IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 4 August 2022, accepted 22 August 2022, date of publication 25 August 2022, date of current version 1 September 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3201520

RESEARCH ARTICLE

A Vehicular Mobility Management Scheme for a Shared-Prefix Model Over IEEE WAVE IPv6 Networks

MYEONGJI KO, HYOGON KIM, (Member, IEEE), AND SUNG-GI MIN
Department of Computer Science and Engineering, Korea University, Seoul 02841, Republic of Korea

Corresponding author: Sung-Gi Min (sgmin@korea.ac.kr)

This work was supported by the National Research Foundation of Korea (NRF) Grant through the Korea Government [Ministry of Science and ICT (MSIT)] under Grant 2020R1A2C3011888.

ABSTRACT The IETF IPWAVE Working Group is developing an IPv6-based solution to establish connectivity for V2V and V2I communication. It proposes a vehicular mobility management scheme based on PMIPv6. In this scheme, a shared-prefix model is proposed for its vehicular link model. The shared-prefix model is to share a common prefix among several RSUs. When using it, a vehicle in the same prefix domain does not need to change its IPv6 address even if its serving RSU is changed. So its CNs do not notice the movement of the vehicle. This concept is very similar with PMIPv6, because PMIPv6 also hides the vehicle's movement to its CNs if PMIPv6 does not use the route optimization. However, current IPWAVE draft of vehicular mobility management has several problems. Firstly, it requires a vehicular neighbor discovery, which is not compatible with standard IPv6 NDP. Secondly, PMIPv6 assumes that it acquires a MN-ID and a profile of a mobile node during the network attachment, but there is no authentication in IEEE WAVE networks. So, such information cannot be acquired in IEEE WAVE networks. We propose a vehicular mobility management scheme for a shared-prefix model over IEEE WAVE IPv6 networks. The proposed scheme introduces VMM-NDP module. It handles the path changes due to the serving RSU changes within a shared prefix domain. It also hides the IPv6 address change to CNs by maintaining the previous IPv6 address when the vehicle's current IPv6 address is changed due to the movement of the vehicle into new shared prefix domain. In this scheme, a vehicle can know its movement. So, there is no need to acquire a MN-ID. And also, since this scheme uses a shared-prefix model, there is no need to obtain a profile.

INDEX TERMS IEEE wireless access in vehicular environment (WAVE), WAVE service advertisement (WSA), vehicular mobility management (VMM), neighbor discovery protocol (NDP), proxy mobile IPv6 (PMIPv6).

I. INTRODUCTION

The IETF IP Wireless Access in Vehicular Environments (IPWAVE) Working Group (WG) is developing an IPv6-based solution to establish direct and secure connectivity for Vehicle-to-Vehicle (V2V) and Vehicle-to-Infrastructure (V2I) communication [1]. It proposes a new vehicular link model for its IPWAVE architecture called a shared-prefix model in which multiple Road Side Units (RSUs) in the same subnet share a common prefix. By using the shared-prefix model, when a vehicle passes through another RSU in the same subnet, it can use its current IPv6 address in the wireless coverage of the RSU. So, its Correspondent Nodes (CNs) do not notice the movement of the vehicle. In addition, since multiple RSUs share one prefix, this model can save prefixes. Therefore, it is quite attractive to apply the shared-prefix model to IEEE Wireless Access in Vehicular Environment (WAVE) networks.

The IPWAVE WG has proposed Vehicular Neighbor Discovery (VND) [2]. VND defines the shared-prefix model and takes into account Neighbor Discovery (ND) optimization that considers the wireless link properties such as asymmetric

The associate editor coordinating the review of this manuscript and approving it for publication was Rentao Gu.

reachability [3] and undetermined link-level connectivity [4]. However, since VND extends the standard Neighbor Discovery Protocol (NDP), it requires modifications of the standard NDP module.

The IPWAVE WG has also proposed Proxy Mobile IPv6 (PMIPv6) [5] based Vehicular Mobility Management (VMM) scheme [6]. VMM also uses the shared-prefix model. In VMM, a vehicular link is a subnet and consists of multiple sub-links within the subnet. A shared prefix is assigned to the vehicular link. VMM has proposed intra-link mobility management and inter-link mobility management. It also considers Distributed Mobility Management (DMM) [7].

However, there are problems in VMM because it has missed a basic assumption of PMIPv6. Comparing PMIPv6 and VMM, in PMIPv6, Mobile Access Gateway (MAG) assumes that Mobile Node's Identifier (MN-ID) and its profile are acquired during the mobile node attachment procedure. The MN-ID is typically an identifier such as a Network Access Identifier (NAI) [8] and the profile is used by discovering the Local Mobility Anchor (LMA) for a Mobile Node (MN). Unlike PMIPv6, VMM cannot discover such information because the vehicular link uses the Outside the Context of a Basic service set (OCB) [9] mode that does not have the authentication procedure. In addition, in PMIPv6, the LMA assigns a Home Network Prefix (HNP) to the MN during its attachment procedure. However, in VMM, a vehicle obtains its shared prefix via VND. After that, it configures its IPv6 address based on IPv6 Stateless Address Auto-Configuration (SLAAC) [10] and registers the IPv6 address using the address registration procedure defined in VND. So VMM does not need to assign the vehicle's IPv6 addresses. As a result, the address registration procedure may replace the attachment procedure. For these reasons, unnecessary procedures exist in VMM.

In this paper, we propose a vehicular mobility management scheme for a shared-prefix model over IEEE WAVE IPv6 networks. In this scheme, a vehicle uses WAVE Service Advertisements (WSAs) with WAVE Routing Advertisements (WRAs) [11] to discover default gateway information and a shared prefix assigned to a vehicular link. The proposed scheme introduces VMM-NDP module. VMM-NDP module is used for address auto-configuration and vehicular mobility management within a vehicular link and between vehicular links. VMM-NDP extends the validity of the currently used IPv6 address derived from the shared prefix assigned to the vehicular link when the vehicle moves to another vehicular link. It prevents changes of the IPv6 address used for on-going connections between CNs and the vehicle. VMM-NDP is also used to track changes of the data path within the vehicular link and between vehicular links. In addition, it mitigates the workload of vehicular links because it does not use any link-level multicast. By doing so, wireless communication efficiency will be improved as more vehicles can use the vehicular links at the same time.

The rest of this paper is organized as follows. Section 2 presents related work. Section 3 describes problems of IETF IPWAVE VND and VMM in detail. Section 4 describes the network architecture of the proposed mobility management scheme. Section 5 presents the procedure of the proposed scheme in detail. Section 6 presents comparative analysis between the proposed scheme and existing mobility schemes. Section 7 presents security considerations of the proposed scheme.Section 8 presents the simulation result. Finally, Section 9 concludes this paper.

II. RELATED WORK

There have been several proposals [12], [13] [14], [15] [16] for mobility management in the WAVE network. We described only the two papers [12], [13] which are the most relevant to the proposed scheme.

Vehicular IP in WAVE (VIP-WAVE) [12] addresses the problems of IP-based V2I communications. VIP-WAVE defines IP configuration and mobility management scheme supported by PMIPv6 to WAVE network. In this scheme, when a vehicle detects its movement, it forms its movement to a network entity by sending a unicast Router Solicitation (RS). This scheme becomes the base vehicular mobility management scheme proposed in IETF IPWAVE VND and VMM drafts. However, VIP-WAVE has a waste of prefixes because, unlike the proposed scheme using the shared-prefix model, it allocates a prefix per vehicle.

A location estimation-based mobility management mechanism [13] is based on WAVE Short Message Protocol (WSMP). It utilizes positioning systems for WAVE network. In this proposal, the location of vehicle is stored in the location server using Location Area Identifier (LAI). The architecture of this scheme assumes that there are four RSU, RSU1 and RSU2 belong to the same location area, and RSU3 and RSU4 belong to another location area. One location server manages multiple location areas and RSUs register vehicles with LAI within their coverage to the location server. The location server also handles vehicular mobility management. When the location server delivers messages to a vehicle, it uses broadcasting to all RSUs belonging to the LAI because it does not know the specific RSU at which the destination vehicle is currently attached. However, this method wastes channel bandwidth because whenever the location server delivers messages, it broadcasts the messages to all RSUs. In contrast, our proposed scheme does not use any multicasting. Also, this mechanism is a mobility management for WSMP, not for IPv6.

III. IETF IPWAVE VEHICULAR NEIGHBOR DISCOVERY AND VEHICULAR MOBILITY MANAGEMENT

The IPWAVE architecture [2], [6] is shown in Fig.1.

There is a Traffic Control Center (TCC) in a vehicular cloud in the Internet. The TCC hosts Mobility Anchors (MAs). This architecture introduces a new vehicular link model called a shared-prefix model. A vehicular link represents a subnet and a shared prefix is assigned to the subnet. Each MA is responsible for managing the mobility of vehicles in the subnet. In the Internet, the shared prefix is used to

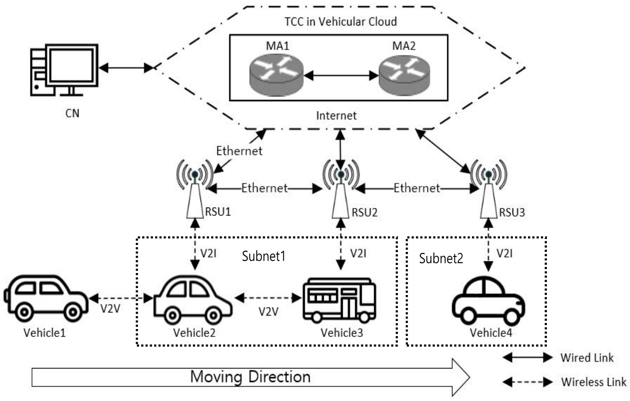


FIGURE 1. **IPWAVE architecture.**

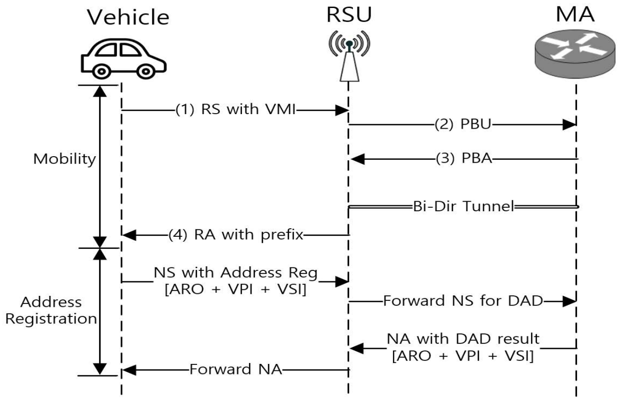


FIGURE 2. **Message flow of a vehicle's network attachment.**

deliver packets to the MA that manages the subnet to which the shared prefix is assigned. RSUs are interconnected to MAs via a wired network.

A vehicular network is a wireless network consisting of multiple RSUs and vehicles. A vehicular network divides into multiple subnets. A subnet may include multiple RSUs, and they share a common prefix. In Fig.1, a vehicular network consists of three RSUs and four vehicles. The subnet1 includes two RSUs (RSU1 and RSU2) and the subnet2 has one RSU (RSU3). Different prefixes are assigned to different subnets.

A. PROBLEMS OF NETWORK ATTACHMENT PROCEDURE

In this section, we describe the vehicle's network attachment procedure, and then explain the problems with this procedure. VND [2] and VMM [6] describe the network attachment procedure of a vehicle. Fig.2 shows the message flow of a vehicle's network attachment procedure. They assume that vehicles have their own digital road map, which includes RSU information. Using this map, the vehicle can figure out which RSU the vehicle will be attached to.

The network attachment procedure consists of two phases. In the first phase, a vehicle obtains a prefix of a RSU for address auto-configuration. The vehicle exchanges Router Solicitation (RS)/Router Advertisement (RA) messages with the RSU, and the RSU exchanges Proxy Binding Update (PBU)/Proxy Binding Acknowledgement (PBA) messages with the MA. After exchanging PBU/PBA messages, a bi-directional tunnel is created between the RSU and the MA. In the second phase, the vehicle performs address auto-configuration by combining its link-layer address with the prefix of its current vehicular link. Then it registers a new IPv6 address to the RSU by sending a Neighbor Solicitation (NS) message. The RSU forwards the NS to the MA for Duplicate Address Detection (DAD) [17]. VMM notes that the MA allocates a unique IPv6 address to the vehicle during multihop-DAD-based registration.

However, the Mobility phase (first phase) in Fig.2 is totally redundant at the network attachment procedure due to the following problems.

- PBU/PBA exchange (messages #2 and #3 in Fig. 2) is redundant. In PMIPv6, when a LMA receives a PBU, it allocates a HNP to the vehicle and creates a Binding Cache Entry (BCE). Then a bi-directional tunnel between the MAG and the LMA is established if there is no bi-directional tunnel between them. However, in VMM, there is no HNP allocation and a BCE cannot be created due to the lack of the vehicle's identifier at this moment. The BCE can be established during the address registration phase (second phase) with its new IPv6 address which the vehicle is registering. Also, the bi-directional tunnel already exists with a high probability as it is shared by all vehicles attached to the RSU. If not, it can be established at the address registration phase (second phase), as there are no data packets for the vehicle at this moment.
- RS/RA exchange (messages #1 and #4 in Fig. 2)is also redundant because the vehicle already knows its RSU and the RSU advertises the shared prefix instead of the vehicle's HNP.
- There is no mention of how to acquire the mandatory MN-ID and the profile of a vehicle during the network attachment. In PMIPv6, it assumes that the MAG acquires the MN-ID and profile during a network attachment event. However, VMM cannot discover such information during the attachment procedure since vehicular link using OCB mode does not perform authentication during the network attachment. In addition, if the vehicle wants to use a pseudonym for its privacy, it is more hard to obtain the MN-ID of the vehicle because the real MN-ID of the vehicle is hidden.
- There is a conflict in the vehicle's IP address configuration. The vehicle configures its IPv6 address based on SLAAC and registers the IPv6 address with the MA. However, VMM explicitly notes that the MA allocates a unique IPv6 address (128bits) to the vehicle. The address registration replaces DAD, so the registering address is unique if no address conflict is detected on the MA. Therefore, the MA's address allocation is not required.

- VMM uses a link-layer address as a Interface Identifier (IID) for the vehicle's IPv6 address generation, which is not recommend by [18].

B. PROBLEMS OF MOBILITY MANAGEMENT PROCEDURE

VMM describes two types of mobility management procedures: One is mobility management within a vehicular link, and the other is mobility management between multiple vehicular links. It also describes DMM case for each type of mobility management, but we do not discuss it here because it is beyond the scope of this paper. Mobility management problems is divided into two types.

1) PROBLEMS OF MOBILITY MANAGEMENT WITHIN A VEHICULAR LINK

In this section, we describe mobility management procedure within a vehicular link (intra-link handover), and then explain the problems with this procedure. As several RSUs are located at a vehicular link, a vehicle may change its point of attachment within the vehicular link. There are two cases of choosing a new RSU for the vehicle to attach to. In the first case, when a current RSU (denoted as c-RSU) detects that the vehicle is leaving, it de-registers the vehicle at the MA by sending a de-registration PBU message. The MA may select a new RSU (denoted as n-RSU) based the digital road map with the vehicle's trajectory. In this case, the MA changes directly the end-point of the tunnel to the n-RSU for the vehicle. In the second case, the n-RSU may detect the vehicle using periodically broadcasted NS messages from the vehicle. In this case, it exchanges PBU/PBA messages with the MA.

However, if the MA's prediction for the vehicle's trajectory is inaccurate due to the vehicle sudden change of its direction, the tunnel may be mis-configured. Also, in the second case, handover delay may be increased as the frequency of periodic NS message is decreased due to the wireless channel congestion.

Because of these issues, it is necessary to propose a new vehicular mobility management scheme.

2) PROBLEMS OF MOBILITY MANAGEMENT BETWEEN MULTIPLE VEHICULAR LINKS

In this section, we describe mobility management procedure between multiple vehicular link (inter-link handover), and then explain the problems with this procedure. When the vehicle moves from a vehicular link to another vehicular link, the vehicle maintains the current IPv6 address in new vehicular link. The c-RSU de-registers the vehicle at the current MA (MA1) by sending a de-registration PBU message. The MA1 predicts new MA (MA2) using vehicle's trajectory, then notifies MA2 that the vehicle moves into the n-RSU which belongs to the MA2. Then the MA2 notifies n-RSU by sending PBA message. The MA2 and the n-RSU establish a bi-directional tunnel. The PBU/PBA messages carry the vehicle's context information for the previous link. The n-RSU sends the unicast RA message for the previous vehicular link to the vehicle when the vehicle enters new vehicular link. The vehicle also performs the network attachment procedure for new vehicular link. The vehicle can maintain the IPv6 address derived from the previous shared prefix that the vehicle is used.

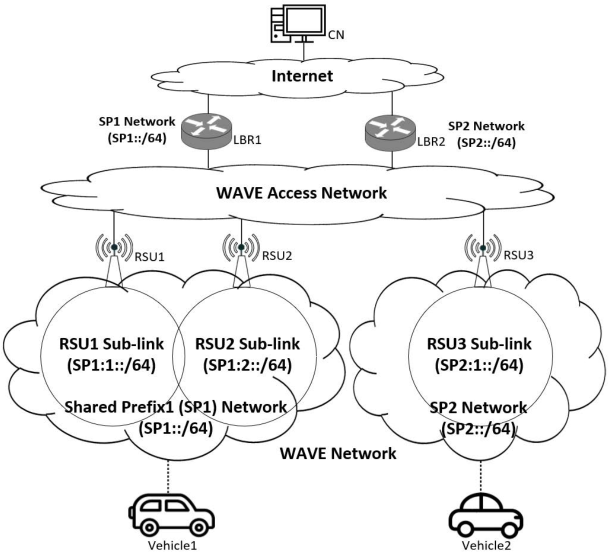


FIGURE 3. The network architecture of the proposed scheme.

However, there are following problems in this procedure.

- If the prediction is inaccurate, connections between the vehicle and CNs are disrupted. Unlike the intra-link handover, there is no reactive handover using PBU/PBA exchanges between n-RSU and MA2. As a result, connections disruption due to inaccurate prediction cannot be recovered.
- There is no description about how to carry the context information in PBU/PBA messages and how to route packets between MA1 and MA2. VMM must clarify the data path between MA1 and n-RSU. In PMIPv6, single LMA serves its MN and there is no tunneling between LMAs.

IV. THE NETWORK ARCHITECTURE OF THE PROPOSED SCHEME

From now on, we describe about our proposed vehicular mobility management scheme. In this section, we explain the network architecture of the proposed scheme. The message exchange flow of the proposed scheme is described in section V.

The network architecture of the proposed vehicular mobility management scheme is shown in Fig.3.

The architecture also uses a shared-prefix model. In the proposed scheme, a shared prefix is a 64-bit network prefix consisting of a 48-bit global routing prefix and a 16-bit subnet identifier. It is assigned to a vehicular link. A vehicular link consists of multiple sub-links, and each with the same global routing prefix and a unique subnet identifier. A sub-link represents a wireless link of a RSU. The subnet identifier of the vehicular link is zero, while the subnet identifiers of sub-links have non-zero values. The subnet identifier of the sub-link is used to route packets within the WAVE access network. The WAVE network consists of several vehicular links. Each vehicular link owns a shared prefix and is managed by a router in the WAVE access network called a Link Boarder Router (LBR). The LBR advertises the 64-bit shared prefix, assigned to its vehicular link, into the Internet.

For example, in Fig.3, there are two LBRs (LBR1 and LBR2). LBR1 represents a vehicular link, of which the 64-bit shared prefix is SP1::/64. The vehicular link has two sub-links. One represents a wireless link of RSU1 with SP1:1/64, and the other represents that of RSU2 with SP1:2/64. LBR2 represents another vehicular link, of which its prefix is SP2::/64. It has a single sub-link. The sub-link represents a wireless link of RSU3 with SP2:1/64. Note that sub-link prefixes are never advertised into the WAVE network. They are used only in the WAVE access network internally.

The major components of the proposed scheme are as follows:

- Vehicle
- Road Side Unit (RSU)
- The WAVE access network
- Link Border Router (LBR)
- Correspondent Node (CN)

A. VEHICLE

A vehicle is a mobile node which may want to use a service in the Internet or want to communicate with other vehicles. It includes an IPv6 configuration module to initialize its IPv6 module. The IPv6 configuration module configures new IPv6 addresses for its WAVE interfaces when it detects a new vehicular link in the WAVE network.

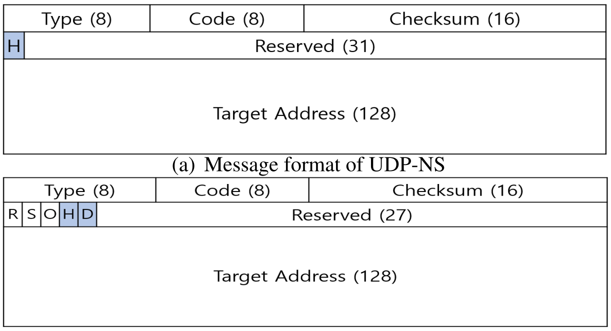
It also includes VMM-NDP module. VMM-NDP module performs address registration procedure and handover procedure. The address registration procedure replaces the DAD which is defined in [10] because it detects whether there are duplicate IPv6 addresses to be registered.

B. ROAD SIDE UNIT (RSU)

A RSU is a router which connects the WAVE network to the WAVE access network. It serves as a default gateway for vehicles within its wireless coverage area in the WAVE network. It periodically broadcasts WSAs into the WAVE network. Each WSA contains one WRA.

A RSU includes VMM-NDP module. It performs the address registration procedure for new IPv6 addresses generated by vehicles in its wireless coverage area. To perform the address registration procedure, it maintains a Registered Vehicle List (RVL) called a RSU-RVL. Each entry contains a tuple of <a registered IPv6 address, a link-layer address, a status>.

If a RSU supports inter-link mobility, it has an inter-link handover list. Each entry contains a pair of <a shared prefix of a vehicular link, a LBR's IPv6 address of vehicular link>.



(a) Message format of UDP-NS

(b) Message format of UDP-NA

FIGURE 4. Message format of NS and NA with new flags.

C. THE WAVE ACCESS NETWORK

The WAVE access network is an IPv6 network, which connects RSUs and LBRs in the proposed scheme.

D. LINK BORDER ROUTER (LBR)

A LBR is a router which represents a vehicular link to the Internet. It includes VMM-NDP module to perform the address registration procedure for IPv6 addresses generated by vehicles within its vehicular link. The LBR maintains a RVL list (LBR-RVL) to detect whether a newly generated IPv6 address is duplicated with an registered address in the RVL list. Each entry contains a tuple of <a registered IPv6 address, a RSU's IPv6 address, a prefix of RSU's sub-link>. The list also is used to forward a data packet to deliver the correct RSU of the vehicle which is the destination of the data packet.

If a LBR supports inter-link mobility, it maintains an inter-link handover prefix list. The list includes 64-bit shared prefixes assigned to other vehicular links in the WAVE network. It is used to check whether inter-link handover for a given shared prefix is supported by the LBR.

E. CORRESPONDENT NODE (CN)

A CN is an IPv6 host in the Internet and it communicates with a vehicle in the WAVE network.

V. THE PROPOSED VEHICULAR MOBILITY MANAGEMENT SCHEME

In this section, we describe the flow of our proposed scheme. The proposed scheme introduces VMM-NDP. VMM-NDP is used for address registration and handover procedure. VMM-NDP uses User Datagram Protocol (UDP) encapsulation to separate its NS/Neighbor Advertisement (NA) messages from standard NDP messages which use Internet Control Message Protocol version 6 (ICMPv6) encapsulation. It also combines multiple NS/NA messages into an UDP datagram to handle multiple address registration and/or handover procedure simultaneously. We call this an UDP-NS/NA datagram. VMM-NDP uses the standard NS/NA format defined in [3] with some new flags. Fig. 4 shows the message format of UDP-NS/NA with new flags.

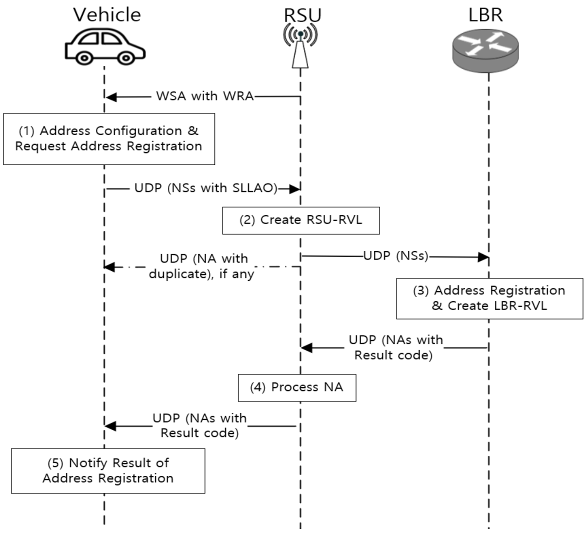

Vehicle
RSU
LBR
WSA with WRA
(1) Address Configuration & Request Address Registration
UDP (NSs with SLLAO)
(2) Create RSU-RVL
UDP (NA with duplicate), if any
UDP (NSs)
(3) Address Registration & Create LBR-RVL
UDP (NAs with Result code)
(4) Process NA
UDP (NAs with Result code)
(5) Notify Result of Address Registration

FIGURE 5. Message flow of the network attachment.

A 'Handover (H)' flag is defined in the first bit of the reserved field in the standard NS message. A 'Handover (H)' flag and a 'De-registration (D)' flag are defined in the next two bits of the 'override (O)' flag in the standard NA message. If the 'H' flag sets in the UDP-NS datagram, the NS message notifies the arrival of the vehicle that owns the target address in the UDP-NS datagram. If the 'H' and 'D' flags set in the unsolicited UDP-NA datagram, the unsolicited NA message notifies the departure of the vehicle that owns the target address in the unsolicited UDP-NA datagram.

A. NETWORK ATTACHMENT PROCEDURE

When a vehicle enters the WAVE network for the first time, it must perform the network attachment procedure. Fig.5 and Algorithm1 shows the network attachment procedure. In the algorithm, the IPv6 configuration module is denoted as 'Conf', VMM module is denoted as 'Vmm'.

Step 1) The RSU sends a WSA with a WRA periodically. The WSA includes the RSU's 3D geographic position. The WRA includes the shared prefix of the vehicular link (SP1::/64), the RSU's link-local IPv6 address of the vehicular link and the RSU's link-layer address **(Line #1 of algorithm)**. When the vehicle receives the WSA, it notifies the IPv6 configuration module. The IPv6 configuration module and VMM-NDP module configure its WAVE interface as follows:

- **(Line #2 of algorithm)** The IPv6 configuration module adds a Destination Cache Entry (DCE) for the RSU's link-local IPv6 address. It also adds a Neighbor Cache Entry (NCE) with the RSU's link-local IPv6 address and the RSU's link-layer address from the WRA of the received WSA. These operations prevent standard NDP operation when the vehicle sends an NS message to the RSU.
- **(Line #3 of algorithm)** The IPv6 configuration module generates IPv6 addresses according to the stateless address auto-configuration procedure. It generates a 64-bit IID for its link-local IPv6 address using the recommended IID generation method defined in [19]. The generated IID is called a stable IID. "fe80::/64" and the newly generated IID are combined to generate the vehicle's link-local IPv6 address for the vehicular link. In addition, another new stable IID is generated for the shared prefix (SP1::/64) included in the WRA. The shared prefix and the stable IID are concatenated to generate a new public IPv6 address for the vehicle.
- The IPv6 configuration module requests address registration procedure for both IPv6 address to VMM-NDP module.
- **(Line #4 of algorithm)** Vehicle's VMM-NDP module creates one NS message for each IPv6 address. The NS message includes a source link-layer address option

Algorithm 1 Network Attachment Procedure

```
Require: A vehicle receives a WSA with a WRA
 1: $gw\_ipAddr \Leftarrow wsa.GetWRA().GetGwIpAddr()$
    $gw\_llAddr \Leftarrow wsa.GetWRA().GetGwLlAddr()$
    $shared\_prefix \Leftarrow wsa.GetWRA().GetPrefix()$
    $dad\_count = 0$
 2: $vConf.AddDCE(gw\_ipAddr)$
    $vConf.AddNCE(gw\_ipAddr, gw\_llAddr)$
 3: $v\_ip \Leftarrow SlaacStableIid(shared\_prefix, dad\_count)$
 4: Send $UdpNs(v\_ip, v\_llAddr)$ to RSU
 5: RSU receives $UdpNs$ from Vehicle
 6: if $rsuVmm.FindRVL(v\_ip)$ then
        Send $UdpNa(duplicated)$ to vehicle($ff02 :: 1$)
 7: else
        $rsuVmm.CreateRVL(v\_ip, v\_llAddr, incomplete)$
        Send $UdpNs(v\_ip)$ to LBR
 8: end if
 9: LBR receives $UdpNs$ from RSU
10: if $lbrVmm.FindRVL(v\_ip)$ then
        Send $UdpNa(duplicated)$ to RSU
11: else
        $lbrVmm.CreateRVL(v\_ip, rsu\_ip, rsu\_prefix)$ Send $UdpNa(success)$ to RSU
12: end if
13: RSU receives $UdpNa$ from LBR
14: if $UdpNa.GetCode() == success$ then
        $rsuVmm.SetRVLStatus(v\_ip, complete)$
        Send $UdpNa(success)$ to vehicle($ff02 :: 1$)
15: else
16:     $rsuVmm.DeleteRVL(v\_ip)$
        Send $UdpNa(duplicated)$ to vehicle($ff02 :: 1$)
17: end if
18: Vehicle receives $UdpNa$ from RSU
19: $vVmm.NotifyToConf(udpNa.GetCode())$
20: if $udpNa.GetCode() == success$ then
        $SetPreferredAddress(v\_ip)$
21: else
        $dad\_count$ += 1 and Go to Line 3
22: end if
```

(SLLAO) [3]. All created NS messages are encapsulated into an UDP-NS datagram. Since the WAVE interface does not have a valid IPv6 address yet, the source address of the UDP-NS datagram is set to an unspecified IPv6 address (::), and the destination address is set to the default gateway (RSU)'s link-local address. The UDP-NS datagram is sent to the default gateway.

Step 2) When RSU's VMM-NDP module receives the UDP-NS datagram, it processes each NS message independently as follows **(Line #5 of algorithm)**:

- **(Line #6 of algorithm)** For an NS message, it looks up a matching entry in its RSU-RVL for the target address. If so, VMM-NDP module generates an NA message with the code "duplicated". It combines all NA messages into an UDP-NA datagram if any NA message is generated, and sends it to the vehicle. Since the vehicle's registering IPv6 address is a tentative address, so the destination IPv6 address of the UDP-NA datagram uses all-node-multicast IPv6 address (ff02::1). Even though the destination IPv6 address of the UDP-NA datagram uses all-node-multicast IPv6 address, it use the link-layer unicast address in the SLLAO, instead of the link-layer address derived from the multicast address mapping defined in [20]. Such mapping of an IPv6 multicast address to a link-layer unicast address is defined in [21].
- **(Line #7-8 of algorithm)** Otherwise, it creates a RSU-RVL entry. The entry includes the target address in the NS message and the link-layer address in the SLLAO. The status of the entry is set to "INCOMPLETE". It also combines all NS messages into an UDP-NS datagram. The source IPv6 address of the UDP-NS datagram uses one of IPv6 addresses on the interface through which the original UDP-NS datagram is received from the vehicle. The destination IPv6 address must be the IPv6 address of the LBR, which manages the vehicular link to which the RSU's sub-link belongs. The UDP-NS datagram is sent to the LBR.

Step 3) When LBR's VMM-NDP module receives the forwarded UDP-NS datagram, it processes each NS message independently as follows **(Line #9 of algorithm)**:

- **(Line #10 of algorithm)** For an NS message, it looks up a matching entry in its LBR-RVL for the target address in the NS message. If it finds a matching entry, it generates an NA message with the code "duplicated". Then it combines all NA messages into an UDP-NA datagram, and forwards it to the RSU which sent the original UDP-NS datagram.
- **(Line #11-12 of algorithm)** Otherwise, it adds new entry to the LBR-RVL. The entry stores the registering IPv6 address, the source address of the UDP-NS datagram, and the prefix of the source address. It also generates an NA message with the code "success" for each registered IPv6 address. Then it combines all NA messages into an UDP-NA datagram, and forwards it to the RSU which sent the original UDP-NS datagram.

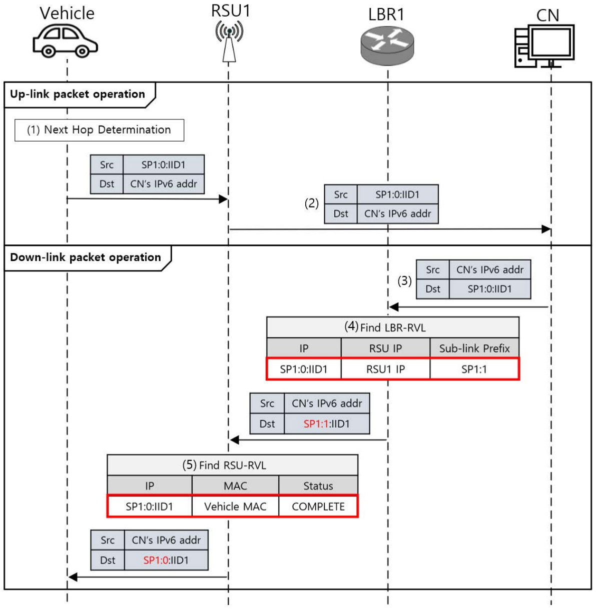


FIGURE 6. Message flow of V2I communication.

Step 4) **(Line #13-17 of algorithm)** When RSU'S VMM-NDP module receives the UDP-NA datagram, it processes each NA message independently. For an NA message, it checks the code of the NA message. If the code is "duplicated", it deletes the corresponding entry in the RSU-RVL. Otherwise, it sets the status of the entry to "COMPLETE". Then, it forwards the received UDP-NA datagram to the vehicle. It uses the same IPv6 address and link-layer address of the UDP-NA datagram, described at step 2.

Step 5) **(Line #18-22 of algorithm)** When vehicle's VMM-NDP module receives the UDP-NA datagram, it notifies the result of address registration for each registering IPv6 address to the IPv6 configuration module. If an IPv6 address is successfully registered, the IPv6 address becomes a preferred address. If address registration is rejected due to "duplicated", it regenerates another stable IID by increasing the DAD_Counter parameter value of the Pseudo Random Function (PRF). Then it restarts address registration for the newly generated IPv6 address.

B. V2I COMMUNICATION

After the network attachment procedure, the vehicle can communicate with a CN. The V2I communication message flow between a vehicle and a CN is shown in Fig.6 and the communication algorithm is shown in Algorithm2. In the algorithm, IPv6 Module is denoted as 'Ipv6' and Forwarding Module is denoted as 'Fwd'.

Step 1) **(Line #1-7 of algorithm)** A vehicle sends a packet to a CN. The IPv6 module in the vehicle performs next-hop determination for the destination address of the packet. Since DCE does not exist for the destination address, it performs on-link determination. The prefix list in the vehicle's IPv6 module is empty due to the undetermined link-level

Algorithm 2 V2I Communication Procedure

```
Require: A vehicle sends a data packet to a CN
 1: if $vIpv6.NhDeter(destAddr)$ then
        Forward the packet to next hop
 2:     else
 3:     if $vIpv6.OnlinkDeter(destAddr)$ then
            Forward the packet to the CN directly
 4:         else
 5:         $vIpv6.CreateDCE(destAddr)$
            Forward the packet to GW (RSU)
 6:     end if
 7: end if
 8: RSU forwards the packet to CN
 9: CN sends a packet to vehicle
10: LBR receives the packet from CN
11: if $lbrFwd.PrefixIsMine(destAddr)$ then
        $matchEntry \Leftarrow lbrFwd.FindRVL(destAddr)$
12:     if $matchEntry$ then
            $prefix \Leftarrow matchedEntry.GetPrefix()$
            $destAddr \Leftarrow$
            $lbrFwd.ReplacePrefix(prefix, destAddr)$
13:         Forwards the packet to its IPv6 Module
14:     else
            Drops the packet
15:     end if
16:     else
        Forwards the packet to its IPv6 Module
17: end if
18: RSU receives the packet from LBR
19: if $rsuFwd.PrefixIsMine(destAddr)$ then
        $destAddr \Leftarrow rsuFwd.ReplacePrefix(0, destAddr)$
        $matchEntry \Leftarrow rsuFwd.FindRVL(destAddr)$
20:     if $matchEntry$ then
            $destLlAddr \Leftarrow matchEntry.GetLlAddr()$
21:         Forwards the packet to $destLlAddr$
22:     else
            Drops the packet
23:     end if
24:     else
        Forwards the packet to its IPv6 Module
25: end if
```

connectivity property. The result of the on-link determination becomes "off-link". Then, it creates a DCE for the destination address and forwards the packet to the default gateway (RSU1).

Step 2) **(Line #8 of algorithm)** RSU1 forwards the packet to the CN using standard IPv6 forwarding procedure. Note that the subnet prefix of the source address of the packet belongs to the WAVE access network. No "Ingress filtering" problem occurs.

Step 3) **(Line #9 of algorithm)** The CN sends a packet to the vehicle.

Step 4) When LBR1 receives the packet, it processes it as follows **(Line #10 of algorithm)**:

- **(Line #11-15 of algorithm)** It checks whether the prefix of the destination address of the packet (SP1:0:IID1) matches the shared prefix assigned to its vehicular link. If so, it looks up a entry in the LBR-RVL with the destination address of the packet. If a matching entry exists, it replaces the prefix of the destination address with the shared prefix field (SP1:1::/64) of the matched entry. Then it forwards the packet to its IPv6 module. Otherwise, it drops the packet, because the destination vehicle is not attached to the vehicular link anymore.
- **(Line #16-17 of algorithm)** If they are not matched, it forwards the packet to the standard IPv6 module because the destination vehicle does not belong to the vehicular link managed by LBR1.

Step 5) When RSU1 receives the packet, it processes it as follows **(Line #18 of algorithm)**:

- **(Line #19-23 of algorithm)** It checks whether the prefix of the destination address of the packet (SP1:1:IID1) matches the prefix assigned to its sub-link. If so, it replaces the prefix of the destination address (SP1:1::/64) with the shared prefix of the vehicular link (SP1::/64). Then it looks up the RSU-RVL with the destination address of the packet. If there is a matching entry, it obtains the link-layer address from the link-layer address field of the matched entry and forwards the packet directly to the destination vehicle via its WAVE interface. Otherwise, it drops the packet because the destination vehicle is not attached to the RSU anymore.
- **(Line #24-25 of algorithm)** If they are not matched, it forwards the packet to its standard IPv6 module because the destination vehicle does not belong to the sub-link managed by RSU1.

C. VEHICULAR MOBILITY MANAGEMENT FOR V2I COMMUNICATION

When a vehicle changes its attachment point, vehicular mobility management handles this case. Depending on whether the vehicle is attached to a RSU in the same vehicular link or not, vehicular mobility management is divided into the following two cases:

- vehicular mobility management within a vehicular link
- vehicular mobility management between vehicular links

1) VEHICULAR MOBILITY MANAGEMENT WITHIN A VEHICULAR LINK

When a vehicle changes its point of attachment to another RSU in the same vehicular link, the vehicle notifies it to the current LBR via the new RSU. Fig.7 and Algorithm3 shows the flow of intra-link mobility management procedure. In Fig.7 and Algorithm3, the vehicle changes its point of attachment from RSU1 to RSU2 and the current LBR is LBR1. In the algorithm, the IPv6 configuration module is denoted as 'Conf', VMM module is denoted as 'Vmm'.

Step 1) **(Line #1-3 of algorithm)** The vehicle receives a WSA with a WRA from RSU2. It forwards the WSA to

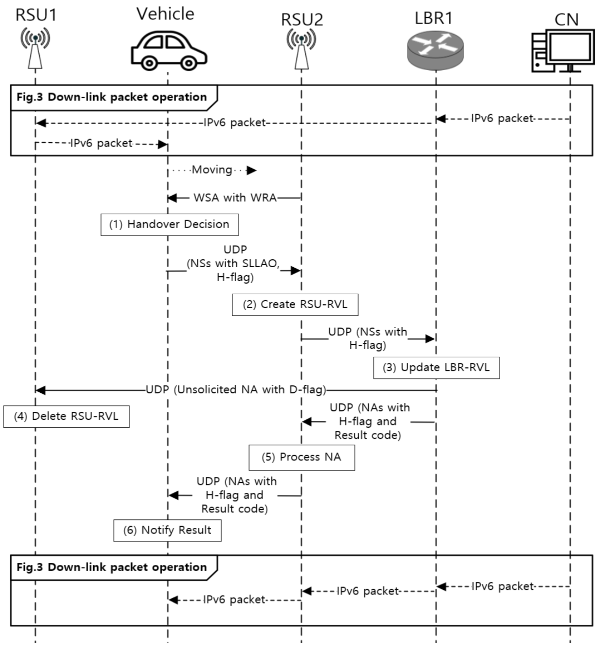


FIGURE 7. Message flow of mobility management within a vehicular link (Intra-link).

its IPv6 configuration module. When the module decides to change its point of attachment from RSU1 to RSU2, it adds a new DCE and NCE for RSU2. Then it notifies its handover decision to VMM-NDP module. VMM-NDP module generates an NS message with the 'H' flag set for each IPv6 address assigned to its WAVE interface, and the NS message includes a SLLAO. Then it encapsulates all NS messages into an UDP-NS datagram and sends it to RSU2. The source address of the UDP-NS datagram must be the link-local address of the WAVE interface.

Step 2) When RSU2 receives the UDP-NS datagram, it processes each NS message independently as follows **(Line #4 of algorithm)**:

- **(Line #5 of algorithm)** For each NS message with the 'H' flag set, it looks up a matching entry in its RSU-RVL. If there is a matching entry, VMM-NDP module generates an NA message with the code "duplicated". Then it combines all NA messages into the UDP-NA datagram if any NA message is generated and sends it to the vehicle.
- **(Line #6-7 of algorithm)** Otherwise, it creates a RSU-RVL entry for the target address in the NS message and the link-layer address in the SLLAO. Then, it forwards the NS message with the 'H' flag set to the LBR1, which manages the prefix of the target IPv6 address in the NS message.

Step 3) When LBR1 receives an UDP-NS datagram, it processes each NS message independently as follows **(Line #8 of algorithm)**:

Algorithm 3 Intra-Link Mobility Management Procedure

```
Require: A vehicle receives a WSA from a RSU2
 1: $gw\_ipAddr \Leftarrow wsa.GetWRA().GetGwIpAddr()$
    $gw\_llAddr \Leftarrow wsa.GetWRA().GetGwLlAddr()$
    $shared\_prefix \Leftarrow wsa.GetWRA().GetPrefix()$
    $dad\_count = 0$
 2: $vConf.AddDCE(gw\_ipAddr)$
    $vConf.AddNCE(gw\_ipAddr, gw\_llAddr)$ $my\_prefix \Leftarrow$ $vConf.GetPrefix()$
 3: if $my\_prefix == previous\_prefix$ then
      Send $UdpNs(v\_ip, v\_llAddr, H)$ to RSU2
 4:   RSU2 receives $UdpNs$ from vehicle
 5:   if $(rsuVmm.FindRVL(v\_ip))$ && $(UdpNs.GetHFlag() == 1)$ then
        Send $UdpNa(duplicated)$ to vehicle
 6:   else
        $rsuVmm.CreateRVL(v\_ip, v\_llAddr, incomplete)$
        Send $UdpNs(v\_ip, H)$ to LBR1
 7:   end if
 8:   LBR1 receives $UdpNs$ from RSU2
 9:   $src\_prefix \Leftarrow lbrVmm.GetPrefix(srcAddr)$
10:   if $(src\_prefix == shared\_prefix)$ && $(UdpNs.GetHFlag() == 1)$ then
11:     $matched\_entry \Leftarrow lbrVmm.FindRVL(v\_ip)$
12:     if $matched\_entry$ then
          $pRsu\_ip \Leftarrow lbrVmm.GetRsuIp(matched\_entry)$
          $lbrVmm.UpdateRVL(v\_ip, rsu\_ip, rsu\_preifx)$
13:       Send $UdpNa(success, H)$ to RSU2
14:       Send $UdpNa(D)$ to $pRsu\_ip$ (RSU1)
15:     else
16:       Send $UdpNa(unknown, H)$ to RSU2
17:     end if
18:     RSU1 and RSU2 receive $UdpNa$ from LBR1
19:     if $UdpNa.GetDFlag() == 1$ then
          $rsuVmm.DeleteRVL(v\_ip)$
20:     end if
21:     if $UdpNa.GetCode() == success$ then
          $rsuVmm.SetRVLStatus(v\_ip, complete)$
          Forward $UdpNa(success)$ to vehicle
22:     else
23:       $rsuVmm.DeleteRVL(v\_ip)$
          Forward $UdpNa(unknown)$ to vehicle
24:     end if
25:     Vehicle receives $UdpNa$ from RSU2
26:     $vVmm.NotifyToConf(udpNa.GetCode())$
27:     if $udpNa.GetCode() == success$ then
          $SetValidAddress(v\_ip)$
28:     else
          $SetInvalidAddress(v\_ip)$
29:     end if
30:   end if
31: else
      Perform Inter-link mobility management
32: end if
```

- **(Line #9-10 of algorithm)** It extracts the shared prefix assigned to the vehicular link from the source address of the UDP-NS datagram. Then it checks whether the extracted prefix matches its shared prefix assigned to its vehicular link and whether the code is ‘H’ flag.
- **(Line #11-13 and #15-17 of algorithm)** If so, it looks up a matching entry in the LBR-RVL with the target address in each NS message. If so, it stores the RSU address field in the matching entry and updates the RSU’s prefix and IPv6 address fields of the matching entry with the source address of the UDP-NS datagram. Then it generates an NA message with the code “success” and ‘H’ flag set. Otherwise, it generates an NA message with the code “unknown” and ‘H’ flag set. Then it combines them into an UDP-NA datagram and forwards it to the sender of the UDP-NS datagram (RSU2).
- **(Line #14 of algorithm)** It also generates an unsolicited NA message for each target address in the NS message with the ‘D’ flag set. Then it combines them into an UDP-NA datagram. It sends it to the previous RSU (RSU1) using the stored RSU address saved at above.
- **(Line #31-32 of algorithm)** If not, inter-link mobility management handles this case (see Step 4 in section V-C2)

Step 4) (Line #18-20 of algorithm) When RSU1 receives the unsolicited UDP-NA datagram, it processes each unsolicited NA message independently. For each unsolicited NA message with ‘D’ flag set, it deletes the matching RSU-RVL entry in its RSU-RVL.

Step 5) (Line #21-24 of algorithm) When RSU2 receives the UDP-NA datagram, it processes each NA message independently. For each NA with the code “success”, it sets the status of the entry to “COMPLETE”. For each NA with the code “unknown”, it deletes the corresponding RSU-RVL entry. Then it forwards the UDP-NA datagram the vehicle.

Step 6) (Line #25-30 of algorithm) When the vehicle receives the UDP-NA datagram, it notifies the result of handover procedure for each IPv6 address to the IPv6 configuration module. If the result is “success” for an IPv6 address, it uses the IPv6 address continuously. Otherwise, the unknown IPv6 addresses must be invalided.

2) VEHICULAR MOBILITY MANAGEMENT BETWEEN VEHICULAR LINKS

When a vehicle changes its point of attachment to another RSU which belongs to another vehicular link, it performs the network attachment procedure for the new vehicular link. If it wants to use public IPv6 addresses derived from shared prefixes of the previous vehicular links, it also performs inter-link mobility management for that public IPv6 addresses. Fig.8 and Algorithm4 shows the flow of inter-link mobility management. In Fig.8 and Algorithm4, the vehicle changes its point of attachment from RSU2 which is managed by LBR1 to RSU3 which is managed by LBR2. In the algorithm, the IPv6 configuration module is denoted as ‘Conf’, VMM module is denoted as ‘Vmm’.

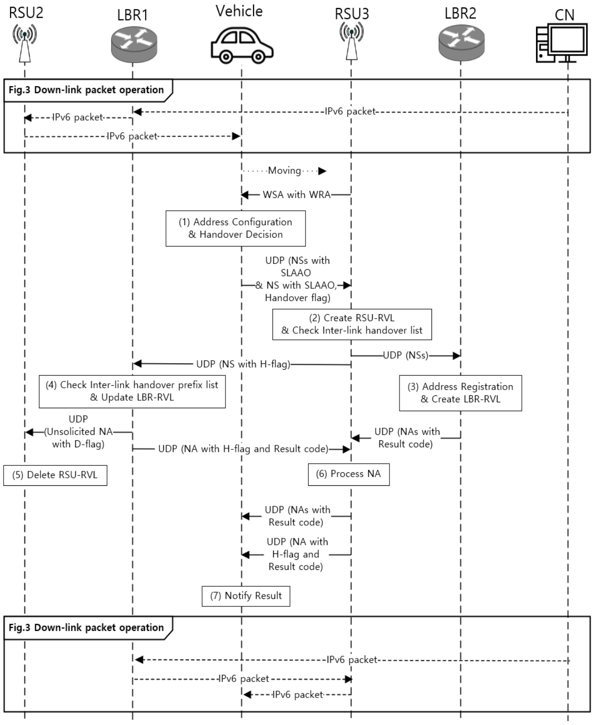

RSU2
LBR1
Vehicle
RSU3
LBR2
CN
Fig.3 Down-link packet operation
IPv6 packet
IPv6 packet
IPv6 packet
Moving
WSA with WRA
(1) Address Configuration & Handover Decision
UDP (NSs with SLAAO & NS with SLAAO, Handover flag)
(2) Create RSU-RVL & Check Inter-link handover list
UDP (NSs)
UDP (NS with H-flag)
(4) Check Inter-link handover prefix list & Update LBR-RVL
(3) Address Registration & Create LBR-RVL
UDP (Unsolicited NA with D-flag)
UDP (NA with H-flag and Result code)
UDP (NAs with Result code)
(5) Delete RSU-RVL
(6) Process NA
UDP (NAs with Result code)
UDP (NA with H-flag and Result code)
(7) Notify Result
Fig.3 Down-link packet operation
IPv6 packet
IPv6 packet
IPv6 packet

FIGURE 8. Message flow of mobility management between vehicular links (Inter-link).

Step 1) (Line #1-2 of algorithm) When the vehicle receives a WSA with a WRA with a different prefix than the previously received, the vehicle notifies it to the IPv6 configuration module.

The IPv6 configuration module and VMM-NDP module handle this case as follows:

- **(Line #3 of algorithm)** The IPv6 configuration module recognizes that the vehicle is attaching to the new vehicular link. It performs address configuration procedure explained in section V-A Step 1 for a new prefix.
- **(Line #4 of algorithm)** For each public IPv6 address in use which is derived from the previous prefix, VMM-NDP module generates an NS message with ‘H’ flag set and a SLLAO. These NS messages and the NS messages for the newly generated IPv6 address are combined into an UDP-NS datagram and sent to the new RSU (RSU3).

Step 2) When RSU3 receives the UDP-NS datagram, VMM-NDP module handles each NS message in the UDP-NS datagram as follows **(Line #5 of algorithm)**:

- **(Line #6 of algorithm)** For each NS message generated for the newly generated IPv6 address, it processes it as described in section V-A Step 2.
- **(Line #7-16 of algorithm)** For each NS message with ‘H’ flag set, it looks up the prefix of the target address in its inter-link handover list. If so, it handles the NS message as described in section V-C1 Step 2. Otherwise, it generates an NA message with the code “invalid prefix”. All generated NA messages are combined into an UDP-NA datagram and sent to the vehicle.

Algorithm 4 Inter-Link Mobility Management Procedure

```
Require: Continue with line #31 of Algorithm3
 1: if $my\_prefix == previous\_prefix$ then
      Perform Intra-link mobility management
 2:   else
 3:   $v\_ip2 \Leftarrow SlaacStableIid(shared\_prefix, dad\_count)$
 4:   Send $UdpNs(v\_ip, v\_llAddr, H), (v\_ip2, v\_llAddr)$ to
      RSU3
 5:   RSU3 receives $UdpNs$ from vehicle
 6:   if $UdpNs.GetHFlag() == 0$ then
        1emPerform network attachment for $v\_ip2$ with
        LBR2
 7:     else
 8:     $v\_prefix \Leftarrow rsuVmm.GetPrefix(v\_ip)$
 9:     if $rsuVmm.FindIlList(v\_prefix)$ then
10:       if $rsuVmm.FindRVL(v\_ip)$ then
            1emSend $UdpNa(duplicated)$ to vehicle
11:         else
12:         $rsuVmm.CreateRVL$
            1em$(v\_ip, v\_llAddr, incomplete)$
            1emSend $UdpNs(v\_ip, H)$ to LBR1
13:       end if
14:       else
          1emSend $UdpNa(invalid\_prefix)$ to vehicle
15:     end if
16:   end if
17:   LBR1 receives $UdpNs$ from RSU3
18:   $src\_prefix \Leftarrow lbrVmm.GetPrefix(srcAddr)$
19:   if $src\_prefix == shared\_prefix$ then
        1emPerform Intra-link mobility management
20:     else
21:     if $lbrVmm.FindIlpList(src\_prefix)$ then
          1emSame as line #11-14 of Algorithm3
22:       else
          1emSend $UdpNa(unknown, H)$ to RSU3
23:     end if
24:   end if
25:   RSU3 receives $UdpNa$ from LBR1
      1emSame as line #18-24 of Algorithm3
26:   Vehicle receives $UdpNa$ from RSU3
      1emSame as line #25-29 of Algorithm3
27: end if
```

Step 3) When the new LBR (LBR2) receives the UDP-NS datagram, it handles each NS message as described in section V-A Step 3.

Step 4) (Line #17 of algorithm) When the previous LBR (LBR1) receives the UDP-NS datagram, it processes each NS message as follows:

- **(Line #18-19 of algorithm)** It extracts the shared prefix assigned to the vehicular link from the source address of the UDP-NS datagram. Then it checks whether the extracted prefix matches its shared prefix assigned to its vehicular link. If so, it follows the section V-C1) Step 3.
- **(Line #20-21 of algorithm)** Otherwise, it checks whether the extracted prefix matches one of the shared prefixes in its inter-link handover prefix list. If so, it looks up the LBR-RVL with the target address of the NS message. Then it follows the operations after the lookup procedure of the LBR-RVL, described in section V-C1 Step 3.
- **(Line #22-24 of algorithm)** It sends an unsolicited NA message, as described in section V-C1 Step 3.

Step 5) When RSU2 receives the unsolicited UDP-NA datagram, the RSU processes it as described in section V-C1 Step 4.

Step 6) (Line #25 of algorithm) When RSU3 receives the UDP-NA datagram from its LBR (LBR2), it handles each NA message as described in section V-A Step 4. If it receives the UDP-NA datagram from another LBR (LBR1), it processes the UDP-NA datagram as described in section V-C1 Step 5.

Step 7) (Line #26-27 of algorithm) The vehicle handles the UDP-NA datagram as described in section V-A Step 5 or in section V-C1 Step 6, depending on 'H' flag value of each NA message.

VI. COMPARATIVE ANALYSIS

We perform comparative analysis between our scheme and existing mobility schemes. Table 1 shows the similarities and differences between them.

In our scheme, the vehicle can know its movement from WSAs sent by RSUs, and it initiates the handover procedure. So, it is a host-based mobility scheme. Similarly, MIPv6 is also a host-based mobility scheme in which MN detects its movements through RA messages sent by the MAG of the link periodically to which it belongs. On the other hand, PMIPv6 and IPWAVE VMM are network-based schemes in which the MAG/RSU that receives RS messages sent by the MN/vehicle periodically initiates the handover procedure.

There is a difference in IPv6 address and IID generation method. In MIPv6, PMIPv6 and our scheme, a vehicle gets a prefix from a RSU/MAG and generates its own IPv6 address based on SLLAC. However, in the case of IPWAVE VMM, a vehicle generates an IPv6 address based on SLLAC, then after, a MA allocates an IPv6 address to the vehicle once more. It is redundant for the MA to assign a unique IPv6 address to the vehicle because the vehicle already has an IPv6 address which is generated using SLAAC. Also, although our scheme and existing schemes both generate an IPv6 address based on SLAAC, there is a difference in the generation of IID. In MIPv6, PMIPv6, and IPWAVE VMM, IID is used as link-layer address. Using the link-layer address for IID is not recommended in [18], and since the link-layer address of the vehicle is revealed, pseudonym cannot be guaranteed. On the other hand, our scheme uses a stable IID. It is the recommended method in [18], and can guarantee pseudonym because vehicle's information is not revealed.

There is also a difference with assigning prefix. In MIPv6, a different prefix is assigned to each link, and in PMIPv6, a different prefix is assigned to each MN. Therefore, prefixes

TABLE 1. Comparative table between our scheme and existing schemes.

Features	MIPv6	PMIPv6	IPWAVE VMM	Our VMM
Handover Detection	Host-based	Network-based	Network-based	Host-based
IPv6 Generation	SLLAC	SLLAC	SLLAC/128-bit assignment	SLLAC
IID Generation	link-layer address	link-layer address	link-layer address	Stable IID
Prefix	Exclusive	Exclusive	Shared	Shared
Valid period of Ipv6	per link	per vehicle	per shared-prefix domain	per shared-prefix domain
Packet Forwarding	Tunneling	Tunneling	Tunneling	Address conversion

equal to the number of links or the number of MNs are needed. In contrast, shared prefix is used in IPWAVE VMM and Our VMM scheme. so prefixes equal to the number of vehicular links are needed.

The valid period of IPv6 address is also different. In the case of MIPv6, as a vehicle's address is only valid within a link with that prefix, the vehicle recreate an IPv6 address every time it moves its attachment. In PMIPv6, a prefix is assigned to each vehicle, so once an IPv6 address is generated, it is valid until the vehicle leaves the PMIPv6 domain. In IPWAVE VMM and our scheme, once a vehicle creates an IPv6 address, it is valid within the vehicular link. In addition, in our scheme, the generated IPv6 address is also valid until active communication exists using that address.

The last difference is packet-forwarding manner. PMIPv6 and IPWAVE VMM use tunneling and packet encapsulation to forward packets, while our VMM scheme uses address conversion. The processing time of tunneling takes more grater than address conversion because tunneling has to encapsulate and decapsulate packets. Also, the bandwidth consumed when tunneling is used is more than when using address translation because the packet-size is increased with tunneling.

VII. SECURITY CONSIDERATION

There are several security considerations of our scheme as follows:

- If the integrity of a UDP-ND message is not guaranteed, a man-in-middle attacker can maliciously change the field of UDP-ND message. If an attacker maliciously modifies the value of the target IPv6 address field of a UDP-NS message, the wrong vehicle's IPv6 address may be registered to LBR. In addition, an attacker can also maliciously change the address of the RSU to which the vehicle currently belongs by modifying the source address of the UDP-NS message. In this case, because the vehicle's current RSU is set to the wrong RSU, the traffic is destined for the wrong RSU and the vehicle cannot receive IPv6 data traffic.
- If a sender of a UDP-ND message is not authenticated, an attacker can send a fake UDP-ND message. If an attacker periodically sends fake UDP-NS registration messages containing randomly generated IPv6 addresses, it causes the exhaustion of 64-bit IID space of the vehicular link. An attacker can also maliciously change the RSU to which a vehicle belongs by sending fake UDP-NS handover messages.
- An attacker can collect UDP-NS handover messages sent by the vehicle and replay the messages to LBR later. If the message is replayed, an attacker can change the current RSU of a vehicle using previous UDP-NS handover messages.

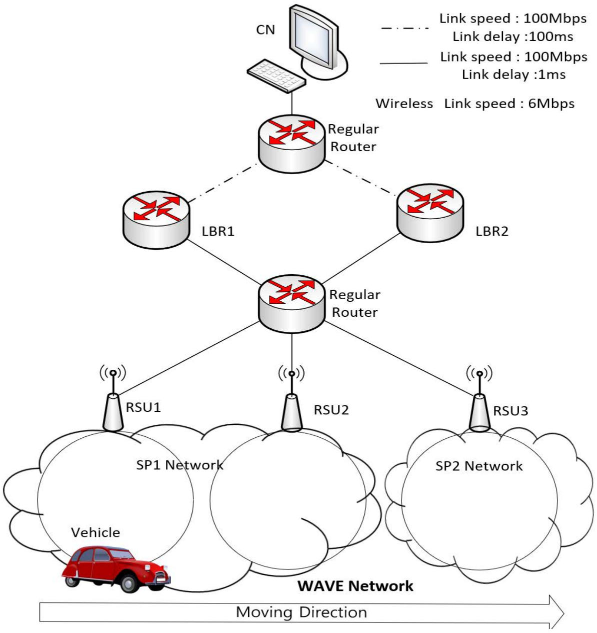


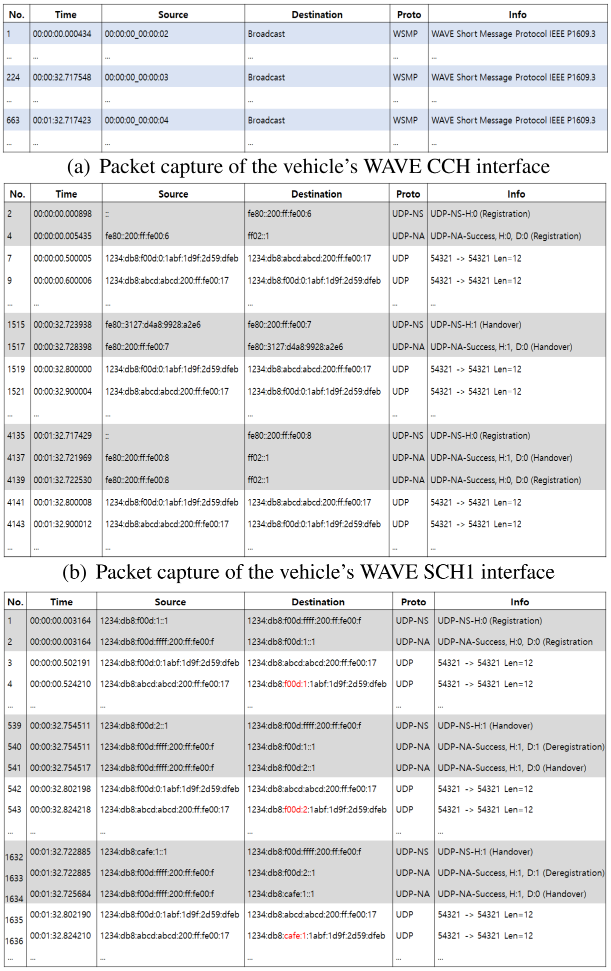
FIGURE 9. Simulation topology.

To handle these considerations, the scheme may use the security service defined in IEEE standard 1609.2 [22]. It authenticates a vehicle and a RSU, and guarantees the integrity of UDP-ND message contents. Also, IPsec [23] with pre-configured security associations may be used between a RSU and a LBR. By the pre-configured IPsec, a RSU and a LBR can authenticate each other and guarantee the integrity of the source IPv6 address.

VIII. SIMULATION

We use the ns-3 network simulator (version 3.32 [24]) and its WAVE model library to simulate the proposed vehicular mobility management scheme.

Fig.9 shows the network topology used in the simulation. The vehicle has two WAVE interfaces and moves from RSU1



No.	Time	Source	Destination	Proto	Info
1	00:00:00.000434	00:00:00_00:00:02	Broadcast	WSMP	WAVE Short Message Protocol IEEE P1609.3
...	...	...	...	...	...
224	00:00:32.717548	00:00:00_00:00:03	Broadcast	WSMP	WAVE Short Message Protocol IEEE P1609.3
...	...	...	...	...	...
663	00:01:32.717423	00:00:00_00:00:04	Broadcast	WSMP	WAVE Short Message Protocol IEEE P1609.3
...	...	...	...	...	...

(a) Packet capture of the vehicle’s WAVE CCH interface

No.	Time	Source	Destination	Proto	Info
2	00:00:00.000898	::	fe80::200:ff:fe00:6	UDP-NS	UDP-NS-H:0 (Registration)
4	00:00:00.005435	fe80::200:ff:fe00:6	ff02::1	UDP-NA	UDP-NA-Success, H:0, D:0 (Registration)
7	00:00:00.500005	1234:db8:f00d:0:1abf:1d9f:2d59:dfeb	1234:db8:abcd:abcd:200:ff:fe00:17	UDP	54321 -> 54321 Len=12
9	00:00:00.600006	1234:db8:abcd:abcd:200:ff:fe00:17	1234:db8:f00d:0:1abf:1d9f:2d59:dfeb	UDP	54321 -> 54321 Len=12
...	...	...	...	...	...
1515	00:00:32.723938	fe80::3127:d4a8:9928:a2e6	fe80::200:ff:fe00:7	UDP-NS	UDP-NS-H:1 (Handover)
1517	00:00:32.728398	fe80::200:ff:fe00:7	fe80::3127:d4a8:9928:a2e6	UDP-NA	UDP-NA-Success, H:1, D:0 (Handover)
1519	00:00:32.800000	1234:db8:f00d:0:1abf:1d9f:2d59:dfeb	1234:db8:abcd:abcd:200:ff:fe00:17	UDP	54321 -> 54321 Len=12
1521	00:00:32.900004	1234:db8:abcd:abcd:200:ff:fe00:17	1234:db8:f00d:0:1abf:1d9f:2d59:dfeb	UDP	54321 -> 54321 Len=12
...	...	...	...	...	...
4135	00:01:32.717429	::	fe80::200:ff:fe00:8	UDP-NS	UDP-NS-H:0 (Registration)
4137	00:01:32.721969	fe80::200:ff:fe00:8	ff02::1	UDP-NA	UDP-NA-Success, H:1, D:0 (Handover)
4139	00:01:32.722530	fe80::200:ff:fe00:8	ff02::1	UDP-NA	UDP-NA-Success, H:0, D:0 (Registration)
4141	00:01:32.800008	1234:db8:f00d:0:1abf:1d9f:2d59:dfeb	1234:db8:abcd:abcd:200:ff:fe00:17	UDP	54321 -> 54321 Len=12
4143	00:01:32.900012	1234:db8:abcd:abcd:200:ff:fe00:17	1234:db8:f00d:0:1abf:1d9f:2d59:dfeb	UDP	54321 -> 54321 Len=12
...	...	...	...	...	...

(b) Packet capture of the vehicle’s WAVE SCH1 interface

No.	Time	Source	Destination	Proto	Info
1	00:00:00.003164	1234:db8:f00d:1::1	1234:db8:f00d:ffff:200:ff:fe00:f	UDP-NS	UDP-NS-H:0 (Registration)
2	00:00:00.003164	1234:db8:f00d:ffff:200:ff:fe00:f	1234:db8:f00d:1::1	UDP-NA	UDP-NA-Success, H:0, D:0 (Registration
3	00:00:00.502191	1234:db8:f00d:0:1abf:1d9f:2d59:dfeb	1234:db8:abcd:abcd:200:ff:fe00:17	UDP	54321 -> 54321 Len=12
4	00:00:00.524210	1234:db8:abcd:abcd:200:ff:fe00:17	1234:db8:f00d:1:1abf:1d9f:2d59:dfeb	UDP	54321 -> 54321 Len=12
...	...	...	...	...	...
539	00:00:32.754511	1234:db8:f00d:2::1	1234:db8:f00d:ffff:200:ff:fe00:f	UDP-NS	UDP-NS-H:1 (Handover)
540	00:00:32.754511	1234:db8:f00d:ffff:200:ff:fe00:f	1234:db8:f00d:1::1	UDP-NA	UDP-NA-Success, H:1, D:1 (Deregistration)
541	00:00:32.754517	1234:db8:f00d:ffff:200:ff:fe00:f	1234:db8:f00d:2::1	UDP-NA	UDP-NA-Success, H:1, D:0 (Handover)
542	00:00:32.802198	1234:db8:f00d:0:1abf:1d9f:2d59:dfeb	1234:db8:abcd:abcd:200:ff:fe00:17	UDP	54321 -> 54321 Len=12
543	00:00:32.824218	1234:db8:abcd:abcd:200:ff:fe00:17	1234:db8:f00d:2:1abf:1d9f:2d59:dfeb	UDP	54321 -> 54321 Len=12
...	...	...	...	...	...
1632	00:01:32.722885	1234:db8:cafe:1::1	1234:db8:f00d:ffff:200:ff:fe00:f	UDP-NS	UDP-NS-H:1 (Handover)
1633	00:01:32.722885	1234:db8:f00d:ffff:200:ff:fe00:f	1234:db8:f00d:2::1	UDP-NA	UDP-NA-Success, H:1, D:1 (Deregistration)
1634	00:01:32.725684	1234:db8:f00d:ffff:200:ff:fe00:f	1234:db8:cafe:1::1	UDP-NA	UDP-NA-Success, H:1, D:0 (Handover)
1635	00:01:32.802190	1234:db8:f00d:0:1abf:1d9f:2d59:dfeb	1234:db8:abcd:abcd:200:ff:fe00:17	UDP	54321 -> 54321 Len=12
1636	00:01:32.824210	1234:db8:abcd:abcd:200:ff:fe00:17	1234:db8:cafe:1:1abf:1d9f:2d59:dfeb	UDP	54321 -> 54321 Len=12
...	...	...	...	...	...

(c) Packet capture of the LBR1’s interface which is connected with the regular router in the WAVE access network

FIGURE 10. **Wireshark’s captured packet.**

to RSU3 at a constant velocity (60km/s). One WAVE interface is tuned to the Control Channel (CCH) and the other is tuned to the Service Channel one (SCH1). A RSU advertises “IPv6 routing” service in its WSA, by including Provider Service Identifier (PSID) (0 × 10–20–40–7E) [25]. It broadcasts the WSA at a rate of 10 times per 1 sec. RSU1 and RSU2 belong to the same vehicular link managed by LBR1, and RSU3 belongs to another vehicular link managed by LBR2. When a vehicle moves from RSU1 to RSU2, intra-link mobility management (section V-C1) is applied. When the vehicle moves from RSU2 to RSU3, inter-link mobility management (section V-C2) is applied.

In the simulation, we set the maximum coverage of each RSU to 400 meters because the unobstructed Vehicular Ad-hoc Network’s (VANET’s) reliable communication range is measured to be under 370 meters according to [28]. The deployment of RSUs can be divided into disjoint case and overlapped case. RSUs stand 1000 meters or 780 meters apart for another RSU, respectively.

There are two regular IPv6 routers (RRs) connecting the components in the simulated networks. One is used to connect all RSUs and LBRs in the WAVE access network, and the other is used to connect LBRs and a CN in the Internet. If two vehicular links are partitioned in the WAVE access network, communication between the two vehicular links is done via the Internet.

To aid the analysis of the proposed UDP-ND datagram, we have programmed the decoding module of the proposed message using Lua language [27] for Wireshark [26]. In Wireshark’s captured packet screen, a proposed ND message encapsulated in an UDP packet is marked an UDP-NS/UDP-NA. We also filter out irrelevant packets such as 802.11 Acknowledgement (ACK) to aid visibility.

Fig.10(a) shows packet trace at the vehicle’s WAVE CCH interface and Fig.10(b) shows packet trace at the vehicle’s WAVE SCH1 interface. WSA messages are received via CCH and the proposed UDP-ND messages and UDP data packets are received via SCH1.

When a vehicle first enters RSU1’s coverage, it receives WSA messages sent by RSU1 (Packet #1 in Fig.10(a)). It performs the network attachment procedure (section V-A). An UDP-NS (Packet #2 in Fig.10(b)) is sent for IPv6 address registration. It uses an unspecified IPv6 address (::) as the source address, and contains multiple NS messages.

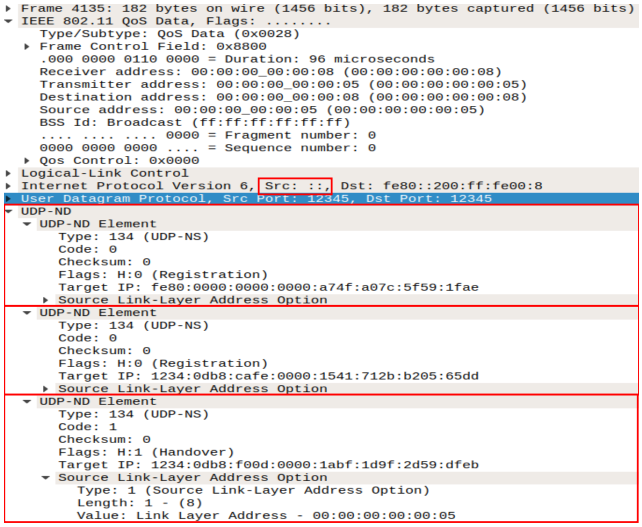
The vehicle receives an UDP-NA datagram from RSU1 (Packet #4 in Fig.10(b)). Note that the destination address of the UDP-NA datagram is all-node multicast address, but its destination link-layer address uses the link-layer of the vehicle, instead of the IPv6 mapped multicast link-layer address. All NS messages in the UDP-NA datagram indicate success of address registration by their code values. The code and flags in the first NA message of the UDP-NA datagram are shown in Wireshark’s capture screen.

From now on, the vehicle can commence UDP exchanges with a CN. The vehicle sends UDP packets to the CN at a rate of 500 Kbps and 10 packets per sec. First two UDP packets exchanged with the CN are shown in Packets #7 (uplink), #9 (downlink) in Fig.10(b).

Fig.10(c) shows packet capture at LBR1’s interface connected to a regular router in the WAVE access network. When an UDP packet destined to the vehicle is arrived at LBR1, it replaces the prefix of the destination IPv6 address of the packet with the prefix of RSU1 founded in its LBR-RVL. Packet #4 in Fig.10(c) shows the operation result.

When the vehicle receives a new WSA sent by RSU2 (Packet #224 in Fig.10(a)), it decides to perform intra-link mobility management. It sends an UDP-NS datagram as a handover notification to RSU2 (Packet #1515 in Fig.10(b)). Note that the source IPv6 address of the UDP-NS datagram uses the vehicle’s link-local address.

When LBR1 receives this UDP-NS datagram (Packet #539 in Fig.10(c)), it creates two UDP-NA datagrams (Packets #540 and #541 in Fig.10(c)). The unsolicited UDP-NA with ‘H’ and ‘D’ flag set (Packet #540) is sent to the previous RSU (RSU1) to notify the de-registration of the vehicle and another UDP-NA with ‘H’ flag set (Packet #541) is sent to



```
▸ Frame 4135: 182 bytes on wire (1456 bits), 182 bytes captured (1456 bits)
▾ IEEE 802.11 QoS Data, Flags: ........
    Type/Subtype: QoS Data (0x0028)
  ▸ Frame Control Field: 0x8800
    .000 0000 0110 0000 = Duration: 96 microseconds
    Receiver address: 00:00:00_00:00:08 (00:00:00:00:00:08)
    Transmitter address: 00:00:00_00:00:05 (00:00:00:00:00:05)
    Destination address: 00:00:00_00:00:08 (00:00:00:00:00:08)
    Source address: 00:00:00_00:00:05 (00:00:00:00:00:05)
    BSS Id: Broadcast (ff:ff:ff:ff:ff:ff)
    .... .... .... 0000 = Fragment number: 0
    0000 0000 0000 .... = Sequence number: 0
  ▸ Qos Control: 0x0000
▸ Logical-Link Control
▸ Internet Protocol Version 6, Src: ::, Dst: fe80::200:ff:fe00:8
▸ User Datagram Protocol, Src Port: 12345, Dst Port: 12345
▾ UDP-ND
  ▾ UDP-ND Element
      Type: 134 (UDP-NS)
      Code: 0
      Checksum: 0
      Flags: H:0 (Registration)
      Target IP: fe80:0000:0000:0000:a74f:a07c:5f59:1fae
    ▸ Source Link-Layer Address Option
  ▾ UDP-ND Element
      Type: 134 (UDP-NS)
      Code: 0
      Checksum: 0
      Flags: H:0 (Registration)
      Target IP: 1234:0db8:cafe:0000:1541:712b:b205:65dd
    ▸ Source Link-Layer Address Option
  ▾ UDP-ND Element
      Type: 134 (UDP-NS)
      Code: 1
      Checksum: 0
      Flags: H:1 (Handover)
      Target IP: 1234:0db8:f00d:0000:1abf:1d9f:2d59:dfeb
    ▾ Source Link-Layer Address Option
        Type: 1 (Source Link-Layer Address Option)
        Length: 1 - (8)
        Value: Link Layer Address - 00:00:00:00:00:05
```

FIGURE 11. UDP-NS message (Packet #4135 in Fig.10(b).

the new serving RSU (RSU2) to notify the result of intra-link handover.

When RSU1 receives the unsolicited UDP-NA datagram with 'D' flag set, it removes its RSU-RVL entry of the vehicle.

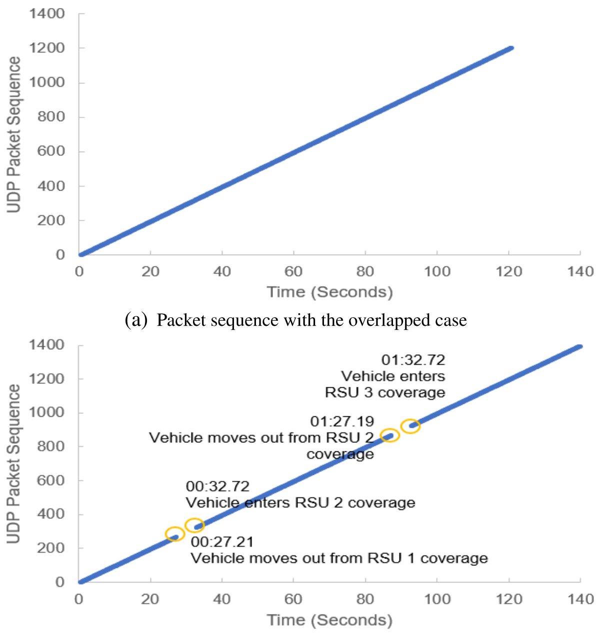
When RSU2 receives the UDP-NA datagram, it process the datagram and forwards it to the vehicle. The vehicle recognizes the success of its intra-link handover by receiving Packet #1517 in Fig.10(b).

At this moment, the UDP packet flow between the vehicle and the CN is changed. As a result, LBR1 changes the prefix of the destination IPv6 address of UDP destined to the vehicle to the prefix of RSU2 in its LBR-RVL. Packet #543 in Fig.10(c) shows this result.

When the vehicle receives a new WSA sent by RSU3 (Packet #663 in Fig.10(a)), it decides to perform inter-link handover for a previous vehicular link as well as the network attachment procedure for a new vehicular link. It sends an UDP-NS datagram (Packet #4135 in Fig.10(b) to notify its attachment and inter-link handover to RSU3. The UDP-NS datagram is shown in Fig.11. For the new vehicular link, as the vehicle has no valid IPv6 address yet, it uses an unspecified IPv6 address (::) as the source IPv6 address of the UDP-NS datagram. It also shows the three NS messages contained in the UDP-NS datagram. The first and second NS messages register a new link-local address for the new vehicular link and a new public IPv6 address derived from the shared prefix of the new vehicular link, respectively. The third NS message notifies inter-link handover for a public IPv6 address derived from the shared prefix of the previous vehicular link.

When RSU3 receives this UDP-NS datagram, it generates two UDP-NS datagrams. One is used for inter-link handover with LBR1, and the other is for the network attachment procedure with LBR2.

When LBR1 receives inter-link handover notification with packet #1632 in Fig.10(c), it generates two UDP-NA datagrams as same as intra-link handover (Packets #1633 and #1634 10(c)).



(a) Packet sequence with the overlapped case

(b) Packet sequence with the disjoint case

FIGURE 12. Packet sequence with each case.

The vehicle receives two UDP-NA datagrams from RSU3 (Packets #4137 and #4139 in Fig.10(b)). One is sent by LBR1 as the result of inter-link handover, and the other is sent by LBR2 as the result of the new network attachment.

At this moment, LBR1 changes the UDP packet flow between the CN and the vehicle. As a result, LBR1 changes the prefix of the destination IPv6 address of UDP destined to the vehicle to the prefix of RSU3 in its LBR-RVL (Packet #1636 in Fig.10(c).

Fig.12(a) and 12(b) shows the sequence of received UDP packets at the vehicle as it moves from RSU1 to RSU3. In all cases, the vehicle receives packets without any loss before handover occurs. But in disjoint case (Fig.12(b)), there are some of packet loss because RSU coverage does not overlap each other.

IX. CONCLUSION AND FUTURE WORKS

In this paper, we propose a vehicular mobility management scheme for a share-prefix model over IEEE WAVE IPv6 networks. The proposed scheme introduces VMM-NDP. VMM-NDP is used for both the network attachment and handover procedures. This scheme supports both intra-link mobility management and inter-link mobility management. It also supports multiple shared prefixes for DMM.

In VMM proposed by the IEFT IPWAVE working group, there were problems because it has missed some assumptions of PMIPv6. As one of them, in PMIPv6, a MN-ID and a profile must be obtained during the network attachment, however, such information cannot be obtained because there is no authentication procedure in the IEEE WAVE networks.

Accordingly, in the proposed scheme, there is no need to obtain the MN-ID and the profile because the vehicle can know its own movement and the shared-prefix model is used.

Additionally, the scheme does not use link-layer address for its IPv6 address generation, so it supports pseudonym. Also, since it does not use the link-layer multicasting at all, it improves wireless communication efficiency.

In the future, we will make efforts to alleviate packet loss problem during handover. We also plan to apply route optimization to our proposed scheme.

REFERENCES

[1] E. Kline, *IP Wireless Access in Vehicular Environments*, charter-ietf-ipwave-01, IETF, Fremont, CA, USA, Mar. 2020.

[2] J. Jeong, Y. Shen, Z. Xian, and S. Cespedes, *Vehicular Neighbor Discovery for IP-Based Vehicular Networks*, draft-jeong-ipwave-vehicular-neighbor-discovery-13, IETF, Fremont, CA, USA, Feb. 2022.

[3] T. Narten, E. Nordmark, W. Simpson, and H. Soliman, *Neighbor Discovery for IP Version 6 (IPv6)*, document RFC 4861, IETF, Fremont, CA, USA, Sep. 2007.

[4] E. Baccelli and M. Townsley, Eds., *IP Addressing Model in Ad Hoc Networks*, document RFC 5889, IETF, Fremont, CA, USA, Sep. 2010.

[5] S. Gundavelli, K. Leung, V. Devarapalli, K. Chowdhury, and B. Patil, Eds., *Proxy Mobile IPv6*, document RFC 5213, IETF, Fremont, CA, USA, Aug. 2008.

[6] J. Jeong, B. Mugabarigira, Y. Shen, and Z. Xiang, *Vehicular Mobility Management for IP-Based Vehicular Networks*, draft-jeong-ipwave-vehicular-mobility-management-07, IETF, Fremont, CA, USA, Feb. 2022.

[7] H. Chan, D. Liu, P. Seite, H. Yokota, and J. Korhonen, Eds., *Requirements for Distributed Mobility Management*, document RFC 7333, IETF, Fremont, CA, USA, Aug. 2014.

[8] A. DeKok, *The Network Access Identifier*, document RFC 7542, IETF, Fremont, CA, USA, May 2015.

[9] N. Benamar, J. Härri, J. Lee, and T. Ernst, *Basic Support for IPv6 Networks Operating Outside the Context of a Basic Service Set Over IEEE Std 802.11*, document RFC 8691, IETF, Fremont, CA, USA, Dec. 2019.

[10] S. Thomson, T. Narten, and T. Jinmei, *IPv6 Stateless Address Autoconfiguration*, document RFC 4862, IEFT, Fremont, CA, USA, Sep. 2007.

[11] *IEEE Standard for Wireless Access in Vehicular Environments (WAVE)—Networking*, IEEE Standard 1609.3-2020, Mar. 2021.

[12] S. Cespedes, N. Lu, and X. Shen, "VIP-WAVE: On the feasibility of IP communications in 802.11p vehicular networks," *IEEE Trans. Intell. Transp. Syst.*, vol. 14, no. 1, pp. 82–97, Mar. 2013.

[13] C.-C. Huang-Fu, Y.-B. Lin, and N. Alrajeh, "Mobility management of unicast services for wireless access in vehicular environments," *IEEE Wireless Commun.*, vol. 19, no. 2, pp. 88–95, Apr. 2012.

[14] T. Bellache, S. Kallel, O. Shagdar, and S. Tohme, "GeoMIP: A novel mobility management solution for internet and VANET communication using geographic partition in mobile IP," Presented at the Wireless Days (WD), Dubai, United Arab Emirates, Apr. 2018.

[15] K. S. Atwal, A. Guleria, and M. Bassiouni, "SDN-based mobility management and QoS support for vehicular ad-hoc networks," Presented at the Int. Conf. Comput., Netw. Commun. (ICNC), Maui, HI, USA, Mar. 2018.

[16] Z. He, B. Fu, A. Cao, and J. Yu, "A solution for mobility management in software defined VANET," Presented at the IEEE 15th Int. Conf. Mobile Ad Hoc Sensor Syst. (MASS), Chengdu, China, Oct. 2018.

[17] N. Moore, *Optimistic Duplicate Address Detection (DAD) for IPv6*, document RFC 4429, IETF, Fremont, CA, USA, Apr. 2006.

[18] F. Gont, A. Cooper, D. Thaler, and W. Liu, *Recommendation on Stable IPv6 Interface Identifiers*, document RFC 8064, IEFT, Fremont, CA, USA, Feb. 2017.

[19] F. Gont, *A Method for Generating Semantically Opaque Interface Identifiers With IPv6 Stateless Address Autoconfiguration (SLAAC)*, document RFC 7217, IEFT, Fremont, CA, USA, Apr. 2014.

[20] M. Crawford, *Transmission of IPv6 Packets Over Ethernet Networks*, document RFC 2464, IEFT, Fremont, CA, USA, Dec. 1998.

[21] S. Gundavelli, M. Townsley, O. Troan, and W. Dec, *Address Mapping of IPv6 Multicast Packets on Ethernet*, document RFC 6085, IEFT, Fremont, CA, USA, Jan. 2011.

[22] *IEEE Standard for Wireless Access in Vehicular Environments (WAVE)—Certificate Management Interfaces for End Entities*, IEEE Standard 1609.2.1-2022, Mar. 2022.

[23] S. Frankel and S. Krishnan, *IP Security (IPsec) and Internet Key Exchange (IKE) Document Roadmap*, document RFC 6071, IEFT, Fremont, CA, USA, Feb. 2011.

[24] (Mar. 10, 2022). *The ns-3 Website*. [Online]. Available: https://www.nsnam.org/releases/ns-3-32/

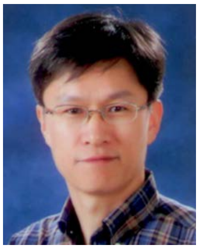
[25] *IEEE Standard for Wireless Access in Vehicular Environments (WAVE)—Identifier Allocations*, IEEE Standard, 1609.12-2016, Mar. 2016.

[26] (Mar. 10, 2022). *The Wireshark Website*. [Online]. Available: https://www.wireshark.org/

[27] (Mar. 10, 2022). *The Lua Website*. [Online]. Available: https://www.lua.org/

[28] R. Meireles, M. Boban, P. Steenkiste, O. Tonguz, and J. Barros, "Experimental study on the impact of vehicular obstructions in VANETs," in *Proc. IEEE Vehicle Netw. Conf. (VNC)*, Dec. 2010, pp. 338–345.

MYEONGJI KO received the B.S. degree in computer science and engineering from Hankyong National University, South Korea, in 2018. She is currently pursuing the Ph.D. degree in computer science and engineering with Korea University, Seoul, South Korea. Her research interests include future internet, vehicle ad hoc networks, mobility protocol design, network architectures, and performance analysis.



HYOGON KIM (Member, IEEE) received the B.S. and M.S. degrees in computer engineering from Seoul National University, Seoul, South Korea, in 1987 and 1989, respectively, and the Ph.D. degree in computer and information science from the University of Pennsylvania, in 1995. From 1996 to 1999, he was a Research Scientist at Bell Communications Research (Bellcore). He is currently a Professor at Korea University. His research interests include wireless communication, vehicular networking, the Internet of Things (IoT), and mobile computing.

SUNG-GI MIN received the B.S. degree in computer science from Korea University, Seoul, South Korea, in 1988, and the M.S. and Ph.D. degrees in computer science from the University of London, in 1989 and 1993, respectively. From January 1994 to February 2000, he worked with the LG Information and Communication Research Center. From March 2000 to February 2001, he was a Professor with the Department of Computer Engineering, Dongeui University, Busan, South Korea. Since March 2001, he has been a Professor with the Department of Computer Science and Engineering, Korea University. His research interests include wired/wireless communication networks, mobility protocols, network architectures, QoS, and mobility management in future networks.