/s00607-013-0310-8. URL http://dx.doi.org/10.1007/s00607-013-0310-8.

[111] J. Wang, E. Osagie, P. Thulasiraman, and R. K. Thulasiram. HOPNET: A hybrid ant colony optimization routing algorithm for mobile ad hoc network. *Ad Hoc Networks*, 7(4):29–34, 2009. doi: 10.1016/j.adhoc.2008.06.001. URL http://dx.doi.org/10.1016/j.adhoc.2008.06.001.

[112] K. M. Sim and W. H. Sun. Ant colony optimization for routing and load-balancing: survey and new directions. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, 33(5):560–572, 2003. doi: 10.1109/TSMCA.2003.817391. URL http://dx.doi.org/10.1109/TSMCA.2003.817391.

[113] S. Bitam, A. Mellouk, and S. Zeadally. Bio-inspired routing algorithms survey for vehicular ad-hoc networks. *Communications Surveys Tutorials, IEEE*, 17(2):843–867, 2015. doi: 10.1109/COMST.2014.2371828. URL http://dx.doi.org/10.1109/COMST.2014.2371828.

[114] D. S. J. De Couton, D. Aguayo, B. A. Chambers, and R. Morris. Performance of multihop wireless networks: Shortest path is not enough. *ACM SIGCOMM Computer Communication Review*, 33(1):83–88, 2003. doi: 10.1145/774763.774776. URL http://doi.acm.org/10.1145/774763.774776.

[115] R. Draves, J. Padhye, and B. Zill. Comparison of routing metrics for static multi-hop wireless networks. *ACM SIGCOMM Computer Communication Review*, 34(4):133–144, 2004. doi: 10.1145/1030194.1015483. URL http://dx.doi.org/10.1145/1030194.1015483.

[116] L. Liu and F. Guangzeng. A novel ant colony based QoS-aware routing algorithm for MANETs. In L. Wang, K. Chen, and Y. Ong, editors, *Advances in Natural Computation*, volume 3612 of *Lecture Notes in Computer Science*, pages 457–466. Springer Berlin Heidelberg, 2005. doi: 10.1007/11539902_56. URL http://dx.doi.org/10.1007/11539902_56.

[117] X. Wang, Z. Yong-Zhao, W. Liang-Min, and L. Jiang. Ant Colony Optimization and Ad-hoc On-demand Multipath Distance Vector (AOMDV) Based Routing Protocol. In *Proceedings of the Fourth International Conference on Natural Computation (ICNC)*, volume 7, pages 589–593, Jinan, Shandong, China, October 2008. doi: 10.1109/ICNC.2008.656. URL http://dx.doi.org/10.1109/ICNC.2008.656.

[118] Z. Liu, M.Z. Kwiatkowska, and C. Constantinou. A Biologically Inspired QoS Routing Algorithm for Mobile Ad Hoc Networks. In *Proceedings of the 19th International Conference on dvanced Information Networking and Applications (AINA)*, pages 426–431, 2005. doi: 10.1109/AINA.2005.13. URL http://dx.doi.org/10.1109/AINA.2005.13.

[119] Z. Liu, M. Z. Kwiatkowska, and C. Constantinou. A Biologically Inspired QoS Routing Algorithm for Mobile Ad Hoc Networks. *International Journal of Wireless and Mobile Computing*, 4(2):64–75, 2010. doi: 10.1504/IJWMC.2010.033054. URL http://dx.doi.org/10.1504/IJWMC.2010.033054.

[120] M. Belkadi, M. Lalam, A. M'zoughi, N. Tamani, M. Daoui, and R. Aoudjit. Intelligent Routing and Flow Control in MANETs. *Journal of Computing and Information Technolog-CIT*, 18(3):233–243, 2010. doi: 10.2498/cit.1001470. URL http://dx.doi.org/10.2498/cit.1001470.

[121] H. Badis, A. Munaretto, K. Al Anelise, and G. Pujolle. QoS for ad hoc networking based on multiple metrics: bandwidth and delay. In *Proceedings of the 5th IEEE International Conference on Mobile and Wireless Communications Networks*, pages 15–18, 2003.

[122] P. Deepalakshmi and S. Radhakrishnan. An ant colony-based multi objective quality of service routing for mobile ad hoc networks. *EURASIP Journal on Wireless Communications and Networking*, 2011(1):1–12, 2011. doi: 10.1186/1687-1499-2011-153. URL http://dx.doi.org/10.1186/1687-1499-2011-153.

[123] S. Kim. An Ant-based Multipath Routing Algorithm for QoS aware Mobile Ad-hoc Networks. *Wireless Personal Communications*, 66(4):739–749, 2011. doi: 10.1007/s11277-011-0361-0. URL http://dx.doi.org/10.1007/s11277-011-0361-0.

[124] P. V. Krishna, V. Saritha, G. Vedha, A. Bhiwal, and A. S. Chawla. Quality-of-service-enabled ant colony-based multipath routing for mobile ad hoc networks. *IET communications*, 6(1):76–83, 2012. doi: 10.1049/iet-com.2010.0763. URL http://dx.doi.org/10.1049/iet-com.2010.0763.

[125] V. Balaji and V. Duraisamy. Ant Optimized Link Quality for Ad Hoc on Demand Distance Vector. *Wireless Personal Communications*, 79(1):763–771, 2014. doi: 10.1007/s11277-014-1885-x. URL http://dx.doi.org/10.1007/s11277-014-1885-x.

[126] S. K. Nivetha, R. Asokan, and N. Senthilkumaran. ACO-GA Hybrid Meta-Heuristic (AGHM) Optimization for Multi-constrained Quality of Service Routing in MANETs. In *Proceedings of the Int. Conf. on Advances in Communication, Network, and Computing, (CNC)*, 2014.

[127] A. Al-Ani. Routing Protocol for Self-Organized Communication Network with Guaranteed QoS. In *Proceedings of the 12. Ilmenauer TK-Manager Workshop. Tagungsband.*, pages 45–53, Ilmenau University of Technology, Communication Networks Research Lab, September 2012. URL http://www.db-thueringen.de/servlets/DocumentServlet?id=20784.

[128] A. Al-Ani. QoS-enabled Routing in MANETs Based on ACO and SNMP. In *Proceedings of the 13. Ilmenauer TK-Manager Workshop. Tagungsband.*, pages 77–87, Ilmenau University of Technology, Communication Networks Research Lab, September 2014. URL http://www.db-thueringen.de/servlets/DocumentServlet?id=24546.

[129] A. Al-Ani and J. Seitz. An Approach for QoS-aware Routing in Mobile Ad hoc Networks. In *Proceedings of the 12th International Symposium on Wireless Communication Systems (ISWCS'15)*, pages 626–630, Brussels, Belgium, 2015.

[130] A. Al-Ani and J. Seitz. QoS-aware Routing in Multi-rate Ad hoc Networks Based on Ant Colony Optimization. *Network Protocols and Algorithms*, 7(4):1–25, December 2015. doi: 10.5296/npa.v7i4.8513. URL http://dx.doi.org/10.5296/npa.v7i4.8513.

[131] S. Deering and R. Hinden. Internet Protocol, Version 6 (IPv6) Specification. Request for Comments RFC 2460, IETF Network Working Group, December 1998. URL http://dx.doi.org/10.17487/RFC2460.

[132] M. Dorigo, M. Birattari, and T. Stutzle. Ant colony optimization. *IEEE Computational Intelligence Magazine*, 1(4):28 – 39, Nov 2006. doi: 10.1109/MCI.2006.329691. URL http://dx.doi.org/10.1109/MCI.2006.329691.

[133] J. Case, M. Fedor, M. Schoffstall, and J. Davin. A simple network management protocol (snmp). Request for Comments RFC 1157, IETF Network Working Group, May 1990. URL http://dx.doi.org/10.17487/RFC1157.

[134] J. Case, K. McCloghrie, M. Rose, and S. Waldbusser. Introduction to Community-based SNMPv2. Request for Comments RFC 1901, IETF Network Working Group, January 1996. URL http://dx.doi.org/10.17487/RFC1901.

[135] J. Case, K. McCloghrie, M. Rose, and S. Waldbusser. Coexistence between Version 1 and Version 2 of the Internet-standard Network Management Framework.

Request for Comments RFC 1908, IETF Network Working Group, January 1996. URL http://dx.doi.org/10.17487/RFC1908.

[136] J. Case, R. Mundy, D. Partain, and B. Stewart. Introduction and applicability statements for internet standard management framework. Request for Comments RFC 3410, IETF Network Working Group, December 2002. URL http://dx.doi.org/10.17487/RFC3410.

[137] D. Harrington, R. Presuhn, and B. Wijnen. An Architecture for Describing Simple Network Management Protocol (SNMP) Management Frameworks. Request for Comments RFC 3411, IETF Network Working Group, December 2002. URL http://dx.doi.org/10.17487/RFC3411.

[138] J. Case, D. Harrington, R. Presuhn, and B. Wijnen. Message Processing and Dispatching for the Simple Network Management Protocol (SNMP). Request for Comments RFC 3412, IETF Network Working Group, December 2002. URL http://dx.doi.org/10.17487/RFC3412.

[139] D. Levi, P. Meyer, and B. Stewart. Simple Network Management Protocol (SNMP) Applications. Request for Comments RFC 3413, IETF Network Working Group, December 2002. URL http://dx.doi.org/10.17487/RFC3413.

[140] U. Blumenthal and B. Wijnen. User-based security model (usm) for version 3 of the simple network management protocol (snmpv3). Request for Comments RFC 3414, IETF Network Working Group, December 2002. URL http://dx.doi.org/10.17487/RFC3414.

[141] B. Wijnen, R. Presuhn, and K. McCloghrie. View-based Access Control Model (VACM) for the Simple Network Management Protocol (SNMP). Request for Comments RFC 3415, IETF Network Working Group, December 2002. URL http://dx.doi.org/10.17487/RFC3415.

[142] R. Presuhn, J. Case, K. McCloghrie, M. Rose, and S. Waldbusser. Version 2 of the Protocol Operations for the Simple Network Management Protocol (SNMP). Request for Comments RFC 3416, IETF Network Working Group, December 2002. URL http://dx.doi.org/10.17487/RFC3416.

[143] R. Presuhn, J. Case, K. McCloghrie, K. McCloghrie, M. Rose, and S. Waldbusser. Transport Mappings for the Simple Network Management Protocol (SNMP). Request for Comments RFC 3417, IETF Network Working Group, December 2002. URL http://dx.doi.org/10.17487/RFC3417.

[144] R. Frye, D. Levi, S. Routhier, and B. Wijnen. Coexistence between Version 1, Version 2, and Version 3 of the Internet-standard Network Management Framework.

Request for Comments RFC 3584, IETF Network Working Group, August 2003. URL http://dx.doi.org/10.17487/RFC3584.

[145] U. Blumenthal, F. Maino, and K. McCloghrie. The Advanced Encryption Standard (AES) Cipher Algorithm in the SNMP User-based Security Model. Request for Comments RFC 3826, IETF Network Working Group, June 2004. URL http://dx.doi.org/10.17487/RFC3826.

[146] J. Schoenwaelder and. Simple Network Management Protocol (SNMP) Context EngineID Discovery. Request for Comments RFC 5343, IETF Network Working Group, September 2008. URL http://dx.doi.org/10.17487/RFC5343.

[147] William Stallings. *SNMP, SNMPV2, SNMPv3, and RMON 1 and 2, Third Edition*. Addison-Wesley Longman Inc., 1999.

[148] M. Rose and K. McCloghrie. Structure and Identification of Management Information for TCP/IP-based internets. Request for Comments RFC 1155, IETF Network Working Group, May 1990. URL http://dx.doi.org/10.17487/RFC1155.

[149] K. McCloghrie, D. Perkins, and J. Schoenwaelder. Structure of management information version 2 (smiv2). Request for Comments RFC 2578, IETF Network Working Group, April 1999. URL http://dx.doi.org/10.17487/RFC2578.

[150] J. Case, K. McCloghrie, M. Rose, and S. Waldbusser. Protocol operations for version 2 of the simple network management protocol (snmpv2. Request for Comments RFC 1448, IETF Network Working Group, April 1993. URL http://dx.doi.org/10.17487/RFC1448.

[151] K. McCloghrie and M. Rose. Management Information Base for Network Management of TCP/IP-based internets. Request for Comments RFC 1156, IETF Network Working Group, May 1990. URL http://dx.doi.org/10.17487/RFC1156.

[152] K. McCloghrie and M. Rose. Management Information Base for Network Management of TCP/IP-based internets: MIB-II. Request for Comments RFC 1213, IETF Network Working Group, March 1991. URL http://dx.doi.org/10.17487/RFC1213.

[153] Douglas Mauro and Kevin Schmidt. *Essential SNMP*. O'REILLY Media, Inc., 2005.

[154] K. McCloghrie and F. Kastenholz. The Interfaces Group MIB. Request for Comments RFC 2863, IETF Network Working Group, June 2000. URL http://dx.doi.org/10.17487/RFC2863.

[155] S. A. Routhier. Management Information Base for the Internet Protocol (IP). Request for Comments RFC 4293, IETF Network Working Group, April 2006. URL http://dx.doi.org/10.17487/RFC4293.

[156] J. Galletly. Evolutionary Algorithms in Theory and Practice: Evolution Strategies, Evolutionary Programming, Genetic Algorithms. *Kybernetes*, 27(8):979–980, 1996. doi: 10.1108/k.1998.27.8.979.4. URL http://dx.doi.org/10.1108/k.1998.27.8.979.4.

[157] The network simulator ns-3, 2015. URL http://www.nsnam.org.

[158] G. F. Riley. The Georgia Tech Network Simulator. In *Proceedings of the ACM SIGCOMM Workshop on Models, Methods and Tools for Reproducible Network Research (MoMeTools '3)*, pages 5–12, New York, NY, USA, 2003. doi: 10.1145/944773.944775. URL http://doi.acm.org/10.1145/1190455.1190467.

[159] M. Lacage and T. R. Henderson. Yet Another Network Simulator. In *Proceedings of the 2006 Workshop on ns-2: The IP Network Simulator (WNS2 '06)*, ACM, New York, NY, USA, 2006. doi: 10.1145/1190455.1190467. URL http://doi.acm.org/10.1145/944773.944775.

[160] K. Wehrle, F. Pählke, H. Ritter, D. Müller, and M. Bechler. *The Linux Networking Architecture: Design and Implementation of Network Protocols in the Linux Kernel*. Pearson Prentice Hall, 2005.

[161] E. Weingartner, H. vom Lehn, and K. Wehrle. A Performance Comparison of Recent Network Simulators. In *Proceedings of the IEEE International Conference on Communications (ICC '09)*, pages 1–5, Dresden Germany, 2009. URL http://dx.doi.org/10.1109/ICC.2009.5198657.

[162] A. R. Khan, S. M. Bilal, and M. Othman. A Performance Comparison of Open Source Network Simulators for Wireless Networks. In *Proceedings of the IEEE International Conference on Control System, Computing and Engineering (ICCSCE)*, pages 34–38, Penang Malaysia, 2012. doi: 10.1109/ICCSCE.2012.6487111. URL http://dx.doi.org/10.1109/ICCSCE.2012.6487111.

[163] The network simulator ns-3: Conceptual overview, 2015. URL https://www.nsnam.org/docs/tutorial/html/conceptual-overview.html.

[164] The network simulator ns-3: Internet stack, 2015. URL https://www.nsnam.org/docs/release/3.10/manual/html/internet-stack.html.

[165] T. Henderson. ns-3 Tutorial: Part 1. In *9th GENI Engineering Conference (GEC9)*, page 27, November 2010. URL https://www.nsnam.org/tutorials/geni-tutorial-part1.pdf.

[166] The network simulator ns-3: Routing overview, 2015. URL https://www.nsnam.org/docs/release/3.10/manual/html/routing.html.

[167] G. Singh, N. Kumar, and A. K. Verma. Oantalg: An orientation based ant colony algorithm for mobile ad hoc networks. *Wireless Personal Communications*, 77(3): 1859–1884, August 2014. doi: 10.1007/s11277-014-1613-6. URL http://dx.doi.org/10.1007/s11277-014-1613-6.

[168] P. Seeling, M. Reisslein, and F. H.P. Fitzek. Offset trace-based video quality evaluation after network transport. *Journal of Multimedia*, 1(2):1–13, May 2006. doi: 10.4304/jmm.1.2.1-13. URL http://dx.doi.org/10.4304/jmm.1.2.1-13.

[169] P. Seeling and M. Reisslein. Video Transport Evaluation With H.264 Video Traces. *IEEE Communications Surveys Tutorials*, 14(4):1142–1165, 2012. doi: 10.1109/SURV.2011.082911.00067. URL http://dx.doi.org/10.1109/SURV.2011.082911.00067.

[170] P. Seeling, M. Reisslein, and B. Kulapala. Network performance evaluation using frame size and quality traces of single-layer and two-layer video: A tutorial. *IEEE Communications Surveys Tutorials*, 6(3):58–78, 2004. doi: 10.1109/COMST.2004.5342293. URL http://dx.doi.org/10.1109/COMST.2004.5342293.

[171] S. Corson and J. Macker. Mobile ad hoc networking (manet): Routing protocol performance issues and evaluation considerations. Request for Comments RFC 2501, IETF Network Working Group, January 1999. URL http://dx.doi.org/10.17487/RFC2501.

[172] F. Lassabe, P. Canalda, P. Chatonnay, F. Spies, and O. Baala. A Friis-based calibrated model for WiFi terminals positioning. In *Proceedings of the IEEE International Symposium on a World of Wireless Mobile and Multimedia Networks (WoWMoM)*, pages 382–387, June 2005. doi: 10.1109/WOWMOM.2005.2. URL http://dx.doi.org/10.1109/WOWMOM.2005.2.

[173] T. Rappapor. *Wireless Communications: Principles and Practice*. Prentice-Hall, Englewood Cliffs, NJ, 1996.

[174] N. Aschenbruck, R. Ernst, E. Gerhards-Padilla, and M. Schwamborn. Bonnmotion: A Mobility Scenario Generation and Analysis Tool. In *Proceedings of the 3rd International Conference on Simulation Tools and Techniques (ICST)*, pages 15–60, Brussels, Belgium, 2010. doi: 10.4108/ICST.SIMUTOOLS2010.8684. URL http://dx.doi.org/10.4108/ICST.SIMUTOOLS2010.8684.

[175] M. Behrisch, L. Bieker, J. Erdmann, and D. Krajzewicz. SUMO – Simulation of Urban MObility. In *Proceedings of the third International Conference on Advances in System Simulation (SIMUL)*, Barcelona, Spain, 2011.

[176] Video trace. URL http://trace.eas.asu.edu/.

List of Figures

List of Tables

Abbreviations

AARF	**A**daptive **A**uto **R**ate **F**allback
AARF-CD	**A**daptive **A**uto **R**ate **F**allback with **C**ollision **D**etection
ABC	**A**nt-**B**ased **C**ontrol
ACK	**A**cknowledgment
ACO	**A**nt **C**olony **O**ptimization
ACO-AHR	**A**nt **C**olony **O**ptimization **a** **h**ybrid **r**outing
AEARA	**A**daptive **E**mergent **A**d hoc **R**outing **A**lgorithm
A_{ijd}	**A**dditive metrics through the path $p(i, j, d)$
AMAR	**A**daptive **M**ultipath **A**nt **R**outing
AMQR	**A**nt-based **M**ulti-objective **Q**oS **R**outing
AMQR	**A**nt colony based **M**ultipath **Q**oS-aware **R**outing
Ant-CAMP	**Ant** based **C**ongestion **A**daptive **M**ulti**p**ath
Ant-DYMO	**Ant-Dy**namic MANET **O**n-demand
AntOR	**Ant**HocNet-based improved routing
AntSeq	**Ant** unique **Seq**uence number
AODV	**A**d hoc **O**n-**D**emand **D**istance **V**ector
AODVM	**AODV**-**M**ultipath
AOMDV	**A**d hoc **O**n-demand **M**ultipath **D**istance **V**ector
AP	**A**ccess **P**oint
API	**A**pplication **P**rogramming **I**nterface
ARA	**A**nt-Colony-Based **R**outing **A**lgorithm
ARAMA	**A**nt **R**outing **A**lgorithm for **M**obile **A**d-hoc networks
ARF	**A**utomatic **R**ate **F**allback
ARP	**A**ddress **R**esolution **P**rotocol
ASN.1	**A**bstract **S**yntax **N**otation One

AVC	**A**daptive **V**ideo **C**oding
B	**B**andwidth
BANT	**B**ackward **ant**
BER	**B**it **E**rror **R**ate
$\mathbf{B}_{ijd}$	Available **B**andwidth for the path $p(i, j, d)$
$\mathbf{B_{min}}$	**Min**imum **B**andwidth
BMR	**B**andwidth-aware **M**ulti-path **R**outing
$\mathbf{B_{thr}}$	QoS **thr**eshold for **B**andwidth
CARA	**C**ollision-**A**ware **R**ate **A**daptation
CBR	**C**onstant **B**it **R**ate
CCA	**C**lear **C**hannel **A**ssessment
C_{ijd}	**C**oncave metrics through the path $p(i, j, d)$
CITR	**C**hannel's **I**dle **T**ime **R**atio
CLWPR	**C**ross-**L**ayer **W**eighted **P**osition based **R**outing
CSMA/CA	**C**arrier **S**ense **M**ultiple **A**ccess with **C**ollision **A**voidance
CS-neighbor	**C**arrier **S**ensing neighbor node
CTS	**C**lear-**T**o-**S**end
D	End-to-End **D**elay
DAR	**D**istributed **A**nt **R**outing
DCF	**D**istributed **C**oordination **F**unction
DIFS	**D**istributed **I**nter-**F**rame **S**paces
$\mathbf{D}_{ijd}$	End-to-end **D**elay for the path $p(i, j, d)$
$\mathbf{D_{max}}$	**Max**imum **D**elay
$\mathbf{D_{prop}}$	**Prop**agation **D**elay
$\mathbf{D_{queue}}$	**Queu**ing **D**elay
DREAM	**D**istance **R**outing **E**ffect **A**lgorithm for **M**obility
$\mathbf{DropPkts}_{\boldsymbol{in}}$	The number of **in**coming **packets** that are **drop**ped
$\mathbf{DropPkts}_{\boldsymbol{out}}$	The number of **out**going **packets** that **drop**ped
DSDV	**D**estination-**S**equenced **D**istance-**V**ector
DSR	**D**ynamic **S**ource **R**outing
DSSS	**D**irect **S**equence **S**pread **S**pectrum
$\mathbf{D_{thr}}$	QoS **thr**eshold for **D**elay
$\mathbf{D_{total}}$	**Total** **D**elay at node n

$\mathbf{D_{trans}}$	**Trans**mission **D**elay
EANT	**E**rror **ant**
EARA	**E**mergent **A**d Hoc **R**outing **A**lgorithm
E-DYMO	**E**nhancement of **Dy**namic **M**ANET **O**n-demand
EPD	**E**xpected **P**ath **D**elay
FANT	**F**orward **ant**
FLS	**F**uzzy **L**ogic **S**ystem
FTP	**F**ile **T**ransfer **P**rotocol
GNU GPLv2	**GNU G**eneral **P**ublic **L**icense, **v**ersion **2**
GPS	**G**lobal **P**ositioning **S**ystem
GPSR	**G**reedy **P**erimeter **S**tateless **R**outing
GTNetS	**G**eorgia **T**ech **Net**work **S**imulator
HOPNET	Hybrid ant colony optimization routing algorithm
HSLS	**H**azy **S**ighted **L**ink **S**tate
IEEE	**I**nstitute of **E**lectrical and **E**lectronics **E**ngineers
IETF	**I**nternet **E**ngineering **T**ask **F**orce
IFS	**I**nter-**F**rame **S**paces
IoT	**I**nternet **o**f **T**hings
IP	**I**nternet **P**rotocol
IPv4	**I**nternet **P**rotocol **v**ersion **4**
IPv6	**I**nternet **P**rotocol **v**ersion **6**
LAR	**L**ocation-**A**ided **R**outing
LQF	**L**ink **Q**uality **F**ormat
LTE	**L**ong **T**erm **E**volution
MAC	**M**edia **A**ccess **C**ontrol layer
MANET	**M**obile **A**d hoc **Net**work
MIB	**M**anagement **I**nformation **B**ase
M_{ijd}	**M**ultiplicative metrics through the path $p(i, j, d)$
MMQARP	**M**ulti-constrained and **M**ultipath **Q**oS **A**ware **R**outing **P**rotocol
MO	**M**anaged **O**bjects
MPR	**M**ulti**p**oint **R**elay
MTU	**M**aximum **T**ransmission **U**nit
NAV	**N**etwork **A**llocation **V**ector

NLC	**N**eighbor node-**L**ink **C**ount
NP	**N**ondeterministic Polynomial time
ns-2	**n**etwork **s**imulator-2
ns-3	**n**etwork **s**imulator-3
OAR	**O**pportunistic **A**uto **R**ate
OFDM	**O**rthogonal **F**requency-**D**ivision **M**ultiplexing
OID	**O**bject **ID**entifier
OLSR	**O**ptimized **L**ink **S**tate **R**outing
P	**P**olynomial time
$p(i, j, d)$	**P**ath from node **i** to destination **d** through neighbor node **j**
POSANT	**Pos**ition Based **Ant** Colony
PSTN	**P**ublic **S**witched **T**elephone **N**etwork
PAN	**P**ersonal **A**rea **N**etworks
PBANT	**P**osition **B**ased **ANT** Colony Routing Algorithm
PCF	**P**oint **C**oordination **F**unction
PDA	**P**ersonal **D**igital **A**ssistants
PDU	**P**rotocol **D**ata **U**nit
PERA	**P**robabilistic **E**mergent **R**outing **A**lgorithm
PHY	**Phy**sical layer
PIFS	**P**oint **I**nter-**F**rame **S**paces
$\mathbf{Pkts}_{\boldsymbol{Rx}}$	The correctly received incoming packets
$\mathbf{Pkts}_{\boldsymbol{Tx}}$	The number of successfully sent packets
PLR	**P**acket **L**oss **R**atio
PACONET	Im**P**roved **A**nt **C**olony **O**ptimization **Net**work
QAMR	**Q**oS-enabled **A**nt colony based **M**ultipath **R**outing
QARP	**Q**oS-**a**ware **R**outing **P**rotocol
QB	Residual **B**andwidth
QD	Residual **D**elay
QMRP	**Q**oS-aware **M**ultipath **R**outing **P**rotocol
QMTR	**Q**oS-aware **M**ultipath **T**hreshold **R**outing
QoRA	**Qo**S-aware **R**outing based on **A**nt Colony Optimization
QoS	**Q**uality **of** **S**ervice
QP	**Q**oS **P**arameter

QP$_{\textbf{threshold}}$	**Q**oS **P**arameter **threshold**
QS	Residual expected success rate
RBAR	**R**eceiver-**B**ased **A**utoRate
RERR	**R**oute **Err**or
RFC	**R**equest **f**or **C**omments
RNG	**R**andom **N**umber **G**enerate (RNG)
RRAA	**R**obust **R**ate **A**daptation **A**lgorithm
RREP	**R**oute **Rep**ly
RREQ	**R**oute **Req**uest
RSP	**R**oute **S**election **P**robability
RSQR	**R**oute **S**tability based **Q**oS **R**outing
RSS	**R**eceive **S**ignal **S**trength
RTS	**R**equest-**T**o-**S**end
RWP	**R**andom **W**ay**P**oint mobility
RWS	**R**oulette-**W**heel **S**election
S	Expected **S**uccess Rate
SI	**S**warm **I**ntelligence
SIFS	**S**hort **I**nter-**F**rame **S**pacing
S$_{ijd}$	Expected **S**uccess Rate for the path $p(i, j, d)$
SINR	**S**ignal-to-**I**nterference-plus-**N**oise **R**atio
SMI	**S**tructure of **M**anagement **I**nformation
S$_{\textbf{min}}$	**Min**imum Expected **S**uccess Rate
SMORT	**S**calable **M**ultipath **O**n-demand **R**outing **P**rotocol
SMR	**S**plit **M**ultipath **R**outing
SNMP	**S**imple **N**etwork **M**anagements **P**rotocol
SNR	**S**ignal-to-**N**oise **R**atio
SON	**S**elf-**O**rganized **N**etworks
S$_{\textbf{thr}}$	QoS **thr**eshold for Expected **S**uccessful Rate
SUMO	**S**imulation of **U**rban **MO**bility
SVC	Scalable Video Coding
TBRPF	**T**opology **B**roadcast based on **R**everse-**P**ath **F**orwarding
TCP	**T**ransmission **C**ontrol **P**rotocol
$T_{\textbf{idle}}$	**T**ime for which the channel is observed to be **idle**

TORA	**T**emporally **O**rdered **R**outing **A**lgorithm
UDP	**U**ser **D**atagram **P**rotocol
V2I	**V**ehicle-**to**-**I**nfrastructure
V2V	**V**ehicle-**to**-**V**ehicle
VANET	**V**ehicular **A**d hoc **N**etwork
VBR	**V**ariable **B**it **R**ate
VoIP	**V**oice over **I**nternet **P**rotocol
WiMAX	**W**orldwide **i**nteroperability for **M**icrowave **A**ccess
WLAN	**W**ireless **L**ocal **A**rea **N**etwork
WPAN	**W**ireless **P**ersonal **A**rea **N**etwork
WRP	**W**ireless **R**outing **P**rotocol
YANS	**A**nother **N**etwork **S**imulator
ZRP	**Z**one **R**outing **P**rotocol

Symbols

α	Pheromone weight factor
β	Heuristic weight factor
β_B	Bandwidth weight factor
β_S	Success rate weight factor
β_D	Delay weight factor
τ	Pheromone value
η	Heuristic factor
ρ	Diffusion factor
ψ	Forward ant goodness value
θ	T_{idle} weight moving factor
ϑ	Delay weight moving factor
φ	Success rate weight moving factor
$\Delta\tau$	Pheromone increment factor
ΔT	Monitoring Window