Network Protocols and Algorithms
ISSN 1943-3581
2015, Vol. 7, No. 4

QoS-aware Routing in Multi-rate Ad hoc Networks Based on Ant Colony Optimization

Aymen Al-Ani and Jochen Seitz
Communication Networks Group, Technische Universität Ilmenau
Ilmenau (Germany)
Email: {Aymen.Al-ani, Jochen.Seitz}@tu-ilmenau.de

Received: November 3, 2015 Accepted: December 9, 2015 Published: December 31, 2015

DOI: 10.5296/npa.v7i4.8513 URL: http://dx.doi.org/10.5296/npa.v7i4.8513

Abstract

QoS-aware routing in mobile ad hoc networks (MANETs) is a major challenge due to node mobility and scarcity of resources. QoS-aware routing based on ant colony optimization (ACO) algorithms is a promising approach to overcome these problems. However, as compared to MANETs, vehicular ad hoc networks (VANETs) face additional challenges due to a rapid topology change, making the estimation or prediction of QoS parameters difficult or stale. VANETs require time-critical message delivery, as late delivery may result in endangering lives. Currently existing routing protocols usually require the exchange of additional control message between neighbor nodes to compute QoS parameters. This makes the routing protocol too slow to react to fast topology change and also does not consider network congestion when forwarding a data packet. To reduce the overhead introduced to collect information from neighbor nodes and to obtain an accurate estimate of QoS parameters, we use the simple network management protocol to estimate these values locally. This paper describes a new approach for calculating QoS parameter locally and avoiding congestion during data transmission. The simulations are implemented using the network simulator ns-3, and the results show that our approach **Qo**S **R**outing based on **A**nt Colony Optimization (**QoRA**) protocol is scalable and performs well in high mobility.

Keywords: Ant Colony Optimization, Mobile Ad hoc Networks, Vehicular Ad hoc Networks, QoS-aware Routing, Simple Network Management Protocol.

1. Introduction

Next generation networks (NGNs) include various popular network topologies like MANETS, VANETs, WSNs, etc. The main feature of these networks is their self-organized multi-hop wireless communication. NGNs are getting more important, because of the growing number of wireless mobile electronic devices. These devices have the ability to afford real-time multimedia applications, such as digital video and audio streaming. Especially, MANETs and VANETs have related features, such as low bandwidth, short transmission range, and poor link quality. Nevertheless, VANETs have additional characteristics, such as repeated topology changes and frequent disconnections. However, VANETs play an important role in developing intelligent transportation systems (ITS). ITS applications are provided by network vehicles over multiple hops by using a unicast mode. Additionally, using multimedia information (audio/video) can play a significant role in traffic management and road safety.

To ensure fast and reliable delivery of ITS information or multimedia data, Quality of Service (QoS) should be considered in ad hoc networks, too. Routing protocols are the focus of QoS mechanisms. The purpose of QoS-aware routing protocols is to calculate QoS parameters and to select a path from source to destination that satisfies the application QoS requirements. Selection and optimization of the routing protocol play a primary role in improving QoS in ad hoc networks. Metaheuristic approaches seem to provide a suitable method to solve the problem of QoS routing, since they provide an excellent solution to an optimization problem by iteratively trying to improve a candidate solution, especially with incomplete or imperfect information or limited computation capacity.

One of the promising optimization methods that can be applied in the wireless ad hoc networks is Ant Colony Optimization (ACO). ACO routing protocols are inspired by the foraging behavior of ants that optimally derive the optimal paths to food. Ants deposit pheromone on the paths to guide incoming ants about the usage of the particular path to the food source. The pheromone is a volatile substance and in terms of routing it could be used to indicate the availability of certain resources on the path, thus, enabling QoS-awareness. On the contrary, most of the QoS-aware routing protocols require additional exchange of control messages to determine QoS parameters and do not consider the problem of congestion during data forwarding. Thus, these protocols increase the routing overhead, waste time and energy in a mobile electronic device during path discovery, and increase packet loss ratio.

Therefore, this paper aims to present our approach QoRA that utilizes the monitoring features of the simple network managements protocol (SNMP), which already is supported in the mobile device to obtain required values locally. Based on these values, we can calculate the QoS parameters and avoid congestion during data packet forwarding. The basic and initial results of the QoRA approach are presented in [1].

This paper is organized as follows. The related work is presented in Section 2. We provide the details of the proposed QoRA architecture and its components in Section 3. The QoRA routing decision based on three QoS constraints which are estimated based on SNMP is explained in detail in Section 4. Section 5 presents the details of the QoS threshold estimation to avoid congestions. Section 6 describes the functioning of the QoRA routing protocol. The implementation and feasibility of our approach are shown via simulations in ns-3 in Section 7. Finally, the paper is concluded in Section 8, where we also provide some aspects of further development.

2. Related work

Several routing protocols have been proposed to support the quality of the communications service in NGNs. The biggest challenge can be summarized as the estimation of link QoS parameters because of node mobility, lack of precise state information, fading, and shared radio channel [2]. To achieve a systematic view on the state of the art of QoS-aware routing in MANETs and VANETs, the investigated publications are organized into three groups. The first group deals with QoS-aware routing in MANETs. The second group summarizes QoS-aware routing protocols in VANETs. Finally, the third group focuses on QoS-aware routing based on the Ant Colony Optimization (ACO) paradigm that seems to be an interesting approach to providing adaptive QoS in highly mobile networks.

2.1 QoS-aware Routing in MANETs

Huang et al. [3] proposed a multi-constrained QoS multipath routing protocol. The objective function is to find the optimal paths to a given destination based on soft-QoS that is mapped into the local link available information. However, they do not provide the measurement method of the local parameter for link delay and reliability. Balachandra et al. [4] presented a multi-constrained and multipath QoS-aware routing protocol (MMQARP) for MANETs. The routing decision depends on three QoS constraints, which are route reliability, delay across the link, and energy efficiency of nodes. These QoS constraints are used to compute node-disjoint multiple paths from source to destination. Therefore, the protocol requires synchronization to calculate the average delay and geographical information to estimate reliability. So, the position identification of the node leads to expensive equipment and message exchange. Obaidat et al. [5] proposes a QoS-aware Multipath Routing Protocol (QMRP) for MANETs. Feedback from cross-layer communications between physical (PHY), medium access control (MAC) and routing layers is used to compute the node-disjoint multiple paths. QMRP improves AODV significantly as it extends the route-broadcasting packet with 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. However, QMRP introduces additional routing overhead as compared to AODV due to the discovery of more than one path in each route discovery process. Ali et al. [6] developed the QoS-aware multipath threshold routing (QMTR) protocol for MANETs. In this approach, traffic is distributed over fail-safe different paths to reduce the load of a congested node. The congestion mechanism is based on available bandwidth, node transmission delay, and load. However, the protocol tries to select the routes to the destination optimally without adopting the end-to-end QoS constraints. So, the local QoS might exist, but the end-to-end QoS on the complete route is not satisfied. Thus, the availability of local QoS parameters is not sufficient to transport real-time multimedia applications. Lal et al. [7] 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 application. The protocol adapts session admission control (SAC) and cross-layer communication (CLC) to locally estimate the bandwidth for each node. Additionally, the route discovery phase produces high overhead and delay or consumes higher power, since information is collected from two-hop neighbors to compute the available bandwidth at each intermediate node for the selected paths.

2.2 QoS-aware Routing in VANETs

Due to its dynamic network topology, Chen.et.al [8] specified that QoS in VANET should be considered for fast and reliable delivery of the message. To address this problem, Niu et al. [9] produced the delay and reliability constrained QoS (DeReQ) routing protocol. The proposed protocol uses link lifetime and current traffic density to compute reliability. Therefore, additional geographic information such as a digital map and specific devices are required to provide geographic location information. Asefi et al. [10] proposed an application-centric routing framework that includes road-side units for the dissemination of video streams. A queuing-based mobility model and a connectivity probability for sparse and dense VANETs have been adopted. In addition, the selection of the next node is performed depending on the video packet rate, the packet error probability, and the total transmission time. However, the protocol did not adopt end-to-end QoS constraints for real-time and multimedia applications.

Another routing method is the location-based approach, where the data is forwarded on the basis of the location information such as the position of the source, destination, and neighbor nodes. In [11] it is shown that position-based protocols are more suitable for multimedia transmission over VANET than reactive protocols, whereas proactive protocols are not at all suitable for high mobility in VANETs. Katsaros et al. [12, 13] 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 in VANETs. The cross-layering mechanism keeps track of the PHY layer parameter like 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. The CLWPR protocol is based on the Optimized Link State Routing (OLSR) module, which is a proactive ad-hoc protocol.

2.3 QoS-aware ACO Routing

A number of protocols have been proposed for solving the QoS problem using Ant

Colony Optimization (ACO). ACO routing protocols are enabled with adaptability that makes them flexible towards the changing environmental conditions and increases their capability against the failures and damages occurring in the network [14]. Deepalakshmi et al. [15] introduced the ant-based multi-objective QoS routing protocol for MANETs to support multimedia communications. This protocol determines various paths based on number of hops, delay, and link bandwidth. The path with the highest preference probability is selected to send the data to the desired destination. However, the protocol utilizes periodic hello messages to detect link failures and to evaluate the available bandwidth for each neighbor node, which increases the overhead in the network. In addition, it assumes that all nodes in a MANET are synchronized for computing QoS parameters. Krishna et al. [16] 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. For the path preference probability calculation, different parameters such as delay, bandwidth and number of hops are measured. However, the main drawback of this protocol is that the algorithm does not define the methods for calculating the available bandwidth and delay. In addition, periodic information needs to be exchanged between neighbor nodes in order to compute the link stability that incurs overhead. Kim [17] introduced a multi-path routing strategy based on the ACO algorithm. During the path discovery phase, each node sends an ant packet randomly from one node to another node. The forward ant selects the next hop based on the distance and queue length available at the neighbor node. In addition, each node periodically transmits hello message to maintain local connectivity with neighbor nodes. The proposed routing protocol chooses the most adaptable paths that satisfy the QoS constraints in terms of bandwidth and delay. However, the proposed algorithm relies on the fact that the bandwidth and delay information are available beforehand. Balaji et al. [18] proposed a 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 two neighbors can be assessed based on received signal strength that can be represented by other network factors such as battery power, distance, and mobility. Here, the regular hello messages are extended to a new packet Link Quality Format. It is a link quality integer metric that defines the link quality between the neighboring nodes. However, the proposed protocol used hello message that consume available bandwidth and energy. Nivetha et al. [19] proposes the combination of two stochastic optimization methods ACO and GA (Genetic Algorithms) called 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. 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 a large number of iterations in order to find the optimal path and thereby consume a big amount of time, which is not desirable for the multimedia applications.

2.3 General Drawbacks of Current Approaches

This section discusses general shortcomings observed in the measurement techniques adopted by the QoS-aware routing protocols presented in previous sections. A significant group of QoS routing protocols (e.g., [3, 16, 17, 19]) assumes 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. Other approaches clearly state that they use additional information (e.g. geographical information about the nodes [4, 9, 12]) or extra control messages to estimate the quality of outgoing links [5, 7, 15, 16, 17, 18]. This leads to increased control traffic at the expense of bandwidth for multimedia traffic. Additionally, the local link quality information (e.g., [6, 10, 18]) 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 the destination. Finally, another approach assumes all the nodes in MANETs are synchronized to compute QoS parameters (e.g., [4, 15]). However, synchronization signaling incurs extra overhead. Furthermore, in a constantly changing topology with nodes joining or leaving the network, the effort for keeping up synchrony is very high (exhausting the batteries of the mobile nodes) and might not always achieve synchrony in all nodes at one time.

Therefore, the idea of the QoRA approach is to develop an ACO-based routing algorithm that does not require further control messages or synchronous nodes. To achieve that, we explore QoS parameters from a network management entity and locally derive the quality of outgoing links. With this approach, the quickly changing topology of a MANET/VANET can be tackled, and QoS can be kept up.

3. QoRA Approach

This section gives an overview of our approach **Q**o**S** **R**outing based on **A**nt Colony Optimization (**QoRA**) and describes its relevant architectural components. These components and the information exchange between them is shown in Figure 1. The first element is the QoRA Entity. The QoRA entity runs on each node to identify suitable paths according to the specified QoS requirements. The second component is the SNMP entity consisting of the SNMP agent and the MIB. Where the necessary information required for QoS calculations is provided by SNMP agent. We describe both entities in the later sections.

3.1 QoRA Entity Components

The first element is the QoRA Entity, which runs on each node to identify suitable paths according to the given QoS requirements. It consists of five components (Ant Management, Neighbor Table, Routing Table, QoRA Routing Protocol, and QoS Manager) that will be described in more details below.

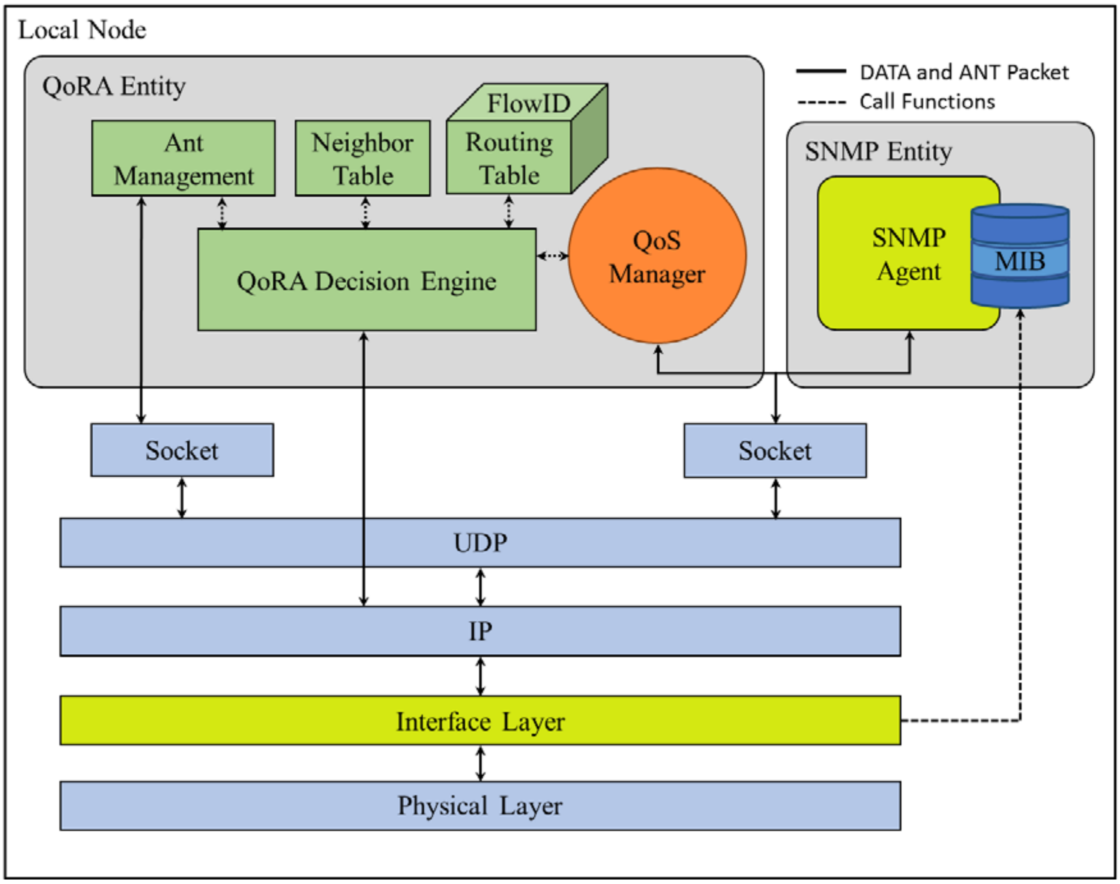

Figure 1. The QoRA Architecture

1) *Ant Management:* The Ant Management is responsible for generating three different types of ants, which are 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. The ants contain specific information to provide QoS-aware routing and deposit pheromone on the paths they identified. All ant and data packets contain a unique FlowID, which identifies the source, the destination, and the counter value to identify the data stream or flow called FlowLabel. Besides that, there are some specific fields related to QoS parameters of the stream on the visited path.

The first type of ants is FANT has three special packet fields $(\mathrm{QP}, \mathrm{QP}_{\mathrm{desired}}, \mathrm{FantStack})$. The QoS parameter QP represents the end-to-end delay (D), and expected success rate (S) aggregated over all visited nodes. The field $\mathrm{QP}_{\mathrm{desired}}$ represents the desired value for the user defined QoS requirements in terms of minimum bandwidth (B_{min}), maximum delay (D_{max}), and minimum expected success rate (S_{min}). The FantStack contains the IP addresses of all visited nodes.

The second type of ants is BANT, which also has three special packet fields $(\mathrm{QP}, \mathrm{QP}_{\mathrm{residual}}, \mathrm{BantStack})$. The QoS parameter QP represents the minimum available bandwidth (B), the end-to-end delay (D), and expected success rate (S) aggregated over all visited nodes. The field $\mathrm{QP}_{\mathrm{residual}}$ represents the residual of QoS parameters in terms of residual bandwidth (QB), residual end-to-end delay (QD), and residual expected success rate (QS). The BantStack contains the IP addresses of all visited nodes as collected by the FANT.

The third type of ants is EANT, which has only one field called EantStack that contains the FlowID for all the affected destinations/flows.

2) *QoRA Neighbor Table:* In the neighbor table, each neighbor is registered along with the pheromone. The pheromone τ_{ij} gives an indication of the goodness of the link from i to j, and thus indicates the number of packets that selected the link recently because each BANT or data packet will increase the accumulated pheromone amount by $\Delta\tau_{ij}$ according to Equation 1. This equation is derived from the forms given in [20].

$$\tau_{ij} = (1-\rho)\,\tau_{ij} + \Delta\tau_{ij} \tag{1}$$

The chemical pheromone is a volatile substance, and the pheromone on this link is reduced based on decay factor ρ, which is defined as the time interval of sending a data frame at node i and successfully receiving the data frame at node j.

3) *QoRA Routing Table:* The QoRA Routing Table contains the identified different routes towards known destinations for each flow and is used for data packets forwarding. For each flow, multiple paths may be recorded in the routing table. An entry of the routing table contains information about the route from node i to destination d over neighbor node j. Each entry for a destination is associated with a list of suitable neighbor nodes. The routing table entry contains the following fields: FlowID, next hop address toward this destination, QoS parameter QP, QoS threshold $\mathrm{QP}_{\mathrm{threshold}}$, the heuristic factor η_{ijd}, and a probability value $\mathcal{P}_{ijd}$ that reflects how likely it is that a data packet is forwarded using this neighbor.

The heuristic factor is calculated according to Equation 2. The QoRA decision is based on three QoS constraints, namely, bandwidth (B), delay (D), and expected success rate (S) that are calculated based on the cooperation with the SNMP agent as explained in Section 4. Where, β_B, β_S and β_D denote the adjustment parameters of bandwidth, expected success rate, and delay, respectively

$$\eta_{ijd} = \frac{[B_{ijd}]^{\beta_B} \times [S_{ijd}]^{\beta_S}}{[D_{ijd}]^{\beta_D}} \tag{2}$$

The probability $\mathcal{P}_{ijd}$ is calculated based on the pheromone and the heuristic factor according to Equation 3, where N_i is a set of neighbor nodes of i, and l is the neighbor node of i through which a route is available to destination d. The parameters α and β define the adjustment parameters of the pheromone and the heuristic factor, respectively. The equation to calculate the probabilities and the one to calculate the heuristic factor are derived from versions given in [20]. The probabilities of all neighbors with an available path to d sum up to 1.

$$\mathcal{P}_{ijd} = \frac{[\tau_{ij}]^{\alpha}[\eta_{ijd}]^{\beta}}{\sum_{l\in N_i}[\tau_{ij}]^{\alpha}[\eta_{ijd}]^{\beta}} \tag{3}$$

Thus, the information in this table includes the QoS information about the route. However, the routing table entry for a destination is updated at a node only after receiving a BANT from the destination. Based on the routing table, data traffic will be distributed according to the probabilities for each neighbor in the routing table.

4) *QoRA Decision Engine:* The most important block is the QoRA Decision Engine in the QoRA entity. It causes the different ants to be sent through the Ant Management component, and it updates Neighbor and Routing Table based on the QoS parameters, the information gathered by the ants. The QoS parameters are obtained by the QoS Manager communicating with the local SNMP Agent.

5) *QoS Manager:* The QoS manager can be described as a software module that is responsible for command generator and notification receiver applications. This manager communicates with the SNMP agent to retrieve or modify management information. For the QoRA approach, we have designed a lightweight implementation of the SNMP Manager that is integrated into the QoS Manager. The QoS Manager calls one of the following SNMP operations: Get, Set, and notification receive (trap). Additional, it is responsible for calculating QoS parameters locally based on the communication with the SNMP agent.

3.2 SNMP Entity Components

The Simple Network Management Protocol (SNMP) was first standardized in 1988 and is currently available in version 3 [21]. It is an application protocol on top of the User Datagram Protocol (UDP) and is used to retrieve management information from a management entity known as SNMP agent.

1) SNMP Agent: The SNMP agent is usually available on every networked node and collects the relevant information for this node. In order to select outgoing links according to the required quality of service, detailed information about the characteristics of the links is required. These could be determined using measurements, but this would cause additional traffic influencing the characteristics of the links significantly. Thus, we decided to retrieve this information from the SNMP agent locally as explained in Section 4. Hence, no additional measurements are needed.

2) Management Information Base: The management information is available through Managed Objects, which are stored in a standardized tree-based Management Information Base (MIB) [22]. This tree-structure serves to address uniquely Managed Objects. To give some examples of the standardized Managed Objects useful to assess the quality of different outgoing links, Table 1 lists the relevant objects for our approach. The short descriptions are taken from the relevant Request for Comments [23].

Table 1. Relevant SNMP managed objects.

Object	Short Description
ifSpeed	An estimate of the interfaces 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 sublayer to a higher (sub-)layer, which were addressed to a multicast address at this sub-layer.
ifHCInBroadcastPkts	The number of packets, delivered by this sublayer 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. Estimate QoS Parameters Based on SNMP

The QoRA entity is responsible for finding a path to the destination for given QoS requirements. It therefore needs to determine which next hops are suitable to forward packets that belong to a certain flow. For calculating the QoS values of the outgoing links, it relies on the measurements of the various link parameters performed by the SNMP agents. The following QoS parameters are considered to determine the quality of the outgoing link.

1) Bandwidth: One of the most important QoS parameters is the bandwidth or the channel transmission speed [24]. The transmission speed is automatically selected based on the received signal strength by a mechanism called Receiver-Based Auto-Rate (RBAR) [25]. In multi-rate ad hoc wireless networks, the bandwidth for a given path is limited by the link minimum transmission speed over all of its component links. Thus, each node must determine the transmission speed of its outgoing links. This value is stored in the Managed Object "ifSpeed" (cf. Table 1). So, for determining whether a node can satisfy a bandwidth requirement $B_{min}\,[b/s]$ in the path P_{ijd}, it must fulfill Equation 4. Since, low transmission speed produces lower throughput and increases packet transition delay. Additional, 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. Therefore, we selected the B_{min} as the basic rate of the IEEE 802.11 standard.

$$B_{ijd} = min\{\text{ifSpeed}\,(n), \forall n \in \mathrm{P}_{ijd}\} \geq B_{min} \tag{4}$$

2) Delay: The next QoS parameter is a delay. The estimation of the total delay time for a transmitted data packet between two neighbor nodes can be calculated from three components: the propagation delay D_{prop} given by the time for the physical signal to travel from sender to receiver, the transmission delay D_{trans}, which is the time for the sender between sending the first and the last bit of the packet, and the queuing delay D_{queue} the packet has to wait in the queue of a node before it can be forwarded. So the total delay in a node n $D_{total}(n)$ is given by Equation 5.

$$D_{total}(n) = D_{prop} + D_{trans} + D_{queue} \tag{5}$$

The propagation delay D_{prop} is depending on the distance between the sender and receiver node. This varies significantly in a MANET, but as we assume radio communication with signal speed close to speed of light. Thus, the propagation delay between two neighbor nodes, which are less than $1{,}000\,m$ apart from each other, is approximately 3.33×10^{-6}s and thus negligible.

The transmission delay D_{trans} depends on the length of the packet and the bandwidth of the link. For estimation, we assume the maximum size of a packet, which can be retrieved from the SNMP agent in the MO “ifMtu” for each interface available. Since this object gives the size in octets, it needs to be multiplied by 8 to get bits. The bandwidth again can be found in the MO “ifSpeed”. Therefore, D_{trans} is estimated by Equation 6.

$$D_{trans} \leq \frac{ifMtu\,(Octet) \times 8}{ifSpeed} \tag{6}$$

The queuing delay D_{queue} can be determined by counting the number of packets already awaiting transmission in the sent buffer. This number is also available with the SNMP agent in the MO “ifOutQLen”. The queuing delay can be assumed as given in Equation 7.

$$D_{queue} \leq ifOutQLen \times D_{trans} \tag{7}$$

Then, the total delay at node n can assess using Equation 8.

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

The end-to-end delay D_{ijd} for the path P_{ijd} is calculated according to Equation 9 and must satisfy the delay constraints D_{max}. Otherwise, the FANT is dropped because the QoS requirement cannot be satisfied.

$$D_{ijd} = \sum_{n \in P_{ijd}} D_{total}(n) \leq D_{max} \tag{9}$$

2) *Packet Loss:* Packet loss can be assessed by the ratio between the number of lost packets divided by the total number of packets received and sent through a period of time, known Monitoring Window (ΔT). A high value can indicate a congestion problem, bad channel quality caused by fading and interference especially on wireless links, or a hardware problem. These values can again can be retrieved from the SMNP agent (cf. table 1).

Each node can compute the received incoming packets $Pkts_{Rx}$ using the MOs “ifHCInUcastPkts”, “ifHCInMulticastPkts” and “ifHCInBroadcastPkts” using to Equation 10.

$$Pkts_{Rx} = \text{ifHCInUcastPkts} + \text{ifHCInMulticastPkts} + \text{ifHCInBroadcastPkts} \tag{10}$$

The number of successfully sent packets (PktsTx) can be computed accordingly based on the MOs “ifHCOutUcastPkts”, “ifHCOutMulticastPkts” and “ifHCOutBroadcastPkts” using to Equation 11.

$$Pkts_{Tx} = \text{ifHCOutUcastPkts} + \text{ifHCOutMulticastPkts} + \text{ifHCOutBroadcastPkts} \tag{11}$$

Equation 12 gives the number of incoming packets that are dropped, which are computed based on MOs “ifInDiscards”, “ifInErrors”, and “ifInUnkownProtos”.

$$\text{DropPkts}_{in} = \text{ifInDiscards} + \text{ifInErrors} + \text{ifInUnkownProtos} \tag{12}$$

Equation 13 gives the number of outgoing packets that are dropped, which are computed based on MOs “ifOutDiscards” and “ifOutErrors”.

$$\text{DropPkts}_{out} = \text{ifOutDiscards} + \text{ifOutErrors} \tag{13}$$

With these computed values, each node n can calculate the packet loss ratio ($L\ (n)$) as shown in Equation 14. The optimal value to refresh $L\ (n)$ has been verified by multiple simulation runs and is set to $\Delta T = 5s$ by calculating the difference between successive values ($MO\ =\ MO_{new} - MO_{old}$).

$$L\ (n) = \frac{\text{DropPkts}_{in} + \text{DropPkts}_{out}}{Pkts_{Rx} + Pkts_{Tx} + \text{DropPkts}_{in}} \tag{14}$$

The FANT carries the desired success rate (S_{min}) for packet transmission and the expected success rate (S) at the previous node. Each node refreshes S according to Equation 15. If S_{ijd} falls below the desired success rate (S_{min}), then FANT will be dropped. Otherwise, FANT will be forwarded with the updated S field.

$$S_{ijd} = \prod_{n \in P_{ijd}} \left(1 - L(n)\right) \geq S_{min} \tag{15}$$

5. Estimate QoS threshold

Once the data flow begins from the source to the destination, the path needs a constant monitoring in order to keep track of its quality. If the data is forwarded on the link that no longer provides the desired minimum quality, then this 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 containing the end-to-end QoS parameter, such as D_{ijd}, and S_{ijd} and QoS desired, such as B_{min}, D_{max}, and S_{min},. On the basis of these values, the residual value of the QoS parameters can be calculated based on the subsequent equations and all these values are set in BANTs:

$$QB = B_{min} \tag{16}$$

$$QD = \frac{D_{max} - D_{ijd}}{hop + 1} \tag{17}$$

$$QS = \frac{S_{ijd} - S_{min}}{hop + 1} \tag{18}$$

Where hop is the number of intermediate hops from the source node to the destination. After receiving BANTs at the intermediate node, the node can compute the QoS threshold for each flow and store these values in the routing table. The QoS threshold for bandwidth (B_{thr}), delay (D_{thr}), and successful rate (S_{thr}) for each flow at node n can be computed based on the following equations.

$$B_{thr}(n) = QB \tag{19}$$

$$D_{thr}(n) = D_{total}(n) + QD \tag{20}$$

$$S_{thr}(n) = (1 - L\,(n)) - QS \tag{21}$$

6. QoRA Routing Phases

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.

6.1 Forward Phase

First, the source node of the session enquires its routing table to determine the availability of routing information for the requested destination. In case no allocated route is available, this node performs the forward phase. In this phase, the node broadcasts an FANT packet into the network to find the best routes to the destination. 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 this FANT before in order to avoid loops. Second, if the node cannot satisfy the required QoS, 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 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 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 limited by dropping FANTs that followed suboptimal paths. Additionally this policy increases the reliability as a mesh network is created between source and destination with multiple non-disjoint paths. The FANT goodness value is calculated using Equation 22.

$$\psi_{sji} = \frac{[S_{sji}]^{\beta_S}}{[D_{sji}]^{\beta_D}} \tag{22}$$

6.2 Backward Phase

The backward phase starts when the FANTs arrive at the destination. The arrived FANT is converted into a backward ant (BANT). After that, the destination node computes residual values for bandwidth (QB), delay (QD), and expected success rate (QS), as shown in Section 5. The destination node sends the BANT to all the neighbors from which it received the FANT. At destination and intermediate node, the BantStack is popped to assign the next node that has to be forwarded. Throughout the route, the BANT collects quality information about each link in the path and refreshes the routing tables using this quality information, as well as computes the probabilities ($\mathcal{P}_{ijd}$) using Equation 3, updates the pheromones (τ_{ijd}) using Equation 1, and calculates the QoS thresholds, B_{thr}, D_{thr}, and S_{thr} for each flow (see Section 5). The entire process is repeated if the source node loses valid paths for the destination while data still need to be sent. Figure 2 depicts the flowchart of the route discovery Forward/Backward Phases.

Figure 3 shows a simple scenario that is used to explain the calculation of residual QoS values on the route. 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 requested by an application in order to provide QoS, namely, $B_{min} = 2Mbps$, $D_{max} = 150ms$, $S_{min} = 90\%$. In the forward phase, the FANT is broadcasted at the base data rate i.e. the base data rate for IEEE 802.11b equal 1Mbps. The delay observed at node i, node j and node k is $22ms$, $12ms$ and $22ms$ 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 $56ms$. 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 route is the multiplication of individual success rates and in our case it is 0.94.

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) is always equal to the desired minimum bandwidth (B_{min}) in our approach. 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, the residual delay is 31.33ms. The residual success rate (QS) is given 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 equals 0.0133.

Now on the link between node d and node k (in the backward phase), 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 $4ms$. Thus, the threshold delay $D_{thr}(k)$ at node k is $35.33ms$ (summation of the residual delay - 31.33ms and the observed delay - 4ms). The threshold success rate $S_{thr}(k)$ at node k is 0.9667 obtained by subtracting the residual expected (0.0133) from the expected success rate (0.98). The process is iterated for all the nodes in the backward phase.

6.3 Packet Forwarding Phase

Data packets are forwarded from the source to the destination based on stochastic decision policy, where the intermediate node takes a new routing decision according to Equation 3 and increments the amount of pheromone using Equation 1. When a data packet arrives at an intermediate node, QoRA reads the flow information in the packet to determine its FlowID. Then, using data in its routing table, it randomly directs the packet to the next node based on the probabilistic roulette-wheel selection (RWS) [26]. All data packets contain a unique FlowID that identifies source, destination, and FlowLabel used as a counter value to identify the data stream/flow. The intermediate node forwards packet by looking at the FlowID of a data packet. For each flow, multiple paths may be recorded in the routing table.

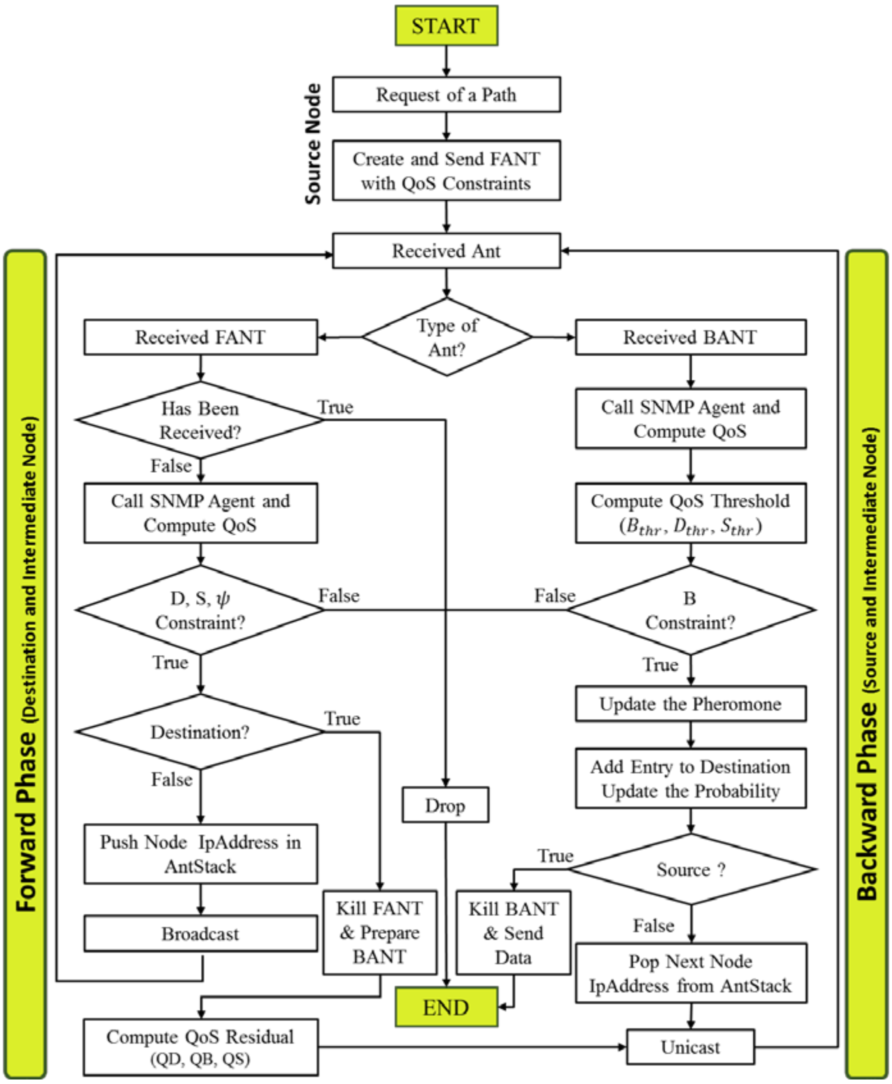

Figure 2. The QoRA Route discovery Forward/Backward Phases

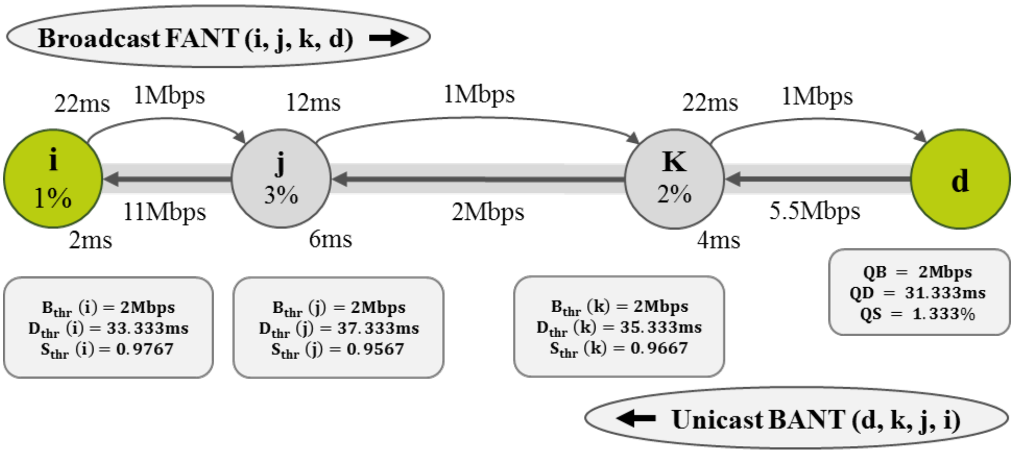

Figure 3. Example explaining the calculation of residual and threshold of QoS values

6.4 QoS Monitoring Phase

The suitability of a route should be monitored during the existence of a flow. Congestion is a very common problem in communication networks. It occurs when the network fails to handle the offered external load. In this situation, if the necessary steps are not taken to minimize the traffic flow into the network, 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 any more. The congestion problem occurs when the transmission delay and packet loss increase and exceed the specified thresholds in Equations 20 and 21. The auto rate protocol also reduces the link transmission speed when the distance between two adjacent nodes increases. As a result, QoRA can avoid and predict link failures by monitoring the decreasing channel transmission speed and comparing to the specified threshold using Equation 19. For each packet forwarding, QoRA calls the SNMP agent to compute QoS parameters locally and compares these values with the QoS threshold for the particular flow. If the required QoS cannot be maintained through a period of time known as monitoring window (ΔT), then the affected node will broadcast EANT to inform the previous node about a congestion problem.

For packet loss QoS violation the node is kept under the observation for the period of time ΔT. If the desired packet success rate cannot be maintained through this period, then the affected node will inform the previous node about a congestion problem by broadcasting EANT. For bandwidth and delay QoS constraints the QoS violation is triggered after continuous violations measured by Counter CU as shown in the Figure 4. In the first time period, the counter measures 3 continuous violations and after that the violations are not observed. Therefore, the node does not triggered QoS violation but in the second period the node register QoS violation as 5 continuous violations are measured by Counter. In this way, the monitoring phase addresses the problem of congestion and broadcast EANT.

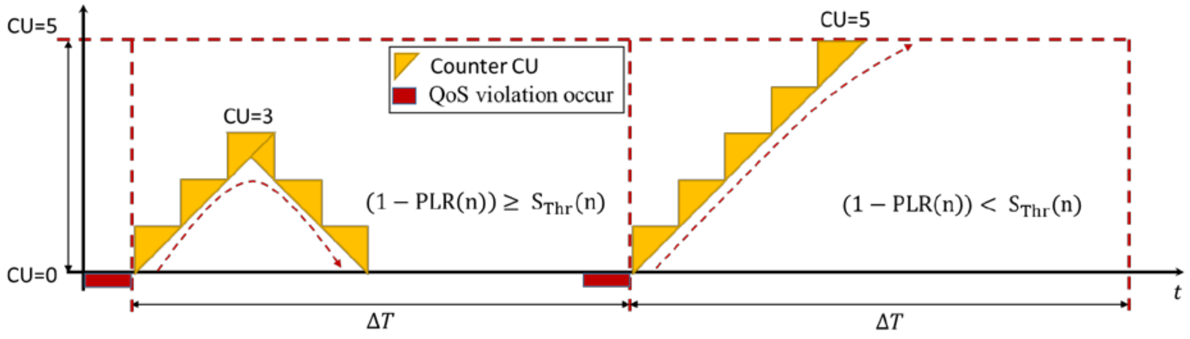

Figure 4. Example explain the QoS violation

6.5 Link Failure Phase:

Every intermediate node that forwards data packets to the destination should keep track of its connectivity to the immediate operational next hops. This connectivity information can be obtained by the available link or network layer mechanisms. IEEE 802.11 uses acknowledgments (ACK) and an optional handshaking mechanism (RTS/CTS) to indicate successful communications. The MAC layer considers 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.

When the QoRA detects the loss of a link to a neighboring node, it deletes the information about the neighboring node from the neighbor table. Then, the node updates its routing table to identify the routes that become invalid because of the link failure and checks another path for the affected flow. In case that the node does not have another path, it adds the flow information (FlowID) to the EantStack. Then, the node broadcasts EANT to inform the previous nodes in the path about link failures or congestion occurrences. Based on the receiving EANT, the intermediate nodes update the routing tables of the affected flows and check another path for these flows. If the node does not have an alternative path for the affected flows, so it adds this FlowID to EantStack and rebroadcasts EANT. Otherwise, the node drops the EANT. When the EANT arrives at the source node, it updates its routing table for the affected flow and broadcasts a new FANT to identify a new path to the destination.

7. Implementations and Simulation Results

For the QoRA approach, we developed a simulation model within the framework of the network simulator (ns-3) [27], and the simulation results were compared with the popular topology-based routing protocol AODV and geographical-based routing protocol CLWPR [12, 13]. We decided to compare our approach with AODV and CLWPR because some other ACO multipath-routing protocol has not been implemented in ns-3 yet.

7.1 QoRA Approach Implementation:

The QoRA approach consists of two main modules. The first one is the QoRA Entity that consists of a number of classes. The first and main class is the QoraDecisionEngine that is inherited from the ns-3 Ipv4RoutingProtocol and it implements the protocol. Pheromone trail values are stored in QoraNeighbourTable and each neighbor is listed along with the pheromone concentration indicating the experienced goodness of this link to several destinations. QoraRoutingTable performs the actual routing. An entry from this table is saved in the sub-class QoraRoutingTableEntry and contains information about the route from node i to destination d via neighbor j and each entry for a destination is also associated with a list of neighbor nodes. QoraAntsPacket, which inherits from ns-3 Header, is responsible to generate three type of ants: forward ant FANT, backward ant BANT, and error ant EANT. Also, QoraQueue is used to save the data packet at the source node until the establishment of the route to the destination. The final class is QosManager, generating the commands for the following SNMP operations namely Get, Set, and Trap and also calculating the QoS parameters.

The second module is the SNMP Entity, which we developed for ns-3 as it has not been available in the ns-3 yet. In this section, we briefly explain the implementation of SNMP in ns-3, which includes the SNMP protocol and the interface MIB group. The SNMP implementation consists of several classes. The first class is SnmpAgent, which handles the command responder, the notification originator and provides access to the MIB. The second class SnmpPDU is used to encapsulate the SNMP protocol data unit (PDU) message. Managers and agents use this message format to exchange information. The implementation of the MIB interface in ns-3 consists of further classes. The first class is the MIBHandler, which handles the interaction between the SNMP entity and the MIB. This class is based on the ns-3 tracing system [27] that consists of both tracing sources and tracing sinks. Trace sources are connected to the other pieces of code that so that they can utilize the information provided by the source, such as counting the number of incoming packets $Pkts_{Rx}$, outgoing packet $Pkts_{Tx}$, or dropped packets $DropPkts_{in}$ and $DropPkts_{out}$. A trace source also informs about a state change, such as the change in the data rate speed ifSpeed and the size of interface output queue. The second class is ifTable that defines the management object ifTable. This class contains a list of interface entries that are defined in the sub class ifTablEntry. The ifTablEntry class includes some of the management objects used to assess the quality of various outgoing links as shown in Table 1.

7.2 Simulation Environment and Settings:

The simulated network consists of 200 nodes/vehicles with varying mobility from 5 to 30 m/s. All nodes were configured according to the standard ns-3 YansChannel using the IEEE 802.11 standard. The channel speed over the wireless link was controlled depending on RBAR [25] using ns-3 IdealWifiManager. The channel rates varied with different ranges according to the signal-to-noise ratio of the neighbor nodes.

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. Application data transfer starts at the random time in the interval between 10 to 40 s of the simulation time and continues to the end of the simulation. The simulation runs for 250 s. The simulation was repeated multiple times for each scenario with an average of 20 runs per step using different seeds.

Table 2 shows the parameters settings for different scenarios for MANETs and VANETs. Additional, Table 3 shows the parameter settings that define the different weight factors used by the QoRA approach; these values are determined by testing different values and contain the QoS constraints.

Table 2. Parameters settings for different scenarios for MANETs and VANETs

Parameters	MANETs scenario	VANETs scenario
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
simulation area	1000m x 2000m	2000m x 2000m
Type of Mobility	ns-3: “Random Waypoint Mobility.”	Bonnmotion tool: “5x5 Manhattan Grid Mobility.”
Propagation loss model	Friis	Two Ray Ground

Table 3. Parameter settings that used by the QoRA approach

Parameters	Value	Parameters	Value
Pheromone increment factor $\Delta\tau_{ij}$	0.02	Delay weight factor β_D	0.7
Pheromone weight factor α	0.4	Monitoring Window $MW = \Delta T$	5S
Heuristic weight factor β	0.6	Bandwidth constraint B_{min}	2-3 Mbps
Bandwidth weight factor β_B	0.5	Delay constraint D_{max}	150-250ms
Expected success rate factor β_S	0.5	Desired success rate S_{min}	95%-100%

7.3 Simulation Results and Analysis:

This section presents the results obtained from MANETs and VANETs scenario as shown in Figure 5 and 6. We compared the results between AODV, CLWPR, and QoRA for both MANETs and VANETs against the following four parameters.

- **Packet delivery ratio (PDR):** It is defined as the ratio of the number of data packets successfully received by the destination node to the number of data packets generated by the source node. The robustness of a protocol can be determined in terms of PDR. Figures 5A and 6A shows the packet delivery ratio for MANETs and VANETS. 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 PDR reduces as expected, due to the frequent link failures.
- **Throughput:** It is defined as the ratio of the total number of data packets delivered to the destination to the data packet delivery time. Figures 5B and 6B shows 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 that increases the value of this metric stochastically by spreading the data packets through the multipath. As the mobility increases, the throughput decreases due to frequent link failures.
- **Average end-to-end delay:** It indicates the time required to transfer the data packets from the source node to the destination node. This term includes different delays occurring due to buffering, route discovery phase, queuing, contention, and propagation. Figures 5C

and 6C indicates that QoRA performs much better than AODV even at high mobility. However, CLWPR outperforms QoRA due to the fact that CLWPR is a location based proactive routing protocol, whereas QoRA is an on-demand routing protocol.

➢ **Jitter:** It is defined as the inconsistency in packet delay observed at the receiving node due to network congestion, route changes, queuing, etc. It is a very crucial metric that influences the quality of real-time applications such as voice and video. Figures 5D and 6D shows that for jitter parameter, the QoRA approach again outperforms the AODV protocol. Here, CLWPR again shows better performance due to its proactive nature.

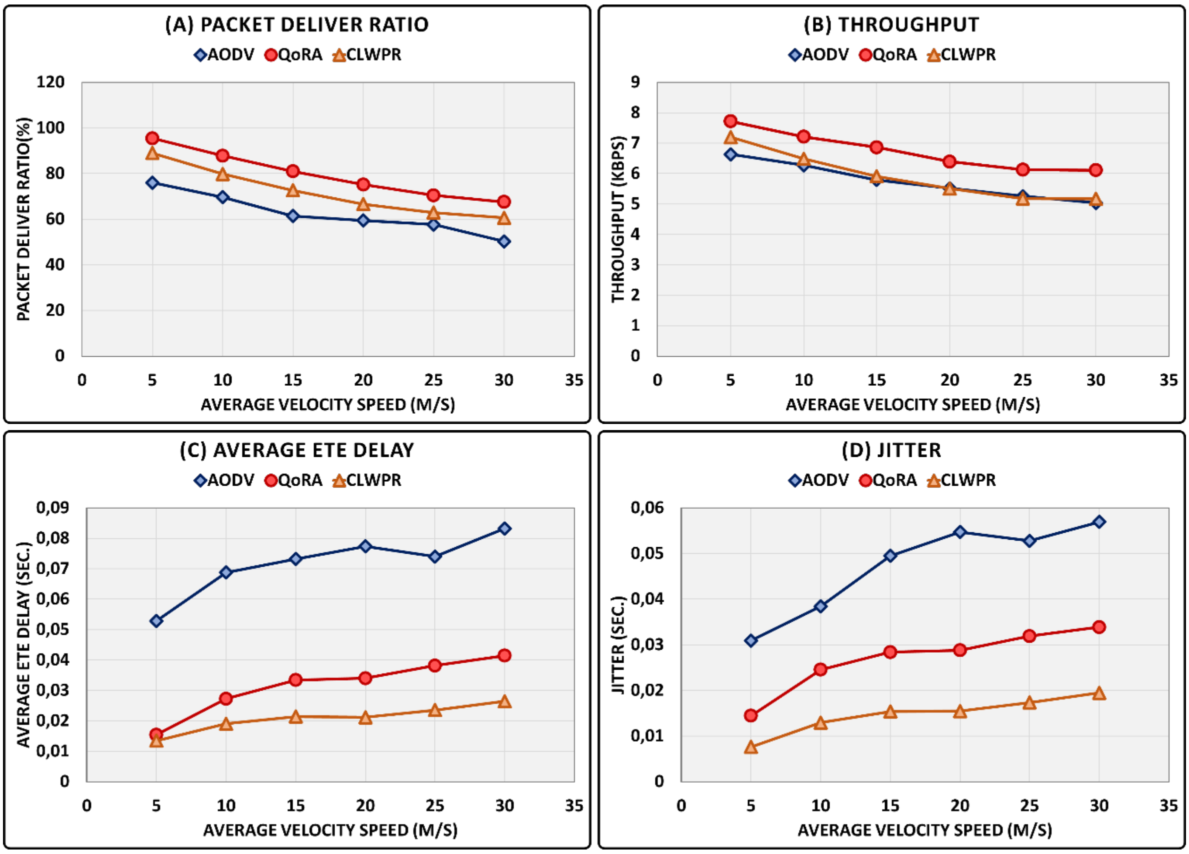

Figure 5. The first scenario for MANET performance

The main reason that QoRA outperforms AODV all the four parameters is due to its unique design approach. QoRA selects the paths with high transmission speed and minimum delay as compared to the selection of a path with a minimum number of hops as determined by AODV. The approach followed by AODV in multi-rate networks is not desirable as a path with less number of hops might not be optimal path in terms of transmission speed. The reason that CLWPR outperforms QoRA for jitter and end-to-end delay 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. The QoRA approach does not employ any periodic control message in contrast to the hello messages used by AODV and CLWPR, which reduces the network overhead to a great extent. This method also minimizes the probability of packet collision and interference with regular data transmission. CLWPR also proposes the carry-n-forward mechanism as a link repair strategy. However, this method increases the 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 [13]. 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.

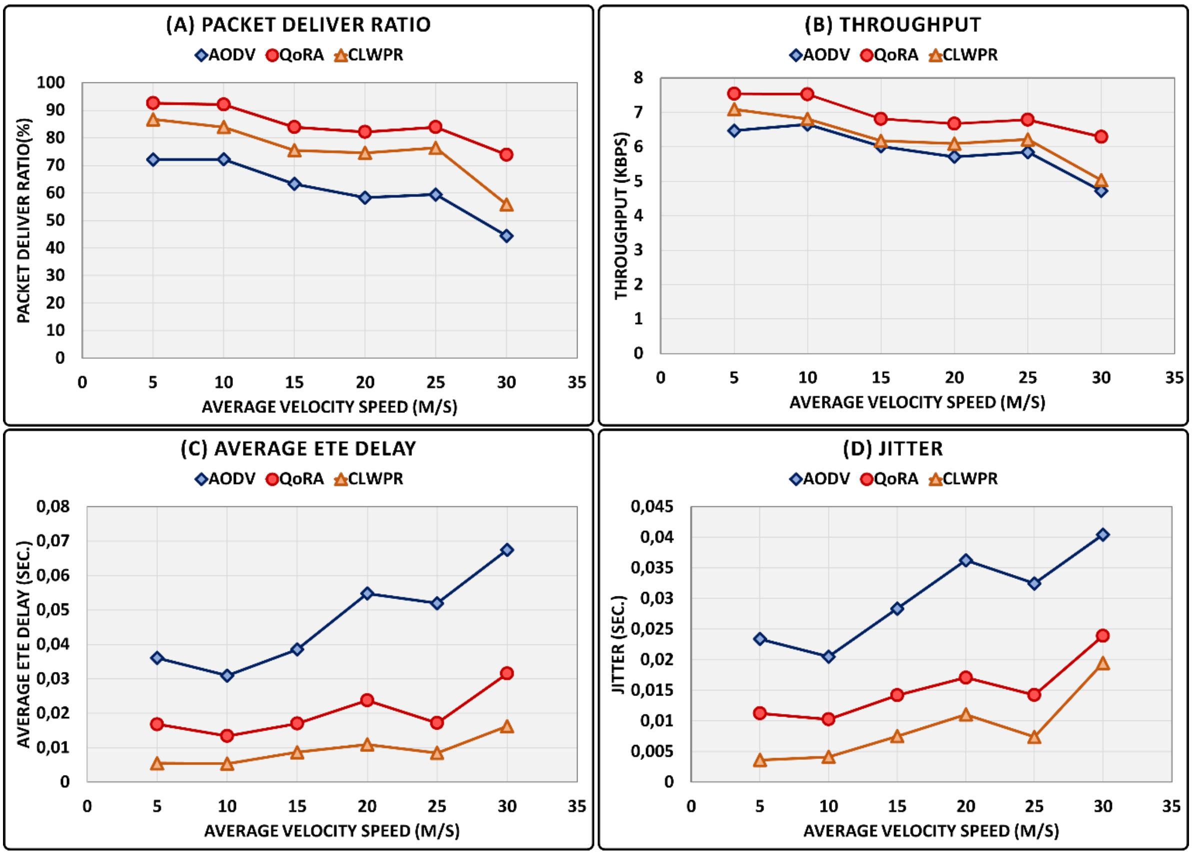

Figure 6. The second scenario for VANET performance

8. Conclusion

In this paper, we presented the simulation results of our proposed routing protocol QoRA along with the QoS computation model and the congestion avoidance mechanism using the SNMP agent. With the help of the SNMP agent, the QoS parameters are computed locally without exchanging any additional control message and without synchronization. Our approach thus evades any network overhead for QoS computation as compared to the other QoS-aware routing protocols. As an additional QoS measure, our approach utilizes the parallel and global search abilities of ACO to find multiple paths to the destination satisfying the specified QoS requirements. We compared our simulation results against the four metrics that confirm the ability of QoRA to satisfy successfully the QoS requirements.

In the future, we intend to compare the proposed model with other QoS-aware routing protocols based on the ACO algorithm and use more realistic VANET topologies and multimedia applications such as a video trace source file.

Acknowledgement

This work has been carried out within the Communication Networks Group at Technische Universität Ilmenau, partially supported by German Academic Exchange Service DAAD (Deutscher Akademischer Austausch Dienst), and University of Technology-Iraq.

References

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

[2] M. K. Gulati and K. Kumar, "QoS routing protocols for mobile ad hoc networks: a survey," International Journal of Wireless and Mobile Computing, vol. 5, no. 2, pp. 107–118, 2012. http://dx.doi.org/10.1504/IJWMC.2012.046783

[3] X. Huang and Y. Fang, "Multiconstrained QoS multipath routing in wireless sensor networks," Wireless Networks, vol. 14, no. 4, pp. 465–478, 2008. http://dx.doi.org/10.1007/s11276-006-0731-9

[4] M. Balachandra, K. Prema, and K. Makkithaya, "Multiconstrained and multipath QoS aware routing protocol for MANETs," Wireless Networks, vol. 20, no. 8, pp. 2395–2408, 2014. http://dx.doi.org/10.1007/s11276-014-0754-6

[5] M. Obaidat, M. Ali, I. Shahwan, M. S. Obaidat, and S. Obeidat, "QoS-Aware Multipath Communications over MANETs," Journal of Networks, vol. 8, no. 1, pp. 26–36, 2013. http://dx.doi.org/10.4304/jnw.8.1.26-36

[6] M. Ali, B. Stewart, A. Shahrabi, and A. Vallavaraj, "QoS Aware Multipath Threshold Routing for Mobile Ad hoc Networks," International Journal of Applied Information Systems (IJAIS), vol. 7, no. 1, pp. 8–15, April 2014. http://research.ijais.org/volume7/number1/ijais14-451118.pdf

[7] C. Lal, V. Laxmi, M. S. Gaur, and S.-B. Ko, "Bandwidth-aware routing and admission control for efficient video streaming over MANETs," Wireless Networks, vol. 21, no. 1, pp. 95–114, 2015. http://dx.doi.org/10.1007/s11276-014-0774-2

[8] W. Chen, R. K. Guha, T. J. Kwon, J. Lee, and Y.-Y. Hsu, "A survey and challenges in routing and data dissemination in vehicular ad hoc networks," Wireless Communications and Mobile Computing, vol. 11, no. 7, pp. 787–795, 2011. http://dx.doi.org/10.1002/wcm.862

[9] Z. Niu, W. Yao, Q. Ni, and Y. Song, "DeReQ: a QoS routing algorithm for multimedia communications in vehicular ad hoc networks," in International Conference on Wireless Communications and Mobile Computing (IWCMC'07), Honolulu, Hawaii, August 12–16 pp. 393–398, 2007. http://dx.doi.org/10.1145/1280940.1281025

[10] M. Asefi, J. W. Mark, and X. Shen, “An Application-Centric Inter-Vehicle Routing Protocol for Video Streaming over Multi-Hop Urban VANETs,” in IEEE International Conference on Communications (ICC), Kyoto, Japan, June 5–9 2011, pp. 1–5. http://dx.doi.org/10.1109/icc.2011.5962635

[11] S. Xu, P. Guo, B. Xu, and H. Zhou, “QoS Evaluation of VANET Routing Protocols,” Journal of Networks, vol. 8, no. 1, pp. 132-139, 2013. http://dx.doi.org/10.4304/jnw.8.1.132-139

[12] K. Katsaros, M. Dianati, R, Tafazolli and R. Kernchen, “CLWPR – A Novel Cross-Layer Optimized Position Based Routing Protocol for VANETs”, Proceedings of IEEE Vehicular Networking Conference (VNC), pp. 139- 146, Nov. 2011. http://dx.doi.org/10.1109/VNC.2011.6117135

[13] K. Katsaros, M. Dianati, and K. Roscher, “A Position-based Routing Module for Simulation of VANETs in NS-3,” Proceedings of the 5th International Conference on Simulation Tools and Techniques (ICST), pp. 345-352, 2012. http://dl.acm.org/citation.cfm?id=2263074

[14] S. Bitam, A. Mellouk, and S. Zeadally, “Bio-Inspired Routing Algorithms Survey for Vehicular Ad-hoc Networks,” Communications Surveys Tutorials, IEEE, vol. 17, no. 2, pp. 843–867, 2015. http://dx.doi.org/10.1109/COMST.2014.2371828

[15] 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, vol. 2011, no. 153, pp. 1–12, 2011. http://dx.doi.org/10.1186/1687-1499-2011-153

[16] 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, vol. 6, no. 1, pp. 76–83, 2012. http://dx.doi.org/10.1049/iet-com.2010.0763

[17] S. Kim, “An Ant-based Multipath Routing Algorithm for QoS Aware Mobile Ad-hoc Networks,” Wireless Personal Communications, vol. 66, no. 4, pp. 739–749, 2011. http://dx.doi.org/10.1007/s11277-011-0361-0

[18] V. Balaji and V. Duraisamy, “Ant Optimized Link Quality for Ad Hoc on Demand Distance Vector,” Wireless Personal Communications, vol. 79, no. 1, pp. 763–771, 2014. http://dx.doi.org/10.1007/s11277-014-1885-x

[19]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. http://searchdl.org/public/book_series/AETS/8/119.pdf

[20] M. Dorigo, M. Birattari, and T. Stutzle, “Ant colony optimization,” Computational Intelligence Magazine, IEEE, vol. 1, no. 4, pp. 28 –39, Nov 2006. http://dx.doi.org/10.1109/MCI.2006.329691

[21] J. Case, R. Mundy, D. Partain, and B. Stewart, “Introduction and Applicability Statements for Internet Standard Management Framework,” IETF Network Working Group, Request for Comments RFC 3410, December 2002. http://dx.doi.org/10.17487/RFC3410

[22] S. A. Routhier, “Management Information Base for the Internet Protocol (IP),” IETF

Network Working Group, Request for Comments RFC 4293, April 2006. http://dx.doi.org/10.17487/RFC4293

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

[24] A. Baruch, H. David, and R. Herbert, “High throughput route selection in multi-rate ad hoc wireless networks,” Wireless On-Demand Network Systems. Springer Berlin Heidelberg, 2004. 253-270. http://dx.doi.org/10.1007/978-3-540-24614-5_19

[25] G. Holland, N. Vaidya, and P. Bahl, “A rate-adaptive mac protocol for multi-hop wireless networks,” in ACM MOBICOM, 2001, pp. 236–251. http://dx.doi.org/10.1145/381677.381700

[26] T. Back, Evolutionary Algorithms in Theory and Practice: Evolution Strategies, Evolutionary Programming, Genetic Algorithms. Oxford University Press, USA, 1996.

[27] The ns-3 network simulator. [Online]. Available: http://www.nsnam.org/

Copyright Disclaimer