Cloud-Supported Seamless Internet Access in Intelligent Transportation Systems

Yen-Wen Lin · Jie-Min Shen · Hao-Chun Weng

Abstract Great progress of technologies makes intelligent transportation system (ITS) services increasingly desirable. It becomes essential to seamlessly connecting to the Internet for accessing high quality Internet-enabled ITS services. In this context, gateway is recognized as the logical portal to the Internet. The efficiency of gateway regulation hence significantly sways the quality of ITS services. To seamlessly access the Internet in ITSs, a cloud-supported gateway model is proposed in this paper. Accordingly, the weighty load of gateway government, including gateway registration, discovery, selection, dispatching, and handoff, is offloaded from the clients to the appointed cloud servers. Numerical analyses and simulation results suggest that the gateway model proposed in this paper effectively improves the system performance in terms of the packet delivery rate, end-to-end delay, signaling overhead, and gateway connectivity; and, correspondingly enhances the usage experience of Internet-enabled ITS services.

Keywords ITS · Cloud computing · VANET · Internet gateway · Connectivity

1 Introduction

ITSs [1] aim to intelligently support versatile transport services and traffic management. Advances of vehicular ad hoc networks (VANETs) technologies [2] make ITS services effectively accessible. VANET applications [2] widely support drive safety and infortainment for vehicular users. In VANETs [2], vehicle-to-vehicle (V2V) is the communications between vehicles while vehicle-to-infrastructure (V2I) is the communications between the vehicles and road side units (RSUs). RSU [2], in general, refers to roadside access point or base station that can be connected to the Internet. Thence, V2I communications are suitably used for accessing the Internet in VANETs. Some of ITS services require effectively access to the

Y.-W. Lin (✉) · J.-M. Shen · H.-C. Weng
Department of Computer Science, National Taichung University of Education,
Taichung, Taiwan, Republic of China
e-mail: ywlin@mail.ntcu.edu.tw

Internet [1,3–5]. *Gateway* here is regarded as the logical entrance to the Internet. Traditionally, Internet access in VANETs is generally provided through roadside gateways [6,7]. For high mobility and changing topology of the VANETs [2], mobile gateways [6–11] are thus introduced to get stable connections to the Internet for vehicular users.

Accordingly, the efficiency of gateway management in VANETs remarkably affects the usage experience of employing Internet-enabled ITS services. A cloud-supported gateway model, namely the gateway-as-a-service (GaaS), is proposed in this paper to get stable and seamless connection to the Internet in VANETs. Specialized cloud servers are designated for gateway government including gateway registration, discovery, selection, dispatching, and handoff. Thus, the heavy load of gateway management is shifted from the clients to the employed cloud servers. Additionally, when more than one gateway is obtainable around, the gateway with longest link lifetime is selected and reserved in advance by the cloud servers for the requesting client vehicle. Thence, a new link lifetime prediction scheme is also proposed in this paper. Related evaluations are carried out to investigate the effects of vehicle velocity, vehicle density, and gateway density on related performance including packet delivery rate, end-to-end delay, signaling overhead, and gateway connectivity ratio. As will be shown in the numerical analyses and simulation results, the proposed method outperforms a few well-known methods in context of VANETs.

The rest of this paper is organized as follows. Background and related work are introduced in Sect. 2. The proposed method is overviewed in Sect. 3. Numerical analyses and simulation results are presented in Sects. 4 and 5 respectively. Section 6 briefly concludes this paper.

2 Preliminary

ITSs [1] support the services of various modes of transport and traffic management. It makes the users to be better informed, safer, and smarter [1]. Typical applications of ITSs [1] include emergency vehicle notification, automatic road enforcement, variable speed limits, collision avoidance, and dynamic traffic light sequence. Some of these applications need Internet access through the Internet gateways or some roadside infrastructures [3–5].

Zhu et al. [3] survey the issues related to mobility and handoff management in VANETs. Among others, the proposals for V2V and V2I communications in VANETs are systematically addressed. Also, Ammari [12] offers a survey of research work integrating mobile ad hoc networks (MANETs) and the Internet. Specifically, the problems of Internet gateway discovery in MANETs are discussed in [3,12–14]. In service discovery protocols [3,13,14], the mobile nodes tag and register to related Internet gateway for Internet access. Relatively, some solutions [3,13,14] extend ad hoc routing protocols to support mobile IP in MANETs.

VANETs are widely accepted as a special type of MANETs. However, special characteristics of VANETs [2], including high mobility and changing topologies, the proposals for MANETs cannot satisfactorily gratify VANET environment. As mentioned in the study [7], several proposals stretch ad hoc routing to the Internet via gateways while a few proposals augment Mobile IP [15] for connecting vehicles to the Internet. Mobile IP and diverse extensions [7,15] aim at supporting Internet access for mobile nodes. Among them, MMIP6 [7,13] integrates IPv6-based VANETs with the Internet. A proactive agent discovery protocol is adopted in MMIP6. Also, in MMIP6, geocast is used to decrease the signaling overhead caused by flooding related control messages. Some methods [7,16–18] plant Mobile IP in the gateway to support mobility management and Internet connection in context of MANETs. The study [18] investigates the performance of using DYMO [19] in VANET environment. As suggested in [7,18], a few limitations, including deficient lifetime between the vehicles

and related gateway, and, inefficient handoff mechanism of connections from one gateway to another, are partially solved. Besides, the well-known project FleetNet [7,14] try to support Internet connection for VANET users via fixed roadside gateway. In FleetNet [7,14], based on a revised MIPv6, it proposed a location-based routing protocol and a gateway discovery protocol. However, as mentioned in the studies [7,14], the special mobility of vehicles was not fully taken into consideration to further optimize related performance of routing in VANET environment.

Plenty of efforts have been proposed to address routing problems in VANETs [2,20–22]. The signaling overhead of the topology-based approach (e.g. AODV [20,21,23]) increases significantly when the variation of network topology increases. Thus, the position-based approach (e.g. GPSR [20,21,24]) seems more proper for being adopted in VANETs [2,20,21]. Conventionally, Internet access in VANETs is provided through roadside gateways [6,7]. AODV+ [25], extended from well-known AODV [23], aims at connecting MANETs to the Internet. The performance of three gateway discovery protocols, including proactive, reactive, and hybrid gateway discovery, is investigated in the study [25].

As the special features of VANETs are considered, it is essential to discover a stable route to make VANET applications accessible. Thence, it is important to find a gateway with longer connection time to supply steady Internet access. The route lifetime deeply affect the routing performance in VANETs. The widely-known study in [26] introduces a link lifetime prediction scheme for routing in VANETs. Aside from roadside stationary gateways [6,7], mobile gateways [6–11] thus are introduced to alternatively support connections to the Internet for VANET users.

Due to high mobility and changing topology of the VANETs [2], it becomes challenging to efficiently discover proper gateways to get stable connection to the Internet. Cloud computing is widely recognized as next generation computing paradigm [27–31]. As defined in [29], based on the Internet protocols, cloud computing characterizes a new model for supplement, consumption, and delivery of information technology (IT). A cloud-empowered gateway model, namely the GaaS, is proposed here to improve Internet connectivity in VANETs for Internet-enabled ITS services. Considered aspects of the proposed GaaS model include gateway registration, discovery, selection, dispatching, reservation, and seamless service handoff. Specially, the gateway with longer link lifetime is selected as the entrance to the Internet. To precisely estimate the link lifetime between the client vehicle and the gateway, a link lifetime prediction scheme is therefore offered in this paper.

3 System Overview

The proposed GaaS gateway model is overviewed here.

3.1 Problem Statement

It becomes essentially important to support seamless Internet access to pleasingly furnish diverse ITS services. A *gateway* here is able to directly connect to the Internet. Traditionally, a gateway is facilitated on fixed RSUs. Due to high speed of the vehicles, the vehicles may quickly move in and out of the communication rage of a gateway. Thus, it is difficult to seamlessly access the Internet via the stationary gateways. To remedy the limits of the stationary gateways [6,7], the mobile gateways [6–11] are undertaken to improve the efficiency of Internet access in VANETs. The effectiveness of gateway supervision therefore decisively settles the quality of services (QoS) of Internet-enabled ITS services. The powerful processing
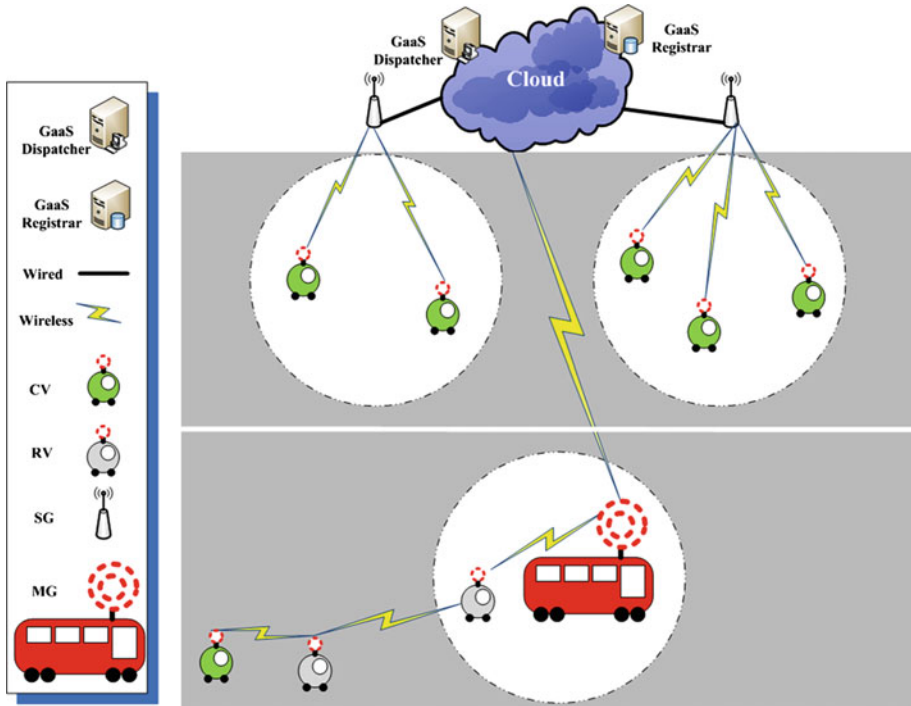
Fig. 1 System model

and storage capability of cloud computing are supposed to be able to intelligently help the VANET users with efficient gateways management. Therefore, a cloud-supported gateway model, namely the *GaaS* (*Gateway-as-a-Service*), is proposed in this paper to govern the gateways for VANET clients. The bulky load of gateway regulation is shifted from the clients to associated powerful cloud servers.

Besides, the special features of VANETs, including high mobility and changing network topology, profoundly affect the performance of VANET routing and consequently hinder the deployment of ITS services. Link lifetime [26] is one of the dominative factors in selecting the next-hop node on the route. Though plenty proposals [2,26] have been proposed to compute the link lifetime between two nodes, the utility is insufficiently limited. Specifically, in the GaaS, to dispatch related gateways in advance for supporting seamless Internet access, the time point of entering and exiting the coverage of a gateway is necessary to precisely predict the link lifetime between the vehicles and the gateways. Therefore, a link lifetime prediction scheme is newly proposed in this paper for selecting proper next-hop gateways for ITS clients.

3.2 System Model

The system model of the GaaS is depicted in Fig. 1. The main functional modules involved in the GaaS include:

- **Gateway**: A *Gateway* is the logical entity which is able to connect to the Internet directly via WiFi/WiMAX networks or cellular networks. On the basis of the mobil-

ity of the gateways, two types of gateways are considered in the GaaS. A *Stationary Gateway* (SG) is part of the roadside infrastructure such as access points (APs) of WiFi/WiMAX networks, or base stations (BSs) of cellular networks. Oppositely, a *Mobile Gateway* (MG) is the vehicle on the road which is capable of directly connecting to the Internet.
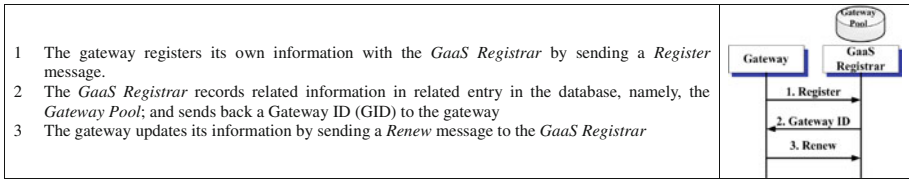
– **Client Vehicle**: A *Client Vehicle* (CV) is the vehicle which wants to access the Internet and request the GaaS services.
– **Relay Vehicle**: In case that a CV does not locate within the coverage of any gateway, to connect to a gateway, one or more *Relay Vehicles* (RVs) will be employed for forwarding packets from/to a CV to/from a gateway.
– **Cloud Server**: A *Cloud Server* is the service provider in the cloud. In general, to provide services in the cloud, a server needs to register its services to some well-known *name servers* in the cloud. In the GaaS, to supply associated services to the VANET clients, two special servers, namely, the *GaaS Registrar* and the *GaaS Dispatcher*, are facilitated in the system. A gateway willing to provide gateway services registers itself with the *GaaS Registrar*. Correspondingly, the *GaaS Registrar* maintains related information of the gateways. And, according to the information maintained by the *GaaS Registrar*, the *GaaS Dispatcher* is responsible of dispatching related gateways for the client vehicles.
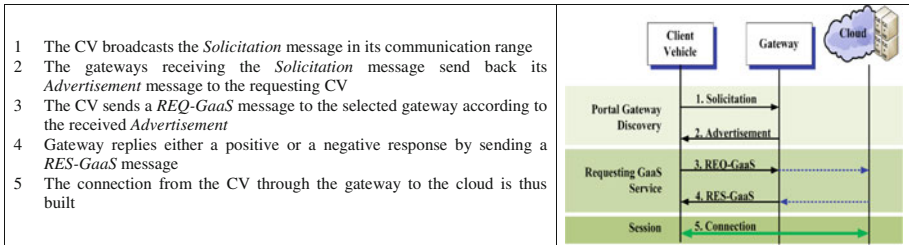
3.3 Offloaded GaaS Services

In the GaaS, the powerful processing and storage capability of cloud computing are employed for governing the gateways for the clients. After requesting the GaaS service, the load of gateway management is offloaded from the clients to associated cloud servers. The messages used in the following discussions are defined in Table 1.

Table 1 Used messages

Message	Definition
Advertisement	The gateway willing to supply the GaaS service sends the *Advertisement* message to the requesting CV
Candidate List	The list of the qualified gateways
GID	Gateway ID
Query	The *GaaS Dispatcher* sends a *Query* message to the *GaaS Registrar* for retrieving related information of the candidate gateways
Register	The gateway registers its services by sending a *Register* message to the *GaaS Registrar*
Renew	The gateway periodically updates its current status by sending a *Renew* message to the *GaaS Registrar*
REQ-GaaS	Requesting the GaaS service
RES-GaaS	Responding to the REQ-GaaS request
Reserve	The *GaaS Dispatcher* requests the selected gateway to reserve needed resources for the CV by sending a *Reserve* message
Solicitation	The client vehicle broadcasts a *Solicitation* message in its communication range to discover the gateways supplying the GaaS service

1　The gateway registers its own information with the *GaaS Registrar* by sending a *Register* message. 2　The *GaaS Registrar* records related information in related entry in the database, namely, the *Gateway Pool*; and sends back a Gateway ID (GID) to the gateway 3　The gateway updates its information by sending a *Renew* message to the *GaaS Registrar*	

Fig. 2 Gateway registration

1　The CV broadcasts the *Solicitation* message in its communication range 2　The gateways receiving the *Solicitation* message send back its *Advertisement* message to the requesting CV 3　The CV sends a *REQ-GaaS* message to the selected gateway according to the received *Advertisement* 4　Gateway replies either a positive or a negative response by sending a *RES-GaaS* message 5　The connection from the CV through the gateway to the cloud is thus built	

Fig. 3 Portal gateway discovery

3.3.1 Gateway Registration

As depicted in Fig. 2, a gateway willing to supply the gateway service registers its information with the *GaaS Registrar* in advance. Note that, related information of a gateway includes its current location, velocity, moving direction, and QoS level (i.e. available resources). The *GaaS Registrar* then maintains related information in the dedicated database (namely the *Gateway Pool*) and assigns a GID (Gateway ID) to the gateway. After that, the gateway renews its information either periodically or on demand.
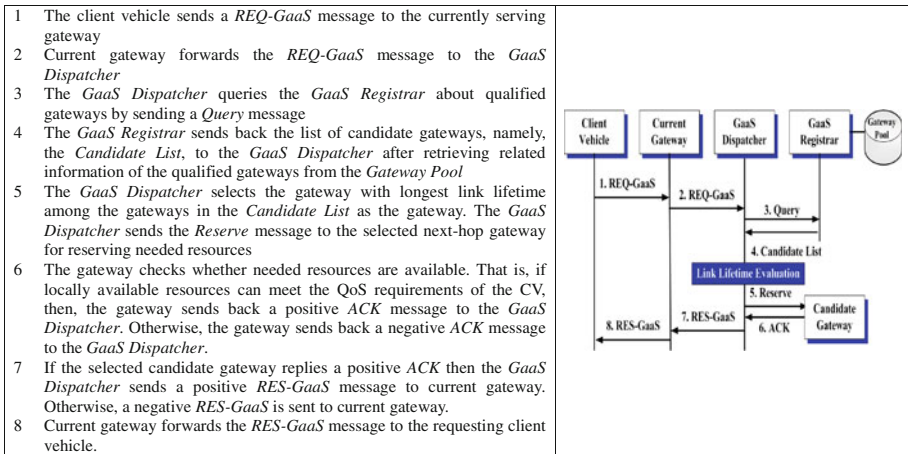
3.3.2 Portal Gateway Discovery

To offload the heavy load of gateway management from the clients to related cloud servers, a portal gateway is needed to serve as the initial entrance to the Internet for subsequent process. In the GaaS, as shown in Fig. 3, the CV requests to access the Internet by broadcasting a *Solicitation* message in its communication range. The gateways receiving the *Solicitation* message responses by sending back an *Advertisement* message to the requesting CV. The CV then sends a *REQ-GaaS* message to the selected gateway.

3.3.3 Requesting GaaS Services

As depicted in Fig. 4, a CV offloads the overheads of gateway management to the cloud servers by sending a *REQ-GaaS* message to currently serving gateway. Afterwards, the *GaaS Dispatcher* takes over the gateway discovery and dispatching for the requesting CV.

3.3.4 Seamless Service Handoff

After receiving the *REQ-GaaS* message from the requesting CV, the *GaaS Dispatcher* engages related gateways for the requesting CV. The CV periodically reports its own current status, including the location, speed, direction, and QoS level, to the *GaaS Dispatcher* via the

1	The client vehicle sends a *REQ-GaaS* message to the currently serving gateway
2	Current gateway forwards the *REQ-GaaS* message to the *GaaS Dispatcher*
3	The *GaaS Dispatcher* queries the *GaaS Registrar* about qualified gateways by sending a *Query* message
4	The *GaaS Registrar* sends back the list of candidate gateways, namely, the *Candidate List*, to the *GaaS Dispatcher* after retrieving related information of the qualified gateways from the *Gateway Pool*
5	The *GaaS Dispatcher* selects the gateway with longest link lifetime among the gateways in the *Candidate List* as the gateway. The *GaaS Dispatcher* sends the *Reserve* message to the selected next-hop gateway for reserving needed resources
6	The gateway checks whether needed resources are available. That is, if locally available resources can meet the QoS requirements of the CV, then, the gateway sends back a positive *ACK* message to the *GaaS Dispatcher*. Otherwise, the gateway sends back a negative *ACK* message to the *GaaS Dispatcher*.
7	If the selected candidate gateway replies a positive *ACK* then the *GaaS Dispatcher* sends a positive *RES-GaaS* message to current gateway. Otherwise, a negative *RES-GaaS* is sent to current gateway.
8	Current gateway forwards the *RES-GaaS* message to the requesting client vehicle.

Fig. 4 Requesting the offload GaaS service

currently serving gateway. Properly, the *GaaS Dispatcher* chooses and reserves the best one from among the gateways listed in the *Candidate List* ahead. For now, in the GaaS, the gateway with longest link lifetime will be chosen as the next-hop gateway for supplying the gateway service. (The details of gateway selection in the GaaS will be described later.) Thus, the load of gateway handoff is shifted to the *GaaS Dispatcher* and seamless service is accomplished.

3.4 Gateway Selection

In the GaaS, if more than one candidate gateway is available for the CV, the *GaaS Dispatcher* selects the best gateway for serving the door to the Internet. Related criteria for selecting the gateway is the QoS level supported by the gateways currently recorded in the *Candidate List*. Generally, the QoS level considered includes packet delivery rate, end-to-end delay, available bandwidth, and predicted link lifetime etc. Currently, when taking into account the special features of VANETs, in the GaaS, the gateway offering longest link lifetime with the client vehicle is chosen as the next-hop gateway for supplying the gateway service. Though a few methods [2,26] have been proposed to predict the link lifetime between two nodes, they cannot sufficiently meet the requisites of the GaaS. Thus, a new link lifetime prediction scheme is proposed here. In addition to compute the link lifetime, the newly proposed scheme is able to foretell the time point of entering and exiting the coverage of a gateway; that is necessary to support seamless GaaS service.

3.4.1 Link Lifetime Prediction

In the GaaS, three functions for predicting the link lifetime of a gateway and associated time points of entering/exiting the coverage of a gateway are introduced next.

Definition: Mobility Function f_1. Given the velocity vector of a node $V_c = (\alpha, \beta)$ and current location of a node (X_c, Y_c), f_1 computes new location of the node (X, Y) after time duration t; which is expressed in (1).

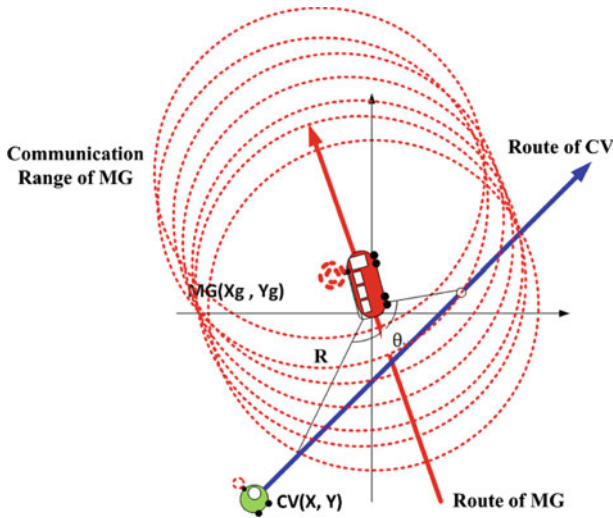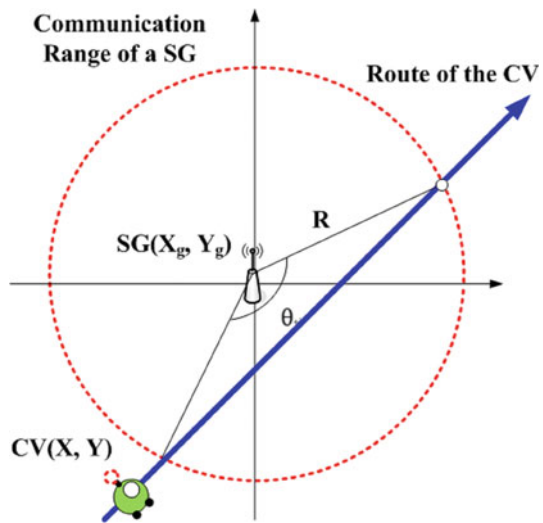$$X = X_c + \alpha * t$$
$$Y = Y_c + \beta * t \tag{1}$$

Fig. 5 Communication range of a MG

Fig. 6 Communication range of a SG

Definition: Connectivity Function f_2. Given the coordinate of the node and the gateway is (X, Y) and (X_g, Y_g) respectively, and, as shown in Fig. 5, the communication rage and the velocity vector of a gateway is R and $V_g = (\gamma, \delta)$ respectively, f_2 determines the boundary of the coverage of the gateway; which is expressed in (2).

$$\left(X - \left(X_g + \gamma * t\right)\right)^2 + \left(Y - \left(Y_g + \delta * t\right)\right)^2 = R^2 \tag{2}$$

Note that, if the gateway is stationary, as shown in Fig. 6, its velocity vector is (0, 0), thus, the boundary of the coverage of the gateway is expressed in (3).

$$\left(X - X_g\right)^2 + \left(Y - Y_g\right)^2 = R^2 \tag{3}$$
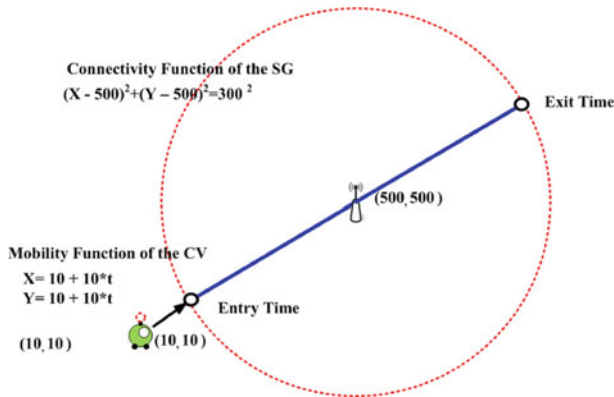
Fig. 7 Connecting to a SG

Definition: Link Lifetime Function f_3. Given current location of the node and the gateway (i.e. (X_c, Y_c) and (X_g, Y_g)), velocity vector of the node and the gateway [i.e. $V_c = (\alpha, \beta)$ and $V_g = (\gamma, \delta)$], f_3 computes the link lifetime between the vehicle and the gateway; which is expressed in (4).

$$\begin{cases} \left(X - \left(X_g + \gamma * t\right)\right)^2 + \left(Y - \left(Y_g + \delta * t\right)\right)^2 = R^2 \\ X = X_c + \alpha * t \\ Y = Y_c + \beta * t \end{cases} \quad (4)$$

Solve the expression (4), two solutions of t are gotten, denoted as t_1 and t_2. Among them, the smaller one is taken as the *Entry Time* and the larger one is taken as the *Exit Time* of the vehicle. Obviously, $t_{entry} = \min(t_1, t_2)$ and $t_{exit} = \max(t_1, t_2)$. Three cases are addressed further. Firstly, if both t_{entry} and t_{exit} are positive, then link lifetime is $t_{exit} - t_{entry}$. Secondly, if t_{exit} is positive and t_{entry} is negative, then link lifetime is t_{exit}. Thirdly, if no solution is gotten, it means that the vehicle will never enter the coverage of the gateway.

3.4.2 Examples

An example is described to illustrate the operations of three functions mentioned above. Assume that current location of CV, SG, and MG is (10, 10), (500, 500), and (300, 300) respectively. The communication range of the SG and MG is 300m and 250m respectively. The velocity vector of CV and MG is (10, 10) and (8, 0) respectively. As depicted in Fig. 7, the link lifetime between the CV and the SG can be computed by expression (5). Two solutions of t is 27.78 and 70.2 respectively. The former one is taken as entry time and the latter one is taken as exit time. Thence, the link lifetime between the CV and the SG is $70.2 - 27.78 = 42.42$.

$$\begin{cases} (X - 500)^2 + (Y - 500)^2 = 300^2 \\ X = 10 + 10 * t \\ Y = 10 + 10 * t \end{cases} \quad (5)$$

Similarly, in Fig. 8, the link lifetime between the CV and the MG can be computed by expression (6). Two solutions of t is 23.3 and 43.63 respectively. The former one is taken as
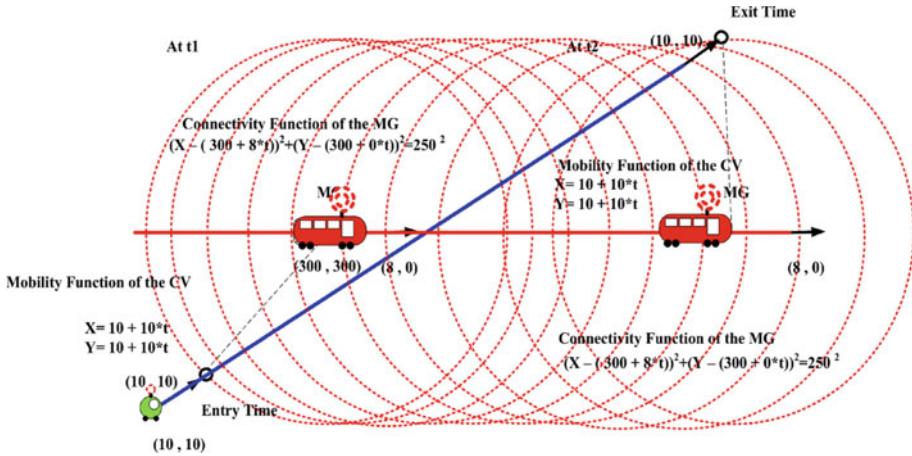
Fig. 8 Connecting to a MG

Table 2 Protocols comparisons

	Gateway discovery	Routing type	Gateway selection	Gateway reservation
AODV	Reactive	Topology based	Hop count	No
AODV+	Reactive, proactive, hybrid	Topology based	Hop count	No
GPSR	Reactive	Position based	Distance to destination	No
GaaS	Reactive	Position based	Link lifetime	Yes

entry time and the latter one is taken as exit time. Thence, the link lifetime between the CV and the MG is $43.63 - 23.30 = 20.33$.

$$\begin{cases} (X - (300 + 8*t))^2 + (Y - (300 + 0*t))^2 = 250^2 \\ X = 10 + 10*t \\ Y = 10 + 10*t \end{cases} \quad (6)$$

3.5 Discussions

Comparisons between the GaaS and three well-known routing protocols, including the AODV [23], the GPSR [24], and the AODV+ [25], are briefly summarized in Table 2. All these four protocols support discover gateways on demand. Specially, reactive, proactive, and hybrid gateway discovery are also supported in ADOV+ [25]. Both AODV [23] and AODV+ [25] belong to topology-based routing while GPSR [24] and GaaS are categorized as position-based routing. Besides, both AODV [23] and AODV+ [25] select the gateway with least hop count. Relatively, the gateway with shortest distance to the destination node is picked in GPSR [24]. In contrast, the gateway with longest link lifetime is chosen in the GaaS. Also, assisted by the employed cloud servers, beforehand gateway reservation is carried out in the GaaS to support seamless gateway service.

Table 3 Parameter definition

Notation	Definition
G_i	The gateway whose ID is i, i = {0, 1, 2, ..., n − 1, n}
S_0	State of the initial portal gateway G_0
$S_{i,j}$	State of the gateway G_i, j = 1, 2, or 3. Specifically, $S_{i,1}$ is the state of the CV can directly connect to the gateway G_i. $S_{i,2}$ is the state of the CV can connect to the gateway G_i via a few RVs. $S_{i,3}$ is the state of the CV cannot connect to the gateway G_i
α	The average time the CV cannot directly connect to a gateway
λ	The average rate the connection between the CV and the serving gateway breaks, where $\lambda = 1/\alpha$
β	The average time the CV cannot connect to a gateway via a few RVs
γ	The average rate the connection between the CV and the relayed gateway breaks, where $\gamma = 1/\beta$
k	The number of gateways to which the CV can directly connected before a new gateway discovery is issued

4 Numerical Analyses

4.1 Analysis Model

To evaluate the performance of the GaaS, related numerical analyses are presented here. The definitions of related parameters used in the following analyses are collected in Table 3. Related descriptions of nine possible state transitions are listed in Table 4. Define $S_{i,x} \rightarrow S_{i+1,y}$ as the state transition from $S_{i,x}$ to $S_{i+1,y}$ when the CV moves from G_i to G_{i+1}. Also, the Markov chain model adopted in the following analyses is illustrated in Fig. 9.

4.2 Performance Analyses

Without losing the generalities, the lifetime of a link is mutually independent. Assume that the average time the CV cannot directly connect to a gateway is α. Therefore, the breaking rate of the connection is $\frac{1}{\alpha}$. That is, the connection breaking rate (λ) is computed as expression (7).

$$\lambda = \frac{1}{\alpha} \tag{7}$$

In this paper, the *exponential distribution* is used to modeling the connection breaking rate. Thus, the *probability density function* (*pdf*) of the connection breaking rate is given by expression (8).

$$f(t) = \lambda e^{-\lambda t} \tag{8}$$

Table 4 State transition matrix of $S_{i,x} \rightarrow S_{i+1,y}$

	$y = 1$	$y = 2$	$y = 3$
$x = 1$	The CV currently can directly connect to G_i and it can directly connect to G_{i+1} later. The transition probability is $e^{-\lambda t}$	The CV currently can directly connect to G_i and it can connect to G_{i+1} via a few RVs later. The transition probability is $e^{-\gamma t}$	The CV currently can directly connect to G_i and it cannot connect to G_{i+1} (neither directly nor via a few RVs) later. The transition probability is $1 - e^{-\lambda t} - e^{-\gamma t}$
$x = 2$	The CV currently can connect to G_i via a few RVs and it can directly connect to G_{i+1} later. The transition probability is $e^{-\lambda t}$	The CV currently can connect to G_i via a few RVs and it can connect to G_{i+1} via a few RVs later. The transition probability is $e^{-\gamma t}$	The CV currently can connect to G_i via a few RVs and it cannot connect to G_{i+1} (neither directly nor via a few RVs) later. The transition probability is $1 - e^{-\lambda t} - e^{-\gamma t}$
$x = 3$	The CV currently cannot connect to G_i (neither directly nor via a few RVs) and it can directly connect to G_{i+1} later. The transition probability is $e^{-\lambda t}$	The CV currently cannot connect to G_i (neither directly nor via a few RVs) and it can connect to G_{i+1} via a few RVs later. The transition probability is $e^{-\gamma t}$	The CV currently cannot connect to G_i (neither directly nor via a few RVs) and it cannot connect to G_{i+1} (neither directly nor via a few RVs) later. The transition probability is $1 - e^{-\lambda t} - e^{-\gamma t}$
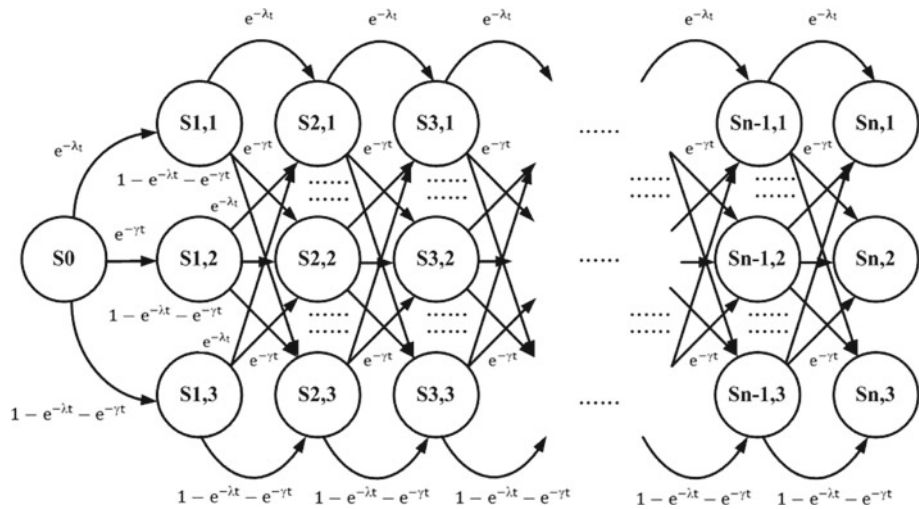
Fig. 9 Markov chain model

Thus, its *cumulative distribution function* (*cdf*) of the connection breaking rate is calculated as expression (9).

$$F(t) = 1 - e^{-\lambda t} \tag{9}$$

Therefore, the *cdf* of a CV can directly connect to a gateway is computed as expression (10).

$$F(t) = e^{-\lambda t} \tag{10}$$

In case, the CV can connect to a gateway via a few RVs. Assume that the average time the CV cannot connect to a gateway via a few RVs is β. Hence, the breaking rate of the relayed connection is $\frac{1}{\beta}$. That is, the relayed connection breaking rate (γ) is computed as expression (11).

$$\gamma = \frac{1}{\beta} \tag{11}$$

Again, the *exponential distribution* is used to modeling the relayed connection breaking rate. Thus, the *pdf* is given by expression (12).

$$f(t) = \gamma e^{-\gamma t} \tag{12}$$

Thus, the *cdf* of the relayed connection breaking rate is calculated as expression (13).

$$F(t) = 1 - e^{-\gamma t} \tag{13}$$

For that reason, the *cdf* of a CV can connect to a gateway via a few relays is computed as expression (14).

$$F(t) = e^{-\gamma t} \tag{14}$$

Consequently, the *cdf* of the CV cannot get gateway service neither directly connected to a gateway or connected to a gateway via a few RVs is given by expression (15)

$$F(t) = 1 - e^{-\lambda t} - e^{-\gamma t} \tag{15}$$

4.2.1 Performance of Direct Routes

In case that, initially, the CV can directly connect to a gateway. In our system, the *GaaS Dispatcher* reserves k-hop-ahead gateways for the CV. Then, the CV can directly connect to the next hop gateway, or the CV can connect to the next hop gateway via a relayed route, or the CV cannot connect to any gateway. In our design, the CV needs discovering gateways only when either it connects to the next hop gateway via a relayed route or the CV cannot connect to any gateway. The *cdf* of the scenario ($S_0 \rightarrow S_{n,2}$) and the scenario ($S_0 \rightarrow S_{n,3}$), are given by expressions (16) and (17) respectively. The resulting *cdf* of gateway discovery is computed as expression (18).
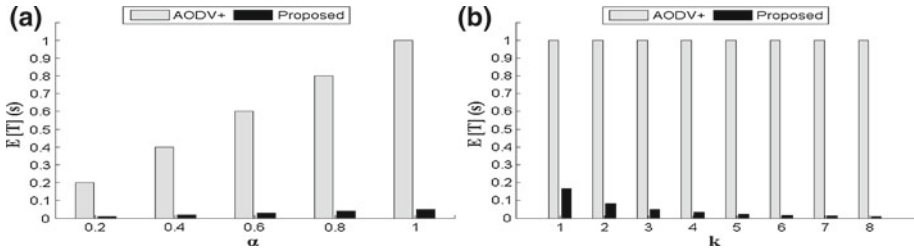
$$F(t) = (e^{-\lambda t}) \times (e^{-\lambda t})^k \times (e^{-\gamma t}) \tag{16}$$

$$F(t) = (e^{-\lambda t}) \times (e^{-\lambda t})^k \times (1 - e^{-\lambda t} - e^{-\gamma t}) \tag{17}$$

$$\begin{aligned} F(t) &= (e^{-\lambda t}) \times (e^{-\lambda t})^k \times [(e^{-\gamma t}) + (1 - e^{-\lambda t} - e^{-\gamma t})] \\ &= [(e^{-\lambda t})^{k+1} - (e^{-\lambda t})^{k+2}] \end{aligned} \tag{18}$$

Thus, in our design, the expected time spent for discovering gateway (E[T]) can be calculated as expression (19).

$$\begin{aligned} E[T] &= \int_0^\infty F(t)dt \\ &= \int_0^\infty \{(e^{-\lambda t}) \times [(e^{-\lambda t})^k - (e^{-\lambda t})^{k+1}]\}dt \end{aligned}$$

Fig. 10 **a** E[T] with various α. **b** E[T] with various k

$$= \int_0^\infty ((e^{-(k+1)\lambda t}) - (e^{-(k+2)\lambda t}))dt \qquad (19)$$

Comparatively, in AODV+, no gateway reservation mechanism is facilitated, the CV needs discovering gateways for each time it moves out of the coverage of currently serving gateway. That is, the CV have to discover the next-hop gateway when it connects to the next hop gateway directly, or when it connects to the next hop gateway via a relayed route, or when it cannot connect to any gateway. The *cdf* of the scenario ($S_i \rightarrow S_{i+1,1}$), the scenario ($S_i \rightarrow S_{i+1,2}$), and the scenario ($S_i \rightarrow S_{i+1,3}$), are given by expressions (20), (21), and (22) respectively. The resulting *cdf* of gateway discovery is computed as expression (23).

$$F(t) = (e^{-\lambda t}) \times (e^{-\lambda t}) \qquad (20)$$

$$F(t) = (e^{-\lambda t}) \times (e^{-\gamma t}) \qquad (21)$$

$$F(t) = (e^{-\lambda t}) \times (1 - e^{-\lambda t} - e^{-\gamma t}) \qquad (22)$$

$$F(t) = (e^{-\lambda t}) \times [(e^{-\lambda t}) + (e^{-\gamma t}) + (1 - e^{-\lambda t} - e^{-\gamma t})$$

$$= (e^{-\lambda t}) \qquad (23)$$

Thus, in AODV+, the expected time spent for discovering gateway (E[T]) is given by expression (24).

$$E[T] = \int_0^\infty F(t)dt$$

$$= \int_0^\infty e^{-\lambda t} dt$$

$$= \frac{1}{\lambda}$$

$$= \alpha \qquad (24)$$

Summary In case that the CV can directly connect to a gateway. As shown in Fig. 10a, both in the AODV+ and the proposed method, E[T] increases when α (i.e. the average time the CV cannot directly connect to a gateway) increases. Also, as shown in Fig. 10b, E[T] of AODV+ keeps constant as *k* (i.e. the number of gateways to which the CV can directly connected before a new gateway discovery is issued) increases. Oppositely, E[T] of the proposed method decreases as *k* increases. Basically, the expected time spent for discovering gateway (E[T]) of the AODV+ is longer than that of the proposed method.

4.2.2 Performance of Relayed Routes

In case that, initially, the CV can connect to a gateway via a relayed route. In our system, the *GaaS Dispatcher* reserves k-hop-ahead gateways for the CV. Then, the CV can directly connect to the next hop gateway, or the CV can connect to the next hop gateway via a relayed route, or the CV cannot connect to any gateway. In our design, the CV needs discovering gateways only when either it connects to the next hop gateway via a relayed route or the CV cannot connect to any gateway. The *cdf* of the scenario ($S_0 \rightarrow S_{n,2}$) and the scenario ($S_0 \rightarrow S_{n,3}$), are given by expressions (25) and (26) respectively. The resulting *cdf* of gateway discovery is computed as (27).

$$F(t) = (e^{-\gamma t}) \times (e^{-\lambda t})^k \times (e^{-\gamma t}) \tag{25}$$

$$F(t) = (e^{-\gamma t}) \times (e^{-\lambda t})^k \times (1 - e^{-\lambda t} - e^{-\gamma t}) \tag{26}$$

$$F(t) = (e^{-\gamma t}) \times (e^{-\lambda t})^k \times [(e^{-\gamma t}) + (1 - e^{-\lambda t} - e^{-\gamma t})]$$
$$= (e^{-\gamma t}) \times [(e^{-\lambda t})^k - (e^{-\lambda t})^{k+1}] \tag{27}$$

Thus, in our design, the expected time spent for discovering gateway (E[T]) can be computed as (28).

$$E[T] = \int_0^\infty F(t)dt$$
$$= \int_0^\infty \{(e^{-\gamma t}) \times [(e^{-\lambda t})^k - (e^{-\lambda t})^{k+1}]\}dt \tag{28}$$

Oppositely, in the AODV+, no gateway reservation mechanism is facilitated, the CV needs discovering gateways when it connects to the next hop gateway directly, or it connects to the next hop gateway via a relayed route, or the CV cannot connect to any gateway. The *cdf* of the scenario ($S_i \rightarrow S_{i+1,1}$), the scenario ($S_i \rightarrow S_{i+1,2}$), and the scenario ($S_i \rightarrow S_{i+1,3}$), are given by expressions (29), (30), and (31) respectively. The resulting *cdf* of gateway discovery is computed as expression (32).

$$F(t) = (e^{-\gamma t}) \times (e^{-\lambda t}) \tag{29}$$

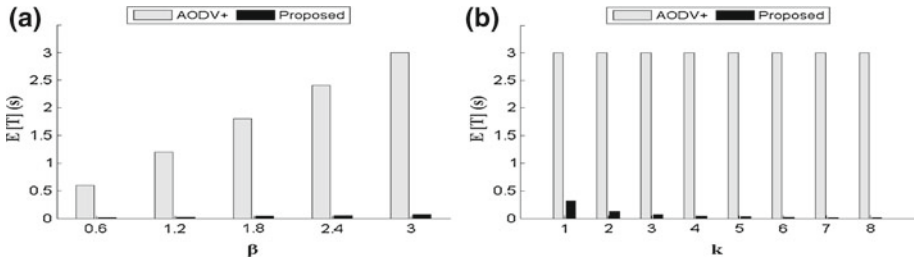$$F(t) = (e^{-\gamma t}) \times (e^{-\gamma t}) \tag{30}$$

$$F(t) = (e^{-\gamma t}) \times (1 - e^{-\lambda t} - e^{-\gamma t}) \tag{31}$$

$$F(t) = (e^{-\gamma t}) \times [(e^{-\lambda t}) + (e^{-\gamma t}) + (1 - e^{-\lambda t} - e^{-\gamma t})]$$
$$= (e^{-\gamma t}) \tag{32}$$

Thus, in the AODV+, the expected time spent for discovering gateway (E[T]) can be computed as expression (33).

$$E[T] = \int_0^\infty F(t)dt$$
$$= \int_0^\infty e^{-\gamma t}dt$$
$$= \frac{1}{\gamma}$$
$$= \beta \tag{33}$$

Fig. 11 a E[T] with various β. **b** E[T] with various k

Summary In case that the CV can connect to a gateway via a relayed route. The expected time spent for discovering gateway (E[T]) of the AODV+ and the proposed method is compared in Fig. 11. As shown in Fig. 11a, both in the AODV+ and proposed method, E[T] increases when β (i.e. the average time the CV cannot connect to a gateway via a few RVs) increases. Also, as shown in Fig. 11b, E[T] of the AODV+ keeps constant as k increases. Oppositely, E[T] of the proposed method decreases as *k* (i.e. the number of gateways to which the CV can directly connected before a new gateway discovery is issued) increases. Apparently, the expected time spent for discovering gateway (E[T]) of the AODV+ is longer than that of the proposed method.

5 Simulations

5.1 Test Bed and Metrics

The following simulations are implemented in NS2 [32]. Total simulation time is 500 s. 60–100 vehicles are randomly deployed in a 2000 m × 2000 m area. The number of SG and MG is 2–4 and 1–2 respectively. The average velocity of CV and MG is 6–15 and 5 m/s respectively. IEEE 802.11 is used as the MAC layer protocol; whose bandwidth is 2 Mbps. Assumed transmission type is constant bit rate (CBR); whose transmission rate is 512 Kbps. The communication range of a node is 250 m. 10 sessions are created in the experiments. Manhattan mobility model [33] is adopted in the simulations.

The metrics used in the performance evaluation are defined next. Firstly, the *Link lifetime* is defined as the time duration before a link breakage. In this paper, a link can exist between two vehicles or between a vehicle and a gateway. Secondly, the *Time to connect* is defined as the time needed for a CV to get connected with a gateway after issuing a connect request. In this paper, a vehicle can directly connect to a gateway if it gets into the coverage of a gateway. Or, it can indirectly connect to a gateway via a multi-hop relayed route going through a few RVs. Thirdly, the *Packet delivery rate* is defined as the rate of successfully delivered data packets sent from the source node to the destination node. Fourthly, the *End-to-end delay* is the average needed end-to-end delay of all successfully delivered data packets sent from the source node to the destination node. Fifthly, the *Signaling load* is defined as the average number of control messages needed to deliver a single data packet. Sixthly, the *Gateway connectivity* is defined as the percent of time a CV can connect to at least one gateway. The performance of the proposed GaaS is compared with a few well-known routing protocols, including the AODV [23], the GPSR [24], and the AODV+ [25], in terms of packet delivery rate, end-to-end delay, signaling overhead, and gateway connectivity. Furthermore, the effects
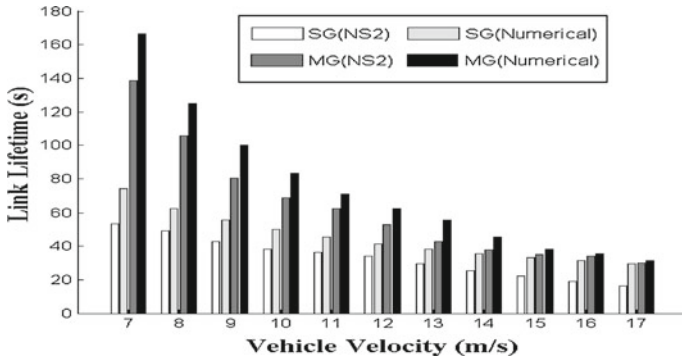
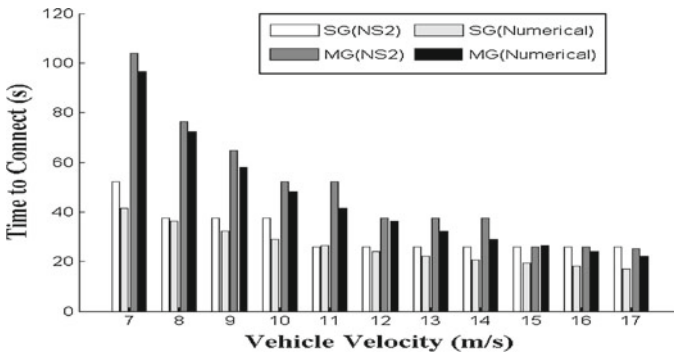**Fig. 12** Link lifetime prediction

Fig. 13 Time needed to connect to a gateway

of a few factors, including the vehicle velocity, the vehicle density, and the gateway density, are investigated.

5.2 Comparing SG and MG

Related simulations are carried out to examine the feasibility of the deployment of SG (Stationary Gateway) and MG (Mobile Gateway) in VANETs. Related results of numerical analyses and simulations are illustrated in Fig. 12 to verify the link lifetime prediction scheme proposed in this paper. Generally speaking, related results of the numerical analyses and simulations are consistent. For both SG and MG, the link lifetime decreases as the vehicle velocity increases. And, the link lifetime of the MG is longer than that of the SG. It implies that the usage of MGs in VANETs is potentially desirable.

In addition, to investigate the time needed for a CV (Client Vehicle) can catch up with and connect to a gateway after issuing a connection request, related results of numerical analyses and simulations are shown in Fig. 13. The results of the numerical analyses and simulations are approximately agreeing. For both MG and SG, the time needed to connect to a gateway decreases as the vehicle velocity increases. And, the time needed to connect to the MG is longer than that of connecting to the SG. Remind that the velocity of a SG is zero. Therefore, the relative velocity between the CV and the MG is smaller than that between the CV and the SG. Hence, more time is needed for a CV to catch up with and connect to a MG.
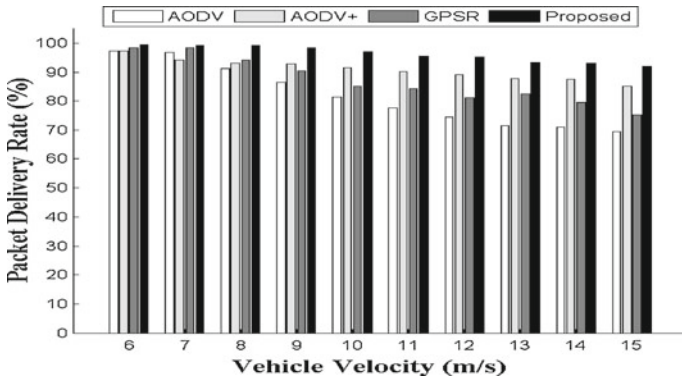
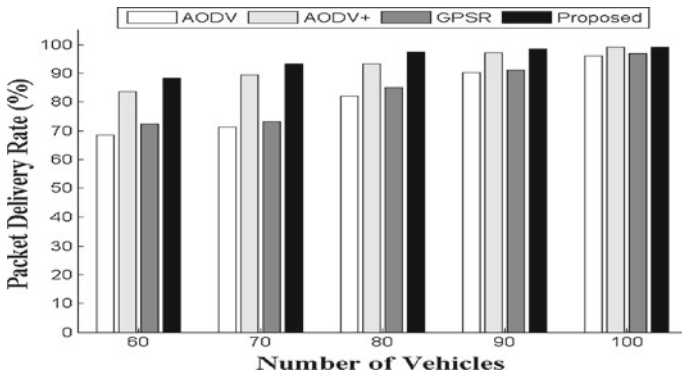**Fig. 14** Packet delivery rate for various vehicle velocity

Fig. 15 Packet delivery rate for various vehicle density

5.3 Packet Delivery Rate

In Fig. 14, for all four methods, the packet delivery rate decreases as the vehicle velocity increases. Because increased vehicle velocity results in more frequent route breakage and corresponding lower packet delivery rate. The proposed method presents higher packet delivery rate than the other three methods for various vehicle velocity.

Also, as shown in Fig. 15, the packet delivery rate increases as the vehicle density increases for all four methods. In this paper, when the CV moves out of the coverage of a gateway, the RVs forward messages from/to related gateways. Thus, increased vehicle density supplies more vehicles for being employed as RVs. That is, increased vehicle density raises the success rate of arranging the RVs for forwarding packets; that results in higher packet delivery rate. And, the proposed method presents higher packet delivery rate than the other three methods for various vehicle density.

Moreover, Fig. 16 displays that, for all four methods, the packet delivery rate increases as the gateway density increases. Increased gateway density offers more gateways for handoff. That is, increased gateway density increases the overall coverage of gateways and promotes the success rate of connecting to a gateway; that generates higher packet delivery rate. As well, the proposed method presents higher packet delivery rate than the other three methods
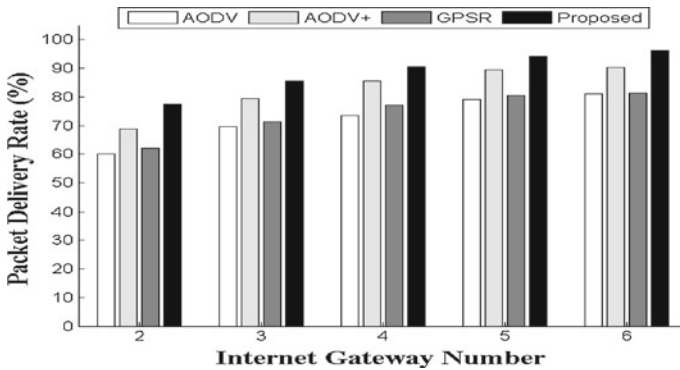
Fig. 16 Packet delivery rate for various gateway density

Table 5 Packet delivery rate comparisons

	Vehicle velocity	Vehicle density	Gateway density
AODV	Lowest	Lowest	Lowest
AODV+	Higher	Higher	Higher
GPSR	Lower	Lower	Lower
Proposed	Highest	Highest	Highest

for various gateway density. The comparisons of packet delivery rate of four methods are listed in Table 5.

5.4 End-to-End Delay

For all four methods, as depicted in Fig. 17, the end-to-end delay increases as the vehicle velocity increases. Because increased vehicle velocity results in more frequent broken connections and more gateway discovery; that thus yields longer end-to-end delay. The proposed method presents shorter end-to-end delay than the other three methods for various vehicle velocities.

In Fig. 18, the end-to-end delay decreases as the vehicle density increases for all four methods. In this paper, when the CV moves out of the coverage of any gateways, the RVs forward messages from/to related gateways. Thus, increased vehicle density supplies more RVs for utilization. That is, increased vehicle density raises the success rate of arranging the RVs; that causes less end-to-end delay. The proposed method causes less end-to-end delay than the other three methods for various vehicle densities.

As seen in Fig. 19, for all four methods, the end-to-end delay decreases as the gateway density increases. Increased gateway density offers more gateways for selection for handoff. That is, increased gateway density increases the overall coverage of gateway and promotes the success rate of connecting to a gateway; that results in less time for gateway discovery and corresponding less end-to-end delay. Besides, the proposed method needs less end-to-end delay than the other three methods for various gateway densities. The comparisons of end-to-end delay for four methods are listed in Table 6.
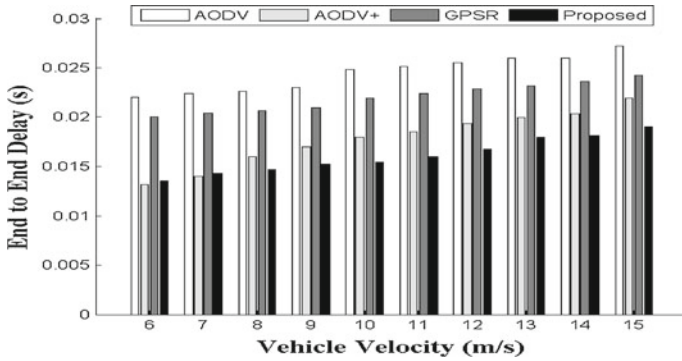
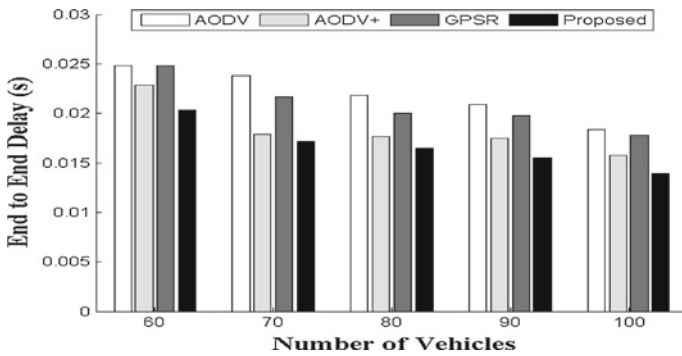**Fig. 17** End-to-end delay for various vehicle velocity

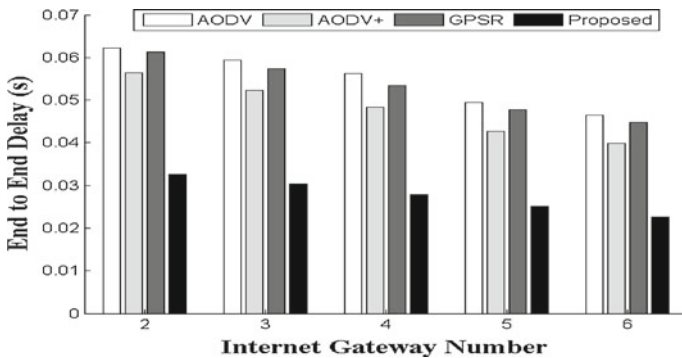**Fig. 18** End-to-end delay for various vehicle density

Fig. 19 End-to-end delay for various gateway density

5.5 Signaling Overhead

In Fig. 20, for all four methods, the signaling load increases as the vehicle velocity increases. Because increased vehicle velocity results in more frequent broken connections and more gateway discovery; that results in more signaling load. The proposed method presents lowest signaling load than the other three methods for various vehicle velocity.

Table 6 End-to-end delay comparisons

	Vehicle velocity	Vehicle density	Gateway density
AODV	Longest	Longest	Longest
AODV+	Shorter	Shorter	Shorter
GPSR	Longer	Longer	Longer
Proposed	Shortest	Shortest	Shortest
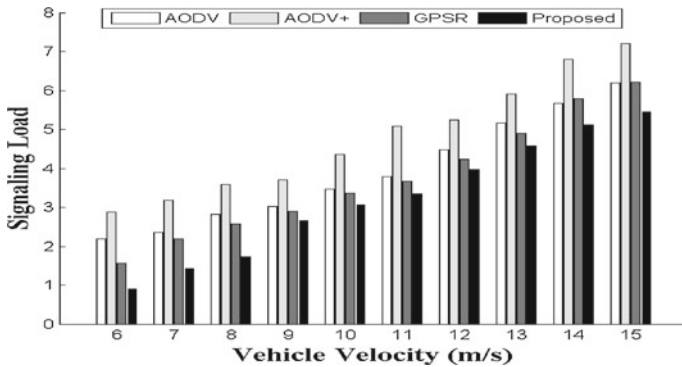
Fig. 20 Signaling load for various vehicle velocity
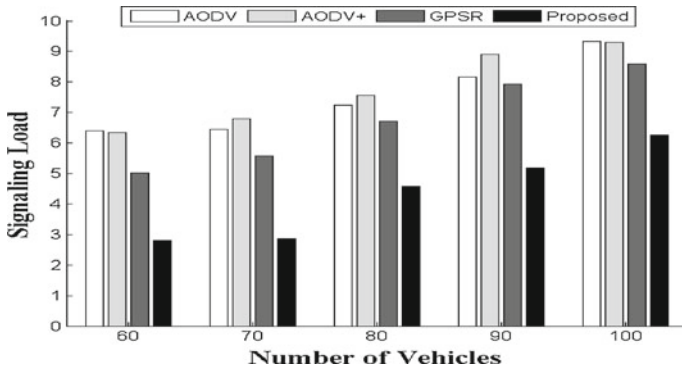
Fig. 21 Signaling load for various vehicle density

You can also find in Fig. 21, the signaling load increases as the vehicle density increases for all four methods. Because that increased vehicle density raises the number of signaling messages sent in the system; that needs more signaling load. Also, the proposed method presents less signaling load than the other three methods for various vehicle density.

Figure 22 illustrate that the signaling load decreases as the gateway density increases for all four methods. Increased gateway density offers more gateways for handoff. In other words, increased gateway density increases the overall coverage of gateway and promotes the success rate of connecting to a gateway; that results in less signaling load for gateway discovery. The proposed method presents less signaling load than the other three methods for various gateway density. Note that, AODV+ yields most signaling load. Proactive gateway discovery is adopted in AODV+. Thus, related signaling messages are periodically broadcast
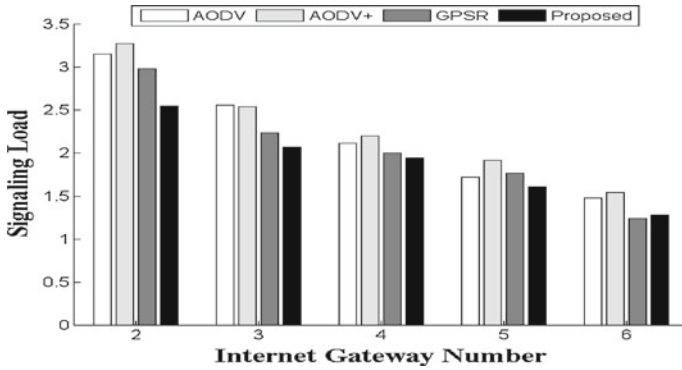
Fig. 22 Signaling load for various gateway density

Table 7 Signaling load comparisons

	Vehicle velocity	Vehicle density	Gateway density
AODV	Higher	Higher	Higher
AODV+	Highest	Highest	Highest
GPSR	Lower	Lower	Lower
Proposed	Lowest	Lowest	Lowest

in the system to discover the gateways; that causes more signaling load. The comparisons of signaling load for four methods are listed in Table 7.

5.6 Gateway Connectivity Ratio

In Fig. 23, the relationship of the gateway connectivity and the vehicle velocity is not remarkably obvious. The proposed method presents higher gateway connectivity than the other three methods for various vehicle velocities. Similarly, the relationship of the gateway connectivity and the vehicle density is not impressively apparent in Fig. 24. The proposed method presents higher gateway connectivity than the other three methods for various vehicle densities.
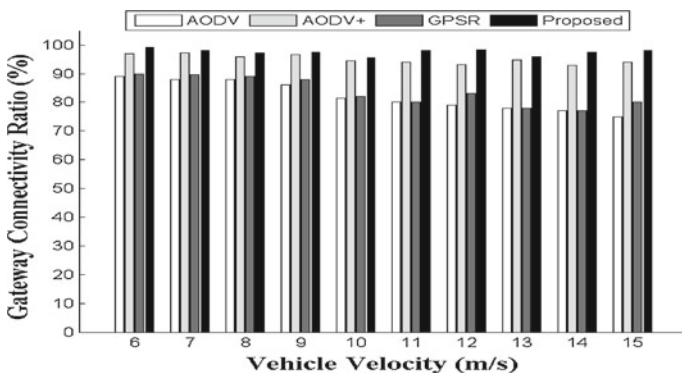

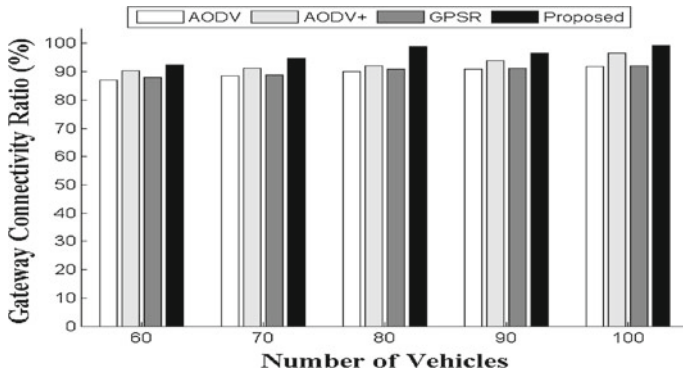**Fig. 23** Gateway connectivity ratio for various vehicle velocity

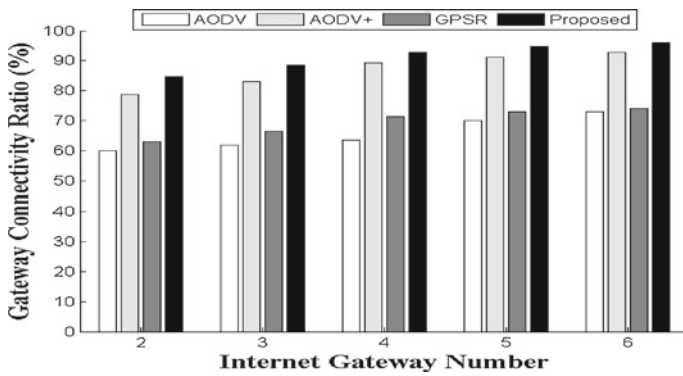**Fig. 24** Gateway connectivity ratio for various vehicle density

Fig. 25 Gateway connectivity ratio for various gateway density

As seen in Fig. 25, for all four methods, the gateway connectivity increases as the gateway density increases. Gateway density directly determines the success rate of discover proper gateway for handoff. Increased gateway density offers more gateways for handoff. Therefore, increased gateway density increases the overall coverage of gateway and promotes the success rate of connecting to a gateway; that offers higher gateway connectivity. As well, the proposed method presents higher gateway connectivity than the other three methods for various gateway densities. The comparisons of gateway connectivity for four methods are listed in Table 8.

5.7 Performance Comparison

In both the AODV and the AODV+, the gateway with least hop count is selected for handoff. Oppositely, in GPSR, the gateway being closest to the destination is picked as the next-hop. Comparatively, in the proposed method, the gateway with longest link lifetime is chosen as the successor; that yields less broken connections and less gateway discovery. Thence, it presents higher packet delivery rate, shorter end-to-end delay, less signaling load, and raised gateway connectivity. Moreover, in the proposed method, assisted by related cloud servers, the next gateway is selected and reserved in advance for potential handoff; that yields less handoff delay and reduced service interruption time. As a result, it presents greatly improved packet delivery rate, less end-to-end delay, less signaling load, and better gateway connectivity. The performance of the proposed method and three well-known methods are compared in Table 9.

Table 8 Gateway connectivity comparisons

	Vehicle velocity	Vehicle density	Gateway density
AODV	Lowest	Lowest	Lowest
AODV+	Higher	Higher	Higher
GPSR	Lower	Lower	Lower
Proposed	Highest	Highest	Highest

Table 9 Performance comparisons

	Packet delivery rate	End-to-end delay	Signaling load	Gateway connectivity
AODV	Lowest	Longest	Higher	Lowest
AODV+	Higher	Shorter	Highest	Higher
GPSR	Lower	Longer	Lower	Lower
Proposed	Highest	Shortest	Lowest	Highest

The results imply that the proposed method outperforms the other three methods in terms of packet delivery rate, end-to-end delay, signaling load, and gateway connectivity.

6 Conclusions

A cloud-supported gateway model, namely the GaaS, is proposed in this paper for stably providing seamless Internet access in ITSs. In the GaaS, a few cloud servers are designated for gateway management. Thus, the heavy load of gateway government is shifted from the clients to the designated cloud servers. Numerical analyses and simulation results suggest that the proposed gateway model uplifts the system performance including packet delivery rate, end-to-end delay, signaling overhead, and gateway connectivity; and, shall promote the usage experience of the Internet-enabled ITS services. To implement the GaaS system, participatory vehicles need furnishing on board units (OBUs) supporting the generation and dissemination of the positioning information (i.e. the location, speed, and moving direction of the vehicles). Besides, the OBUs are capable of receiving and processing information delivered by the cloud servers. Correspondingly, related WiFi/WiMAX APs and/or Cellular BSs have to be properly deployed alongside the road for transferring related information from/to associated cloud servers to/from the clients. The cloud servers and gateways need being equipped with related data structures and algorithms proposed in the GaaS. Thoughtfully, the scalability problem will profoundly impact the system performance. Scalable schemes based on hierarchy or cluster should be creatively elaborated in the future.

References

1. ITS. Available at: http://en.wikipedia.org/wiki/Intelligent_transportation_system.
2. Moustafa, H., & Zhang, Y. (2009). *Vehicular networks: techniques, standards, and applications*. Boca Raton, FL: CRC Press.

3. Zhu, K., Niyato, D., Wang, P., Hossain, E., & Kim, D. I. (2011). Mobility and handoff management in vehicular networks: A survey. *Wireless Communications and Mobile Computing*, *11*(4), 459–476.
4. Baldessari, R., Festag, A., & Abeille, J. (2007). Nemo meets VANET: A deployability analysis of network mobility in vehicular communication. In *Proceedings of ITST*, pp. 1–6.
5. Perkins, C. E., Malinen, J. T., Wakikawa, R., Nilsson, A., & Tuominen, A. (2002). Internet connectivity for mobile ad hoc networks. *Wireless Communications and Mobile Computing*, *2*(5), 465–482.
6. Iera, A., Molinaro, A., Polito, S., & Ruggeri, G. (2009). A multi-layer cooperation framework for QoS-aware Internet access in VANETs. *Ubiquitous Computing and Communication Journal*, *4*(3), 1–10.
7. Benslimane, A., Barghi, S., & Assi, C. (2011). An efficient routing protocol for connecting vehicular networks to the Internet. *Pervasive and Mobile Computing*, *7*, 98–113.
8. Namboodiri, V., Agarwal, M., & Gao, L. (2004). A study on the feasibility of Mobile Gateways for vehicular ad hoc networks. In *Proceedings of VANET'04, 1st ACM international workshop on vehicular ad hoc networks*.
9. Benslimane, A., Taleb, T., & Sivaraj, R. (2011). Dynamic clustering-based adaptive mobile gateway management in integrated VANET-3G heterogeneous wireless networks. *IEEE Journal on Selected Areas in Communications*, *29*(3), 559–570.
10. Luo, J., Gu, X., Zhao, T., & Yan, W. (2010). A mobile infrastructure based VANET routing protocol in the urban environment. In *Proceedings of international conference on communications and mobile computing (CMC)*, pp. 432–437.
11. Pan, H. Y., Jan, R. H., Jeng, A. A. K., Chen, C., & Tseng, H. R. (2011). Mobile gateway routing for vehicular networks. In *Proceedings of the 8th IEEE Asia Pacific wireless communication symposium (APWCS 2011)*.
12. Ammari, H. M. (2007). A survey of current architectures for connecting wireless mobile ad hoc networks to the Internet: Research articles. *International Journal of Communication Systems*, *20*(8), 943–968.
13. Bechler, M., & Wolf, L. (2005). Mobility management for vehicular ad hoc networks. In *Proceedings of IEEE VTC Spring* (Vol. 4, pp. 2294–2298).
14. Bechler, M., Wolf, L., Storz, O., & Franz, W. J. (2003). Efficient discovery of Internet gateways in future vehicular communication systems. In *Proceedings of IEEE VTC Spring* (Vol. 2, pp. 965–969).
15. Perkins, C. (Ed.) (2002). *IP mobility support for IPv4. RFC 3344*.
16. Jonsson, U., Alriksson, F., Larsson, T., Johansson, P., Gerald, J., & Maguire, Q. (2000). MIPMANET: Mobile IP for mobile ad hoc networks. *Proceedings of ACM MobiHoc*.
17. Korner, U., Hamidian, A., & Nilsson, A. (2004). Performance of Internet access solutions in mobile ad hoc networks. In *Proceedings of wireless systems and mobility in next generation Internet, first international workshop of the EURO-NGI network of excellence*, Dagstuhl Castle, Germany, June 7–9, 2004.
18. Sommer, C., & Dressler, F. (2007). The DYMO routing protocol in VANET scenarios. In *Proceedings of IEEE VTC*.
19. Chakeres, I., & Perkins, C. (2010). *Dynamic MANET on-demand (DYMO) routing*. draft-ietf-manet-dymo-19, March 22–September 23, 2010.
20. Lin, Y. W., Chen, Y. S., & Lee, S. L. (2010). Routing protocols in vehicular ad hoc networks: A survey and future perspectives. *Journal of Information Science and Engineering*, *26*(3), 913–932.
21. Kakarla, J., Sathya, S. S., Laxmi, B. G., & Ramesh Babu, B. (2011). A survey on routing protocols and its issues in VANET. *International Journal of Computer Applications 28*(4), 38–44.
22. Ochi, L. S., Vianna, D. S., Drummond, L. M. A., & Victor, A. (1998). A parallel evolutionary algorithm for the vehicle routing problem with heterogeneous fleet. *Future Generation Computer Systems*, *14*, 285–292.
23. Perkins, C., Royer, E.B., & Das, S. (2003). Ad-hoc on-demand distance vector (AODV) routing. In *Proceedings of Network Working Group*, RFC 3561, July 2003.
24. Karp, B., & Kung, H. T. (2002). GPSR: Greedy perimeter stateless routing for wireless networks. In *Proceedings of ACM Mobil-Com*.
25. Hamidian, A., Körner, U., & Nilsson, A. (2005). Performance of Internet access solutions in mobile ad hoc networks. *Lecture notes in computer science* (Vol. 3427/2005, pp. 189–201).
26. Su, W., Lee, S. J., & Gerla, M. (2001). Mobility prediction and routing in ad hoc wireless networks. *International Journal of Network Management*, *11*(1), 3–30.
27. Buyya, R., et al. (Eds.). (2010). *Cloud computing: Principles and paradigms*. London: Wiley.
28. Gillam, L., et al. (Eds.). (2010). *Cloud computing: Principles, systems and applications*. Berlin: Springer.
29. Cloud computing. Wikipedia, available at: http://en.wikipedia.org/wiki/Cloud_computing.
30. Buyyaa, R., Yeoa, C. S., Venugopala, S., Broberg, J., & Brandic, I. (2009). Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility. *Future Generation Computer Systems*, *25*, 599–616.

31. Gaoa, J., Xiao, Y., Liu, J., Liang, W., & Chen, C. L. P. (2012). A survey of communication/networking in smart grids. *Future Generation Computer Systems*, *28*, 391–404.
32. NS2. Available at: http://www.isi.edu/nsnam/ns.
33. Bai, F., Sadagopan, N., & Helmy, A. (2003). IMPORTANT: A framework to systematically analyze the impact of mobility on performance of routing protocols for ad hoc networks. In *Proceedings of IEEE INFOCOM* (Vol. 2, pp. 825–835).

Author Biographies

Yen-Wen Lin is currently an associate professor in the Department of Computer Science at National Taichung University of Education, Taichung, Taiwan, R.O.C. Her current research interests include mobile computing, wireless Internet, and IoT.

Jie-Min Shen received his M.S. degree in Department of Computer Science from the National Taichung University of Education, Taiwan, R.O.C. in 2011. His primary research interests include mobile computing and VANET.

Hao-Chun Weng received his M.S. degree in Department of Computer Science from the National Taichung University of Education, Taiwan, R.O.C. in 2012. His research interests include mobile computing, VANET, and ITS.