Received: December 9, 2021. Revised: January 21, 2022.

INASS

International Journal of
Intelligent Engineering & Systems

http://www.inass.org/

QoS-Satisfied Multicast Routing Protocol for MANETs

Suganya Rangasamy[1]* **Vasantha Kalyani David**[1]

[1]*Department of Computer Science, Avinashilingam Institute for home Science and Higher Education for Women, Coimbatore, India*
* Corresponding author's Email: rsugu1983@gmail.com

Abstract: Mobile Adhoc Network (MANET) can operate with multiple data rates for Quality-of-Service (QoS)-constrained multimedia communication. For such services, QoS-sensitive multicast protocols are necessary. A Delay-Sensitive Multicast Tree was built in many works in literature that estimates one-hop delay by using the measure of busy/idle fraction of the distributed radio channel and also adjusts the transmission data rates while reducing the end-to-end delay. But, different metrics rather than delay can be considered furthering enhancing the network performance. Therefore in this article, QoS-Satisfied Multicast (QSSM) routing protocols is proposed which considers additional network metrics like bandwidth utilization, packet drop rate and jitter to select the multicast route for data transmission. First, the considered metrics for each node are collected from the neighboring nodes. After that, multiple QoS-satisfied multicast trees are constructed using these metrics where each multicast tree can satisfy the predefined percentage of the QoS requirements. So, the overall delay and bandwidth consumption are minimized by ensuring the QoS guarantees a requested traffic flow and ongoing flows. Further, a randomized network coding scheme is applied to remove the unwanted data packets that increase the network capacity efficiently for multicasting. Finally, the simulation results exhibit that the QSSM protocol achieves a 0.62sec one-hop delay and 0.06sec end-to-end delay for the simulation time of 200sec compared to the other protocols.

Keywords: MANET, Delay-sensitive protocol, QoS, Network capacity, Randomized network coding.

1. Introduction

Generally, MANETs are one of the categories of wireless networks and constructed without the help of any backbone infrastructure for achieving data transmission between each mobile node. Each node in the network can act as both router and host. The difficulties and limitations in conventional wireless networks are more significant in MANETs because of their dynamic topological changes, power, bandwidth and delay constraints. Few challenges are intrusions, collisions, congestion and privacy issues. These networks can be deployed in many applications such as distributed computing, disaster management, mobile access web, defense, smart transportation, healthcare monitoring, smart home, smart city, etc. Nowadays, the deployment of these networks has been encouraged vastly to attain the QoS guarantees as needed by the application based on its scenario requirements. Though these networks increase QoS performance, it needs an efficient route selection algorithm to select the most optimal path to transmit the data packets while multiple QoS guarantees are not easily achieved [1-2].

Usually, routing in MANETs is designed based on the unique addresses of the nodes. It generates the routing path with the number of intermediate nodes from the source to the destination nodes. Typically, routing protocols are either unicast or multicast which depends on the method used for data transmission [3]. Unicast transmission can deliver the data packets to only one destination node whereas multicast transmission can deliver the data packets simultaneously to the group of destination nodes [4]. Such multicast routing in MANETs can be executed in the network layer, the MAC layer and the application layer. Thus, the Multicast Routing Protocols (MRPs) are divided into three categories, namely Network (IP) Layer Multicast

(IPLM) [5], Application Layer Multicast (ALM) [6] and MAC Layer Multicast (MACLM) [7]. Normally, multicasting can reduce the channel capacity utilization, sender and router processing, power consumption and transmission delay. The modern tendency in data-distribution applications is transmitting the data packets through multicast routing [8]. To achieve this multicast routing, a multicast tree is created which is more efficient than transmitting the same data packets separately from source to multiple destinations. Nowadays, many applications rely on real-time multicast transmission services [9].

In these multicast transmission services, delay-sensitive multicast protocols are the most essential while delay requirements of the requested multicast services can be satisfied with certain confidence levels i.e., a specific proportion of data packets whose end-to-end delay is less than the fixed values. In the past years, many delay-aware MRPs have been proposed for increasing the performance of MANETs. Among those protocols, Chen et al. [10] proposed a DSM protocol for enhancing the network capacity in MANETs. At first, a method was proposed to estimate the one-hop delay based on the measure of the busy/idle ratio of the distributed radio channel. Then, a DSM protocol was proposed by constructing the multicast tree that reduces the sum of the total transmission time of the forwarders and the total blocking time of the blocked hosts with the utilization of neighboring data of the forwarders. Also, the transmission data rates of the forwarders were properly fine-tuned to reduce the resource consumption and increase the network capacity such that more traffic flows were permitted into the network. On the other hand, it needs to consider additional metrics other than delay to further increase the QoS performance.

Hence, the main aim of this article is to increase the QoS performance of the network by considering other network metrics during multicast routing decisions. The QSSM protocol is proposed that uses other metrics such as bandwidth, packet dropping rate and jitter. In this protocol, these metrics are gathered from the adjacent nodes. Then, multiple QoS-satisfied multicast trees are built by using these metrics where each multicast tree can satisfy the predefined proportion of the QoS requirements. As a result, the total delay and bandwidth utilization are effectively decreased when QoS guarantees to a requested traffic flow and ongoing flows are achieved. Moreover, the redundant data packets are effectively removed by applying a randomized network coding. Thus, each destination node can receive the new coded packets from different routes via multicast routing.

The remaining part of this article is prepared as follows: Section 2 presents the survey on previous researches related to the delay-aware multicast routing in MANETs. Section 3 explains the methodology of the QSSM protocol in detail. Section 4 shows the simulation results of the QSSM protocol compared with the existing protocol. Section 5 concludes the entire research work.

2. Literature survey

Kharraz et al. [11] proposed an efficient path discovery process for improving the performance and multicast efficiency of On-Demand MRP (ODMRP). In this protocol, the flooding mechanism was effectively managed based on the delay characteristics of the contributing nodes. Only the nodes that satisfy the delay requirements can flood the Join-Query messages. Also, the contributing nodes were modeled as M/M/1 queuing systems. However, it was not able to select the optimal paths for improving the multicast routing.

Walikar & Biradar [12] proposed an Energy-Aware MRP (EAMRP) to maximize end-to-end connectivity in the network and minimize link failure. In this protocol, a set of multiple routes were established from the source to multicast destinations by using an energy-efficient adjacent node selection scheme. It has different steps such as computing the remaining energy of a node using node energy model, pruning the nodes with remaining energy lower than the threshold value, finding multiple paths to the destination by using request and reply packets, selecting the stable paths by taking remaining energy of the nodes and preventing the paths from the link/node failures due to energy drain. However, the end-to-end delay of this protocol was still not reduced efficiently.

Gopinath & Nagarajan [13] proposed a residual energy-based reliable MRP for increasing the network lifetime and packet delivery ratio. In this protocol, a multicast backbone was created for achieving more stability based on the node awareness and trustable loop. Moreover, a reliable route criterion was measured for selecting the best reliable route among all available routes and transmitting the packets to the destination node. Conversely, the routing reliability and network reliability were still not effective.

Robinson et al. [14] proposed a bandwidth and delay-aware routing protocol with a scheduling algorithm for multihop MANET for preventing link failures. The major objective of this protocol was integrating the routing protocol with priority-based traffic management to achieve the end-to-end reservation and distributed transmission scheduling to allocate the channels efficiently. Conversely, this protocol has a high delay and a very less packet delivery ratio.

Singal et al. [15] proposed a Multi-Constraints Link Stable MRP (MCLSMRP) in MANETs. In this protocol, a QoS-aware metric was used to determine the stable link among all incoming links based on the link stability factor. The stability factor was estimated by using contention count, received signal strength and hop count as QoS parameters. Then, the node with the highest stability factor was chosen as a forwarding node. Similarly, the link with the highest link stability factor was chosen as the most stable link. However, an appropriate contention count should choose to achieve well connectivity and availability of resources.

Yadav et al. [16] proposed an Efficient Fuzzy-based Multi-constraint MRP (EFMMRP) that considers multiple QoS performance constraints such as end-to-end delay, channel bandwidth and energy. In this protocol, such constraints were converted as a fuzzy cost to assign the multicast route for data transmission between the source and set of destinations having a low value of fuzzy cost. On the other hand, the control overhead was high.

Vinya and Rao [17] developed a novel multicast routing with Particle Swarm Optimization (PSO) to choose the best path in the cognitive radio networks. Also, the data was secured by the RSA algorithm. But, the energy utilization was not reduced. Jayaramu and Banga [18] designed an effective Adhoc On-demand Distance Vector (AODV) routing method with the Biogeography-Based Optimization (BBO) to detect the best path and minimize the transfer period. But, the energy utilization was not reduced efficiently.

El-Sayed et al. [19] designed Multi-objective multicast Dynamic Source Routing (MDSR) protocol based on Ant Colony Optimization (ACO) to find the shortest path and optimize the total weight of the multicast tree. But, the distance cost was still high.

3. Proposed methodology

In this section, the QSSM protocol for MANETs is explained briefly. This protocol can minimize the overall bandwidth utilization and end-to-end delay to a specific degree when providing both delay and bandwidth guarantees to a requested flow and ongoing flows. Multiple multicast trees are built so that the remaining bandwidth is completely used with minimized bandwidth utilization, end-to-end delay, packet loss and jitter. Also, the randomized network coding is applied to remove the unwanted data packets and the new coded packets are received by each destination node via distinct paths.

Given a requested multicast flow in a MANET, the problem of creating the multicast tree that can reduce the end-to-end delay and bandwidth utilization of the traffic flow with guaranteed QoS requirements is called an NP-hard problem. A tree construction algorithm is proposed for constructing the multicast tree at a time. It can choose the hosts with better channel states as forwarders for reducing the overall end-to-end delay and bandwidth utilization. Each constructed tree can link few destinations and provide a predefined percentage of QoS requirements of each linked destination. This process is repeated until all destinations are QoS satisfied.

This QSSM protocol considers the remaining bandwidths, packet loss rate, jitter, residual delay and bandwidth utilization of the forwarders and their one-hop or two-hop neighbors while constructing trees for providing bandwidth and delay i.e., QoS guarantees to all traffic flows.

3.1 Proposed methodology

A network is initialized with n number of nodes wherein h_S is the source node and h_D is the destination node. Also, a set of forwarder nodes $\mathcal{F} \in h_i, h_j, h_k$ is initialized. Consider 3 different sets of destination nodes D, D', D'' for multicast transmission. All the three different destination sets of traffic flow (Υ) is defined by $\ddot{D}$. During initialization, $D = \emptyset, D' = \ddot{D}, \mathcal{F} = \{h_S\}$. The set D' is the destinations whose QoS requirements are not fulfilled and it is modified by eliminating QoS-satisfied destinations when a new tree is created. Similarly, D'' is the destinations in D' that were added to the tree.

Similarly, Q_{req} is the QoS requirements of each destination (QoS includes the values of delay, bandwidth utilization, packet drop rate and jitter), QoS_{fix} is the predefined percentage for QoS requirements $\left(QoS_{req}\right)$. There is a series of 2D arrays $\tilde{Q}\{Q^1, Q^2, \ldots, Q^\varphi\}$ where φ refers to the number of trees constructed.

By considering these assumptions, there is a basic process contained in the QSSM protocol. When it is invoked with a particular destination h_D, a minimum-weight path to h_D from h_S is discovered to give a predefined percentage of the QoS requirement of h_D by constructing the multicast tree. To construct the multicast tree and find the minimum-weight route, different notations are used, which are summarized in Table 1.

Table 1. Parameters used for multicast tree construction and minimum weight path discovery

Parameters	**Description**
h_i, h_j, h_k	Host nodes
Υ	Requested traffic flow
h_S	Source node
h_D	Destination node
h_{D^*}	New destination node
$\ddot{D}$	Set of all destinations of Υ
Q_{req}	QoS requirements of each destination
QoS_{fix}	Predefined % for Q_{req}
D, D', D''	Three sets of destinations
$\mathcal{F}$	Set of forwarders
$\tilde{Q}$	Series of 2D arrays
φ	Number of trees constructed
$p_{i,j}$	Success ratio of $h_i \rightarrow h_j$
$1/p_{i,j}$	Expected number of transmissions needed to successfully deliver the packet over $h_i \rightarrow h_j$
QoS_{def_D}	QoS deficiency of h_D
$U(h_D)$	Set of the upstream forwarders of h_D in the constructed trees
$\tilde{Q}[i, D]$	Total QoS of h_i utilized for h_D in the constructed trees
$Q^t[i, D]$	The QoS of h_i that is utilized for h_D in t^{th} constructed trees
$p_{i,D}$	Success ratio of link$h_i \rightarrow h_D$
$\tilde{Q}[i, D] \times p_{i,D}$	QoS gain of h_D from h_i
t	Iteration number
$\mathcal{G}$	Directed graph
x	Vertex in $\mathcal{V}$ individually corresponds to a host h_i
$arc < h_i, h_j >$	h_j is within the communication range of h_i
$\mathcal{w}_{h_i}, \mathcal{w}_{h_k}, \mathcal{w}_{h_k,h_j}$	Weight values of h_i, h_j, h_k
$\mathcal{w}_{h_j}$	Minimum weight of h_D-to-h_j routes discovered thus far
$\mathfrak{R}$	Novel h_j-to-h_D path through h_k
τ	Maximum gain
η_D	Variable relies on QoS requirements of h_D

The following sections provide details about multicast tree construction and minimum-weight path discovery processes briefly.

4. Multicast tree construction algorithm

Given a requested traffic flow Υ, the tree construction algorithm can create the sequence of one or multiple trees for providing QoS guarantees to the destinations of Υ and enhancing the QoS efficiency. For a host h_i and a destination h_D, $Q^t[i, D]$ records the QoS of h_i that is utilized for h_D in t^{th} constructed tree, where $1 \leq t \leq \varphi$ and $\tilde{Q}[i, D]$ accumulates the QoS of h_i which is utilized for h_D in all constructed trees. At first, this algorithm sets $D = \emptyset, D' = \ddot{D}, \mathcal{F} = \{h_S\}$ and $\tilde{Q} = 0$.

In this *Algorithm 1*, two nested repeat-until-loop statements are used. The outer repeat-until-loop statement is used for constructing all trees. They are all rooted at h_S. Each iteration of this statement creates a tree that links all destinations in D' and has the ability to provide QoS_{fix} percentage of their QoS requirement. Set D' gathers the destinations whose QoS requirements are still not satisfied and it is updated by eliminating the QoS-satisfied destinations whenever a new tree is created. The functioning of this statement ends with Υ admitted while $D' = \emptyset$. After that, $\mathcal{F}$ have the forwarders of all trees.

The inner repeat-until-loop statement is used for constructing the tree that links all destinations inD'. Originally, the tree has the root i.e., h_S only. After that, each iteration of this statement inserts a new destination i.e., $h_{D^*} = D'$ to the tree with the help of the for-loop statement which is used to discover a minimum-weight path for each destination i.e., h_D in $D' - D''$ by invoking the fundamental process, namely $Discovering\_a\_Minimum\_Weight\_Path$ (described in *Algorithm 2*). The weight of a path is referred to the total weight of all links contained in it and the weight of a link $h_i \rightarrow h_j$ is equal to $1/p_{i,j}$, where $p_{i,j}$ denotes the success ratio of $h_i \rightarrow h_j$ and $1/p_{i,j}$ denotes the expected number of transmissions needed to successfully deliver the packet over$h_i \rightarrow h_j$. The minimum-weight path for h_D can provide QoS_{fix} percentage of the QoS requirement of h_D and avoid the multicast routing problem at the same time.

Set D'' gathers the destinations in D' that are inserted into the tree. Therefore, the for-loop statement is performed $|D' - D''|$ times and a total of $|D' - D''|$ minimum-weight paths is constructed. While this process is terminated, the minimum-weight path linking h_{D^*} with $h_{f(D^*)}$ among $|D' - D''|$ minimum-weight paths is determined. Also, $D, D'', \mathcal{F}, Q^t$ and $\tilde{Q}$ are updated before terminating the current iteration of the inner repeat-until-loop statement. This statement has $|D'|$ iterations and its process ends while all destinations in D' are inserted into the tree i.e., $D'' = D'$.

```
Algorithm 1 for tree construction $(h_S, \ddot{D}, QoS_{req}, QoS_{fix})$:
    Begin
    {
```

```
    $D = \emptyset;\ D' = \ddot{D}; \mathcal{F} \leftarrow \{h_S\};$
    Assign $\tilde{Q}[i, D] = 0$ for all hosts $h_i$ and all destinations $h_d$ in $\ddot{D}$;
    $t \leftarrow 0$; //$t$ −iteration number
    Repeat
    $D'' \leftarrow \emptyset; t \leftarrow t + 1;$
    Assign $Q^t[i, D] = 0$ for all hosts $h_i$ and all destinations $h_d$ in $D' - D''$;
    Repeat
    $\boldsymbol{for}(each\ destination\ h_D\ in\ D' - D'')$
    $QoS_{def_D} \leftarrow QoS_{req} - \sum_{h_i \in U(h_D)} \tilde{Q}[i, D] \times p_{i,D};$
   Apply Algorithm 2 to find the minimum weight path;
    $\boldsymbol{end\ for}$
    Decide a minimum-weight path e.g., linking $h_{D^*}$ with $h_{f(D^*)}$ where $h_{D^*} \in D' - D''$ among all $|D' - D''|$ paths obtained previously;
    $D \leftarrow D \cup \{h_{D^*}\};\ D'' \leftarrow D'' \cup \{h_{D^*}\};$
    Insert all hosts in $h_{f(D^*)}$-to-$h_{D^*}$ path to $\mathcal{F}$;
    Allocate each $Q^t[i, D^*]$ with the QoS of $h_i$ that is utilized for $h_{D^*}$ where $h_i$ denotes the host in the $h_S$- to-$h_D$ path through $h_{f(D^*)}$;
    Insert $Q^*[i, D^*]$ to $Q[i, D^*]$ for each host $h_i$ in the $h_S$-to-$h_D$ path through$h_{f(D^*)}$;
    Until $D'' = D'$
    $\boldsymbol{for}(each\ destination\ h_D\ in\ D')$
    $\boldsymbol{if}\left(\sum_{h_i \in U(h_D)} \tilde{Q}[i, D] \times p_{i,D} \geq QoS_{req}\right)$
    $D' \leftarrow D' - \{h_D\};$
    $\boldsymbol{end\ if}$
    $\boldsymbol{end\ for}$
    Until $D' = \emptyset$
    }
    End
```

In this *Algorithm 1*, the minimum-weight path is discovered which can provide QoS_{fix} percent of its QoS requirement for a given destination h_D. Also, three input parameters are used such as $\mathcal{F}, D$ and QoS_{def_D} for the fundamental process where $\mathcal{F}$ gathers the forwarders of the constructed trees, D gathers the destinations that are linked to the constructed trees and QoS_{def_D} measures the QoS deficiency of h_D. The value of QoS_{def_D} is computed as $QoS_{req} - \sum_{h_i \in U(h_D)} \tilde{Q}[i, D] \times p_{i,D}$ where $U(h_D)$ denotes the set of the upstream forwarders of h_D in the constructed trees, $\tilde{Q}[i, D]$ denotes the total QoS value of h_i utilized for h_D in the constructed trees and $p_{i,D}$ denotes the success ratio of link $h_i \rightarrow h_D$. Intuitively, $\tilde{Q}[i, D] \times p_{i,D}$ denotes the QoS gain of h_D from h_i.

4.1 Minimum-weight path discovery

This process aims to discover a minimum-weight path for h_D which can provide a $\min\{Q_{def_D}, Q_{fix} \times Q_{req}\}$ QoS to h_D. Initially, the minimum-weight paths from all hosts to h_D are constructed by using Dijkstra's algorithm. They together form a spanning tree rooted at h_D. After that, the path with the smallest weight and satisfying two conditions is chosen as the required minimum-weight path for h_D. Consider that the chosen minimum-weight path is h_f -to- h_D. The initial condition is $h_f \in \mathcal{F} \cup D$ and the second condition is that $h_f = h_S$ or there is an h_S-to-h_f path in a few constructed trees such that h_S-to-h_D path in both cases can provide a $\min\{Q_{def_D}, Q_{fix} \times Q_{req}\}$ QoS to h_D. If no path from few hosts to h_D can satisfy these two conditions, then it is not feasible to provide an adequate QoS to h_D. Therefore, this algorithm will stop with Υ not admitted.

Given a source vertex in a weighted graph, Dijkstra's algorithm can create minimum-weight routes from the source vertex to the other vertices. To apply Dijkstra's algorithm, a directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{A})$ is used for representing a MANET where each vertex x in $\mathcal{V}$ individually corresponds to a host h_i and each $arc < h_i, h_j >$ from h_i to h_j in $\mathcal{A}$ indicates that h_j is within the communication range of h_i.

Each $arc < h_i, h_j >$ is allocated with a weight $w_{h_i,h_j} = 1/p_{h_i,h_j}$ referring that w_{h_i,h_j} transmissions are needed to successfully deliver a packet through link $h_{h_i} \rightarrow h_{h_j}$. The weight of a route in $\mathcal{G}$ is defined as the total weight of the arcs contained in it. Dijkstra's algorithm is performed by using the input graph $\mathcal{G}$ and source vertex h_D as:

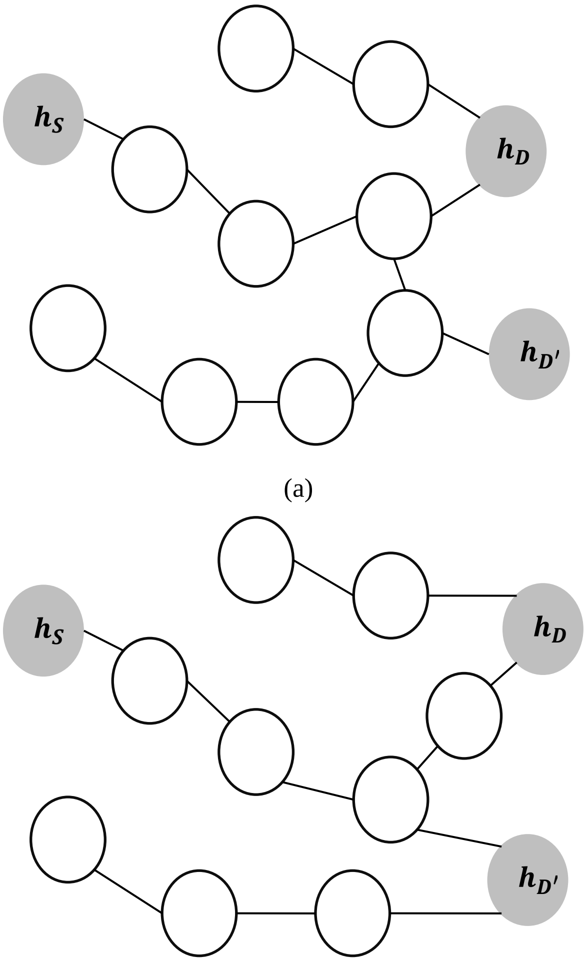
At first, each host h_i is allocated with a weight $w_{h_i} = 0$, if $h_i = h_d$ and $w_{h_i} = \infty$, if $h_i \neq h_d$. After that, a host with the least weight is frequently chosen and eliminated from $\mathcal{V}$.

While a host h_k is chosen, the weights of its neighbors are updated by replacing w_{h_j} with $min\left\{w_{h_j}, w_{h_k} + w_{h_k,h_j}\right\}$ for each neighbor h_j. Intuitively, w_{h_j} is the minimum weight of h_D-to-h_j routes discovered thus far and it is modified to $w_{h_k} + w_{h_k,h_j}$ if the new h_D-to-h_j route through h_k has a less weight. While $\mathcal{V}$ becomes null, the process of Dijkstra's algorithm ends with a spanning tree of $\mathcal{G}$ created. To provide $Discovering\_a\_Minimum\_Weight\_Path$, two changes of Dijkstra's algorithm are needed.

The algorithm for finding the minimum weight path is described below:

```
Algorithm 2 for finding a minimum weight path$(h_D, \mathcal{F}, D, Q_{def_D})$:
Begin
{
    For all hosts $h_i \neq h_D$ , assign $w_{h_D} = 0$ and $w_{h_i} = \infty$;
    For all pairs of $arc < h_i, h_j > and < h_j, h_i >$ in $\mathcal{A}$, exchange $w_{h_i,h_j}$ and $w_{h_j,h_i}$;
    Create minimum-weight paths from all hosts to $h_D$ by performing Dijkstra's algorithm, where the alternative of $w_{h_j}$ with $w_{h_k} + w_{h_k,h_j}$ for some host $h_j \neq h_d$ and its neighbor $h_k$ is executed under the following two conditions:
    •  The $h_j$-to-$h_D$ path through $h_k$ can provide a $\min\{Q_{def_D}, Q_{fix} \times Q_{req}\}$ QoS to $h_D$;
    •  There is no QoS degradation to the $h_j$-to-$h_D$ path through $h_k$, the constructed trees and all ongoing traffic flows;
    Choose and keep one path from the early obtained minimum-weight paths in which the path has the least weight and satisfies the following two conditions:
    •  $h_f \in \mathcal{F} \cup D;$
    •  $h_f = h_S$, or there is an $h_S$-to-$h_f$ path in a few constructed trees such that the resulting $h_S$-to-$h_D$ path can provide a $\min\{Q_{def_D}, Q_{fix} \times Q_{req}\}$ QoS to $h_D$.
    If the selected $h_f$-to-$h_D$ path does not exist, then stop the algorithm with $\Upsilon$ not admitted;
}
End
```

In *Algorithm 2*, first, Dijkstra's algorithm creates minimum-weight paths from h_D to the other hosts but $Discovering\_a\_Minimum\_Weight\_Path$ needs a minimum-weight path from some hosts to h_D. Thus, w_{h_i,h_j} and w_{h_j,h_i} need to be exchanged for each pair of $arc < h_i, h_j >$ and $< h_j, h_i >$ in $\mathcal{A}$. Then, the alternative of w_{h_j} with $w_{h_k} + w_{h_k,h_j}$ is performed to ensure that the required minimum-weight path for h_D can provide a $\min\{Q_{def_D}, Q_{fix} \times Q_{req}\}$ QoS to h_D and prevent QoS degradation simultaneously. One condition is that the novel h_j-to-h_D path through h_k denoted by $\mathfrak{R}$ can provide a $\min\{Q_{def_D}, Q_{fix} \times Q_{req}\}$ QoS to h_D . Another condition is that there is no QoS degradation to $\mathfrak{R}$, the constructed trees and all ongoing traffic flows. QoS degradation occurs to a path if the QoS requirement of some host on the path exceeds its residual delay, residual bandwidth, less packet loss rate and jitter. By using these two conditions, $\mathfrak{R}$ can provide an adequate QoS to h_D while the multicast routing problem is prevented. Fig. 1(a) shows the execution of calling *Algorithm 2* (finding a Minimum_Weight-path) for 2 destinations.



(a)

(b)

Figure. 1: (a) Minimum_weight routes obtained by calling *Algorithm 2* from h_D and (b) Minimum_weight routes obtained by calling *Algorithm 2* from $h_{D'}$

It is necessary to discover the maximum gain τ for determining whether these two conditions hold or not such that $\mathfrak{R}$ can provide a $\tau \times Q_{def_D}$ QoS to h_D whereas there is no QoS violation to $\mathfrak{R}$, the constructed trees and all ongoing traffic flows. If $\tau \times Q_{def_D} \geq min\{Q_{def_D}, Q_{fix} \times Q_{req}\}$, then these two conditions hold. Or else, both conditions do not hold. The value of τ is computed as $min\{\eta_D\}$ where η_D relies on the QoS of $\mathcal{F}$.

5. Experimental results

In this section, the performance efficiency of the QSSM routing protocol is compared with the

Received: December 9, 2021. Revised: January 21, 2022.

Table 2. Simulation parameters

Parameters		Range
Simulation area size		1000×1000 m^2
Number of nodes		100
MAC type		IEEE 802.11
Channel type		Wireless channel
Antenna type		Omni-directional
Propagation model		Two ray ground
Required bandwidth		500 kbps
Header size of data packets		10 bytes
Payload size of data packets		512 bytes
Batch size for randomized network coding		32
Multiplying factor		1.2
Queue size		64 packets
Traffic type		Constant Bit Rate (CBR)
Q_{fix} for creating multicast trees		0.3
Refresh time between two consecutive tree refreshes		4sec
Data rates		11 Mbps
Transmission power		0.0316 W
Simulation time		200 sec
Control packet size	hello	160 bytes
	table_query	10 bytes
	table_reply	1500 bytes
	path_query	10 bytes
	path_discovered	20 bytes

existing DSM [10], ODMRP [11], MCLSMRP [16], EFMMRP [17], AODV-BBO [19] and MDSR [20] in terms of end-to-end delay, one-hop delay, control byte overhead, success ratio and admission ratio. The simulation is carried out by using Network Simulator (NS version 2.35). The simulation parameters are given in Table 2.

5.1 One-hop delay

It refers to the time taken to transmit the data packet between two neighbouring hosts.

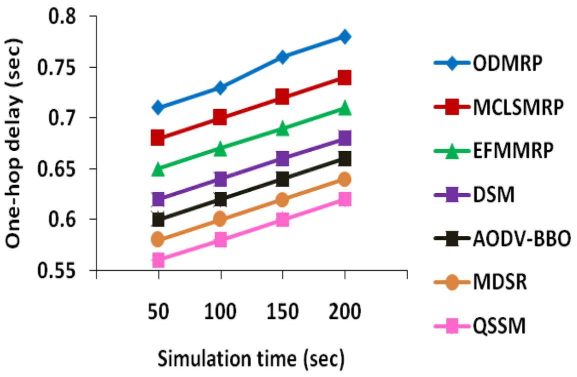


Figure. 2 Comparison of one-hop delay

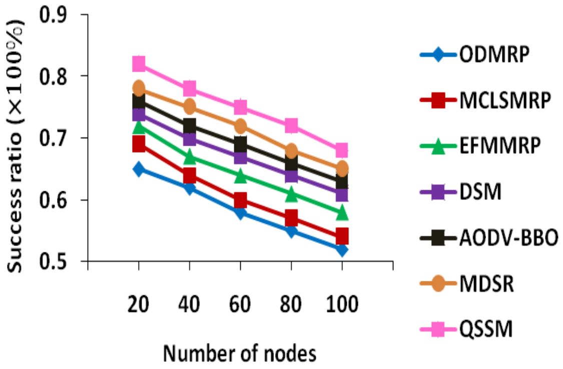


Figure. 3 Comparison of success ratio

Fig. 2 shows the one-hop delay values (in sec) obtained by the QSSM, DSM, EFMMRP, MCLSMRP and ODMRP. From this analysis, it is observed that the QSSM routing protocol has reduced one-hop delay than the other MRP while increasing the simulation time because of satisfying the QoS requirements to the requested traffic. If the simulation time is 200sec, then the one-hop delay for QSSM is 20.51% less than the ODMRP, 16.22% less than the MCLSMRP, 12.68% less than the EFMMRP, 8.82% less than the DSM, 6.1% less than the AODV-BBO and 3.1% less than the MDSR protocols.

5.2 Success ratio

It is defined as the ratio between the number of data packets successfully received by the destination without delay and the number of data packets delivered from the source.

Fig. 3 shows the success ratio (in %) achieved by the QSSM, DSM, EFMMRP, MCLSMRP and ODMRP. From this analysis, it is observed that the QSSM routing protocol has increased success ratio than the other routing protocol while increasing the number of nodes. If the number of node is 100, then the success ratio for QSSM is 30.77% higher than the ODMRP, 25.93% higher than the MCLSMRP, 17.24% higher than the EFMMRP, 11.48% higher than the DSM, 7.9% higher than the AODV-BBO and 4.6% higher than the MDSR protocols. This is achieved because of ensuring the QoS requirement of each requested traffic flow in the network during multicast routing.

5.3 End-to-end delay

It refers to the time required for transmitting the data packet from the source to few destination nodes.

Fig. 4 shows the end-to-end delay (in sec) obtained by the QSSM, DSM, EFMMRP, MCLSMRP and ODMRP. From this analysis, it is

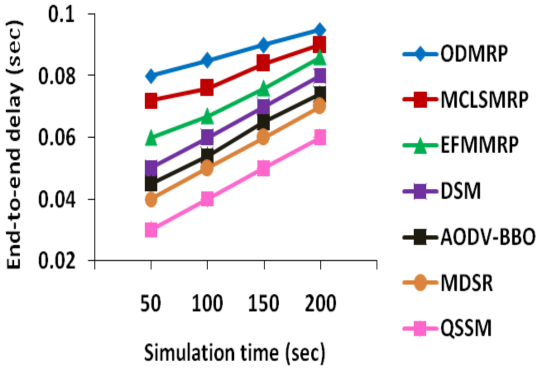


Figure. 4 Comparison of end-to-end delay

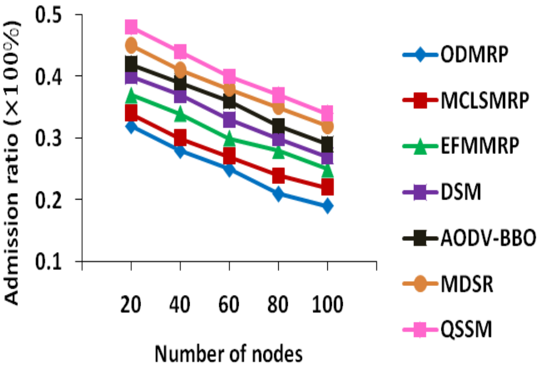


Figure. 5 Comparison of admission ratio

observed that the QSSM routing protocol has minimized end-to-end delay than the other routing protocol while increasing the simulation time because of considering jitter, end-to-end delay as QoS metrics to satisfy the multicast routing in MANET. If the simulation time is 200sec, then the end-to-end delay for QSSM is 36.84% less than the ODMRP, 33.33% less than the MCLSMRP, 30.23% less than the EFMMRP, 25% less than the DSM, 18.9% less than the AODV-BBO and 14.3% less than the MDSR protocols.

5.4 Admission ratio

It is the ratio of the number of admitted multicast flows to the number of requested multicast flows.

Fig. 5 shows the admission ratio (in %) achieved by the QSSM, DSM, EFMMRP, MCLSMRP and ODMRP. From this analysis, it is observed that the QSSM routing protocol has increased admission ratio than the other routing protocol while increasing the number of nodes. If the number of node is 100, then the admission ratio for QSSM is 78.95% higher than the ODMRP, 54.55% higher than the MCLSMRP, 36% higher than the EFMMRP, 25.93% higher than the DSM, 17.2% higher than the AODV-BBO and 6.3% higher than the MDSR protocols. This is due to the consideration of QoS demands for each requested and ongoing traffic flow during multicast routing.

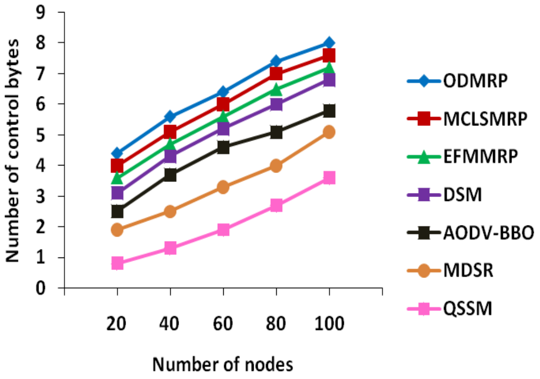


Figure. 6 Comparison of control byte overhead

5.5 Control byte overhead

It is defined as the number of control bytes transmitted per data byte delivered. The number of control bytes includes both the total length of protocol control packets and embedded protocol control information in data packets.

Fig. 6 shows the number of control bytes for the QSSM, DSM, EFMMRP, MCLSMRP and ODMRP. From this analysis, it is observed that the QSSM routing protocol has less number of control bytes because of reducing the redundant packets using randomized network coding than the other routing protocol while increasing the number of nodes. If the simulation time is 200sec, then the number of control bytes for QSSM is 55% less than the ODMRP, 52.63% less than the MCLSMRP, 50% less than the EFMMRP, 47.06% less than the DSM, 37.9% less than the AODV-BBO and 29.4% less than the MDSR protocols.

6. Conclusion

In this article, the QSSM routing protocol was proposed for MANET to increase the network performance by improving the multicast routing. Initially, QoS metrics such as bandwidth, packet dropping rate, jitter and end-to-end delay were measured for each node in the network. Based on these metrics, multiple QoS-satisfied multicast trees were constructed wherein each tree can guarantee the predetermined fraction of QoS requirements. Also, the overall QoS performance was improved by guaranteeing the QoS requirements to a requested traffic flow and ongoing flows. Moreover, a randomized network coding scheme was used for preventing redundant data packets during transmission to increase the network capacity efficiently. Finally, the simulation results proved that the QSSM protocol has a 0.62sec one-hop delay and 0.06sec end-to-end delay while simulating the network in 200sec. Similarly, it has a 68% of success ratio and 34% of admission ratio than to the other protocols for 100 nodes that exist in the network. In contrast, a selection of proper data rates for hosts was not effective because of the different transmission ranges of hosts. So, the future extension of this work will focus on selecting the proper data rate for hosts using deep learning algorithms.

Conflicts of Interest

The authors declare no conflict of interest.

References

[1] G. K. Wadhwani and N. Mishra, “A Survey of multicast routing protocols in MANET”, *IITM Journal of Management and IT*, Vol. 8, No. 1, pp. 42-50, 2018.

[2] S. Laqtib, K. E. Yassini, and M. L. Hasnaoui, “Performance evaluation of multicast routing protocols in MANET”, In: *Proc. of International Conference on Advanced Intelligent Systems for Sustainable Development*, pp. 847-856, 2018.

[3] S. Debnath, M. Saha, M. Islam, P. K. Sarker, and I. Pramanik, “Evaluation of multicast and unicast routing protocols performance for group communication with QoS constraints in 802.11 mobile ad-hoc networks”, *International Journal of Computer Network & Information Security*, Vol. 13, No. 1, pp. 1-15, 2012.

[4] A. M. A. Mo'men, H. S. Hamza, and I. A. Saroit, “A survey on security enhanced multicast routing protocols in Mobile Ad hoc Networks”, In: *Proc. of 7th IEEE International Symposium on High-capacity Optical Networks and Enabling Technologies*, pp. 262-268, 2010.

[5] S. A. Hadiwardoyo, “An overview of multicast routing techniques for group communications applications”, In: *Proc. of IEEE 25th Telecommunication Forum*, pp. 1-4, 2017.

[6] M. Amad, A. Boudries, and L. Badis, “Application layer multicast based services on hierarchical peer to peer architecture”, In: *Proc. of Applied Mechanics and Materials*, Vol. 892, pp. 64-71, 2019.

[7] D. Chander and R. Kumar, “QoS enabled cross-layer multicast routing over mobile ad hoc networks”, *Procedia Computer Science*, Vol. 125, pp. 215-227, 2018.

[8] S. M. Adam and R. Hassan, “Delay aware reactive routing protocols for QoS in MANETs: a review”, *Journal of Applied Research and Technology*, Vol. 11, pp. 844-850, 2013.

[9] B. S. Kim, M. Aldwairi, and K. I. Kim, “An efficient real-time data dissemination multicast protocol for big data in wireless sensor networks”, *Journal of Grid Computing*, Vol. 17, No. 2, pp. 341-355, 2019.

[10] Y. H. Chen, C. C. Hu, E. H. K. Wu, S. M. Chuang, and G. H. Chen, “A delay-sensitive multicast protocol for network capacity enhancement in multirate MANETs”, *IEEE System Journal*, Vol. 12, No. 1, pp. 926-937, 2018.

[11] M. A. Kharraz, H. S. Azad, and A. Y. Zomaya, “On-demand multicast routing protocol with efficient route discovery”, *Journal of Network and Computer Applications*, Vol. 35, No. 3, pp. 942-950, 2012.

[12] G. A Walikar and R. C. Biradar, “Energy aware multicast routing in mobile ad-hoc networks using NS-2”, In: *Proc. of IEEE International Conference on Electrical, Computer and Communication Technologies*, pp. 1-7, 2015.

[13] S. Gopinath and N. Nagarajan, “Energy based reliable multicast routing protocol for packet forwarding in MANET”, *Journal of Applied Research and Technology*, Vol. 13, No. 3, pp. 374-381, 2105.

[14] Y. H. Robinson, E. G. Julie, and S. Balaji, “Bandwidth and delay aware routing protocol with scheduling algorithm for multi hop mobile ad hoc networks”, *World Academy of Science, Engineering and Technology, International Journal of Computer, Electrical, Automation, Control and Information Engineering*, Vol. 10, No. 8, pp. 1512-1521, 2016.

[15] G. Singal, V. Laxmi, M. S. Gaur, S. Todi, V. Rao, M. Tripathi, and R. Kushwaha, “Multi-constraints link stable multicast routing protocol in MANETs”, *Ad Hoc Networks*, Vol. 63, pp. 115-128, 2017.

[16] A. K. Yadav, S. K. Das, and S. Tripathi, “EFMMRP: Design of efficient fuzzy based multi-constraint multicast routing protocol for wireless ad-hoc network”, *Computer Networks*, Vol. 118, pp. 15-23, 2017.

[17] V. L. Vinya and G. V. Rao, “An energy efficient multicast route establishment using AODV with PSO algorithm and RSA for secured transmission”, *International Journal of Intelligent Engineering & Systems*, Vol. 12, No. 5, pp. 257-266, 2019.

[18] A. B. Jayaramu and M. K. Banga, "Delay aware routing protocol using optimized AODV with BBO for MPLS-MANET", *International Journal of Intelligent Engineering and Systems*, Vol. 13, No. 5, pp. 29-37, 2020.

[19] H. H. E. Sayed, A. Younes, and F. A. Alghamdi, "Multiobjective multicast DSR algorithm for routing in mobile networks with cost, delay, and hop count", *Complexity*, Vol. 2021, pp. 1-8, 2021.