

# Routing Management for DTN Networks in VANET

Moath Muayad Al-Doori, Francois Siewe, and Ali Hilal Al-Bayatti

**Abstract**— Routing protocols in VANET are considered as one of the critical dilemmas that need to be tackled, especially in sparse environment. Thus designing an efficient routing mechanism has an impact on enhancing the network performance in terms of disseminating messages to their desired destinations. This paper proposes a novel routing protocol in VANET for sparse environment called Vehicle Second Heading Direction Routing Protocol (VSHDRP), which is designed to leverage the probability of delivering a data packet to its destination and to increase connectivity and route stability by utilizing the knowledge of the Second Heading Direction (SHD) in the process of selecting the next-hop node. This new routing protocol contains two modes; the highway straight mode and the roundabout\intersection mode. Moreover, the two modes of VSHDRP protocol are formalized in the Calculus of Context-aware Ambients (CCA) and simulated using the CCA interpreter ccaPL in order to analyse and validate the behaviour of the protocol.

**Index Terms**—VANET, Routing Protocols, DTN, SHD, VSHDRP, CCA

## I. INTRODUCTION

Vehicle Ad hoc Networks (VANET) is a specific state of Mobile Ad hoc Networks (MANET) [1, 2]. Communication in VANET can be accomplished via one of two main alternatives: Vehicle to Infrastructures (V2I) and Vehicle to Vehicle (V2V), which use a Dedicated Short Range Communication (DSRC) method between either nearby vehicle or roadside equipment facilities. The DSRC is based on IEEE 1609 standards of the Wireless Access in Vehicular Environments (WAVE) family [3, 4]. Traffic congestion caused by vehicles accidents is considered to be a vital concern on the roads. Therefore safety applications are the focus of most researchers in VANET; increasing the efficiency of these applications has a vital impact on their contribution to limiting the number of fatal accidents and providing safer, cleaner and more comfortable travelling on roads. Vehicle drivers have no ability to predict the conditions ahead of time [5, 6]; with the aid of sensors, ubiquitous computing and wireless communication devices, this combination of equipped devices assists in providing more capabilities to vehicles on the roads to foresee hazard (e.g. the speed of other vehicles). In that way, warning messages could be sent every 0.5 second to predict vehicle speed in order to eliminate the occurrence of accidents [7].

Manuscript received October 5, 2011, revised October 20 2011.

M. M Al-Doori is a PhD candidate in the Software Technology Research Laboratory (STRL), Faculty of Technology, De Montfort University, Leicester, UK (e-mail: maldoori@dmu.ac.uk).

F. Siewe is a Senior Research with the Software Technology Research Laboratory (STRL) at Faculty of Technology, De Montfort University, Leicester, UK (e-mail : fsiewe@dmu.ac.uk)

A. H. Al-Bayatti is Research Fellow with the Software Technology Research Laboratory (STRL) at Faculty of Technology, De Montfort University, Leicester, UK (e-mail: alihmohd@dmu.ac.uk)

Many routing techniques have been designed in MANET to tackle the limitations of the transmission packet delivery delay, packets being dropped, wasting bandwidth, mobility and security. These techniques could not be fitted on VANET owing to the particular characteristics of VANET, (e.g. restricted mobility pattern). As a result of the above-mentioned argument, it is anticipated that designing an efficient routing protocol will aid in accomplishing the task of delivering packets to their destinations in a more realistic method.

This paper proposes a novel mechanism in VANET to mitigate these limitations, by using the *Second Heading Direction* (SHD) besides other parameters already used in other proposed routing mechanisms such as node's position, current direction and speed. Indeed, the SHD provides information about whether a node will change its direction ahead, and this knowledge is utilized to select the best next-hop node with respect to network connectivity and route stability. The contributions of this paper are summarized as follows:

- We propose a novel notion of SHD in routing protocol in VANET to increase the packet delivery ratio, route stability and connectivity (Sect. IV-B).
- We propose a novel routing algorithm for highway (straight mode), called VSHDRP1, based on the filtration process, which comprises four stages: position, direction, SHD and speed (Sect. IV-C).
- We propose a novel routing algorithm for roundabout mode, called VSHDRP2, based on the aspect of SHD (sect. VII).
- The new two modes protocol is fully formalized in the Calculus of Context-aware Ambients (CCA) (Sect. V and sect. VIII).
- We analyze the behaviours of the protocol using the CCA interpreter (Sect. VI and Sect. IX).

## II. THE CALCULUS OF CONTEXT-AWARE AMBIENTS (CCA)

CCA is a process calculus for modelling mobile and context-aware systems. The main features of the calculus include *mobility*, *context-awareness* and *concurrency*. The concept of *ambient* is an abstraction of a place where computation may happen. An ambient can be mobile and can contain other ambients called child ambients organised in a hierarchical structure. Such a hierarchy can be used to model any entity in a pervasive system (whether physical, logical, mobile or immobile) as well as the environment (or context) of that entity. In addition to child ambients, an ambient can also contain a process specifying the capabilities of that ambient, i.e. the actions the ambient is allowed to perform, such as mobility capabilities, context-aware capabilities and communication capabilities. Moreover, specifications written in CCA are executable. All these features have motivated our choice of CCA for the

specification and analysis of the Vehicular Second Heading Direction Routing Protocol (VSHDRP) [8].

We refer interested readers to [9] for the full details of the calculus. Table I depicts the syntax of CCA, based on three syntactic categories: processes (denoted by  $P$  or  $Q$ ), capabilities (denoted by  $M$ ) and context-expressions (denoted by  $E$ ).

TABLE I: SYNTAX OF CCA

$P, Q ::= 0 \mid P \mid Q \mid (v n) P \mid !P \mid n[P] \mid \{P\} \mid E?M.P \mid \mathbf{find} \tilde{x} : E \mathbf{for} P$
$M ::= \mathbf{in} n \mid \mathbf{out} \mid \alpha \mathbf{recv}(\tilde{y}) \mid \alpha \mathbf{send}(\tilde{y}) \mid \mathbf{del} n$
$\alpha ::= \uparrow \mid n \uparrow \mid \downarrow \mid n \downarrow \mid :: \mid n :: \mid \epsilon$
$E ::= \mathbf{True} \mid \bullet \mid n = m \mid \neg E \mid E_1 \mid E_2 \mid E_1 \wedge E_2 \mid \oplus E \mid \diamond E$

**Processes.** The process 0, aka inactivity *process*, does nothing and terminates immediately. The process  $P/Q$  denotes the process  $P$  and the process  $Q$  running in parallel. The process  $(v n) P$  states that the scope of the name  $n$  is limited to the process  $P$ . The replication  $!P$  denotes a process which can always create a new copy of  $P$ , i.e.  $!P$  is equivalent to  $P \mid !P$ . Replication can be used to implement both iteration and recursion. The process  $n[P]$  denotes an ambient named  $n$  whose behaviours are described by the process  $P$ . The pair of square brackets ‘[’ and ‘]’ outlines the boundary of that ambient. The process  $\{P\}$  behaves exactly like the process  $P$ , so the pair of curly brackets ‘{’ and ‘}’ are used simply as parentheses.

A context expression  $E$  specifies the condition that must be met by the environment of the executing process. A *context-guarded prefix*  $E?M.P$  is a process that waits until the environment satisfies the context expression  $E$ , then performs the capability  $M$  and continues like the process  $P$ . The dot symbol ‘.’ denotes the sequential composition of processes. We let  $M.P$  denote the process  $\mathbf{True}?M.P$ , where  $\mathbf{True}$  is a context expression satisfied by all context.

**Capabilities.** Ambients exchange messages using the output capability  $\alpha \mathbf{send}(\tilde{z})$  to send a list of names  $\tilde{z}$  to a location  $\alpha$ , and the input capability  $\alpha \mathbf{recv}(\tilde{y})$  to receive a list of names from a location  $\alpha$ . The location  $\alpha$  can be ‘ $\uparrow$ ’ for any parent, ‘ $n \uparrow$ ’ for a specific parent  $n$ , ‘ $\downarrow$ ’ for any child, ‘ $n \downarrow$ ’ for a specific child  $n$ , ‘ $::$ ’ for any sibling, ‘ $n ::$ ’ for a specific sibling  $n$ , or  $\epsilon$  (empty string) for the executing ambient itself.

The mobility capabilities  $\mathbf{in}$  and  $\mathbf{out}$  are defined as follows. An ambient that performs the capability  $\mathbf{in} n$  moves into the sibling ambient  $n$ . The capability  $\mathbf{out} n$  moves the ambient that performs it out of that ambient's parent.

The capability  $\mathbf{del} n$  deletes an ambient of the form  $n[0]$  situated at the same level as that capability.

**Context model.** In CCA, a context is modelled as a process with a hole in it. The hole (denoted by  $\odot$ ) in a context represents the position of the process that context is the context of. For example, suppose a system is modelled by the process  $P \mid n[Q \mid m[R \mid S]]$ . So, the context of the process  $R$  in that system is  $P \mid n[Q \mid m[\odot \mid S]]$ , and that of ambient named  $m$  is  $P \mid n[Q \mid \odot]$ . Thus the contexts of CCA processes are described by the grammar in Table II.

TABLE II: SYNTAX OF CONTEXTS

$$C ::= 0 \mid \odot \mid n[C] \mid C \mid P \mid (v n)C$$

Properties of contexts are called context expressions (CEs in short).

**Context expressions.** The CE  $\mathbf{True}$  always holds. A CE  $n=m$  holds if the names  $n$  and  $m$  are lexically identical. The CE  $\bullet$  holds solely for the hole context, i.e. the position of the process evaluating that context expression. Propositional operators such as negation ( $\neg$ ) and conjunction ( $\wedge$ ) expand their usual semantics to context expressions.

A CE  $E_1 \mid E_2$  holds for a context if that context is a parallel composition of two contexts such that  $E_1$  holds for one and  $E_2$  holds for the other. A CE  $n[E]$  holds for a context if that context is an ambient named  $n$  such that  $E$  holds inside that ambient. A CE  $\oplus E$  holds for a context if that context has a child context for which  $E$  holds. A CE  $\diamond E$  holds for a context if there exists somewhere in that context a sub-context for which  $E$  holds. The operator  $\diamond$  is called *somewhere modality* while  $\oplus$  is aka *spatial next modality*.

### III. THE VSHDRP1 MODEL

This section shows the assumption, conception and the mechanism of the VSHDRP:

#### A. Assumptions

VSHDRP1 works under the following assumptions: the transmission range of each vehicle in the network is up to 250m, and each vehicle has sufficient knowledge about its surrounding neighbours through exchanging a HELLO beacon message periodically, i.e. vehicle *id*, its position, direction and speed. We assume in this proposed protocol that each vehicle is supplied with a *GPS* device and navigation system (*NS*), and vehicles are equipped with preloaded digital road maps; therefore we assume that each vehicle can know its own location, direction through the fitted *GPS* device and *NS*, and can predetermine its route to its destination from the beginning.

#### B. VSHDRP1 Conception

The mode of approach proposed in this section is developed for highway straight scenario. This paper takes the advantage of utilising the *GPS* system, *NS* and digital road map, so that each vehicle can acquired the knowledge of its position, direction and can establish a predetermined route; meanwhile each vehicle can provide its surrounding neighbours with all these information by broadcasting a periodic HELLO beacon message. The new proposed protocol aims to initiate a robust and long-lived route between source node of the data packet and its destination. The main idea of the VSHDRP is to take the (SHD) into account in the process of selecting the next-hop node, which has an impact on making the packet route more stable.

For example, if we assume that source node  $S$  is travelling on a highway and is intending to deliver a packet to a destination node  $D$  located at the end of the road, according to the above assumptions the forwarding vehicle

(which holds the packet) can select the next-hop vehicle (node) based on its SHD. The next-hop node may have an SHD that leads the vehicle to the next exit (if we know that the route packet will continue ahead). In this situation, the forwarding vehicle will ignore this vehicle and will look for another vehicle that has an SHD, which will not drive the vehicle to the next exit. In other words, the forwarding vehicle will select the next-hop vehicle that continues on the same road as the packet route without making any diversion. Thus the probability of delivering messages to their destinations will increase.

This proposed work mitigates the Delay Tolerant Network (DTN) issues because it is based mainly on the carry-and forward strategy. Indeed, when a disconnected area appears to split between the vehicle that holds the packet and other vehicles moving toward the packet's destination, there is no opportunity to forward the packet to a next-hop node. Having information about the position of the source node and destination is insufficient to avoid disconnected areas.

Therefore, using the SHD is vital in deciding the next hop node. Suppose a forwarding node in a DTN Network needs to forward the packet to the next-hop node which has the position and direction towards the destination region, without knowing the SHD of the next-hop. This next-hop node might take another route if it has a SHD that leads it to another route before it reaches its destination. The SHD can take two values, either 0 or 1; SHD = 0 means that the vehicle will not divert its route in the next exit or intersection, which makes it a candidate node; while SHD = 1 represents the condition that the vehicle will divert its route in the next exit or intersection. In this situation, this node is not suitable for delivering the packet; as a result, the forwarding node will need to re-forward the packet, and that will lead to a reduction in the bandwidth consumption, delay in the packet transmission time and the system stability and connectivity.

C. VSHDRP1 Mechanism for straight mode (Highway Scenario)

The VSHDRP1 comprises three parts: the first one represents the general algorithm of the process of sending the packet, while the second part depicts the core of the VSHDRP, which is the filtration process for selecting the next-hop node; the third part represents the process of packet delivery confirmation between any two nodes.

1. Packet Sending Process: VSHDRP1 considers that each vehicle (node) in the network has sufficient knowledge about its own location, direction, speed and SHD. As illustrated in Figure 1, when a source node *S* needs to send a packet to a destination node *D*, it will look for node *D* in its cache (neighbours table), and if the node *D* is found as a neighbour in its cache, node *S* will start forwarding the data packets to node *D*. If *D* is not found in the cache of source node *S*, then node *S* will set the current direction and the SHD, and then it will start to look for an appropriate next-hop node by using the filtration process for selecting the next-hop node, which comprises four main stages: position stage, current direction stage, SHD stage and speed stage

respectively. The output of the filtration process can be either *Yes* or *No*.

- a. *Yes*: means an appropriate next-hop node is found from the neighbour nodes. Therefore the source node will forward the packet to this node.
- b. *No*: means that no appropriate next-hop node is found from the surrounding neighbour nodes. For that reason, the source node will keep the packet in its buffer and continue listening for new neighbours to become available. If a new neighbour node is found, the source node will run the same procedure until it delivers the packet to its destination.

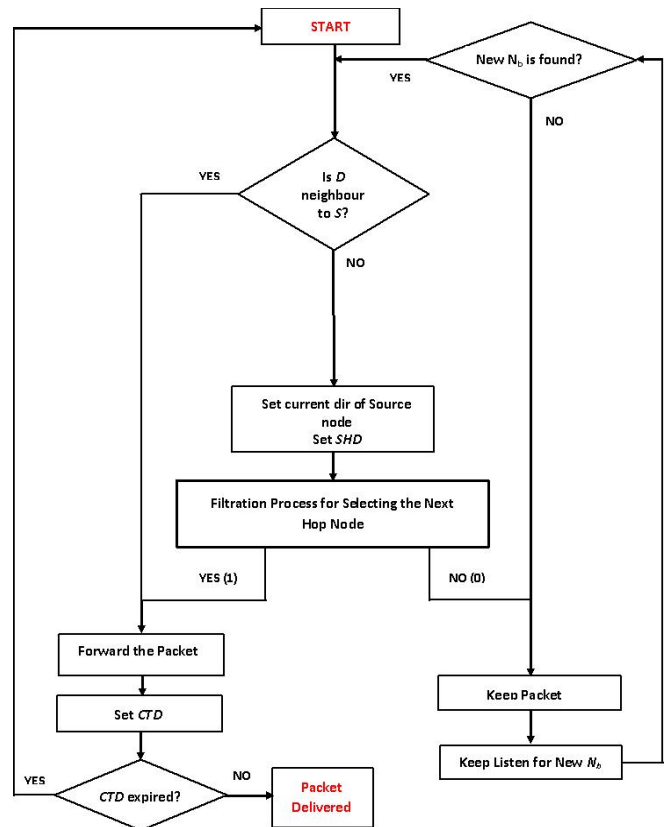


Fig. 1. The Algorithm of VSHDRP

2. Filtration process for selecting the next hop node: there are four stages for selecting the next-hop node.
  - a. Position Knowledge Stage: in this stage, the packet's holder node will select neighbour nodes that have a closer position to the packet's destination than itself.
  - b. Current Direction Knowledge Stage: in this stage, the selected nodes in the previous stage will be processed in order to check if they have an appropriate current direction (i.e. the direction towards the packet's destination).
  - c. Second Heading Direction Knowledge Stage (SHD): this stage nominates the candidates' nodes in the previous stage according to their SHD.
  - d. Speed knowledge stage: this stage will select the node with the highest speed in case more than one candidate is available.

As shown in Figure 2, the filtration process states that, if the destination node  $D$  is not found in the source node's cache, then the source node  $S$  will start the filtration process to look for an appropriate next-hop node.  $S$  will start the filtration process by looking for neighbours with a closer destination to  $D$  than itself. If no neighbours' nodes ( $Nb$ ) are found, it will buffer the packet; otherwise it will check if any of  $Nb$  nodes has a current direction towards the packet's destination. If an appropriate node is found then it will check to see if its  $SHD = 0$ ; if more than one  $Nb$  node is found, then the  $Nb$  with highest speed will be selected. If  $SHD = 1$  for all  $Nb$ , then  $S$  will check if  $SHD = 0$  for a neighbour of neighbour ( $Nb$  of  $Nb$ ) and if more than one  $Nb$  of  $Nb$  are found, then the  $Nb$  of  $Nb$  with the highest speed will be selected. If the current direction for all  $Nb$  is opposite to the packet's destination, then  $S$  will check the  $Nb$  of  $Nb$  that have current direction towards the packet's destination, and repeat the same procedure for  $SHD$  and speed.

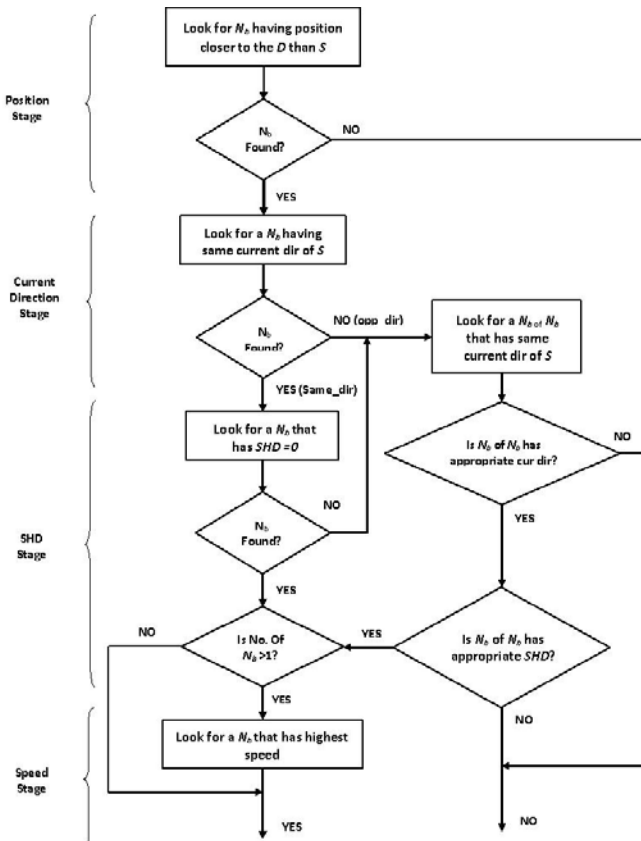


Fig. 2. Filtration Process Algorithm for Selecting Next-Hop Node

3. Packet Delivery Confirmation ( $CTD$ ): when a node intends to forward the packet to another node, it needs to make sure that the packet is delivered successfully to that node; therefore it is important in this stage to send back an acknowledgment to the sending process. Thus a time counter needs to be established at the same moment of sending the packet; this counter is called Confirmation Time Duration ( $CTD$ ). When the forwarding node wants to send the packet, it sets the  $CTD$  to a specified timeout value, and the value of this counter will be decremented; if the confirmation is received before the  $CTD$  has elapsed, then the  $CTD$  will be halted. Otherwise, if the time is elapsed and no

confirmation message (*acknowledgment*) has been received, that means the packet has been dropped or discarded for some reason; therefore the sender node will look for an alternative node and resend the packet.

#### IV. FORMAL SPECIFICATION OF VSHDRP1

We now give a formal specification of the VSHDRP protocol in CCA in a compositional manner.

##### A. System Model

VANET is modelled in CCA as a parallel composition of all the nodes in the network (e.g. vehicles and road side units), i.e

$$VANET \cong node_0 \mid node_1 \mid \dots \mid node_{k-1} \quad (1)$$

Each node,  $node_i$ , in the VANET is modelled as an ambient of the following structure:

$$id[P_{id} \mid NH1 [P_1] \mid NH2 [P_2] \mid Hspeed [P_3] \mid Lspeed [P_4] \mid SHD [P_5] ]$$

where

- $id$  is the node's id. For the sake of simplicity, we use  $SN$  to denote the source node,  $DN$  for the destination node and  $IN_j$  for intermediate nodes,  $j \geq 0$ .
- $P_{id}$  is a process that specifies the capabilities of the node, e.g. its ability to communicate or to sense the presence of other nodes in its range.
- $NH1$  is an ambient that contains the ids of the neighbouring node moving in the direction of the destination node (aka direction 1) and closer to the destination node.
- $NH2$  is an ambient that contains the ids of the neighbouring node moving in the direction opposite to direction 1 (this is called direction 2) and closer to the destination node.
- $Hspeed$  is an ambient that contains the ids of the neighbouring nodes moving in high speed (i.e. speed greater than or equal to a specified threshold).
- $Lspeed$  is an ambient that contains the ids of the neighbouring nodes moving in low speed (i.e. speed less than the threshold).
- $SHD$  is an ambient that contains the ids of the neighbouring nodes with Second Heading Direction (SHD).
- Each process  $P_j$ ,  $1 \leq j \leq 5$ , is either the inactivity process  $\mathbf{0}$  or a parallel composition of ambients of the form  $n[\mathbf{0}]$  where  $n$  is a node's id.

##### B. Context Expression

As explained in Section II, a context expression is a predicate that states the condition the environment of the executing process must meet.

The context expressions used in the specification of the VSHDRP protocol are summarised as follows:

- $hasNb1(n)$  holds if the node  $n$  is a neighbour closer to destination in direction 1, i.e.  
 $hasNb1(n) \cong \diamond (\bullet \mid NH1[\mathbf{True} \mid n[\mathbf{True}]] \mid \mathbf{True})$
- $hasNb2(n)$  holds if the node  $n$  is a neighbour

closer to destination in direction 2, i.e.

$$hasNb2(n) \equiv \diamond (\bullet \mid NH2[\mathbf{True} \mid n[\mathbf{True}]] \mid \mathbf{True})$$

- $hasNb(n)$  holds if the node  $n$  is a neighbour closer to destination (regardless its direction), i.e.  
 $hasNb(n) \equiv hasNb1(n) \vee hasNb2(n)$

- $noNb1()$  holds if there are no neighbours in direction 1, i.e.

$$noNb1() \equiv \neg \diamond (\bullet \mid NH1[\mathbf{True} \mid \oplus \mathbf{True}] \mid \mathbf{True})$$

- $hasSHD(n)$  holds if the node  $n$  is a neighbour with second heading direction (SHD), i.e.

$$hasSHD(n) \equiv \diamond (\bullet \mid SHD[\mathbf{True} \mid n[\mathbf{True}]] \mid \mathbf{True})$$

- $noSHD()$  holds if there are no neighbours with SHD

$$noSHD(n) \equiv \neg \diamond (\bullet \mid SHD[\mathbf{True} \mid \oplus \mathbf{True}] \mid \mathbf{True})$$

- $hasNbofNb1(n)$  holds if node  $n$  is a neighbour in direction 2 that has a neighbour closer to destination and moving in direction 1

$$hasNbofNb1(n) \equiv hasNb2$$

$$\wedge \diamond (n[\mathbf{True} \mid NH1[\mathbf{True} \mid \oplus \mathbf{True}]] \mid \mathbf{True})$$

- $highestSpeed(n)$  holds if the node  $n$  has the highest speed:

$$highestSpeed(n) \equiv \diamond (\bullet \mid Hspeed[\mathbf{True} \mid n[\mathbf{True}]] \vee (\neg \diamond (\bullet \mid Hspeed[\mathbf{True} \mid \oplus \mathbf{True}] \mid \mathbf{True}) \wedge \diamond (\bullet \mid Lspeed[\mathbf{True} \mid n[\mathbf{True}]] \mid \mathbf{True}))$$

We now specify the behaviours proper to each type of node (source node, intermediate node and destination node).

A source node  $SN$  is the node that initiates a run of the VSHDRP protocol, willing to send a message  $msg$  to a destination node  $DN$ . So its capabilities are modelled by the following process:

$$P_{SN} \equiv (v n) \{ (3) \mid (4) \mid (5) \mid x[\uparrow \mathbf{send}().0] \} \quad (2)$$

where the restricted name  $x$  is used to guarantee that not more than one of the processes Eq3, Eq4, Eq5 are executed. Indeed,  $x$  is an ambient that sends a single signal to its parent; this signal will be captured by exactly one of these processes, eventually. These processes are specified as follows:

- if the destination node is a neighbour, send message to destination node and wait for acknowledgement. This is formalised as:

$$hasNb(DN)?x \downarrow \mathbf{recv}().DN :: \mathbf{send}(SN, msg).DN :: \mathbf{recv}(y).0 \quad (3)$$

- if the destination node is not a neighbour then look for an intermediate node moving in direction 1 with SHD and highest speed, and send the message to that intermediate node and wait for acknowledgement, viz.

$$\mathbf{find} n : E1(n, DN) \text{ for } x \downarrow \mathbf{recv}().n :: \mathbf{send}(SN, DN, msg).n :: \mathbf{recv}(y).0 \quad (4)$$

where:  $E1(s, t) \equiv \neg hasNb(t) \wedge hasNb(s) \wedge hasSHD(s)$

$\wedge highestSpeed(s)$

- if no such intermediate nodes then look for an intermediate node moving in direction 2 which has a neighbour in direction 1 closer to the destination node, and send the message to that intermediate node and wait for acknowledgement, viz.

$$\mathbf{find} n : E2(n, DN) \text{ for } x \downarrow \mathbf{recv}().n :: \mathbf{send}(SN, DN, msg).n :: \mathbf{recv}(y).0 \quad (5)$$

where:

$$E2(s, t) \equiv \neg hasNb(t) \wedge (noNb1() \vee noSHD()) \wedge hasNbofNb1(s)$$

An intermediate node  $IN$  receives a triple ( $sender, dest, msg$ ) where  $sender$  is the sender's  $id$ ,  $dest$  is the destination node's  $id$  and  $msg$  is the message being sent. The intermediate node confirms the receipt by sending an acknowledgement to the sender and forwards the message to an appropriate node. This is specified as:

$$P_{IN} \equiv ! :: \mathbf{recv}(sender, dest, msg). \mathbf{send} :: \mathbf{send}(ack).(v x) \{ (7) \mid (8) \mid (9) \mid x[\uparrow \mathbf{send}().0] \} \quad (6)$$

where the restricted name  $x$  plays the same role as in (2) for selecting at most one of the processes (7), (8) and (9). The replication operator ' $!$ ' means that an intermediate node repeats this pattern of behaviour its whole lifetime. Moreover, an intermediate node determines the next node to forward the message to as follows:

- if the destination node  $dest$  is a neighbour, send message to destination node and wait for acknowledgement, i.e.

$$hasNb(dest)?x \downarrow \mathbf{recv}().dest :: \mathbf{send}(IN, msg).dest :: \mathbf{recv}(y).0 \quad (7)$$

- if the destination node  $dest$  is not a neighbour then look for another intermediate node moving in direction1 with SHD and highest speed, and send the message to that intermediate and wait for acknowledgement, viz.

$$\mathbf{find} n : E1(n, dest) \text{ for } x \downarrow \mathbf{recv}().n :: \mathbf{send}(IN, dest, msg).n :: \mathbf{recv}(y).0 \quad (8)$$

- if no such intermediate nodes then look for another intermediate node moving in direction 2 which has a neighbour in direction 1 closer to the destination node, and send the message to that intermediate and wait for acknowledgement, viz.

$$\mathbf{find} n : E2(n, dest) \text{ for } x \downarrow \mathbf{recv}().n :: \mathbf{send}(IN, dest, msg).n :: \mathbf{recv}(y).0 \quad (9)$$

The destination node  $DN$  receives a pair ( $sender; msg$ ) and sends an acknowledgement to the sender. This behaviour is formalised as:

$$P_{DN} \equiv :: \mathbf{recv}(sender, msg).sender :: \mathbf{send}(ack).0 \quad (10)$$

This formal specification of the VSHDRP protocol is executable by the CCA interpreter (ccaPL) and is used in the following section to simulate runs of the protocol.

## V. BEHAVIOURAL VALIDATION AND ANALYSIS FOR VSHDRP1

In this section, we run the formal specification of the VSHDRP protocol presented above in a variety of scenarios reflecting differing configurations of the VANET network. In these scenarios, a source node SN is willing to send a "hello" message to a destination node DN. Other nodes of the network are named IN1, IN2 and so on. The result of each run of the protocol is a sequence of reductions showing the interactions that happened among the VANET's nodes. This output can then be analysed to check that properties of the protocol are preserved, at an early stage prior to implementation and deployment. For illustration, we consider three scenarios as follows.

### A. Scenario 1

The destination node is neighbour to the source node as shown in Figure 3. This scenario is set up by adding the id of the destination node in NH1 component of the source node and the id of the source node in the NH1 component of the destination node. The output of the run is depicted in Table \ref{tab:scenario1}, where line numbers are added to make it easy to refer to individual lines.

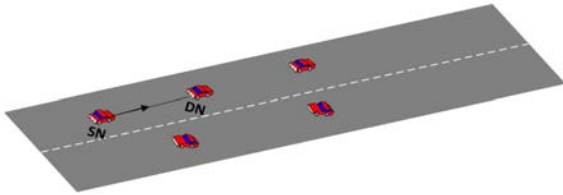


Fig. 3. Scenario 1

TABLE III: OUTPUT OF SCENARIO 1 OF VSHDRP1

1. $\leftarrow$	{renaming of a restricted name: $x$ to $x\$0$ }
2. $\rightarrow$	{Child to parent: $x\$0 == () \Rightarrow SN$ }
3. $\rightarrow$	{Sibling to sibling: $SN == (SN, hello) \Rightarrow DN$ }
4. $\rightarrow$	{Sibling to sibling: $DN == (ack) \Rightarrow SN$ }

The symbol ' $\leftarrow$ ' corresponds to the structural congruence relation which is an equivalence relation between processes, while ' $\rightarrow$ ' represents the reduction relation of CCA which determines a process transition as formally defined in [9]. The explanation of each transition is given between a pair of curly brackets. In particular, the notation ' $A == X \Rightarrow B$ ' means that an ambient ' $A$ ' sent a message ' $X$ ' to another ambient ' $B$ '. Other notations such as 'Child to parent' and 'Sibling to sibling' provide information about the relationship (w.r.t. ambient hierarchy) between the sender ' $A$ ' and the receiver ' $B$ '.

The line 1 in Table III corresponds to the execution of the restriction ( $v x$ ) in Eq. 2 which renames the name  $x$  to a new name  $x\$0$ . In line 2, the source node SN detects that the destination node DN is a neighbour and decides to execute the process in Eq. 3 by receiving a signal from the ambient  $x\$0$ . Then the "hello" message is sent directly to the

destination node (line 3) which acknowledges in line 4 and the run terminates.

### B. Scenario 2

In this scenario, the destination node is not a neighbour to the source node; rather the following neighbourhood relationship is considered:  $\{(SN, IN1), (IN1, IN2), (IN2, IN3), (IN3, DN)\}$  as shown in Figure 4. In addition, the node IN2 moves in direction 1, while IN1 and IN3 move in direction 2. Each of these nodes has a second heading direction (SHD). The output of the protocol's run is given in Table IV.

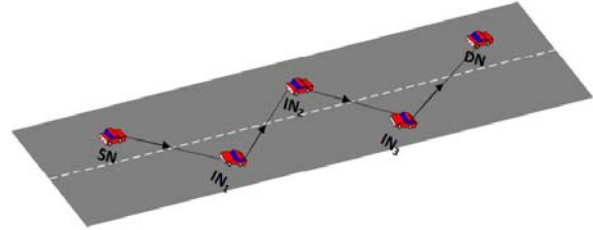


Fig. 4. Scenario 2

In line 2 the notation  $\{binding: n \rightarrow IN1\}$  corresponds to the execution of *search prefix* and means that the node IN1 is found (by the source node SN) as an appropriate node to forward the message to. The source node SN then forwards the message to the node IN1 in line 4. In a similar way, the message goes from IN1 to IN2 then to IN3 and finally to DN as showed in lines 9, 14 and 18 respectively.

TABLE IV: OUTPUT OF SCENARIO 2 OF VSHDRP1

1. $\leftarrow$	{renaming of a restricted name: $x$ to $x\$0$ }
2. $\rightarrow$	{binding: $n \rightarrow IN1$ }
3. $\rightarrow$	{Child to parent: $x\$0 == () \Rightarrow SN$ }
4. $\rightarrow$	{Sibling to sibling: $SN == (SN, DN, hello) \Rightarrow IN1$ }
5. $\rightarrow$	{Sibling to sibling: $IN1 == (ack) \Rightarrow SN$ }
6. $\leftarrow$	{renaming of a restricted name: $x$ to $x\$11$ }
7. $\rightarrow$	{binding: $n \rightarrow IN2$ }
8. $\rightarrow$	{Child to parent: $x\$11 == () \Rightarrow IN1$ }
9. $\rightarrow$	{Sibling to sibling: $IN1 == (IN1, DN, hello) \Rightarrow IN2$ }
10. $\rightarrow$	{Sibling to sibling: $IN2 == (ack) \Rightarrow IN1$ }
11. $\leftarrow$	{renaming of a restricted name: $x$ to $x\$20$ }
12. $\rightarrow$	{binding: $n \rightarrow IN3$ }
13. $\rightarrow$	{Child to parent: $x\$20 == () \Rightarrow IN2$ }
14. $\rightarrow$	{Sibling to sibling: $IN2 == (IN2, DN, hello) \Rightarrow IN3$ }
15. $\rightarrow$	{Sibling to sibling: $IN3 == (ack) \Rightarrow IN2$ }
16. $\leftarrow$	{renaming of a restricted name: $x$ to $x\$29$ }
17. $\rightarrow$	{Child to parent: $x\$29 == () \Rightarrow IN3$ }
18. $\rightarrow$	{Sibling to sibling: $IN3 == (IN3, DN, hello) \Rightarrow DN$ }
19. $\rightarrow$	{Sibling to sibling: $DN == (ack) \Rightarrow IN3$ }

## VI. VSHDRP2 MODEL

VSHDRP2 represents the second mode of the proposed approach; this mode is designed for roundabout/intersection scenario, where the location of destination node is unknown. As in the VSHDRP1, VSHDRP2 is working under the assumption that the type of the network is DTN (disconnected area can be generated any time when no vehicles are available at specific area) and each node in the

network has sufficient knowledge about its position and directions and predetermined route for its journey, by utilising the GPS and NS. As illustrated in flowchart in figure 1, when a source node  $S$  needs to send a packet to a destination node  $D$ , it will look for the destination node in its cache (neighbours table), and if the destination node is found as a neighbour in its cache,  $S$  will start forwarding the data packets to node  $D$ . If  $D$  is not found in the cache of source node  $S$ , then node  $S$  will set the current direction and the zones according to the SHD, then it will look for an appropriate next-hop node by using the filtration process for selecting the next-hop node.

- As shown in flowchart in Figure 5, if a neighbour nodes that have the desired position (closer to the destination than node  $S$ ) is found, then  $S$  will check for the current direction of these neighbours:
  - If the a neighbour node that has the same current direction (current road) as the  $S$  node is found (current direction = 0), then the  $S$  node will check if there is a roundabout ahead or an intersection by exploiting the knowledge of the digital road map, if one of these is found ahead, then it check the SHD (next road):
    - If a neighbour with an appropriate SHD is found by the  $S$  node in its cache, the  $S$  node forwards the packet to this node. If more than one node have an appropriate SHD ( $SHD = 1$ ), then the  $S$  node will select a node from each four zones (that we are classified in the beginning of the chapter). In case more than one appropriate node is available in each zone, the node with highest speed is chosen.
    - If no neighbour node that has an appropriate SHD can be found in its cache, in this case node  $S$  will look for neighbours of neighbours that have an appropriate SHD ( $SHD = 0$ ). And if more than one neighbour is found, the node with the highest speed will be chosen; otherwise it will keep holding the packet.
  - If no neighbour node that have the same current direction is found, which means current direction = 1, then node  $S$  will look for nodes that have the same current direction in its neighbours of neighbours table
    - If an appropriate node is found, then  $S$  will check for its SHD. If more than one appropriate node is found it will select one from each zone.
    - If no neighbour can be found, that means a disconnected area separates the source node from the other nodes in the network. Thus, the source node will buffer the packet until a new node appears in its cache of neighbour.
- If no neighbour nodes that have the appropriate position, then  $S$  node will keep holding the packet until a new node is available.

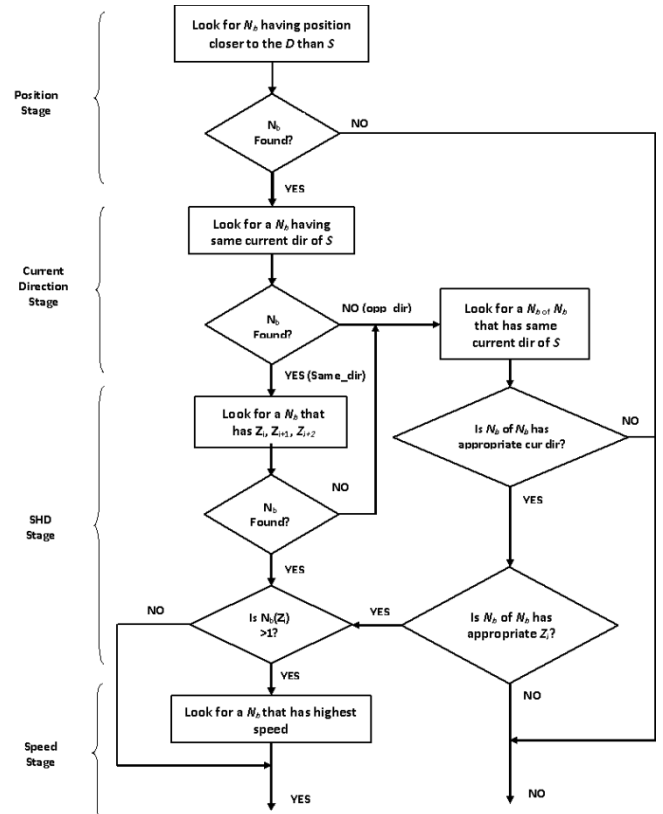


Fig. 5. Filtration process algorithm of VSHDRP2

## VII. FORMAL SPECIFICATION OF VSHDRP2

We now give a formal specification of the VSHDRP2 protocol in CCA.

### A. System Model

A VANET is modelled in CCA as a parallel composition of all the nodes in the network (e.g. vehicles and road side units), i.e.

$$VANET \cong node_0 \mid node_1 \mid \dots \mid node_{k-1}$$

Each node,  $node_i$ , in the VANET is modelled as an ambient of the following structure:

$$id[P_{id} \mid NH1[P_1] \mid NH2[P_2] \mid Hspeed_j[P_3] \mid Lspeed_j[P_4] \mid SHD_j[P_5]] \quad j = 1$$

where

- The index  $j=1, 2, 3, 4$  is the zone number.
- $id$  is the node's id. For the sake of simplicity, we use  $SN$  to denote the source node,  $DN$  for the destination node and  $IN_m$  for intermediate nodes,  $m \geq 0$ .
- $P_{id}$  is a process that specifies the capabilities of the node, e.g. its ability to communicate or to sense the presence of other nodes in its range.
- $NH1$  is an ambient that contains the ids of the neighbouring node moving in the direction of the destination node (aka direction 1) and closer to the destination node.
- $NH2$  is an ambient that contains the ids of the neighbouring node moving in the direction opposite to direction 1 (this is called direction 2) and closer to the destination node.
- $Hspeed_j$  is an ambient that contains the ids of the



neighbouring nodes moving in high speed (i.e. speed greater than or equal to a specified threshold) with second heading direction (SHD) in zone  $j$ ,  $j=1,2,3,4$ .

- $Lspeed_j$  is an ambient that contains the ids of the neighbouring nodes moving in low speed (i.e. speed less than the threshold) with SHD in zone  $j$ ,  $j=1,2,3,4$ .
- $SHD_j$  is an ambient that contains the ids of the neighbouring nodes with SHD in zone  $j$ ,  $j=1, 2, 3, 4$ .
- Each process  $P_m$ ,  $1 \leq m \leq 5$ , is either the inactivity process  $0$  or a parallel composition of ambients of the form where  $n$  is a node's id.

Each run of the VSHDRP2 protocol may involve up to three types of nodes: one source node, zero or many intermediate nodes and one destination node. Each node in a VANET should be able to act as a source node, an intermediate and a destination node.

### B. Context Expression

The context expressions used in the specification of the VSHDRP2 protocol are summarised as follows, where  $j=1, 2, 3, 4$  is the zone number:

- $hasNb1(n)$  holds if the node  $n$  is a neighbour closer to destination in direction 1, i.e.

$$hasNb1(n) \cong \diamond (\bullet | NH1 [True | n [True]] | True)$$

- $hasNb2(n)$  holds if the node  $n$  is a neighbour closer to destination in direction 2, i.e.

$$hasNb2(n) \cong \diamond (\bullet | NH2 [True | n [True]] | True)$$

- $hasNb(n)$  holds if the node  $n$  is a neighbour closer to destination (regardless its direction), i.e.

$$hasNb(n) \cong hasNb1(n) \vee hasNb2(n)$$

- $noNb1()$  holds if there are no neighbouring nodes in direction 1, i.e.

$$noNb1() \cong \neg \diamond (\bullet | NH1 [True | \oplus True] | True)$$

- $hasSHD_j(n)$  holds if the node  $n$  is a neighbour with second heading direction in zone  $j$ , i.e.

$$hasSHD_j(n) \cong \diamond (\bullet | SHD_j [True | \oplus True] | True)$$

- $noSHD_j()$  holds if there are no neighbours with SHD in zone  $j$ , i.e.

$$noSHD_j() \cong \neg \diamond (\bullet | SHD_j [True | \oplus True] | True)$$

- $hasNbofNb_j(n)$  holds if node  $n$  is a neighbour in direction 2 that has a neighbour closer to destination and moving in direction 1, i.e.

$$hasNbofNb_j(n) \cong hasNb2(n) \wedge \diamond (n [True | SHD_j [True | \oplus True]] | True)$$

- $highestSpeed_j(n)$  holds if the node  $n$  has the highest speed and a SHD in zone  $j$ , i.e.

$$highestSpeed_j(n) \cong \diamond (\bullet | Hspeed_j [True | n [True]] | True) \vee (\neg \diamond (\bullet | Hspeed_j [True | \oplus True] | True) \wedge \diamond (\bullet | Lspeed_j | True | n [True]] | True)$$

A source node  $SN$  is the node that initiates a run of the VSHDRP2 protocol, willing to send a message  $msg$  to a destination node  $DN$ . So its capabilities are modelled by the following process:

$$P_{SN} \cong (v z1)(v z2)(v z3)(v z4) \left\{ \begin{array}{l} Eq.12 | Eq.13 | Eq.14 \\ z1[\uparrow send().0] | z2[\uparrow send().0] \\ z3[\uparrow send().0] | z4[\uparrow send().0] \end{array} \right\} \quad (11)$$

where the restricted names  $z1$ ,  $z2$ ,  $z3$  and  $z4$  are used to guarantee that not more than one of the processes specified in Eq.12, Eq.13 and Eq.14 are executed, for each zone.

These processes are specified as follows:

- If the destination node is a neighbour, send message to destination node and wait for acknowledgement. This is formalised as:

$$hasNb(DN)? z1 \downarrow recv(). DN :: send(SN, msg). DN :: recv(y). 0 \quad (12)$$

- if the destination node is not a neighbour then for each zone  $j$ , look for an intermediate node moving in direction 1 with SHD in zone  $j$  and highest speed, and send the message to that intermediate node and wait for acknowledgement, viz.

$$\left\{ \begin{array}{l} \text{find } n : E_1^1(n, DN) \text{ for } z1 \downarrow recv(). n :: send(SN, DN, msg). n :: recv(y). 0 \\ \text{find } n : E_2^1(n, DN) \text{ for } z2 \downarrow recv(). n :: send(SN, DN, msg). n :: recv(y). 0 \\ \text{find } n : E_3^1(n, DN) \text{ for } z3 \downarrow recv(). n :: send(SN, DN, msg). n :: recv(y). 0 \\ \text{find } n : E_4^1(n, DN) \text{ for } z4 \downarrow recv(). n :: send(SN, DN, msg). n :: recv(y). 0 \end{array} \right\} \quad (13)$$

where:  $E_j^1(s, t) \cong \neg hasNb(t) \wedge hasNb1(s) \wedge hasSHD_j(s) \wedge highestSpeed_j(s)$ , for  $j = 1, 2, 3, 4$

- if no such intermediate nodes exist for a zone  $j$  then look for an intermediate node moving in direction 2 which has a neighbour with SHD in zone  $j$  moving in direction 1 and closer to the destination node, and send the message to that intermediate node and wait for acknowledgement, viz.

$$\left\{ \begin{array}{l} \text{find } n : E_1^2(n, DN) \text{ for } z1 \downarrow recv(). n :: send(SN, DN, msg). n :: recv(y). 0 \\ \text{find } n : E_2^2(n, DN) \text{ for } z2 \downarrow recv(). n :: send(SN, DN, msg). n :: recv(y). 0 \\ \text{find } n : E_3^2(n, DN) \text{ for } z3 \downarrow recv(). n :: send(SN, DN, msg). n :: recv(y). 0 \\ \text{find } n : E_4^2(n, DN) \text{ for } z4 \downarrow recv(). n :: send(SN, DN, msg). n :: recv(y). 0 \end{array} \right\} \quad (14)$$

where:  $E_j^2(s, t) \cong \neg hasNb(t) \wedge noSHD_j(s) \wedge hasNbofNb_j(s)$

An intermediate node  $IN$  receives a triple  $(sender, dest, msg)$  where  $sender$  is the sender's id,  $dest$  is the destination node's id and  $msg$  is the message being sent. The intermediate node confirms the receipt by sending an



acknowledgement to the sender and forwards the message to an appropriate node in each zone. This is specified as:

$$P_{IN} \hat{=} ! :: \mathbf{rcv}(sender, dest, msg).sender :: \mathbf{Send}(ack). \\ (v z1)(v z2)(v z3)(v z4) \{Eq. 16 | Eq. 17 | Eq. 18\} \\ z1[\uparrow \mathbf{send}().0] | z2[\uparrow \mathbf{send}().0] | z3[\uparrow \mathbf{send}().0] | z4[\uparrow \mathbf{send}().0] \} \quad (15)$$

Moreover, an intermediate node determines the next node to forward the message to as follows:

- if the destination node  $dest$  is a neighbour, send message to destination node and wait for acknowledgement, i.e.

$$hasNb(dest)? z1 \downarrow \mathbf{rcv}().dest :: \\ \mathbf{send}(IN, msg).dest :: \mathbf{rcv}(y).0 \quad (16)$$

- if the destination node  $dest$  is not a neighbour then for each zone  $j$ , look for another intermediate node moving in direction 1 with SHD in zone  $j$  and highest speed, and send the message to that intermediate and wait for acknowledgement, viz.

$$\left\{ \begin{array}{l} \mathbf{find} n : E_1^1(n, dest) \text{ for } z1 \downarrow \mathbf{rcv}().n :: \mathbf{send}(IN, dest, msg).n :: \mathbf{rcv}(y).0 \\ \mathbf{find} n : E_2^2(n, dest) \text{ for } z2 \downarrow \mathbf{rcv}().n :: \mathbf{send}(IN, dest, msg).n :: \mathbf{rcv}(y).0 \\ \mathbf{find} n : E_3^3(n, dest) \text{ for } z3 \downarrow \mathbf{rcv}().n :: \mathbf{send}(IN, dest, msg).n :: \mathbf{rcv}(y).0 \\ \mathbf{find} n : E_4^4(n, dest) \text{ for } z4 \downarrow \mathbf{rcv}().n :: \mathbf{send}(IN, dest, msg).n :: \mathbf{rcv}(y).0 \end{array} \right\} \quad (17)$$

- if no such intermediate nodes exist for a zone  $j$  then look for an intermediate node moving in direction 2 which has a neighbour with SHD in zone  $j$  moving in direction 1 and closer to the destination node, and send the message to that intermediate node and wait for acknowledgement, viz.

$$\left\{ \begin{array}{l} \mathbf{find} n : E_1^2(n, dest) \text{ for } z1 \downarrow \mathbf{rcv}().n :: \mathbf{send}(IN, dest, msg).n :: \mathbf{rcv}(y).0 \\ \mathbf{find} n : E_2^2(n, dest) \text{ for } z2 \downarrow \mathbf{rcv}().n :: \mathbf{send}(IN, dest, msg).n :: \mathbf{rcv}(y).0 \\ \mathbf{find} n : E_3^2(n, dest) \text{ for } z3 \downarrow \mathbf{rcv}().n :: \mathbf{send}(IN, dest, msg).n :: \mathbf{rcv}(y).0 \\ \mathbf{find} n : E_4^2(n, dest) \text{ for } z4 \downarrow \mathbf{rcv}().n :: \mathbf{send}(IN, dest, msg).n :: \mathbf{rcv}(y).0 \end{array} \right\} \quad (18)$$

The destination node  $DN$  receives a pair  $(sender, msg)$  and sends an acknowledgement to the sender. This behaviour is formalised as:

$$P_{DN} \hat{=} ! :: \mathbf{rcv}(sender, msg).sender :: \mathbf{send}(ack).0$$

### VIII. BEHAVIOUR VALIDATION AND ANALYSIS FOR VSHDRP2

The formal specification of the VSHDRP protocol presented above is executable by the CCA interpreter. In this section, we run the protocol in a variety of scenarios reflecting differing configurations of the VANET network. This output can then be analysed to detect flaws in the protocol at an early stage, prior to implementation and deployment. For illustration, we consider three scenarios as follows.

#### A. Scenario 1

In this scenario, the destination node is not a neighbour to the source node; rather the following neighbourhood relationship is considered:  $\{(SN, IN1), (SN, IN2), (SN, IN3), (IN3, DN)\}$ . In addition, the nodes  $SN, IN1, IN2$  and  $IN3$

move in direction 1;  $IN1$  has SHD in zone 1,  $IN2$  in zone 2 and  $IN3$  in zone 3. The output of the protocol's run is given in Table V.

TABLE V: OUTPUT OF SCENARIO 1 FOR VSHDRP2

1. ---> {binding: n → IN1}
2. ---> {binding: n → IN2}
3. ---> {binding: n → IN3}
4. ---> {Child to parent: x\$0 ==> () ==> SN}
5. ---> {Child to parent: y\$1 ==> () ==> SN}
6. ---> {Child to parent: z\$2 ==> () ==> SN}
7. ---> {Sibling to sibling: SN == (SN, DN, Packet) ==> IN1}
8. ---> {Sibling to sibling: IN1 == (ack) ==> SN}
9. ---> {Sibling to sibling: SN == (SN, DN, Packet) ==> IN2}
10. ---> {Sibling to sibling: IN2 == (ack) ==> SN}
11. ---> {Sibling to sibling: SN == (SN, DN, Packet) ==> IN3}
12. ---> {Sibling to sibling: IN3 == (ack) ==> SN}
13. ---> {Child to parent: x\$220 ==> () ==> IN3}
14. ---> {Child to parent: y\$221 ==> () ==> IN3}
15. ---> {Child to parent: z\$222 ==> () ==> IN3}
16. ---> {Sibling to sibling: IN3 == (IN3, packet) ==> DN}
17. ---> {Sibling to sibling: DN == (ack) ==> IN3}

In line 1, 2 and 3 in table VI the notation  $\{\text{binding: } n \rightarrow IN1, IN2 \text{ and } IN3\}$  corresponds to the execution of a *search prefix* and means that the node  $IN1, IN2$  and  $IN3$  are found (by the source node  $SN$ ) as an appropriate nodes to forward the message to. In line 4, 5 and 6, the source node  $SN$  detects that the Intermediate node  $IN1$  is a neighbour in zone1,  $IN2$  is a neighbour in zone2 and  $IN3$  is a neighbour in zone3 and decides to execute the process in Eq.12. Then the packet is sent directly to those nodes as shown in lin7, line9 and line11 respectively, which acknowledges in line 8, 10 and 12 respectively.

In line 13, the  $IN3$  detects that the destination node  $DN$  is a neighbour, after that  $IN3$  send the packet to  $DN$  as shown in line 16, which acknowledged in line 17 and the run terminates.

TABLE VI: OUTPUT OF SCENARIO 2 FOR VSHDRP2

1. ---> {binding: n → IN1}
2. ---> {binding: n → IN2}
3. ---> {binding: n → IN4}
4. ---> {Child to parent: x\$0 ==> () ==> SN}
5. ---> {Child to parent: y\$1 ==> () ==> SN}
6. ---> {Child to parent: z\$2 ==> () ==> SN}
7. ---> {Sibling to sibling: SN = (SN, DN, Packet) ==> IN1}
8. ---> {Sibling to sibling: IN1 == (ack) ==> SN}
9. ---> {Sibling to sibling: SN = (SN, DN, Packet) ==> IN2}
10. ---> {Sibling to sibling: IN2 == (ack) ==> SN}
11. ---> {Sibling to sibling: SN = (SN, DN, Packet) ==> IN4}
12. ---> {Sibling to sibling: IN4 == (ack) ==> SN}
13. ---> {binding: n → IN3}
14. ---> {Child to parent: z\$196 ==> () ==> IN4}
15. ---> {Sibling to sibling: IN4 == (IN4, DN, Packet) ==> IN3}
16. ---> {Sibling to sibling: IN3 == (ack) ==> IN4}
17. ---> {Child to parent: x\$175 ==> () ==> IN3}
18. ---> {Child to parent: y\$176 ==> () ==> IN3}
19. ---> {Child to parent: z\$177 ==> () ==> IN3}
20. ---> {Sibling to sibling: IN3 == (IN3, packet) ==> DN}
21. ---> {Sibling to sibling: DN == (ack) ==> IN3}

## B. Scenario 2

In this scenario, the destination node is not a neighbour to the source node; rather the following neighbourhood relationship is considered:  $\{(SN, IN1), (SN, IN2), (SN, IN4), (IN4, IN3), (IN3, DN)\}$ . In addition, the nodes SN, IN1, IN2 and IN3 move in direction 1, while IN4 moves in direction 2; IN1 has SHD in zone 1, IN2 in zone 2 and IN3 in zone 3. The output of the protocol's run is given in Table VI.

In line 1, 2 and 3 in table VII the notation  $\{\text{binding: } n \rightarrow IN1, IN2 \text{ and } IN4\}$  corresponds to the execution of a *search prefix* and means that the node IN1, IN2 and IN4 are found (by the source node SN) as an appropriate nodes to forward the message to. However node IN3 is direction 2 (opposite direction), but IN3 have a neighbour node that belong to zone3, which will receive the packet and forward it to the destination node DN as shown in line 20.

## IX. RELATED WORK

Many researchers have been concentrated on dealing with VANET by tackling its challenges, especially in designing a suitable routing protocol [10, 11] to overcome the problems of high node mobility and restricted movement (e.g. connectivity, latency and unnecessary overhead) [12, 13].

Zhao and Cao [14] introduced the Vehicle-Assisted Data Delivery Routing Protocol (VADD), which tackles the problems of packet delivery ratio, data packet latency and overhead. In addition, a Connectivity-Aware Routing protocol (CAR) [15] was initiated by Naumov and Gross to support the vehicular network in city and highway environments; this protocol focused on reducing overhead during message exchange.

Motion Vector Routing Algorithm [16] introduced by Lebrun and Chuah was designed to deliver a message from a vehicle to a static destination  $D$  in a sparse environment. It focuses on predicting which of its neighbouring vehicles will travel towards the fixed destination by utilising data from the knowledge of its neighbouring vehicles, such as velocity and trajectory. Furthermore, in this algorithm, the store-and-forward strategy is used to save and store data.

The authors of Delay-bounded Routing in Vehicular Ad-hoc Networks [17] introduced two algorithms, D-Greedy and DMinCost, in an attempt to deliver data from a moving node to a fixed infrastructure; the algorithms' environment is a densely urban area, so the authors concentrated on achieving ideal bandwidth consumption in terms of reducing the number of transmitted messages by exploiting traffic information, a store-and-forward technique is used in this technique.

## X. CONCLUSION

In this article we have proposed a novel routing protocol for Sparse environment in VANET which comprises two modes; the straight highway mode and the roundabout/intersection mode, this new protocol exploits the concept of SHD to leverage the probability of delivering

packets to their destination, increase link stability and reduce the number of failing transmissions; so as to maintain safety applications and reduce the risks associated with fatal accidents. Our protocol was formalised in CCA, then simulated using the interpreter of CCA through different scenarios designed to validate and evaluate the behaviour of the protocol. It followed that the protocol works correctly.

In future, the system performance will be examined using the Network Simulator-2 (NS-2), and its performance will be compared with other routing protocols in VANET.

## REFERENCES

- [1] W. Sun, H. Yamaguchi, K. Yukimasa, S. Kusumoto. GVGrid: A QoS Routing Protocol for Vehicular Ad Hoc Networks, *In Proceeding of 14th IEEE International Workshop on QoS*, 2006.
- [2] Y. Lin and Y. Chen and S. Lee, *Routing Protocols in Vehicular Ad Hoc Networks: A Survey and future perspectives*, Journal of Information Science and Engineering, 2010.
- [3] Festag, A. Hessler, R. Baldessari, L. Le, W. Zhang and D. Westho. *Vehicle-to-Vehicle and Road-Side Sensor Communication for Enhanced Road Safety*, 2009.
- [4] H. Hartenstein and K. P. Laberteaux. A Tutorial Survey on Vehicular Ad hoc Networks, *IEEE Communications Magazine*, 46(6), 164-171.
- [5] Ebner and H. Rohling. A Self-Organized Radio Network for Automotive Applications, *In Proceedings of the 8th World Congress on Intelligent Transportation Systems*, 2001.
- [6] M. Dikaiakos, A. Florides, T. Nadeem and L. Iftode, Location-Aware Services over Vehicular Ad-Hoc Networks using Car-to-Car Communication, *IEEE journal on selected areas in communications*, Vol. 25, No. 8, October 2007.
- [7] M. M. Al-Doori, A. H. Al-bayatti and H. Zedan. Context Aware Architecture for Sending Adaptive Hello Message in VANET, *In proceeding of Casemans 2010, the 4th ACM International Workshop on Context Awareness for Self-Managing Systems*, Copenhagen, Denmark, 26-29 Sep, 2010.
- [8] M. M. Al-Doori, F. Siewe and A. H. Al-Bayatti. VSHDRP: Vehicle Second Heading Directional Routing Protocol in VANET, *In proceeding of ICCSIT2011, The 4th IEEE conference, China, June 2011*.
- [9] F. Siewe, H. Zedan, and A. Cau. The Calculus of Context-aware Ambients. *Journal of Computer and System Science*, 77(4), 597-620, 2011.
- [10] M. Zhang and R. S. Wol. Routing Protocols for Vehicular Ad Hoc Networks in Rural Areas, *IEEE Communication Magazine*, 126-131, 2008.
- [11] T. Taleb, E. Sakhee, A. Jamalipour, K. Hashimoto, N. Kato and Y. Nemoto, *A Stable Routing Protocol to Support ITS Services in VANET Networks*, IEEE Transactions on Vehicular Technology, Vol. 56, No. 6, NOVEMBER 2007.
- [12] S. Jaap, M. Bechler and L. Wolf. Evaluation of Routing Protocols for Vehicular Ad Hoc Networks in City Traffic Scenarios, *In Proceedings of the 11th EUNICE Open European Summer School on Networked Applications*, Colmenarejo, 2005, 584-602.
- [13] W. Chen, R. K. Guha, T. J. Kwon, J. Lee and Y. Hsu, *A survey and challenges in routing and data dissemination in vehicular ad hoc networks*, Wiley InterScience, 2009.
- [14] J. Zhao and G. Cao, *VADD: Vehicle-Assisted Data Delivery in Vehicular Ad Hoc Networks*, IEEE Computer Communications, 2006.
- [15] V. Naumov and T. R. Gross. *Connectivity-Aware Routing (CAR) in Vehicular Ad Hoc Networks*, IEEE International Conference on Computer Communications (INFOCOM), 1919-1927, 2007.
- [16] J. Lebrun, C. Chuah, D. Ghosal and Michael Zhang. Knowledge-based opportunistic forwarding in vehicular wireless ad hoc networks, *IEEE Vehicular Tech. Conference*, 2005, 2289-2293.
- [17] Skordylis and N. Trigoni. Delay-Bounded Routing in Vehicular Ad-Hoc Networks, *ACM International Symposium on Mobile Ad hoc Networking and Computing*, pp.3017-3021, November 2008.