

Research Article

On the Performance of the Current MANET Routing Protocols for VoIP, HTTP, and FTP Applications

Waheb A. Jabbar, Mahamod Ismail, and Rosdiadee Nordin

Department of Electrical, Electronic and System Engineering, Universiti Kebangsaan Malaysia, 43600 Bangi, Selangor, Malaysia

Correspondence should be addressed to Waheb A. Jabbar; waheb@ieee.org

Received 11 December 2013; Accepted 26 December 2013; Published 17 February 2014

Academic Editor: Tzonelih Hwang

Copyright © 2014 Waheb A. Jabbar et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The recent advances of ensemble mobile environment of smart devices with embedded sensors have enabled the MANET to play a key role in the smart cities as well as WSN and WMN. However, these smart devices are still limited in terms of energy, processor, and memory. Moreover, the efficient routing for reliable network connectivity at anytime, anywhere, and about everything is still a challenge in multihop wireless networks. This paper evaluates the QoS and energy efficiency of three active routing protocols: (i) OLSRv2, a successor to OLSR, (ii) DYMO, a successor to both DSR and AODV, and (iii) MP-OLSR multipath extension to OLSRv2. In contrast to the related previous works which focused only on CBR traffic without considering the influence of specific traffic types on the performance of routing protocols, this work focused on this area from a different perspective. It evaluates the performance of three internet-based traffic types that can be used in the smart city applications: VoIP, HTTP, and FTP using different simulation models. The impact of the network density, load traffic, and nodes mobility on the considered protocols was evaluated by considering particular performance metrics for each traffic application. Based on the results, the study concludes by presenting useful recommendations for future work.

1. Introduction

Smart city is becoming the orientation trend of urban development [1, 2]. The advances and popularity in the smart mobile devices (laptops, notebooks, tablets, and smartphones) with embedded sensors, computing resources and wireless technologies, allied to the demand for smart city applications, are presenting a new ensemble mobile environment in smart cities. These sensor enabled smart mobile devices are ideal candidates to catalyze smart applications and services in smart cities [3] and present the mobile ad hoc networks (MANET) [4, 5] as an essential component of the smart city infrastructure as well as wireless sensor networks (WSN) and wireless mesh networks (WMN). The infrastructure of smart cities needs to integrate all these wireless networks to make them accessible and controllable for the applications and services [6].

MANET is defined as a self-organizing collection of wireless mobile nodes which constructs a temporary and a dynamic wireless network with no infrastructure. MANETs

are also known as self-configuring, which implies that they have no central management system with specific configuration responsibilities [7, 8]. There are two (2) types of possible communication among mobile nodes: (i) direct communication which happens among the nodes within its wireless range or (ii) indirect communication which takes place via other intermediate nodes in the network. However, mobile nodes are still restricted in terms of energy, processors, and memory. For ensuring that the transfer of data is secure and reliable, an efficient routing protocol is necessary for discovering routes between pairs of source-destination mobile nodes which are involved in communication either through a single hop or multiple hops with the help of intermediate nodes.

The internet engineering task force (IETF) [9] has developed four (4) routing protocols for MANET [10]: ad hoc on-demand vector: (i) (AODV) [11], (ii) dynamic source routing (DSR) [12], (iii) topology dissemination based on reverse-path forwarding (TBRPF) [13], and (iv) optimized link state routing (OLSR) [14]. The efforts of the IETF are being made to standardize the OLSRv2 [15], a successor to OLSR,

and dynamic MANET On-demand (DYMO) [16] which is currently known as AODVv2 [17], a successor to both AODV and DSR routing protocol. However, the multipath-OLSR (MP-OLSR) is proposed in [18] as a multipath extension to OLSRv2 and it is a hybrid multipath routing protocol with multiple description coding for data transfer. Although these routing protocols are designed for MANETs, they have been widely implemented in WMN [19, 20] and most of them have been adopted for WSN [21].

With the fast growth of the internet of things (IoT) [22], refer to the objects or devices interconnecting and cooperating in self-configuring wireless networks to connect all things at all times [23]. In all applications of the IoT, the data is sent from the source node to the destination through a routing protocol which should be efficient enough in terms of quality of service (QoS) and consumed energy [24] to guarantee the data transmission. However, such applications rely on the Internet-based traffic such as (i) voice over internet protocol (VoIP) [25], (ii) file transfer protocol (FTP) [26], and (iii) hypertext transfer protocol (HTTP) [27] and they require a higher QoS over the wireless links. In this context, our paper evaluates the performance of three (3) routing protocols: (i) OLSRv2, (ii) DYMO, and (iii) MP-OLSR, in terms of QoS and energy consumption for the mentioned types of the Internet-based traffic. Different evaluation metrics for QoS were considered for each traffic type, namely, throughput, average end-to-end delay, average jitter, and average time in queue as well as the total consumed energy for transmitting and receiving packets between source-destination pairs. The identified performance metrics were selected according to their significant impact on the considered internet traffic types in our evaluation.

The remaining part of this paper is structured as follows. In Section 2, we briefly review the OLSRv2, DYMO, and MP-OLSR routing protocols followed by materials, methods, and the simulation models with the performance metrics in Section 3. Then, Section 4 demonstrates the simulation results, performance analysis, and comparison. Finally, Section 5 presents the conclusions and highlights recommendations for future work.

2. Background

In this section, we provide a summary of the considered routing protocols. The current situation of standardization for OLSR, which includes OLSRv1 and OLSRv2, is described. Moreover, this section presents an overview of DYMO and MP-OLSR as well as its main functionalities.

2.1. OLSR and OLSR Version 2 Overview

2.1.1. OLSRv1. OLSR [14] is the currently most employed and leading proactive routing protocol in mobile ad hoc networks and the first version of OLSR (OLSRv1) [14]. It has been standardized as an experimental RFC (RFC 3626). It was developed by the French National Institute for Research in Computer Science and Control (INRIA) and it works in

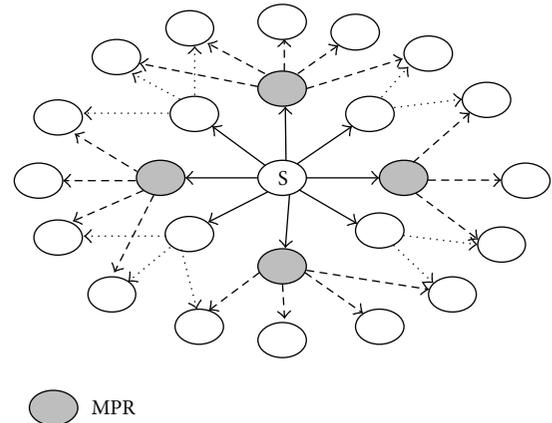


FIGURE 1: The MPR election in OLSR protocol.

a proactive manner (table-driven); that is, topology information is exchanged between the nodes on a periodic basis. The core optimization of OLSR is to minimize the control traffic by selecting a small number of nodes, known as multipoint relays (MPR) which is an improved flooding mechanism for topological information as shown in Figure 1. OLSR generally proposes four (4) types of periodic control messages, namely, (i) hello messages, (ii) topology Control (TC) messages, (iii) multiple interface declaration (MID) messages, and (iv) host and network association (HNA) messages. A routing table is maintained to keep the next hop information to all the possible destination nodes.

2.1.2. OLSRv2. The second version of OLSR (OLSRv2) [15] is an update and successor to OLSRv1 as published in RFC3626. It holds the same basic mechanisms and algorithms of OLSRv1; however, OLSRv2 provides an even more flexible signaling framework and some simplification of the exchanged messages between nodes. It also accommodates both IPv4 and IPv6 addresses in a compact fashion. OLSRv2 is developed for mobile ad hoc networks. Basically, it modifies the OLSRv1 by using and extending the following generalized building blocks: *The MANET Neighborhood Discovery Protocol (NHDP)* defined in [RFC6130] [28], *The Generalized MANET Packet/Message Format* [RFC5444] [29], *The TLVs* as specified in [RFC5497] [30] and, *optionally, message jitter* as specified in [RFC5148] [31].

OLSRv2 is developed to operate independently from other protocols. OLSRv2 makes no assumptions about the underlying link layer, while it may use link layer information and notifications when existing and applicable. OLSRv2 [32] is in its final stage of standardization. It utilizes two basic types of control packets: *hello messages*, and *topology control messages (TC messages)*.

(1) Hello Messages. In OLSRv2, HELLO messages have five (5) functions: (i) discovering links to adjacent OLSR nodes, (ii) performing bidirectional check on the discovered links, (iii) advertising neighbors hence discovering 2-hop neighbors, (iv) selecting single MPR, and (v) advertising their own interfaces which participate in the MANET. The HELLO

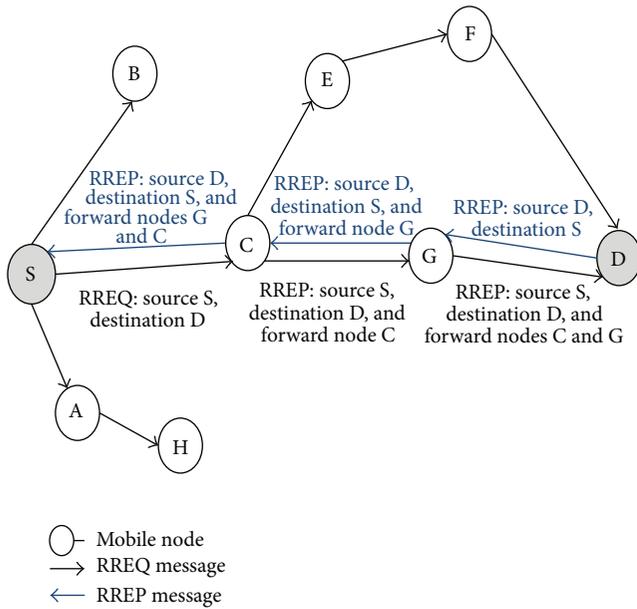


FIGURE 2: DYMO route discovery.

messages are emitted periodically, thereby allowing nodes to continuously track changes in their local neighborhoods.

(2) *TC Messages*. In OLSRv2, TC messages serve to (i) inject link state information into the entire network, (ii) inject addresses of hosts and networks for which they may serve as a gateway to the entire network, (iii) allow nodes with multiple interface addresses to ensure that nodes within two hops can associate these addresses with a single node for efficient MPR Set determination. The TC messages are emitted periodically, thereby allowing nodes to continuously track global changes in the network.

OLSRv2 performs mainly three (3) basic processes: (i) *neighborhood discovery*, (ii) *MPR flooding*, and (iii) *link state advertisements*.

(1) *Neighborhood Discovery*. The OLSRv2 applies NHDP for HELLO messages for Neighborhood Discovery process. It continuously updates information at each node and discovers the nodes which are in direct communication range of itself (1-hop neighbors) and detects which one of these can establish bidirectional communication. It also learns about the “neighbors of its neighbors” (2-hop neighbors).

(2) *MPR Flooding*. A node selects MPRs from its one hop neighbors set with “symmetric,” that is, bi-directional, linkages. Thereby, a node announces to the network that it has reachability to its MPR. Thus, besides being used to facilitate efficient flooding, MPRs are used for route calculation from any given node to any destination in the network.

(3) *Link State Advertisement*. Nodes selected as MPRs also have a special responsibility when declaring link state information in the network and announce this information periodically in their TC messages. Nodes determine which link

state information to advertise through the network. Each node must advertise links between itself and its MPRs set, in order to allow all nodes to find the shortest paths.

2.2. *DYMO Overview*. The dynamic MANET on-demand (DYMO) [16] routing protocol is a unicast reactive routing protocol which is intended to be used by mobile nodes in wireless multihop networks. It is also a reactive routing protocol and it is a successor of AODV and DSR. Nowadays, it is known as AODVv2 [17]. In this, routing message (control packet) is generated only when the node receives a data packet and it does not have any routing information. The most attractive specification defined in the DYMO draft is the internet connectivity. In DYMO protocol, there are two (2) basic operations, namely; (i) *route discovery*, and (ii) *route maintenance*.

2.2.1. *Route Discovery*. It is performed when a DYMO router must transmit a packet towards a destination for which it does not have a route as described in Figure 2. The route request (RREQ) routing message is generated for a target node for which it does not have any routing formation. Source node floods the RREQ message to find the target node. During flooding, each intermediate node records a route to the originating node by adding the routing information into this routing table. When the target node receives the RREQ, it responds with a route reply (RREP) message which is sent as a unicast message towards the originating node. Each node that receives the RREP records a route to the target node and forwards the RREP to the next hop. When the originating node receives the RREP, routes between the originating node and the target node in both directions are established.

2.2.2. *Route Maintenance*. It is performed to avoid prematurely expunging routes from the route table and to avoid dropping packets when an active route breaks. In order to react to the changes in the network topology, nodes maintain their routes and monitor their links. A route error (RERR) message is generated by a node whenever it receives a data packet for a destination to which it has no route in its routing table. This RERR RM notifies other nodes that the current route is broken. Once the source receives the RERR, it reinitiates the route discovery if it still has packets to deliver. Hello messages can be used by nodes to maintain routes to all their neighboring nodes. Sequence numbers are used to avoid routing loops and propagation of stale routing information.

2.3. *MP-OLSR Overview*. The multipath optimized link state routing (MP-OLSR) [18] can be regarded as a hybrid multipath routing protocol based on OLSRv2, includes a major modification of the Dijkstra algorithm (two cost functions are now used to produce multiple disjoint or non-disjoint paths) as presented in Figure 3. It sends out HELLO messages and TC messages periodically to be aware of the network topology, just like OLSR. However, it differs from OLSR in the sense that MP-OLSR does not always keep a routing table to all the possible destinations. It only calculates the routes when there are data packets needed to be sent out. To the best of our

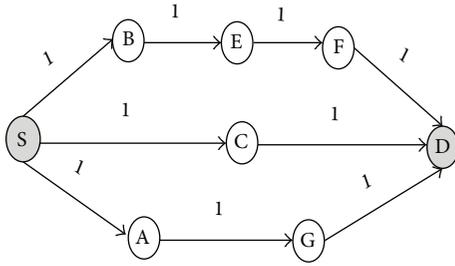


FIGURE 3: MP-OLSR multiple paths in initial state cost function.

knowledge, MP-OLSR is the only OLSR-based protocol that proposes a strategy to increase the chances of constructing multiple node disjoint paths [33]. It maintains several paths for the same pair (source, destination). One of these paths will be selected as the primary path and the others will be the secondary paths. MP-OLSR has two (2) main functionalities to make it possible to find multiple paths from source to destination: (i) *topology sensing* and (ii) *route computation*. Additionally, it has two auxiliary functionalities, namely: *route recovery* and *loop detection* to improve the performance of the protocol. Moreover, MP-OLSR is compatible with standardized OLSR by using IP source routing.

2.3.1. Topology Sensing. The topology sensing is to make the nodes aware of the topology information of the network and this part benefits from MPRs like OLSR. It includes link sensing, neighbor detection, and topology discovery. Both link sensing and neighbor detection are based on the periodic exchange of HELLO messages. Topology discovery generates the information base which concerns the nodes that are more than two hops away (topology set). It is based on the flooding of the TC messages (optimized by selecting the MPR set).

2.3.2. Route Computation. The route computation uses the multipath Dijkstra algorithm [34] to calculate the multipaths based on the information obtained from the topology sensing. The source route (all the hops from the source to the destination) is saved in the header of the data packets. MP-OLSR utilizes an on-demand scheme to avoid the heavy computation of multiple routes for every possible destination.

2.3.3. Route Recovery. The route recovery can effectively reduce the packet loss due to the dynamic topology of the network. Moreover, it overcomes the disadvantage of the source routing scheme used in MP-OLSR. In this mechanism, the intermediate nodes recompute the route to the destination if it is not one of its neighbors' sets.

2.3.4. Loop Detection. The loop detection can be used to avoid potential loops in the network. In the specification of the algorithm, the paths will be loop-free. However, in practice, the situation will be much complicated due to the change of the topology and the instability of the wireless medium. After the route recovery is performed, a new path will be calculated from the current node to the destination

and the new path will be used if there is no loop. Otherwise, it will try to find another path or the packet will be discarded.

3. Materials and Methods

3.1. Simulation Tools. EXata 3.1 simulator [35] is used to perform extensive simulations to evaluate the performance of the considered routing protocols in different scenarios. The EXata communication simulation platform is a network emulator that is integrated into the well-known QualNet network simulator [36] which is widely used in academic research and industry [18]. EXata is a comprehensive suite of tools for emulating large wireless and wired networks. It uses simulation and emulation to predict the behavior and performance of networks to improve their design, operation, and management. Three (3) main simulation parameters are considered in this work: (i) number of nodes, (ii) number of traffic sources, and (iii) percentage of static nodes in each scenario.

3.2. Network Model. MANETs can be modeled by a graph $G(V, E)$, where V is the set of nodes representing the mobile devices and E is a set of arcs; each arc models the communicational range intersection between two devices [37]. Each node $v \in V$ can directly communicate with a set of neighbor devices within its communication range; otherwise, it has to use routing protocol to transmit packets to remote devices, which are not adjacent neighbors. One of the possible paths in the graph $G(V, E)$ can be used to transmit the packets from the node v to the destination node. Due to the node's mobility, the network topology will frequently change. Thus, the cardinality of the nodes V remains the same throughout the period whereas the cardinality of the edges E changes. As the number of nodes set V increases, the average hop length E also increases and this will affect the routing protocol performance. Therefore, the network size is one of the major parameters in simulation studies of routing protocol evaluation in MANET. In our simulation, for network size parameter, we considered different numbers of nodes set (25, 36, 49, 64, 81, and 100 nodes) with uniformly distributed initial position like a grid. However, for the simulation scenarios that studied the other parameters (number of sources traffic and number of static nodes), the number of nodes is set to 60 nodes as shown in Figure 4.

3.3. Mobility Model. The mobility of nodes in a mobile multihop wireless network can be described by the node's speed as well as its pause duration. The average speed of nodes in the network is an indicator of changing the network topology and the rate of links break as well as routing overhead. The Random Waypoint Model (RWP) [38] is one of the most widely used mobility models in our simulation. This model selects random destinations and speeds from (0 to V_{\max}) for each node. After the nodes reach their selected destinations, they pause for a defined duration of time and then the process is repeated. In our simulation, the maximum speed is 30 m/s and the pause time is set to 10 [Sec]. In particular scenarios, we deployed nodes without any mobility

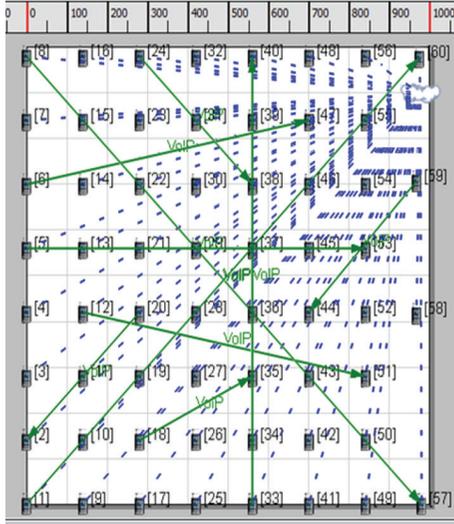


FIGURE 4: The simulation topology with 60 nodes and 10 VoIP connections.

models (static nodes) as a percentage of the total number of nodes (25%, 50%, 75%, and 100%) to compare and analyze the impact of mobility on the performance of routing protocols.

3.4. Traffic Generation Model. The traffic model type has a major effect on the performance of the protocol and its execution. Moreover, the change of traffic sources number impacts the congestion level in the network. This paper focuses on three Internet-based traffic types: VoIP, FTP, and HTTP that can be utilized in smart city and IoT applications in different scenarios. The performance of the considered routing protocols was evaluated for different numbers of traffic models and different types. The number of sources traffic was changed from 5 sources to 30 sources. We briefly describe these models of traffic below.

3.4.1. Voice over IP. VoIP is a low cost IP telephony term for a set of facilities used to manage the delivery of voice information over the internet. It involves sending voice information in a digital form in discrete packets and uses a wide variety of protocols such as H323 [39] and SIP [40] for call signaling and management purposes in order to provide the above mentioned services. Once the session is established, the actual data is carried over packets of real-time transport protocol (RTP) [41]. A companion control protocol to RTP is known as real-time transport control protocol (RTCP) [41]. It is used to collect end-to-end information about the quality of the session to each participant. In the VoIP traffic generator scenarios, we used H323 for call signaling modules and RTP for media transmission. Moreover, audio/video application G.711 was used for encoding codec with packetization interval of 20 ms.

3.4.2. HTTP. Hypertext transfer protocol simulates single-TCP connection web servers and clients. Only one server for each client was specified in our HTTP simulation scenarios.

3.4.3. FTP. It represents the file transfer protocol server and client. In this work, the size of the items sent was specified to 512 bytes, and the number of items to send was set to 100 items.

3.5. Energy Model. The Energy consumption in multihop wireless networks is a significant issue. Since nodes are battery-operated; the battery energy is limited and a node can only transmit a restricted number of bits. The maximum number of bits that can be transmitted is defined by the total battery energy divided by the required energy per bit. The node in a wireless network has four states namely: *transmit*, *receive*, *idle*, and *sleep*, and each state consumes different energy level. In our simulation model, we used a generic radio energy model which is derived to estimate the consumed energy in each state and we set the energy parameters as it used in [42]. The consuming energy for these states can be defined as follows.

P_{transmit} : the consumed circuitry power for transmitting one packet.

P_{receive} : the consumed circuitry power for receiving one packet.

P_{idle} : the consumed circuitry power by the node while it is idle but keep listening to the exchange control messages.

P_{sleep} : the small amount of power that circuitry consumed when the node is sleep and no communication signal is possible. This value is neglected because it is very small compared to other states. Therefore, the total energy consumption for a node to transmit and receive a packet can be calculated as follow:

$$E_{\text{total}} = E_{\text{trans}} + E_{\text{rec}} + E_{\text{idle}}, \quad (1)$$

where

$$\begin{aligned} E_{\text{trans}} &= P_{\text{transmit}} * t_{\text{trans}}, \\ E_{\text{rec}} &= P_{\text{receive}} * t_{\text{rec}}, \\ E_{\text{idle}} &= P_{\text{idle}} * t_{\text{idle}}, \end{aligned} \quad (2)$$

and t_{trans} , t_{rec} , and t_{idle} are the duration of time for each state.

3.6. Simulation Environments and Parameters. The topology of the network simulation environment is shown in Figure 4. The scenarios spread out 60 mobile nodes placed with uniformly distributed initial positions like a grid in an area of 1000 m by 1000 m to construct a network layout. For each scenario, the simulation time is set to 100 [Sec]. The applications of traffic are late 15 [Sec] after the simulation starts. This is to give sufficient time for routing messages exchange. All simulation scenarios were repeated 20 times with different randomly selected seeds to get different experiments, and the results were averaged over all experiments. The used radio type in our simulations is 802.11 b radio with omnidirectional antenna model and 11 Mbps data rate. The two-ray was used as

a pathloss model with 2.4 GHz for the channel frequency and the constant shadowing model with 4.0 shadowing mean. In the network physical layer, we set the transmission power to 15 dBm, and the generic energy model was used as previously explained. The radio transmission range is about 270 m as a result of the selected Wi-Fi parameters setting. As previously mentioned, RWP was used as the mobility model for the mobile nodes. The node's maximum speed was set to 30 m/s and the min speed was set to 0 m/s with 10 [Sec] pause times. All nodes were provided with residual life estimator battery model.

For each type of the application traffic models VoIP, HTTP, and FTP, we performed three main simulation scenarios and compared the performance of each protocol in terms of specific evaluation metrics. In the first scenario, the number of traffic sources was set to 10 flows, and the number of nodes was 25, 36, 49, 64, 81, and 100 nodes to evaluate the impact of network size on the performance of routing protocols for each application type. However, in the second scenario, the number of running applications traffic was set to 5, 10, 15, 20, 25, and 30 applications for randomly selected source-destination pairs and the number of nodes was set to 60 nodes to evaluate the performance metrics under diverse traffic loads. In the last scenario, the impact of changing the number of static nodes as a percentage of the total nodes number in the network was evaluated. In this context, the number of the deployed static nodes (nodes without mobility model) was selected as 15, 30, 45, and 60 nodes as a percentage of the total number of 60 nodes (25%, 50%, 75%, and 100%). The parameters that we used in our simulation were used in similar previous studies and the details are presented in Table 1.

3.7. Performance Evaluation Metrics. The main purpose of this work is to evaluate the performance of routing protocols for the selected types of internet-based traffic applications in terms of QoS and energy consumption. For this purpose, the following evaluation metrics were used.

(1) *Throughput.* Throughput is defined as the total number of bits that were successfully received at the server within definite time duration. The throughput at the receiver can be calculated as shown below.

(i) If the session is complete,

$$\text{Throughput} = \frac{(\text{Total bytes received} * 8)}{(t_{\text{last}} - t_{\text{1st}})}, \quad (3)$$

(ii) and if the session is incomplete,

$$\text{Throughput} = \frac{(\text{Total bytes received} * 8)}{(t_{\text{sim}} - t_{\text{1st}})}, \quad (4)$$

where t_{last} is the time of last packet received, t_{1st} is the time of first packet received, and t_{sim} is the simulation time and the times are in seconds.

(2) *Average End-to-End Delay.* It refers to the average of time duration over all surviving data packets that are transmitted

TABLE 1: Simulation parameters.

Parameter	Values
Simulator	EXata 3.1
Routing protocol	OLSRv2, MP-OLSR, and DYMO
Simulation area	1000 m × 1000 m
Mobility	RWP, max speed 0–30 m/s
Simulation time	100 seconds
No. of Nodes	25, 36, 49, 64, 81, 100, and 60
No. of Sources traffic	5, 10, 15, 20, 25, and 30
Applications traffic	VoIP, HTTP, and FTP
Traffic start-end	15 s–0 s
Transport protocol	UDP/TCP
Network protocol	IPv4
Priority input queue size	50000
Mac protocol	IEEE 802.11
Energy model	Generic:
	$P_{\text{transmit}} = 1300 \text{ mW}$
	$P_{\text{receive}} = 900 \text{ mW}$
	$P_{\text{idle}} = 740 \text{ mW}$
	$P_{\text{Sleep}} = 47 \text{ mW}$
Battery model	Residual life estimator
Physical layer model	PHY 802.11b
Wireless channel frequency	2.4 GHz
Pathloss model	Two ray
Shadowing model	Constant
Shadowing mean	4.0 dB
Transmission range	270 m
Data rate	11 Mbps
Simulation repeat times	20 different seeds

from the source to the destinations. This includes all possible delays caused due to buffering, queuing, retransmissions, propagation, and transfer through channel. Consider the following:

$$\text{Average End-to-End Delay} = \frac{t_{\text{total}}}{N_p}, \quad (5)$$

where t_{total} is the total of transmission delays of all received packets, N_p is the number of packets received.

The transmission delay of a packet refers to the difference between the time of packet received at the server and the time of packet transmitted at the client, and all times are in seconds.

(3) *Average Jitter.* Average jitter is defined as the mean deviation of the packet spacing at the receiving node compared to the sending node due to the network congestion. The average jitter can be estimated as follows:

$$\text{Average Jitter} = \frac{P_j}{(N_p - 1)}, \quad (6)$$

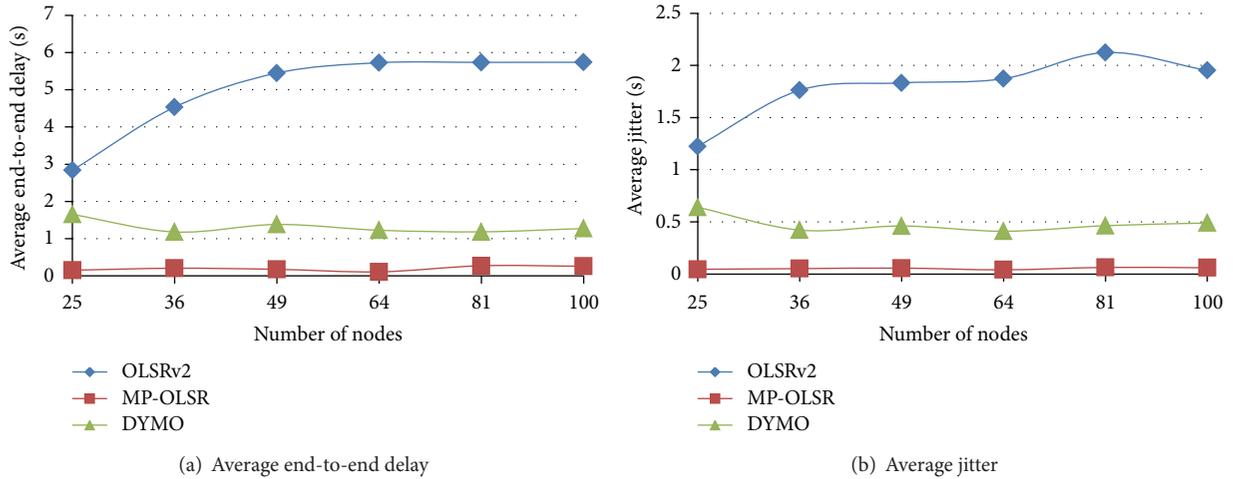


FIGURE 5: Performance metrics comparison for OLSRv2, DYMO, and MP-OLSR for different network size with 10-VoIP traffic.

where P_j is the total packet jitter for all received packets, N_p is the number of packets received.

Packet Jitter. Is the difference between the transmission delay of the current packet and the transmission delay of the previous packet. The packet jitter can be calculated only if there are at least two packets received.

(4) **Average Time in FIFO Queue.** is known as average delay in seconds spent by the packets in the queue.

(5) **Energy Consumption.** This metric is defined as the average of consumed energy in mWh of all nodes at the end of simulation. The average consumption energy of the nodes was changed according to the node's state either transmits, receive, or idle state. The total average consumed energy can be calculated as shown earlier in (1).

(6) **Average Page Request Time(s).** One more key QoS metric was also used to ensure the quality of HTTP traffic which specifies the average page request time in seconds.

4. Experimental Results and Discussion

This section provides a discussion of the main results obtained in this study. This includes the results obtained from comparing the performances of the routing protocols: OLSRv2, DYMO, and MP-OLSR are compared for the considered application traffic models (VoIP, HTTP, and FTP) in different scenarios with different metrics. These results were obtained using the EXata simulator platform which offers high reliability and scalability simulations for wireless communication. Our simulation scenarios were studied based on network density, load traffic, and percentage of static nodes to show the behavior of the considered protocols and the impact of application traffic type on the performance of routing protocols.

4.1. VoIP Traffic Model. The size of VoIP data is small and generates a light traffic and the flow rate of UDP was maintained at 64 Kb/s with packet size at 80 bytes (RTP + UDP + Payload) and packetization interval at 20 [ms] to closely emulate the G.711 codec. Therefore, in this scenario, several experiments were carried out to verify the impact of the considered parameters (network size, traffic load, and mobility) on the performance evaluation metrics of VoIP application traffic and these were compared for the selected routing protocols as shown in Figures 5, 6, and 7.

4.1.1. The Network Size. In the first scenario, random 10 VoIP connections were selected between 10 source-destination pairs to evaluate the effect of network size (number of nodes) as shown in Figures 5(a) and 5(b). The comparison of the average end-to-end delay with respect to the number of nodes for the three routing protocols OLSRv2, DYMO, and MP-OLSR are shown in Figure 5(a). Obviously, and based on the statistics of RTP that carried the VoIP packets, the MP-OLSR provided less end-to-end delay than that provided by both OLSRv2 and DYMO in all scenarios with different network size. This is because the multipath nature of this protocol with its loop detection functionality enabled MP-OLSR to send VoIP packets via different paths and avoid transmitting unnecessary packets. Therefore, the maximum delay was only 0.26 [Sec] in high density network with 100 mobile nodes and the lowest delay was 0.12 [Sec] in the scenario with 64 nodes. However, the OLSRv2 obtained the highest delay in all scenarios and its delay increased proportionally with the number of nodes from 2.8 [Sec] at 25 nodes scenario to 5.8 [Sec] at the scenario with 100 nodes. This is because the propagation delay and queue delay between the source and destination increased, thus leading to an increase in the average end-to-end delay. These results indicate that DYMO is certainly better than OLSRv2 as it provided an average delay lower by 28% than OLSRv2. However, it performs worse than MP-OLSR in terms of end-to-end delay (it provided 7 times higher average end-to-end delay).

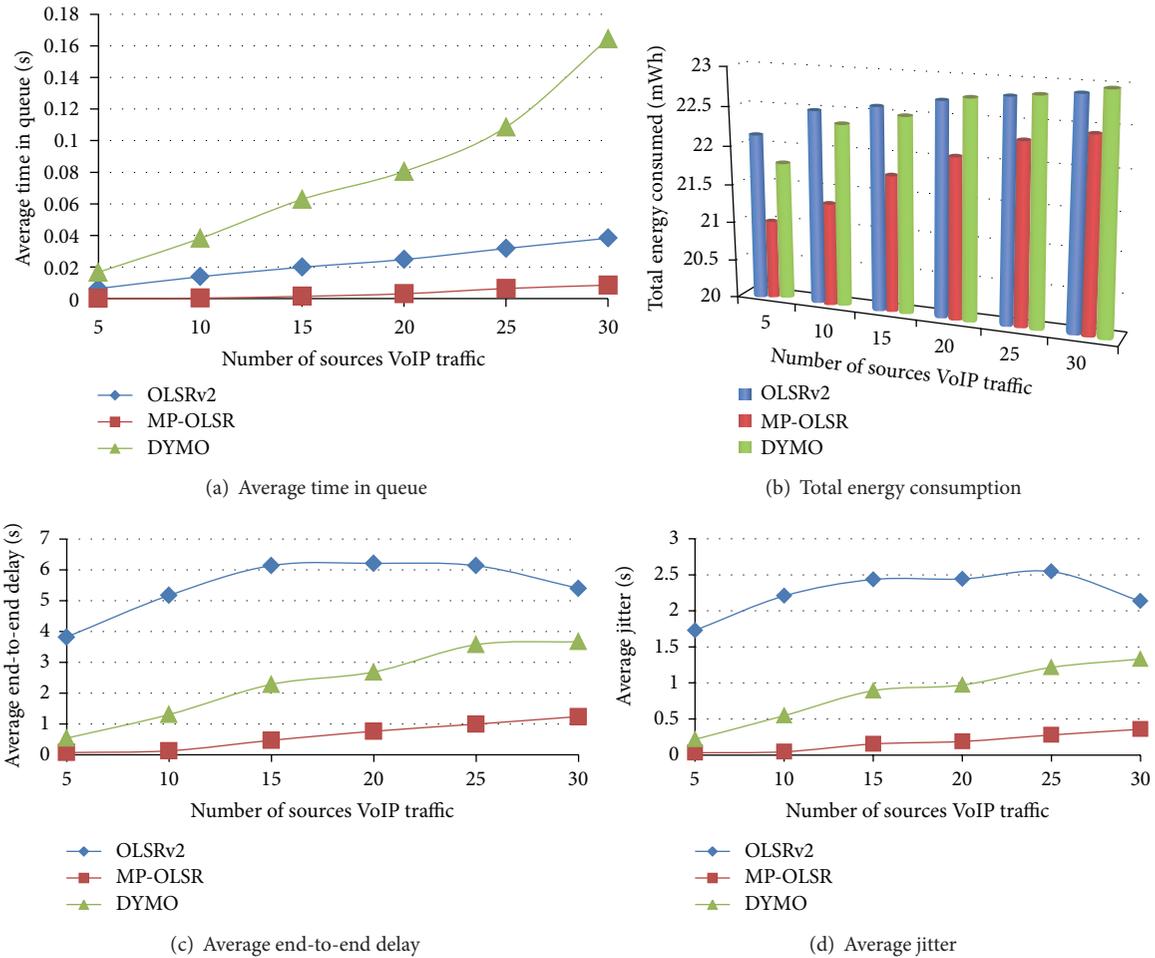


FIGURE 6: Performance metrics comparison for OLSRv2, DYMO, and MP-OLSR for different VoIP traffic load with 60 nodes.

From (6), average jitter totally depends on packets transmission delay. Therefore, the results for this metric as shown in Figure 5(b) are consistent with the end-to-end delay results. For the DYMO protocol, the average jitter median over all scenarios is 0.46 [Sec]. Then again, the OLSRv2 has the highest average jitter with respect to network size in all experiments with a median of 1.8 [Sec]. Meanwhile, MP-OLSR has the lowest average jitter that reaches 34 times lower than OLSRv2 and 9 times lower than DYMO as an average overall scenario with VoIP traffic model.

4.1.2. The Traffic Load. In the second VoIP application scenario, we deployed 60 mobile nodes to evaluate the impact of changing traffic load (number of source-destination pairs) from 5 to 30 connections on the VoIP performance evaluation metrics. The results obtained from this scenario were compared for the selected routing protocols as shown in Figure 6. Since VoIP traffic is a multimedia traffic type, the evaluation metrics that depend on the time and delay are very important to be investigated. Another metric that we compared in this scenario is the FIFO average time in queue as a metric of network congestion. As presented in Figure 6(a), the average time in queue for the considered

protocols was consistent and it got higher as the number of VoIP traffic increased. However, DYMO provides the longest average time in queue among the three protocols. The MPR mechanism used in OLSRv2 and MP-OLSR reduced the control message overhead and average time in queue in both protocols in comparison to DYMO. Moreover, the queue monitoring mechanism that has been proposed in MP-OLSR has minimized the average time in queue in this protocol, since the number of packets in the queue has been used as a metric for the link cost between nodes during route computation. The overall median values for the average jitter are 0.002, 0.02, and 0.7 [Sec] for MP-OLSR, OLSRv2, and DYMO respectively. Although the average time in queue is one of delays that caused average end-to-end delay, its small value did not affect the overall total delay even when the network size or load traffic was changed. The shortest delay in queue is clear in MP-OLSR graph and its value is close to zero in most scenarios. This is due to the multipath nature of MP-OLSR.

The average of total consumed energy was obtained from (1) based on the generic energy model as explained in Section 3.5. It is obvious from Figure 6(b) that increasing the network traffic load increased the total consumed energy

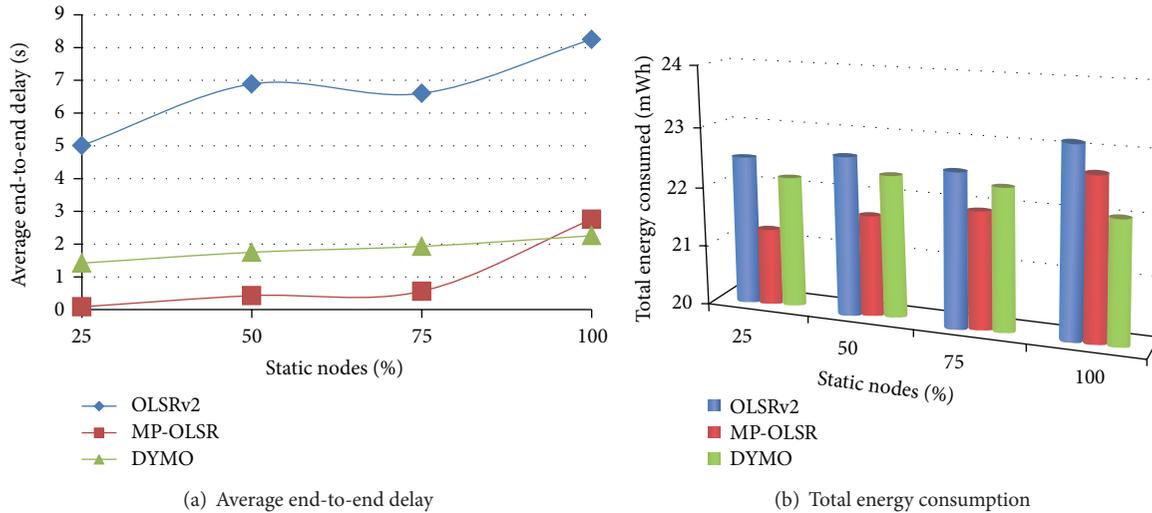


FIGURE 7: Performance metrics comparison for OLSRv2, DYMO, and MP-OLSR for different percentage of static nodes with 10-VoIP traffic and 60 nodes.

for MP-OLSR, DYMO, and OLSRv2. In this context, MP-OLSR consumed 5% less energy in the low traffic scenarios and this percentage decreased to only 2% in the heavy load scenarios compared to OLSRv2. In the scenarios with traffic sources from 20 to 30, DYMO consumed more energy than OLSRv2 because of the higher control message overhead in the congested networks, while in the light load scenario it has slightly lower consumed energy compared to OLSRv2.

As presented in Figure 6(c) the average end-to-end delay is consistent between MP-OLSR and DYMO and it has gotten higher as the number of VoIP traffic increased. However, OLSRv2 end-to-end delay is the highest among the three protocols and it is increased as the traffic load increased until 20 traffic sources, then the delay goes down to be 5.4 [Sec]. Although MP-OLSR performed well compared to DYMO and OLSRv2, its delay is increased as the network traffic load increased and reached 1.24 [Sec] with 30 traffic sources which is equal to 18 times the delay in the lowest traffic scenario with 5 traffic sources. This is because in the heavy traffic networks the number of transmitted packets that need to be handled at the intermediate nodes is increased and the propagation delay is increased leading to higher end-to-end delay.

Since average jitter depends on average end-to-end delay, the behavior of the three protocols in terms of average jitter is as same as their behavior in terms of average end-to-end delay as it is obvious in Figure 6(d). In overall scenarios, the median value of the average jitter for the MP-OLSR is 0.61 [Sec]. This value is lower than OLSRv2 and DYMO by 6 times and 9 times respectively.

4.1.3. Number of Static Nodes. The effect of nodes mobility was investigated in the third VoIP simulation scenario by changing the number of static nodes as a percentage of the total number of the deployed 60 nodes. In this scenario, several experiments were conducted and the performance metrics of VoIP traffic were compared in the routing protocols. We preferred to deploy some nodes to be static during

the whole simulation time to determine their effects on the performance of the investigated routing protocols (15, 30, 45, and 60 static nodes). Figure 7 shows the results for 10 VoIP traffic connections and comparison for the selected routing protocols.

Figure 7(a) presents and compares the average end-to-end delay for 10 VoIP flows with respect to the percentage of static nodes in different routing protocols. Definitely, the performance of MP-OLSR was affected by the number of static nodes in each scenario since it performed better in high mobility scenarios. The average end-to-end delay for the MP-OLSR increased as the number of static nodes increased and it provided 22% higher delay than the DYMO in the last scenario when all nodes were static. This is because of the difficulties in selecting multiple paths in case of route recovery and loop detection. In other words, all options are limited by the static nodes. However, the DYMO performed better than OLSRv2 in all scenarios and its delay increased slightly as the number of static nodes increased. The overall results showed that the median value of the average end-to-end delay achieved by DYMO is 3.7 times lower than OLSRv2 and 3.7 times higher than the delay obtained by MP-OLSR which outperformed DYMO except in the last scenario.

Correspondingly, Figure 7(b) shows that the MP-OLSR consumed higher energy as the number of static nodes increased and it provided lower energy consumption since there are some mobile nodes in the scenarios. In contrast, OLSRv2 spent higher energy in all scenarios because route calculation is based on hop count and it is not necessary to be an energy efficient route. Consequently, DYMO was found to be better than OLSRv2 and MP-OLSR in terms of the consumed energy and end-to-end delay for VoIP applications in the networks with static nodes.

4.2. HTTP Traffic Model. In HTTP traffic applications, there are two (2) types of exchanged data between clients and servers: (i) HTTP requests sent from client to server and

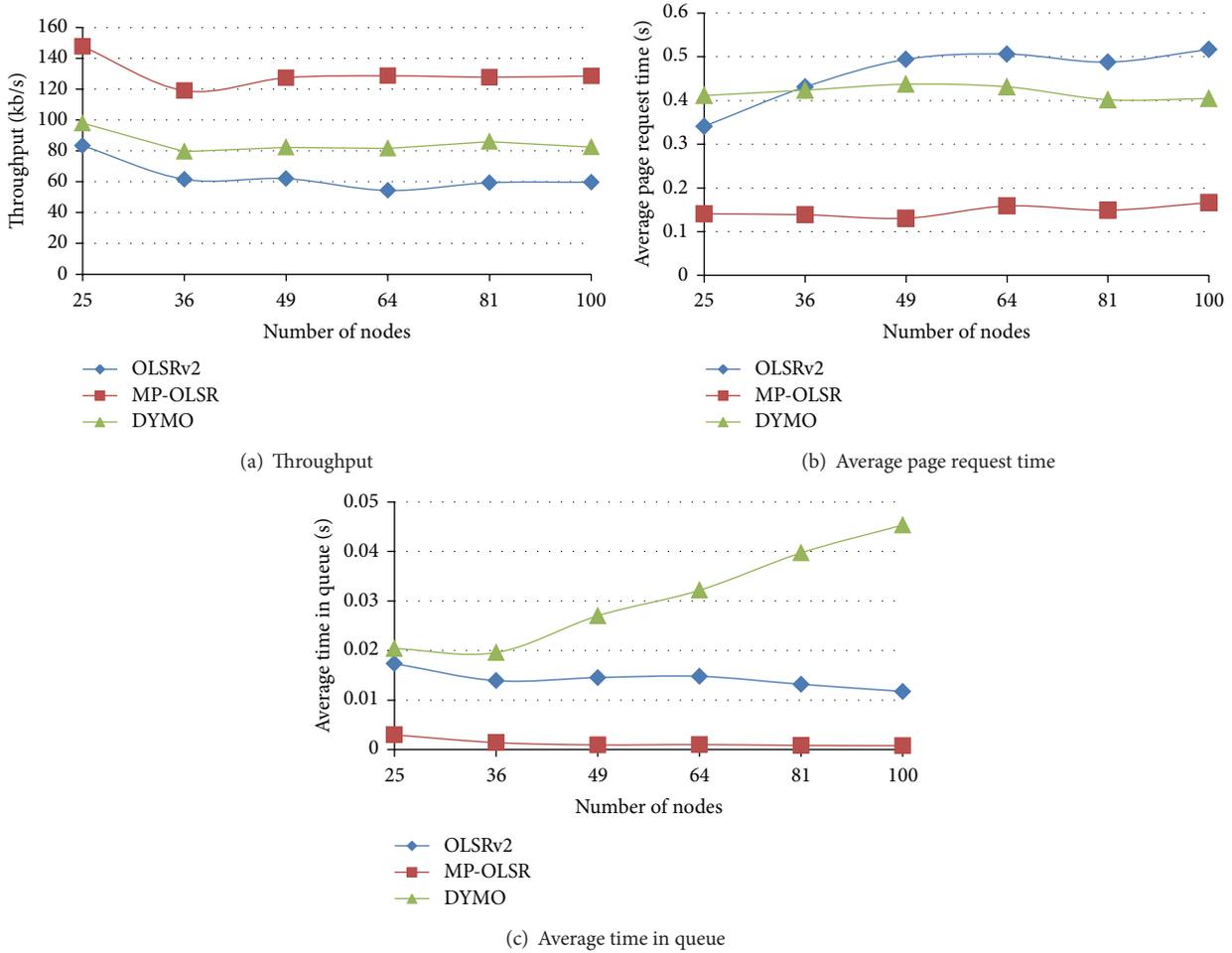


FIGURE 8: Performance metrics comparison for OLSRv2, DYMO, and MP-OLSR for different network size with 10 HTTP traffic.

(ii) the HTTP server replies. Therefore, we selected 10 nodes randomly as clients with the HTTP traffic and for each client only one node was appointed as a server to replay the client's HTTP requests.

4.2.1. The Network Size. The impact of the network size on the HTTP performance metrics for the studied routing protocols is described in Figure 8 in terms of throughput, average page request time, and average time in queue. Figure 8(a) compares the server's throughput as an average for the ten (10) servers in [kbps]. In contrast to the other applications which considered only the total bytes received at servers to calculate the throughput as explained in Section 3.7, the HTTP server's throughput also considered the total bytes sent as well as the total bytes received to calculate the throughput. The results show that MP-OLSR outperformed the other protocols: DYMO and OLSRv2, in terms of server's throughput for different network size with a maximum throughput of 147.6 [kbps] at 25 nodes. In contrast, OLSRv2 maintained the lowest throughput in all scenarios and its performance was better in 25 mobile nodes network with throughput of 83.36 [kbps]. The throughput with DYMO was

changed from 97.94 [kbps] at the scenario of 25 nodes to 79.6 [kbps] at the 36-node scenario. In all scenarios except the first one with 25 nodes, the throughput for each protocol was balanced with median values of 127.7, 82.2, and 59.7 for MP-OLSR, DYMO, and OLSRv2 respectively. This is because the HTTP server's throughput depends on the total bytes sent as a page request from the clients and this value was not affected by the number of nodes or its mobility and the difference in the server's throughput changed from the total bytes received by the server in different scenarios. The average page request time is another metric for the HTTP traffic and it is related to the clients or source nodes.

Figure 8(b) presents the results obtained from comparing this metric for the routing protocols. It was found that OLSRv2 had the longest average page request time among the three protocols except in the first scenario where DYMO needs 0.41 [Sec] which is higher by 1.2 times than OLSRv2 and 2.9 times than MP-OLSR at the same scenario. Both OLSRv2 and MP-OLSR needed higher time for page request in the last scenario with 100 nodes compared to other scenarios since they used the same concept of MPR nodes selecting mechanism which was affected by the network size.

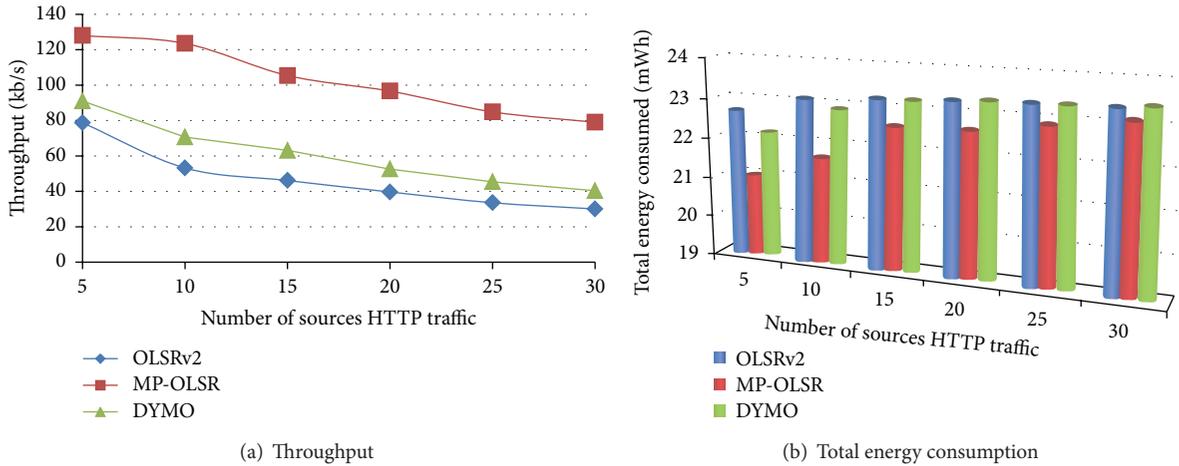


FIGURE 9: Performance metrics comparison for OLSRv2, DYMO, and MP-OLSR for different HTTP traffic load with 60 nodes.

The median values for the average page request time in all scenarios are 0.14, 0.4, and 0.5 for MP-OLSR, DYMO, and OLSRv2, respectively.

As shown in Figure 8(c), the FIFO average time in queue for the scenario with 10 HTTP traffic was consistent with the scenario with 10 VoIP connections. The average time in queue for DYMO decreased by 20% in all scenarios with HTTP traffic compared to VoIP traffic although it was the highest in all cases compared to both MP-OLSR and OLSRv2. However, the overall average time in queue for MP-OLSR with HTTP traffic was 2.6 times higher than it is in the case of VoIP traffic which has a smaller packet size but it had the shortest time in queue compared to OLSRv2 and DYMO. On the other hand, OLSRv2 showed closely similar average time in queue for both traffic types HTTP and VoIP. Moreover, it performed better than DYMO in all HTTP scenarios with a different number of nodes. This impact of the network size on the average time in queue can be more noticeable in DYMO and it increased as the number of nodes increased. However, the changes in the network size did not degrade the performance of both MP-OLSR and OLSRv2 in terms of average time in queue with HTTP applications. This means that, in DYMO, the congestion is higher because of the control message overhead which degraded the protocol performance, while in OLSRv2 and MP-OLSR, only the MPR nodes are responsible for exchange control messages and their number was not much affected by changing the total number of nodes.

4.2.2. The Traffic Load. In Figure 9(a), the average HTTP server's throughput was plotted. The effect of increasing HTTP sources on the throughput is obvious and it is reversely proportional to the number of traffic sources for all protocols in this simulation. MP-OLSR obtained higher throughput than DYMO and OLSRv2. The OLSRv2 had the lowest throughput for the different number of the HTTP sources. However, the MP-OLSR provided the highest throughput in all scenarios with a maximum value of 128 [kbps] in the scenario with 5 HTTP sources. The overall average of MP-OLSR throughput is 102 [kbps] which is 2.2 times higher

than the OLSRv2 and 1.7 times higher than the DYMO. The network traffic load affected the performance of the routing protocols in terms of throughput than the network size.

Correspondingly, as the network load increased, the total energy consumption increased. This is due to increasing the transmitted packets per nodes and this can be noticed in Figure 9(b). MP-OLSR provided 10% incremental difference in the total consumed energy between the first and the last scenario. This is because of sending more packets in different paths in the heavy traffic scenarios but its performance was better than OLSRv2 and DYMO with varying proportions. In contrast, DYMO and OLSRv2 maintained much similar consumed energy values in the high traffic scenarios which have 15 HTTP traffic and above. However, in the low traffic scenarios, DYMO provided a lower consumed energy than that provided by the OLSRv2, which is the same as in the VoIP traffic scenarios.

4.2.3. Number of Static Nodes. The average throughput of the HTTP servers is shown in Figure 10(a). The increasing number of static nodes negatively affected the throughput of MP-OLSR which decreased rapidly from 123 [kbps] in the first scenario (25% static nodes) to only 15 [kbps] at the last scenario (100% static nodes), while DYMO performed better than the MP-OLSR in the last scenario with a throughput of 16.4 [kbps].

In this context, OLSRv2 achieved the highest throughput in this scenario with 24 [kbps] though it provided the lowest throughput in the other scenarios compared to DYMO and MP-OLSR which has the highest overall throughput. Consequently, OLSRv2 is better than DYMO and MP-OLSR in terms of throughput for HTTP traffic in the networks with static nodes.

Like in the VoIP traffic scenarios, the superiority of DYMO in the nonmobility networks with HTTP traffic scenario was evident in terms of both metrics, namely, average page request time as shown in Figure 10(b) and total energy consumed as presented in Figure 10(c). However,

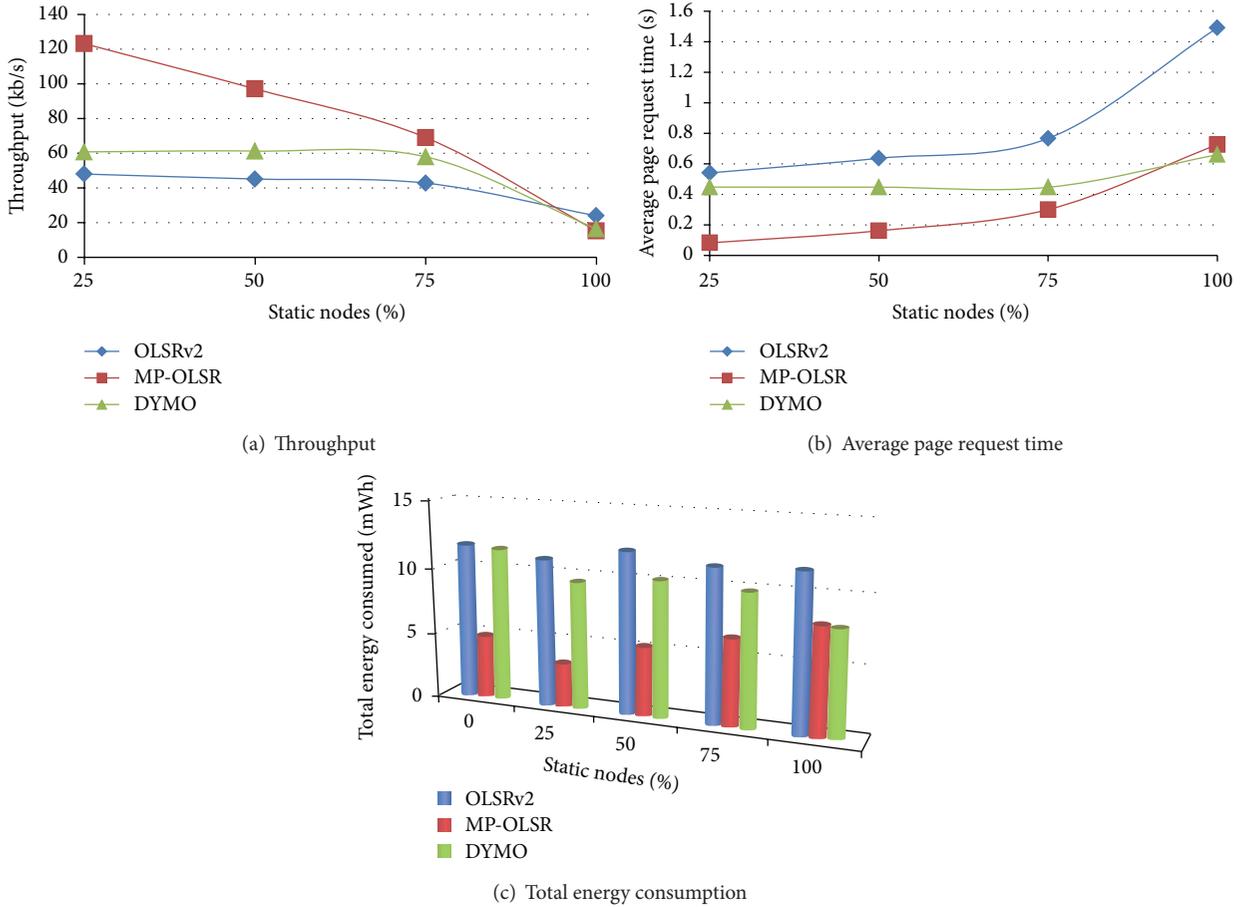


FIGURE 10: Performance metrics comparison for OLSRv2, DYMO, and MP-OLSR for different percentage of static nodes with 10-HTTP traffic and 60 nodes.

MP-OLSR is still the best for any scenario with some mobile nodes as shown in Figure 10 with different evaluation metrics.

4.3. FTP Traffic Model. For modeling FTP traffic, 10 nodes were randomly assigned as FTP clients that connect to other 10 different nodes as servers to send 100 items with the size of 512 bytes.

4.3.1. The Network Size. The average throughput as a metric for FTP flows with different nodes number was compared in Figure 11(a) in various routing protocols. Since the average throughput with FTP traffic is much higher than the average throughput obtained with HTTP because of packet size in both traffic types, thus MP-OLSR achieved 6 times higher than the average throughput in case of HTTP traffic and still outperformed both DYMO and OLSRv2. Moreover, MP-OLSR throughput did not degrade as the network size increased. Conversely, the throughput of both DYMO and OLSRv2 decreased as the network density became higher with an advance for DYMO throughput over OLSRv2 in most scenarios. However, in the scenario with 25 mobile nodes, OLSRv2 showed the highest throughput in comparison to other scenarios and at the same time higher than the DYMO

throughput at the same point. The DYMO and OLSRv2 overall average throughputs in all scenarios are 321 and 253 [kbps] respectively.

As FTP traffic was used to send data files between the client and server, the average end-to-end delay was one of the metrics that we used to evaluate the performance of the routing protocols in terms of network size. In Figure 11(b), the average end-to-end delay for the considered protocols was compared for FTP traffic model as a function of the number of mobile nodes.

Unlike the throughput metric, OLSRv2 obtained better average end-to-end delay than DYMO in all simulation scenarios with an overall median value of 3.1 [Sec] versus 4 [Sec] in DYMO. However, the performance of OLSRv2 was worse than MP-OLSR in terms of end-to-end delay. MP-OLSR had the shortest end-to-end delay in all scenarios except the first scenario with 25 nodes where it had 2.06 [Sec] versus 2.01 [Sec] for OLSRv2 which is the smallest value at the 25th node scenario. As the network size increased, the average end-to-end for MP-OLSR decreased and it performed better in high dense networks although its delay with FTP was 9 times higher than its average end-to-end delay in VoIP applications. This is because of the variance in packet size in both traffic application types FTP and VoIP.

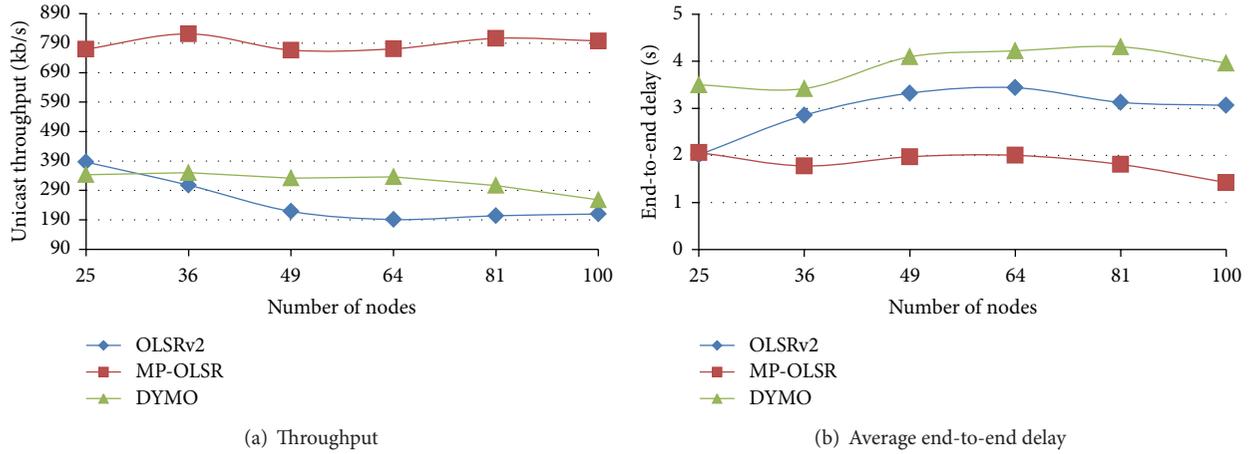


FIGURE 11: Performance metrics comparison for OLSRv2, DYMO, and MP-OLSR for different network size with 10-FTP traffic.

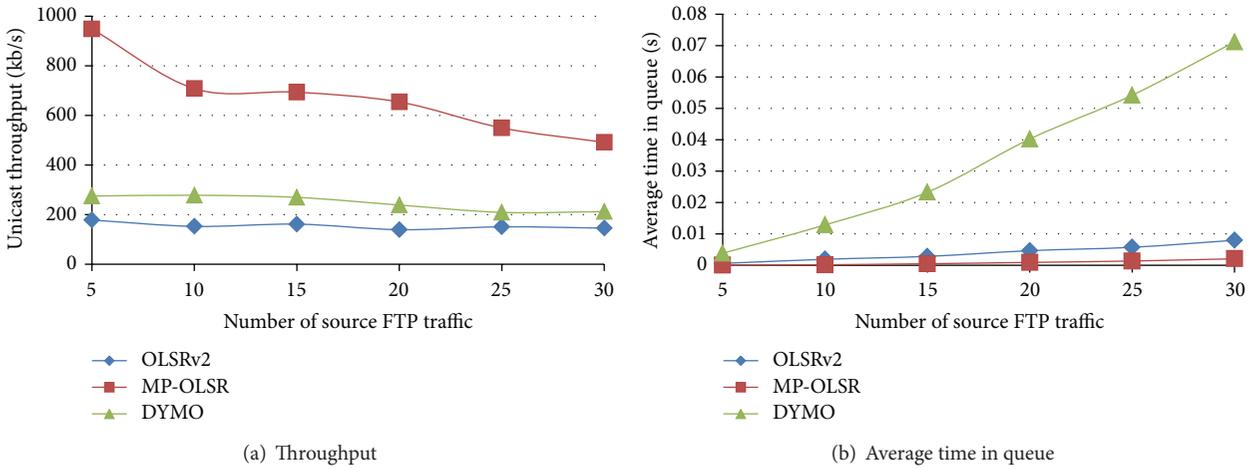


FIGURE 12: Performance metrics comparison for OLSRv2, DYMO, and MP-OLSR for different FTP traffic load with 60 nodes.

4.3.2. The Traffic Load. Like the throughput in HTTP scenarios, MP-OLSR throughput degraded as the number of FTP flows increased even though it still achieved the highest throughput overall scenarios by selecting high throughput paths among its multiple routes as presented in Figure 12(a). Meanwhile, DYMO and OLSRv2 maintained lower throughput than MP-OLSR but their throughput was not degraded much with FTP traffic numbers compared to MP-OLSR. In high load traffic, the number of the poor quality links (lossy or slow) increased and caused a drop in the average throughput; hence the percentages of the throughput drop in the high load scenario (with 30 FTP) compared to the light traffic scenario (with 5 FTP) are 48%, 23%, and 18% for MP-OLSR, DYMO, and OLSRv2, respectively. In all scenarios, OLSRv2 achieved the lowest throughput with a median value of 151 [kbps] compared to 254 [kbps] for DYMO and 674 [kbps] for MP-OLSR.

As displayed in Figure 12(b), the performance of OLSRv2, MP-OLSR, and DYMO in terms of average time in queue for different sources number was consistent to a certain degree. However, the FTP packets queue time was shorter

than HTTP and VoIP traffic models in the studied protocols for the different number of traffic sources. For MP-OLSR, the queue time was near to zero in all scenarios, while OLSRv2 queue time varied between 0.0006 [Sec] and 0.008 [Sec] in the scenario with 30 FTP. On the other hand, DYMO was the worst in terms of the average time in queue in all scenarios with an overall median value of 0.034 [Sec] which is higher than OLSRv2 by at least 9 times and MP-OLSR by 39 times.

4.3.3. Number of Static Nodes. The throughput in FTP scenarios for MP-OLSR is reduced by at least 7 times from the first scenario (25% static nodes) to the last scenario (100% static nodes) and achieved the lowest throughput in the last scenario with 109.3 [kbps] compared to 116 [kbps] and 124 [kbps] for OLSRv2 and DYMO, respectively, as presented in Figure 13(a). The throughput curves of DYMO and OLSRv2 are consistent as the number of static nodes was changed with prevalence for DYMO in all scenarios.

Inversely, OLSRv2 outperformed DYMO in all scenarios in terms of the average end-to-end delay although it provided

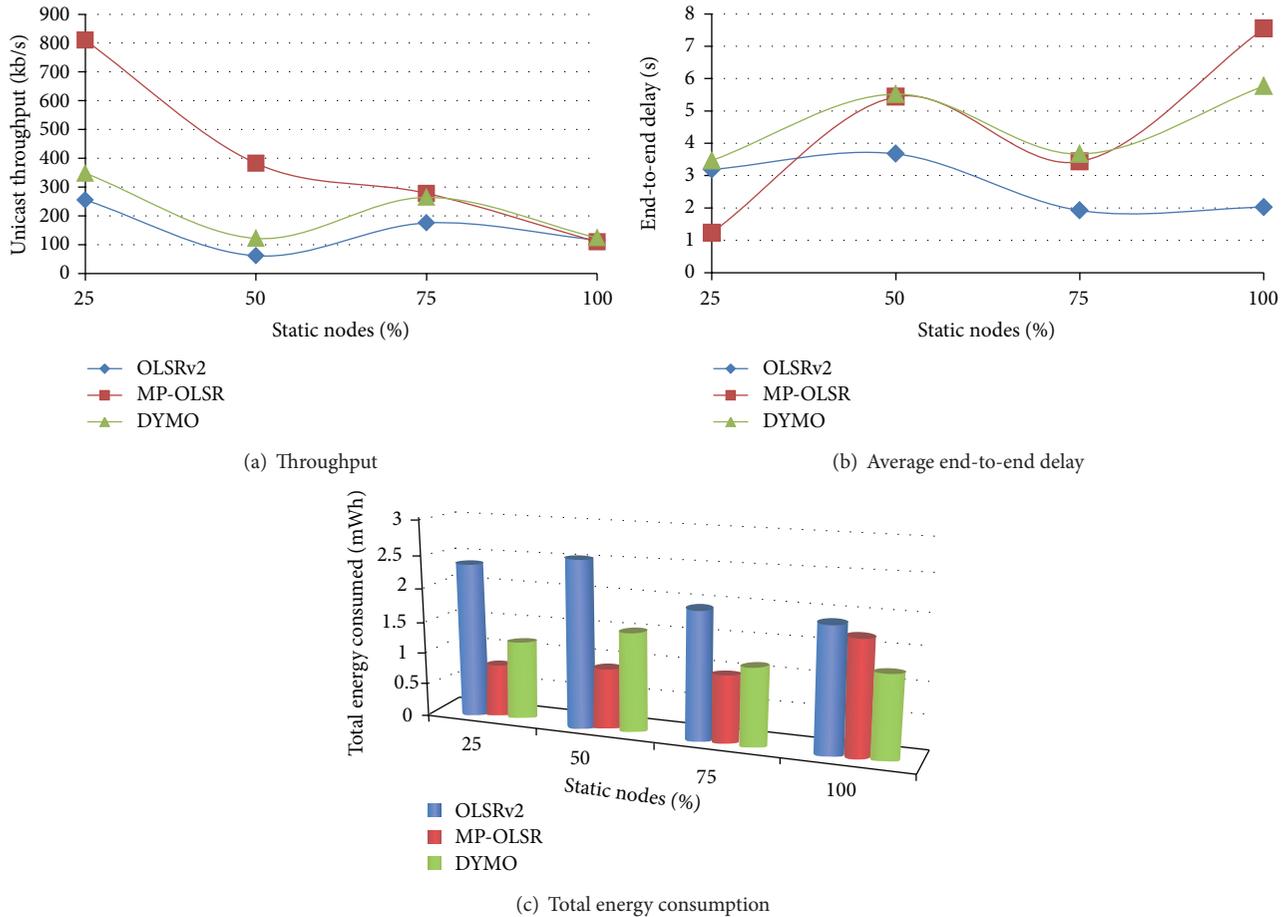


FIGURE 13: Performance metrics comparison for OLSRv2, DYMO, and MP-OLSR for different percentage of static nodes with 10-FTP traffic and 60 nodes.

a higher delay only in the first scenario compared to MP-OLSR as shown in Figure 13(b). However, as usual in the fully static node scenario, MP-OLSR provided 3.7 and 1.3 times higher average end-to-end delay than both OLSRv2 and DYMO. Therefore, the performance of MP-OLSR in terms of average-end-to-end delay was the worst in this scenario with FTP application compared to VoIP application in the previous scenarios.

On the other hand, the total consumed energy was compared to the percentage of static nodes in the selected routing protocols in Figure 13(c). As shown by the bar graph, DYMO performed well in the last scenario (100% static nodes) and consumed less power compared to both MP-OLSR and OLSRv2. In contrast, MP-OLSR achieved lower energy consumption than DYMO and OLSRv2 in the other scenarios. OLSRv2 consumed the highest energy in all scenarios regardless of the number of static nodes as presented before with HTTP and VoIP traffic models scenarios.

5. Conclusions and Future Work

This study evaluated the performance of OLSRv2, DYMO, and MP-OLSR for three internet based traffic applications,

namely, VoIP, HTTP, and FTP. The evaluation was based on changing three main parameters in the simulation model, namely, network size, number of traffic connections, and percentage of static nodes. To determine the effect of these parameters on the performance of the investigated routing protocols, we considered a variety of evaluation metrics for different traffic types, namely, throughput, end-to-end-delay, average jitter, average time in queue, page request time, and energy consumption. The simulation results obtained in the study provided evidence of the significant benefits of MP-OLSR which has the best performance over DYMO and OLSRv2 in terms of all investigated performance metrics for most scenarios in high density, heavy congestions, and high mobility environments as expected for all traffic types. We also observed that DYMO has the best performance in terms of throughput, average end-to-end delay, average page request time, and consumed energy in the network with 100% static nodes compared with OLSRv2 and DYMO. OLSRv2 outperformed DYMO in terms of average time in queue in all simulations. Although the results of this study provided evidence of the efficiency of MP-OLSR, it has a few limitations in terms of QoS in the low mobility scenarios and consumed energy in idle state as it was observed to consume highest

energy. Therefore, further investigation on optimal energy-based route algorithm for better energy saving during idle mode is essential.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgment

The research was supported by Ministry of Higher Education (MOHE), Malaysia, under the Grant scheme no. LRGS/TD/2011/UKM/ICT/02/02.

References

- [1] J. M. Hernández-Muñoz, J. B. Vercher, L. Muñoz et al., "Smart cities at the forefront of the future internet," in *The Future Internet*, pp. 447–462, Springer, Berlin, Germany, 2011.
- [2] H. Schaffers, A. Sällström, M. Pallot, J. M. Hernández-Muñoz, R. Santoro, and B. Trousse, "Integrating living labs with future internet experimental platforms for co-creating services within smart cities," in *Proceedings of the 17th International Conference on Concurrent Enterprising (ICE '11)*, pp. 1–11, Aachen, Germany, June 2011.
- [3] C. Balakrishna, "Enabling technologies for smart city services and applications," in *Proceedings of the 6th International Conference on Next Generation Mobile Applications, Services and Technologies (NGMAST '12)*, pp. 223–227, Paris, France, 2012.
- [4] J. Macker and S. Corson, "Mobile ad hoc networking (MANET): routing protocol performance issues and evaluation considerations," RFC 2501, IETF, 1999.
- [5] J. Macker and S. Corson, "Mobile ad hoc networks (MANET)," 1997.
- [6] G. Cardone, P. Bellavista, A. Corradi, and L. Foschini, "Effective collaborative monitoring in smart cities: converging MANET and WSN for fast data collection," in *Proceedings of the 4th ITU Kaleidoscope Academic Conference: The Fully Networked Human? Innovations for Future Networks and Services (K '11)*, pp. 1–8, Cape Town, South Africa, December 2011.
- [7] M. Bansal, R. Rajput, and G. Gupta, "Mobile ad hoc networking (MANET): routing protocol performance issues and evaluation considerations," 1999.
- [8] X. Zhao, Z. You, and H. Wan, "A novel two-terminal reliability analysis for MANET," *Journal of Applied Mathematics*, vol. 2013, Article ID 216186, 9 pages, 2013.
- [9] The Internet Engineering Task Force (IETF), <http://www.ietf.org>.
- [10] T. Clausen, J. Yi, and A. C. de Verdiere, "LOADng: towards AODV version 2," in *Proceedings of the IEEE Vehicular Technology Conference (VTC Fall '12)*, pp. 1–5, Quebec City, Canada, 2012.
- [11] C. E. Perkins and E. M. Royer, "Ad-hoc on-demand distance vector routing," in *Proceedings of the 2nd IEEE Workshop on Mobile Computing Systems and Applications (WMCSA '99)*, pp. 90–100, New Orleans, La, USA, February 1999.
- [12] D. B. Johnson, "The dynamic source routing protocol for mobile ad hoc networks," draft-ietf-manet-dsr-09.txt, 2003.
- [13] R. Ogier, F. Templin, and M. Lewis, "Topology dissemination based on reverse-path forwarding (TBRPF)," RFC, 2004.
- [14] T. Clausen and P. Jacquet, "Optimized link state routing protocol (OLSR) RFC3626-OLSR," 2003.
- [15] T. Clausen, C. Dearlove, and P. Jacquet, "The optimized link state routing protocol version 2," draft-ietf-manet-olsrv2-00, 2006.
- [16] I. Chakeres and C. Perkins, "Dynamic MANET on-demand (DYMO) routing," draft-ietf-manet-dymo-14, 2008.
- [17] J. Dowdell, S. Ratliff, and C. E. Perkins, "Dynamic MANET On-demand (AODVv2) Routing," 2013.
- [18] J. Yi, A. Adnane, S. David, and B. Parrein, "Multipath optimized link state routing for mobile ad hoc networks," *Ad Hoc Networks*, vol. 9, no. 1, pp. 28–47, 2011.
- [19] J. Friginal, D. de Andrés, J.-C. Ruiz, and P. Gil, "Towards benchmarking routing protocols in wireless mesh networks," *Ad Hoc Networks*, vol. 9, no. 8, pp. 1374–1388, 2011.
- [20] N. Liu and W. K. Seah, "Performance evaluation of routing metrics for community wireless mesh networks," in *Proceedings of the 7th International Conference on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP '11)*, pp. 556–561, Adelaide, Australia, December 2011.
- [21] S. Mahfoudh and P. Minet, "Survey of energy efficient strategies in wireless ad hoc and sensor networks," in *Proceedings of the 7th International Conference on Networking (ICN '08)*, pp. 1–7, Cancun, Mexico, April 2008.
- [22] L. Atzori, A. Iera, and G. Morabito, "The internet of things: a survey," *Computer Networks*, vol. 54, no. 15, pp. 2787–2805, 2010.
- [23] A. Gluhak, S. Krco, M. Nati, D. Pfisterer, N. Mitton, and T. Razafindralambo, "A survey on facilities for experimental internet of things research," *IEEE Communications Magazine*, vol. 49, no. 11, pp. 58–67, 2011.
- [24] R. Asokan, A. M. Natarajan, and C. Venkatesh, "Quality-of-service routing using path and power aware techniques in mobile ad hoc networks," *Journal of Computer Systems, Networks, and Communications*, vol. 2008, Article ID 160574, 7 pages, 2008.
- [25] B. Goode, "Voice over internet protocol (VoIP)," *Proceedings of the IEEE*, vol. 90, no. 9, pp. 1495–1517, 2002.
- [26] J. Postel and J. Reynolds, "File transfer protocol," 1985.
- [27] R. Fielding, J. Gettys, J. Mogul et al., "Hypertext transfer protocol-HTTP/1.1," RFC 2616, 1999.
- [28] T. Clausen, C. Dearlove, and J. Dean, "RFC 6130 mobile ad hoc network (MANET) neighborhood discovery protocol (NHDP)," 2011.
- [29] T. Clausen, C. Dearlove, J. Dean, and C. Adjih, "Generalized mobile ad hoc network (MANET) packet/message format," Internet Engineering Task Force, RFC, vol. 5444, 2009.
- [30] T. Clausen and C. Dearlove, "RFC5497: representing multi-value time in mobile ad hoc networks (MANETs)," Std. Track, 2009, <http://www.ietf.org/rfc/rfc5497.txt>.
- [31] T. Clausen, C. Dearlove, and B. Adamson, "RFC5148: jitter considerations in mobile ad hoc networks (MANETs)," Informational, 2008, <http://www.ietf.org/rfc/rfc5148.txt>.
- [32] T. Clausen, U. Herberg, and R. Cole, "Definition of managed objects for the optimized link state routing protocol version 2," 2013.
- [33] G. Cervera, M. Barbeau, J. Garcia-Alfaro, and E. Kranakis, "A multipath routing strategy to prevent flooding disruption attacks in link state routing protocols for MANETs," *Journal of*

Network and Computer Applications, vol. 36, no. 2, pp. 744–755, 2013.

- [34] S. Vutukury and J. Garcia-Luna-Aceves, “An algorithm for multipath computation using distance-vectors with predecessor information,” in *Proceedings of the 8th International Conference on Computer Communications and Networks*, pp. 534–539, Boston, Mass, USA, 1999.
- [35] EXata 3.1 simulator, <http://www.scalable-networks.com/products>.
- [36] Qualnet simulator, <http://www.scalable-networks.com/products>.
- [37] C. E. Perkins, “Mobile ad hoc networking terminology,” draft-ietf-manet-term-00.txt, 1997.
- [38] C. Bettstetter, G. Resta, and P. Santi, “The node distribution of the random waypoint mobility model for wireless ad hoc networks,” *IEEE Transactions on Mobile Computing*, vol. 2, no. 3, pp. 257–269, 2003.
- [39] G. A. Thom, “H.323: the multimedia communications standard for local area networks,” *IEEE Communications Magazine*, vol. 34, no. 12, pp. 52–56, 1996.
- [40] J. Rosenberg, H. Schulzrinne, G. Camarillo et al., “RFC 3261: SIP: session initiation protocol,” IETF, 2003, <http://www.ietf.org/rfc/rfc3261.txt>.
- [41] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson, “RFC 3550: RTP: a transport protocol for real-time applications,” status: proposed standard, 2003.
- [42] S. Mahfoudh and P. Minet, “EOLSR: an energy efficient routing protocol in wireless ad hoc and sensor networks,” *Journal of Interconnection Networks*, vol. 9, no. 4, pp. 389–408, 2008.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

