1-1-2014

A secure IPv6 address configuration protocol for vehicular networks

Xiaonan Wang
University of Wollongong

Yi Mu
University of Wollongong, ymu@uow.edu.au

Guangjie Han
Hohai University

Deguang Le
Changshu Institute of Technology

A secure IPv6 address configuration protocol for vehicular networks

Abstract

This paper proposes a secure address configuration protocol for IPv6-based vehicular networks. In this protocol, the network architecture is proposed. In this architecture, a vehicle obtains a unique address from a neighbor vehicle or an access point without DAD, and a leaving vehicle's address space can be automatically reclaimed for reassignment. Based on this architecture, the address configuration algorithm is presented. In this algorithm, an access point or a vehicle owns the unique address space and assigns a unique address to a neighbor vehicle without DAD, so the address configuration cost and delay are lowered. The identification of a vehicle can be authenticated, so the security is achieved. This paper evaluates the performance of this protocol. The data results show that this protocol effectively improves the address configuration performance. 2014 Springer Science+Business Media New York.

Keywords

ipv6, vehicular, networks, address, configuration, secure, protocol

Publication Details

Wang, X., Mu, Y., Han, G. & Le, D. (2014). A secure IPv6 address configuration protocol for vehicular networks. Wireless Personal Communications, 79 (1), 721-744.

A secure IPv6 address configuration protocol for vehicular networks

Xiaonan Wang[1] , Yi Mu[2] , Guangjie Han[3] Deguang Le[1]

1. School of Computer Science & Engineering, Changshu Institute of Technology, Changshu, China

2. Centre for Computer and Information Security Research, School of Computer Science and Software Engineering, University of Wollongong, Wollongong, NSW, Australia

3. Department of Information & Communication Systems, Hohai University, Changzhou, China

Abstract: This paper proposes a secure address configuration protocol for IPv6-based vehicular networks. In this protocol, the network architecture is proposed. In this architecture, a vehicle obtains a unique address from a neighbor vehicle or an access point without DAD, and a leaving vehicle's address space can be automatically reclaimed for reassignment. Based on this architecture, the address configuration algorithm is presented. In this algorithm, an access point or a vehicle owns the unique address space and assigns a unique address to a neighbor vehicle without DAD, so the address configuration cost and delay are lowered. The identification of a vehicle can be authenticated, so the security is achieved. This paper evaluates the performance of this protocol. The data results show that this protocol effectively improves the address configuration performance.

Key Words: Vehicular network, IPv6 address, address configuration, security, vehicle

1 Introduction

With the development of the vehicular networks and the IPv6 Internet, a variety of services can be acquired via the IPv6 Internet, including watching TV, acquiring real-time traffic information, etc [1]. Before a vehicle performs the proper communication with the Internet, it must acquire a globally unique IPv6 address. In IPv6, there are two typical address configuration protocols, namely, the stateful protocol and the stateless protocol based on DAD (duplicate address detection) [2-3]. However, due to the high address configuration cost and considerable latency, these two protocols cannot work efficiently in vehicular networks, so there is a strong motivation to propose an IPv6 address configuration for vehicular networks.

An important issue of achieving the address configuration in vehicular networks is to reduce the cost and delay. Moreover, an IPv6 address configuration scheme for vehicular networks should be able to meet the following objectives:

1) Dynamic

In vehicular networks, some vehicles may join or leave without alerting due to the unpredictable reasons, so the address configuration scheme must automatically assign a unique IPv6 address to joining vehicles and retrieve the address space occupied by leaving vehicles.

2) Uniqueness

The prerequisite for a vehicle performing the proper communication with the IPv6 Internet is to ensure that a vehicle's address is globally unique. Therefore, the address configuration scheme must achieve the global uniqueness of an assigned IPv6 address.

3) Scalability

In general, with the growth in the number of nodes, the address configuration cost increases exponentially due to flooding [4]. As a result, the scalability is limited. Therefore, the address configuration scheme must lower the cost in order to enhance the scalability.

4) Security

The address configuration scheme should offer security to prevent malicious attacks [6]. The major malicious attacks related to dynamic address configuration mainly include [4-6]:

- Address spoofing attack: if a malicious node spoofs a good node's address to flood a DAD message in the entire network, then network resources will be exhausted and network performance will be substantially degraded. Moreover, a malicious node can also spoof an address to intercept the network data.
- Address exhaustion attack: if a malicious node claims as many IP addresses as possible, then the address resources will be exhausted. As a result, a good node might not be able to acquire an address.
- False address conflict attack: if a malicious node falsely sends address conflict messages, then a good node can be prevented from obtaining an address.
- Replay attack: if a malicious node intercepts an address request message, then it can resend the same address request message to claim as many IP addresses as possible in order to exhaust address resources and consume network resources. Moreover, if a malicious node intercepts an address response message, then it can retransmit the same address response message to a new node in order to cause address conflict.

This paper aims to propose an address configuration protocol for vehicular networks in order to solve the following problems:

1) Achieve the address configuration objectives, namely dynamic, uniqueness, scalability and security.

2) Reduce the address configuration cost and delay.

This paper has the following contributions:

1) The architecture for vehicular networks is proposed. In this architecture, a vehicle obtains a unique address from a neighbor vehicle or an AP(Access point) without DAD, and the address spaces occupied by leaving vehicles can be automatically reclaimed for reassignment. Therefore, the dynamic address configuration is achieved. The cost and delay are also reduced.

2) Based on this architecture, the address configuration algorithm is proposed. In this algorithm, an

AP or a vehicle owns the unique address space and can assign a unique address to a neighbor vehicle without DAD, so the uniqueness of an assigned address is ensured.

3) The control messages are transmitted within one-hop scope, so the scalability is obtained. The cost and delay are also lowered.

4) The identification of a vehicle can be authenticated, so the security is achieved.

XN Wang et al.[6] propose a secure IPv6 address configuration scheme for a MANET(Mobile ad hoc network). This scheme is based on one IP domain, and achieves the distributed address configuration for a MANET. Moreover, the security of this scheme is achieved via authentication. However, as a special kind of MANET, a vehicular network has its own features. Taking these features into account, this work is totally different from the previous work [6] in the following aspects:

1) The network architectures are different. A vehicular network is a special type of MANET and it spans multiple IP domains. In a vehicular network, when a vehicle enters a new IP domain, its network prefix correspondingly changes. Therefore, the architecture in this scheme is based on multiple IP domains. In the previous work, a MANET is limited to one IP domain, so the network prefix of a mobile node keeps unchanged. Therefore, the architecture in the previous work is based on one IP domain.

2) The IPv6 address structures are different. In this scheme, the network architecture is based on road segments. In order to improve the address configuration performance and contribute to the hierarchical routing, an IPv6 address is made up of RS ID and node ID. In the previous work, the network architecture is based on the tree topology where a gateway node is the root. Correspondingly, an address consists of three parts which are the global routing prefix, the gateway node ID and the node ID.

3) The IPv6 address configuration algorithms are different. A distinctive feature of a vehicular network is high speed, so a vehicle frequently moves from one IP domain to another IP domain. When a vehicle enters a new IP domain, it must be configured with a new IP address. Therefore, a vehicle has to frequently acquire a new address and have the old address reclaimed. This scheme fully takes this characteristic into account and introduces the address exchange algorithm. This algorithm achieves both the address configuration and address reclamation at the same time, so the extra cost and delay caused by the address reclamation is avoided. In the previous work, a mobile node obtains an IPv6 address by joining a tree structure, and the address configuration and address reclamation are two independent processes.

The rest of this paper is organized as follows. The related work on the address configuration is discussed in Section 2, the address configuration protocol for vehicular networks is presented in

Section 3, and the security of this protocol is analyzed in Section 4. The address configuration performance is analyzed in Section 5, and this paper concludes with a summary in Section 6.

2 Related work

The IPv6 address configuration protocols, such as the stateful protocol and stateless protocol [2], can configure a node with a unique address in the absence of intervention. However, these protocols have high cost and long delay, so they are unsuitable for multi-hop networks, such as a MANET or a vehicular network.

At present, some address configuration schemes are proposed for multi-hop networks, mainly including a MANET and a vehicular network.

2.1 Address schemes for MANET

Uttam Ghosh et al.[4] propose a secure IPv4 address configuration scheme. In this scheme, a new node first requests an address from a neighbor node. If the neighbor nodes do not have the address space, then the new node requests an address from a remote node. In the latter case, both the cost and the delay are increased. In [5], a random function is employed to generate random numbers for address assignment. This scheme improves the address configuration performance, but it does not eliminate the address conflict. When the address conflict happens, it is solved by weak DAD or passive DAD. In this situation, the address configuration delay and cost are potentially increased. In [6], a node obtains an address from a neighbor node without DAD, so the address configuration cost is reduced and the delay is avoided.

Sonia Mettali Gammar et al.[7] present an address configuration scheme for a MANET. In this scheme, the messages are transmitted within two-hop scope, so the address configuration cost and delay are reduced to an extent. However, when the address reclamation process is performed, almost every node needs to broadcast a message. Therefore, the network performance is degraded. In [8], the architecture based on clusters is proposed and based on this architecture the address format is created. This scheme also addresses the address reclamation issue. However, the cluster maintenance consumes some network resources to some extent. In [9], a node first gets its position information and then acquires an address based on location information. Since DAD is employed to ensure the address uniqueness, the address configuration performance is degraded. In [10], a node acquires a unique address based on its location information without DAD, so the address configuration performance is improved substantially.

Hyojeong Shin et al. [11] utilize the RGB (Red, Green, Blue) coordinates to perform the address configuration. This scheme improves the address configuration performance, but it limits the network scalability. If the node density is high, the address configuration cost is increased substantially. Elmurod Talipov et al. [12] use the time stamp and random ID to identify a node, so multiple nodes

with the same ID and the same time stamp can not be distinguished and the address configuration may be falsely performed. In addition, different kinds of message pairs are employed to perform the communication, so the mapping between different kinds of message pairs degrades the address configuration performance.

In [13], the network is organized into the tree topology and three types of nodes are defined: a root node, leader nodes, and normal nodes. Normal nodes work as intermediate nodes, and leader nodes are responsible for the address configuration and address reclamation. Inevitably, this centralized communication mode increases the address configuration cost to some extent.

2.2 Address configuration for vehicular networks

As a special kind of MANET, a vehicular network has its characteristics, such as high speed, so the address configuration scheme for vehicular networks must take these characteristics into account. At present, some address configuration schemes for vehicular networks are proposed.

In CAC (Centralized Address Configuration) [14], a remote DHCP server configures all vehicles serially, so the address configuration delay grows to some extent. Moreover, if the speed is fast or the number of vehicles is large, then the packet loss rate also grows.

In [15], each AP works as a DHCP server and can assign an address to a vehicle. An AP does not record the address assignment states, and all the address allocation states are maintained by a balanced server which is also responsible for reclaiming the address resources released by vehicles. After a balanced server returns the recovered address resources to an AP, the address loss can happen because the address buffer is limited.

In [16], when a vehicle is entering a new serving area, it acquires a new address. If a vehicle is unable to communicate with any vehicle in the new area, it cannot acquire an address. Yuh-Shyan Chena et al. [17] improve the scheme [16] and propose an address scheme for vehicular networks. In this scheme, if a vehicle is leaving the serving area, then it passes its address to an intermediate vehicle in the serving area in order to extend the address lifetime. If a vehicle is entering this area, then the intermediate vehicle can assign the address to the entering vehicle. If a vehicle cannot acquire an address from an intermediate vehicle, then it obtains an address from the remote DHCP servers. In the latter case, the address configuration delay is prolonged. In addition, a vehicle broadcasts a message within multi-hop scope according to TTL (Time to live), so the address configuration performance is degraded.

From the above discussion, it can be seen that there are not any secure address configuration schemes for vehicular networks. Therefore, this paper presents a secure address configuration for vehicular networks in order to provide the address configuration security and also reduce the address configuration cost and delay.

3 Address configuration protocol

3.1 Architecture

The proposed architecture has the following design goals:

1) The architecture is based on multiple IP domains.

2) The architecture can help achieve the address configuration objectives, including dynamic, uniqueness, scalability and security.

In this protocol, a vehicular network is made up of APs and vehicles. An AP connects the vehicular networks to the IPv6 Internet. A vehicle communicates with the IPv6 Internet via an AP. The area covered by an AP is called an RS (Road Segment), as shown in Figure 1.

This protocol defines three types of nodes:

LN (Leaving node): If a vehicle is entering the overlay area of the serving RS and the next RS, then for the serving RS this vehicle is called a leaving node.

EN (Entering node): If a vehicle is entering the overlay area of the serving RS and the next RS, then for the next RS this vehicle is called an entering node.

NN (Normal node): A vehicle which is marked neither an LN nor an EN.

In Figure 1, for the RS identified by the AP AP2, the vehicle V1 is an LN, and for the RS identified by the AP AP3, V1 is an EN.

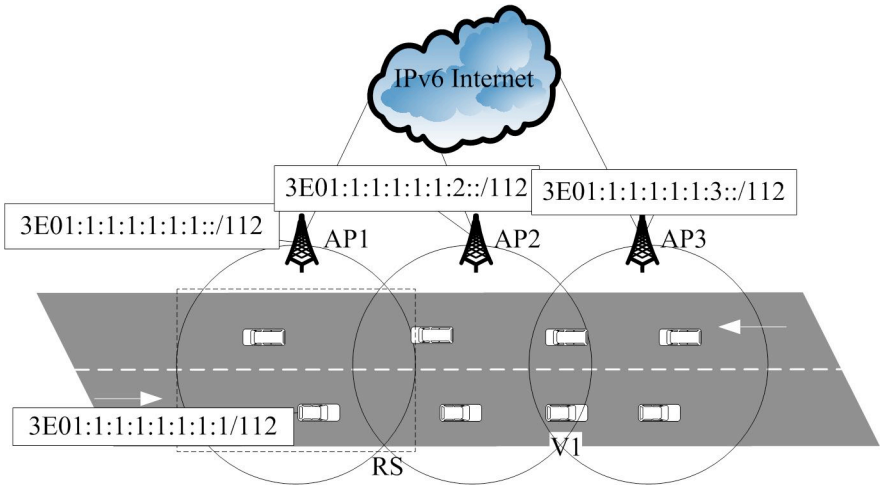


Figure 1 Architecture

3.2 Address structure

Based on the proposed architecture, the address structure for vehicular networks is presented, as shown in Table 1.

Table 1 IPv6 address structure

(128-*i*)bits	*i* bits
RS ID	Vehicle ID

In Table 1, the first part of an IPv6 address is RS ID which is the global routing prefix and identifies an RS. The RS IDs of all vehicles in one RS are the same and the value is equal to the RS ID of the AP in the same RS. The second part is vehicle ID which identifies a vehicle. An AP's address is preconfigured and its vehicle ID is zero.

In Table 1, i is a positive integer, and its value is decided by the size of a vehicular network and the density of vehicles. Taking the generality into account, this protocol sets i to 16, as shown in Figure 1. In this way, one RS can include up to $2^{16}-1$ vehicles.

3.3 Address configuration algorithms

When a vehicle starts or enters a new RS, it must obtain a new IPv6 address to ensure the communication correctness. An AP or a vehicle periodically broadcasts a DSRC message, namely BasicSafetyMessage[18], and the payload includes its public key certificate authorized by CA (Certificate authority). The payload of a DSRC message broadcast by an LN or an EN also contains the addresses of both the next AP and the serving AP, and the one by an NN includes the size of its vehicle ID space.

In this scheme, after a vehicle receives a DSRC message from an AP or a vehicle, it acquires the public key certificate of the AP or vehicle and then computes the hash value with the AP or vehicle's public key and the well-known hash function. If the computed hash value is equal to the hash value embedded in the certificate, then the received public key is authenticated. In this way, a vehicle can get a neighbor AP or a vehicle's public key.

After a vehicle X becomes an EN, it does the following operations:

1) If X is moving from the RS RS1 to the RS RS2 and a vehicle Y is moving from the RS RS2 to the RS RS1, then X exchanges the address with Y in order to achieve the address configuration according to the algorithm in Section 3.3.1.

2) Otherwise, if X receives multiple DSRC messages from neighbour NNs, then it obtains an address from the neighbour NN with the maximum vehicle ID space according to the algorithm in Section 3.3.2.

3) Otherwise, X acquires an address from the next AP according to the algorithm in Section 3.3.3.

3.3.1 Address exchange

It is assumed that for the RS RS1 where the AP is AP1, the vehicle X is an LN and the vehicle Y is an EN, and for the RS RS2 where the AP is AP2, X is an EN and Y is an LN. After X receives a DSRC message from Y, X can acquire Y's public key PubK_Y and it can also determine that Y's next RS is its serving RS and Y's serving RS is its next RS. Therefore, X acquires an address in the next RS according to the following algorithm:

1) X first constructs an Ex_Addr message whose payload is its vehicle ID space and AP2's address.

Then, X calculates the hash value with the hash function H and the Ex_Addr message, and obtains the signature Sign_Ex_Addr with its private key PriK_X, as shown in formula (1). X appends Sign_Ex_Addr to the Ex_Addr message and encrypts the signed Ex_Addr message into an E_Ex_Addr message with PubK_Y, as shown in formula (2). Finally, X sends the E_Ex_Addr message to Y.

$$Sign\_Ex\_Addr = Sign\_Generate(PriK\_X, H(Ex\_Addr)) \quad (1)$$

$$E\_Ex\_Addr = Encrypt\_Generate(PubK\_Y, Ex\_Addr + Sign\_Ex\_Addr) \quad (2)$$

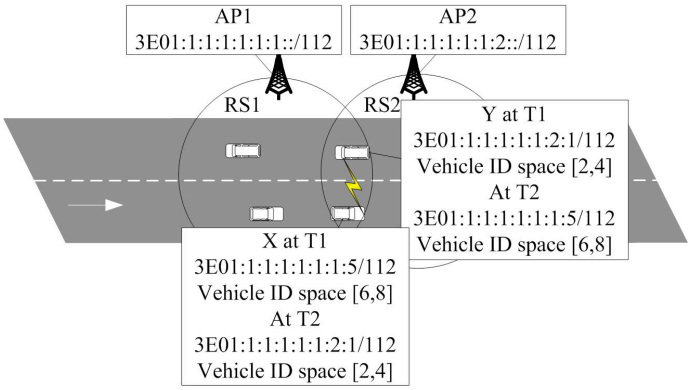
2) After Y receives the E_Ex_Addr message, it decrypts the message with its private key PriK_Y and obtains the Ex_Addr message and the signature Sign_Ex_Addr. With X's public key PubK_X, Y can acquire the hash value. Then, Y calculates the hash value with both the hash function H and the received Ex_Addr message. If the computed hash value is equivalent to the received one, then X is authenticated. Y can determine that X's serving RS is its next RS based on X's address and X's next RS is its serving RS based on the received Ex_Addr message, so Y constructs an Ex_Ack message whose payload is its vehicle ID space, and then sets both its address and vehicle ID space to X's ones. Y acquires the hash value with both the hash function H and the Ex_Ack message, and calculates the signature Sign_Ex_Ack with its private key PriK_Y, as shown in formula (3). Y appends Sign_Ex_Ack to the Ex_Ack message, and encrypts the signed Ex_Ack message into an E_Ex_Ack message with PubK_X, as shown in formula (4). Finally, Y sends the E_Ex_Ack message to X.

$$Sign\_Ex\_Ack = Sign\_Generate(PriK\_Y, H(Ex\_Ack)) \quad (3)$$

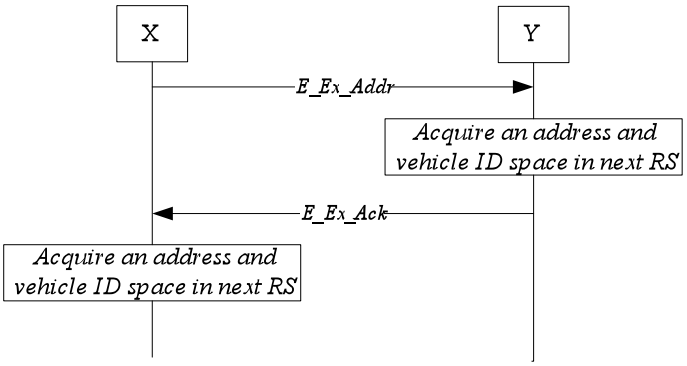
$$E\_Ex\_Ack = Encrypt\_Generate(PubK\_X, Ex\_Ack + Sign\_Ex\_Ack) \quad (4)$$

3) After X receives the E_Ex_Ack message, it fulfills the authentication process. If Y is authenticated, then X sets both its address and vehicle ID space to Y's ones.

4) In this way, X and Y acquire the new addresses and the corresponding vehicle ID spaces in the next RS, as shown in Figure 2(a), Figure 2(b) and Figure 2(c).



(a)



(b)

Algorithmic description of exchanging addresses

For a node X

1 **If** a DSRC message is received from a node Y whose next RS is X's serving RS and whose serving RS is X's next RS **then**
2 Construct an Ex_Addr message;
3 Encrypt the Ex_Addr message into the E_Ex_Addr message;
4 Send the E_Ex_Addr message to Y;
5 Start Timer;
6 **End**
7 **If** an E_Ex_Ack message is received from Y **then**
8 Decrypt the E_Ex_Ack message into the Ex_Ack message;
9 Set both its address and vehicle ID space to Y's ones;
10 Stop Timer;
11 **End**

For a node Y

1 **If** an E_Ex_Addr message is received from X **then**
2 Decrypt the E_Ex_Addr message into the Ex_Addr message;
3 Set both its address and vehicle ID space to X's ones;
4 Construct an Ex_Ack message;
5 Encrypt the Ex_Ack message into the E_Ex_Ack message;
6 Send the E_Ex_Ack message to X;
7 **End**

(c)

Figure 2 Exchange addresses

In Figure 2(a), at the time T1, X's serving RS is RS1, and both its address and vehicle ID space belong to RS1. Y's serving RS is RS2, and both its address and vehicle ID space belong to RS2. At the time T2, X and Y exchange the addresses and vehicle ID spaces. In this way, both X and Y acquire an address and the corresponding vehicle ID space in the next RS. The flow and algorithm description of X and Y exchanging addresses are shown in Figure 2(b) and Figure 2(c).

3.3.2 Address configuration from neighbour NN

It is assumed that for the RS RS1 where the AP is AP1 the vehicle X is an LN, and for the RS RS2 where the AP is AP2 X is an EN. If X receives multiple DSRC messages from neighbour NNs in RS2, then it requests an address from the NN Y with the maximum vehicle ID space $[L,U](L<U)$ in RS2 through the following process:

1) X builds a Req_Addr message, and obtains the hash value with the hash function H and the Req_Addr message. Then, X computes the signature Sign_ Req_Addr with its private key PriK_X, as shown in formula (5). X appends Sign_ Req_Addr to the Req_Addr message and encrypts the signed Req_Addr message into an E_Req_Addr message with Y's public key PubK_Y, as shown in formula (6). Finally, X sends the E_Req_Addr message to Y.

$$Sign\_Req\_Addr = Sign\_Generate(PriK\_X, H(Req\_Addr)) \quad (5)$$

$$E\_Req\_Addr = Encrypt\_Generate(PubK\_Y, Req\_Addr + Sign\_Req\_Addr) \quad (6)$$

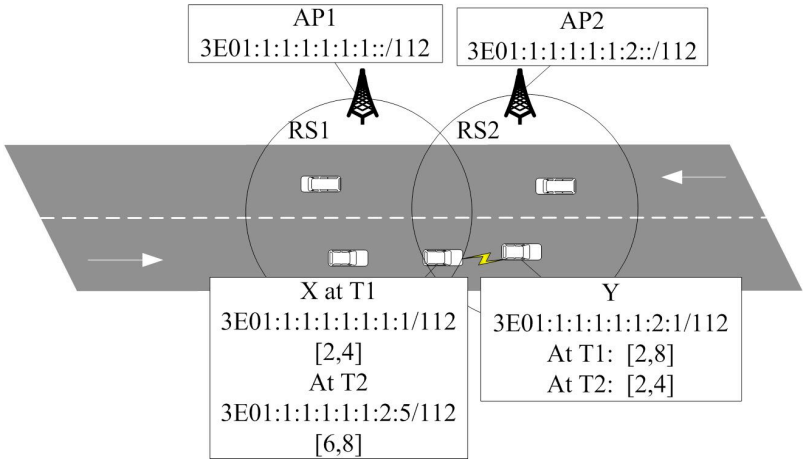
2) After Y receives the E_ Req_Addr message, it decrypts the message with its private key PriK_Y and obtains both the Req_Addr message and the signature Sign_Req_Addr. With X's public key PubK_X, Y can acquire the hash value. Then, Y calculates the hash value with both the hash function H and the received Req_Addr message. If the calculated hash value is equivalent to the received one, then X is authenticated. Y constructs a Res_Addr message whose payload is the assigned vehicle ID space $[\left\lfloor \frac{L+U}{2} \right\rfloor, U]$, and updates its vehicle ID space with $[L, \left\lfloor \frac{L+U}{2} \right\rfloor - 1]$. Y obtains the hash value with both the function H and the Res_Addr message, and acquires the signature Sign_Res_Addr with its private key PriK_Y, as shown in formula (7). Y appends Sign_Res_Addr to the Res_Addr message, and encrypts the signed Res_Addr message into an E_Res_Addr message with PubK_X, as shown in formula (8). Finally, Y sends the E_Res_Addr message to X.

$$Sign\_Res\_Addr = Sign\_Generate(PriK\_Y, H(Res\_Addr)) \quad (7)$$

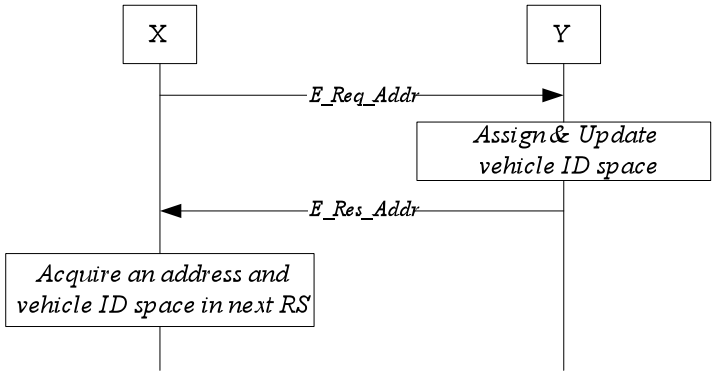
$$E\_Res\_Addr = Encrypt\_Generate(PubK\_X, Res\_Addr + Sign\_Res\_Addr) \quad (8)$$

3) After X receives the E_Res_Addr message, it fulfills the authentication process. If Y is authenticated, then X sets its vehicle ID to $\left\lfloor \frac{L+U}{2} \right\rfloor$ and combines the vehicle ID with Y's RS ID to form an address. Then, X sets its vehicle ID space to $[\left\lfloor \frac{L+U}{2} \right\rfloor + 1, U]$.

4) X acquires the address and the corresponding vehicle ID space in the next RS, as shown in Figure 3(a), Figure 3(b) and Figure 3(c).



(a)



(b)

Algorithmic description of address configuration from neighbour NN

For a node X

1 **If** a DSRC message is received from a neighbour NN Y in X's next RS **then**

2 Construct a Req_Addr message;

3 Encrypt the Req_Addr message into the E_Req_Addr message;

4 Send the E_Req_Addr message to Y;

5 Start Timer;

6 **End**

```
7  If an E_Res_Addr message is received from Y then
8   Decrypt the E_Res_Addr message into the Res_Addr message;
9   Form an IPv6 address;
10  Stop Timer;
11 End
```

For a node Y

```
1    If an E_ Req_Addr message is received from X then
2      Decrypt the E_ Req_Addr message into the Req_Addr message;
3      Construct a Res_Addr message whose payload is the assigned vehicle ID space;
4      Encrypt the Res_Addr message into the E_ Res_Addr message;
5      Send the E_ Res_Addr message to X;
6    End
```

(c)

Figure 3 Address configuration from neighbour NN

In Figure 3(a), at the time T1, X's serving RS is RS1, and Y's serving RS is RS2. At the time T2, for RS2, X becomes an EN. Since X does not find an LN which is moving from RS2 to RS1, it requests an address from the neighbour NN Y with the maximum vehicle ID space in RS2. Finally, X obtains the address and the corresponding vehicle ID space in the next RS. The flow and algorithm description of address configuration from neighbour NN are shown in Figure 3 (b) and Figure 3(c).

3.3.3 Address configuration from an AP

After an LN acquires an address in the next RS, it sends the serving AP a Release message in order to release its address and vehicle ID space. After the serving AP receives the Release message, it stores the Release message for the specified time in order to assign the vehicle ID space in the Release message to an EN. If the AP does not receive an address request from an EN within the specified time, then it reclaims the vehicle ID space in the Release message for reuse.

It is assumed that for the RS RS1 where the AP is AP1 the vehicle X is an EN and the vehicle Y is an LN, and for the RS RS2 where the AP is AP2 X is an LN. It is assume that X receives neither an Ex_Addr message nor a DSRC message from any one neighbour NN in RS1, and AP1 receives or stores a Release message from Y. Then, X obtains an address from AP1 according to the following algorithm:

1) X constructs a Req_Addr message and acquires the signature Sign_Req_Addr according to formula (5). Then, X appends Sign_ Req_Addr to the Req_Addr message and encrypts the signed Req_Addr message into an E_Req_Addr message with AP1's public key PubK_AP1, as shown in formula (9). Finally, X sends the E_Req_Addr message to AP1.

$$E\_Req\_Addr = Encrypt\_Generate(PubK\_AP1, Req\_Addr + Sign\_Req\_Addr) \quad (9)$$

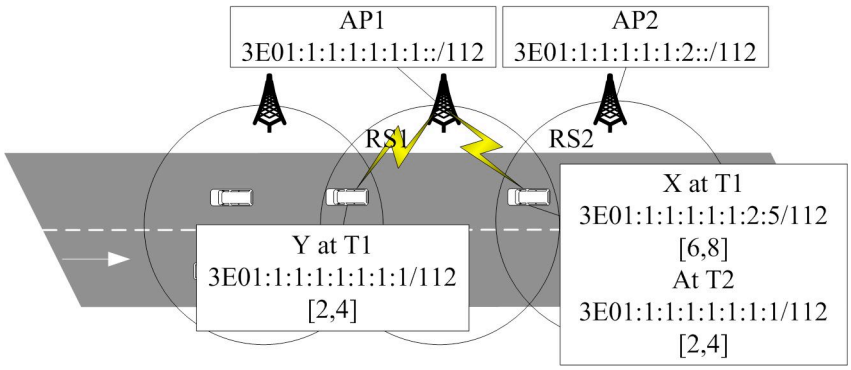
2) After AP1 receives the E_Req_Addr message, it fulfills the authentication process. After X is authenticated, AP1 builds a Res_Addr message whose payload is Y's Release message, and uses its private key PriK_AP1 to generate the signature Sign_Res_Addr according to formula (10). Then, Y appends Sign_Res_Addr to the Res_Addr message, and encrypts the signed Res_Addr message into an E_Res_Addr message with PubK_X, as shown in formula (8). Finally, AP1 sends the E_Res_Addr message to X and resets the life time of the vehicle ID space in the Release message.

$$Sign\_Res\_Addr = Sign\_Generate(PriK\_AP1, H(Res\_Addr)) \quad (10)$$

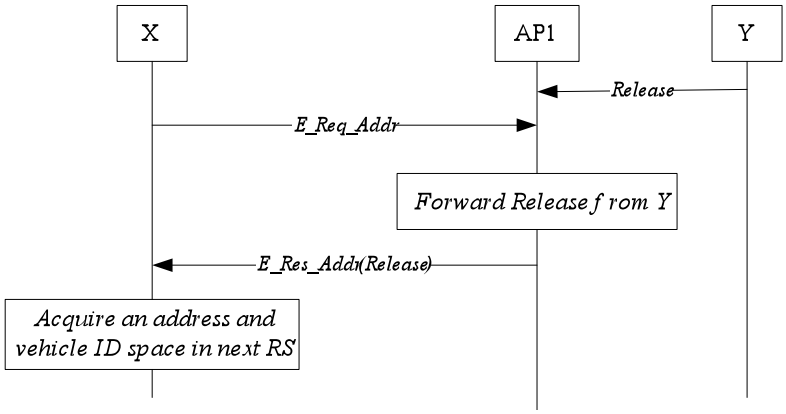
3) After X receives the E_Res_Addr message and authenticates AP1, it acquires Y's Release message. Then X sets its address to Y's address and its vehicle ID space to Y's vehicle ID space.

4) X acquires the address and the corresponding vehicle ID space in the next RS, as shown in Figure 4(a) and Figure 4(b).

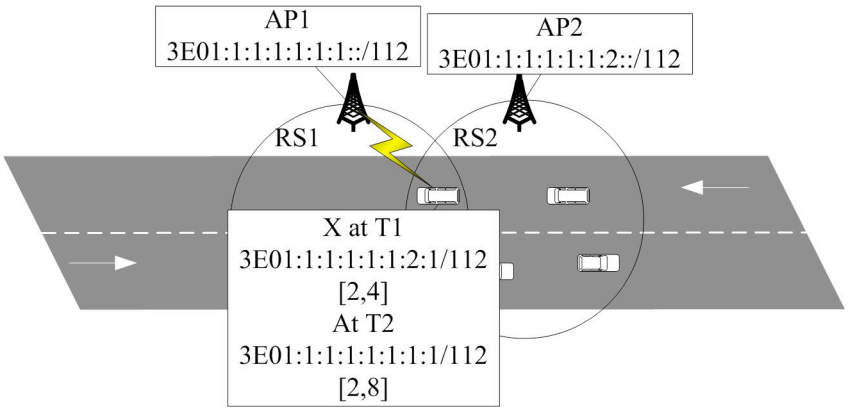
In Figure 4(a), at the time T1, X's serving RS is RS2, and Y's serving RS is RS1. At the time T2, for RS1 X becomes an EN, and AP1 receives and stores a Release message from the LN Y. After X requests an address from AP1, AP1 directly assigns Y's address and vehicle ID space to X. The address configuration flow is shown in Figure 4 (b).



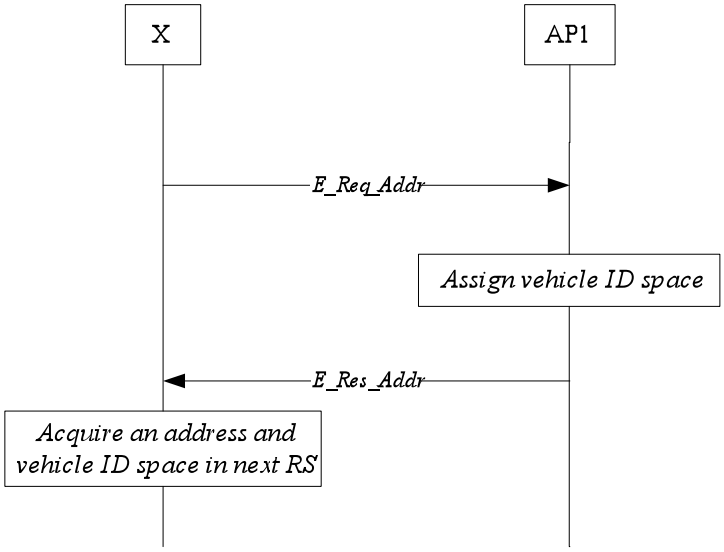
(a)



(b)



(c)



(d)

Algorithmic description of address configuration from an AP

For a node X

1 Construct a Req_Addr message;
2 Encrypt the Req_Addr message into the E_Req_Addr message;
3 Send the E_Req_Addr message to AP1;
4 Start Timer;
5 **If** an E_Res_Addr message is received from AP1 **then**
6 Decrypt the E_Res_Addr message into the Res_Addr message;
7 Form an IPv6 address;
8 Stop Timer;
9 **End**

For an AP AP1

```
1  If an E_ Req_Addr message is received from X then
2    Decrypt the E_ Req_Addr message into the Req_Addr message;
3    Construct a Res_Addr message whose payload is the assigned vehicle ID space;
4    Encrypt the Res_Addr message into the E_ Res_Addr message;
5    Send the E_ Res_Addr message to X;
6  End
```

(e)

Figure 4 Address configuration from an AP

If AP1 does not receive or store a Release address from an LN, then X acquires an address from AP1 according to the following process.

1) X constructs an E_Req_Addr message according to formulae (5) and (9), and then sends the E_Req_Addr message to AP1.

2) After AP1 receives the Req_Addr message, it fulfils the authentication process. If X is authenticated, AP1 constructs a Res_Addr message whose payload is the assigned vehicle ID space $[L,U](L<U)$, and marks the vehicle ID space $[L,U]$ as the assigned state. Then, AP1 generates an E_Res_Addr message according to formulae (8) and (10), sends the E_Res_Addr message to X, and resets the life time of X's address and vehicle ID space.

3) After X receives the E_Res_Addr message and authenticates AP1, it sets its vehicle ID to L and its vehicle ID space to $[L+1,U]$. Then, X combines the vehicle ID with AP1's RS ID to form an address.

4) X acquires the address and the corresponding vehicle ID space in the next RS, as shown in Figure 4(c), Figure 4 (d) and Figure 4 (e).

In Figure 4(c), at the time T1, X's serving RS is RS2. At the time T2, for RS1 X becomes an EN. Since AP1 does not have a Release message, it assigns the address and vehicle ID space to X. In this way, X acquires the address and the corresponding vehicle ID space in the next RS. The flow and algorithm description are shown in Figure 4 (d) and Figure 4 (e).

3.4 Address reclamation

After a vehicle X is configured with an address where the vehicle ID is L, it constructs an Update message whose payload is its vehicle space $[L+1, U]$, and acquires the signature Sign_Update with its private key PriK_X, as shown in formula (11). Then, X appends Sign_Update to the Update message and encrypts the signed Update message into an E_Update message with the serving AP AP1's public key PubK_AP1, as shown in formula (12). Finally, X sends the E_Update message to AP1.

$$Sign\_Update = Sign\_Generate(PriK\_X, H(Update)) \quad (11)$$

$$E\_Update = Encrypt\_Generate(PubK\_AP1, Update + Sign\_Update) \quad (12)$$

After AP1 receives the E_Update message and authenticates X, it resets the life time of the vehicle ID space [*L, U*]. If AP1 does not receive an E_Update message from X within the life time, then it reclaims X's address and vehicle space, namely [*L, U*].

After X becomes an LN and acquires an address in the next RS, it constructs a Release message whose payload is the reclaimed vehicle space [*L,U*]. Then, X obtains the signature Sign_Release with its private key PriK_X, as shown in formula (13). X appends Sign_Release to the Release message and encrypts the signed Release message into an E_Release message with PubK_AP1, as shown in formula (14). Finally, X sends the E_Release message to AP1.

$$Sign\_\mathrm{Re}lease = Sign\_Generate(\mathrm{Pr}iK\_X, H(\mathrm{Re}lease)) \quad (13)$$

$$E\_\mathrm{Re}lease = Encrypt\_Generate(PubK\_AP1, \mathrm{Re}lease + Sign\_\mathrm{Re}lease) \quad (14)$$

After AP1 receives the E_Release message and authenticates X, it stores the Release message for the specified time in order to directly assign the vehicle ID space in the Release message to an EN. If AP1 does not receive an address request from an EN within the specified time, then it reclaims the vehicle ID space in the Release message, namely [*L, U*].

4 Security analysis

This protocol achieves the security and avoids the malicious attacks related to dynamic address configuration, including address spoofing attack, address exhaustion attack, false address conflict attack and replay attack.

4.1 Address spoofing attack

In this protocol, a vehicle can ensure the uniqueness of an assigned IPv6 address without performing DAD process. If a malicious vehicle spoofs a good vehicle's address to flood a false DAD message, its neighbor vehicles first receive this DAD message. After the neighbor vehicles detect that this message is a DAD message, they abandon it because this protocol does not employ the DAD process to ensure the address uniqueness. Therefore, the flooding of false DAD messages is prevented.

If a malicious vehicle M spoofs an IP address of a good vehicle X to allocate an address to a vehicle Y, then it needs to return Y an address response message which must be signed by X's private key. Since the malicious vehicle M does not know X's private key, it cannot generate the correct signature. Therefore, Y can determine that M is a malicious vehicle.

In this way, the address spoof attack is avoided.

4.2 Address exhaustion attack

If a malicious vehicle M wants to request an address from a good vehicle or an AP, then it needs to send the good vehicle or AP an address request message which is signed by M's private key. However, M's public key can not be authenticated by CA, so the good vehicle or AP can detect that M is a malicious node and decline the address request.

In this way, the address exhaustion attack is avoided.

4.3 False address conflict attack

If a malicious vehicle M broadcasts an address conflict message, then its neighbor vehicles first receive this message. Because the neighbor vehicles cannot authenticate the malicious vehicle M, they abandon the address conflict message.

Moreover, this protocol achieves the address configuration without DAD, so it does not employ an address conflict message to ensure the uniqueness of an assigned address. Therefore, if a vehicle receives an address conflict message, then it can determine that this message comes from a malicious vehicle.

In this way, the flooding of false address conflict messages is prevented.

4.4 Replay attack

A malicious node M intercepts an encrypted address request message from a good vehicle X, and resends the same message to a vehicle Y to request an address in order to exhaust the address resources and consume network resources. After Y receives the encrypted address request message, it can abandon this message because the time stamp in the received message is expired.

A malicious node M intercepts an encrypted address response message from a good vehicle Y, and resends the same encrypted address response message to another node X in order to cause the address conflict. After X receives this encrypted address response message, it can discard this message because it can not decrypt the encrypted address response message with its private key.

In this way, the replay attack is avoided.

5 Experiment and evaluation

The existing address configuration protocol [17] is selected to compare with this protocol due to the following reasons:

1) It is a new address configuration protocol for vehicular networks.

2) It has better performance than the protocol [16].

5.1 Analysis

The analytical method in [19] is employed to analyze the performance parameters of the proposed protocol and the existing protocol [17].

1) Existing protocol

In the existing protocol, a leaving vehicle first forwards its address to an intermediate vehicle which then assigns the address to an entering vehicle. Therefore, the upper bounds of the cost and delay of a vehicle obtaining an address are $2O(d)+4O(1)$ and $2O(d)+4O(1)$ respectively where d is the sum of $d1$ and $d2$, $d1$ is the distance between a leaving vehicle and an intermediate vehicle keeping a leaving

vehicle's address, and *d2* is the distance between an entering vehicle and an intermediate vehicle keeping a leaving vehicle's address.

Proof. The upper bounds of the cost and delay of a leaving vehicle (an intermediate vehicle) sending an IP_passing message (an IP_passing_Ack message) are $O(d1)$ and $O(d1)$ respectively. The upper bounds of the cost and delay of an entering vehicle (an intermediate vehicle) sending an IP_Request message (an IP_Reply message) are $O(d2)$ and $O(d2)$ respectively. The upper bounds of the cost and delay of processing a message are $O(1)$ and $O(1)$ respectively.Therefore, the upper bounds of the cost and delay of a vehicle acquiring an address are $2O(d)+4O(1)$ and $2O(d)+4O(1)$ respectively, as shown in Table 2 and 3.

2) Proposed protocol

- Address exchange

In the situation (S1) that a vehicle acquires an address through the address exchange configuration, an Ex_Addr message and an Ex_Ack message are used to perform the address configuration for two vehicles. Therefore, the upper bounds of the cost and delay for a vehicle acquiring an address are $O(1)$ and $O(1)$ respectively.

Proof. The upper bounds of the cost and delay of a vehicle sending an Ex_Addr message are $O(1)$ and $O(1)$ respectively. Also, the upper bounds of the cost and delay of a vehicle returning an Ex_Ack message are $O(1)$ and $O(1)$ respectively. The upper bounds of the cost and delay of processing a message are $O(1)$ and $O(1)$ respectively. Since the address exchange configuration achieves the address configuration for two vehicles, the upper bounds of the cost and delay of a vehicle acquiring an address are $2O(1)$ and $2O(1)$ respectively, as shown in Table 2 and 3.

- Address configuration from a neighbour vehicle

In the situation (S2) that a vehicle gets an address from a neighbour vehicle, a Req_Addr message and a Res_Addr message are used to perform the address configuration for a vehicle. Therefore, the upper bounds of the cost and delay for a vehicle acquiring an address are $4O(1)$ and $4O(1)$ respectively.

Proof. The upper bounds of the cost and delay of a vehicle sending a Req_Addr message are $O(1)$ and $O(1)$ respectively. Also, the upper bounds of the cost and delay of a vehicle returning a Res_Addr message are also $O(1)$ and $O(1)$ respectively. The upper bounds of the cost and delay of processing a message are $O(1)$ and $O(1)$ respectively. Therefore, the upper bounds of the cost and delay of a vehicle obtaining an address are $4O(1)$ and $4O(1)$ respectively, as shown in Table 2 and 3.

- Address configuration from an AP with Release

In the situation (S3) that a vehicle acquires an address from an AP which stores a Release message, a Req_Addr message and a Res_Addr message are used to perform the address configuration for a vehicle, and the upper bounds of the cost and delay for a vehicle acquiring an address are $5O(1)$ and $5O(1)$ respectively.

Proof. The upper bounds of the cost and delay of a vehicle sending a Release message are $O(1)$ and $O(1)$ respectively. The upper bounds of the cost and delay of a vehicle sending a Req_Addr message are $O(1)$ and $O(1)$ respectively. The upper bounds of the cost and delay of an AP returning a Res_Addr message are also $O(1)$ and $O(1)$ respectively. The upper bounds of the cost and delay of processing a Req_Addr message or a Res_Addr message are $O(1)$ and $O(1)$ respectively. Therefore, the upper bounds of the cost and delay of a vehicle acquiring an address are $5O(1)$ and $5O(1)$ respectively, as shown in Table 2 and 3.

- Address configuration from an AP without Release

In the situation (S4) that a vehicle acquires an address from an AP which does not store a Release message, an AP processes a Release message to reclaim the vehicle ID space, and uses a Req_Addr message and a Res_Addr message to perform the address configuration for a vehicle. Therefore, the upper bounds of the cost and delay for a vehicle acquiring an address are $6O(1)$ and $6O(1)$ respectively.

Proof. The upper bounds of the cost and delay of a vehicle sending a Release message are $O(1)$ and $O(1)$ respectively. The upper bounds of the cost and delay of a vehicle sending a Req_Addr message are $O(1)$ and $O(1)$ respectively. The upper bounds of the cost and delay of an AP returning a Res_Addr message are also $O(1)$ and $O(1)$ respectively. The upper bounds of the cost and delay of processing a message are $O(1)$ and $O(1)$ respectively. Therefore, the upper bounds of the cost and delay of a vehicle acquiring an address are $6O(1)$ and $6O(1)$ respectively, as shown in Table 2 and 3.

Table 2 Address configuration cost

		Total cost
Existing protocol [17]		$2O(d)+4O(1)$
Proposed protocol	S1	$2O(1)$
	S2	$4O(1)$
	S3	$5O(1)$
	S4	$6O(1)$

Table 3 Address configuration delay

		Total delay
Existing protocol [17]		$2O(d)+4O(1)$
Proposed protocol	S1	$2O(1)$
	S2	$4O(1)$
	S3	$5O(1)$
	S4	$6O(1)$

5.2 Simulation

We use *ns-2* [20] to evaluate the performance, and the simulation parameters are shown in Table 4. The performance parameters include the address configuration cost, the address configuration delay, the packet loss rate and the address life. Among them, the address life means the interval from the time when an address is assigned to the time when the address is reclaimed. The address reclamation and reassignment can increase the address cost and delay. Therefore, the address life is an important indicator of the address configuration performance. The signature algorithm is RSA and the hash function is SHA-1.

Table 4 Simulation parameters

Parameters	Values
Length of highway	5km
Number of lanes	4 (2 in each direction)
Transmission range	200 m-500m
Speed	10-30 m/s
Vehicle arrival rate	0.1vps(vehicles/s)-0.5vps
MAC	802.11p
Mobility model	Freeway Mobility Model
Simulation time	500s

5.2.1 The effect of speed

When the transmission range is 300m and the vehicle arrival rate is 0.3vps, the address configuration cost, the address configuration delay, the packet loss rate and the address life are shown in Figure 5, 6, 7 and 8.

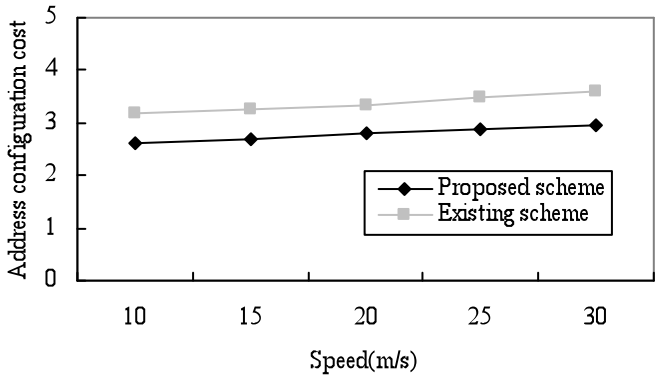


Figure 5 Address configuration cost based on speed

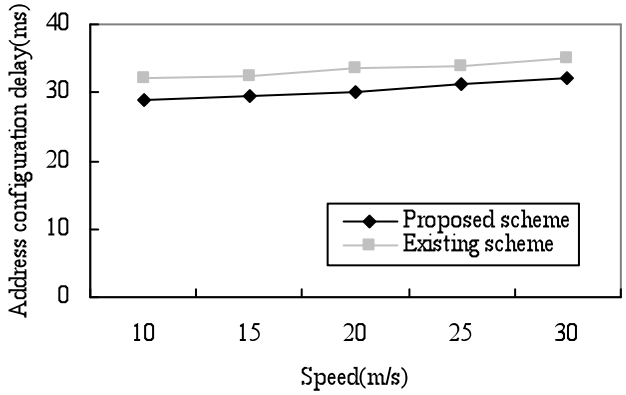


Figure 6 Address configuration delay based on speed

When the speed varies, the total number of a vehicle's neighbor nodes keeps invariable due to the constant transmission range and vehicle arrival rate. As a result, the number of the vehicles acquiring an address in different situations tends to be steady, so the average address configuration cost and delay tend to be constant. The increase in speed causes the growth in the packet loss rate, so the retransmission caused by the packet loss results in the slight increase in both the cost and delay. In the existing protocol, an entering vehicle acquires an address from an intermediate vehicle keeping a leaving vehicle's address. When the speed increases, the distance between an entering vehicle and an intermediate vehicle tends to be stable. But the retransmission caused by the packet loss results in the slight increase in both the cost and delay, as shown in Figure 5 and 6.

In this protocol, a vehicle obtains an address from a neighbor node while in the existing protocol a vehicle gets an address from an intermediate node multi-hop away. Therefore, the address configuration cost and delay in this protocol are lower.

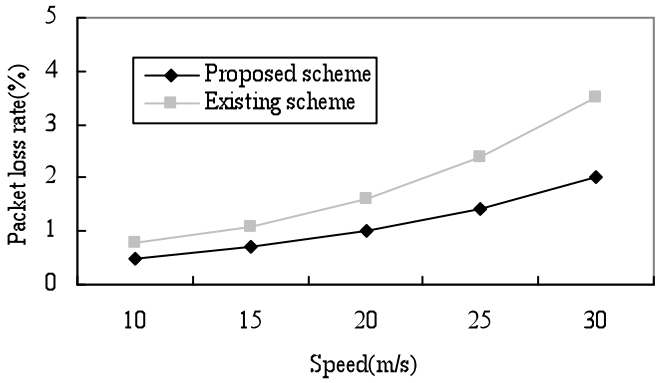


Figure 7 Packet loss rate based on speed

The increase in the speed can cause the growth in the packet loss rate [21]. In this protocol a vehicle gets an address from a neighbor node while in the existing protocol a vehicle obtains an address from an intermediate node multi-hop away. In addition, the address configuration cost and delay in this protocol are smaller, as shown in Figure 5 and 6, so the packet loss rate in the proposed protocol is lower, as shown in Figure 7.

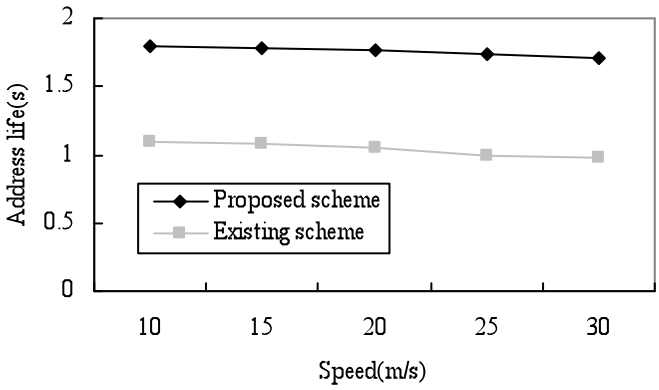


Figure 8 Address life based on speed

When both the transmission range and vehicle arrival rate keep constant, the number of a vehicle's neighbor nodes keeps invariable. Therefore, the number of the vehicles acquiring an address in different situations tends to be stable, so the average address life tends to be steady. In the existing protocol, an entering vehicle acquires an address from an intermediate vehicle keeping a leaving vehicle's address. Since the distance between an entering vehicle and an intermediate node tends to be constant, the address life also tends to be steady, as shown in Figure 8.

In this protocol, only in S4 an LN's address is reclaimed, so the address life is longer than the one in the existing protocol.

5.2.2 The effect of transmission range

When the speed is 30m/s and the vehicle arrival rate is 0.3vps, the address configuration cost, the address configuration delay, the packet loss rate and the address life are shown in Figure 9, 10, 11 and 12.

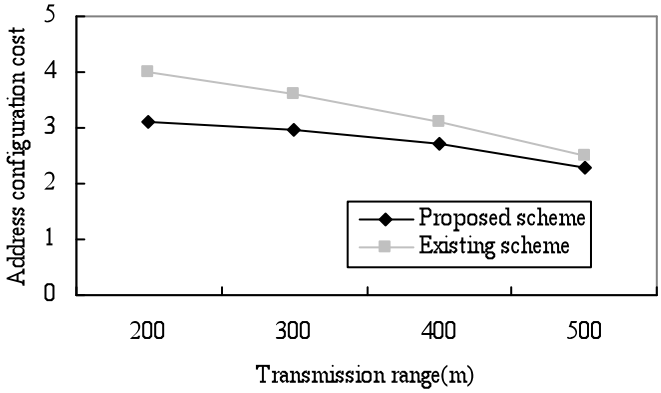


Figure 9 Address configuration cost based on transmission range

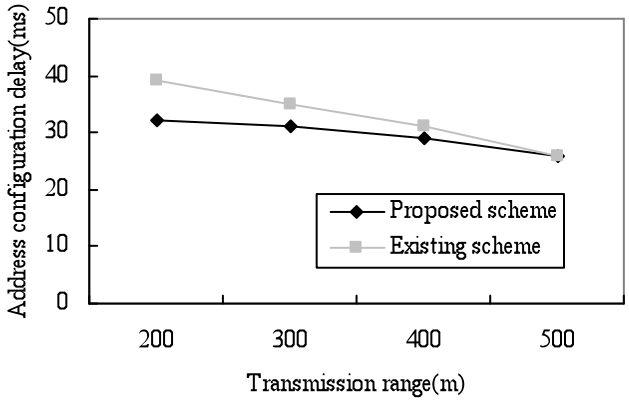


Figure 10 Address configuration delay based on transmission range

In this protocol, with the increase in transmission range, the number of the vehicles acquiring an address in S1 and S2 grows, so the average address configuration cost and delay decrease. In the existing protocol, with the increase in transmission range, the distance between an entering vehicle and a leaving vehicle releasing an address reduces, so the average configuration cost and delay also decrease. In this protocol a vehicle obtains an address from a neighbor node while in the existing protocol a vehicle obtains an address from an intermediate node multi-hop away. Therefore, the address configuration cost and delay in this protocol are lower, as shown in Figure 9 and 10.

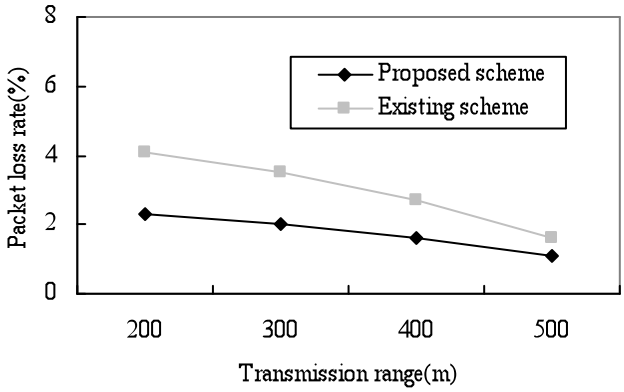


Figure 11 Packet loss rate based on transmission range

In this protocol, with the increase in transmission range, the probability of a vehicle acquiring an address in S1 and S2 grows, so the average address configuration cost and delay are reduced. As a result, the packet loss rate is also decreased. In the existing protocol, with the increase in the transmission range, the distance between an entering vehicle and a leaving vehicle releasing an address is decreased, so the packet loss rate is also reduced. In this protocol the address configuration cost and delay are smaller and a vehicle acquires an address from a neighbor node, so the packet loss rate is lower, as shown in Figure 11.

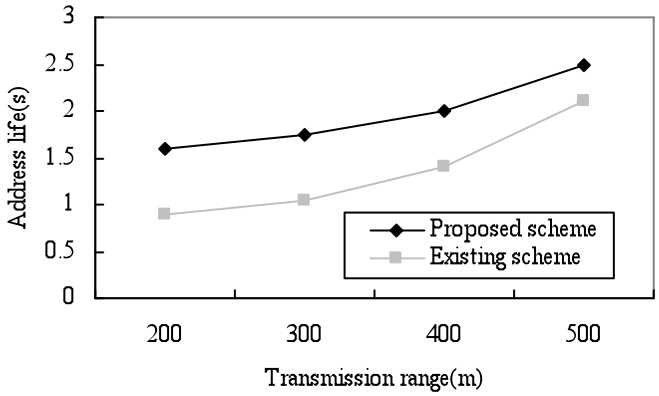


Figure 12 Address life based on transmission range

In this protocol, with the increase in transmission range, the probability of a vehicle acquiring an address in S1 and S2 grows, so the average address life increases. In the existing protocol, with the increase in transmission range, the distance between an entering vehicle and a leaving vehicle releasing an address decreases, so the probability of an entering vehicle acquiring an address from an intermediate vehicle keeping a leaving vehicle's address grows. As a result, the address life is increased, as shown in Figure 12.

5.2.3 The effect of vehicle arrival rate

When the transmission range is 300m and the speed is 30m/s, the address configuration cost, the address configuration delay, the packet loss rate and the address life are shown from Figure 13, 14, 15 and 16.

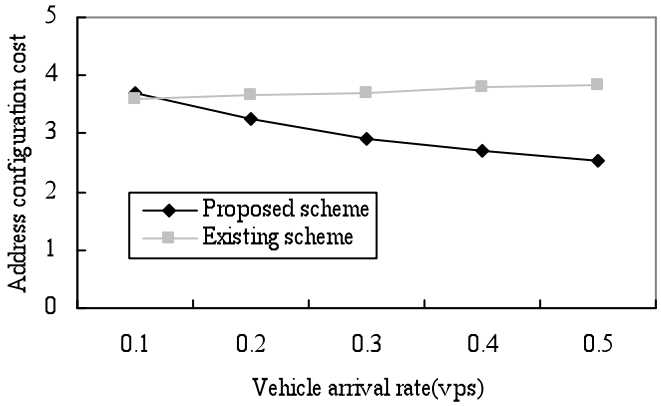


Figure13 Address configuration cost based on vehicle arrival rate

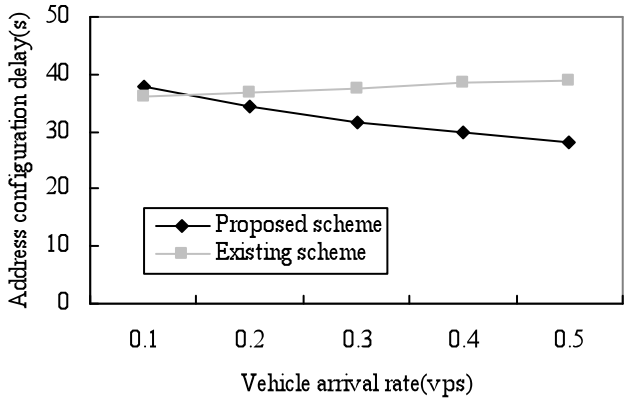


Figure14 Address configuration delay based on vehicle arrival rate

In this protocol, with the increase in vehicle arrival rate, the total number of vehicles grows, so the probability of the vehicles acquiring an address in S1 and S2 increases. As a result, the average address configuration cost and delay decrease. In the existing protocol, the vehicle arrival rate has no relation to the distance between an entering vehicle and a leaving vehicle releasing an address, so the average configuration cost and delay tend to be steady. The retransmission caused by the packet loss results in the slight growth in the cost and delay, as shown in Figure 13 and 14.

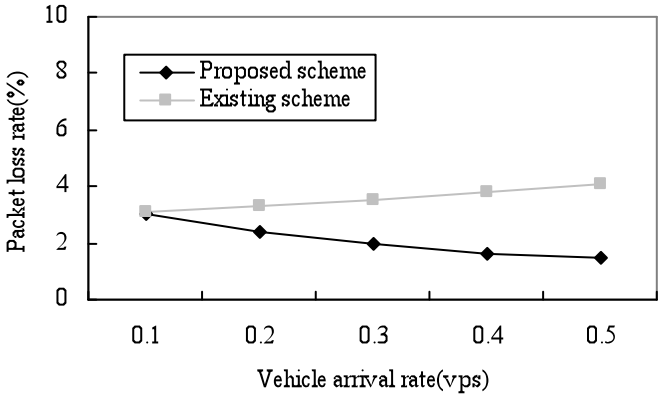


Figure15 Packet loss rate based on vehicle arrival rate

With the increase in vehicle arrival rate, the address configuration cost in the proposed protocol decreases, as shown in Figure 13. Therefore, when the vehicle arrival rate increases, the packet loss rate reduces. In the existing protocol, the vehicle arrival rate has no relation to the average address configuration cost, so the cost is steady. When the vehicle arrival rate increases, the total traffic also grows. Since an entering vehicle obtains an address from a vehicle multi-hop away, the growth in the traffic degrades the network performance. As a result, the packet loss rate increases, as shown in Figure 15.

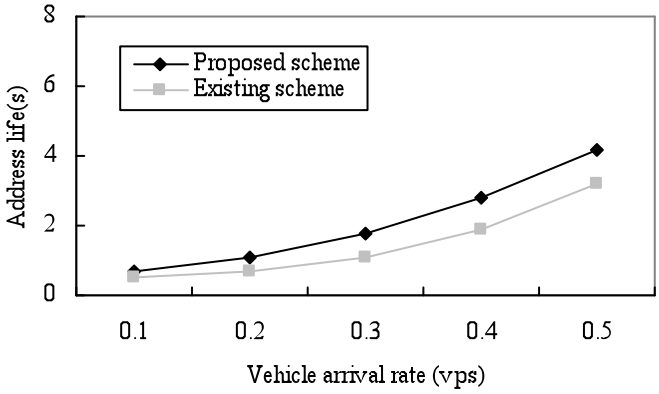


Figure16 Address life based on vehicle arrival rate

In this protocol, with the increase in vehicle arrival rate, the probability of a vehicle acquiring an address in S1 and S2 grows. As a result, the address life is prolonged. In the existing protocol, when the vehicle arrival rate increases, the number of intermediate vehicles staying between an entering vehicle and a leaving vehicle also grows. Therefore, the probability of an entering vehicle acquiring an address from an intermediate vehicle keeping a leaving vehicle's address also increases. As a result, the address life is increased.

6 Conclusion and future work

This paper proposes a secure address configuration protocol for IPv6-based vehicular networks. In this protocol, a vehicle obtains a unique address from a neighbor node without DAD, and the address space can be automatically reclaimed for reassignment. Therefore, the cost and delay are reduced. Moreover, the control messages are transmitted within one-hop scope, so the scalability is achieved. The identification of a vehicle can be authenticated, so the security is ensured.

This protocol assumes that the distance between a vehicle and its serving AP is one-hop. In our future work, we plan to study the secure IPv6 address configuration for vehicular networks where the distance between a vehicle and its serving AP is multi-hop.

Acknowledgements

This work is supported by National Natural Science Foundation of China (61202440).

References

[1] Maazen Alsabaan, Waleed Alasmary, Abdurhman Albasir,Kshirasagar Naik. Vehicular Networks for a Greener Environment: A Survey. IEEE Communications Surveys & Tutorials, 2013, 15(3): 1372-1388.

[2]Thomson S, Narten T, Jinmei T. IPv6 Stateless Address Autoconfiguration, IETF RFC 4862,2007.

[3]N. Moore. Optimistic Duplicate Address Detection (DAD) for IPv6. IETF RFC 4429. 2006.

[4]Uttam Ghosh, Raja Datta. A secure dynamic IP configuration scheme for mobile ad hoc networks. Ad hoc networks. 2011,9(7): 1327–1342.

[5]Hongbo Zhou, Matt W.Mutka, and Lionel M.Ni. Secure prophet address allocation for MANETs. Security and communication networks,2010,3(1):31-43.

[6] Xiaonan Wang, Yi Mu. A secure IPv6 address configuration scheme for a MANET. *Security and communication networks*. 2013, 6(6):777–789.

[7]Sonia Mettali Gammar, Elabidi Amine,Farouk Kamoun. Distributed address auto configuration protocol for Manet networks. Telecommun Syst. 2010, 44(1-2): 39-48.

[8] Xiaonan Wang, Huanyan Qian. Cluster-based and distributed IPv6 address configuration scheme for a MANET. Wireless personal networks.2013, 71(4):3131-3156.

[9]Syed Rafiul Hussain, Subrata Saha, Ashikur Rahman. SAAMAN: Scalable Address Autoconfiguration in Mobile Ad Hoc Networks. Journal of Network and Systems Management.2011, 19(3):394-426.

[10]Wang Xiaonan, Zhong Shan. An IPv6 address configuration scheme for wireless sensor networks based on location information. Telecommunication Systems. 2013, 52(1):151–160.

[11]Hyojeong Shin, Elmurod Talipov, Hojung Cha.Spectrum: Lightweight Hybrid Address Autoconfiguration Protocol Based on Virtual Coordinates for 6LoWPAN.IEEE Transactions on mobile computing, 2012, 11(11):1749-1761.

[12]Elmurod Talipov, Hyojeong Shin, Seungjae Han,etc. A lightweight stateful address autoconfiguration for 6LoWPAN. Wireless Network, 2011, 17(1): 183-197.

[13]Mamoun F. Al-Mistarihi, Mohammad Al-Shurman, Ahmad Qudaimat. Tree based dynamic address autoconfiguration in mobile ad hoc networks. Computer Networks, 2011, 55(8): 1894–1908.

[14]Mohandas, B.K., Liscano, R. IP Address Configuration in VANET using Centralized DHCP. In: 33rd IEEE Local Computer Networks Conference, 2008, pp. 608–613.

[15]Shu-Jun Chao,Jia-Ming Zhang,Chiu-Ching Tuan. Hierarchical IP distribution mechanism for VANET. 2010 Second International Conference on Ubiquitous and Future Networks (ICUFN), 2010, pp.349 -354.

[16]Y.-S. Chen, C.-H. Cheng, C.-S. Hsu, G.-M. Chiu, Network mobility protocol for vehicular ad hoc networks, in: Wireless Communications and Networking Conference, WCNC, 2009, pp. 1-6.

[17]Yuh-Shyan Chena,Chih-Shun Hsu, Wei-Han Yi.An IP passing protocol for vehicular ad hoc networks with network fragmentation. Computers and Mathematics with Applications. 2012, 63(2): 407–426.

[18]Dedicated short range communications (DSRC) message set dictionary. Draft, SAE J2735,2008

[19]S Kim, J Chung. Message Complexity Analysis of Mobile Ad Hoc Network Address Auto-configuration Protocols, IEEE Transactions on Mobile Computing, 2008, 7(3): 356-371.

[20]K. Fall, K. Varadhan, The ns manual, www.isi.edu/nsnam/ns/nsdocumentation.html

[21]X.N Wang, S Zhong. All-IP communication between wireless sensor networks and IPv6 networks based on location information. Computer Standards & Interfaces, 2013, 35(1): 65–77.