Cluster Computing (2020) 23:13–27
https://doi.org/10.1007/s10586-019-02937-x

AQ-Routing: mobility-, stability-aware adaptive routing protocol for data routing in MANET–IoT systems

Abdellatif Serhani[1] · Najib Naja[1] · Abdellah Jamali[2]

Received: 30 October 2017 / Revised: 25 December 2018 / Accepted: 4 May 2019 / Published online: 10 May 2019

Abstract

Internet of Things, is an innovative technology which allows the connection of physical things with the digital world through the use of heterogeneous networks and communication technologies. In an IoT system, a major role is played by the wireless sensor network as its components comprise: sensing, data acquiring, heterogeneous connectivity and data processing. Mobile ad-hoc networks are highly self reconfiguring networks of mobile nodes which communicate through wireless links. In such a network, each node acts both as a router and host at the same time. The interaction between MANETs and Internet of Things opens new ways for service provision in smart environments and challenging issues in its networking aspects. One of the main issues in MANET–IoT systems is the mobility of the network nodes: routing protocol must react effectively to the topological changes into the algorithm design. We describe the design and implementation of AQ-Routing, and analyze its performance using both simulations and measurements based on our implementation. In general, the networking of such a system is very challenging regarding routing aspects. Also, it is related to system mobility and limited network sensor resources. This article builds upon this observation an adaptive routing protocol (AQ-Routing) based on Reinforcement Learning (RL) techniques, which has the ability to detect the level of mobility at different points of time so that each individual node can update routing metric accordingly. The proposed protocol introduces: (i) new model, developed via Q-learning technique, to detect the level of mobility at each node in the network; (ii) a new metric, called Q_{metric}, which account for the static and dynamic routing metrics, and which are combined and updated to the changing network topologies. The protocol can efficiently handle network mobility by a way of preemptively adapting its behaviour thanks to the mobility detection model. The presented results of simulation provide an effective approach to improve the stability of links in both static and mobile scenario and, hence, increase the packet delivery ratio in the global MANET–IoT system.

Keywords Internet of Things · Mobile ad-hoc networks · AQ-Routing · RPL Routing protocol · ETX · Adaptive routing · Reinforcement learning · Q-learning

1 Introduction

Internet of Things (IoT) [1], refers to the stringent connectedness between digital and physical world [2], represents a part of the future internet paradigm. The interconnection of smart objects and its interoperability with global communications serve as a main incorporated idea in IoT systems. all the objects of different kinds of our daily life spanning from smart phones, sensors or devices are associated with network enabled objects (like RFID) can communicate with each other and makes a part of Internet [3]. There is a huge use of IoT in several domains like medical aids, home automation, industrial automation,

✉ Abdellatif Serhani
abdollatiif@gmail.com

Najib Naja
n.naja@inpt.ac.ma

Abdellah Jamali
abdellah.jamali@uhp.ac.ma

[1] RAI2S, INPT, University of Mohammed V, Rabat, Morocco

[2] RI2M, ENSA, University of Hassan I, Settat, Morocco

Table 3 AQ-Routing parameters

Attribute	Value
$\gamma^{MAX}, \gamma^{MAXI}$	0.95
α	0.95
T_{HELLO} (s)	2
Encounter lifetime (s)	6
Observation time (s)	6
Loop probe interval (s)	0.5

mobile health care, electricity transmission and distribution etc. In an IoT system, a major role is played by the wireless sensor network (WSN) as its components comprise: sensing, data acquiring, heterogeneous connectivity and data processing.

Mobile ad hoc networks (MANETs) emerge as important technologies visioning to support robust and efficient operations in mobile wireless networks by incorporating routing functionality into mobile nodes. Such networks are envisioned to have dynamic, sometimes rapidly-changing, random, multihop topologies which are likely composed of relatively bandwidth-constrained wireless links. Ad hoc networks technology can provide an extremely flexible method for establishing communications for fire/safety/rescue operations or other scenarios requiring rapidly-deployable communications with survivable, efficient dynamic networking.

Ad hoc networks consists of mobile platforms referred to as "nodes" which are free to move about arbitrarily. Each node is equipped with wireless transmitters and receivers using antennas which may be omnidirectional, highly-directional, possibly steerable, or some combination thereof. Ad hoc networks have several salient characteristics (i) dynamic topologies: the network topology may change randomly and rapidly at unpredictable times; (ii) Bandwidth-constrained, variable capacity links; (iii) Energy-constrained operation: all of the nodes in a MANET may rely on batteries for their energy and (iv) Limited physical security.

Interaction between wireless sensor and mobile ad-hoc networks with the internet of things allows the creation of a new MANET–IoT systems and IT-based networks [1]. in such systems we can find different options in which an IoT device can be connected to a MANET node. It can also be a MANET node, and be either connected to an Internet node or be an actual Internet node. Routing for IoT devices in an Internet connected MANET requires some MANET topology. The cluster topology proposed in [4] is for MANET nodes which are either IoT devices or controllers of connected IoT devices. Routing for IoT nodes in MANETs and IoT devices connected to MANET nodes use MANET routing protocols. Low Powered and Lossy Networks is shortly called LLN, is a type of networks which contains a limited number of routers and the connection between the routers is also limited [5]. The Routing Protocol for Low-power and Lossy Networks (RPL) has been proposed by the IETF Working Group [6] for LLN implementations based on 6LoWPAN. RPL is a distance vector routing protocol designed for Low powered Lossy Networks which uses the principles of IPV6. As described in the standard [6], RPL creates a routing topology in the form of a destination-oriented directed acyclic graph (DODAG): a directed graph without cycles, oriented toward a root node. By default, each node maintains multiple parents toward the root; a preferred one is used for actually forwarding data packets upward toward the root, while the others are kept as backup routes. The topology is created and maintained via control packets called DODAG information objects (DIO), advertised by each node. The packet contains the routing metric (e.g., link quality, residual energy) and an objective function used by each node to select the parents among its neighbors. To support the dual traffic pattern from the root to the devices, called point-to-multipoint communication in RPL, the standard requires additional control messages called Destination Advertisement Object (DAO), which supports and down traffic and it is used to broadcast receiver information towards DODAG. These messages are propagated "upward" in the DODAG topology, via a parent that may coincide with the preferred parent, therefore establishing "downward" routes along the way.

In MANET, developing efficient routing protocols has always been a challenging task because of the specific characteristics aforementioned. An efficient routing protocol should be able to react appropriately to these topology changes and the protocol's amount of control information should be limited due to the bandwidth constraint. A wide range of ad hoc routing protocols have been proposed [7–9] but all are only suboptimal. A given protocol can outperform other protocols depending on the network context under which it is evaluated. To deal with the changing context, some routing models try to adapt their behavior to the context of the network, which both varies in time and space. Nodes can adapt their protocols at any moment.

Hoebeke et al. [10] proposed an adaptive multi-mode routing that has the ability to let individual nodes change their routing technique. The protocol is capable of minimally offering the performance of either proactive or reactive routing. As a result, this protocol is only beneficial when adapting to medium and large time-scale network changes.

To adapt the protocol to any given state of the network, the authors in [11] based on Zone Routing Protocol (ZRP) to propose a model known as Adaptive Zone Routing Protocol (AZRP) that can adapt to any given state of the network in order to enhance the performance with the use of variable zone radius for every node based on rate of route failure measured in route failures/node. The model helps to reduce route acquisition times and decrease bandwidth loss. From the same perspective, Seba et al. [12] proposed a protocol, named Adaptive Routing Protocol for MANETs (ARPM), that takes advantage of reactive and proactive approaches by dynamically adapting the routing mechanism with respect to the mobility degree of nodes based on a metric named 'neighbor change frequency'. It takes a trade-off between proactive and reactive protocol when the network becomes more dynamic.

Another interesting idea is to use the ideas of Cross-Layer Design to switch the routing node between proactive and reactive modes. That is, the authors in [13], proposed an adaptive cross layer protocol (ACRP) by incorporating fuzzy logic into the routing mode selector to change current node routing accordingly. The model used the number of link breakage (LB) per second as mobility metric and the interface queue (IFQ) length as the traffic metric. ACRP model help to enhance routing efficiency by providing a Quality of Service (QoS) in terms of average End-to-End Delay.

To build stable and optimum routes, the authors in [14] proposed an adaptive cross layer design to couple routing discovery with received signal strength information and speed of mobile nodes. Each node when receiving signal strength is used to calculate a distance to the transmission range boundary of the transmitting node and its safe distance from transmission range boundary and make a decision on forwarding the RREQ. The model leads to the formation of more stable routes and reduce the number of link breaks and increase the reliability of a routing path.

In [15], the Self-adaptive proactive scheme is proposed and is claimed to be the first routing scheme that adapt routing metrics to the mobility of the network, which has the ability to detect the mobile ad-hoc network mobility states and self-adapt routing metrics accordingly. The model tries to detect the mobility states of the network based on the mobility state indicator (MSI) computed at each node and uses an adaptive routing metrics subject to the detected mobility states to improve routing performance while reducing the number of dropped packets caused by failing to find route for sending data. However the following drawbacks were identified: (i) each node calculate its AER (Average Encounter Rate) and all AERs are exchanged between nodes such that each node can examine AERs of all other nodes: this process generates an unacceptable amount of overhead. (ii) due to the mobility of nodes, network state can change very often between static and mobile resulting in node 'flickering' in terms of routing metric, thus a consistent portion of traffic, in the time necessary to allow nodes to be synchronized for switching to a specific metric, may be lost. (iii) The model proposed only two states of mobility in the network: relatively static or mobile. This is a highly simplified picture of the reality of ad hoc networks which we can have a mobile part and a relatively static one.

This inspires the Reinforcement Learning based Adaptive Routing (AQ-Routing) model proposed in this paper. The key contributions of this paper are as follows:

- ★ Proposing a new model to detect the mobility of the network based on Reinforcement Learning (RL) techniques [16, 17]. The mobility states within the network is learned;
- ★ Proposing a new metric named Q_{metric} based on two known metrics: Expected Transmission count metric (ETX) [18] that helps nodes determine the highest throughput path for routing and Mobile Factor metric (MF) [19] that aims to find a stable path for routing instead of shortest path; therefore routing performance enhances in mobility conditions. The two metrics are then combined together using a recent knowledge of mobility level changes to yield a new Q_{metric}.

The rest of this paper is organized as follows. In Sect. 2, we present some assumptions and definitions. We review the Reinforcement Learning (RL) and Q-learning techniques and some useful metrics. In Sect. 3, we develop the new mobility degree detection model. Section 4 presents our AQ-Routing model. In Sect. 5, we discuss the simulation method and evaluate the performance of the proposed algorithm in comparison with OLSR Standard (OLSR-SD) and OLSR-ETX based on ETX metric. Finally, we present our conclusion in Sect. 6.

2 Assumption and definition

2.1 Reinforcement learning (RL)

Machine learning (ML) was introduced as a technique for artificial intelligence (AI) [20]. Over time, its focus evolved and shifted more to optimize a performance criterion using example data or past experience on given tasks without the need for re-programming. In the last decade, machine learning techniques have been used extensively for a wide range of tasks including classification, regression and density estimation in a variety of application areas such as bioinformatics, speech recognition, spam detection, computer vision, fraud detection and advertising networks.

During the past decade, wireless ad-hoc networks have seen increasingly intensive adoption of advanced machine learning techniques. In [21], a short survey of machine learning algorithms applied in MANETs for information processing and for improving network performance was presented. They discussed the most widely applied distributed ML techniques (i.e., reinforcement learning, swarm intelligence, mobile agents and real time heuristic search), together with their advantages, disadvantages and properties relevant to the wireless ad hoc scenario. The authors of [22] discussed applications of three popular machine learning algorithms (i.e., reinforcement learning, neural networks and decision trees) at all communication layers in the WSNs. Those paradigms and algorithms are well suited to be used in a distributed environment for solving different online problems, like routing in WSNs or MANETs and fall into the categories of supervised, unsupervised and reinforcement learning [23]. In the first category, all data is labeled and the algorithms learn to predict the output from the input data. The goal is to approximate the mapping function so well that when we have new input data that we can predict the output variables for that data. Supervised algorithms can be further grouped into regression and classification problems. In contrast to supervised learning, unsupervised learning algorithms are not provided with labels (i.e., there is no output vector). Basically, the goal of an unsupervised learning algorithms is to classify the sample sets to different groups (i.e., clusters) by investigating the similarity between the input samples. Unsupervised learning problems can be further grouped into clustering and association problems. The third category includes reinforcement learning algorithms, in which the agent learns by interacting with its environment (i.e., online learning). Finally, some problems sit in between both supervised and unsupervised learning. Many real world machine learning problems fall into this area. This is because it can be expensive or time-consuming to label data as it may require access to domain experts, whereas unlabeled data is cheap and easy to collect and store. These hybrid algorithms (often termed as semi-supervised learning) aim to inherit the strengths of these main categories, while minimizing their weaknesses [24].

Under the Markovian and stationarity assumptions, a discrete-time Markov Decision Process (MDP) [25] is defined by: (i) a decision epochs $T = \{1, 2, \ldots, N\}, N \leq \infty$; (ii) a finite state space $S = \{s_i\}_{i=1,\ldots,n}$; (iii) a finite and nonempty set of available control actions $A(s_i) = \{a_k\}_{k=1,\ldots,|A(s_i)|}$ associated to each state $s_i \in S$; (iv) a real-valued one-step reward function $r : S \times A \to \mathbb{R}$, where $r = r(s_i, a_k)$ is the reward incurred by the system as it is in state $s_i \in S$ and action $a_k \in A(s_i)$ is chosen; (v) the probability $p(s_i, s_j, a_k)$ that, in the next time step, the system will be in state $s_j \in S$ when action $a_k \in A(s_i)$ in state $s_i \in S$ is chosen; the set of these transition probabilities constitute the transition matrix $\mathbf{P}$. (vi) $\gamma \in [0, 1]$ is the discount factor, which represents the difference in importance between future rewards and present rewards. In the following, we will denote with $s(t)$, $a(t)$, and $r(t)$ the state, the action and the reward of the system at time t, respectively. A stationary policy is a function $\pi : S \to A$, which maps every state $s_i \in S$ to a unique control action $a_k \in A(s_i)$. When the system operates under policy π, the MDP reduces to a discrete-time Markov chain and the following expected discounted reward is earned:

$$R_\pi = \limsup_{n\to\infty} \frac{1}{t} E_\pi \left\{ \sum_{t'=1}^{t} \gamma^{t'-1} r(t') \right\}, \tag{1}$$

where $0 \leq \gamma \leq 1$ is the discount rate and the subscript π specifies that the controller operates under policy π. The MDP problem is the determination of the optimal policy π^* minimizing cost (1). They have a large number applications, both practical and theoretical, and various algorithms have been developed to solve them. If the probabilities or rewards are unknown, the problem is one of reinforcement learning [26]. Two elements make reinforcement learning powerful: i) they don't need complete knowledge of the system statistical characteristics; ii) they can cope with non-stationary scenarios. Thanks to these two key components, reinforcement learning can be used in large environments. Reinforcement learning uses a formal framework defining the interaction between a learning agent and its environment in terms of states, actions, and rewards. Agent can use its experience to improve its performance over time. The reinforcement learning agent and its environment interact over a sequence of discrete epochs. At each epoch t, the agent receives some representation of the environment's state $s_t \in S$, and on that basis selects an action $a_t \in A(s_t)$. One epoch later, in part as a consequence of its action, the agent receives a numerical reward r_{t+1} and finds itself in a new state s_{t+1}. The value of taking action a_t in state s_t under policy $\pi(s_t, a_t)$, denoted $Q^\pi(s_t, a_t)$, is the expected return when starting from s_t, taking the action a_t and following policy π:

$$\begin{aligned} Q^\pi(s,a) &= E_\pi\{R_t \mid s_t = s, a_t = a\} \\ &= E_\pi\left\{ \sum_{k=0}^{\infty} \gamma^k r_{t+k+1} \mid s_t = s, a_t = a \right\}, \end{aligned} \tag{2}$$

where $E_\pi\{\}$ denote the expected value given that the agent follows policy π, and γ is a parameter called the discount rate that determines the present value of future rewards.

The agent's goal is to maximize the total amount of reward it receives over the long run, thus, finding all the optimal policies π^* that achieves a lot of reward over the long run. Optimal policies share the same optimal action-value function, denoted Q^*, and defined as:

$$Q^*(s,a) = \max_\pi Q^\pi(s,a) \tag{3}$$

2.2 Q-learning algorithm

$\mathfrak{Q}$-learning [16] is a form of model-free reinforcement learning. Specifically, $\mathfrak{Q}$-learning can be used to find an optimal action-selection policy for any given finite Markov decision process (MDP) and can also be viewed as a method of asynchronous dynamic programming (DP). Learning proceeds similarly to Sutton's [26] method of temporal differences (TD): an agent tries an action at a particular state, and evaluates its consequences in terms of the immediate reward or penalty it receives and its estimate of the value of the state to which it is taken. Thus, by trying all actions in all states repeatedly it works by learning an action-value function that ultimately gives the expected utility of taking a given action in a given state and following the optimal policy thereafter.

Consider a computational agent moving around some discrete, finite world, choosing one from a finite collection of actions at every epoch. At epoch t, the agent is in state s_t, and by interacting with the world, can choose its action $a_t \in A(s_t)$ accordingly. The agent receives a probabilistic reward r_t, whose mean value $R_{s_t}(a_t)$ depends only on the state and action, and the state of the world changes probabilistically to s_{t+1} according to the law:

$$Prob[s_{t+1} \mid s_t, a_t] = P_{s_t s_{t+1}}[a_t] \tag{4}$$

The task facing $\mathfrak{Q}$ learner is that of determining one optimal stationary policy π^* without initially knowing r_t nor $P_{s_t s_{t+1}}[a_t]$. For a policy π, define $\mathfrak{Q}$ values (or action-values) as:

$$Q^{\pi}(s_t, a_t) = R_{s_t}(a_t) + \gamma \sum_{t'} P_{s_t s_{t'}}[\pi(s_t)] V^{\pi}(s_{t'}), \tag{5}$$

where $V^{\pi}(s_{t'})$ denote the $\mathfrak{Q}$ values of $s_{t'}$. In other words, the $\mathfrak{Q}$ value is the expected discounted reward for executing action a_t at state s_t and following policy π thereafter. The object in $\mathfrak{Q}$-learning is to estimate the $\mathfrak{Q}$ values for an optimal policy $\pi^*(s_t) \equiv a_t^*$ and that a_t^* is an action at which the maximum values, $\max_a Q^{\pi^*}(s_t, a_t)$, is attained.

In $\mathfrak{Q}$-learning the agent's experience consists of a sequence of distinct stages or episodes. In the n^{th} episode, the agent selects and performs an action a_t and adjusts its Q_{n-1} values using a learning factor α_n, with the following update rule:

$$\begin{aligned} Q_n(s_t, a_t) &= (1 - \alpha_n) Q_{n-1}(s_t, a_t) \\ &\quad + \alpha_n \left[r_t + \gamma \max_{a_k \in A(s_{t+1})} Q_{n-1}(s_{t+1}, a_k) \right], \end{aligned} \tag{6}$$

where $\max_{a_k \in A(s_{t+1})} Q_{n-1}(s_{t+1}, a_k)$ is the best the agent thinks it can do from state s_{t+1}. The initial Q values, $Q_0(s_t, a_t)$, for all states and actions are assumed given. The convergence to the optimal policy is guaranteed if all actions and state are "sufficiently" visited [16].

$\mathfrak{Q}$-learning requires clever exploration mechanisms. Randomly selecting actions, without reference to an estimated probability distribution, shows poor performance. However, due to the lack of algorithms that provably scale well with the number of states (or scale to problems with infinite state spaces), simple exploration methods are the most practical. One such method is determined by selecting in each state s_t the best (greedy) action. However, in order to explore the state space, different actions should be selected ; fore this purpose, some algorithms use the so-called ϵ-greedy policy, determined by selecting in each state s_t the greedy action with probability $1 - \epsilon$, and a random action with probability ϵ, with $0 \le \epsilon \le 1$:

$$a^{\epsilon-greedy}(s_t) = \begin{cases} \text{argmax}_{a_k \in A(s_t)} Q(s_t, a_t), & \text{Prob} = 1 - \epsilon \\ \text{random action in} A(s_t), & \text{Prob} = \epsilon \end{cases} \tag{7}$$

The tuning of the parameter ϵ is needed to balance exploration (i.e., random action selection) and exploitation (i.e., greedy action selection based on the current best known policy). The $\mathfrak{Q}$-learning approach consists in learning the action-values function $\mathfrak{Q}$, and therefore the optimal policy, by experience, following proper update rules, starting from a given initial $\mathcal{Q}_0$ function (e.g., with values set to 0).

2.3 ETX algorithm

It is generally acknowledged that the most common metric used for routing in MANETs as the optimal route is "hop-count" [7, 27] (HOP metric). So far, it has been assessed as a good indicator for routing performance in wired and wireless networks [28, 29]. However, the network's mobility and density are not concerned in this metric. In a dense network there may be many routes of the same minimum length, with widely varying qualities; the arbitrary choice made by most minimum hop-count metrics is not likely to select the best. Because regardless of the often large differences in throughput among those paths, and ignoring the possibility that a longer path might offer higher throughput [18]. In other words, the HOP metric is not an optimal solution for routing in mobile ad hoc networks. Researchers are still investigating various metrics and routing strategies which have the ability to adapt to the mobility and/or the density changes of MANETs. De Couto et al. [18] proposed a well-known metric which account for lossy links: expected transmission count (ETX). This metric estimates the number of transmissions needed to send unicast packets by measuring the loss rate of broadcast packets between pairs of neighboring nodes. The metric's overall goal is to choose routes with high end-to-end throughput. The ETX metric incorporates the effects of link loss ratios, asymmetry in the loss ratios between the two directions of each link, and interference among the successive links of a path. Measurements on a wireless test-bed show that ETX finds routes with significantly higher throughputs than a minimum hop-count metric, particularly for paths with two or more hops.

The ETX of a link is the predicted number of data transmissions required to send a packet over that link, including retransmissions. The ETX of a route is the sum of the ETX for each link in the route. The ETX of a link is calculated using the forward and reverse delivery ratios of the link. The forward delivery ratio, d_f, is the measured probability that a data packet successfully reaches at the receiver. the reverse delivery ratio, d_r, is the probability that the ACK packet is successfully received. Mathematically, the ETX metric of a link can be calculated by:

$$ETX = \frac{1}{d_f \times d_r}, \tag{8}$$

ETX has several important characteristics: (i) it is based on delivery ratios, which directly affect throughput; (ii) it detects and appropriately handles asymmetry by incorporating loss ratios in each direction; (iii) it can use precise link loss ratio measurements to make fine-grained decisions between routes; (iv) penalizes routes with more hops, which have lower throughput due to interference between different hops of the same path [30]; (v) tends to minimize spectrum use, which should maximize overall system capacity.

In this paper, the delivery ratio d_f and d_r are measured using hello message. Each node broadcasts hello message at an average period T_{hello}. Every node remembers the messages it receives during the last w seconds $(3 \times T_{hello})$ in our implementation, allowing it to calculate the d_r from the sender at any time. Each hello message contains the number of hello messages received from each of its neighbors during the last w seconds This allows each neighbor to calculate the d_f.

The results in [29] show that with stationary nodes the ETX metric significantly outperforms hop-count. ETX does not specifically account for mobility, thus, it leads to a bad routing performance because the metric does not quickly enough react to the mobility of network.

2.4 Mobility metrics

In recent years, some studies on efficient control of network mobility have been done in the field of mobile ad hoc networks. Research community has developed a plethora of metrics to address the mobility constraint in order to enhance the performance of the overall system. Works on mobility metrics have been intended to either find stable links and paths in MANETS or to reduce route discovery control overhead by restricting the flooding of control messages used for route discovery. Adya et al. [31] proposed a new metric relied on per-hop round trip time (RTT) concerning the duration of sending and receiving a probe packet from a sender to 1-hop neighbours. The sender then updates the estimated weighted average RTTs and selects a routing path with a minimum value of RTT. However, the periodic dissemination of probe packets and probe acknowledgement packets for getting the RTT value may consume more bandwidth and cause more network contention, and therefore employing this metric is not effective in dynamic and dense networks [29]. In another approach, Khelil et al. [32] exhibited several mobility metrics that quantify a large time-scale mobility based on the pair-wise contacts between mobile nodes. Two nodes encounter each other when the distance between becomes smaller than the communication range R. The encounter is said to be lost, if the nodes leave the communication range of each other. The encounter e_{nm} of node n with node m is defined as follows:

$$e_{nm} = \{n, m, t, \Delta t\}, \tag{9}$$

with t the time of incidence of the encounter and Δt the duration of the encounter. A contact between two nodes is defined as the list of all encounters between them. A contact between two nodes begins with the first encounter between them, and ends with the last one. A contact is considered as lost if there is no encounter between both nodes. We denote by c_{nm} the contact of node n with node m. We represent cnm as a set of e_{nm}:

$$c_{nm} = \{e_{nm}\}, \tag{10}$$

The simulation results and analysis in [32] signify that the number of new encounters in a given duration is linearly proportional to the mobility and the density of the network. In other words, in a given node density, by observing new encounters in certain duration, a node can basically predict its relative velocity [33] to other nodes around.

From the same perspective, Grossglauser et al. [34] introduced the so-called last encounter age or the time elapsed since last encounter to efficiently route unicast messages to destinations. They achieve this by letting each node maintain a local database of the time and location of its last encounter with every other node in the network. This database is consulted by packets to obtain estimates of their destination's current location. This work shows the utility of encounter history or contacts even for non-delay tolerant networking. Boleng et al. [35] introduced a mobility metric named link duration defined as the time that two nodes are within the transmission range of one another. Longer lasting links create more network stability, while shorter duration links create less network stability. An average link duration metric accurately captures this effect. It combines the link change rate [35] and weights the changes by their stability.

To solve the routing problem of MANETs in a highly dynamic scenario, some routing schemes used a metric named Mobility Factor (MF) [19, 36] to consider the link stability before forwarding a packet. The mobility metric takes into consideration the mobility of nodes in terms of neighbour sets. Let $A \Delta B$ denote the symmetric difference between two sets A and B, and $A \cup B$ denote the union of these sets. The mobility factor is then calculated as the percentage of neighbours which remains the same between the sending of two consecutive Hello Packets:

$$MF = \begin{cases} \sqrt{1 - \frac{|(N_x \cap \overline{N_x^p}) \cup (\overline{N_x} \cap N_x^p)|}{|N_x \cup N_x^p|}}, & \text{if} \quad N_x \cup N_x^p \neq \varnothing \\ 0, & \text{otherwise} \end{cases} \tag{11}$$

where N_x is the current neighbor set of node x and N_x^p denotes the neighbor set of node x at the time of the previous hello was sent. Every node needs to maintain a N_x^p. When a hello timer expires a node uses this value and current neighbor set to calculate the MF. The MF can give a higher value to a relatively stable node. In case of a static network, MF will be 1 for every node. The notation $|\,.\,|$ in (11) denotes the cardinality of a set.

The MF metric helped a routing protocol choose the best stable path for transmitting data packets. As a result, the performance of proposed routing models improved: the minimum node lifetime is increased, leading to a fairer energy balancing, and that the application level throughput is increased. However, to calculate the MF value, each node is required to maintain a look-up table to record the historical neighbour list N_x^p besides the current neighbour list N_x. This could lead to the problem of resource usage and computational complexity at a node when the scale of MANET grows up.

In an effort to find an appropriate metric that well adapts to the mobility of MANETs, T. The Son et al. citeson introduced Path Encounter Rate (PER) metric based on the concept of “encounter” stated in [32]. Let E_n be the set of all encounters (see Eq. refref:9) experienced by node n within T (our observation period): $E_n = \{e_{nm}\}$. The Average Encounter Rate (AER_n) of node n is defined as the average number of new encounters per time unit T.

$$AER_n = \frac{|E_n|}{T}, \tag{12}$$

where $|E_n|$ is the cardinality of the set E_n or the number of elements of the set E_n. Conceptually, the AER indicates the relative speed of a node with respect to its neighbours in a given network density, whereas MF metric reflects the stable of the neighbour set [37]. The Path Encounter Rate (PER) of a path or a route across the network is defined as a sum of square Average Encounter Rate values of all nodes along that path or that route.

$$PER = \sum_{i=1}^{m} AER_i^2, \tag{13}$$

where m is the number of nodes along the path including source and destination nodes. The routing algorithm then selects the path which has the lowest PER value among the available paths to the destination. As a result, the performance of proposed routing models in [37] improved compared to routing metric based on hop count metric.

3 Mobility degree detection model

As argued earlier in Assumption and definition and by analyzing the Q-learning [16] behaviour and its direct application in [36], the Q-Routing algorithm, a successful routing policy in a rapid changing MANET should account for the stochastic dynamics associated with the node mobility. In this section, we describe our MDP formulation for the process in order to infer network mobility state information in a distributed manner.

To take into account the relative speed of a node with respect to its neighbours and the instability of the neighbor nodes, which directly affects the stability of the link, proper metrics are proposed in this section. The idea follows the one introduced in Wu et al. [19] and in Macone et al. [36], which define a discount factor $\gamma = \gamma(t)$. In our model, two so-called mobility-stability-varying discount factors are associated to different nodes. Then, we define a one step updating rule, by modifying the rule mentioned in Eq. (6) accordingly:

$$Q_i(d,j) \leftarrow (\alpha - 1) \cdot Q_i(d,j) + \alpha \cdot [r_j + \gamma_{ij}(t) \max_{l \in N(j)} Q_j(d,l)], \tag{14}$$

where, $Q_i(d,j)$ is the value of taking action j (i.e., forwarding packet to node j) in state i in order to reach state d, and α, defined in Sect. 2.2 as a learning factor, indicates the weight of the recent knowledge with respect to older one. α takes values in the interval [0, 1]: the smaller the α is, the greater importance is given to the older knowledge; on the counterpart, the bigger the α value is, the greater importance is given to the upcoming knowledge.

Two different metrics are reported in this section, taking into account the link stability of a node and the node mobility; the metrics are then combined together to yield the mobility-stability-varying discount factor. A link stability factor γ_j^{STA} follows the idea proposed in [19, 36], and the mobility factor γ_i^{MOB} follows the idea proposed in

[32, 33]. Note that the mobility factor is per-node factor, whereas the link stability factor is per-link factor.

The overall discount factor from state s_i and state s_j is computed as:

$$\gamma_{ij}(t) = \gamma^{MAX} \cdot \gamma_i^{MOB}(t) \cdot \gamma_j^{STA}(t) \tag{15}$$

where $\gamma^{MAX} \in [0, 1]$ is the maximum discount factor used in order to normalize AER_i, reached when all the factors are equal to 1, i.e., with best link stability and fixed nodes. The algorithm is guaranteed to tend towards convergence as mentioned in [16].

3.2 Link stability metric

As already mentioned, to take into consideration the stability of a link, the Mobility Factor (MF) metric [19, 36] is considered. As mentioned in Sect. 2.4, this metric takes into consideration the mobility of nodes in terms of neighbour sets.

The link stability factor, following Eq. (11), is given by:

$$\gamma_j^{STA}(t) = \begin{cases} \sqrt{1 - \dfrac{\mid N_j(t) \Delta N_j(t - T_{HELLO}) \mid}{\mid N_j(t) \cup N_j(t - T_{HELLO}) \mid}}, & \text{if} \quad N_j(t) \cup N_j(t - T_{HELLO}) \neq \varnothing \\ 0, & \text{otherwise} \end{cases} \tag{17}$$

3.1 Q-routing

In this work, our model of the mobility detection is an extension of Q-Routing [17, 19]. The entire mobile ad hoc network is the environment. Its components include the mobile nodes, the links between the nodes and packets. Each packet $P(s, d)$, indexed by its source and destination node is an agent. In the Q-Routing algorithm, the network is modelled as a graph $G = (V, E)$, where V is the set of nodes with cardinality $\mid V \mid = N$ and E is the set of edges. Each node $i \in V$ is connected to a set of neighbour nodes N_i. In the following, we describe the other components in further detail:

1. $S = \{S_i = \delta_i \mid i \in V\}$, where δ_i is a vector of n elements equal to 0 but the *ith* element equal to 1, is the finite state set; each state indicates the (unique) current position of the packet;

2. $A = \cup_{i=1,\ldots,N} A(s_i)$ is the action set, where $A(s_i) = \{a_{ij} \mid j \in N_i\}$ is the action set of state s_i, and a_{ij} represents the decision of the agent to move from state s_i to state s_j (i.e., node i sends the packet to neighbour j).

3. the one-step reward obtained performing action $a(t) = a_{ij}$ in state $s(t) = s_i$ to reach state $s(t+1) = s_j$ is 1 only if at a time $t+1$ the packet is in the destination node d, it is 0 otherwise; the reward is the function of the reached state *sj* only:

$$r_j = \begin{cases} 1, & j = d \\ 0, & \text{otherwise} \end{cases} \tag{16}$$

The transition matrix is not defined, since RL algorithms implicitly estimate the transition probabilities within the action-value function estimation (2.2).

where $N_j(t)$ is the set of neighbours of node j at time t and T_{HELLO} is the hello interval. The mobility factor (17) is computed by each node and sent to its neighbours, thus, each node compute its discount factor (15). The main importance of the link stability metric is to let routing algorithm prefer more stable paths.

3.3 Link mobility metric

The proposed model considers also the node mobility metric. This metric, Average Encounter Rate (AER), uses the concept of "encounter" mentioned in Sect. 2.4 and defined in [32]. The mobility factor is calculated, following the Eq. (13), by:

$$\gamma_i^{MOB^*}(t) = \frac{\mid E_i \mid}{T}, \tag{18}$$

where $\mid E_i \mid$ is the cardinality of the E_i (the set of new encounters that node i experienced within observation time T.

In Eq. (18), we need to normalize $\gamma_i^{MOB^*}$ between 0 and 1. Then we introduce a factor $\gamma^{MAXI} \in [0, 1]$, which represents the maximum average encounter rate, reached when $\mid E_i \mid = N - 1$ and N indicate the number of nodes in the network. The mobility factor is given by:

$$\gamma_i^{MOB}(t) = \gamma^{MAXI} \cdot \gamma_i^{MOB^*}(t), \tag{19}$$

The mobility factor (19) is computed by each node and compute its discount factor (15). The main importance of the mobility factor is to indicates the relative speed of a node with respect to its neighbours in a given network density, since AER increases linearly with node density (see also analysis in [32]).

3.4 Impact factor of mobility

Watkins et al. [16] proved that Eq. (14) converges to optimal $Q_i^*(d,j)$,the optimal action-value function, independent of the policy being followed, and then an optimal policy π^* can be derived such as:

$$\pi_i^*(d) = \underset{j}{\operatorname{argmax}}\ Q_i^*(d,j), \tag{20}$$

Every node i maintains a Q-Table that consist of action-value $Q_i(d,j)$, where $d \in S_i$ and $j \in A_i$ and whose value is ranging from 0 to 1. The size of Q-Table for each node i is determined by the number of its neighbours. Each value in the Q-Table $Q_i(d,j)$ reflect an estimate of the value of future cost of mobility, called the *impact factor of mobility* and denoted $\boldsymbol{\alpha_{ij}}$, if in node i chooses a node j as a next-hop in order to reach destination d. These factors make nodes to learn the mobility degree changes around them.

4 AQ-Routing model

4.1 Q-mobility metrics

Operational experience with has revealed that use of hop-count as routing metric leads to unsatisfactory network performance. Experiments with *ETX* metric [38] proved that is very easy to implement and provides sufficiently good results.

The ETX metric of a link is the estimated number of transmissions required to successfully send a packet over that link, until an acknowledgement is received. The algorithm for computing *ETX* runs as follows. Node i computes the value of the *ETX* metric of its link to neighbour j by continuously estimating the loss rates over this link, in both directions: from j to i (this rate is called d_r^{-1}), and from i to j (this rate is called d_f^{-1}). Node i computes d_r^{-1} as the measured proportion of packets successfully arriving from j, and sends this value in HELLO message to node j. Symmetrically, node j computes the rate of packets successfully arriving from i, and sends its value to node i that can take as d_f^{-1} value for this link. The value of the ETX metric of the link is then:

$$ETX_j = d_r^{-1} \times d_f^{-1}, \tag{21}$$

which corresponds to the expected number of attempts to successfully receive and acknowledge a packet over this link.

It can be seen that $ETX \geq 1$ because of $d_f, d_r \leq 1$ for all links. The equality occurs as $d_f, d_r = 1$ or the forward and reverse packet delivery ratios over a link achieve 100%, therefor the selected path $P_{selected}$ for routing is chosen by:

$$P_{selected} = \underset{P_j}{\operatorname{argmin}} \left(\sum_{l \in P_j} ETX_l \right) \tag{22}$$

where P_j is the set of available path from the source to the destination and l is the link in such Pj.

ETX metric incorporates the notion of link quality metrics and with stationary nodes, it out-performs hop-count although it uses longer paths. In mobile condition, ETX metric leads to a bad routing performance [29] because the metric does not quickly enough react to the mobility of network.

To deal with the mobility, some routing models employed a metric named Mobility Factor (MF) (2.4). This metric is based on detecting the change of neighbor sets in a period of HELLO messages to examine the link stability before sending a packet. The mobility factor in Eq. (11) is calculated as the percentage of neighbours which remains the same between the sending of two consecutive Hello Packets. In our implementation, each node $i \in V$ keeps in memory a set of encounter tuple $\cup_{j \in N_i}\{MainAddress_j, Time_j\}$, where $Time_j$ is the time this tuple expires and must be removed. The update value of MF_i (11) occurs every T_{HELLO} emission interval, and requires to get $\mid N_i \mid$, the cardinality of set of neighbours of node i, from a set of neighbour tuple.

Equation 11 implies that the higher the MF value is, the more stable the path is. Thus, the mobility factor of nodes has an inverse relation with mobility degree. This means that, the lower the mobility factor reflect the higher degree of mobility factor. Hence, similarly to [39], we use the following logistic function to model the mobility factor calculated by node j:

$$\varphi(MF_j) = \frac{a}{1 - e^{\frac{-MF_j}{b}}} \tag{23}$$

where a is a scaling parameter and b is preferred coefficient; therefor the routing path in mobile condition in selected by:

$$P_{selected} = \underset{P_j}{\operatorname{argmax}} \left(\sum_{l \in P_j} \varphi(MF_l) \right) \tag{24}$$

4.2 AQ-Routing model

Clearly, routing under the most stable path by using MF metric (or other mobility metrics) might suffer a longer delay than routing via the shortest path. Hence, nodes might consider other factors, for example, link bandwidth [19], link availability and residual energy [36], overhead redundancy and route duration [40], besides the MF in order to improve routing performance. This work, however, focuses on the adaptation of nodes (as described in introduction) on different states of network mobility.

4.2.1 AQ-Routing algorithm

In order to achieve a trade-off between ETX and MF based on mobility degree factor learned in our model, we propose a new metric called $Q_{metric_{ij}}$ which accounts for mobility degree and takes advantages from both metrics described above. Overall, we use the following equation for describing $Q_{metric_{ij}}$ used in AQ-Routing:

$$Q_{metric_{ij}} = \alpha_{ij} \cdot \varphi(MF_j) + (1 - \alpha_{ij}) \cdot \lambda ETX_{ij}, \tag{25}$$

$Q_{metric_{ij}}$ is a weighted sum, where α_{ij} is a degree mobility factor determined according to our model described in section 3.4. Also λ is a normalization factor to make the values of ETX_{ij} between 0 and 1 and $j \in N(i)$, where $N(i)$ is a set of neighbour nodes of i.

The path Q_{metric} (PQ_{metric}) of a path P_j across the networks is defined as a sum of $\{Q_{metric_l} \mid l \in P_j\}$ of all links along to that path.

$$PQ_{metric_{P_j}} = \sum_{l \in P_j} Q_{metric_l}, \tag{26}$$

In principle, the primary goal of a routing protocol is to look for a path which has the lowest PQ_{metric} to forward data packets. The selected path will offer the best quality for transmitting the data packets because:

- it takes advantages of link stability and least mobility and density guaranteed by mobility degree factor α_{ij};
- it takes advantages of both link quality and link stability guaranteed by ETX and MF;
- it affect directly throughput when a lower degree of mobility occurs because ETX will be preferred;

Mathematically, the selected path P for routing must be satisfied:

$$P_{selected} = \underset{P_j}{\operatorname{argmin}} \left(PQ_{metric_{P_j}}\right), \tag{27}$$

where P_j is the set of available path from the source to the destination.

4.2.2 Proactive approach

In AQ-Routing, the mechanism for populating the local link information and the neighborhood information performed in a proactive fashion. Similar to OLSR, *Hello Packets* are exchanged among neighbours. To keep track of fast connectivity changes, a *Hello Packets* must be sent at least every hello interval T_{hello}. Each note i initialize the action-values (Q-table) for any destination node: $Q_i(d, j) = 0$, $d \in D$, $j \in N(i)$ and D is a set of destinations.

At each time step (T_{hello}), each node broadcasts to its neighbours a *Hello Packet* containing: (i) its maximum estimates of the action-values ($\max_{l \in N(i)} Q_i(d, l)$) towards each destination node; (ii) all the information needed to calculate the variable discount factor, as described in Sects. 3.2 and 3.3. *Hello Packet* must never be forwarded. Each node i processes incoming *Hello Packet* messages from neighbour node j and executes the following steps:

- get the mobility factor for neighbour and calculate the discount factor (see Eq. 15);
- if the sender of the *Hello Packet* is the destination node, then reward value of 1 is assigned, otherwise reward value is 0 (see Eq. 16);
- update $Q_i(d, j)$ (see Eq. 14);
- get the reverse loss rates from neighbour node j and calculate ETX_{ij} (see Eq. 21);
- calculate $Q_{metric_{ij}}$ (see Eq. 25)

4.2.3 Loop management

To Judge the performance of a routing protocol, loop-freedom is one of the aspects to consider in MANETs. The effect of transient routing loops is to significantly increase the impact on the surrounding network and its traffic thus degrading end-to-end transmission [41]. In a reactive protocols, such as AODV [9] and DSDV [42], loop freedom is guaranteed by using a destination sequence numbers. The destination sequence number is created by the destination to be included along with any route information it sends to requesting nodes. Proactive protocols, in contrast, like OLSR [8] and STAR [43] incur temporary loops, because several overlapping paths to a destination can coexist, since there is a neighboring nodes that have different views of the topology and hence, compute different forwarding tables. Moreover, a frequent topology changes might trigger advertisements by several routers and causes an inconsistent topology information that causes loops.

In AQ-Routing, we propose a post-detected-based method that guarantees loop-free routing. Therefore, detected loop must be removed from routing table and packet in the loop must be discarded. Our mechanism is based on Duplicate Packet Detection DPD introduced in [44], and duplicate packets are detected. These detected duplicate packet indicate packet looping. Then, the packet must be discarded and the loop recovery procedure begin by executing these following steps:

- a *Probe Message* is used to notify other nodes that the loop routing has occurred; the node sends this *Probe Message* ,which contains the main address of unreachable Destination Add_d and he main address of the node

n that detected loop Add_n, towards the same destination as a looping packet;
- when a node i receives the *Probe Message*, through the inspection of this packet, it resets to zero the associated action-values $Q_i(d, n)$ and sends as soon as a new *Hello Message*, to speed up the update of routing tables.
- when the *Probe Message* reaches the node i and $Add_i = Add_n$, the loop management procedure is terminated.

Table 2 OLSR and AQ-Routing parameters

Attribute	Value
Willingness	Default (3)
Hello interval (s)	2
Topology control interval (s)	5
Neighbor holding time (s)	6
Topology holding time (s)	15
Duplicate holding time (s)	30

5 Performance evaluation

5.1 Simulation model

In this section we evaluate the performance of our proposed routing algorithm by means of NS-3 (release 3.26), a discrete-event network simulator for Internet systems [45]. Each scenario was experimented 20 times in different seed numbers. Nodes have been warmed up 10 s to reach the stable state before transmitting the data. Encounter lifetime Δt is set to $6s$ ($3 \times T_{hello}$) which is equal to the observation time T (Table 3). Consequently, each encounter node will no longer be recognized as a new encounter in the next observation. We consider a 1000×1000 m network environment in which a total of 30 nodes are deployed.

We consider a 1000 $\times$1000 m network environment in which a total of 30 nodes are deployed. We made all nodes move continuously without pausing at any location during simulation of 160s. Table 1 define the mobility model, namely, the Random Waypoint mobility model [46], a random model for the movement of mobile nodes, and how their location, velocity and acceleration change over time. Table 2 define the routing protocols parameters (both for OLSR and for AQ-Routing) used in the simulations. The wireless link model used in all the simulations is the Wi-Fi 802.11b [47] at 54 Mbps. The transmission range TX is 100 m. A constant bit rate (CBR) flow is used as applications, and the packet size of the flow is 512 bytes. In this simulation the source and destination are chosen randomly, and source keeps transmitting packets to destination at a rate of 10 packets per second during simulation time.

In this situation, protocol performances are evaluated in terms of:

- *Mean packet loss ratio* The ratio of the average number of data packets lost by the destination node to the number of data packets transmitted by the source.
- *End-to-end delay* The elapsed time between a data packet is sent by the source to the time data packet is received at the destination node.

Table 1 Mobility model

Attribute	Value
x_{min} (m)	0
y_{min} (m)	0
x_{max} (m)	1000
y_{max} (m)	1000
Speed (m/s)	[1–15]
Pause time (s)	Constant (0)
Start time (s)	Constant (10)
Stop time (s)	End of simulation

5.2 Numerical results

To investigate the effect of mobility on the mobility metrics (MF and AER), nodes in the simulation firstly have been set to move in RWP mobility model and then change to be static under CP mobility model at the second of 10 (see Fig. 1) and then change back to be mobile under RWP mobility model at the second of 80 resulting in network state changing from mobile to static and back to mobile. Fig 1 shows the mobility metrics of a random picked-up node in the network, the mobility model of other nodes are much similar.

As shown in Fig. 2 the AER value and MF value varies depending on the velocity of its neighbors. Thus, we noticed that the values of AER and MF start to change once

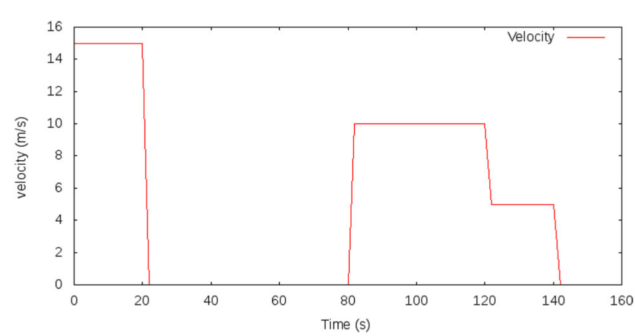


Fig. 1 Velocity against different mobility states

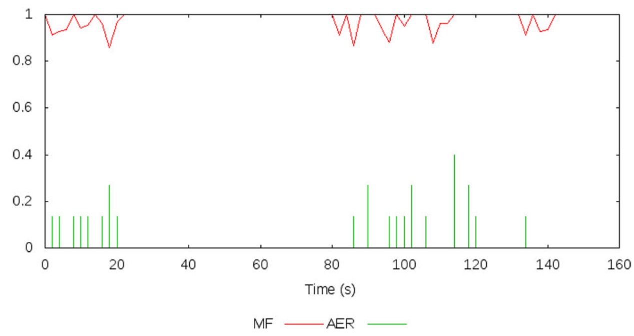


Fig. 2 AER and MF against different velocities

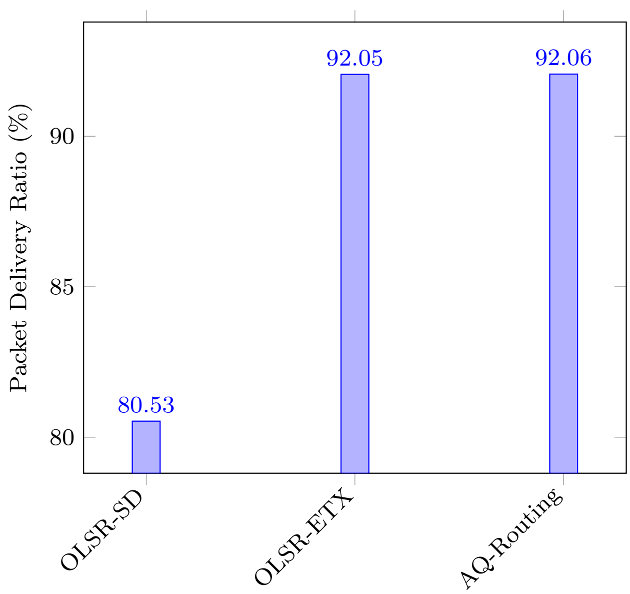


Fig. 3 Effect on packet delivery ratio at the static scenario

the speed of the node changes from null to 10 meter per second.

It is also recorded that the variation of the MF is opposite to that of the AER. This justifies our choice of the logistic function in the formula (23).

In Fig. 3, OLSR-SD performs worse than both OLSR-ETX and AQ-Routing. In static scenarios, AQ-Routing performs like OLSR-ETX, because they promote ETX metric and nodes find the highest throughput paths for routing [18]. As a result, PDR of the AQ-Routing at the static scenario is about 10% higher than the OLSR-SD. However, the routing performance of the AQ-Routing is the same as that of the OLSR-ETX.

According to the Fig. 4, we know that the routing overhead redundancy and the choice of shortest path rather than the highest throughput path affect the performance in terms of end-to-end delays. Thus, As shown in Fig. 4, we can see that OLSR-ETX suffers from relatively high end-to-end delay, as compared to OLSR-SD. This observation

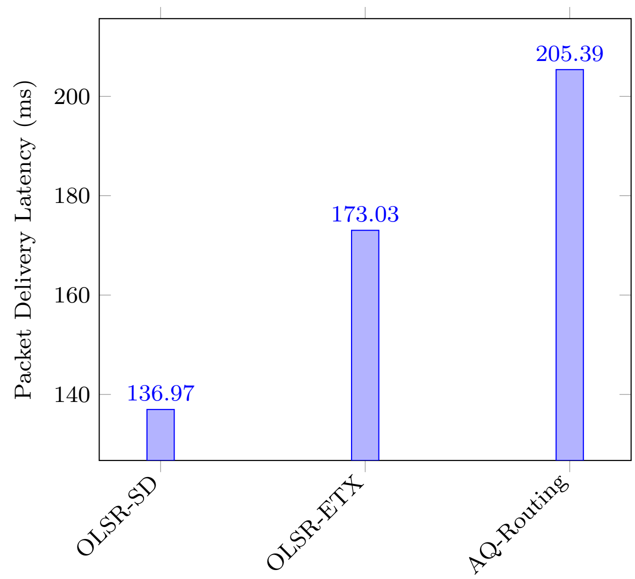


Fig. 4 Effect on packets delivery latency in static scenario

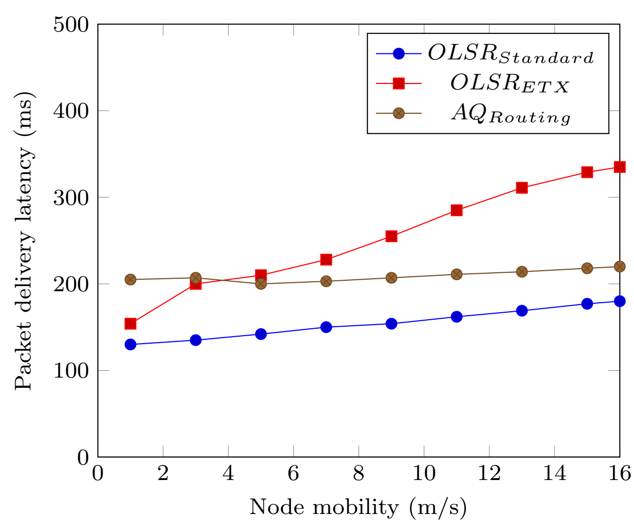


Fig. 5 Effect on the packet delivery latency

is in-line with results obtained by authors of ETX metric in [18].

Figures 5 and 6 illustrate the effect of the proposed algorithm in comparison with the effect of the previous algorithms (OLSR-SD and OLSR-ETX) in terms of packet delivery ratio and in terms of end-to-end delay. As shown in Fig. 6, we can see that OLSR-ETX algorithm performs worse than the other routing algorithms. In addition, we can observe that the proposed algorithm achieves high performance. As presented in Fig. 6, as the route duration affect directly the performance in terms of packet delivery ratio in mobility scenarios, because the probability of taking two node connected a long while decrease. It is observed that AQ-Routing protocol keeps its best

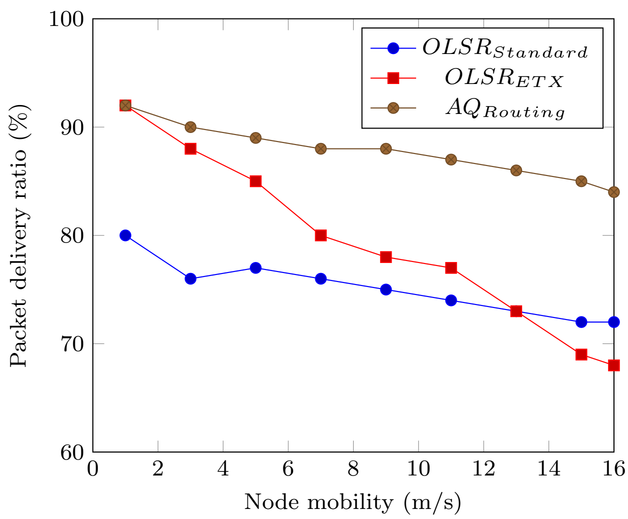


Fig. 6 Effect on the packet delivery ratio

performance in terms of PDR compared to OLSR-SD. It turns out that the metric has been changed to promote the use of MF metric and, thus, the most stable path has been chosen for routing instead of the shortest path. This is due to the fact that the AQ-Routing algorithm, thanks to the Q_{metric}, selects the more stable route with the best quality than the route selected by HOP metric. As Fig. 5 shows, End-to-End delay our proposal protocol gives better performance as speed increases. But remains less than OLSR-SD in terms of end-to-end delay. This is not a surprise since AQ-Routing shortened the packet waiting time in buffer by means of selecting the more stable route than that for standard OLSR-SD and thus reducing the route errors. Another reason which makes AQ-Routing performs good delay is that it can balance traffic loads between nodes thanks to the mobility detection model based on reinforcement learning technique.

6 Conclusion

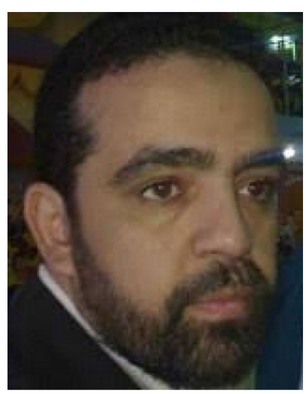
In this article, we have proposed a new routing protocol (AQ-Routing), based on Reinforcement Learning, suitable for Mobile Ad hoc Networks (MANETs), in particular with the aim to be aware of node mobility and link stability. To this purpose, a new mobility detection model was proposed to allow each node adjust a new metric called Q_{metric} based on the updated mobility factor. Thus, each node in the networks can adapt its routing behaviour based on the network conditions around it and AQ-Routing can efficiently improve the stability of links in both static and mobile scenario and, hence, increase the packet delivery ratio compared to existing routing models.

Moreover, the proposed approach suggests a number of interesting directions for further work. For instance, the proposed model does not depend on any specific routing protocols or metrics and allows introduction of further metrics. Thus future work will be introduce one or more energy-aware protocol in order to maximizing network lifetime and minimizing energy consumption routing in all MANET–IoT systems using a combination of MANET and WSN routing principles. We will grouping the nodes of the IoT network into clusters and each cluster is an Arrangement Graph $A_{n,k}$.

References

1. Alameri, I.A.: MANETS and Internet of Things: the development of a data routing algorithm. Eng. Technol. Appl. Sci. Res. **8**(1), 2604–2608 (2018)
2. Luigi, Atzori, Antonio, Iera, Giacomo, Morabito: The internet of things: a survey. Comput Netw. **54**(15), 2787–805 (2010)
3. Mukherjee, S., Biswas, G.P.: Networking for IoT and applications using existing communication technology. Egypt. Inf. J. **19**(2), 107–127 (2018)
4. Karlsson, J., Dooley, L.S., Pulkkis, G.: Secure routing for MANET connected Internet of Things systems. In: 2018 IEEE 6th International Conference on Future Internet of Things and Cloud (FiCloud), Barcelona, pp. 114–119 (2018)
5. Noorul, T.M., Pramila, R.S., Islam, N.: An analysis of routing protocols in MANETs and Internet of Things. In: 2017 International Conference on IoT and Application (ICIOT), Nagapattinam, pp. 1–8 (2017)
6. Airehrour, D., Gutierrez, J., Ray, S.K.: Secure routing for Internet of Things: a survey. J. Netw. Comput. Appl. **66**, 198–213 (2016)
7. Boukerche, A., Turgut, B., Aydin, N., Ahmad, M.Z., Boloni, L., Turgut, D.: Routing protocols in ad hoc networks: a survey. Comput. Netw. **55**(13), 3032–3080 (2011)
8. Clausen, T., Jacquet, P. (eds.): Optimized Link State Routing Protocol (OLSR). IETF RFC 3626, October 2003
9. Perkins, C., Belding-Royer, E., Das, S. (eds.): Ad hoc On-Demand Distance Vector (AODV) Routing. TETF RFC 3561, July 2003
10. Hoebeke, J., Moerman, I., Demeester, P.: Adaptive routing for mobile ad hoc networks. EURASIP J. Wireless Commun. Netw. **2012**(1), 126 (2012)
11. Giannoulis, S., et al.: A hybrid adaptive routing protocol for ad hoc wireless networks. In: Proceedings of 2004 IEEE International Workshop on Factory Communication Systems (2004)
12. Seba Lagraa, H.: ARPM: an adaptive routing protocol for MANETs. In: IEEE International Conference on Pervasive Services, Lyon, France (2006)
13. Fathy, C., El-Hadidi, M.T., El-Nasr, M.A.: Fuzzy-based adaptive cross layer routing protocol for delay sensitive applications in MANET. In: 2012 IEEE International Conference on Communications (ICC) (2012)
14. Ramachandran, B., Shanmugavel, S.: Mobility adaptive cross layer design for reliable route discovery in ad-hoc networks. In: Third International Conference on Wireless Communication and Sensor Networks, 2007 (WCSN '07) (2007)
15. Son, T.T. et al.: Self-adaptive proactive routing scheme for mobile ad-hoc networks. IET Netw. **4**(2), 128–136 (2015)

16. Watkins, C.J.C.H., Dayan, P.: Q-learning. Mach. Learn. **8**(3), 279–292 (1992)
17. Boyan, J.A., Littman, M.L.: Packet routing in dynamically changing networks: a reinforcement learning approach. In: Proceedings of the 6th International Conference on Neural Information Processing Systems (NIPS'93). Morgan Kaufmann, San Francisco, pp. 671–678 (1993)
18. De Couto, Douglas S.J., Aguayo, Daniel, Bicket, John, Morris, Robert: A high-throughput path metric for multi-hop wireless routing. Wireless Netw. **11**(4), 419–434 (2005)
19. Wu, C., Kumekawa, K., Kato, T.: A MANET protocol considering link stability and bandwidth efficiency. In: 2009 International Conference on Ultra Modern Telecommunications & Workshops, St. Petersburg, pp. 1–8 (2009)
20. Ayodele, T.O.: Introduction to machine learning. In: New Advances in Machine Learning, pp. 1–9. InTech, Rijeka (2010)
21. Forster, A.: Machine learning techniques applied to wireless adhoc networks: guide and survey. In: 3rd International Conference on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP), pp. 365–370. IEEE (2007)
22. Forster, A., Murphy, A.L.: Machine Learning Across the WSN Layers in Emerging Communications for Wireless Sensor Networks, pp. 165–181. InTech, Rijeka (2011)
23. Abu-Mostafa, Y.S., Magdon-Ismail, M., Lin, H.-T.: Learning from Data. AMLBook, Chicago (2012)
24. Chapelle, O., Scholkopf, B., Zien, A.: Semi-Supervised Learning, vol. 2. MIT, Cambridge (2006)
25. Puterman, Martin L.: Markov Decision Processes: Discrete Stochastic Dynamic Programming. Wiley, New York (2008)
26. Sutton, R.S., Barto, A.G.: Reinforcement Learning. MIT, Cambridge (1997)
27. Johnson, D., Hu, Y., Maltz, D.: The Dynamic Source Routing Protocol (DSR) for mobile ad hoc networks for IPv4. IETF REC (2007)
28. Sarkar, S.K., Basavaraju, T.G., Puttamadappa, C.: Ad Hoc Mobile Wireless Networks: Principles, Protocols and Applications, 1st edn. Auerbach Publications, Boston (2007)
29. Draves, R., Padhye, J., Zill, B.: Comparison of routing metrics for static multi-hop wireless networks. SIGCOMM Comput. Commun. Rev. **34**(4), 133–144 (2004)
30. Li, J., Blake, C., Couto, D.S.J.D., Lee, H.I., Morris, R.: Capacity of Ad Hoc wireless networks. In: 7th Annual International Conference on Mobile Computing and Networking (MobiCom '01), pp. 61–69. ACM, New York (2001)
31. Adya, A., Bahl, P., Padhye, J., Wolman, A., Zhou, L.: A multiradio unification protocol for IEEE 802.11 wireless networks. In: 1st International Conference on Broadband Networks (BROADNETS '04), pp. 344–354. IEEE Computer Society, Washington (2004)
32. Khelil, A., Marron, P.J., Rothermel, K.: Contact-based mobility metrics for delay-tolerant ad hoc networking. In: 13th IEEE International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems, pp. 435–444 (2005)
33. Ingelrest, F., Mitton, N., Simplot-Ryl, D.: A Turnover based adaptive HELLO protocol for mobile ad hoc and sensor networks. In: 2007 15th International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems, Istanbul, pp. 9–14 (2007)
34. Grossglauser, M., Vetterli, M.: Locating nodes with EASE: last encounter routing in ad hoc networks through mobility diffusion. In: IEEE 2003 22nd Annual Joint Conference of the IEEE Computer and Communications Societies (IEEE Cat. No. 03CH37428) (INFOCOM), vol. 3, pp. 1954–1964 (2003)
35. Boleng, J., Navidi, W., Camp, T.: Metrics to enable adaptive protocols for mobile ad hoc networks. In: Proceedings of International Conference on Wireless Networks (ICWN'02), pp. 293–296 (2002)
36. Macone, D., Oddi, G., Pietrabissa, A.: MQ-Routing: Mobility-, GPS- and energy-aware routing protocol in MANETs for disaster relief scenarios. Ad Hoc Netw. **11**(3), 861–878 (2013)
37. Son, Tran The, Le Minh, Hoa, Sexton, Graham, Aslam, Nauman: A novel encounter-based metric for mobile ad-hoc networks routing. Ad Hoc Netw. **14**, 2–14 (2014)
38. Richard, D., Jitendra, P., Brian, Z.: Routing in multi-radio, multihop wireless mesh networks. In: Proceedings of the MOBICOM Conference (2004)
39. Edalat, N., Tham, C.-K., Xiao, W.: An auction-based strategy for distributed task allocation in wireless sensor networks. Comput. Commun. **35**(8), 916–928 (2012)
40. Chun, L.D., Hyeon, L.D.: Novel routing protocol scheme for realtime applications in mobile ad hoc networks. Clust. Comput. **20**(77), 1–12 (2017)
41. Mase, K., Speakman, L., Akima, K., Owada, Y.: Loop Detection for MANET. Internet Engineering Task Force (2007). https://datatracker.ietf.org/doc/html/draft-mase-manet-loopdetect-00
42. Perkins, C., Bhagwat, P.: Highly Dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for mobile computers. In: ACM SIGCOMM'94 Conference on Communications Architectures, Protocols and Applications, pp. 234–244 (1994)
43. Garcia-Luna, J., Spohn, M.: Source Tree Adaptive Routing Internet Draft. draft-ietf-manet-star-00.txt, work in progress, October 1999
44. Macker, J. (ed.): Simplified Multicast Forwarding. RFC 6621, May 2012. https://doi.org/10.17487/RFC6621, http://www.rfc-editor.org/info/rfc6621
45. ns-3 Development Team, ns-3 Network Simulator (2014). https://www.nsnam.org/doxygen- release/group__mobility.html
46. Camp, T., Boleng, J., Davies, V.: A survey of mobility models for ad hoc network research. Wirel. Commun. Mob. Comput. **2**, 483–502 (2002). https://doi.org/10.1002/wcm.72
47. IEEE 802.11TM Wireless Local Area Networks: the Working Group for WLAN Standards. http://www.ieee802.org/11/. Accessed Oct 2011



Abdellatif Serhani was born in Taroudant, Morocco, on January 1, 1974. He received the diplôme d'ingénieur d'état degree in telecommunications from the Ecole Nationale d'Industrie Minéral (ENIM), Rabat, in 1999. He was with ISM. Since September 2015, he has been with the Communication Systems Department, INPT, where he is currently a Ph.D student of wireless communications. His research interests are in the area of IoT, MANETs, with special emphasis on Distributed Computing, Graph Theory and their application in wireless communications.

Najib Naja was born in Mohammedia, Morocco, on February 4, 1964. He received the diplôme d'ingénieur d'état degree from the Ecole Nationale Supérieure des Télécommunications de Bretagne (ENST Bretagne), Brest, France, in 1989, the Ph.D. degree in telecommunications from the Université de Rennes I, Rennes, France, in 1994, and the Doctorat d'État es Sciences from Mohammed V University, Rabat, Morocco, in 1997. Since 1994, he has been with the Signal and Communications Group, Institut National des Postes et Telecommunications (INPT), Rabat, where he is currently a Professor. His research interests are speech coding, CDMA, multiuser detection, MIMO communications, and networking.

Abdellah Jamali is a researsh and professor at Hassan 1 er University, Settat, Morroco, since October 2011. He was born in Ouarzazate, Morocco. He received a thesis in Computer Science from the University of Hassan II, Mohammedia, Morocco. He is a founding member, in 2007, of a research group e-NGN (e-Next Generation Networks) for Africa and Middle East. His research interests include the computer networks, IPv6 Networks, Quality of Service in MPLS and QoS in Ad Hoc and Networks performance analysis.